REGIONE EMILIA-ROMAGNA
Atti amministrativi

GIUNTA REGIONALE

Atto del Dirigente:	DETERMINAZIONE n° 4213 del 19/05/2009
Proposta:	DPG/2009/4810 del 15/05/2009
Struttura proponente:	DIREZIONE GENERALE CENTRALE ORGANIZZAZIONE, PERSONALE, SISTEMI INFORMATIVI E TELEMATICA
Oggetto:	LINEE GUIDA PER LA GOVERNANCE DEL SISTEMA INFORMATICO REGIONALE
Autorità emanante:	IL DIRETTORE - DIREZIONE GENERALE CENTRALE ORGANIZZAZIONE, PERSONALE, SISTEMI INFORMATIVI E TELEMATICA
Firmatario:	GAUDENZIO GARAVINI in qualità di Direttore generale
Luogo di adozione:	BOLOGNA data: 19/05/2009

DIREZIONE GENERALE CENTRALE ORGANIZZAZIONE, PERSONALE, SISTEMI INFORMATIVI E TELEMATICA

IL DIRETTORE

Premesso che:

La delibera della Giunta regionale n. **1057/2006** "*Prima fase di riordino delle strutture organizzative della giunta regionale. Indirizzi in merito alle modalità di integrazione interdirezionale e di gestione delle funzioni trasversali"*, delinea nell'Allegato D modifiche e integrazioni alla normativa in materia di relazioni organizzative e funzionali tra le strutture e di esercizio delle funzioni dirigenziali, correlate in particolare al tema della trasversalità;

la stessa delibera dispone la necessità di apportare alcune integrazioni alla delibera della Giunta regionale n. 447/2003 "*Indirizzi in ordine alle relazioni organizzative e funzionali tra le strutture e sull'esercizio delle funzioni dirigenziali*" prevedendo un'apposita sezione relativa alle responsabilità dirigenziali in materia di sistemi informativi e sicurezza dei dati e delle procedure informatiche che preveda il ruolo della Direzione generale centrale Organizzazione, personale, sistemi informativi e telematica in quanto a detta Direzione generale spettano:

- il presidio della coerenza dell'architettura del sistema informativo regionale assicurando l'unitarietà dell'impostazione delle funzioni tecniche settoriali incardinate nelle altre direzioni generali;
- lo sviluppo e la gestione delle infrastrutture e dei servizi di garanzia, la progettazione e la realizzazione dei progetti trasversali, gli standard generali di riferimento, l'assistenza tecnica e la collaborazione per lo sviluppo dei servizi e dei sistemi informativi settoriali e locali, anche su richiesta;
- l'individuazione degli standard tecnologici, metodologici e di sicurezza, le piattaforme tecnologiche a supporto dei sistemi, i criteri di selezione e assegnazione delle attrezzature informatiche;
- l'espressione di una valutazione preventiva, per le procedure e sistemi informativi sviluppati a cura delle Direzioni generali, di rispetto degli standard e piattaforme tecnologiche e di rispondenza ai criteri di qualità e sicurezza individuati per l'insieme del sistema informativo regionale;
- l'effettuazione, mediante modalità formalizzate ed omogenee, della presa in carico di sistemi informativi sviluppati a cura delle Direzioni generali e il loro passaggio in produzione, a seguito di una valutazione finale del rispetto dei criteri sopra indicati;

e di tali valutazioni deve essere dato formale riscontro negli atti amministrativi assunti dalle Direzioni generali in relazione alla realizzazione dei sistemi informativi, ivi inclusi gli atti di impegno e liquidazione della spesa;

Rilevato che:

la delibera della Giunta regionale n. 2416/2008 "*Indirizzi in ordine alle relazioni organizzative e funzionali tra le strutture e sull'esercizio delle funzioni dirigenziali. adempimenti*

conseguenti alla delibera 999/2008. adeguamento e aggiornamento della delibera 450/2007" stabilisce nell'Appendice 6 *"Sistemi informativi e sicurezza dei dati e delle procedure informatiche"* che la rilevanza strategica del sistema informativo regionale richiede che sia garantita l'unitarietà del sistema informativo nel suo complesso, attraverso l'omogenea applicazione di regole che riguardano tutto l'Ente e che, a tal fine, è necessario che l'esercizio di responsabilità ed autonomia da parte delle Direzioni generali sia accompagnato da azioni tecniche di verifica e audit, svolte dalla Direzione generale competente in materia di sistemi informativi/informatici e telematica, a garanzia della coerenza architetturale complessiva del sistema informativo regionale e dell'agevole svolgimento, da parte di quest'ultima, di eventuali successivi sviluppi o servizi complementari;

Considerato inoltre che la stessa Appendice 6 della delibera della Giunta regionale n. 2416/2008 stabilisce che la Direzione competente in materia di sistemi informativi / informatici e telematica svolge i compiti di programmazione, sviluppo e coordinamento generale previsti al comma 2 dell'art.16 della L.R. 11/2004 e rimanda ad apposite circolari da emanarsi a cura della Direzione generale competente in materia di sistemi informativi e telematica per la definizione delle modalità operative attuative di quanto disposto;

Rilevato inoltre che:

Negli ultimi anni infatti, l'avvio di numerosi progetti di e-government accompagnato dalla crescente necessità dei settori regionali di interoperare con altri soggetti istituzionali della società civile (Associazioni di categorie, Enti locali, Aziende sanitarie, Ministeri, UE, ecc.), ha sostanzialmente modificato sia in termini qualitativi che quantitativi, il sistema informativo-informatico regionale: da un sistema informativo progettato e realizzato per le specifiche esigenze funzionali dell'Ente Regione, si è progressivamente implementato un sistema informativo che eroga servizi anche a soggetti esterni che cooperano con la Regione;

tale crescita, purtroppo non sempre accompagnata da una visione strategica univocamente condivisa con tutte le strutture regionali, ha, in alcuni casi, comportato la proliferazione di soluzioni applicative eterogenee, governate più dall'offerta che non dalla domanda e tale disomogeneità di soluzioni ha comportato anche una ricaduta e un appesantimento delle infrastrutture informatiche dei nostri enti locali, costretti a uniformarsi alle diverse richieste delle Direzioni regionali: il rischio è che un sistema informativo a sottosistemi stagni all'interno della Regione-Ente induca sistemi informativi a sottosistemi stagni negli Enti della regione-territorio, con la creazione di sistemi magari integrati verticalmente tra diversi livelli territoriali ma sempre più non comunicanti.

Valutato quindi che in ottica di Community network questo appesantimento va risolto;

Considerato che in base alla DGR 999/2008, in particolare l'allegato D, le Direzioni generali trasmettono i relativi programmi di acquisizione alla Direzione generale centrale Organizzazione, personale, sistemi informativi e telematica alla quale spetta individuare possibili sinergie e la definizione di politiche di acquisto comune.

Richiamate quindi:

- La Legge 4/2004 "Disposizioni per favorire l'accesso ai soggetti disabili agli strumenti informatici";
- Il DM 8 luglio 2005 "Requisiti tecnici e i diversi livelli per l'accessibilità agli strumenti informatici"
- Il D. Lgs. 82/2005 "Codice dell'amministrazione digitale" e in particolare l'art. 53 "Caratteristiche dei siti";
- Il D.Lgs. 196/2003 "Codice in materia di protezione dei dati personali";
- la Legge regionale 11/2004 "Sviluppo regionale della Società dell'informazione";

- la delibera della Giunta regionale n. 1057/2006 "*Prima fase di riordino delle strutture organizzative della giunta regionale. Indirizzi in merito alle modalità di integrazione interdirezionale e di gestione delle funzioni trasversali" – Allegato D;*
- La delibera della Giunta regionale n. 1264/2005 "Linee guida della Giunta della Regione Emilia-Romagna in materia di protezione dei dati personali";
- La determinazione n.2653/2007 "Disciplinare tecnico per utenti per l'utilizzo dei sistemi informativi nella Giunta della Regione Emilia-Romagna";
- La determinazione n. 2651/2007 "Disciplinare tecnico in materia di sicurezza di applicazioni informatiche nella Giunta della Regione Emilia-Romagna";
- La determinazione n. 1416/2009 "Disciplinare Tecnico per Amministratori di sistema della Giunta e dell'Assemblea Legislativa";
- La delibera della Giunta regionale n. 2416/2008 "*Indirizzi in ordine alle relazioni organizzative e funzionali tra le strutture e sull'esercizio delle funzioni dirigenziali. adempimenti conseguenti alla delibera 999/2008. adeguamento e aggiornamento della delibera 450/2007"*- Appendice 6 *"Sistemi informativi e sicurezza dei dati e delle procedure informatiche";*
- La legge regionale 28/2007 "Disposizioni per l'acquisizione di beni e servizi";
- La delibera della Giunta regionale n. 999/2008 "Attività contrattuale. programmazione e riassetto organizzativo. procedura in economia in attuazione dell'art.10 l.r. n. 28 del 2007".

Vista inoltre la propria determinazione n 451/2008 "Linee guida per la *governance* del sistema informatico regionale" che definiva il carattere sperimentale degli standard metodologici e tecnologici di riferimento in esso contenuti.

Considerato che la sperimentazione, condotta per la durata di un anno, non ha evidenziato particolari criticità e che, a seguito dell'evoluzione normativa e tecnologica, sono state apportate alcune modifiche agli standard metodologici e tecnologici di riferimento già individuati.

Dato atto dei pareri allegati,

DETERMINA

a. di approvare, nel testo allegato quale parte integrante al presente atto, il documento "Linee guida per la governance del sistema informatico regionale e tutti i suoi allegati, in attuazione di quanto previsto dalle delibere della Giunta regionale n. 1057/2006 e n. 2416/2008, come meglio specificato in premessa;

b. di disporre che il suddetto allegato sia portato a conoscenza di tutte le strutture della Giunta regionale;

c. di applicare l'allegato a partire dall'adozione del presente atto

d. di disporre che tale allegato sia aggiornato ogni qualvolta che l'evoluzione tecnologica o la normativa vigente lo renda necessario.

IL DIRETTORE GENERALE
(Gaudenzio Garavini)

Linee guida per la *governance* del sistema informatico regionale

INDICE

1. Contesto

Gli Indirizzi in ordine alle relazioni organizzative e funzionali tra le strutture e sull'esercizio delle funzioni dirigenziali sono stati approvati con DGR 447/2003 e successivamente aggiornati in risposta all'evoluzione del contesto normativo e organizzativo.

La **DGR 1057/2006** "*Prima fase di riordino delle strutture organizzative della giunta regionale. indirizzi in merito alle modalità di integrazione interdirezionale e di gestione delle funzioni trasversali"*, a fronte del nuovo assetto organizzativo, pur confermando i contenuti di detti Indirizzi, ha disposto la necessità di apportare alcune integrazioni alla DGR 447/2003:

"Va inoltre prevista un'apposita sezione relativa alle Responsabilità dirigenziali in materia di sistemi informativi e sicurezza dei dati e delle procedure informatiche che preveda il ruolo della Direzione generale Organizzazione, sistemi informativi e telematica.

Alla Direzione generale, attraverso le proprie unità organizzative, spetta:

il presidio della coerenza dell'architettura del sistema informativo regionale assicurando l'unitarietà dell'impostazione delle funzioni tecniche settoriali incardinate nelle altre direzioni generali;

lo sviluppo e la gestione delle infrastrutture e dei servizi di garanzia, la progettazione e la realizzazione dei progetti trasversali, gli standard generali di riferimento, l'assistenza tecnica e la collaborazione per lo sviluppo dei servizi e dei sistemi informativi settoriali e locali, anche su richiesta.

Per raggiungere tali obiettivi, la Direzione generale, attraverso i competenti Servizi:

individua gli standard tecnologici, metodologici e di sicurezza, le piattaforme tecnologiche a supporto dei sistemi, i criteri di selezione e assegnazione delle attrezzature informatiche;

esprime una valutazione preventiva, per le procedure e sistemi informativi sviluppati a cura delle Direzioni generali, di rispetto degli standard e piattaforme tecnologiche e di rispondenza ai criteri di qualità e sicurezza individuati per l'insieme del sistema informativo regionale;

effettua, mediante modalità formalizzate ed omogenee, la presa in carico di sistemi informativi sviluppati a cura delle Direzioni generali e il loro passaggio in produzione, a seguito di una valutazione finale del rispetto dei criteri sopra indicati.

Di tali valutazioni deve essere dato formale riscontro negli atti amministrativi assunti dalle Direzioni generali in relazione alla realizzazione dei sistemi informativi, ivi inclusi gli atti di impegno e liquidazione della spesa."

Successivamente, la DGR 2416/2008 "*Indirizzi in ordine alle relazioni organizzative e funzionali tra le strutture e sull'esercizio delle funzioni dirigenziali. adempimenti conseguenti alla delibera 999/2008. Adeguamento e aggiornamento della delibera 450/2007*" sancisce che la rilevanza strategica del sistema informativo regionale, richiede che sia comunque garantita l'unitarietà del sistema informativo nel suo complesso, attraverso l'omogenea applicazione di regole che riguardano tutto l'Ente.

"A tal fine è necessario che l'esercizio di responsabilità ed autonomia da parte delle Direzioni generali sia accompagnato da azioni tecniche di verifica e audit, svolte dalla Direzione generale competente in materia di sistemi informativi/informatici e telematica, a garanzia della coerenza architetturale complessiva del sistema informativo regionale e dell'agevole svolgimento, da parte di quest'ultima, di eventuali successivi sviluppi o servizi complementari."

La Direzione competente in materia di sistemi informativi / informatici e telematica svolge i compiti di programmazione, sviluppo e coordinamento generale previsti al comma 2 dell'art.16 della L.R. 11/2004.

In particolare, in riferimento a quanto previsto alla lettera c) del citato comma ("presidio della coerenza dell'architettura del sistema informativo regionale") la Direzione generale competente in materia di Sistemi informativi /informatici e telematica è responsabile:

a) *della definizione degli standard e piattaforme tecnologiche a supporto delle filiere applicative su cui si sviluppa il sistema informativo dell'Ente Regione;*
b) *della progettazione e realizzazione dei sottosistemi informativi di supporto alle funzioni trasversali dell'Ente e dei sottosistemi informativi strategici a valenza multisettoriale;*
c) *della progettazione e realizzazione, in concorso con le Direzioni generali richiedenti, di sottosistemi informativi per le strutture regionali;*
d) *delle verifiche preventive progettuali in merito al rispetto degli standard definiti in materia di tecnologie, metodologie di sviluppo e documentazione, livelli minimi di sicurezza e accessibilità;*
e) *delle verifiche, preliminari alla presa in carico di prodotti e sottosistemi realizzati da terzi, sul rispetto degli standard definiti in materia di tecnologie, metodologie di sviluppo e documentazione, livelli minimi di sicurezza e accessibilità;*
f) *cura il ciclo di vita dei sistemi realizzati di cui ai punti b) e c), assicurandone la manutenzione evolutiva, l'adeguamento tecnologico, la manutenzione della documentazione;*

I punti d) ed e) vengono svolti attraverso l'espressione di un preliminare riscontro sulla congruenza tecnica, da esprimersi da parte della Direzione generale competente in materia di sistemi informativi / informatici e telematica, relativamente ai segmenti di sistema informativo realizzati dalle Direzioni generali nell'ambito della loro autonomia finanziaria, anche mediante il concorso di fondi non provenienti dal Bilancio regionale.

Le modalità operative attuative di quanto previsto ai punti precedenti, e in particolare i rapporti tra la Direzione generale competente in materia di Sistemi informativi/ informatici e telematica e le altre Direzioni generali, saranno specificate da apposite circolari da emanarsi a cura della Direzione generale competente in materia di Sistemi informativi/informatici e telematica, sentito il Comitato di Direzione."

2. Obiettivi

La *governance* dei sistemi informativi (*IT governance)* è definita come «*il complesso delle decisioni relative alla gestione dell'IT, finalizzate a garantirne l'allineamento con le strategie dell'Ente, con la conseguente attribuzione di responsabilità tra i diversi attori interni ed esterni alla direzione IT*».

Gli obiettivi che si intendono perseguire attivando un sistema di *IT governance* possono essere così sintetizzati:

- allineare l'IT alla strategia dell'Ente in modo che possa portare i benefici attesi;
- migliorare l'erogazione dei servizi IT agli utenti:
  - razionalizzando i criteri di scelta e priorità delle richieste;
  - analizzando gli impatti sulle risorse disponibili, sui sistemi/applicativi, sui processi e sull'organizzazione;
  - garantendo un maggior e miglior controllo dello stato d'avanzamento delle richieste;
- responsabilizzare maggiormente gli utenti sulle richieste e sul loro impatto tecnico ed economico;
- valutare e bilanciare i rischi tecnici, organizzativi ed economici;
- monitorare periodicamente e sistematicamente le prestazioni e l'uso dell'IT;
- verificare la conformità delle richieste rispetto alla normativa vigente;
- assicurare l'integrità, la confidenzialità e la disponibilità dei sistemi, dei dati e delle risorse.

3. Modello organizzativo

Un modello organizzativo adeguato a perseguire tali obiettivi deve:

- **individuare ruoli e processi adeguati, coerenti e integrati tra le diverse strutture.**

  Processi di lavoro snelli e trasparenti, supportati da strumenti condivisi che assicurino, da un lato, che i settori dell'Amministrazione, pur nella propria autonomia, costruiscano sistemi informativi interoperabili, non ridondanti, sicuri, accessibili; dall'altro, che consentano ai settori centrali competenti in materia di IT, di gestire in maniera più consapevole i sistemi informativi per fornire un servizio qualitativamente migliore. Ciò sarà possibile solo se l'intero processo che porta alla realizzazione di un qualsivoglia sistema informativo sarà condiviso e se le interazioni tra le strutture settoriali e quelle centrali non saranno vissute come mere pratiche "dovute" bensì come scambio informativo e arricchimento reciproco, finalizzato all'erogazione di un Servizio pubblico più efficiente ed efficace.

- **presidiare la conoscenza, ripensando l'organizzazione come rete di competenze.**

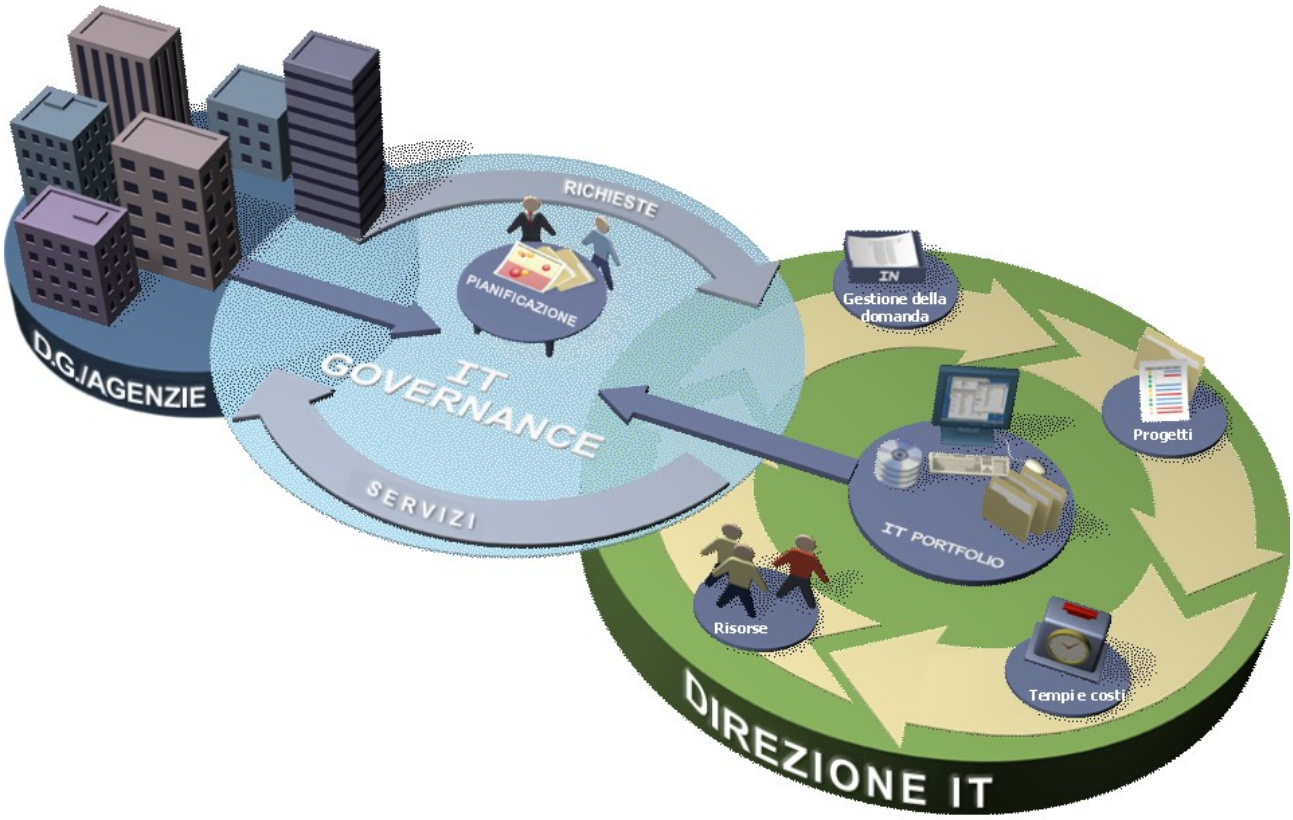
  Poiché il valore delle persone dipende dalla competenza professionale e dalla

capacità di affrontare e risolvere problemi, bisogna generare e condividere la conoscenza.

- **porre attenzione al servizio e al cliente/utente, la cui soddisfazione è il metro e la condizione fondamentale per misurare l'efficacia del servizio stesso.**

  La Direzione generale competente in materia di IT, si trasforma quindi da soggetto che presidia l' infrastruttura tecnologica in soggetto erogatore di servizi.

  In questa logica è importante comprendere il fabbisogno dell'utente/cliente e presidiare/gestire la domanda indirizzandola correttamente, individuandone tempi e costi, allocando le risorse, verificando la conformità a leggi e regolamenti, verificando l'adeguatezza delle infrastrutture esistenti, al fine di programmare investimenti per eventuali potenziamenti dell'infrastruttura.



4. Applicabilità

Il presente documento di Linee guida si applica all'interno della Giunta della Regione Emilia-Romagna quale:

- strumento per condividere un modello organizzativo;
- strumento di riferimento per capitolati tecnici e/o allegati tecnici al conferimento di incarichi e/o documentazione di prodotto e/o servizi dichiarati da fornitori (in quanto contiene gli standard di riferimento dei sistemi informativi regionali);
- strumento per individuare l'offerta di soluzioni già realizzate e pronte per il riuso all'interno dell'Ente (in quanto contiene informazioni su servizi di interoperabilità già realizzati e disponibili a chi ne voglia far uso: es. firma digitale, organigramma, porta di dominio, integrazione con il protocollo, WCM, forum, mailing list, ecc);
- prontuario per effettuare verifiche di prodotti e servizi che si intendono acquisire o realizzare (in quanto contiene check list di riferimento: es. check list per l'accessibilità e la sicurezza delle applicazioni, per la verifica di documentazione obbligatoria per la gestione e manutenzione delle applicazioni, ecc.);
- strumento di riferimento per la richiesta e la distribuzione di dotazioni di strumenti informatici ad uso individuale (PC, portatili, stampanti, collegamenti remoti da casa, ecc.).

Lo scopo di queste linee guida è quindi quello di fornire uno strumento di supporto ai soggetti incaricati di:

a. progettare, sviluppare o acquisire sistemi informativi;
b. valutare/scegliere fornitori di servizi per la realizzazione di sistemi informativi;
c. testare o adeguare sistemi informativi ai criteri di sicurezza previsti dalla normativa vigente;
d. testare o adeguare sistemi informativi ai requisiti di accessibilità previsti dalla normativa vigente;
e. installare, gestire o manutenere sistemi informativi;
f. progettare e/o realizzare siti web;
g. effettuare le verifiche preventive progettuali in merito al rispetto degli standard definiti in materia di tecnologie, metodologie di sviluppo e documentazione, livelli minimi di sicurezza e accessibilità;
h. effettuare le verifiche preliminari alla presa in carico di prodotti e sottosistemi realizzati da terzi;
i. effettuare le verifiche, preliminari al rilascio in produzione di prodotti e sottosistemi realizzati da terzi, anche se in hosting su sistemi esterni.;
j. acquistare/distribuire attrezzature individuali all'interno delle strutture regionali;
k. acquistare strumentazioni hardware.

Standard metodologici e tecnologici di riferimento

Negli ultimi anni, l'avvio di numerosi progetti di *e-government* accompagnato dalla crescente necessità dei settori regionali di interoperare con altri soggetti istituzionali della società civile (Associazioni di categorie, Enti locali, Aziende sanitarie, Ministeri, UE, ecc.), ha sostanzialmente modificato sia in termini qualitativi che quantitativi, il sistema informativo-informatico regionale. Da un sistema informativo progettato e realizzato per le specifiche esigenze funzionali dell'Ente Regione, si è progressivamente implementato un sistema informativo che eroga servizi anche a soggetti esterni che cooperano con la Regione.

Tale crescita, purtroppo non accompagnata da una visione strategica univocamente condivisa con tutte le strutture regionali, ha, in alcuni casi, comportato la proliferazione di soluzioni applicative eterogenee, governate più dall'offerta che non dalla domanda e installazioni di server dedicati "mono-progetto": tutto ciò ha comportato nel tempo numerose difficoltà nella gestione dell'infrastruttura, con conseguente aggravio dei costi di gestione, di una qualità di erogazione del Servizio non ottimale e di una mancanza di possibili economie di scala.

Questa disomogeneità di soluzioni ha comportato anche una ricaduta e un appesantimento delle infrastrutture informatiche dei nostri enti locali, costretti a uniformarsi alle diverse richieste delle DG regionali: **il rischio è che un sistema informativo a sottosistemi stagni all'interno della Regione-Ente induca sistemi informativi a sottosistemi stagni negli Enti della regione-territorio, con la creazione di sistemi magari integrati verticalmente tra diversi livelli territoriali ma sempre più non comunicanti.** E' quindi evidente che in ottica di *Community network* questo appesantimento va risolto.

A ciò si aggiunga che da alcuni anni è stato attivato un processo di migrazione e rivisitazione delle applicazioni su Mainframe verso nuove filiere applicative e ciò ha comportato un aumento della complessità ed eterogeneità delle piattaforme tecnologiche.

In tale contesto, le necessità crescenti in termini di potenza elaborative, scalabilità, alta affidabilità e continuità di Servizio dell'attuale infrastruttura IT regionale, comportano l'urgenza di evolvere l'attuale infrastruttura tecnologica a supporto dei progetti informatici regionali nell'ottica di centralizzare la gestione delle piattaforme applicative, la gestione della sicurezza informatica, la definizione e il controllo degli standard e piattaforme tecnologiche verso una maggiore omogeneità.

La gestione integrata e centralizzata dell'infrastruttura tecnologica e applicativa porta a notevoli benefici, quali in particolare:

- sinergie per l'Ente negli investimenti su hardware, licenze software e risorse umane, tramite condivisione degli ambienti dedicati alle filiere applicative;
- qualità e continuità di erogazione dei servizi garantita da una infrastruttura adeguata ad un Datacenter;
- controllo centralizzato della sicurezza di dati e applicazioni dell'Ente;
- backup e disaster recovery centralizzato per i dati dell'Ente;
- presidio specialistico delle varie piattaforme applicative (risorse umane dedicate in

pratica dalle 8 di mattina alle 8 di sera);

- tempi di intervento praticamente immediati in caso di problemi alle piattaforme;
- politiche di gestione omogenee per tutti i sistemi dell'Ente.

5. Piattaforma infrastrutturale/server dipartimentali

Nell'ottica di ottimizzare la gestione dei sistemi e l'utilizzo delle risorse hardware/software, aumentare la sicurezza fisica dei dati, l'affidabilità e la disponibilità delle applicazioni, si individuano tre filiere software di riferimento per le applicazioni custom in uso presso l'Amministrazione e gestite centralmente dal Servizio Sistema Informativo-Informatico Regionale (d'ora in poi Servizio SIIR).

Le filiere supportate per applicazioni e siti web basate su architetture a tre livelli (web server, application server, database server) sono le seguenti:

- **filiera A**: applicazioni basate su tecnologia **JAVA**
- **filiera B**: applicazioni basate su tecnologia **Microsoft**
- **filiera C**: applicazioni basate su tecnologia **OpenSource**

Il dettaglio relativo allo stack tecnologico implementato sulle diverse piattaforme è descritto in Allegato 1.

L'architettura hardware e software attualmente implementata a supporto delle **Filiere applicative A** e **B** è in grado di garantire:

- scalabilità (crescita costante del numero di utenti target del Servizio),
- qualità e continuità nell'erogazione del Servizio (alta affidabilità dei sistemi),
- supporto sistemistico e specialistico immediato.

In particolare in questi ultimi anni è stata potenziata l'infrastruttura a supporto della **Filiera applicativa A**, poiché, basandosi su tecnologia Java, se implementata secondo le specifiche/regole J2EE (Java 2 Enterprise Edition), permette di sviluppare applicazioni indipendenti dalla piattaforma operativa, tutelando quindi l'investimento da eventuali cambiamenti dei sistemi operativi supportati e degli application server adottati. Per questo motivo si consiglia di utilizzare per applicazioni di classe enterprise e mission critical, la filiera applicativa A.

Per quanto riguarda invece l'architettura hardware e software implementata a supporto della **Filiera applicativa C** si evidenzia che attualmente è garantito supporto sistemistico e specialistico immediato ma si prevede di rendere disponibile un ambiente scalabile e ad alta affidabilità. Visto l'orientamento della UE e le raccomandazioni a livello nazionale, si prevede di incrementare le competenze sulle tecnologie open-source, al fine di promuovere iniziative volte allo sviluppo, alla diffusione e alla conoscenza del software open-source, anche nell'ottica di favorire il pluralismo informatico, così come previsto dalla Legge regionale 11/2004.

L'amministrazione regionale è dotata anche di un **sistema ERP** esteso implementato attraverso il prodotto mySAP Business Suite che consente di centralizzare ed integrare il sistema informativo dell'Ente.

Il trend di crescita delle soluzioni implementate negli ultimi anni presso l'Ente ne fanno a tutti gli effetti una filiera enterprise che si affianca alle precedenti ed assume un ruolo centrale nella gestione dei processi critici dell'Ente.

Il sistema ERP di SAP è modulare e composto da soluzioni specializzate e finalizzate alla gestione di determinati flussi organizzativi e procedurali.

Le componenti di SAP utilizzate nelle realizzazioni dei macro-processi implementati sono:

- R/3 Release 4.6C:FM, FI, AA, CO, MM, PM, MI, SD, RE
- HCM Release ECC 6.0 (MD, OM, Payroll, TM, TV, ESS)
- CRM Release 4.0 (ICWC, ICSS, TREX)
- SEM Release 4.0 (BPS, CPM, BSC)
- BW Release 3.5
- MI (Mobile Infrastructure) Release 7.0

Si sottolinea che le nuove release di SAP si basano tutte sulla piattaforma NetWeaver 2004s ovvero una piattaforma enterprise basata sulla *Service Oriented Architecture* (SOA).

L'interfaccia applicativa client che consente il colloquio con i vari sottosistemi SAP è la SAPGUI. Una componente client che va installata sul personal computer degli utenti che intendono accedere a SAP. Nella sua modalità tradizionale SAP si presenta quindi come un'applicazione client-server.

Qualora una struttura regionale intenda informatizzare un processo integrato, è necessario verificare a priori con il Servizio SIIR la fattibilità attraverso i moduli di SAP.

L'Amministrazione regionale è inoltre dotata di **strumenti di Business Intelligence**, quali SAS e Business Object.

Il **sistema SAS** è utilizzato in differenti settori dell'amministrazione regionale per l'inserimento, l'aggiornamento, la validazione e l'elaborazione dei dati.

SAS viene utilizzato per realizzare report, tabelle e grafici da pubblicare in forma cartacea o su web; in quest'ultimo caso sono disponibili pagine statiche, pagine dinamiche navigabili (drill-down) e reportistica su richiesta.

La nuova versione di SAS (9.1.3) consente la realizzazione di pagine web con tecnologia Java (abbandonando la oramai superata tecnologia CGI) ed una maggiore integrazione con i sistemi di autenticazione adottati dall'Ente. Per la componente Java, SAS si integra con gli application server più diffusi: Tomcat, JBoss, IBM WebSphere, Bea WebLogic. A fronte di una filiera tecnologica Java implementata su IBM WebSphere si è deciso di integrare SAS con tale application server.

Le componenti di SAS attualmente in uso sono le seguenti:

- SAS BI Server
- SAS Data Integration Server
- Base SAS Software

- SAS Integration Technologies Software
- SAS/ACCESS Interface to PC Files Software
- SAS/ACCESS Interface to Oracle Software
- SAS/Connect Software
- SAS Graph Software
- SAS/IntrNet Software
- SAS/STAT Software
- SAS/SECURE for Windows Software
- SAS Scalable Performance Data Server

La **suite Business Object** è utilizzato in differenti settori dell'amministrazione regionale come strumento di reportistica, analisi multidimensionale e cruscotti direzionali attraverso interfaccia sia client che web.

I moduli di Business Object sono utilizzato sia direttamente su diversi sistemi gestionali regionali su cui sono stati sviluppati oltre settanta universi, sia come front end di Data Warehouse.

Vi sono inoltre applicativi gestionali con interfaccia web che richiamano direttamente il portale Business Objects Enterprise (in modalità single sign on) per la visualizzazione/produzione di reportistica operativa e di sintesi.

Le componenti di Business Object attualmente in uso sono:

- Portale Web Business Objects Enterprise (ex Infoview)
- Business Objects XI r3.1 Desktop Intelligence e Web Intelligence
- Universe Designer XI r3.1
- Web Intelligence Rich Client per visualizzazione report web off-line
- Xcelsius Enterprise per la creazione e fruizione di cruscotti direzionali
- Business Objects Enterprise Central Management Console
- Knowledge Accelerator per e-learning e documentazione.

5.1 Infrastruttura tecnologica a supporto delle filiere applicative

Il Servizio SIIR ha da tempo avviato la realizzazione di una infrastruttura tecnologica centralizzata al fine di conseguire i seguenti obiettivi principali:

- qualità e continuità di erogazione dei servizi da parte delle filiere applicative,
- gestione centralizzata della sicurezza di dati e applicazioni,
- backup e disaster recovery centralizzato per i database e le piattaforme operative,
- presidio specialistico delle filiere applicative,
- politiche di gestione omogenee per tutte le piattaforme hardware / software,

- ottimizzazione e sinergie nell'utilizzo delle piattaforme hardware / software.

Il dettaglio delle tecnologie utilizzate a supporto delle filiere applicative è riportato in Allegato 2.

5.2 Servizi centralizzati forniti a supporto delle filiere applicative

Il Servizio SIIR, nell'ambito delle tre principali filiere applicative supportate, fornisce servizi di supporto e system integration durante le fasi di progettazione, implementazione e gestione di sistemi informativi. I servizi forniti sono:

- Installazione, configurazione e gestione delle piattaforme hardware / software di sviluppo, test e produzione
- Monitoring tecnico e applicativo delle piattaforme hardware / software di sviluppo, test e produzione
- Gestione security e business continuità delle piattaforme hardware / software
- Implementazione soluzioni inerenti software di integrazione
- Gestione evoluzione piattaforme hardware / software di sviluppo, test e produzione
- Gestione procedure di backup / restore
- Supporto sistemistico ai gruppi di sviluppo applicativo

Nella Tabella che segue sono elencate nel dettaglio le principali attività operative svolte dal Servizio SIIR a supporto delle filiere applicative.

Fase	Piattaforma	Attività svolte per le 3 filiere applicative supportate
Progettazione	Sistema Operativo	• Dimensionamento apparati hardware (storage / CPU / memory / network) • Network planning (assegnazione indirizzi IP privati e pubblici)
	Web Server	• Dimensionamento dei web server per gestione traffico utenza • Progettazione security policy
	Application Server	• Progettazione architettura clustering orizzontale o verticale • Dimensionamento degli application server
	Database Server	• Dimensionamento database server • Progettazione architettura fisica
Implementazione	Sistema Operativo	• Installazione e configurazione del sistema operativo • Configurazione servizi di clustering • Configurazione procedure di backup per il sistema operativo • Creazione utenze e configurazione permessi di accesso
	Web Server	• Installazione e configurazione dei web server nella rete DMZ • Configurazione procedure di backup e di monitoring dei web server
	Application Server	• Installazione e configurazione degli application server • Configurazione procedure di backup e di monitoring
	Database Server	• Installazione e configurazione dei database server • Configurazione procedure di backup e di monitoring • Creazione utenze e configurazione permessi di accesso
Gestione	Sistema Operativo	• Tuning, monitoring e patching sistema operativo • Verifica backup/restore sistema operativo
	Web Server	• Analisi log dei web server nella rete DMZ • Configurazione web server
	Application Server	• Deployment e maintenance applicazioni • Analisi log e patching application server • Tuning security, performance tuning e monitoring
	Database Server	• Supporto all'implementazione di modifiche evolutive/correttive • Tuning security, performance tuning e auditing dei database server • Verifica backup/restore • Monitoring e patching database server

Si evidenzia che l'Ente, a meno di particolari esigenze opportunamente motivate, non fornisce accesso ai sistemi di test e produzione ai fornitori esterni.

5.3 Gestione operativa dei sistemi delocalizzati presso le Direzioni/Agenzie Regionali

Il processo di consolidamento e razionalizzazione dell'infrastruttura server e storage localizzata presso il CED regionale si è concluso. Già dal 2008 si è avviato un processo di centralizzazione e consolidamento presso il CED regionale di sistemi Server delocalizzati presso alcune Direzioni ed Agenzie regionali realizzando la dismissione di diversi sistemi server e l'integrazione delle applicazioni nell'ambito delle filiere applicative previste in allegato 1.

Si sottolinea che il processo di razionalizzazione e consolidamento coinvolge tutte le Direzioni ed Agenzie regionali che attualmente gestiscono sistemi server su cui sono installate applicazioni e servizi che possono essere centralizzati nell'ambito dei sistemi e delle filiere applicative già previste al CED regionale. Il processo di consolidamento si conclude con la dismissione fisica del server e relativo ritiro.

Di norma, e ciò deve essere verificato caso per caso, la centralizzazione dei sistemi non coinvolge i file server delocalizzati.

Si evidenziano, di seguito, i vantaggi e le peculiarità che la centralizzazione dei sistemi e dei servizi comportano:

Assistenza sistemistica estesa	Il CED regionale garantisce assistenza 6 giorni su 7. Dal lunedì al venerdì dalle 07 alle 19. Il sabato dalle 08 alle 14
Riduzione dei costi per hardware e licenze	Centralizzando i sistemi diminuiscono i costi di gestione poiché occorre considerare il fattore obsolescenza dell'hardware, manutenzione e licenze software di base
Architettura ridondante	Incremento della disponibilità e della affidabilità dei sistemi legata alla ridondanza dei server e dello storage normalmente implementata presso le architetture tecnologiche del CED
Procedure centralizzate nella gestione dei sistemi	Controllo sulla sicurezza di dati e applicazioni, monitoraggio integrato dei sistemi e delle applicazioni, backup/restore centralizzato, business continuità, supporto sistemistico immediato e specializzato
Gestione centralizzata e governo della spesa su strumentazioni hardware e relativo software di base	Ai sensi della DGR 999/2008 si individuano possibili sinergie e la definizione di politiche di acquisto comune

Nella fase di transizione, ovvero durante il processo di centralizzazione dei server delocalizzati verso il CED regionale, gli stessi server ancora in gestione alle singole strutture regionali dovranno essere presidiati dal personale tecnico della singola struttura

secondo le policy di seguito descritte:

Implementazione di un nuovo sistema server

Non sono implementati nuovi server presso le strutture regionali a meno di esigenze molto particolari e non risolvibili presso il CED. Nel caso in cui il personale tecnico delle singole strutture regionali e del CED verifica la necessità di implementare un nuovo sistema presso la struttura regionale, tale sistema dovrà essere gestito secondo le seguenti politiche:

Attività di implementazione del sistema operativo:

- Dimensionamento apparati hardware (storage / CPU / memory / network)
- Network planning (assegnazione indirizzi IP privati e pubblici)
- Installazione e configurazione del sistema operativo
- Installazione e configurazione agenti di monitoring e integrazione con l'infrastruttura di monitoring e management regionale
- Configurazione eventuali servizi di clustering
- Configurazione procedure di backup per il sistema operativo
- Creazione utenze e configurazione permessi di accesso
- Integrazione con il sistema antivirus regionale
- Integrazione del sistema nel dominio regionale

Attività di implementazione dell'Application Server:

- Progettazione architettura clustering orizzontale o verticale
- Dimensionamento degli application server
- Installazione e configurazione degli application server
- Configurazione procedure di backup e di monitoring
- Creazione utenze e configurazione permessi di accesso
- Attivazione autenticazione integrata

Attività di implementazione del Database Server:

- Dimensionamento database server
- Progettazione architettura fisica
- Installazione e configurazione dei database server
- Configurazione procedure di backup e di monitoring
- Creazione utenze e configurazione permessi di accesso
- Attivazione autenticazione integrata

Attività periodiche di gestione del sistema server

Attività di gestione inerenti il Sistema Operativo:

- Verifiche periodiche della compatibilità e patching sistema operativo (windows, linux, ecc..)
- Aggiornamento agenti di monitoring
- Verifiche periodiche Backup / Restore sistema operativo
- Monitoring piattaforma operativa (CPU, Memory, Storage, Network)

- Tuning periodico security piattaforma operativa

Attività di gestione inerenti l'Application Server:

- Verifiche periodiche della compatibilità e patching application server (es. WebSphere, Jboss, Tomcat,…)
- Verifiche periodiche Backup / Restore application server
- Performance tuning application server (CPU, memory, ecc..)
- Analisi periodica log application server (individuazione errori su servlet, EJB, ecc..)
- Deployment periodici applicazioni Java

Attività di gestione inerenti il Database Server:

- Verifiche periodiche compatibilità e patching database (Oracle, SQL Server, MySQL, ecc..)
- Verifiche periodiche Backup / Restore database
- Tuning security database e gestione grants di accesso ai dati
- Analisi periodica log database
- Performance tuning database server (Memory, CPU, ecc..)
- Delivery periodiche di modifiche evolutive/correttive al database

Il livello di aggiornamento dei sistemi delocalizzati sarà soggetto a verifiche periodiche nell'ambito dei controlli eseguiti da personale specializzato al fine di garantire la security dell'infrastruttura IT dell'Amministrazione e l'aderenza agli standard adottati.

Ove, visto il livello di complessità tecnica delle attività in oggetto, la struttura tecnica delle Direzioni/Agenzie non fosse in grado di operare in autonomia sui server, la struttura sistemistica del Servizio SIIR procederà con la massima priorità alla dismissione del server e alla migrazione delle funzionalità applicative presso i sistemi gestiti centralmente.

5.4 Networking

L'inserimento di nuovi sistemi informatici nella rete degli uffici regionali richiede di identificarne:

- la collocazione più opportuna tra le varie zone di rete gestite (Intranet con IP privati; Intranet con IP pubblici; DMZ; punto d'interconnessione della rete Lepida, Internet, ecc.);
- le esigenze di comunicazione, e con quali prestazioni di banda, con gli utenti interni all'ente regione, con gli utenti di altri enti che fanno parte della rete regionale, con altri enti pubblici o con il mondo Internet;
- al contrario, la necessità di realizzare ambiti di rete compartimentati, in cui non sia permesso l'accesso dall'infrastruttura di rete normalmente utilizzata dagli uffici;
- la necessità di usare protocolli di comunicazione particolari, che possono saturare la rete o danneggiare altre applicazioni preesistenti, o creare falle nei sistemi di sicurezza;
- la necessità di accedere a servizi di teleassistenza forniti da aziende che non fanno

parte dell'ambito regionale.

Sostanzialmente è indispensabile realizzare uno studio approfondito, in collaborazione con i tecnici esperti in tecnologie di rete che gestiscono la rete regionale, di tutti gli aspetti del progetto che hanno interconnessioni con le tecnologie di rete.

Ciò va fatto fin dalle fasi iniziali della progettazione del sistema stesso, in quanto le scelte sugli strumenti ed i protocolli utilizzati possono permettere, o al contrario impedire, un corretto funzionamento dell'applicazione stessa o di altre applicazioni d'interesse dell'Amministrazione.

6. Applicazioni

Le applicazioni del sistema informativo regionale che si intendono installare sui server gestiti centralmente dal Servizio SIIR, dovranno rispettare gli standard tecnologici relativi alle filiere applicative supportate (vedi paragrafo **5 Piattaforma infrastrutturale/server dipartimentali**).

Tutte le applicazioni facenti parte del sistema informativo regionale, anche se in hosting su server esterni, devono rispettare requisiti di accessibilità definiti dalla Legge Stanca n. 4 del 2004 e dal successivo Decreto ministeriale 8 luglio 2005 (vedi paragrafo **6.3 Accessibilità**) e le misure minime di sicurezza indicate nel D.Lgs. 196/2003 "Codice per la protezione dei dati personali" (vedi paragrafo **6.4 Sicurezza**).

E' importante che ogni applicazione sia corredata dalla documentazione necessaria all'installazione, alla gestione e manutenzione.

Per le applicazioni che lo necessitano, sono disponibili *web services* per l'integrazione con:

- il sistema di protocollo informatico in uso presso la Regione Emilia-Romagna,
- l'infrastruttura di firma digitale,
- l'organigramma delle strutture e del personale,
- il sistema di gestione documentale.

Inoltre è in corso un progetto del Piano Telematico 2007-2009, denominato **CoopERa**, nell'ambito del quale è prevista la realizzazione di un'infrastruttura di cooperazione applicativa, che consentirà attraverso l'implementazione di una porta di dominio unica, l'interoperabilità tra gli Enti.

6.1 Documentazione

Di seguito si fornisce un riferimento di contenuti minimi per i principali documenti, di cui un'applicazione *custom* del sistema informativo regionale deve essere corredata. Il dettaglio di ogni singolo documento sarà commisurato alla complessità dell'applicazione. Per i prodotti che prevedono una propria metodologia, quali ad esempio SAP, la tipologia di documentazione va sottoposta per verifica al Servizio SIIR, a garanzia della copertura delle informazioni in essa contenute, necessarie all'attivazione e gestione dell'applicazione stessa.

Analisi dei requisiti: il documento deve contenere l'elenco formale e relativa descrizione di

tutti requisiti dell'applicazione, siano essi funzionali che qualitativi, emersi nella fase di definizione delle esigenze utente.

In particolare, deve contenere:

- definizione del contesto attuale
- descrizione delle esigenze
- vincoli
- requisiti di sicurezza
- numero e tipologia degli utenti coinvolti
- dati trattati, in forma di schema concettuale iniziale
- indicazioni generali della soluzione, sia in termini funzionali che architetturali
- matrice ruoli/funzionalità
- riferimenti a ulteriore documentazione di interesse prodotta o preesistente (esempio: definizione dei requisiti nella documentazione di gara, studi di fattibilità, documentazione a corredo del software originale da assoggettare a MEV, resoconti riunione, ecc.).

Analisi funzionale: il documento definisce totalmente l'applicazione in modo da ottenere una descrizione funzionale completa e non ambigua.

Contiene in modo completo ed esaustivo l'analisi dell'applicazione interessata sia relativamente ai processi ed alle modalità con cui tali processi risulteranno visibili agli utenti finali, sia al disegno logico dei dati secondo il modello relazionale, sia per quanto riguarda gli aspetti non funzionali (architettura, sicurezza, vincoli, prestazioni, ecc.), sia alla documentazione delle interfacce.

In particolare, deve contenere:

- descrizione dell'architettura tecnologica,
- descrizione del sistema di sicurezza da implementare a fronte di un'analisi dei rischi,
- disegno della base dati: schema concettuale definitivo e schema relazionale,
- descrizione della logica applicativa per ogni funzionalità individuata,
- dettaglio schema di navigazione/pagine e/o prototipo interfaccia/maschere,
- eventuali collegamenti con sistemi esterni,
- eventuale descrizione dei flussi di dati previsti dall'applicazione.

Manuale di istallazione e gestione: è lo strumento necessario alle strutture preposte all'installazione ed esercizio dell'applicazione. È un manuale rivolto a personale tecnico e deve contenere tutte le informazioni necessarie per installare, configurare e gestire l'applicazione. Possono essere indicati eventuali requisiti particolari di gestione del backup/restore dei dati ed eventuale necessità di un piano di Disaster Recovery, se diversi dalle politiche definite sui server.

Manuale utente: il documento deve fornire una descrizione generale dell'applicazione e una guida operativa all'utilizzo delle singole funzionalità disponibili.

6.2 Sviluppo

Le applicazioni che l'Amministrazione regionale prende in carico sui propri sistemi

dovranno essere progettate e sviluppate:

- per essere eseguite in concorrenza (condividendo la stessa infrastruttura tecnologica) con altre applicazioni e quindi non dovranno effettuare operazioni che potrebbero ridurre o bloccare il funzionamento di altre applicazioni e/o servizi;
- per poter essere eseguite su application server configurati in cluster;
- per poter accedere a database in remoto e configurati in cluster;
- per poter essere implementate su infrastutture di rete che utilizzano servizi di network load balancing;
- per essere compatibili con le patch e gli aggiornamenti dei sistemi operativi, dei rdbms e application server;
- per essere accessibili secondo i requisiti definiti nel D.M. 8/8/2005 (vedi paragrafo Accessibilità) e usabili;
- per essere sicure tenendo conto degli aspetti tecnici e procedurali definiti nel Disciplinare tecnico in materia di sicurezza delle applicazioni informatiche nella Giunta della Regione Emilia-Romagna", approvato con Determinazione n. 2651 del 2007 (vedi paragrafo Sicurezza).

Nello sviluppo di un'applicazione si raccomanda di prestare particolare attenzione ai seguenti aspetti:

- convenzione per la scrittura del codice;
- documentazione del codice;
- meccanismi di autenticazione centralizzata;
- chiusura delle connessioni al database;
- validazione dell'input e dell'output (lato server);
- gestione e controllo degli errori;
- riutilizzo del codice;
- scalabilità dell'applicazione.

In particolare il Servizio SIIR ha definito delle linee guida specifiche per lo sviluppo di applicazioni in tecnologia Microsoft .NET (filiera B) riportate in Allegato 3: è disponibile e documentata una libreria di utility per la gestione della sicurezza, utilizzo del DB server, gestione delle eccezioni, invio di mail, generazione documenti pdf, utilizzo di ftp, etc.

Come strumento di reportistica in ambiente Microsoft è disponibile Reporting Services.

Nel caso in cui le applicazioni siano sviluppate in ASP, è importante sapere se e quali componenti (proprietari e open source: dll, com+, …) vengono utilizzate per verificarne la compatibilità con i sistemi regionali.

In particolare sono presenti sui sistemi regionali, e quindi utilizzabili, i seguenti componenti:

- ASP EMAIL per l'invio delle mail,

- SoftArtisans versione 2.4.1.1. per l'upload di file,
- MS Soap Toolkit 3.0.

Le applicazioni sviluppate nell'ambito della filiera Java, per garantire l'indipendenza dall'application server devono attenersi alle specifiche dettagliate nell'Allegato 4.

Considerata l'eterogeneità del software di base ed applicativo implementato presso i sistemi informatici dell'Ente è opportuno adottare strumenti che facilitano ed automatizzano sia l'integrazione applicativa sia l'integrazione dei dati; ove si verifichino tali esigenze occorre concordare preventivamente con il Servizio SIIR quali strumenti e metodologie adottare in fase di analisi dell'applicazione da sviluppare.

Infine un'applicazione prima di essere installata sui sistemi dell'Ente deve essere opportunamente testata dal fornitore in un ambiente analogo a quello regionale al fine di assicurare:

- il rispetto dei requisiti funzionali concordati con il committente,
- la conformità alla normativa vigente e ai regolamenti/disciplinari/linee guida dell'Ente (accessibilità, usabilità, sicurezza, etc..),
- il corretto funzionamento di eventuali integrazioni con altri sistemi,
- il rispetto dei requisiti prestazionali richiesti (nel caso in cui si prevede che l'applicativo debba essere utilizzato da un numero di utenti elevato prevedere un test di carico).

6.3 Accessibilità

Con il termine "accessibilità" si intende la caratteristica di un sito/applicazione di rendere possibile l'accesso ai suoi contenuti e funzionalità a tutti gli utenti, indipendentemente dalla presenza di disabilità (fisiche, sensoriali, cognitive) e dalle dotazioni hardware e software.

Una pubblica amministrazione deve realizzare siti/applicazioni accessibili: il punto di riferimento normativo è la legge n.4 del 9 Gennaio 2004 "Disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici" e le relative indicazioni tecniche fornite in successivi decreti presidenziali e ministeriali.

La Legge 4/2004 (nota come "legge Stanca") obbliga la PA ad inserire nei contratti che stipula con i fornitori di servizi web una clausola per il rispetto dei principi di accessibilità, in caso contrario è previsto l'annullamento del contratto e si incorre in responsabilità dirigenziali e disciplinari.

Inoltre, la Legge 4/2004 impone di considerare anche l'accessibilità nelle procedure per l'acquisto di beni e servizi informatici; chi non lo fa o chi acquista beni/servizi non accessibili deve motivare la scelta.

Quando si scrive un contratto o un capitolato tecnico per un prodotto o Servizio Web (sito, applicazione o CD-ROM/DVD) è quindi necessario inserire una clausola che preveda il rispetto dei requisiti di accessibilità (Allegato 5).

La Regione Emilia-Romagna inoltre è da molti anni impegnata nel rimuovere le barriere di accesso ai propri siti e applicazioni web e nel realizzare siti e applicazioni non solo accessibili ma anche efficaci, efficienti e soddisfacenti per gli utenti che ne fruiscono,

nonché coerenti con gli altri prodotti in uso in Regione. A questo scopo sono state realizzate apposite "Linee Guida per realizzare siti e applicazioni web accessibili per la Regione Emilia-Romagna" che oltre a contenere indicazioni tecniche per il rispetto della normativa sull'accessibilità forniscono anche ulteriori suggerimenti per garantire uniformità e coerenza nella progettazione, realizzazione, manutenzione e gestione di tali prodotti, in modo che ogni iniziativa o progetto su web si inserisca armonicamente nel contesto della comunicazione dell'ente, favorendo l'integrazione con i sistemi esistenti. Queste linee guida sono costantemente aggiornate in base agli sviluppi normativi e tecnici e sono reperibili online all'indirizzo:

http://www.regione.emilia-romagna.it/lineeguida/

Per garantire un livello di qualità uniforme nei servizi forniti all'utenza è pertanto necessario che nei contratti o capitolati tecnici per prodotti o servizi Web sia richiamato esplicitamente anche il rispetto di tali linee guida (Allegato 5) facendo riferimento alla versione online più aggiornata.

Il rispetto delle indicazioni fornite nelle Linee guida viene richiesto anche ai collaboratori regionali che a vario titolo progettano, realizzano, gestiscono siti e applicazioni web; il responsabile del sito/applicazione è infatti tenuto a rispondere anche del livello di accessibilità e qualità di tale prodotto per quanto di sua competenza.

Nell'Allegato 6 viene riportata la lista dei requisiti che può essere utilizzata per effettuare la verifica di accessibilità su siti o applicazioni web ma si consiglia di consultare sempre la versione aggiornata della checklist, scaricabile dalle pagine contenenti le linee guida:

http://www.regione.emilia-romagna.it/lineeguida/

Il Servizio SIIR è comunque a disposizione di tutte le Direzioni/Agenzie per fornire indicazioni tecniche, suggerimenti, supporto in merito.

La Legge 4/2004 non si limita ai siti ed applicazioni internet ma impone il rispetto dell'accessibilità anche per sistemi operativi, applicazioni o prodotti da scaffale, che non utilizzano tecnologie internet.

Anche quando si acquistano beni/servizi di questo tipo pertanto è necessario prevedere nelle procedure di acquisto una clausola sull'accessibilità del prodotto, e in caso contrario motivarne la scelta (Allegato 5).

6.4 Sicurezza

Qualora un sistema informativo tratti dati personali, esso deve essere realizzato assicurando il totale rispetto del D.Lgs. 196/2003 "Codice in materia di protezione di dati personali" (di seguito denominato Codice). Con ciò si intende che il sistema informativo non solo deve assicurare il rispetto di tutti gli obblighi di sicurezza *"in modo da ridurre al minimo, mediante l'adozione di idonee e preventive misure di sicurezza, i rischi di distruzione o perdita anche accidentale, dei dati stessi, di accesso non autorizzato o di trattamento non consentito o non conforme alle finalità della raccolta*" ma anche il rispetto delle regole ulteriori per i soggetti pubblici.

Il Codice dispone infatti delle regole particolari per il trattamento di dati personali da parte degli enti pubblici non economici, in particolare all'articolo 19.

L'art. 19, infatti, recita:

"Comma 1: Il trattamento da parte di un soggetto pubblico riguardante dati diversi da quelli sensibili e giudiziari è consentito, fermo restando quanto previsto dall'articolo 18, comma 2, anche in mancanza di una norma di legge o regolamento che lo preveda espressamente.

Comma 2: La comunicazione da parte di un soggetto pubblico ad altri soggetti pubblici è ammessa quando è prevista da una norma di legge o regolamento. In mancanza di tale norma la comunicazione è ammessa quando è comunque necessaria per lo svolgimento di funzioni istituzionali e può essere iniziata se è decorso il termine di cui all'articolo 39, comma 2, e non è stata adottata la diversa determinazione ivi indicata.

Comma 3: la comunicazione da parte di un soggetto pubblico a privati o a enti pubblici economici e la diffusione da parte di un soggetto pubblico sono ammesse unicamente quando sono previste da una norma di legge o regolamento".

In base a quanto disposto dall'articolo 19, comma 3, quindi, i soggetti pubblici possono effettuare operazioni di diffusione di dati personali diversi da quelli sensibili e giudiziari soltanto se questo è previsto da una norma di legge o di regolamento.

Si sottolinea anche che l'interconnessione tra sistemi telematici di enti diversi, equivale ad un trattamento di comunicazione ai sensi del Codice. E' opportuno quindi che qualora si intenda effettuare diffusione di dati personali di cui la Regione Emilia-Romagna è Titolare (ai sensi dell'art. 28 del Codice), anche attraverso portali gestiti da altri, o si intenda effettuare interconnessioni con sistemi di altri enti, si verifichi preventivamente, ciascuno per la parte di propria competenza e titolarità, la legittimità, la necessità, la pertinenza, la completezza e la non eccedenza rispetto alle finalità per le quali i dati stessi sono stati raccolti e trattati, nonché la legittimità dei trattamenti di diffusione e comunicazione.

La sicurezza dei dati e delle informazioni deve essere poi considerata in ogni fase del ciclo di vita di un'applicazione: dalla progettazione allo sviluppo, dal test alla gestione.

Gli aspetti tecnici e procedurali necessari per progettare, sviluppare, testare e gestire in modo sicuro un'applicazione regionale sono stati definiti nel "Disciplinare tecnico in materia di sicurezza delle applicazioni informatiche nella Giunta della Regione Emilia-Romagna", approvato con Determinazione del Direttore Generale all'Organizzazione, Personale, Sistemi informativi e Telematica n. 2651 del 2007.

Le indicazioni in esso contenute si basano sul fondamento che un'applicazione è sicura quando è in grado di preservare confidenzialità, integrità e disponibilità delle risorse, assicurando costantemente:

- l'identificazione dell'utente che accede alle risorse;
- la limitazione degli accessi alle risorse;
- la comunicazione sicura con l'esterno;
- la conservazione sicura dei dati.

Il Disciplinare definisce sette principi chiave da rispettare durante la progettazione e lo sviluppo di un'applicazione e contiene inoltre utili liste di controllo che riassumono in modo schematico le indicazioni contenute nel documento.

In particolare ogni applicazione che tratta dati personali, deve rispettare le misure minime di sicurezza, di cui si fornisce in Allegato 7 la lista di controllo.

Quando si scrive un contratto o un capitolato tecnico per l'acquisizione di prodotti o servizi IT è necessario inserire una clausola che preveda la conformità a quanto previsto dal D.Lgs. 196/2003 “Codice in materia di protezione dei dati personali”. (Allegato 8).

6.4.a Active Directory

Active Directory è un ambiente di directory protetto che consente l’autenticazione e l’autorizzazione degli utenti per l’accesso alla rete e alle relative risorse.

Active Directory supporta vari protocolli Internet e meccanismi di autenticazione sicuri utilizzati per la verifica dell'identità al momento dell'accesso, inclusi Kerberos V5, i certificati X.509 v3, le smart card, l'infrastruttura a chiave pubblica (PKI) e il protocollo LDAP (Lightweight Directory Access Protocol) mediante SSL (Secure Sockets Layer).

Oltre a garantire la protezione dell'accesso alla rete mediante l'autenticazione, Active Directory consente di proteggere le risorse condivise semplificando l'autorizzazione degli utenti. Dopo l'autenticazione di un accesso utente in Active Directory, i diritti assegnati all'utente mediante i gruppi di protezione e le autorizzazioni assegnate per la risorsa condivisa determinano se l'utente potrà accedere alla risorsa. Questo processo di autenticazione protegge le risorse condivise da accessi non autorizzati e consente l'accesso solo ad utenti e gruppi autorizzati.

Durante il processo di assegnazione delle autorizzazioni di accesso per le risorse, ad esempio condivisioni file, stampanti, applicazioni e così via, si assegnano le autorizzazioni a un gruppo di protezione anziché a singoli utenti. Questo approccio consente di concedere le autorizzazioni un'unica volta al gruppo, anziché più volte ai singoli utenti. Ogni account aggiunto a un gruppo riceve i diritti assegnati al gruppo in Active Directory e le autorizzazioni sulla risorsa definite per tale gruppo.

Gli account di accesso del personale dipendente sono direttamente connessi ai dati contrattuali, garantendo quindi che ogni variazione contrattuale (spostamento di struttura, cessazione del rapporto di lavoro, ecc.) venga immediatamente registrato sul rispettivo account di accesso.

Questo automatismo non è ovviamente applicabile per tutte quelle utenze che non sono registrate nel database del personale (consulenti ed altro personale non dipendente), per i quali si è deciso di regolamentare la presenza e l’accesso alla rete nel modo seguente:

1) la richiesta di creazione di un account perviene agli amministratori del dominio (o al personale delegato) dal responsabile (o dal referente informatico) della struttura per la quale il nuovo utente collabora. La richiesta indica anche la data presunta di fine attività, che in ogni caso non può essere superiore a sei mesi;
2) sarà cura dell'utente stesso, avvisato dell'imminente scadenza del proprio account al momento del Log-in, richiedere al proprio referente una proroga per un periodo che non potrà in ogni caso essere superiore a sei mesi.

Active directory è fonte autoritativa per il nuovo sistema di Identity Management: le modifiche fatte sul dominio interno vengono replicate sul sistema di Identity e sui sistemi target ad esso collegati. La gestione degli utenti del dominio Active Directory esterno è

spostata sul sistema di Identity che permette un sistema distribuito di delega ai referenti più capillare. Le policy di gestione dell'utente sono le stesse che erano state impostate per l'Active Directory esterno.

6.4.b Autenticazione applicativa

Nel caso in cui un'applicazione preveda un meccanismo di autenticazione, è opportuno utilizzare, ove possibile, meccanismi centralizzati, in modo che l'autenticazione non sia parte del codice applicativo ma sia basata su meccanismi dedicati.

In particolare, considerato che i server dipartimentali e tutti i client delle sedi principali della Regione, fanno riferimento a un dominio nativo Microsoft Windows 2003 (e sue successive evoluzioni) che certifica tutti gli utenti regionali e che tutti gli utenti sono censiti nel sistema di Identity Management, è possibile utilizzare il sistema di Access Management oppure, se questo non è possibile per le caratteristiche dell'applicazione o per la tipologia di utenti gestiti, il sistema custom di autenticazione centralizzato basato su Microsoft Active Directory.

Il sistema di Identity & Access Management consente un controllo centralizzato sia nella gestione delle identità che nel controllo degli accessi ai sistemi ed alle applicazioni, migliorando, inoltre, l'usabilità degli stessi da parte degli utenti, in conformità con i requisiti di legge.

L'utilizzo del sistema di Identity Management regionale, fornisce il valore aggiunto di poter usufruire di un sistema di autenticazione federato in quanto si fa carico di federarsi con altri sitemi di autenticazione diversi (es. FEDERA).

Il meccanismo utilizzato dal sistema di Access Management si appoggia su un Directory LDAP che contiene le credenziali degli utenti, che sono le stesse dei domini Active Directory, e su un Policy Server che gestisce il profilo di base dell'utente per l'accesso alle applicazioni. In questo modo ogni utente che si collega all'Access Manager vede solo le applicazioni per le quali è autorizzato ad accedere.

Il Directory è popolato tramite il sistema di Identity Management che, attraverso i suoi meccanismi di delega, permette ai referenti delle applicazioni di gestire in autonomia l'accesso degli utenti alle stesse.

Il sistema di Identity & Access Management (Sun Java System Identity Management Suite) si occupa solo dell'autenticazione dell'utente e non delle autorizzazioni (ad esempio profilature degli utenti, etc…) che rimangono a carico dell'applicazione client.

La documentazione tecnica di dettaglio con le specifiche per utilizzare il sistema di Identity Management per la gestione centralizzata degli utenti e per utilizzare l'autenticazione dell'Access Management è fornita in Allegato 9.

Il meccanismo basato su Active Directory consiste nel sistema di autenticazione centralizzata, applicazione custom sviluppata internamente.

Ci sono due possibilità per sfruttare i servizi di questo sistema centralizzato:

- appoggiarsi per il login e la modifica della password all'applicazione web di autenticazione centralizzata che effettua l'integrazione con l'applicazione client tramite "browser redirection";

- richiamare i servizi web che implementano i metodi necessari per l'autenticazione.

In entrambi i casi ci si occupa solo dell'autenticazione dell'utente e non delle autorizzazioni (ad esempio profilature degli utenti, etc…) che rimangono a carico dell'applicazione client.

L'autenticazione consiste nella verifica dell'esistenza dell'utente nei domini "Active Directory" e nella verifica dell'appartenenza ad un gruppo di dominio definito a priori per ciascuna applicazione client.

In particolare sono presenti due domini "Active Directory", uno contenente gli account degli utenti regionali (dominio rersdm) e l'altro contenente account di utenti esterni alla Regione (dominio extrarer).

Utilizzando sistemi di autenticazione centralizzati si ottiene un doppio vantaggio: l'utente non è costretto a ricordare una nuova userid/password e le policy impostate sui domini comportano l'adempimento di alcune delle misure minime di sicurezza previste dal "Codice in materia di protezione dei dati personali" (D. Lgs. 196/03).

La documentazione tecnica di dettaglio che spiega come utilizzare il sistema di autenticazione centralizzato custom è fornita in Allegato 9.

6.5. Interoperabilità

Un web service è un componente applicativo. Si può definire come un sistema software in grado di mettersi al Servizio di un applicazione comunicando messaggi formattati secondo lo standard XML su di una medesima rete tramite il protocollo HTTP. Un web service consente quindi alle applicazioni che vi si collegano di usufruire delle funzioni che mette a disposizione.

Proprio grazie all'utilizzo di standard basati su XML, tramite un'architettura basata sui Web Service, applicazioni software scritte in diversi linguaggi di programmazione e implementate su diverse piattaforme hardware possono quindi essere utilizzate, tramite le interfacce che queste "espongono" pubblicamente e mediante l'utilizzo delle funzioni che sono in grado di effettuare (i "servizi" che mettono a disposizione) per lo scambio di informazioni e l'effettuazione di operazioni complesse sia su reti aziendali come anche su Internet: la possibilità dell'interoperabilità fra diversi software (ad esempio, tra Java e .net) e diverse piattaforme hardware (come Windows e Linux) è resa possibile dall'uso di standard "aperti".

Alcune applicazioni e infrastrutture regionali, per la loro natura di trasversalità, sono state progettate e realizzate anche nell'ottica di mettere a disposizione funzioni utili ad altre applicazioni attraverso servizi web.

Il sistema di protocollo informatico in uso presso la Regione Emilia-Romagna mette a disposizione sulla intranet alcuni servizi web che consentono l'integrazione con altre applicazioni che lo necessitano. In particolare i servizi web sono i seguenti:

- Protocollazione: consente di effettuare una protocollazione in entrata, in uscita e tra uffici.
- Protocollazioni allegati: consente di effettuare una protocollazione in entrata, in uscita e tra uffici con allegati.
- Ricerca documento: consente di effettuare una ricerca per reperire le informazioni di uno o più documenti.

- Ricerca anagrafica esterna: consente di effettuare una ricerca per reperire le informazioni di una o più anagrafiche esterne a partire dal codice fiscale oppure dal cognome e nome/ragione sociale.
- Acquisizione da sportello: consente di effettuare una protocollazione (acquisizione) da sportello.
- Ricerca fascicoli: consente di effettuare una ricerca dei fascicoli corrispondenti a criteri di ricerca.
- Ricerca Unità organizzativa (uo) di una postazione: permette di effettuare la ricerca delle informazioni associate ad una postazione conoscendone l'identificativo; viene inoltre restituita anche la postazione "padre".
- Modifica protocollazione: permette di modificare o annullare i campi di una registrazione; se la registrazione non conteneva allegati inoltre permette di inserire in una soluzione "one shot" uno o più allegati. L'ordine di inserimento rispecchierà l'ordine con cui vengono registrati.
- Estrai allegati: data una segnatura di registrazione o un identificativo documento permette di estrarre su file system tutti i suoi files allegati.
- Crea Copia: data una segnatura o un identificativo documento e una unità organizzativa destinataria, permette di creare una copia della registrazione a essa assegnata.

Il sistema di gestione documentale Hummingbird, utilizzato dal protocollo informatico, può essere usato anche da altre applicazioni attraverso i seguenti servizi web:

- Login: consente l'autenticazione nel sistema documentale centrale ed effettua il recupero del token DST di sicurezza.
- Import: consente l'inserimento (e la profilazione) di un nuovo documento nel sistema documentale centrale.
- Export: consente il recupero di un documento (file e metadati) dal sistema documentale centrale.
- Modify: consente la modifica di un documento memorizzato nel sistema documentale centrale.
- Delete: consente la cancellazione di un documento memorizzato nel sistema documentale centrale.
- getDocuments: consente la ricerca di documenti memorizzati nel sistema documentale centrale.

L'infrastruttura di firma digitale presente in Regione Emilia-Romagna dispone di servizi web utilizzabili dalle applicazioni che necessitano l'integrazione. Si elencano in maniera sintetica:

- Firma: consente la firma digitale di un file, restituendo il file firmato secondo lo standard PKCS#7.
- Verifica di un file firmato: consente la verifica della firma apposta ad un file restituendo l'esito della verifica, i certificati dei firmatari e i dati originari recuperati dalla verifica.
- Cifra: consente di cifrare un file restituendo un file cifrato secondo lo standard

PKCS#7.

- Decifra: consente di decifrare un file cifrato secondo lo standard PKCS#7.
- Verifica di una marca temporale: consente di verificare l'integrità di una marca temporale e di controllare la credibilità e la validità del certificato del firmatario.
- Firma e verifica di un documento XML: consente di firmare un file XML nel formato XMLSignature, secondo lo standard: XML-Signature Syntax and Processing. Consente inoltre di verificare l'integrità di un file XML firmato.

La funzione di firma di documenti è usufruibile anche sotto forma di pagina web da integrare con altre applicazioni al fine di realizzare la firma web dei file stessi, restituendo all'applicazione chiamante il file firmato.

La banca dati relativa a strutture e personale risiede su mainframe. Sono state implementate classi .net e servizi web per la consultazione di alcuni di questi dati. In sintesi:

- Dettaglio persona: consente di effettuare una ricerca per matricola per reperire i della persona, compresi i dati di assegnazione e responsabilità.
- Dettaglio unità funzionale: consente di effettuare una ricerca per codice unità per reperire i dati dell'unità funzionale.
- Elenca unità funzionale: consente di elencare le unità funzionali secondo diversi criteri.
- Elenca persone assegnate all'unità funzionale: consente di elencare le persone assegnate ad una unità funzionale.

Per la documentazione tecnica di dettaglio si faccia riferimento agli Allegati 10, 11, 12, 13.

6.5.a Cooperazione applicativa

Le specifiche **Sistema Pubblico di Cooperazione (SPCoop)** del CNIPA definiscono le linee guida per le amministrazioni che vogliano interagire con i sistemi informativi di altre amministrazioni attraverso cooperazione applicativa.

In tali specifiche un **dominio** è definito come il confine di responsabilità di un ente o soggetto amministrativo e racchiude al suo interno tutte le applicazioni da esso gestite. Il confine applicativo del Dominio è rappresentato dalla **Porta di Dominio (PdD)**, attraverso la quale devono transitare tutte le comunicazioni applicative "formali" da e verso il dominio. A livello concettuale la PdD funge da proxy per l'accesso alle risorse applicative che si trovano all'interno dello stesso dominio.

La PdD ha lo scopo di assicurare che lo scambio elettronico di informazioni tra le Pubbliche Amministrazioni abbia le stesse caratteristiche di formalizzazione di quello tradizionale (carta, firma, protocollo, fax….). In questo modo l'amministrazione che invia le informazioni in modo elettronico ad un'altra, sarà garantita del fatto che la destinataria (e non altri) le abbia ricevute, così come la ricevente potrà trattare le informazioni elettroniche ricevute con pari dignità di quelle che oggi riceve con i metodi tradizionali, considerati fino ad ora gli unici probanti ai fini del procedimento amministrativo. Questo deve essere possibile indipendentemente da come viene realizzata la porta di dominio (fornitore,

linguaggi, tecnologia…) in quanto la sua interfaccia è stata definita formalmente.

L'interoperabilità fra amministrazioni deve svilupparsi attraverso le PdD, sulla base di standard definiti a livello nazionale dal CNIPA, in modo tale che:

- siano identificati i servizi ed i dati che ogni amministrazione decide di rendere disponibili sulla rete;
- siano rispettate, per ogni Servizio applicativo esposto, le politiche di sicurezza, di accesso e di controllo di qualità e correttezza dei servizi erogati, stabilite dall'amministrazione erogante.

A livello concettuale **esiste una sola porta per ogni dominio**. Sono però in via di definizione nuove regole organizzative (da parte del CNIPA) che prevedeno anche aspetti di sussidiarietà in questo ambito.

Per scambiare messaggi applicativi fra PdD viene utilizzata la **busta di eGovernment** che è la definizione del formato di codifica e del contenuto dei messaggi SOAP scambiati tra le porte di dominio. Anche il formato della busta di eGovernement è stato definito dal CNIPA.

Il formato della busta di eGovernment non è “parlato” nativamente dalle applicazioni degli Enti, pertanto la PdD deve anche occuparsi di convertire le richieste applicative nel formato busta eGov. Facendo riferimento a questa problematica, i compiti della PdD vengono solitamente classificati in due componenti: il componente di cooperazione, che riguarda la comunicazione tra le PdD e quello di integrazione, che riguarda la comunicazione tra i Servizi Applicativi dell’Ente e la PdD. Il componente di integrazione si differenzia a sua volta in due diversi moduli: **la porta delegata e la porta applicativa**. In particolare la porta delegata è utilizzata come proxy per l’accesso al Servizio destinazione, mentre la porta applicativa deve essere in grado di gestire la consegna dei contenuti delle buste di eGovernment ricevute al corrispondente Servizio applicativo interno al dominio destinazione.

Lo strumento utilizzato per definire un formato dei dati condiviso tra tutte le amministrazioni, a prescindere dai sistemi "legacy" e dalle basi dati, è XML. SOAP è invece utilizzato come standard per veicolare le informazioni codificate con XML sulla rete Internet, mediante il protocollo HTTP.

Altra “componente” della cooperazione applicativa secondo le specifiche CNIPA, è l’**Accordo di Servizio**, un documento standard totalmente formalizzato in XML che regola il rapporto erogatore/fruitore di un Servizio applicativo in tutte le parti che lo caratterizzano: l’interfaccia, le modalità di interazione, i punti di accesso, i livelli di Servizio e le caratteristiche di sicurezza previste. L’Accordo di Servizio consentirà lo sviluppo armonico di un insieme trasparente di relazioni di Servizio che, sul piano tecnico, consentirà di aumentare sia l’efficienza nell’erogazione dei servizi telematici che il relativo livello di qualità. L’inserimento di un catalogo degli Schemi/Ontologie negli Accordi di Servizio consentirà, inoltre, di descrivere la semantica delle informazioni veicolate e dei servizi stessi.

Il CNIPA, in accordo con le Regioni nell'ambito della Commissione di coordinamento SPC, ha responsabilità della stesura e manutenzione di un set di documenti di specifiche tecniche e organizzative della cooperazione applicativa SPCoop. Questi documenti delineano compiutamente il quadro tecnico-organizzativo del SPCoop.

Tali specifiche rappresentano il riferimento per i piani di convergenza dei progetti infrastrutturali di cooperazione già realizzati o in corso di realizzazione. Tra questi progetti il progetto **ICAR** (al quale partecipano 18 Regioni italiane) rappresenta il motore della convergenza dei progetti regionali di cooperazione applicativa nel SPCoop.

Nel PITER 2007-2009 - Piano Telematico dell'Emilia-Romagna è presente un progetto denominato "**ICAR-ER**". Tale progetto comprende lo svolgimento delle attività della Regione Emilia-Romagna nel contesto del progetto ICAR nazionale e, coerentemente a questo, intende realizzare una infrastruttura di cooperazione applicativa per il territorio regionale, conforme alle specifiche SPCoop. I principali risultati di tale progetto sono:

- la PdD standard SpCoop per l'Ente Regione Emilia-Romagna;
- una PdD standard SpCoop che verrà messa a disposizione per gli Enti del territorio;
- componenti infrastrutturali aggiuntive di supporto alla cooperazione applicativa secondo le specifiche SPCoop.

In coerenza il modello del SPCoop le applicazioni della Regione Emilia-Romagna che prevedono cooperazione applicativa formalizzata fra enti utilizzano detta Porta di Dominio. Il progetto regionale, a tale scopo, mette a disposizione il supporto e la documentazione necessaria per la realizzazione di tali integrazioni.

7. Siti web

Progettare, realizzare e gestire un sito web oggi richiede una serie di attività complesse che coinvolgono figure professionali diverse e sempre più specializzate che seguano tutte le fasi del *ciclo di vita* di un sito:

- **Progettazione**
  Chi intende realizzare un sito web deve innanzitutto definire gli obiettivi del sito e i suoi destinatari, vincoli e requisiti, architettura dei contenuti e struttura del sito, grafica e layout; può essere utile ispirarsi a quanto fatto da altri su siti simili. E' necessario che sin dalla progettazione siano coinvolti i Servizi competenti in materia di siti web e comunicazione. Va definito se il sito web è pubblico oppure riservato (e in tal caso a chi) oppure è interno alla rete intranet regionale.

  Infine, siccome l'Amministrazione mette a disposizione di tutti i settori l'infrastruttura tecnologica necessaria alla gestione di siti web (vedi paragrafo 5), è opportuno motivare adeguatamente il ricorso a hosting presso aziende esterne.

- **Sviluppo**
  Comprende la realizzazione di prototipi (homepage e alcune pagine interne) (eventualmente affidati ad una ditta specializzata oppure utilizzando modelli e impostazioni "*standard*"), la predisposizione dell'ambiente informatico di sviluppo e delle procedure informatiche correlate (da parte dei tecnici), la raccolta e preparazione dei contenuti (da parte dei redattori) e la realizzazione/implementazione del sito.

- **Collaudo**
  Consiste nel verificare contenuti, funzionalità e prestazioni rispetto agli obiettivi e rispetto agli standard tecnologici e alla normativa vigente.

- **Pubblicazione**

Comprende la predisposizione dell'ambiente informatico di produzione e la pubblicazione del sito, con l'attivazione delle procedure correlate (es. indicizzazione da motore di ricerca). Dopo la verifica finale è possibile attivare i link che renderanno effettivamente raggiungibile il sito da parte del pubblico; si evidenzia che la scelta del posizionamento dei link all'interno dei portali regionali influenza la reperibilità del sito nei motori di ricerca e nella rete.

- **Manutenzione e Promozione**
  Comprende sia la manutenzione dei sistemi da parte dei tecnici informatici che la manutenzione e l'aggiornamento dei contenuti da parte dei redattori e del responsabile del sito. Fanno parte delle attività di manutenzione e promozione anche il monitoraggio degli accessi al sito (tramite le statistiche degli accessi web) e del posizionamento nei motori di ricerca.

Per tutti i siti web pubblicati, anche se in hosting su server esterni, valgono le indicazioni fornite al paragrafo **6.3 Accessibilità**, **6.4 Sicurezza**.

7.1 Nomi dei siti

Per definire il nome di un sito web sono percorribili tre strade:

- inserire il sito all´interno di Ermes
- dare maggiore identità al sito all'interno di Ermes
- registrare un nome di dominio

Per inserire il sito all´interno di Ermes non è necessaria nessuna autorizzazione particolare: una volta deciso il nome, il sito apparirà come "sotto-web", con un URL del tipo: http://www.regione.emilia-romagna.it/nomesito

Per dare maggiore identità all'interno di Ermes si può scegliere un URL del tipo http://nomesito.regione.emilia-romagna.it; in questo caso è necessario inviare una richiesta all´indirizzo e-mail: **mailto:LirEMR@regione.emilia-romagna.it**

L'ultima e più complessa possibilità è quella di registrare un dominio, un URL del tipo http://www.nomesito.it; per registrare un dominio .it si deve fare richiesta alla casella di posta elettronica **LirEMR@regione.emilia-romagna.it,** mentre per gli altri namespace diversi da .it (.org, .eu, ecc.) la Regione Emilia-Romagna non ha funzioni di provider/maintainer quindi chi fosse interessato deve registrarlo autonomamente.

È buona norma individuare il nome in modo che abbia attinenza all´argomento di cui il sito si occupa e che sia chiaro ed inequivocabile.

Per informazioni più dettagliate e per reperire i moduli da allegare alle richieste, consultare le "Linee Guida per realizzare siti e applicazioni web accessibili per la Regione Emilia-Romagna" reperibili online all'indirizzo:

http://www.regione.emilia-romagna.it/lineeguida/

7.2 Il sistema di Web Content Management (WCM)

Il modo più efficace di realizzare e gestire un sito web è tramite il sistema di Web Content Management regionale.

Un CMS -o WCM- (in italiano "sistema di gestione dei contenuti web") è un insieme di metodi e tecniche per automatizzare i processi di creazione, gestione e pubblicazione di contenuti attraverso il web.

Esso si fonda sulla logica della separazione tra i contenuti informativi delle pagine e la forma con cui tali contenuti vengono presentati, consentendo agli autori di inserire i contenuti senza preoccuparsi di come questi verranno resi sullo schermo dell'utente, perché la logica di presentazione viene gestita separatamente da specialisti e tecnici indipendentemente dal numero di pagine a cui andrà applicata e dalle informazioni che in esse verranno visualizzate.

Si consiglia per tutti i nuovi siti l'utilizzo del sistema wcm; altre modalità vanno eventualmente concordate in sede di progettazione, perché tutte le attività di realizzazione sono influenzate da questa scelta.

Quando si progetta un sito wcm, il referente tecnico del progetto guida il referente del sito nelle seguenti attività:

- definizione dell'architettura del sito, della struttura delle pagine e delle tipologie di contenuti che dovrà gestire;
- definizione delle "proprietà" dei contenuti che andranno inseriti (es.: una notizia ha un titolo, una data, un'immagine, un testo ed eventualmente dei link);
- definizione delle regole per la visualizzazione di ogni tipologia di contenuto (es. le news vanno ordinate cronologicamente, in homepage vanno solo le 3 più recenti, ecc.);
- definizione della "grafica": a partire dalla struttura delle pagine e dei contenuti, si applicano le regole per visualizzarli (tipicamente, con uno o più css).

La Regione Emilia-Romagna adotta il CMS Open Source "Plone", un cms di nuova generazione e con funzionalità avanzate, con il quale sono realizzati i nuovi siti web: il sistema di Content Management è gestito direttamente dal Servizio SIIR.

Per coloro che non hanno particolari esigenze grafiche e architetturali è possibile realizzare in tempi rapidi e senza particolari conoscenze tecniche semplici siti web con una struttura "standard" predefinita, personalizzabile nell'aspetto relativamente a loghi e colori.

7.3 Motori di ricerca

I motori di ricerca sono lo strumento più utilizzato dai navigatori per cercare le pagine a cui sono interessati.

Per questo la Regione Emilia-Romagna utilizza il motore di ricerca Google per consentire di effettuare ricerche testuali nelle pagine e nei documenti presenti sui siti regionali.

Il funzionamento dei motori di ricerca consiste nel memorizzare in un database (*indicizzazione*) le informazioni contenute nelle pagine di un sito partendo da una pagina prefissata (per es. la homepage di ERMES) finché non ha indicizzato tutte le pagine del sito; dopodiché il motore di ricerca, quando cerchiamo una o più parole, consulta il database e restituisce l'elenco delle pagine indicizzate in cui sono presenti le parole cercate, ordinate secondo particolari criteri che ne determinano la "*pertinenza*" rispetto alle parole cercate.

Pertanto è molto importante che le pagine che noi riteniamo più significative escano per prime tra i risultati delle ricerche degli utenti.

Per fare ciò è necessario seguire alcuni accorgimenti nella costruzione delle pagine e nella organizzazione dei contenuti all'interno delle pagine; per maggiori informazioni consultare le "Linee Guida per realizzare siti e applicazioni web accessibili per la Regione Emilia-Romagna" reperibili online all'indirizzo:

http://www.regione.emilia-romagna.it/lineeguida/

Si raccomanda di rivolgersi al Servizio SIIR per attivare l'indicizzazione del proprio sito e per consentire la ricerca all'interno del sito e/o in una sua porzione.

7.4 Statistiche di accesso ai siti

Le analisi statistiche dei dati di traffico relativi ai siti internet consentono di determinare il grado di fruizione del sito stesso. Per i gestori di un sito rappresentano una misura per valutare il successo del sito o di una sua sezione oppure per rilevare le criticità nella fruizione di alcune pagine.

Dalle analisi statistiche degli accessi web è possibile monitorare il comportamento del visitatore: quante volte ha visitato il sito, quali sono le pagine che ha visto, il tempo di permanenza in quelle pagine, come è arrivato sul sito, quali sono le keywords (parole chiave) che ha utilizzato per cercarlo con i motori di ricerca, quali sono i percorsi maggiormente utilizzati e quali meno. Queste informazioni risultano estremamente utili al fine di valutare se la struttura delle pagine e la sequenza di navigazione facilitano o scoraggiano il visitatore alla permanenza sul sito e a comprendere quali sono i contenuti ritenuti più interessanti e quindi da valorizzare e quali invece da eliminare o modificare.

In Regione Emilia-Romagna i dati di traffico sui server web regionali vengono registrati su file di log ed elaborati da un programma di analisi statistica degli accessi web che produce dei report per ogni sito web.

L'attivazione dell'analisi statistica degli accessi ad un sito o porzione di sito va richiesta al Servizio SIIR da parte del responsabile del sito. Per maggiori informazioni consultare le "Linee Guida per realizzare siti e applicazioni web accessibili per la Regione Emilia-Romagna" reperibili online all'indirizzo:

http://www.regione.emilia-romagna.it/lineeguida/

8. Servizi e strumenti già disponibili

Internet ed il Web, oltre a contenere pagine informative e documenti da scaricare, possono diventare luoghi di scambio e di lavoro.

A questo scopo, il Servizio SIIR mette a disposizione una serie di strumenti (conformi alla normativa sulla privacy e sull'accessibilità) personalizzabili per le più diverse esigenze, che si consiglia di usare invece di ricorrere a servizi esterni:

1. Forum
2. Newsletter (gestione iscritti e invio newsletter)
3. Iscrizione online a convegni

4. Sondaggi e questionari
5. Groupware (ambienti di lavoro collaborativo)

I responsabili di un sito interessati ad attivare uno o più di questi strumenti devono farne apposita richiesta al Servizio SIIR.

Per una descrizione estesa di questi strumenti vedere l'Allegato 15; maggiori informazioni sono inoltre disponibili nelle "Linee Guida per realizzare siti e applicazioni web accessibili per la Regione Emilia-Romagna" reperibili online all'indirizzo:

http://www.regione.emilia-romagna.it/lineeguida/

8.1 Ftps

La Regione Emilia-Romagna è parte attiva in vari progetti di collaborazione con gli enti pubblici, il mondo delle imprese ed i professionisti. Talora la collaborazione prevede lo scambio di file di grosse dimensioni, come ad esempio file multimediali o grafici.

In questi casi è necessario disporre di uno strumento che garantisca l'efficienza, la sicurezza e la protezione del trasferimento delle informazioni.

A questo scopo il Servizio SIIR mette a disposizione degli utenti un *server ftps*, ovvero un server che garantisce lo scambio crittografato delle informazioni, in cui ad ogni gruppo viene assegnata un'area riservata, composta da una o più cartelle dove si possono caricare e prelevare i file tramite uno strumento per la gestione del trasferimento sicuro dei file chiamato FileZilla, che deve essere installato e configurato sul pc degli utenti.

Procedure

9. Acquisizione di prodotti/servizi IT

Le Direzioni generali nell'ambito della loro autonomia finanziaria, anche mediante il concorso di fondi non provenienti dal Bilancio regionale, realizzano segmenti del sistema informativo regionale, acquistando prodotti e/o servizi IT.

Al fine di garantire l'unitarietà del sistema informativo regionale nel suo complesso, in accordo con quanto previsto dalla DGR 999/2008, allegato D, le Direzioni generali trasmettono i relativi programmi di acquisizione alla Direzione generale centrale Organizzazione, personale, sistemi informativi e telematica alla quale spetta individuare possibili sinergie e la definizione di politiche di acquisto comune.

La Direzione generale competente in materia di sistemi informativi e telematica (nel seguito *Direzione generale SIT*), attraverso queste Linee guida, individua gli standard metodologici e tecnologici di riferimento, a cui i fornitori dovranno attenersi, e che quindi dovranno essere richiamati nei capitolati tecnici o nei contratti, e inoltre ha il compito di effettuare le necessarie verifiche, attraverso le sue strutture.

Le verifiche sono di tre tipologie:

1. **preventive progettuali**: da effettuare sulla documentazione di gara o sul contratto a monte dell'attivazione della procedura di acquisto o dell'affidamento diretto;
2. **preliminari alla presa in carico**: da effettuare sulla documentazione tecnica (a valle dell'analisi tecnica se si tratta di sistema informativo), in particolare sui requisiti/fabbisogni hardware e software;
3. **preliminari al rilascio in produzione**: da effettuare a valle dell'acquisizione e dell'implementazione e sono propedeutiche al passaggio in produzione.

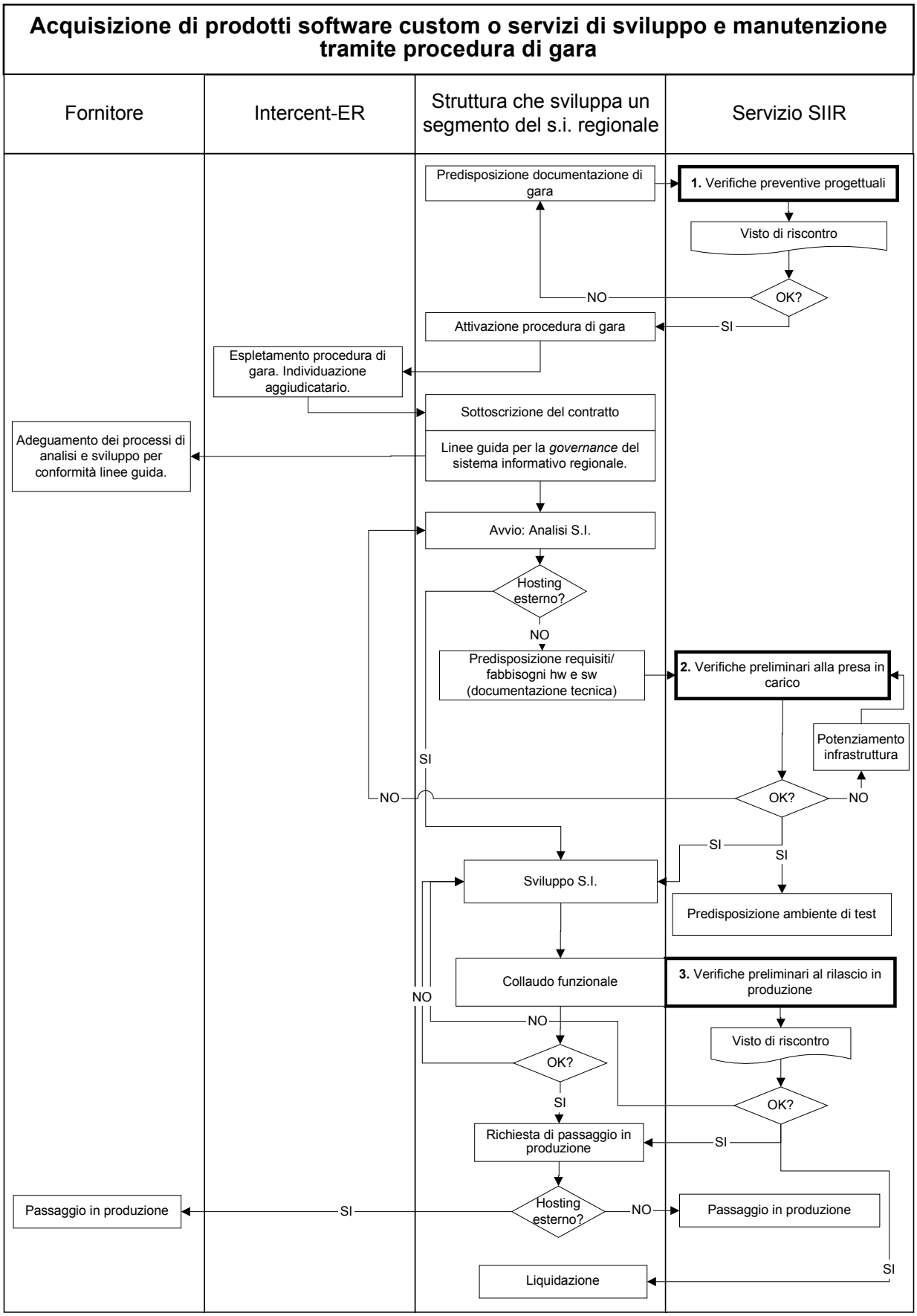
L'esito delle verifiche viene espresso attraverso un visto preliminare di riscontro di congruenza tecnica.

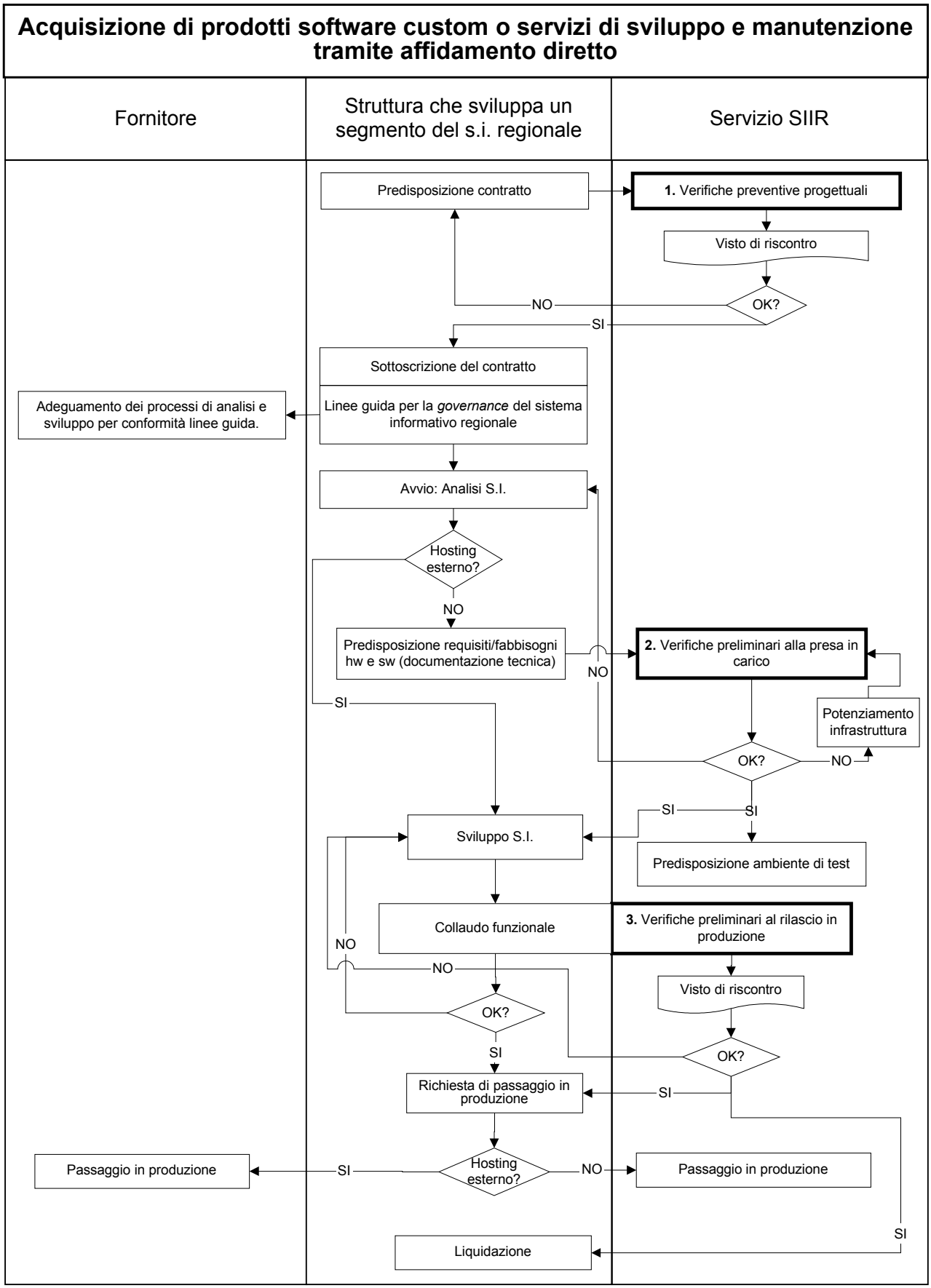
Le possibili casistiche di acquisizione e realizzazione di prodotti/servizi IT con o senza procedura di gara sono le seguenti:

- acquisizione di prodotti software custom o servizi di sviluppo e manutenzione tramite procedura di gara,
- acquisizione di prodotti software custom o servizi di sviluppo e manutenzione tramite affidamento diretto,
- realizzazione di prodotti software mediante ricorso a servizi di sviluppo e manutenzione già acquisiti,
- acquisizione, senza procedura di gara, di prodotti software nella forma proposta dal venditore, sia in proprietà che in licenza d'uso, onerosa o gratuita,
- adozione di prodotti software proposti e concessi in riuso da altri Enti,
- realizzazione di prodotti software da parte di personale regionale
- acquisizione di prodotto hardware tramite procedura di gara/mercato elettronico,

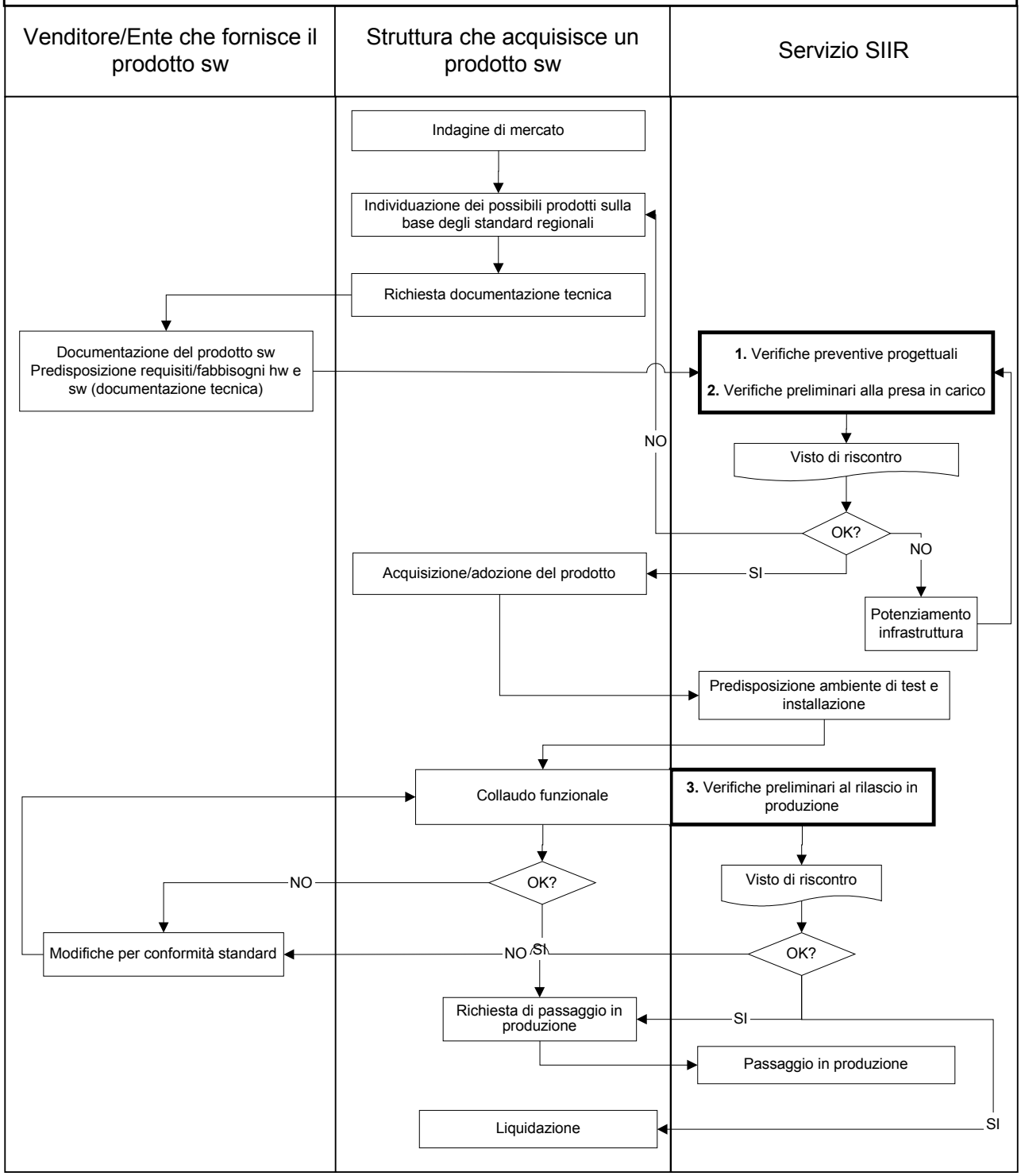
- acquisizione di prodotto hardware tramite affidamento diretto.

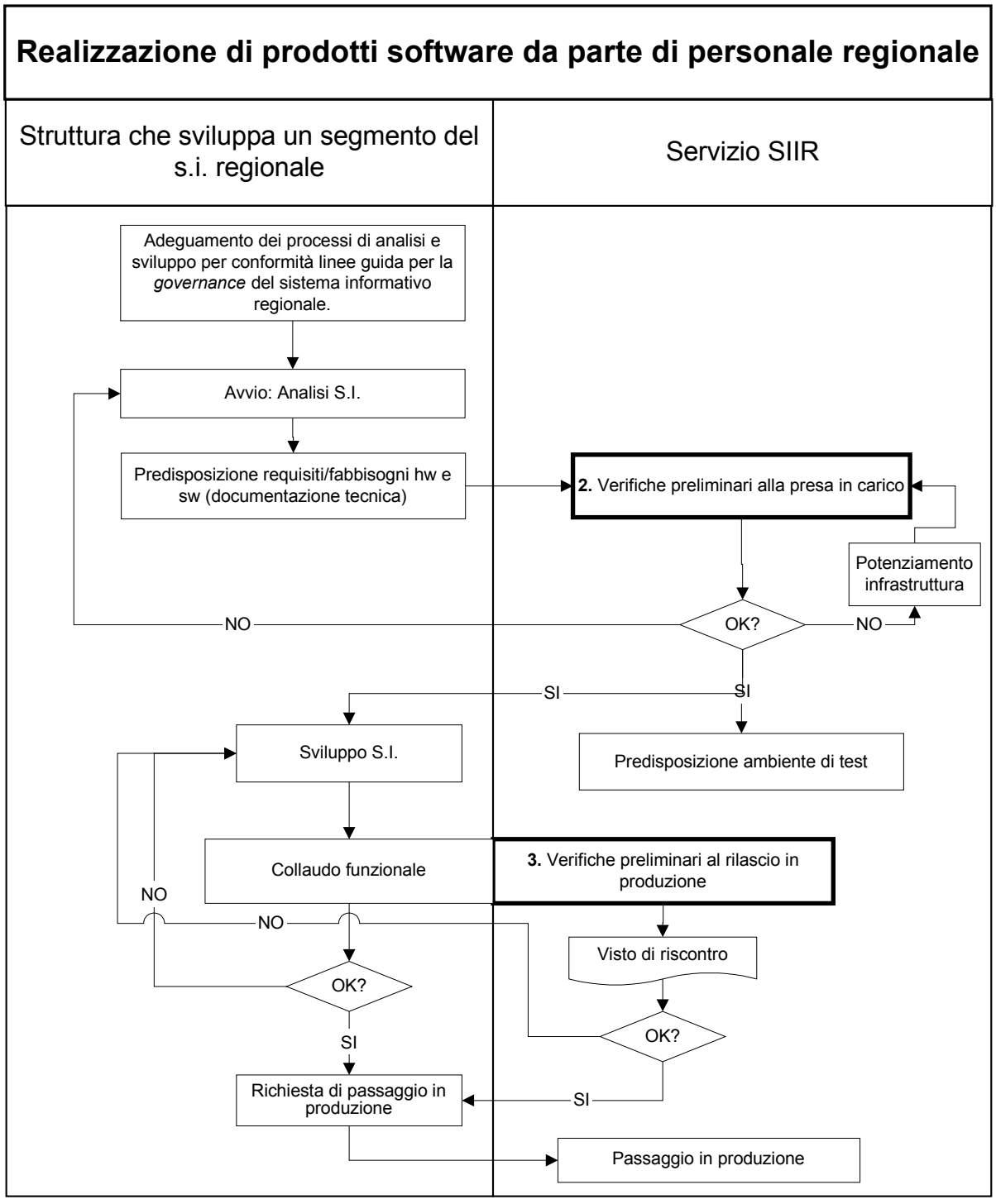
Si analizzano nel dettaglio nelle pagine seguenti per evidenziare le fasi e gli attori coinvolti.

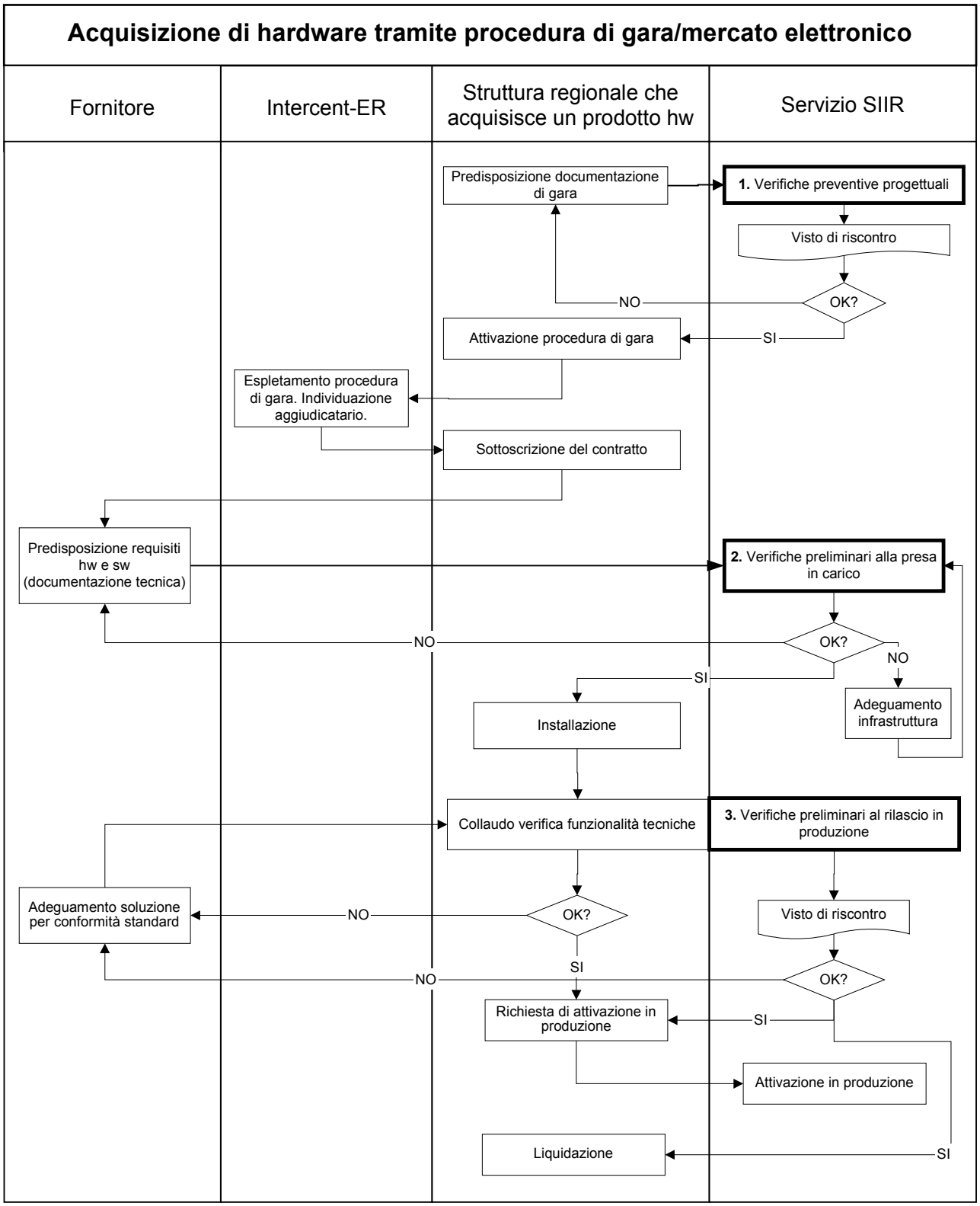

Acquisizione di prodotti software custom o servizi di sviluppo e manutenzione tramite procedura di gara
Fornitore
Intercent-ER
Struttura che sviluppa un segmento del s.i. regionale
Servizio SIIR
Predisposizione documentazione di gara
1. Verifiche preventive progettuali
Visto di riscontro
OK?
NO
SI
Attivazione procedura di gara
Espletamento procedura di gara. Individuazione aggiudicatario.
Sottoscrizione del contratto
Adeguamento dei processi di analisi e sviluppo per conformità linee guida.
Linee guida per la governance del sistema informativo regionale.
Avvio: Analisi S.I.
Hosting esterno?
NO
Predisposizione requisiti/ fabbisogni hw e sw (documentazione tecnica)
2. Verifiche preliminari alla presa in carico
Potenziamento infrastruttura
SI
NO
OK?
NO
SI
SI
Sviluppo S.I.
Predisposizione ambiente di test
Collaudo funzionale
3. Verifiche preliminari al rilascio in produzione
NO
NO
Visto di riscontro
OK?
OK?
SI
SI
Richiesta di passaggio in produzione
Passaggio in produzione
SI
Hosting esterno?
NO
Passaggio in produzione
SI
Liquidazione

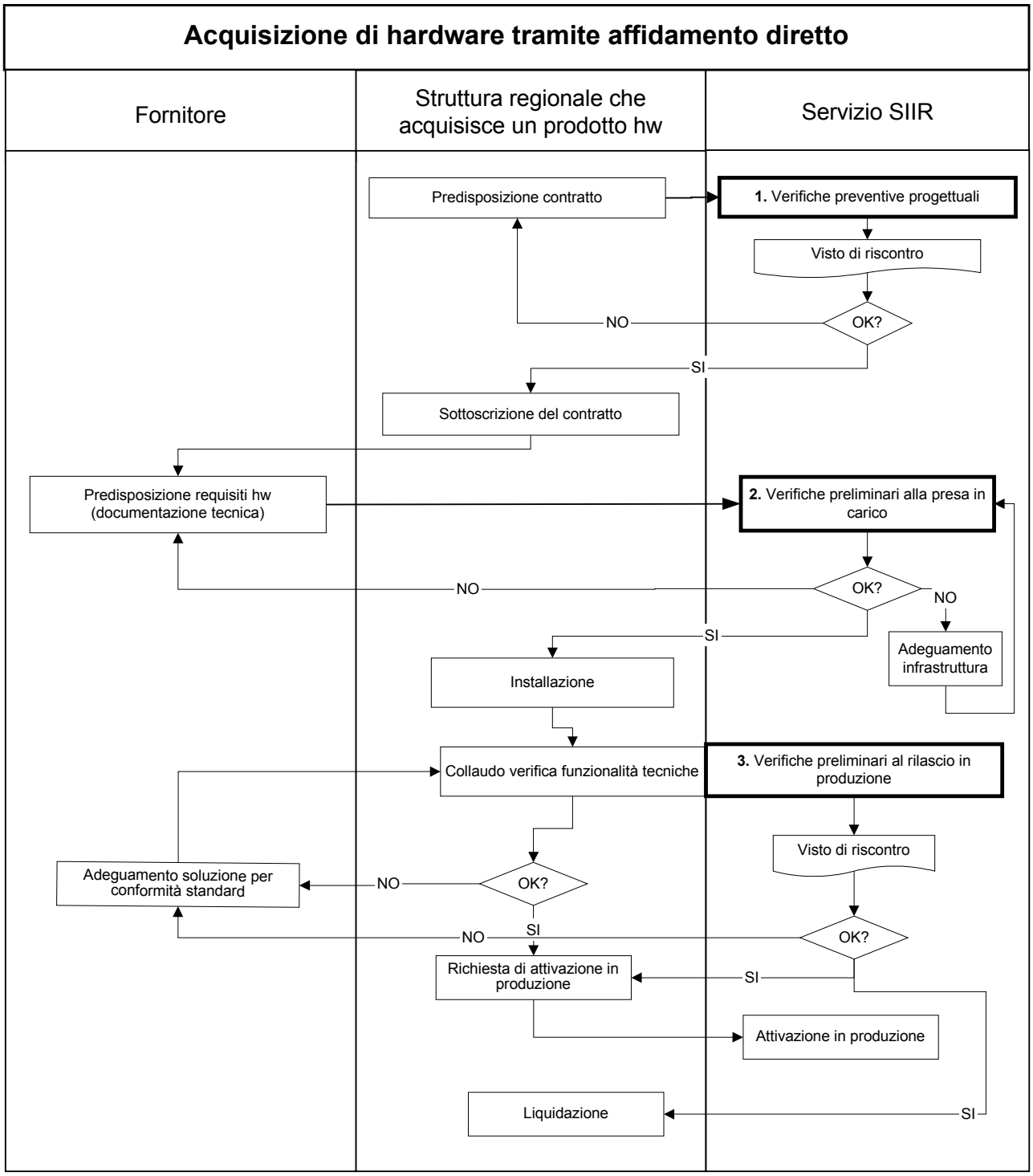

Acquisizione di prodotti software custom o servizi di sviluppo e manutenzione tramite affidamento diretto
Fornitore
Struttura che sviluppa un segmento del s.i. regionale
Servizio SIIR
Predisposizione contratto
1. Verifiche preventive progettuali
Visto di riscontro
OK?
NO
SI
Sottoscrizione del contratto
Adeguamento dei processi di analisi e sviluppo per conformità linee guida.
Linee guida per la governance del sistema informativo regionale
Avvio: Analisi S.I.
Hosting esterno?
NO
Predisposizione requisiti/fabbisogni hw e sw (documentazione tecnica)
2. Verifiche preliminari alla presa in carico
NO
SI
Potenziamento infrastruttura
OK?
NO
SI
SI
Sviluppo S.I.
Predisposizione ambiente di test
Collaudo funzionale
3. Verifiche preliminari al rilascio in produzione
NO
NO
Visto di riscontro
OK?
SI
OK?
Richiesta di passaggio in produzione
SI
Passaggio in produzione
SI
Hosting esterno?
NO
Passaggio in produzione
SI
Liquidazione

Acquisizione, senza procedura di gara, di prodotti sw nella forma proposta dal venditore, sia in proprietà che in licenza d'uso, onerosa o gratuita. Riuso.




Realizzazione di prodotti software da parte di personale regionale
Struttura che sviluppa un segmento del s.i. regionale
Servizio SIIR
Adeguamento dei processi di analisi e sviluppo per conformità linee guida per la governance del sistema informativo regionale.
Avvio: Analisi S.I.
Predisposizione requisiti/fabbisogni hw e sw (documentazione tecnica)
2. Verifiche preliminari alla presa in carico
Potenziamento infrastruttura
OK?
NO
NO
SI
SI
Sviluppo S.I.
Predisposizione ambiente di test
Collaudo funzionale
3. Verifiche preliminari al rilascio in produzione
NO
NO
Visto di riscontro
OK?
OK?
SI
Richiesta di passaggio in produzione
SI
Passaggio in produzione


Acquisizione di hardware tramite procedura di gara/mercato elettronico
Fornitore
Intercent-ER
Struttura regionale che acquisisce un prodotto hw
Servizio SIIR
Predisposizione documentazione di gara
1. Verifiche preventive progettuali
Visto di riscontro
OK?
NO
SI
Attivazione procedura di gara
Espletamento procedura di gara. Individuazione aggiudicatario.
Sottoscrizione del contratto
Predisposizione requisiti hw e sw (documentazione tecnica)
2. Verifiche preliminari alla presa in carico
OK?
NO
SI
Adeguamento infrastruttura
Installazione
Collaudo verifica funzionalità tecniche
3. Verifiche preliminari al rilascio in produzione
Adeguamento soluzione per conformità standard
NO
OK?
Visto di riscontro
SI
NO
OK?
Richiesta di attivazione in produzione
SI
Attivazione in produzione
Liquidazione
SI


Acquisizione di hardware tramite affidamento diretto
Fornitore
Struttura regionale che acquisisce un prodotto hw
Servizio SIIR
Predisposizione contratto
1. Verifiche preventive progettuali
Visto di riscontro
OK?
NO
SI
Sottoscrizione del contratto
Predisposizione requisiti hw (documentazione tecnica)
2. Verifiche preliminari alla presa in carico
OK?
NO
NO
SI
Adeguamento infrastruttura
Installazione
Collaudo verifica funzionalità tecniche
3. Verifiche preliminari al rilascio in produzione
Visto di riscontro
Adeguamento soluzione per conformità standard
NO
OK?
NO
SI
OK?
Richiesta di attivazione in produzione
SI
Attivazione in produzione
Liquidazione
SI

9.1 Verifica preventiva progettuale

Le Direzioni generali che acquisiscono prodotti e servizi IT, preliminarmente all'adozione degli atti e, in caso di gara, all'invio della documentazione a Intercent-ER, trasmettono al Servizio SIIR la documentazione tecnica necessaria alla valutazione. Tale documentazione include, ma non si limita a, i seguenti elementi:

- capitolati tecnici,
- documentazione di prodotto dichiarata dal fornitore,
- protocollo/convenzione per il riuso di soluzione software e/o documentazione di prodotto fornita dall'Ente che concede in riuso,
- contratti per il conferimento di incarichi.

9.1.a Espressione visto di riscontro di congruenza tecnica

Il Servizio SIIR è responsabile dello svolgimento delle verifiche preventive progettuali in merito al rispetto degli standard definiti in materia di tecnologie, metodologie di sviluppo e documentazione, livelli minimi di sicurezza e accessibilità, attraverso l'espressione di un visto preliminare di riscontro di congruenza tecnica.

L'espressione del visto avviene:

a. sui capitolati tecnici per procedure di acquisizione di prodotti software (progetti a corpo), ivi inclusi i siti web e i prodotti multimediali distribuiti su supporto fisico;
b. sui capitolati tecnici per procedure di acquisizione di servizi di sviluppo e manutenzione;
c. sulle specifiche tecniche di prodotti software che si intende acquisire nella forma proposta dal venditore, sia in proprietà che in licenza d'uso, onerosa o gratuita;
d. sui protocolli/convenzioni per il riuso di soluzione software e/o sulle specifiche tecniche di prodotti software proposti e concessi in riuso da altri Enti;
e. sui contratti per acquisire, tramite affidamento diretto, prodotti software o servizi di sviluppo e manutenzione da fornitori specifici;
f. sulle specifiche tecniche di apparecchiature hardware che le Direzioni generali intendono acquisire al di fuori delle procedure generali dell'Ente.

entro un arco temporale che può variare da un minimo di 10 giorni a un massimo di 30 giorni (in funzione della documentazione da visionare) dalla ricezione dei capitolati o delle specifiche oggetto di esame. L'invio della documentazione e l'espressione del visto potranno essere effettuati per via telematica. L'espressione del visto viene registrato sul sistema di protocollo.

Il visto è necessario per tutte le tipologie di prodotti software e hardware, ad eccezione delle **"piccole periferiche"** quali ad esempio mouse, monitor, pen drive USB, tastiere USB, Hub USB, moduli di memoria RAM aggiuntivi, dischi esterni USB, scanner dotati di programmi di scansione a corredo.

Qualora il visto di riscontro di congruenza tecnica evidenzi carenze o requisiti non rispondenti agli standard definiti, la Direzione generale acquirente è tenuta ad adeguare le specifiche secondo quanto indicato dal Servizio SIIR e a richiedere nuovamente la valutazione della documentazione fino a quando non venga formulato un visto di riscontro positivo di congruenza tecnica.

Le Direzioni generali danno atto dell'avvenuta espressione del visto positivo, citandone gli estremi, negli atti di indizione di gara, di avvio di procedura selettiva e di affidamento diretto.

9.2 Verifica preliminare alla presa in carico

Le Direzioni generali che acquisiscono prodotti e servizi IT o che li realizzano con proprio personale, inviano al Servizio SIIR la documentazione tecnica necessaria alla verifica preliminare alla presa in carico sui requisiti/fabbisogni hardware e software secondo lo schema in Allegato 14. Tale verifica risulta essere propedeutica alla predisposizione degli ambienti, su cui il prodotto hardware e/o software verrà installato, all'interno dell'infrastruttura regionale.

9.2.a Espressione visto di riscontro di congruenza tecnica

Il Servizio SIIR è responsabile dello svolgimento delle verifiche preliminari alla presa in carico sui requisiti/fabbisogni hardware e software, attraverso l'espressione di un visto preliminare di riscontro di congruenza tecnica.

L'espressione del visto avviene:

a. sulle specifiche di realizzazione di sottosistemi informativi realizzati dalle Direzioni mediante ricorso a specifica gara o ad affidamento diretto;
b. sulle specifiche di realizzazione di sottosistemi informativi realizzati dalle Direzioni mediante ricorso a servizi di sviluppo software precedentemente acquisiti;
c. sulle specifiche di realizzazione di sottosistemi informativi realizzati da personale regionale delle Direzioni;
d. sulla documentazione tecnica del prodotto software fornita dal Venditore che lo commercializza o dall'Ente che lo propone in riuso;
e. sulla documentazione tecnica delle apparecchiature hardware esibita dal fornitore.

entro un arco temporale che può variare da un minimo di 10 giorni a un massimo di 30 giorni dalla ricezione della richiesta corredata della documentazione tecnica (Allegato 14) necessaria alla verifica. L'invio della documentazione e l'espressione del visto potranno essere effettuati per via telematica. L'espressione del visto viene registrato sul sistema di protocollo.

Qualora il visto di riscontro di congruenza tecnica evidenzi carenze o requisiti non rispondenti agli standard definiti, la Direzione generale acquirente è tenuta ad adeguare le specifiche secondo quanto indicato dal Servizio SIIR e a richiedere nuovamente la valutazione della documentazione fino a quando non venga formulato un visto di riscontro positivo di congruenza tecnica.

Solo in seguito ad una valutazione positiva la Direzione generale acquirente potrà procedere nell'implementazione del sistema secondo le specifiche approvate e il Servizio SIIR potrà procedere nella predisposizione degli ambienti, su cui il prodotto hardware e/o software verrà installato, all'interno dell'infrastruttura regionale al fine di effettuare i necessari collaudi.

9.3 Verifica preliminare al rilascio in produzione

Le Direzioni generali che acquisiscono prodotti e servizi IT o che li realizzano con proprio

personale, effettuato il collaudo funzionale del sistema, inviano al Servizio SIIR la richiesta di rilascio in produzione abilitando all'accesso il/i referente/i incaricato/i di effettuare le verifiche e fornendo le indicazione e i chiarimenti che dovessero essere necessari, nonché tutta la documentazione relativa al sistema da attivare in produzione (in caso di software basarsi sullo schema dell'Allegato 14).

9.3.a Espressione visto di riscontro di congruenza tecnica

Il Servizio SIIR è responsabile dello svolgimento delle verifiche preliminari al rilascio in produzione di prodotti e sottosistemi realizzati da terzi, sul rispetto degli standard definiti in materia di tecnologie, metodologie di sviluppo e documentazione, livelli minimi di sicurezza e accessibilità, attraverso l'espressione di un visto preliminare di riscontro di congruenza tecnica.

L'espressione del visto avviene:

a. sul rispetto degli standard definiti (filiere applicative, documentazione, sviluppo), per i prodotti ospitati sui sistemi regionali;
b. sugli aspetti inerenti la sicurezza (analisi dei rischi, meccanismi di sicurezza implementati, misure minime di sicurezza qualora il sistema tratti dati personali);
c. sugli aspetti inerenti l'accessibilità (rispetto dei criteri di accessibilità definiti dalla Legge n. 4 del 9 Gennaio 2004 e dal DM 8 agosto 2005 e delle indicazioni contenute nelle “Linee Guida per realizzare siti e applicazioni web accessibili per la Regione Emilia-Romagna”);
d. sugli aspetti di performance valutando caso per caso, ed in funzione delle caratteristiche dell'applicazione stessa, l’opportunità dell’esecuzione dei test di carico prestazionali;
e. sulla congruenza delle apparecchiature hardware con l'infrastruttura regionale preesistente.

entro un arco temporale che può variare da un minimo di 10 giorni a un massimo di 30 giorni dalla ricezione della richiesta di rilascio in produzione corredata della documentazione tecnica e delle abilitazioni necessarie alla verifica. L'invio della documentazione e l'espressione del visto potranno essere effettuati per via telematica. L'espressione del visto viene registrato sul sistema di protocollo.

Qualora il visto di riscontro di congruenza tecnica evidenzi carenze o requisiti non rispondenti agli standard definiti, la Direzione generale acquirente è tenuta ad adeguare il sistema secondo quanto indicato dal Servizio SIIR e a richiedere nuovamente la valutazione fino a quando non venga formulato un visto di riscontro positivo di congruenza tecnica.

Le Direzioni generali danno atto dell'avvenuta espressione del visto, citandone gli estremi, negli atti di liquidazione delle fatture a saldo relative ai prodotti e servizi oggetto di valutazione.

10. Realizzazione di sottosistemi informativi e siti web da parte della DG COPSIT

Le Direzioni generali e i Servizi che, per raggiungere gli obiettivi del proprio programma di lavoro, necessitano di realizzare sistemi informativi e/o siti web ricorrendo alla collaborazione del Servizio SIIR, sono tenuti a fare richiesta al Responsabile del Servizio

SIIR in modo tale da programmare almeno trimestralmente le attività, individuando priorità, tempi e risorse.

Dotazioni

11. Attrezzature individuali

Poiché le informazioni e le attrezzature informatiche sono fattori critici per il successo dell'Ente e per la qualità dei processi amministrativi, devono essere protette contro perdite, alterazioni o distruzioni. È necessario inoltre che la disponibilità degli strumenti sia improntata a principi di equità, di razionalizzazione e contenimento dei costi, adottando criteri che verranno di seguito specificati e rispetto ai quali verranno promosse graduali azioni di armonizzazione.

Nel governo di tali strumenti, sotto il profilo hardware e software, ai referenti informatici di Direzione è assegnato un ruolo di coordinamento organizzativo, di presidio di eventuali competenze specifiche o settoriali e di raccordo con la struttura centrale; i referenti costituiscono dunque un interlocutore privilegiato ed hanno come impegno prioritario da un lato quello di segnalare in modo tempestivo ed il più possibile pianificato le esigenze evolutive in campo tecnologico della propria Direzione, dall'altro quello di presidiare la corretta assegnazione delle risorse nel tempo, alla luce delle priorità della Direzione stessa, in particolare, e dell'Amministrazione in generale.

11.1 Assegnazione

A ciascun collaboratore in Servizio dell'Ente, viene assegnato di norma uno strumento informatico individuale (personal computer desktop) collegato alla rete informatica aziendale, qualora la tipologia delle mansioni assegnate o gli aspetti logistici non ostino. Tale strumento sarà unico: chi quindi, per comprovate esigenze di lavoro, necessiti di uno strumento mobile (pc portatile notebook) dovrà rinunciare alla postazione desktop. Eventuali deroghe potranno essere autorizzate dal Servizio SIIR previa richiesta del Direttore competente.

Ai collaboratori regionali inseriti nel progetto "Telelavoro" l'Amministrazione fornisce un collegamento "telefonia e dati" che consenta l'accesso sicuro ai servizi della rete regionale Intranet dal domicilio scelto per il telelavoro, un personal computer portatile per il lavoro sia da casa che dall'ufficio, un telefono ed un modem o un router per il collegamento. Qualora fosse necessario per lo svolgimento delle attività, il portatile potrà essere corredato da eventuali periferiche aggiuntive (ad es. monitor eserni, stampanti, tastiere con tastierino numerico ecc.) che sarà possibile lasciare a casa per tutta la durata del contratto. Tutta la strumentazione assegnata dovrà essere riconsegnata al termine del telelavoro, per essere eventualmente sostituita con la strumentazione standard.

A ciascuna struttura (Servizio o Direzione Generale o Agenzia o Segreteria di Assessore) sarà possibile assegnare un personal computer portatile notebook che potrà essere utilizzato da collaboratori della struttura stessa secondo necessità per convegni/riunioni/missioni e che verrà formalmente assegnato al Responsabile di struttura o di Segreteria.

A ciascuna Direzione Generale o Segreteria di Assessore sarà possibile assegnare due ulteriori personal computer portatili, di cui uno preferibilmente per il Direttore o l'Assessore.

Gli assegnatari a vario titolo di personal computer portatili sono tenuti a riconsegnarli alla

struttura informatica centrale nel momento in cui decadano le motivazioni che hanno determinato l'assegnazione affinché possano essere utilizzati per altre comprovate necessità.

Lo strumento assegnato è conforme agli standard adottati e supportati dall' Ente, si basa su sistemi operativi Microsoft, è corredato da un prodotto di Office Automation standard (Microsoft Office Standard o Professional nelle quantità e nelle versioni acquistate e disponibili), software antivirus e di eventuali prodotti software aggiuntivi necessari per i progetti core dell'Ente (ad esempio SAP, Host on Demand, Adobe Acrobat, SAS, ...) la cui diffusione e gestione, governata da appositi gruppi di progetto, è commisurata alle effettive necessità di utilizzo ed alla disponibilità delle licenze acquisite dagli stessi. Eventuali evoluzioni nelle piattaforme software adottate e supportate saranno studiate, vagliate e proposte a cura della struttura informatica centrale previo coinvolgimento del settore responsabile della formazione informatica.

Eventuali prodotti di informatica individuale aggiuntivi dovranno essere richiesti e concordati preventivamente per permetterne l'acquisto nei casi di provata necessità ed in base alle disponibilità economiche. Verrà comunque data prioritariamente risposta ad esigenze che si riconducano a prodotti già conosciuti e disponibili ed il più possibile standardizzati.

Per quanto riguarda le periferiche di stampa, quanto più possibile si dovrà ricorrere agli strumenti condivisi di corridoio o di staff, fatte salve le necessità di riservatezza o di eventuale impedimento fisico o logistico, per consentire di ridurre i costi eccessivi riscontrati dall'attuale massiccia diffusione di strumenti individuali.

Ulteriori periferiche aggiuntive necessarie (scanner, masterizzatori ecc.) seguiranno un iter di richiesta motivata e di programmazione di acquisto analogo a quello del software aggiuntivo.

Il numero delle periferiche di stampa individuali dovrà progressivamente essere riportato entro il 40% rispetto al numero delle postazioni lavorative installate. Le effettive necessità di dotazione di strumenti informatici individuali sarà indicata dai responsabili delle Strutture e l'adeguamento verrà condotto dalla struttura informatica centrale in accordo con i referenti informatici entro un anno dall'approvazione del presente documento.

Le periferiche di lettura di smart card sono attualmente installate sulle postazioni dei Dirigenti, dei responsabili di Posizione Organizzativa, presso alcune postazioni di lavoro con esigenze specifiche; l'amministrazione è impegnata a diffondere la disponibilità di tali periferiche presso tutte le postazioni utente in rete, procedendo per tranche.

Ulteriori periferiche aggiuntive necessarie a corredo delle postazioni di lavoro seguiranno un iter di richiesta motivata e di programmazione di acquisto analogo a quello del software aggiuntivo.

Si richiama la necessità di porre attenzione all'uso di strumenti di scrittura e salvataggio di dati (masterizzatori, PenDrive ecc.) che da un lato facilitano la condivisione di documenti ma dall'altro presentano, analogamente ai personal computer portatili, problematiche legate alla sicurezza dei dati e di controllo di accesso agli stessi.

11.2 Adempimenti in caso di cessazione del rapporto di lavoro

In caso di cessazione del rapporto di lavoro, il referente informatico della struttura dovrà

essere informato dal lavoratore o dal Responsabile di Servizio, affinché possa programmare le opportune modalità di utilizzo degli strumenti e degli eventuali dati presenti sugli stessi. I dati presenti sulle cartelle personali e sulla casella di posta elettronica individuale saranno eliminati dopo un mese salvo eventuali comunicazioni del diretto interessato che segnalino la necessità di assegnarli ad altro collaboratore regionale.

11.3 Modalità di utilizzo

Le attrezzature individuali assegnate al collaboratore devono essere usate esclusivamente quale supporto all'attività lavorativa in modo pertinente alle specifiche finalità della propria attività, nel rispetto delle esigenze di funzionalità e di sicurezza della rete e dei sistemi. In particolare sulle postazioni di lavoro deve essere usato esclusivamente il software autorizzato e fornito dall'Ente. Eventuale software aggiuntivo ritenuto necessario, sia da acquistare sia disponibile gratuitamente, deve essere richiesto al proprio referente informatico o al proprio dirigente responsabile funzionale; riscontratane l'utilità per le attività di lavoro da parte del dirigente, il referente informatico chiederà il supporto dell'assistenza utenti del Servizio SIIR per la valutazione tecnica del software sotto il profilo della sicurezza, della rispondenza ai requisiti di accessibilità e per programmarne l'eventuale acquisto ed installazione.

Per maggiori dettagli si faccia riferimento al Disciplinare tecnico per utenti sull'utilizzo dei sistemi informativi nella Giunta della Regione Emilia-Romagna approvato determinazione 2653 del 2007.

Come misure preventive verranno individuate *policy* di sistema che blocchino alcune opzioni potenzialmente pericolose per la sicurezza. Le operazioni che l'utente non potrà più compiere in autonomia, qualora necessarie, potranno essere svolte dagli amministratori veri e propri, ed in particolare dai referenti informatici (presso la postazione utente) e dai tecnici dell'assistenza utenti (sia presso l'utente sia in modalità remota).

11.4 Modalità di richiesta

Per le necessità di strumenti che si riconducano alle modalità standard (personal computer desktop o portatile motivato da comprovate esigenze di lavoro, eventuali periferiche, software di uso comune) la modalità di richiesta è inoltrata via posta elettronica dal referente informatico, utilizzando la modulistica reperibile sul sito ComputerAmico e seguendo le istruzioni ivi riportate. La possibilità di evasione ed i relativi tempi sono comunicati dal Servizio SIIR con le medesime modalità.

In caso di necessità di personal computer portatili aggiuntivi alle postazioni necessarie o di strumenti di tipologia particolare (workstation, plotter, applicativi non standard) è necessario venga inoltrata una richiesta motivata indirizzata (con modalità paperless) al Responsabile del Servizio SIIR; la richiesta verrà valutata alla luce delle risorse disponibili. Anche per queste richieste sarà fornita una modulistica su ComputerAmico.

11.5 Modalità di presa in carico di attrezzature

Le attrezzature acquistate in autonomia dalle Direzioni Generali potranno essere utilizzate nella rete dell'Ente solo previa verifica da parte delle strutture tecniche del Servizio SIIR.
La verifica dovrà accertare la rispondenza degli strumenti agli standard tecnologici in uso e

alle policy adottate e la possibilità di gestione da parte dei tecnici dell'assistenza. Gli strumenti dovranno essere resi disponibili per i controlli dei tecnici corredati di tutte le informazioni relative al fornitore, alla tipologia e durata della garanzia.
Qualora il riscontro non fosse positivo o i tempi di presa in carico non potessero essere brevi, gli strumenti dovranno essere utilizzati in modalità non in rete.

11.6 Modalità di supporto ed assistenza

In caso di guasti o malfunzionamenti gli utenti sono invitati ad utilizzare le modalità di segnalazione illustrate sul sito http://intra.regione.emilia-romagna.it/computeramico/index.htm dando preferenza, per quanto possibile, alla segnalazione tramite mail alla casella di posta elettronica Helpdesk@regione.emilia-romagna.it che usufruisce di un percorso privilegiato di evasione, rispetto al numero telefonico 051 659 5850.

Le problematiche verranno prese in considerazione in tempo reale, ma subiranno un vaglio per stabilire le priorità di intervento tecnico secondo i seguenti criteri di priorità:

- guasto bloccante ad una postazione pc di segreteria, ad una casella di posta elettronica di struttura, ad una postazione che necessita di essere operativa per necessità progettuali essendo unica o particolare (ad es. portineria, postazione di economo centrale o periferico, postazione di protocollo, postazione a disposizione del pubblico)
- guasto bloccante a stampante di rete condivisa
- guasto bloccante a strumento o applicazione individuale
- guasto non bloccante

In caso di guasti o malfunzionamenti di strumenti di telelavoro che non possano essere risolti con il supporto telefonico, e salvo impedimenti legati allo stato di salute del telelavoratore, lo strumento da ripristinare dovrà essere riportato a cura del collaboratore interessato presso il presidio Help Desk posto nella sede di Viale Aldo Moro 52 a Bologna o presso la propria sede di lavoro, dove sarà possibile effettuare la riparazione o l'eventuale sostituzione.

In caso di eventuali problemi che si riscontrassero nella gestione di un intervento di malfunzionamento i collaboratori sono tenuti ad informare il referente informatico di riferimento, qualora non fosse già stato avvisato, per una gestione più efficace della problematica insieme ai responsabili dell'assistenza tecnica agli utenti.

12. Networking

In generale, tutte le postazioni di lavoro, al momento dell'installazione, vengono collegate all'infrastruttura di rete a servizio degli uffici regionali.

Le postazioni non connesse sono ormai in numero ridottissimo, e devono esserci motivazioni importanti per configurarle in questo modo, o mantenerle in questa condizione. Ricordiamo che attraverso l'infrastruttura di rete tutte le postazioni di lavoro vengono periodicamente aggiornate con l'installazione degli aggiornamenti di sicurezza e dell'antivirus; le postazioni non connesse richiedono quindi un'attività aggiuntiva e costosa di aggiornamento manuale da parte dei tecnici informatici.

Ogni Personal Computer nasce quindi, di default, come postazione connessa in rete locale

(LAN). Dovunque ciò è possibile, anche le stampanti vengono collegate alla LAN, in modo da permetterne l'uso condiviso tra più utenti; inoltre è possibile garantirne il monitoraggio e l'aggiornamento ai fini della sicurezza da parte dei tecnici, con evidenti economie di gestione.

Le LAN degli uffici regionali sono in genere realizzate in tecnologia Ethernet/Fast Ethernet. Ad oggi – novembre 2007 – le tecnologie Gigabit Ethernet, ancora notevolmente più costose, vengono utilizzate solo in casi particolari: ad esempio per collegare i server presso il CED e nelle interconnessioni di dorsale, all'interno dei palazzi o tra le sedi che fanno parte dello stesso campus (come avviene al Fiera District o tra le varie sedi contigue in viale Silvani).

Tutte le LAN a servizio delle sedi degli uffici sono tra loro connesse attraverso circuiti geografici di varie tecnologie; i circuiti sono monitorati con regolarità per verificarne l'occupazione di banda, e vengono potenziati non appena gli utenti ne presentano la necessità, se sono attivabili circuiti con migliori prestazioni.

Per quanto riguarda le postazioni di lavoro non presenti nelle sedi, la rete regionale consente collegamenti di vario genere per gli utenti remoti, e propone soluzioni particolari per i telelavoratori e gli utenti mobili.

Ricordiamo che l'accesso alla posta elettronica ed alle informazioni personali per i dipendenti è permesso a tutti, da qualsiasi postazione di lavoro connessa ad Internet, attraverso il protocollo di comunicazione criptato https, con le stesse modalità utilizzate presso gli uffici regionali.

12.1 Accesso alla rete per i telelavoratori

Presso l'abitazione dei collaboratori regionali con contratto di telelavoro o con contratto di giornalista, viene attivato, a spese dell'amministrazione, un circuito ADSL interconnesso alla Intranet regionale; l'utente può quindi accedere, anche dalla propria abitazione, a tutti i dati e servizi a cui accede normalmente in ufficio. Per comunicare con i colleghi o gli utenti, vengono anche forniti una normale linea telefonica ed il relativo telefono.

Le Direzioni Generali interessate dovranno inviare al SIIR una comunicazione sollecita, in generale non appena deciso di sottoscrivere un contratto di telelavoro, per consentire la predisposizione di tutto il necessario: infatti i tempi di attivazione delle linee ADSL sono lunghi e spesso imprevedibili, in quanto dipendono dalla disponibilità di risorse sul territorio da parte del fornitore di connettività.

Nel caso nella zona di residenza del telelavoratore il servizio ADSL non sia disponibile, viene attivato, sempre a spese dell'amministrazione, un collegamento in accesso remoto via linea ISDN; sulla seconda linea disponibile sul circuito viene attivato un apparecchio telefonico per comunicare con i colleghi.

Se anche il servizio ISDN non è disponibile in zona, viene attivato un circuito telefonico analogico tradizionale. Sia su ISDN sia su linea analogica commutata vengono attivate configurazioni di accesso remoto (vedi par. 12.3).

12.2 Accesso alla rete via VPN client

Per utenti mobili, o dotati presso la propria abitazione di collegamento ad Internet in banda

larga, è possibile realizzare un collegamento alla Intranet attraverso una configurazione VPN Client in modalità Office Mode.

Entro il 2009 sarà possibile attivare collegamenti alla intranet regionale anche tramite VPN SSL.

Sul Personal Computer dell'utente è necessario installare un apposito software, che dopo l'immissione delle credenziali (la stessa coppia di userid e password normalmente usata per accedere alla LAN) realizza una connessione criptata con il Firewall regionale.

Ricordiamo che i costi relativi al collegamento ad Internet sono a carico dell'utente, e che nessun rimborso potrà essere richiesto alla Regione per questo servizio; pertanto è consigliabile sottoscrivere con il fornitore di connettività un contratto ADSL a banda larga, a tariffa flat, via cavo.

Si sconsiglia l'utilizzo di accessi via schede GPRS/UMTS su PC portatile, in quanto la rete cellulare non è sempre disponibile e le applicazioni utilizzate possono avere problemi in questo ambiente. Va tenuto presente che la tariffa per la telefonia mobile in convenzione Intercenter, permette la navigazione IP a tariffa flat fino a 10 GB di traffico al mese; per tutto il traffico ulteriore, si paga a consumo sulla base del traffico effettuato: è molto facile provocare addebiti di importo rilevante.

In casi particolari, in cui un telelavoratore risieda in zone in cui il servizio ADSL non permette l'interconnessione alla Intranet regionale normalmente utilizzata per il telelavoro, ma solo di accedere ai servizi Internet di base, è possibile utilizzare questa tecnologia per le attività di telelavoro.

Va tenuto presente che le prestazioni nell'accesso ai servizi saranno limitate dalla banda massima disponibile sull'intero percorso d'interconnessione e che esse possono cambiare nel tempo in modo imprevedibile, in particolare se si transita da Internet.

Installare il software VPN Client in modalità Office Mode impone, sul Personal computer, stringenti politiche di sicurezza, in quanto il Personal Computer, una volta connesso, può accedere a tutti i servizi e le risorse alla stessa maniera delle postazioni connesse in rete locale; quindi:

- la configurazione è consentita unicamente su Personal Computer portatili di proprietà regionale, configurati con certificazione al dominio obbligatoria;
- il PC deve essere configurato in modo adeguato: il firewall locale deve essere attivo, l'antivirus attivo ed aggiornato, il sistema operativo aggiornato;
- quando la VPN è attiva, sono disponibili tutti i servizi della rete Intranet regionale; la navigazione Internet via browser è possibile unicamente attraverso il proxy; solo in questo caso è protetta e controllata;
- quando il Personal Computer non è connesso in VPN, la responsabilità di ogni accesso illegale ricade sull'utente; perciò l'amministrazione regionale non si assume alcuna responsabilità per danni eventualmente arrecati a terzi in questa modalità.

Il collegamento VPN Client ha un costo per l'Ente ed il numero di licenze disponibili è limitato, pertanto viene utilizzato solo nei casi effettivamente necessari, in cui non è possibile ricorrere ad altri tipi di collegamento. In modo analogo alle richieste di configurazione di collegamenti di accesso remoto, i collegamenti VPN client debbono

essere richiesti dai Responsabili di struttura e i tecnici del SIIR concorderanno con i referenti informatici della struttura stessa, il numero delle connessioni possibili e le modalità della loro attivazione, fino all'esaurimento delle risorse disponibili.

12.3 Accesso remoto via linea telefonica commutata

Per le zone non servite da ADSL, per gli utenti che, anche non essendo telelavoratori, hanno necessità di collegarsi dalla propria abitazione alla rete degli uffici regionali e per i piccoli uffici periferici in cui è presente una singola postazione di lavoro, è possibile collegarsi alla rete regionale attraverso collegamenti commutati basati su linee ISDN o linee analogiche tradizionali. L'accesso viene garantito da un apparato presente presso il CED regionale, dotato di un adeguato numero di linee telefoniche in ingresso.

Per motivi di sicurezza, l'accesso prevede la trasmissione delle proprie credenziali (la stessa coppia di userid e password normalmente usata per accedere alla LAN); una volta controllate le credenziali, e verificato che l'utente sia abilitato a questo tipo d'accesso, il sistema abbatte la connessione e richiama l'utente al numero telefonico a lui collegato all'interno del sistema (callback). Ciò garantisce quindi che un malintenzionato non possa accedere alla rete, anche se riesce ad acquisire in modo illegale credenziali valide.

La configurazione di callback utilizzata non ha lo scopo di far risparmiare gli utenti, che già hanno un vantaggio a poter lavorare da casa invece di recarsi sul proprio luogo di lavoro; pertanto, nessun rimborso potrà essere richiesto alla Regione per questo, anche in caso di malfunzionamento del sistema di richiamata.

In modo analogo alle richieste di configurazione di collegamenti via VPN Client, i collegamenti di accesso remoto vanno richiesti dai Responsabili di struttura e i tecnici del SIIR concorderanno con i referenti informatici della struttura stessa, il numero delle connessioni realizzabili e le modalità della loro attivazione.

Questo tipo di collegamento fornisce servizi "a banda stretta" e progressivamente verrà sostituito da soluzioni "a banda larga" con migliori prestazioni; ha l'innegabile vantaggio di essere utilizzabile anche nelle zone del territorio regionale non servite da circuiti di elevata qualità, e di essere attivabile in tempi molto brevi.

12.4 Accesso alla rete per fornitori di servizi di teleassistenza

Ove siano presenti sistemi connessi nella rete Intranet regionale, su cui siano installate soluzioni complesse, può presentarsi la necessità di permettere a ditte esterne di accedere ai sistemi stessi, per fornire servizi di assistenza e manutenzione da remoto.

Ricordiamo che il contratto stipulato col fornitore dovrà in ogni caso contemplare la designazione dei tecnici incaricati di questa attività quali responsabili esterni per il trattamento di dati personali, ed i conseguenti adempimenti previsti. Sarà inoltre opportuno prevedere nel contratto particolari clausole, quali richieste di penali, nel caso di violazioni delle norme di sicurezza attraverso gli accessi remoti configurati.

Sono previste due principali modalità di collegamento:

1) accesso via Internet: se il sistema è dotato di un indirizzo IP pubblico, è posizionato in un'area di rete accessibile da Internet, ed è configurato con le opportune policies di sicurezza, è possibile consentirne l'accesso attraverso una regola da predisporre sul

firewall regionale, che identificherà in modo preciso:

- la porta o le porte attraverso cui avverrà l'accesso; è necessario usare protocolli criptati ogni qualvolta vengano trattati dati personali; le porte consigliate sono ssh, https o similari;
- l'indirizzo / gli indirizzi di provenienza; si potrà trattare di un singolo IP pubblico assegnato al firewall/proxy della ditta fornitrice, oppure di una sottorete IP pubblica assegnata ufficialmente alla ditta stessa oppure ad una sua consociata; faranno fede gli appositi database delle autorità di rilascio degli indirizzi IP, come il RIPE per il nostro territorio;
- la regola sarà temporizzata, consentendo l'accesso solo in giorni ed orari preventivamente concordati; a meno di particolari esigenze, si sconsiglia di permettere l'accesso nelle fasce notturne o nei giorni festivi.

2) accesso via router su linea ISDN o analogica: viene realizzato un collegamento di rete concordando con la ditta fornitrice:
   - gli indirizzi IP da configurare, compatibili con quelli utilizzati nella rete intranet regionale;
   - le stringhe di sicurezza da scambiarsi nelle fasi d'interconnessione;
   - i numeri ISDN (chiamato e chiamante) da utilizzare, che verranno verificati ad ogni tentativo di connessione.

Sul router regionale in cui verrà attestato il collegamento, apposite regole limiteranno unicamente al sistema interessato le possibilità di accesso da parte della ditta fornitrice.

Ricordiamo che, qualora il contratto con la ditta fornitrice non fosse più valido per scadenza dei termini o inosservanza delle condizioni stabilite, è indispensabile segnalarlo tempestivamente al SIIR, in modo da impedire da quel momento in poi l'accesso da parte del fornitore al sistema interessato.

Nel corso del 2009 verrà attivata anche la modalità VPN SSL per accedere alla intranet regionale. La VPN SSL consentirà di profilare più accuratamente l'accesso alle risorse regionali garantendo un livello di sicurezza comparabile a quello fornito dal VPN Client Office Mode.

12.5 Accesso alle caselle di posta regionale tramite dispositivi mobili

Esiste inoltre la possibilità di accedere alla casella di posta regionale attraverso dispositivi mobili PDA o smartphone; questo accesso offre inoltre la possibilità di sincronizzare anche altri dati come rubrica, agenda e note tra il PDA e Outlook, purché quest'ultimo sia configurato in modo da consegnare la posta sulla casella postale del server e non in locale.

Le politiche implementate, ritenute opportune per rendere efficiente e sicuro il servizio offerto, sono di seguito elencate:

- abilitazione sul sistema di posta regionale del servizio per il singolo utente, contestuale alla richiesta di un dispositivo regionale o, in fase successiva, solo se il dispositivo in possesso dell'utente lo permette;

- sistema operativo Microsoft Windows Mobile 5 e versioni successive, e su sistema operativo Symbian con software Mail for Exchange che sono attualmente pienamente integrati con la piattaforma di posta esistente, nella quale risiedono le caselle regionali;
- nel caso di perdita di uno SmartPhone/PDA è indispensabile che l'utente ne dia tempestivamente comunicazione all'Ente, per autorizzare il blocco del servizio fornendo poi in tempi brevi anche la denuncia all'autorità competente;
- l'assistenza e la configurazione della sincronizzazione del PDA/Smartphone con il server di posta regionale è garantita solo per gli apparati di proprietà dell'ente.

Oltre ai servizi di posta elettronica, per quanto riguarda l'accesso ad altri servizi di rete da telefoni cellulari e palmari è possibilie accedere solo ai servizi che la regione rende disponibili sulla rete Internet.

13. Contesto normativo di riferimento

D. Lgs. 82/2005 "Codice dell'amministrazione digitale"

Art. 53 (Caratteristiche dei siti): *"Le pubbliche amministrazioni centrali realizzano siti istituzionali su reti telematiche che rispettano i principi di accessibilità, nonché di elevata usabilità e reperibilità, anche da parte delle persone disabili, completezza di informazione, chiarezza di linguaggio, affidabilità, semplicità dì consultazione, qualità, omogeneità ed interoperabilità"*

Legge 4/2004 "Disposizioni per favorire l'accesso ai soggetti disabili agli strumenti informatici"

Art. 4 (Obblighi per l'accessibilità) comma 1: *"Nelle procedure svolte dai soggetti … per l'acquisto di beni e per la fornitura di servizi informatici, i requisiti di accessibilità … costituiscono motivo di preferenza a parità di ogni altra condizione nella valutazione dell'offerta tecnica, tenuto conto della destinazione del bene o del Servizio. La mancata considerazione dei requisiti di accessibilità o l'eventuale acquisizione di beni o fornitura di servizi non accessibili è adeguatamente motivata"*

Art. 4 (Obblighi per l'accessibilità) comma 2: *"I soggetti …non possono stipulare, a pena di nullità, contratti per la realizzazione e la modifica di siti Internet quando non è previsto che essi rispettino i requisiti di accessibilità…. I contratti in essere, in caso di rinnovo, modifica o novazione, sono adeguati, a pena di nullità…"*

Art. 9 (Responsabilità): *"L'inosservanza delle disposizioni della presente legge comporta responsabilità dirigenziale e responsabilità disciplinare…. Ferme restando le eventuali responsabilità penali e civili previste dalle norme vigenti"*

DM 8 luglio 2005 "Requisiti tecnici e i diversi livelli per l'accessibilità agli strumenti informatici"

Art. 2 (Requisiti tecnici e livelli di accessibilità) comma 3: *"I requisiti tecnici si applicano anche nei casi in cui i soggetti … forniscono informazioni o erogano servizi mediante applicazioni Internet rese disponibili su reti Intranet o su supporti, come CD-ROM, DVD, utilizzabili anche in caso di personal computer non collegato alla rete"*

D. Lgs. 196/2003 "Codice in materia di protezione dei dati personali"

Gli obblighi in materia di sicurezza dei dati e dei sistemi sono contenuti principalmente nel Titolo V del Codice e in particolare nell' **Art. 31 (Obblighi di sicurezza)**, **Art. 33 (Misure minime)**, **Art. 34 (Trattamenti con strumenti elettronici)**.

Legge finanziaria 2007: I commi 892 e 895 della Finanziaria prevedono 30 m€ per il sostegno agli investimenti per l'innovazione negli EELL con priorità a chi utilizza o sviluppa applicazioni software a codice aperto.

Legge Regionale 11/2004: "Sviluppo regionale della società dell'informazione""

Art. 5 (Pluralismo informatico) comma 1:

Al fine di garantire ai cittadini la massima libertà di accesso all'informazione pubblica, la Regione promuove attivamente l'uso di formati di documentazione

elettronica e di basi dati su formati non proprietari. La Regione promuove la competitività e la trasparenza del mercato, assumendo quali linea-guida il principio del pluralismo informatico e di libera scelta nella realizzazione di piattaforme informatiche; promuove il riuso di software di cui le pubbliche amministrazioni sono proprietarie ed è impegnata alla riduzione di barriere dovute a diversità di formati non standard nella realizzazione di programmi e delle piattaforme e all'impiego ottimale sia del software a sorgente aperto che di quello a sorgente chiuso nella pubblica amministrazione.

Determinazione del Direttore Generale all'Organizzazione, Personale , Sistemi Informativi e Telematica n. 2651/2007: "Disciplinare Tecnico in materia di sicurezza delle applicazioni informatiche nella Giunta della Regione Emilia-Romagna"

Determinazione del Direttore Generale all'Organizzazione, Personale , Sistemi Informativi e Telematica n. 2653/2007: "Disciplinare Tecnico per utenti sull'utilizzo dei sistemi informativi nella Giunta della Regione Emilia-Romagna"

Linee guida per realizzare siti e applicazioni web accessibili per la Regione Emilia-Romagna: http://www.regione.emilia-romagna.it/lineeguida/

Allegato 1: Stack tecnologico delle filiere applicative supportate.

	Piattaforma **Microsoft** (**Windows 2003**)	Piattaforma **Linux** **(Linux RedHat AS 5)**
FILIERA A **Applicazioni su tecnologia JAVA (specifiche J2EE)**	WS: **Microsoft IIS** AS: **IBM WebSphere** DB: **Oracle 10g**	WS: **Apache** AS: **JBoss** DB: **PostgreSQL** **Oracle 10g**
FILIERA B **Applicazioni su tecnologia Microsoft**	WS: **Microsoft IIS** AS: **Microsoft .NET** DB: **SQL Server 2005**	-
FILIERA C **Applicazioni su tecnologia OpenSource (framework LAMP)**	-	WS: **Apache** AS: **PHP, Python, Perl Tomcat** DB: **MySQL, PostgreSQL**
Legenda**: WS:** Web Server – **AS:** Application Server – **DB:** Database Server		

Nel caso sia necessario fornire il dettaglio (ad esempio nei capitolati o nei contratti) delle versioni supportate dell'application server, web server e db server è possibile richiedere le informazioni al Servizio SIIR.

Allegato 2: Tecnologie a supporto delle filiere applicative.

Tecnologia di Storage	Al fine di conseguire una maggiore disponibilità dei dati, una metodologia di accesso standardizzata, una maggiore sicurezza ed una centralizzazione della gestione e del controllo, è stata introdotta una infrastruttura di Storage Area Network (SAN) per i dati delle filiere applicative. Si passa quindi da una logica di storage locale ad ogni singolo server ad una logica di centralizzazione. La gestione centralizzata che ne deriva comporta tutta una serie di vantaggi operativi rispetto allo storage locale, in particolare: • **Scalabilità**: dell'ordine delle decine di TeraByte • **Performance**: connessioni veloci grazie alla tecnologia Fibre Channel • **Flessibilità**: espansione dinamica dello storage, nessun downtime per l'aggiunta di dischi • **Backup:** ridotto tempo di ripristino dei dati in caso di failure • **Fault Tolerance**: funzionalità avanzate di mirroring dei dischi • **Availability**: affidabilità e continuità di Servizio basate su ridondanza di bus, alimentazioni, controller, .. • **Security**: maggiore sicurezza nella protezione dei dati a livello fisico ed applicativo
Tecnologia di Virtualizzazione	Per i 3 livelli fisici di una infrastruttura applicativa di test / sviluppo 3-tier si è optato per la tecnologia di virtualizzazione Vmware su sistemi dipartimentali High Level. Una soluzione su tale piattaforma tecnologica comporta numerosi vantaggi, in particolare: • **Indipendenza dall'hardware:** il layer VMWARE maschera il tipo di hardware fisico presente sul server, per cui il sistema operativo ospite vede e usa quello standard virtualizzato • **Isolamento dell'ambiente applicativo:** gli ambienti sono completamente indipendenti (sistema operativo, registry, dati, ecc.) per cui un crash o errore applicativo su un server virtuale non compromette in alcun modo l'integrità degli altri • **Incapsulamento**: l'intero stato di una macchina virtuale (memoria, dischi, ecc.) risiede su due file che possono essere gestiti agevolmente ripristinando velocemente la configurazione voluta • **Management**: possibilità di management centralizzato di tutte le macchine virtuali. Riconfigurazione e riallocazione dinamica delle macchine virtuali su altre macchine fisiche senza interruzione di Servizio • **Clustering:** possibilità di configurare ambienti cluster tra macchine virtuali usando soluzioni di clustering industry-standard. Si possono creare cluster tra macchine virtuali dello stesso server fisico o di server fisici diversi o ancora tra server fisico e macchina virtuale • **Ottimizzazione:** tramite la tecnologia di virtualizzazione si ottiene un utilizzo ottimale delle risorse hardware assegnate ai progetti di sviluppo • **Provisioning:** la tecnologia di virtualizzazione permette di accelerare notevolmente l'assegnazione degli ambienti di test/sviluppo alle strutture di sviluppo
Tecnologie di Backup	Da diversi anni è stata implementata una soluzione di backup centralizzato per tutti i sistemi server dipartimentali presenti al CED. Si tratta di una soluzione integrata per la gestione dello storage distribuito che opera con funzioni di backup/restore su file, database ed applicazioni.
Tecnologie di Monitoring e Management	Per le piattaforme di produzione e di test / sviluppo dedicate alle filiere applicative sono in via di implementazione strumenti di monitoring e di management per una gestione ottimale dei livelli di servizio hardware / software.

Tecnologia Blade	Per i 3 livelli fisici di una infrastruttura applicativa di produzione 3-tier (in pratica web server, application server e database server) si è optato per la tecnologia Blade. I blade sono dei server particolarmente sottili costruiti per essere inseriti nei bay di chassis appositamente predisposti che si connettono ad un backplane comune. Tali server sono dotati di propri processori, RAM, controller di rete, dischi e sono dunque indipendenti, ma condividono i componenti di alimentazione e raffreddamento, i floppy drive e gli switch con gli altri blade. Una soluzione su tale piattaforma tecnologica comporta numerosi vantaggi che risultano abilitanti verso gli obiettivi di scalabilità e fault tolerance, in particolare: • **Elevati livelli di densità**: un numero elevato di blade server è allocabile in un singolo chassis (o enclosure) consentendo di raggiungere un grado di densità molto più elevato rispetto ai server tradizionali. I blade costituiscono l'elemento ideale per la realizzazione di infrastrutture che implementino architetture avanzate di tipo clustered ed in generale per il load balancing • **Deployment veloce**: la tecnologia blade consente di operare il deployment di un nuovo server con estrema rapidità: da un punto di vista fisico si inserisce un nuovo blade nel rack con necessità di cablaggio ridotte al minimo; da un punto di vista software l'ausilio di sofisticati strumenti di management consente di caricare l'immagine del sistema operativo e dell'applicazione in modo automatizzato. • **Facilità di manutenzione**: la manutenzione fisica dei server blade risulta facilitata dal fatto che in caso di failure un blade è estraibile dal rack in modo non dissimile da un disco hot swap. La tecnologia blade si accompagna a sistemi di manutenzione che consentono di gestire intere batterie di server da console amministrative centralizzate con possibilità di generare in automatico messaggistica relativa a malfunzionamenti hardware/software e in alcuni casi di operare “predictive failure management”. • **Scalabilità modulare**: la scalabilità di un sistema basato su di un'architettura blade si basa sul concetto di “scale out” in luogo del tradizionale “scale up”. Operare in “scale up” significa dover potenziare l'unico server di cui si dispone nell'ambito dell'espandibilità da esso consentita e, una volta raggiunti i limiti massimi, è necessario ricorrere ad un server di fascia più alta. Lo “scale out” è una modalità estremamente più flessibile in quanto ad un'accresciuta esigenza di risorse elaborative consente di rispondere semplicemente aggiungendo moduli standard (blade) uguali a quelli già impiegati e di costo contenuto, fino al raggiungimento dell'obiettivo di performance prefissato.

Allegato 3: Linee guida per lo sviluppo .NET sui sistemi della Regione Emilia-Romagna.

1 Introduzione

Questo documento ha lo scopo definire le linee guida per lo sviluppo di applicazioni .NET che dovranno essere ospitate sui server della Regione Emilia-Romagna.
Data l'inerente flessibilità/varietà delle soluzioni realizzabili tramite il software, ci si rende conto che quanto scritto in questo documento può non prevedere situazioni per cui i requisiti qui descritti risultino non soddisfacibili o rendano impossibile raggiungere gli obiettivi che il sistema si propone di raggiungere. In questi casi è comunque necessario comunicare al personale tecnico della Regione la situazione affinché si possa discutere insieme la soluzione migliore da adottare.

2 Architettura di riferimento

2.1 Ambiente INTRANET

Versione .NET Framework: 1.1 (v1.1.4322) + 2.0 (v2.0.50727) + 3.5 SP1
Front end: web farm composta da 3 web server con Microsoft Application Center *Back end*: cluster di Microsoft SQL Server 2005 "active-active"

2.2 Ambiente INTERNET

Versione .NET Framework: 1.1 (v1.1.4322) + 2.0 (v2.0.50727) + 3.5 SP1
Front end: web farm composta da 3 web server con Microsoft Application Center *Back end*: cluster di Microsoft SQL Server 2005 "active-active"

3 Scrittura del codice

3.1 Linguaggio di riferimento:

C#

3.1.1 Convenzioni

Per quanto riguarda la scrittura del codice è tassativo seguire le linee guida del documento (su MSDN): "Design Guidelines for Class Library Developers"

3.1.2 Commenti

Il linguaggio C# .NET Framework mette a disposizione una sintassi apposita che consente, attraverso l'utilizzo di tools particolari di generare documentazione per le classi create all'interno del progetto. Per una descrizione dettagliata della sintassi da utilizzare, fare riferimento al manuale in linea sotto la voce: "Tags for Documentation Comments". Si raccomanda l'utilizzo di questa possibilità per la documentazione del codice.

4 Il namespace RER.Tools

RER.Tools è il namespace che contiene una serie di componenti da utilizzare affinché le applicazioni siano conformi ai requisiti oggetto di questo documento. Esiste un documento a parte, in forma di guida di riferimento, che li descrive in dettaglio. In questo documento, i componenti sotto RER.Tools, sono presentati in maniera sparsa seguendo il filo logico del discorso.

Per i dettagli sistemistici relativi a come eseguire l'installazione degli assemblies che compongono RER.Tools si veda il capitolo apposito.

5 Sicurezza

Deve essere prevista la gestione delle autorizzazioni ad un livello precedente a quello del DB: quindi nella pagine ASP.NET o nei componenti della business logic.

5.1 Applicazioni web ASP.NET

Le applicazioni web devono essere le prime a farsi carico della gestione delle autorizzazioni. Naturalmente ciò cambia a seconda del tipo di autenticazione implementato:

5.1.1 Scenario con "Windows Authentication" e gruppi Windows

In questo caso prevedere la creazione di gruppo Windows per rappresentare i ruoli dell'applicazione (tali gruppi devono avere "AG_" – Application Group – come prefisso).

Segue esempio: per un'applicazione che prevede i 6 ruoli:
Utente standard della Giunta (A)
Utente standard del Consiglio (B)
Utente amministratore della Giunta (C)
Utente amministratore del Consiglio (D)
Utente standard (senza diritto di protocollazione) della Giunta (E)
Utente standard (senza diritto di protocollazione) della Consiglio (F)
Questi 6 ruoli sono rappresentabili creando i 4 gruppi qui sotto indicati:

	AG_ComunicatiGiunta	AG_ComunicatiConsiglio
--	A	B
AG_ComunicatiAmministratori	C	D
AG_ComunicatiNoProtocollazione	E	F

Ovvero, un utente che deve avere, per esempio, il ruolo D deve essere inserito nei gruppi *AG_ComunicatiConsiglio* e *AG_ComunicatiAmministratori*.

Quindi parametrizzare questi nomi negli appSettings e attivare l'impersonation.

```
<configuration>
  <appSettings>
    <add key="NomeRuoloGiunta" value="RERSDM\AG_ComunicatiGiunta" />
    <add key="NomeRuoloConsiglio" value="RERSDM\AG_ComunicatiConsiglio" />
    <add key="NomeRuoloAmministratori" value="RERSDM\AG_ComunicatiAmministratori" />
    <add key="NomeRuoloNonProtocollanti" value="RERSDM\AG_ComunicatiNoProtocollazione" /
> </appSettings>
<system.web>
<!-- ... -->
<authentication mode="Windows" />
```

```
<identity impersonate="true" />
<!-- ... -->
</system.web>
</configuration>
```

A questo punto nell'applicazione web o nella business logic è sufficiente testare all'appartenenza al gruppo tramite il membro statico *CurrentIdentityIsInRole* della classe ***RER.Tools.Security***:

```
RER.Tools.Security.CurrentIdentityIsInRole(ConfigurationSettings.AppSettings["NomeRuoloAmministratori"])
```

5.1.2 Scenario con "Windows Authentication" e senza gruppi Windows

NB: L'approccio sopra indicato è fattibile solo se i requisiti di autorizzazione dell'applicazione sono rappresentabili tramite creazione di gruppi windows e non sono in perenne cambiamento. Si tenga presente che la creazione dei gruppi Windows e la gestione dei loro membri sono a carico degli amministratori del dominio e non sotto il controllo dell'applicazione.
Pertanto, nel caso i requisiti di autorizzazione dell'applicazione non siano rappresentabili tramite creazione di gruppi windows e/o si deve potere cambiare continuamente chi fa cosa, occorre gestire li stessi applicativamente creando un componente apposito nella business logic basato sul nome dell'utente che si è autenticato:

```
System.Security.WindowsIdentity.GetCurrent().Name
```

5.1.3 Scenario senza "Windows Authentication"

In questi casi (solitamente "Forms Authentication") non potendo avvalersi dell'*impersonation* la soluzione non può che essere applicativa. Esiste però un servizio di autenticazione centralizzata.
In particolare sono presenti due domini "Active Directory", uno contenente gli account degli utenti regionali (dominio intranet) e l'altro contenente account di utenti esterni alla regione (dominio extranet), e un sistema centralizzato di autenticazione (utilizzabile per entrambi i domini).

Ci sono due possibilità per sfruttare i servizi di questo sistema centralizzato:
appoggiarsi per il login e la modifica della password all'applicazione web di autenticazione centralizzata disponibile all'url
https://wwwservizi.regione.emilia-romagna.it/AutenticazioneCentralizzata
(l'integrazione con l'applicazione *client* avviene tramite "browser redirection")
richiamare i servizi web che implementano i metodi necessari per l'autenticazione disponibili all'url
https://wwwservizi.regione.emilia-romagna.it/WebServices/AutenticazioneCentralizzata/Authentication.asmx

Se si intende avvalersi di questa possibilità si faccia riferimento alla documentazione relativa (Allegato 9).

5.2 Applicazioni non web

Nel caso di applicazioni non web (Console Applications e Window Forms Applications) l'impersonation è sempre attiva pertanto valgono le stesse regole del paragrafo 5.1.1.

6 Utilizzo del DB Server

6.1 Creazione di connessioni

Per ragioni di sicurezza, di gestione delle stringhe di connessione in fase di deployment, e di problemi di *delegation* tra il web server e il DB server; qualsiasi connessione al DB server (SQL Server 2005)

deve avvenire attraverso il componente ***RER.Tools.SqlConnectionBroker***.

L'utilizzo di tale componente è banale, basta passare il nome del DB:
SqlConnection cn = RER.Tools.SqlConnectionBroker.GetConnection("NomeDB");
oppure
SqlConnection cn = RER.Tools.SqlConnectionBroker.GetConnection("NomeDB", true);
per ottenere una connessione già aperta.

Al fine di rendere minimi gli interventi nel caso di modifiche sul DB server è opportuno centralizzare, il nome del database. Occorre creare una classe di nome *Global* in cui mettere informazioni globali, come è il nome del database:

```
public class Global
{
        public const string DBName = "NomeDB";
}
```

pertanto gli esempi sopra indicati diventano:
SqlConnection cn = RER.Tools.SqlConnectionBroker.GetConnection(Global.DBName);
SqlConnection cn = RER.Tools.SqlConnectionBroker.GetConnection(Global.DBName, true);

6.1.1 Configurazione di SqlConnectionBroker

La configurazione del SqlConnectionBroker (da inserire nel .config file dell'applicazione) avviene applicazione per applicazione e deve essere concordata con il personale tecnico della Regione. Comunque il componente decide con quali credenziali creare la connessione al DB in base al gruppo di appartenenza dell'utente che sta facendo la richiesta.
La configurazione più semplice (che permette a tutti l'accesso, i.e. *allowAnonymous="true"*) ha i seguente aspetto:

```
<RER>
  <SqlConnectionBrokerSettings version="1.2">
        <DB
          name="Comunicati"
          catalog="Comunicati"
          applicationName="Gestione comunicati stampa"
              >
              <Credential
                    allowAnonymous="true"
                    userName="..."
                    password="..."
                    isCrypted="false" />
        </DB>
</RER>
```

Il componente permette configurazione più sofisticate e anche che un'applicazione acceda a più di un database (sempre SQLServer), ma come già detto, la creazione di questa configurazione va concordata con il personale tecnico della Regione.

6.2 Chiusura delle connessioni e "data readers"

E' tassativo chiudere sempre le connessioni. Per maggiore sicurezza si richiede anche l'utilizzo del costrutto *using* tipico di C#. Esso garantisce la chiamata all'interfaccia IDisposabile, e ciò assicura che le risorse siano liberate il prima possibile e non solo durante la Garbage Collection.
Esempio:

```
public static bool IsProtocollato(int idComunicato)
{
```

```
  using (SqlConnection sqlConnection = SqlConnectionBroker.GetConnection(Global.DBName,
true))
  {
            SqlCommand cmd = new SqlCommand("Comunicato_IsProtocollato", sqlConnection);
            cmd.CommandType = CommandType.StoredProcedure;

            cmd.Parameters.Add("@id_comunicato", SqlDbType.Int).Value = idComunicato;
            cmd.Parameters.Add("@protocollato", SqlDbType.Bit);

            cmd.Parameters["@protocollato"].Direction = ParameterDirection.InputOutput;

            cmd.ExecuteNonQuery();
            cmd.Connection.Close();

            return (bool) cmd.Parameters["@protocollato"].Value;
  }
}
```

Nel caso si debba restituire un SqlDataReader o *IDataReader*, la connessione deve essere chiusa dal "client". In questi casi assicurarsi di aggiungere l'opzione *CommandBehavior.CloseConnection* nel metodo di esecuzione del SqlCommand.
NB: non restituire esattamente un oggetto di tipo *SqlDataReader* bensì l'interfaccia *IDataReader* (da esso implementata). Ciò al fine di minimizzare gli interventi in caso di cambiamento del database server.

```
public static IDataReader ElencaUltimi(
  string idStruttura,
  int nrOre,
  bool soloProtocollati)
{
  using (
      SqlCommand cmd = new SqlCommand(
            "Comunicato_ElencaUltimi",
            SqlConnectionBroker.GetConnection(Global.DBName, true)
   )
      )
  {
            cmd.CommandType = CommandType.StoredProcedure;

            cmd.Parameters.Add("@id_struttura", SqlDbType.VarChar, 10).Value = idStruttura;
            cmd.Parameters.Add("@nr_ore", SqlDbType.Int).Value = nrOre;
            cmd.Parameters.Add("@solo_protocollati", SqlDbType.Bit).Value = soloProtocollati;

return
   cmd.ExecuteReader(CommandBehavior.CloseConnection|CommandBehavior.SingleResult);
  }
}
```

Per quanto detto sopra, l'interfaccia IDataReader deve essere usata anche dal "client", rispetto all'esempio precedente, il "client" dovrà eseguire qualcosa del genere:

```
IDataReader reader = ComunicatoManager.ElencaUltimi(idStruttura, nrOre, false);
while (reader.Read())
{
// ...
}
reader.Close();
```

oppure

```
myGrid.DataSource = ComunicatoManager.ElencaUltimi(idStruttura, nrOre, false);
((IDataReader) myGrid.DataSource).Close();
```

6.3 SQL Injection e creazione di statement SQL dinamici

La regola generale è di utilizzare sempre e solo delle stored procedure, ovverosia non creare mai delle stringhe SQL a livello di logica applicativa. A volte questo non è però possibile. La situazione tipica sono i moduli di ricerca libera e/o avanzata, quelli in cui l'utente può mettere diversi tipi di parametri di ricerca. In quei casi creare una SP i cui parametri riescano a soddisfare tutte le possibili richieste è molto difficile se non impossibile.
In tal caso occorrerà comporre lo statement SQL dinamicamente (all'interno della logica applicativa), però tramite ADO.NET ciò può essere fatto in maniera più elegante e soprattutto in maniera sicura rispetto ai problemi di Sql Server Injection, in quanto è possibile definire nei SqlCommand di tipo CommandText dei parametri formali (non i semplici "?" che dava a disposizione ADO).
Un esempio chiarisce subito: anziché scrivere una cosa del genere:

```
cmd.CommandText += " AND campo LIKE '%" + valoreDaCercare.Replace("'", "") + "'";
```

si può scrivere:

```
cmd.CommandText += " AND campo LIKE @valoreDaCercare";
cmd.Parameters.Add("@valoreDaCercare", SqlDbType.VarChar, 255).Value =
        "%" + valoreDaCercare + "%";
```

Nonostante ci sia più codice da scrivere ci sono i seguenti vantaggi:
non occorre raddoppiare l'apice singolo (è quindi immediatamente "Sql Injection Safe")
si associa un tipo al parametro (in questo caso VarChar) e ciò rende più robusto il sistema (sempre più "Sql Injection Safe")
lo stesso parametro può essere usato più volte nel commandText senza per questo doverlo aggiungere più volte alla collection dei parameters (cosa che succedeva con ADO)
l'esecuzione è più veloce in quanto Sql Server genera una SP temporanea che riesce anche a riutilizzare
il codice è più leggibile.

NB: Nel caso ci si trovasse comunque nella condizione di dovere esplicitamente raddoppiare l'apice per mettere un valore in una stringa rappresentante dell'SQL utilizzare, anziché il metodo *string.Replace*, la funzione *RER.Tools.Sql.MakeSafe*. Al momento, essa, non fa altro che raddoppiare l'apice singolo, ma in futuro, per garantirsi contro nuovi tipi di attacchi di tipo Sql Injection, l'avere utilizzato questa funzione, permette di intervenire velocemente su tutte le applicazioni.

7 Gestione delle eccezioni

Come indicato dalla Microsoft, tutte le eccezioni specifiche dell'applicazione *devono* essere delle ApplicationException o derivare da essa.
La gestione delle eccezioni deve avvenire utilizzando i servizi dei componenti definti in ***RER.Tools.ApplicationLogger***.

In questo namespace sono definite le classi da usare per ottenere un sistema centralizzato per la raccolta degli errori delle applicazioni in produzione (durante lo sviluppo non lo si attiva, ma deve essere comunque previsto). Gli errori vengono registrati su un DB e il responsabile dello sviluppo viene avvisato via email.

Tutto questo serve per monitorare il funzionamento dell'ambiente in produzione (per esempio per rendersi conto nel caso ci sia un tentativo d'attacco di un hacker) e per nascondere all'utente finale i dettagli degli errori non previsti (gli si dà solo una comunicazione generica).

7.1.1 Utilizzo in una Web Application

Per utilizzare il componente in una web application è sufficiente gestire l'evento globale (definito nel file Global.asax) Application_Error:

```
  protected void Application_Error(Object sender, EventArgs e)
  {
#if(!DEBUG)
            Exception exception = Server.GetLastError();
            // Server.GetLastError() ritorna sempre una "HttpUnhandledException",
            // per accedere all'eccezione vera e propria occorre recuperare l'InnerException
            if (exception != null && exception.InnerException != null)
                        exception = exception.InnerException;
            try
            {
                        RER.Tools.ApplicationLogger.WebApplicationLogger.LogEvent(
"Interfaccia per la gestione dei comunicati stampa",
this,
exception
);
                        Server.ClearError();
                        if (exception is ApplicationException)
                                    Response.Redirect(
string.Format("ErroreApplicazione.aspx?msg={0}", Server.UrlEncode(exception.Message)));
                        else
                                    Response.Redirect("ErroreSistema.htm");
            }
            catch
            {
                        // Se si è riusciti a loggare non resta che mostrare l'errore anche all'utente
                        throw exception;
            }
#endif
  }
```

Come si vede il codice viene compilato solo se non si è in debug.
Si estrae da Server.GetLastError l'InnerException perchè questa funziona "wrappa" l'eccezione vera e propria in una HttpUnhandledExcpetion.
Si passa l'eccezione al logger assieme al riferimento alla HttpApplication (this).
Si controlla quindi se l'eccezione è una ApplicationException, in questo caso se ne mostra la descrizione all'utente (tramite un redirect alla pagina ErroreApplicazione.aspx che deve essere prevista). Quindi in pratica occorre generare delle ApplicationException al fine di fornire all'utente una segnalazione di errori coerenti. Per tutte le altre eccezioni (quelle generate dal CLR) si mostra una pagina d'errore generica (ErroreSistema.htm, che deve essere prevista) in cui si avvisa che c'è stato un problema e che gli amministratori sono stati avvisati.
NB: E' necessario che la pagina di segnalazione di un eccezione non di tipo ApplicationException sia una pagina statica HTML. Se fosse una pagina ASPX, lo stesso inconveniente che ha generato l'eccezione potrebbe ripresentarsi all'esecuzione della pagina di segnalazione dello stesso.

7.1.2 Utilizzo in una Console application

In una console application (tipicamente si tratterà di job non interattivi) l'approccio è simile a quello delle web application, ma esiste un'altra classe. Al fine di intrappolare tutte le eccezioni non gestite usare l'evento AppDomain.CurrentDomain.UnhandledException. Segue l'esempio:

```
  [STAThread]
  static int Main(string[] args)
  {

#if(!DEBUG)
            AppDomain.CurrentDomain.UnhandledException +=
new UnhandledExceptionEventHandler(GestoreEccezioni);
#endif
  // Per il formato di output delle date (utile se questo prg è schedulato)
            RER.Tools.Globalizer.SetThreadSpecificCulture("it-IT");

            Console.WriteLine("---- INIZIO ELABORAZIONE ({0}) ----", DateTime.Now);

            // ...

            Console.WriteLine("---- FINE ELABORAZIONE ({0}) ----", DateTime.Now);
            Console.WriteLine();
            return 0;

  }

  static void GestoreEccezioni(object sender, UnhandledExceptionEventArgs args)
  {
            Exception e = (Exception) args.ExceptionObject;
            Console.WriteLine(e.Message);
            Console.WriteLine("---- FINE ELABORAZIONE CON ERRORI ({0}) ----", DateTime.Now);
            Console.WriteLine();
            RER.Tools.ApplicationLogger.ConsoleApplicationLogger.LogEvent(e);
            Environment.Exit(1);
  }
```

Al ConsoleApplicationLogger non occorre passare niente altro che l'eccezione in quanto tutte le informazioni che servono sono accessibili tramite la classe *Environment*.

7.1.3 Utilizzo in una Windows Form application

Per ora non è previsto nessun componente specifico per gli errori delle Windows Form application, utlizzzare il componente ConsoleApplicationLogger.

8 Invio di email

Visto che in Regione Emilia-Romagna non è installata la libreria CDO, prima di inviare un'email è necessario impostare l'SMTP server: è sufficiente impostare la proprietà statica SmtpServer della classe SmtpMail col nome del nuovo server. Il nome dell'SMTP Server è stato comunque centralizzato, in pratica è sufficiente la seguente istruzione prima di inviare l'email:

```
SmtpMail.SmtpServer = RER.Tools.Configuration.SMTPServer;
```

Mail Multipart
La creazione di mail multipart (testo e HTML) avviene in modo automatico se è impostata ad HTML (tipo enum per MailFormat) la proprietà BodyFormat del MailMessage. In particolare viene generato sia il body in formato HTML, sia quello in plain-text rimuovendo i tag ipertestuali.

8.1 RER.Tools.Mail

Questo namespace contiene le classi per la gestione dell'invio di messaggi di posta elettronica. E' particolarmente utile quando si invia un'email a molti indirizzi, infatti il metodo Send della classe SmtpMail (in RER.Tools.Mail) spezza l'invio di un messaggio in tanti invii ognuno dei quali contiene al massimo un certo numero di indirizzi. Questo valore è di default RER.Tools.Configuration.MaxRecipientsPerMessage oppure lo si può passare esplicitamente. A seguito dell'invio è poi possibile ottenere in dettaglio (destinatario per destinatario) cosa è successo. Il tutto è basato sulla classe Recipient.

8.1.1 Recipient

Questa classe è utilizzata sia per indicare le informazioni di un destinatario (proprietà Email e Name) che per ottenere l'esito dell'invio (proprietà Status e Exception).
La proprietà Status è un'enumeration definita nel seguente modo:

```
public enum RecipientStatus
{
        Uninitialized,              // Stato iniziale
        InitializedAndChecked,  // Inizializzato e indirizzo sintatticamente corretto
        Queued,                             // Messaggio trasmesso all'SMTP server
        Error                       // Problemi, vedere la proprietà 'Exception'
}
```

Quando occorre inviare un'email creare uno o più oggetti Recipient (nel caso si tratti di più indirizzo, raggrupparli in un ArrayList). Invocare quindi uno dei metodi della classe SmtpMail definita più sotto), essa tenta l'invio e imposta la proprietà Status (ed eventualmente la proprietà Exception).
Pertanto è possibile sapere cos'è successo all'invio.

Esempio:

```
ArrayList recipients = new ArrayList();
recipients.Add(new Recipient("tizio@dominio.com"));
recipients.Add(new Recipient("caio@dominio.com "));
System.Web.Mail.MailMessage message = new System.Web.Mail.MailMessage();
message.Body = "Prova (Corpo)";
message.Subject = "Prova";
RER.Tools.Mail.SmtpMail.Send(message, recipients);
foreach (Recipient r in recipients)
if (r.Status != RecipientStatus.Queued)
                Console.WriteLine("{0}: {1}", r.Email, r.Exception.Message);
```

Nel caso un Recipient debba essere usato più volte chiamare il metodo ResetStatus per riazzerare il suo stato.

8.1.2 SmtpMail

Questa classe è un wrapper di System.Web.Mail.SmtpMail, e ha i seguenti 4 metodi statici.

static Recipient Send(MailMessage message, string recipient)
Questo è il metodo più semplice per inviare il messaggio a un solo destinatario. Si crea il messaggio e si chiama questo metodo indicando il destinatario come stringa. Il metodo restituisce l'oggetto Recipient corrispondente al destinatario specificato. Con esso si può controllare l'esito dell'invio:

```
MailMessage msg = new MailMessage();
// composizione 'msg'...
if (RER.Tools.Mail.SmtpMail.Send(msg, "tizio@caio.it").Status != RecipientStatus.Queued)
        Console.WriteLine("Errore...");
```

Naturalmente in questo modo non si riesce ad ottenere la descrizione dell'errore. Per farlo occorre memorizzare il risultato del metodo in una variabile di tipo Recipient:

```
MailMessage msg = new MailMessage();
// composizione 'msg'...
Recipient r = RER.Tools.Mail.SmtpMail.Send(msg, "tizio@caio.it");
if (r.Status != RecipientStatus.Queued)
        Console.WriteLine(r.Exception.Message);
```

Oppure si usa direttamente il metodo seguente

static bool Send(MailMessage message, Recipient recipient)
Questo è come il precedente ma si aspetta, nella specificazione del destinatario, un oggetto Recipient anziché una stringa:

```
MailMessage msg = new MailMessage();
// composizione 'msg'...
Recipient r = new Recipient("tizio@caio.it");
RER.Tools.Mail.SmtpMail.Send(msg, r);
if (r.Status != RecipientStatus.Queued)
        Console.WriteLine(r.Exception.Message);
```

static bool Send(MailMessage message, ArrayList bccRecipients, int maxBccRecipientsPerMail)
static bool Send(MailMessage message, ArrayList bccRecipients)
Usare questo metodo quando il messaggio deve essere inviato a più indirizzi.

- *message*: Messaggio da inviare (non impostare i destinatari BCC, saranno ignorati, usare invece il parametro bccRecipients
- *bccRecipients*: Passare un Arraylist di oggetti di tipo RER.Tools.Mail.Recipient o di tipo stringa. Il metodo ha, in ogni caso, come "side-effect" la trasformazione degli oggetti string presenti bccRrecipients in oggetti di tipo Recipient. Ciò permette, al chiamante, di poter ottenere il risultato dell'invio (interrogando le proprietà 'Recipient.Status' e 'Recipient.Exception') anche avendo passato delle stringhe
- *maxBccRecipientsPerMail*: Numero massimo di indirizzi per email (deve essere maggiore di 0). Se non specificato si usa il valore configurato a livello di server (RER.Tools.Configuration.MaxRecipientsPerMessage).

ritorna 'true' se tutto è andato bene, altrimenti occorre scorrere l'Arraylist interrogando la proprietà 'Status' degli oggetti. Se Status != RecipientStatus.Queued allora ci sono stati dei problemi, per i dettagli interrogare la proprietà 'Exception'.

8.1.3 Metodi d'utlilità

La classe SmtpMail contiene anche il metodo:

static bool Split(MailMessage message, ArrayList bccRecipients)
Questa funzione cerca di spezzare una linea di testo in "tranci" di non più di 1000 caratteri. Il punto in cui "tranciare" è identificato da uno spazio. Serve perché l'SMTP vuole che il corpo del messaggio sia composto da linee di non più di 1000 caratteri. Per sicurezza questa funzione cerca un punto di possibile "tranciatura" dall 800 carattere a ritroso.
Nel caso non ne trovi nessuno genera una ArgumentException.
NB: Eventuali spazi precedenti o successivi al punto di "tranciamento" sono eliminati.

9 RER.Tools.DirectoryServices

E' un namespace contente 5 classi di utilità per accedere ai dati sul Active Direcotory

9.1 AdsiHelper

static DirectoryEntry GetDirectoryEntry(string path)
Restituisce la directory entry corrispondente al path indicato. Utile in quanto non richiede le credenziali di autenticazione (che sono definite centralmente).

9.2 AdsiUserHelper

Queste classe permette di avere informazioni sugli utenti del dominio RERSDM (utenti regionali) ed EXTRARER(utenti non regionali ma abilitati).

Per instanziare un oggetto di tipo AdsiUserHelper è possibile utilizzare i 2 costruttori:

- **public AdsiUserHelper(string valore, AdsiUserHelper.TipoIdentificatore tipoIdentificatore)** dove TipoIdentificatore è un enum che comprende i valori account e matricola;

- **public AdsiUserHelper(string domainName, string accountName)**.

Di seguito vengono elencate le proprietà disponibili:

- public string **DistinguishedName**: resituisce il nome che identifica univocamente l'entry nella directory

- public string **EmployeeID**: resituiscela matricola di un dipendente regionale. Restituisce null se il valore non è presente.

- public string **EmployeeType**: resituisce il tipo di dipendente regionale. Restituisce null se il valore non è presente.

- public string **EmployeeNumber**: resituisce un numero univoco del dipendente regionale (es. Codice Fiscale). Restituisce null se il valore non è presente.

- public string **RERTipoIncarico**: resituisce il codice del tipo di incarico presso la regione per dipendenti regionali. Restituisce null se il valore non è presente.

- public string **RERPosLav**: resituisce la posizione lavorativa del dipendente regionale. Restituisce null se il valore non è presente.

- public string **Name**: restituisce l'account dell'utente all'interno del dominio (es. Formica_F)

- public string **FirstName**: restituisce il nome dell'utente.

- public string **LastName**: restituisce il cognome dell'utente.

- public string **FullName**: restituisce il nome completo (nome + cognome).

- public string **Description**: restituisce la descrizione dell'utente (es. Consulente software).

- public string **ManagedBy**: restituisce l'account del responsabile. Restituisce null se il valore non è presente.
- public string **EmailLDAP**: restituisce l'email reginonale dell'utente. Restituisce null se il valore non è presente.
- public string **PianoStanza**: restituisce il piano e la stanza dell'ufficio dell'utente. Restituisce null se il valore non è presente.
- public string **Indirizzo**: restituisce l'indirizzo dell'ufficio dell'utente. Restituisce null se il valore non è presente.
- public string **Citta**: restituisce la città dell'ufficio dell'utente. Restituisce null se il valore non è presente.
- public string **TelephoneNumberLDAP**: restituisce il numero di telefono dell'ufficio dell'utente. Restituisce null se il valore non è presente.
- public string **Fax**: restituisce il numero di fax dell'ufficio dell'utente. Restituisce null se il valore non è presente.
- public string **Cellulare**: restituisce il numero di cellulare dell'utente. Restituisce null se il valore non è presente.
- public string **WorkHomeNumber**: restituisce il numero del telefono del telelavoro. Restituisce null se il valore non è presente.
- public StringCollection **AltriTelefoni**: restituisce una collection di numeri di telefono dell'utente. Restituisce null se il valore non è presente.
- public string **HomeDirectory**: restituisce il path della virtual directory personale dell'utente all'interno del dominio regionale.
- public string **HomeDrive**: restituisce il nome assegnato alla virtual directory personale.
- public StringCollection **Groups**: restituisce una collection di gruppi di dominio ai quail l'utente appartiene per gli utenti RERSDM. Restituisce null se il valore non è presente.
- public SortedList **ManagedGroups**: restituisce una collection di gruppi di dominio dei quail l'utente è il responsabile. Restituisce null se il valore non è presente.
- public SortedList **ManagedUsers**: restituisce una collection di utenti dei quail l'utente è il responsabile. Restituisce null se il valore non è presente.
- public SortedList **Computer**: restituisce una collection di computer dei quail l'utente è proprietario. Restituisce null se il valore non è presente.
- public SortedList **OrganizationalUnit**: restituisce l'unità organizzativa di appartenenza. Restituisce null se il valore non è presente.
- public SortedList **GroupsEXTRARER**: restituisce l'elenco dei gruppi di appartenenza per gli utenti del dominio EXTRARER. Restituisce null se il valore non è presente.

- public string **PathLDAP**: restituisce il percorso completo.
- public string **UrlApplication**: restituisce l'url della pagina personale dell'utente. Restituisce null se il valore non è presente.
- public string **UrlApplication2**

- public string **InfoApplication**
- public string **Comment**
- public DateTime **AccountExpirationDate**
- public bool **PasswordDontExpires**
- public DateTime **PasswordLastChanged**
- public TimeSpan **PasswordAge**
- public string **BadLoginCount**
- public string **BadPwdCount**
- public int **UserFlags**
- public bool **IsDisabled**
- public bool **IsAccountExpired**
- public bool **IsAccountLocked**
- public bool **IsPasswordExpired**
- public bool **IsDisabledToModify**
- public DateTime **LastLogin**
- public DateTime **LastLogoff**
- public DateTime **LastFailedLogin**
- public string **LoginScript**

Seguono esempi:

```
Console.Write("Dominio utente? ");
string dominio = Console.ReadLine();
Console.Write("Nome utente? ");
string username = Console.ReadLine();
AdsiUserHelper user = new AdsiUserHelper(dominio, username);
```

```
Console.WriteLine("Nome: {0} ", user.FirstName);
Console.WriteLine("Cognome: {0} ", user.LastName);
Console.WriteLine("Nome completo: {0} ", user.FullName);
Console.WriteLine("Matricola: {0} ", user.EmployeeID);
Console.WriteLine("Email: {0} ", user.EmailLDAP);
Console.WriteLine("Password Age: {0} ", user.PasswordAge);
Console.WriteLine("Gruppi:");
foreach(DictionaryEntry nomeGruppo in user.ManagedGroups)
        Console.WriteLine(" - " + nomeGruppo.Value);

Console.WriteLine("Users:");
foreach(DictionaryEntry nomeUser in user.ManagedUsers)
        Console.WriteLine(" - " + nomeUser.Value);

Console.WriteLine("Computer:");
foreach(DictionaryEntry nomeComputer in user.Computer)
        Console.WriteLine(" - " + nomeComputer.Value);

Console.WriteLine("OrganizationalUnit:");
foreach(DictionaryEntry nomeOrganizationalUnit in user.OrganizationalUnit)
        Console.WriteLine(" - " + nomeOrganizationalUnit.Value);

Console.WriteLine("GroupsEXTRARER:");
foreach(DictionaryEntry nomeGroupsEXTRARER in user.GroupsEXTRARER)
        Console.WriteLine(" - " + nomeGroupsEXTRARER.Value);
```

Quindi è possibile anche avere la scheda di un utente conoscendo solo la matricola (come dipendente regionale)

9.3 AdsiGroupHelper

Questa classe permette di avere informazioni sui gruppi del dominio RERSDM (regionali) ed EXTRARER(extra-regione).

Per instanziare un oggetto di tipo AdsiGroupHelper è possibile utilizzare i 3 costruttori:

- public **AdsiGroupHelper**()
- public **AdsiGroupHelper**(string group)
- public **AdsiGroupHelper**(string domainName, string group)

Di seguito vengono elencate le proprietà disponibili:

- public string **Description**: restituisce la descrizione del gruppo.
- public long **GroupType**: restituisce il tipo di gruppo.
- public string **ManagedBy**: restituisce l'utente referente (per RERSDM)-
- public string **ManagedByExtrarer**: restituisce l'utente referente (per EXTRARER).
- public string **WWWHomePage**: restituisce URL applicazione o UNC share associata.
- public string **Url**: restituisce URL applicazione o UNC share associata (alternativo).
- public SortedList **UsersOrdinati**: restituisce una lista di utenti ordinati per nome.

9.4 AdsiComputerHelper

Questa classe permette di avere informazioni sui computer del dominio RERSDM (regionali).

Per instanziare un oggetto di tipo AdsiComputerHelper è possibile utilizzare i 2 costruttori:

- public **AdsiComputerHelper**()
- public **AdsiComputerHelper**(string computer)

Di seguito vengono elencate le proprietà disponibili:

- public string **Inventario**
- public string **DNSHostName**
- public string **Description**
- public string **ManagedBy**
- public string **OperatingSystem**
- public string **OperatingSystemVersion**
- public string **OperatingSystemServicePack**
- public string **StreetAddress**
- public string **Citta**: (collocazione – solo Server)
- public string **Stanza**: (collocazione – solo Server)
- public string **TelephoneNumber**: Telefono più vicino (collocazione – solo Server)

9.5 AdsiOrganizationalUnitHelper

Questa classe permette di avere informazioni sulle unità funzionali del dominio RERSDM (regionali).

Per instanziare un oggetto di tipo AdsiOrganizationaUnitHelper è possibile utilizzare i 2 costruttori:

- public **AdsiOrganizationaUnitHelper**()
- public **AdsiOrganizationaUnitHelper** (string organizationalUnit)

Di seguito vengono elencate le proprietà disponibili:

- public string **Description**: restituisce la descrizione.
- public string **ManagedBy**: restituisce il referente.
- public string **CodiceHost**: restituisce il codice della struttura sull'host.
- public string **Indirizzo**
- public string **Citta**
- public string **CAP**
- public string **SiglaProvincia**
- public string **DisplayName**: restituisce il nome visualizzato.

10 Altro

In RER.Tools sono anche definite le seguenti classi i cui servizi possono risultare utili.

10.1 RER.Tools.Sql

Questa classe ha 2 metodi statici *CreateLikeClause* e *MakeSafe*:

CreateLikeClause

```
static string CreateLikeClause(
string userInput,
string dbFieldName,
LikeClauseOptions mode,
out int numberOfTerms
)
```

oppure

```
static string CreateLikeClause(
string userInput,
string dbFieldName,
LikeClauseOptions mode
)
```

Questo metodo restituisce la condizione "where" che serve per cercare una o più parole in una campo testuale di una tabella (dbFieldName). In pratica si prende l'input dell'utente (userInput) e lo si scorre per trovare tutte le "parole" che lo compongono (eventualmente raggruppando in una parola sola quelle comprese tra doppi apici ["]) e per ognuna si crea una condizione "LIKE". Tutte queste condizioni "LIKE" vengono raggruppate insieme in "AND" o in "OR" a seconda del parametro "mode" (AllTerms = AND, AnyTerm = OR).

userInput: l'input inserito dall'utente
dbFieldName: campo su cui fare la ricerca
mode: AllTerms=condizioni messe in AND, AnyTerm=condizioni messe in OR
numberOfTerms: restituisce quanti termini sono stati creati

valore restituito: la stringa contenente la condizione SQL che realizza la ricerca, accodarla allo statement che si sta preparando. E'una stringa sicura relativamente ai problemi di SQL Injection (si utlizza RER.Tools.Sql.MakeSafe, precedentemente citata e sotto descritta).

Esempio:

```
SqlCommand cmd = new SqlCommand();
cmd.CommandText = "SELECT … FROM … WHERE …";
string temp =
  SQL.CreateLikeClause("prova parole chiave", "colonna", SQL.LikeClauseOptions.AnyTerm);
if (temp != string.Empty)
cmd.CommandText += " AND " + temp;
```

In questo caso la condizione restituita sarebbe:

(colonna LIKE '%prova%' OR colonna LIKE '%parole%' OR colonna LIKE '%chiave%')

Se invece lo user input fosse stato [prova "parole chiave"] la condizione sarebbe stata:

(colonna LIKE '%prova%' OR colonna LIKE '%parole chiave%')

ovvero le parole racchiuse tra doppi apici sono considerate come una parola sola. Come al solito, per includere un doppio apice all'interno di un raggruppamento l'utente deve immetterlo raddoppiato. Per esempio: [prova "di un ""testo"" raggruppato"] crea la seguente condizione where:

(colonna LIKE '%prova%' OR colonna LIKE '%di un "testo" raggruppato%')

MakeSafe(string)
Crea una versione "Sql Injection Safe" di una stringa (raddoppia l'apice singolo) da usare tutte le volte che si crea dinamicamente uno statement SQL e, specificamente, quando si scrive qualcosa del genere:

```
string mysql = "'" + RER.Tools.Sql.MakeSafe(myVar) + "'";
```

10.2 RER.Tools.StringWriterWithEncoding

E' una semplice wrapper della classe StringWriter che permette di indicare l'encoding (che per lo StringWriter di sistema è sempre UTF16). Ciò risulta utile per produrre file XML tramite XmlTextWriter

per poi fisicizzarli sul file system. Esempio:

```
private string CommandLineArguents2XML()
{
        string[] arguments = Environment.GetCommandLineArgs();

        StringWriterWithEncoding xml = new StringWriterWithEncoding(Encoding.UTF8);
        XmlTextWriter writer = new XmlTextWriter(xml);
        writer.Formatting = Formatting.Indented;
        writer.WriteStartDocument();
        writer.WriteStartElement("CommandLineArguments");
        for (int i = 1; i < arguments.Length; i++)
        {
                writer.WriteStartElement("Argument");
                writer.WriteAttributeString("position", i.ToString());
                writer.WriteCData(arguments[i]);
                writer.WriteEndElement();
        }
        writer.WriteEndElement();
        writer.WriteEndDocument();
        writer.Close();
        return xml.ToString();
}
```

10.3 RER.Tools.Security.ImpersonateUser

Questa classe serve per impersonare un utente di cui si conoscono le credenziali. È un wrapper per la chiamate di sistema LogonUserA di advapi32.dll. Contiene semplicemente i seguenti due metodi:

bool ImpersonateValidUser(string userName, string domain, string password)
Cambia il contesto di sicurezza del thread corrente con quello dell'utente associato alle credenziali passate come paremetri. Restiusce true se le credenziali sono corrette e l'*impersonation* ha avuto luogo. Altrimenti restituisce false.

void UndoImpersonation()
Da chiamare dopo avere eseguito l'impersonation per ritornare al contesto di sicurezza precedente

Esempio

```
//using System.Security.Principal;
Security.ImpersonateUser impersonationUser = new Security.ImpersonateUser();
if (impersonationUser.ImpersonateValidUser("Pinco_P", "RERSDM", "blablabla"))
{
        Console.WriteLine("OK!");
        Console.WriteLine("Current user: {0}", WindowsIdentity.GetCurrent().Name);
        Console.WriteLine("Undoing impersonation...");
        impersonationUser.UndoImpersonation();
        Console.WriteLine("Current user: {0}", WindowsIdentity.GetCurrent().Name);
}
```

10.4 RER.Toos.UrlNormativa

E' una classe che ha il compito di creare gli URL per accedere ai testi dei varia normativa a partire da un numero e da un anno. L'uso è molto semplice e completamente "statico" (nel senso che non occorre istanziare nessuna classe): vi serve calcolare l'URL per vedere (per esempio) la legge regionale 45 del 2004? Il metodo che vi restituisce l'URL è:

miolink.NavigateUrl = RER.Tools.UrlNormativa.**LeggeRegionale**.CreaLink(**2004**, **45**)

Se anziché il testo di una legge regionale vi serve quello ad una delibera (diciamo la 232 del 2005):

miolink.NavigateUrl = RER.Tools.UrlNormativa.**DeliberaGiuntaRegionale**.CreaLink(**2005**, **232**)

La classe raggruppa una serie di "tipi di normativa", al momento:
LeggeRegionale
DeliberaConsiglioRegionale
DeliberaGiuntaRegionale
DeliberaGiuntaRegionaleDaIntranet (su internet le delibere se vedono solo dopo che sono pubblicate, su intranet appena approvate, per questo ci sono due link distinti)
DeterminazioneDelDirigente
DeterminazioneDelPresidente
DeterminazioneDellAssessore
BollettinoUfficialeRegionale

10.5 Creazione di documenti PDF lato server

La Regione Emilia-Romagna ha acquistato la licenza di 2 componenti per la creazione dei documenti PDF:

- IBEX PDF Creator (v. 4.5.0.3): per la generazione via XSL-FO
- XMLPDF (v. 4.9.0): per la generazione tramite dei template XML (più semplice di XSL-FO, ma meno potente)

Per i dettagli si rimanda al sito del produttore dei componenti http://www.xmlpdf.com

Tutte le volte che si crea un oggetto xmlpdf.PDFDocument, indicare il file della licenza nel seguente modo:

```
ibex4.FODocument foDocument = new FODocument();
ibex4.licensing.Generator.LicenseFileLocation =
    RER.Tools.Configuration.IbexPdfCreatorLicenseFileLocation;
```

Tutte le volte che si crea un oggetto xmlpdf.PDFDocument, indicare il file della licenza nel seguente modo:

```
xmlpdf.PDFDocument doc = new xmlpdf.PDFDocument();
xmlpdf.licensing.Generator.LicenseFileLocation =
   RER.Tools.Configuration.XmlPdfLicenseFileLocation;
```

11 Installazione e configurazione di RER.Tools

Assieme a questo documento è possibile richiedere i seguenti file:

- RER.Tools.dll
- RER.Tools.ApplicationLogger.dll
- RER.Tools.config
- ApplicationLoggerDBSetup.sql

Gli assembly che compongono il namespace RER.Tools.* sono 2:

- RER.Tools.dll
- RER.Tools.ApplicationLogger.dll

Sono assembly con *strongname*, quindi è possibile (nonché consigliabile) installarli nella GAC (Global Assembly Cache).

A questi assembly sono associate alcune sezioni di configurazione. Esattamente come è consigliabile installare gli assembly nella GAC è analogamente consigliabile aggiungere tali sezioni direttamente

nel *machine.config*.

L'intera sezione di configurazione è contenuta nel file RER.Tools.config. Naturalmente se si decide di aggiungere tali configurazioni nel machine.config occorre distribuirle coerentemente con quanto già definito nel proprio machine.config (ciò vale fondamentalmente per la sezione <sectionGroup name="RER"> che andrà aggiunta alla esistente elemento <configSections> del machine.config).

Tali sezioni di configurazione richiedono l'immissione di alcune informazioni personalizzate. Ci sono commenti autoesplicativi, comunque i punti in cui intervenire sono indicati dalla presenza della stringa **!TODO!** e in particolare occorre:

- indicare il nome del server SMTP
- indicare la stringa di connessione per il DB dell'ApplicationLogger (vedere paragrafo 11.1sotto)
- indicare la/le *email* a cui inviare le notifiche dell'ApplicationLogger

La sezione di configurazione più importante di tutte è <SqlServerInstanceMappings>. Essa serve al SqlConnectionBroker per decidere a quale istanza di Sql Server richiedere una connessione. Di default è presente un solo *mapping*:

<Mapping defaultDataSource="(local)" />

che suppone che il DB server sia sulla stessa macchina in cui risiede l'applicazione. Sostituire a "(local)" il nome della propria istanza di SqlServer se, in locale, non sia installato il DB server.
L'utilizzo di più elementi <Mapping> permette la semplificazione del deployment delle applicazioni nel caso si abbia una infrastruttura hardware più articolata (DB server di sviluppo e produzione, Web Farms, …). In tal caso fare riferimento ai commenti presenti nel file RER.Tools.config oppure ai tecnici regionali.

11.1 DB per l'ApplicationLogger

Il componente RER.Tools.ApplicationLogger *logga* gli eventi da esso gestiti in un DB, tra i file allegati è presente anche ApplicationLoggerDBSetup.sql che contiene lo script SQL necessario alla creazione del DB.
Identificare il DB Server in qui si vuole installare questo DB, in esso creare:

- Un nuovo database vuoto di nome "ApplicationLogger"
- Un nuovo login di nome "usrApplicationLogger"
- Lanciare lo script ApplicationLoggerDBSetup.sql
- Modificare di conseguenza la stringa di connessione nella sezione di configurazione RER\GlobalSettings\ ApplicationLoggerConnStr.

Allegato 4: Linee guida per l'interoperabilità tra Application Server J2EE.

1 Introduzione

Scopo di questo documento è quello di fornire alcune regole e suggerimenti di base da seguire per minimizzare il lavoro da effettuare per migrare applicazioni J2EE da un application server ad un altro application server.
Questo documento prende in considerazione 2 application server: IBM WebSphere Application Server e JBoss Application Server, in modo da limitare il numero di specifiche da prendere in esame.

2 Specifiche J2EE supportate dai vari application server

Si prendono qui in considerazione a titolo informativo le seguenti versioni degli application server:
- IBM WebSphere Application Server v6.0 e v6.1
- JBoss Application Server v4.0.5

Tutte le versioni elencate di application server sono certificate almeno riguardo alle specifiche J2EE 1.4.
La versione di specifiche J2EE 1.4 determina le seguenti versioni riguardo alle specifiche di dettaglio:

Sigla	Descrizione completa	Versione
JCA	J2EE Connector Specification	1.5
J2EE Deployment	J2EE Deployment API	1.1
J2EE Management	J2EE Management	1.0
EJB	Enterprise JavaBeans	2.1
EJB to CORBA Mapping	Enterprise JavaBeans to CORBA Mapping	1.1
JAXP	Java API for XML Processing	1.2
JAXR	Java API for XML Registries	1.0
JAX-RPC	Java API for XML-based RPC	1.1
JACC	Java Authorization Contract for Containers	1.0
Java IDL	Java IDL API	
JNDI	Java Naming and Directory Interface	1.2
JMS	Java Message Service	1.1
Servlet	Java Servlet	2.4
JTA	Java Transaction API	1.0.1
JTS	Java Transaction Service	1.0
JMX	Java Management Extensions	1.2
Web Services	Web Services	1.1
JDBC	Java DataBase Connectivity	3.0
JAF	JavaBeans Activation Framework	1.0
JavaMail	JavaMail API	1.3
JSP	JavaServer Pages	2.0
RMI/IIOP	RMI over IIOP	1.0
SAAJ	SOAP with Attachments API for Java	1.2

Tabella 1 - Versioni delle specifiche J2EE 1.4

2.1 Scelta del JDK

Le specifiche J2EE 1.4 indicano il JDK 1.4 come ambiente di riferimento.
Sotto certe condizioni è possibile utilizzare anche il JDK 1.5:
- WebSphere v6.0 supporta **solo il JDK 1.4**
- WebSphere v6.1 supporta anche il JDK 1.5

- JBoss 4 supporta sia il JDK 1.4 che il JDK 1.5

Poiché i nuovi application server da supportare sono **WebSphere v6.1** e **JBoss 4** si suggerisce comunque l'utilizzo del **JDK 1.5** in quanto offre dei miglioramenti dal punto di vista delle prestazioni e maggiori possibilità di monitoraggio rispetto al JDK 1.4.

3 Best practices

Per minimizzare gli sforzi da effettuare quando si vuole migrare un'applicazione da un application server ad un altro è buona norma seguire le seguenti regole:

- Evitare di utilizzare librerie proprietarie dell'Application Server (ad es. per effettuare chiamate di sistema allo scopo di reperire la connessione nativa dal datasource, ma anche l'utilizzo di oggetti proprietari al posto di oggetti standard).
- Adottare le API specificate nella tabella 1, evitando assolutamente di adottare versioni superiori (anche qualora fossero supportate dall'application server) e possibilmente evitare di adottare versioni inferiori in quanto anche se è vero che di solito le specifiche sono retrocompatibili dal punto di vista binario, in alcuni possono non esserlo dal punto di vista della compatibilità del sorgente (ad esempio JDBC 3 contiene delle modifiche non compatibili con JDBC 2).
- Possono verificarsi problemi legati alla diversa gestione dei class loaders nei diversi application server: il classloader di default di JBoss è abbastanza diverso dal classloader di WebSphere. Entrambi gli application server hanno però ampie possibilità di configurazione del classloading, pertanto dovrebbe essere possibile risolvere questi problemi intervenendo sulla configurazione.
- Possono anche verificarsi problemi legate alle diverse librerie installate nei server a livello di runtime (ad esempio in entrambi i server sono presenti le librerie di Xerces, ma con versioni leggermente diverse). Anche questo tipo di problemi vengono di solito risolti intervenendo sulla configurazione del classloading.

In ogni caso, anche seguendo le suddette regole il porting non è immediato in quanto alcune risorse sono comunque non portabili: ad esempio i deployment descriptor delle applicazioni hanno formati specifici dei singoli application server, pertanto non sono portabili e richiedono un adattamento o una riscrittura.
Esistono inoltre alcune linee guida relative alla portabilità per alcune specifiche tecnologie.
Di seguito vengono riportate le linee guida estratte dal documento di Sun "Designing Enterprise Applications with the J2EE platform".

3.1 Enterprise Javabeans

3.1.1 Typecast Remote References

A client program that needs to access a remote enterprise bean must use the *PortableRemoteObject.narrow* method for type narrowing. Type narrowing is needed when a client program looks up a home interface from JNDI, or a finder method returns a collection of references to remote enterprise beans.
The following code example shows how to do type narrowing when looking up a home object from JNDI.

```
try {
        Context ctxt = new InitialContext();
        Object objref = ctxt.lookup("java:comp/env/ejb/remote/admin");
        OrderProcessingCenterAdminFacadeHome adminHome =
        (OrderProcessingCenterAdminFacadeHome) PortableRemoteObject.
        narrow (objref, OrderProcessingCenterAdminFacadeHome.class);
```

```
        OrderProcessingCenterAdminFacade admin = adminHome.create();
} catch (....) {
```

Code Example - Using the narrow Method for Type Narrowing

Type narrowing is needed because many application servers use RMI-IIOP as the communication protocol to access remote beans. However, some application servers do not use RMI-IIOP and hence allow the use of Java language typecasts as well. For portability you cannot rely on an application server allowing Java language typecasts; you should always use the PortableRemoteObject.narrow method. The overhead on this method call is usually quite small. In addition, EJB containers that do not use RMI-IIOP typically optimize away all such overhead.
Note that the narrow method must not be used in the clients of local enterprise beans since they are in the same JVM. The local enterprise beans are always typenarrowed using the regular typecast of the Java programming language.

3.1.2 Mark Non-Serializable Fields Transient

To preserve a bean's state during passivation, the bean class must be serializable.
This requires that all the non-transient fields of the bean class are serializable. Fields that are primitive types, such as String and int, are serializable. However, a reference field, which is a field whose value is a reference to a class instance, is serializable only if the referenced class implements java.io.Serializable. You must mark all fields of non-serializable types as transient. The JVM's serialization machinery ignores fields marked as transient.
For example, a database connection represented by java.sql.Connection is not serializable. It must be marked transient when declared inside an enterprise bean class.

3.1.3 Bean-Managed Persistence and Portability

Extra effort is required to achieve portability for an enterprise bean that uses bean managed persistence, because the bean needs to ensure portability across all databases as well as JDBC drivers.
The foremost factor affecting portability relates to the SQL language. Many database vendors provide proprietary extensions to SQL to provide additional functionality and to achieve higher performance. Consider using only standard SQL constructs to achieve portability. If you do need to use proprietary extensions, consider using the Data Access Object design pattern to encapsulate vendor-specific code.

3.1.4 SQL and Database Connections

For maximum portability, it's important to close SQL statements before you close the database connection. Enterprise beans often need to open a database connection, execute a set of SQL statements, then close the connection. Some JDBC driver implementations throw an exception if a JDBC connection is closed while some of the driver database statements are open. To achieve portability across JDBC drivers, always close database statements before closing the database connection. The finally block in the following Code Example illustrates how this can be done.

```
public Page searchItems(String searchQuery, ....) {
        Connection con = null;
        PreparedStatement ps = null;
        ResultSet rs = null;
        String query = "SELECT .....";
        try {
                con = dataSource.getConnection();
                ps = con.prepareStatement(query);
                rs = ps.executeQuery();
                // rest of the method body
        } catch (.... ) { // handle exceptions
        } finally {
```

```
                    if (rs != null) rs.close();
                    // Close PreparedStatement before Connection.
                    if (ps != null) ps.close();
                    if (c != null) c.close();
}
...
}
```

Code Example - Closing Database Connections

3.1.5 Relying on Instance Fields

Bean providers should not rely on a bean's instance fields or container-managed persistence accessor methods within ejbActivate, ejbLoad, ejbPassivate, and ejbStore methods. This is because the container can choose several ways to manage the life cycle of its enterprise beans. For example, in the ejbActivate method the container is not required to load an entity bean's instance fields from its persistence store. Similarly, in the ejbPassivate method the container is not required to store the instance fields to its persistence store. In addition, the container is not required to allow accesses to resources from the ejbActivate or ejbPassivate methods.

3.1.6 Avoid Exposing Resource-Specific Details

Bean providers should be especially careful to avoid including backend resource-specific details in their components' interfaces, since doing so may limit where the components might be used. One easily overlooked form of resource dependence is the set of exceptions a method may throw. Because bean-managed persistence methods do not necessarily use a SQL database to manage their persistence, SQLException should not be thrown in the bean-managed persistence method signatures.
Instead of throwing SQLException, define system- and application-level exceptions for the class and throw those exceptions in response to error conditions.
While using the Data Access Object (DAO) design pattern, catch the resource-specific exceptions, such as SQLException, in the DAO class and translate them to appropriate system-level or application-level exceptions.
Consider the following Code Example from the sample application. In the method searchItems, an SQLException is translated to a CatalogDAOSysException, which extends java.lang.RuntimeException to indicate a system-level exception.

```
public class CatalogDAOImpl implements CatalogDAO {
        ....
        public Page searchItems(String searchQuery, int start,
        int count, Locale l)
        throws CatalogDAOSysException {
        ....
        try {
                Connection con = getDBConnection();
                PreparedStatement ps = con.prepareStatement("SELECT ...");
                ...
                ps.executeQuery();
                ...
        } catch (SQLException se) {
                throw new CatalogDAOSysException("Malformed query.");
        }
        ...
}
```

Code Example - Throwing Exceptions

The code throws an application exception if the user input to searchQuery is incorrect. For errors such as an unavailable database connection, or general SQL exceptions, a system exception should be thrown.

3.2 J2EE Naming Environment

The JNDI naming context is the component's API for accessing the naming environment. To avoid collision with names of other enterprise resources in JNDI, and *to avoid portability problems*, all names in a J2EE application should begin with the string **java:comp/env**. Because all instances of a particular application component share naming environment entries, components may not change values in the naming context.

The J2EE recommends (but does not require) the following structure for the "**env**" namespace:

- Enterprise JavaBeans™ are placed under the "**ejb**" subtree. For example, a Payroll EJB might be named "java:comp/env/ejb/Payroll".
- Resource factory references are placed in subtrees differentiated by their resource manager type. Here are some examples:
  - "**jdbc**" for JDBC™ DataSource references
  - "**jms**" for JMS connection factories
  - "**mail**" for JavaMail connection factories
  - "**url**" for URL connection factories

For example, a JDBC Salary database might have the name "java:comp/env/jdbc/Salary".

The name "**UserTransaction**" is bound to a *javax.transaction.UserTransaction* object. The component that looks up this object from the namespace (by using the name "java:comp/UserTransaction") can use it to start, commit, or abort transactions.

3.3 Security

3.3.1 Programmatic Authorization

A J2EE container makes access control decisions before dispatching method calls to a component. The logic or state of the component doesn't factor in these access decisions. However, a component can use two methods, *EJBContext.isCallerInRole* (for use by enterprise bean code) and *HttpServletRequest.isUserInRole* (for use by Web components), to perform finer-grained access control. A component uses these methods to determine whether a caller has been granted a privilege selected by the component based on the parameters of the call, the internal state of the component, or other factors such as the time of the call.

The application component provider of a component that calls one of these functions must declare the complete set of distinct roleName values to be used in all calls. These declarations appear in the deployment descriptor as securityrole-ref elements. Each security-role-ref element links a privilege name embedded in the application as a roleName to a security role. Ultimately, the deployer establishes the link between the privilege names embedded in the application
and the security roles defined in the deployment descriptor. The link between privilege names and security roles may differ for components in the same application.
In addition to testing for specific privileges, an application component can compare the identity of its caller, acquired using *EJBContext.getCallerPrincipal* or *HttpServletRequest.getUserPrincipal*, to
the distinguished caller identities embedded in the state of the component when it was created.
If the identity of the caller is equivalent to a distinguished caller, the component can allow the caller to proceed. If not, the component can prevent the caller from further interaction.
The caller principal returned by a container depends on the authentication mechanism used by the caller. Also, **containers from different vendors may return different principals for the same user authenticating by the same mechanism**.
To account for variability in principal forms, an application developer who chooses to apply distinguished caller state in component access decisions should allow multiple distinguished caller identities, representing the same user, to be associated with components. This is recommended especially where application flexibility or portability is a priority.

3.4 Esempio di codice non portabile

Il seguente esempio è relativo a del codice non portabile tra i vari application server, in quanto è specifico di WebSphere.
Il caso è quello della scrittura di un BLOB su un database Oracle. Esistono 2 modalità per effettuare questa operazione:

1. Si inserisce una riga contenente il BLOB vuoto, poi si seleziona la riga con una SELECT FOR UPDATE e si inserisce il contenuto nel BLOB tramite uno stream.
2. Si inserisce la riga tramite una INSERT, con il BLOB già valorizzato in modo da evitare di fare 2 operazioni.

La seconda modalità richiede che sia effettuata la chiamata al metodo **createTemporary** della classe BLOB di Oracle che richiede come parametro la connessione JDBC al database, tale BLOB va poi riempito con i metodi del BLOB e infine può essere invocata la INSERT.
Il problema è che la connessione al database di solito non è disponibile all'applicazione in quanto normalmente quello che viene fornito all'applicazione è un wrapper che permette al server di gestire correttamente il pool.
Su WebSphere il problema viene risolto tramite delle apposite classi di utilità che permettono di invocare il metodo *createTemporary* (o altri metodi) al posto del nostro codice, potendo così fornire anche la connessione JDBC nativa.
Per completezza si riporta un esempio di tale codice:

```
// create a new temporary BLOB
BLOB tempBlob =
        (BLOB) WSCallHelper.jdbcPass(
                BLOB.class,
                "createTemporary",
                new Object[] {
                        dataConnection,
                        Boolean.TRUE,
                        new Integer(BLOB.DURATION_SESSION)},
                new Class[] {
                        java.sql.Connection.class,
                        boolean.class,
                        int.class },
                new int[] {
                        WSCallHelper.CONNECTION,
                        WSCallHelper.IGNORE,
                        WSCallHelper.IGNORE });
```

Questo codice per quanto efficiente non è portabile su altri application server, e probabilmente anche su diverse versioni di WebSphere può essere soggetto a cambiamenti.

Da questo esempio si desume che il vincolo della portabilità ha sicuramente un impatto su come devono essere scritte le applicazioni, ma tale impatto è minimo e di solito non è tale da pregiudicare le prestazioni dell'applicazione e lo sforzo necessario per implementare i servizi.

4 Strumenti automatici di controllo

Sun fornisce due strumenti denominati:

1. "Java™ Application Verification Kit (AVK) for the Enterprise 1.4.2": che permette di verificare la compatibilità di applicazioni J2EE almeno con le specifiche J2EE 1.4.
2. Java Application Verification Kit (AVK) for the Enterprise 5": che permette di verificare la compatibilità di applicazioni J2EE almeno con le specifiche J2EE 1.5.

Tali strumenti effettuano le seguenti operazioni:

- **Controllo statico delle applicazioni** per verificare che siano aderenti alle specifiche J2EE. Questo tool identifica e riporta problemi legati alle firme dei metodi contenuti nei moduli applicativi e altre inconsistenze rilevabili in modo statico (senza eseguire l'applicazione). Verifica tutti i componenti applicativi (contenuti in un EAR, JAR, WAR, o RAR) per trovare inconsistenze con le API e i descrittori definiti nelle specifiche J2EE. Verifica anche il packaging e la struttura dei moduli.
- **Controllo dinamico delle applicazioni**: consiste nell'installare l'applicazione sul Sun Application Server ed effettuare un controllo manuale sul funzionamento dell'applicazione. E' comunque un task poco applicabile al contesto di Regione Emilia Romagna in quanto richiede comunque la traduzione dei deployment descriptors ed inoltre una volta passato il controllo statico tanto vale effettuare il test direttamente sull'application server di destinazione, senza passare per il Sun Application Server.

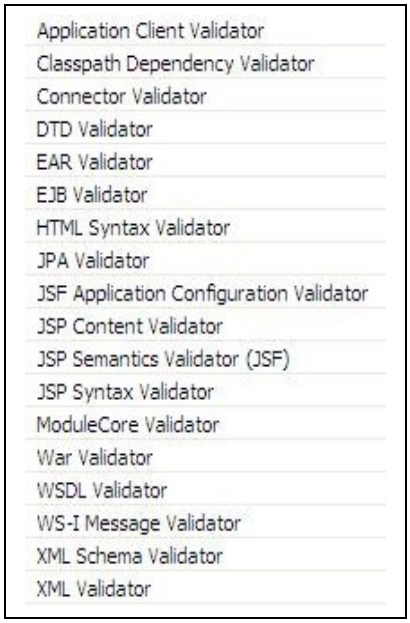
Lungi dall'essere uno strumento perfetto, questo tool può comunque essere di ausilio per effettuare alcuni controlli sulle applicazioni prima di passarle in produzione.
Le specifiche coinvolte nel controllo sono:

- Enterprise JavaBeans™
- Java Servlet API
- JSP
- Web Services for the J2EE platform

Il suggerimento è quello di predisporre un team di persone che sappiano utilizzare correttamente lo strumento e fare in modo che tale team fornisca il supporto per l'uso dell'AVK ai vari team di sviluppo.
L'uso dell'AVK dovrebbe pertanto essere a carico dei team di sviluppo, con il supporto degli "specialisti AVK".
Non si ritiene adatto l'uso dell'AVK al personale sistemistico in quanto le segnalazioni prodotte dall'AVK sono strettamente legate agli standard J2EE sui quali i sistemisti possono non avere le competenze necessarie.

5 Ambienti di sviluppo

Negli ambienti *Eclipse*, *WebSphere Studio Application Developer* (*WSAD)* e *Rational Application Developer* (*RAD)* sono presenti dei validatori che permettono di effettuare una serie di controlli già in fase di sviluppo. Ecco un esempio dei validatori presenti in Eclipse 3.3:



Si consiglia di attivare questi validatori in modo da effettuare automaticamente i relativi controlli.

L'utilizzo di ambienti di sviluppo proprietari come *WebSphere Studio Application Developer* o *Rational Application Developer* può portare a sviluppare servizi non portabili in quanto in alcuni casi i wizard presenti in tali ambienti si appoggiano a features specifiche di WebSphere Application Server.

L'uso dell'analogo ambiente per JBoss, JBoss Tools (che in precedenza si chiamava JBoss IDE e nella versione finale si chiamerà Red Hat Developer Studio) non crea problemi di questo tipo in quanto le tecnologie fornite da JBoss Tools (Hibernate, Drools, JBPM, etc) sono disponibili anche separatamente dall'application server e sono portabili.

Si raccomanda pertanto di limitare l'uso dei wizard automatici di *WSAD* e *RAD* a meno che non si sappia a priori che il codice generato, le librerie su cui tale codice si appoggia, e le relative configurazioni sono aderenti alle specifiche J2EE 1.4 o 1.5 e pertanto portabili.

6 Utilizzo di stored procedure

Si ritiene opportuno in questo contesto dare anche alcune indicazioni sull'utilizzo delle stored procedure nello sviluppo delle applicazioni.
Vi sono ragioni a supporto dell'utilizzo delle stored procedure:

- Elaborazioni massive: quando i dati da elaborare sono molti l'uso delle stored può rivelarsi conveniente dal punto di vista della velocità di elaborazione, in quanto viene minimizzato l'overhead di rete necessario per la trasmissione dei dati dal database al nodo in cui i dati vengono elaborati.

Ci sono anche ragioni per non utilizzarle:

- Non portabilità del codice su diversi database: il codice delle stored di solito non è portabile tra diversi database, costringendo alla riscrittura delle stesse nel momento in cui si voglia migrare il database.
- Minore scalabilità dell'applicazione: se il database è l'elemento in cui vengono effettuate tutte le elaborazioni, diventa complesso distribuire il carico con impatto sulla scalabilità. Va anche tenuto presente che normalmente un database server serve più application server.

Il suggerimento è quello di bilanciare le diverse esigenze e adottare le stored solo quando motivi di efficienza dal punto di vista delle prestazioni lo richiedano, in quanto devono essere manipolate quantità ingenti di dati (decine di migliaia di righe ed oltre), evitando possibilmente di implementare completamente la logica applicativa dentro alle stored, lasciando all'applicazione Java solo la responsabilità di gestire il front-end.

7 Bibliografia

- Java™ 2 Platform Enterprise Edition Specification, v1.4 (Sun Microsystems)
- Designing Enterprise Applications with the J2EE™ Platform, Second Edition (Sun Microsystems) http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/
- WebSphere Application Server v6 Migration Guide (IBM Redbooks)
- WebSphere Application Server Information Center http://publib.boulder.ibm.com/infocenter/was-info/v6r0/index.jsp
- Java Application Verification Kit (AVK) for the Enterprise http://java.sun.com/j2ee/verified/avk_enterprise.html

Allegato 5: Clausola "accessibilità" per contratti e capitolati tecnici

Quando si scrive un contratto o un capitolato tecnico per l'acquisizione di un prodotto o servizio web (sito, applicazione o CD-ROM/DVD) è necessario inserire una clausola che preveda il rispetto dei requisiti di accessibilità.

Il contenuto di tale clausola deve essere simile a questo:

Accessibilità

La realizzazione/modifica/fornitura del sito/applicazione/prodotto/servizio oggetto del contratto dovrà rispondere ai criteri di accessibilità stabiliti dalla Legge 9 Gennaio 2004, n. 4, "Disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici" e successive integrazioni e variazioni, in particolare dal Decreto Ministeriale 8 agosto 2005 - ***Allegato A****.*

Il prodotto/Servizio fornito dovrà inoltre rispettare le indicazioni esposte nelle "Linee Guida per realizzare siti e applicazioni web accessibili per la Regione Emilia-Romagna", nella versione più aggiornata reperibile online all'indirizzo: http://www.regione.emilia-romagna.it/lineeguida/

Il rispetto dei requisiti di accessibilità verrà verificato dal cliente all'atto della consegna da parte del fornitore, e sarà poi accertato dal Servizio SIIR attraverso le verifiche preliminari alla presa in carico, prima della messa online del sito e delle applicazioni o di loro modifiche sostanziali. L'Amministrazione inoltre si riserva in qualunque momento, su propria iniziativa o su segnalazione di terzi, di effettuare verifiche di accessibilità ed usabilità sui servizi web oggetto del presente contratto resi dal fornitore, il quale dovrà provvedere, senza ulteriori oneri per l'Amministrazione, alla messa a norma di quanto eventualmente riscontrato difforme a seguito di tali verifiche.

Anche quando si scrive un contratto o un capitolato tecnico per l'acquisizione di beni/servizi che riguardano sistemi operativi, applicazioni o prodotti a scaffale è necessario inserire una clausola che preveda il rispetto dei requisiti di accessibilità.

Il contenuto di tale clausola deve essere simile a questo:

Accessibilità

Il prodotto/Servizio oggetto del contratto dovrà rispondere ai criteri di accessibilità stabiliti dalla Legge 9 Gennaio 2004, n. 4, "Disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici" e successive integrazioni e variazioni, in particolare dal Decreto Ministeriale 8 agosto 2005 - ***Allegato D****.*

Il rispetto dei requisiti di accessibilità verrà verificato dal cliente sulla base di quanto dichiarato a tal proposito dal fornitore, e sarà poi accertato dal Servizio SIIR attraverso le verifiche preliminari alla presa in carico.

Allegato 6: Lista dei requisiti di accessibilità

La verifica del rispetto dei requisiti va fatta per tutte le pagine del sito o dell'applicazione.
La versione più aggiornata della checklist è scaricabile dalle pagine contenenti le linee guida: **http://www.regione.emilia-romagna.it/lineeguida/**

Requisiti delle pagine (X)HTML

- **TUTTE le pagine** devono avere codice valido (almeno XHTML 1.0 Strict per le nuove realizzazioni).
- **Definire la lingua prevalente della pagina attraverso l'attributo LANG del tag HTML:** questo è utile sia ai motori di ricerca che ai più recenti screen readers che possono così impostare automaticamente il sintetizzatore vocale per parlare nella lingua corretta. I codici di lingua più comuni sono: IT per l'italiano, EN per l'inglese, FR per il francese, ES per lo spagnolo, DE per il tedesco.
- **Ogni pagina deve avere un titolo significativo (non più di 40 caratteri esclusi quelli del nome del sito):**
  che permetta all´utente di riconoscere il contesto: se il titolo si compone di più voci, partire sempre dalla più specifica per arrivare alla più generica.
- **Ogni pagina deve avere dei link (anche nascosti) per saltare al contenuto o al menù:** questi link si devono trovare subito dopo l'apertura del <body> e possono essere associati a degli *accesskey*. Sono utili a chi non vede per raggiungere rapidamente punti significativi della pagina.
- **Ogni pagina deve avere un link per tornare alla Home Page**:
  meglio ancora se contiene il **percorso di navigazione** (la lista dei link alle pagine da attraversare per raggiungere la pagina visualizzata, a partire dalla home page, vedi esempi).
- **Se necessario deve esserci la corretta titolatura** dei differenti livelli ed importanza (da h1, **il titolo dei contenuti della pagina**, ad h6):
  Un titolo è corretto che preceda un blocco di contenuti (un menù o un paragrafo) ma non che sia un link di un menù; la struttura della pagina è utile ai motori di ricerca ed è utile per navigare i contenuti anche saltando da una intestazione all´altra.
- **Tutti (e solo) gli elenchi di elementi devono essere marcati come <ul>, <ol> o <dl>:** gli elementi di lista si devono usare solo per elenchi e non per ottenere particolari effetti grafici.

Requisiti di menu e liste

- **Tutti gli elenchi di voci di menù devono essere fatti in testo e marcati come <ul> o **
  (in questo modo gli elementi sono marcati secondo la loro semantica, di lista appunto).

- **La distanza verticale o orizzontale tra i link del menù deve essere di almeno 0,5em** per mantenere una migliore leggibilità e facilità nel selezionare i singoli elementi.
- **Se esiste più di un menù, marcare con un'intestazione il titolo di ciascuno di essi** e non marcare come intestazioni le singole voci dei menù.

Requisiti dei CSS

- **I CSS devono avere codice valido**.
- **I contenuti e le funzionalità della pagina devono essere ancora fruibili e mantenere il loro significato d'insieme o la loro corretta struttura semantica anche quando si disabilitano i fogli di stile**.
- **Tutte le caratteristiche legate all'aspetto si devono trovare solo in fogli di stile esterni** non è ammesso l'uso di stili *inline* o all'interno della pagina.
- **Se i contenuti possono venire stampati, deve essere presente uno stile per la stampa** di norma questo vale per i siti Web e non per le applicazioni.
- **Le dimensioni dei font e degli elementi dei campi di input devono essere specificate e devono essere definite con unità di misura relative (em o %)**
  alcuni browser (es. Internet Explorer 6) non sono in grado di ingrandire correttamente i caratteri o i campi dei moduli se non si utilizzano unità di questo tipo.
- **I contenuti si devono adattare alle preferenze dell'utente**
  si devono poter ingrandire i caratteri o cambiare le dimensioni della finestra del browser senza che ci sia perdita di informazioni; è possibile realizzare pagine elastiche che riempiono cioè tutta la pagina, o pagine ingrandibili, di dimensioni fisse ma che si ingrandiscono con l'aumentare delle dimensioni dei font sui browser

Requisiti dei frame

Non è consentito l'uso dei frame (o degli iframe) nella realizzazione di nuovi siti

Requisiti per i colori

- **Funzioni ed informazioni devono essere disponibili anche in assenza del particolare colore utilizzato per presentarli nella pagina.**
- **Si deve sempre distinguere il contenuto informativo (foreground) e lo sfondo (background), ricorrendo a un sufficiente contrasto**

Requisiti del testo

- **Le dimensioni dei caratteri delle pagine devono essere espresse in em o %.**
- **I testi devono restare comprensibili** (senza sovrapposizioni o perdita di informazioni) anche su Internet Explorer 6, ad 800x600 impostato coi caratteri grandi.

Requisiti delle immagini

- **Fornire una alternativa testuale equivalente per ogni immagine,**
  l'alternativa testuale deve essere commisurata alla funzione o contenuto dell'immagine.

- **Sono vietati oggetti e scritte lampeggianti o in movimento**.
- **Testi in forma di immagini** sono ammessi solo per loghi o banner.
- Le immagini protette da **© Copyright** dovranno avere l´esplicito consenso dell´autore.
- Se presenti, il **logo della Regione o di Ermes** devono essere inseriti senza modifiche all'immagine, collegati al portale e con il corretto alternativo.

Requisiti delle mappe immagine

- **Utilizzare mappe immagine sensibili di tipo lato client**
  salvo il caso in cui le zone sensibili non possano essere definite con una delle forme geometriche.
- **In caso di utilizzo di mappe immagine lato server, fornire i collegamenti di testo alternativi** necessari per ottenere tutte le informazioni o i servizi raggiungibili interagendo direttamente con la mappa.

Requisiti dei link

- **Usare testi che siano significativi anche se letti indipendentemente dal proprio contesto**.
- **Rendere selezionabili e attivabili tramite la tastiera i collegamenti presenti in una pagina.**
- **La distanza verticale di liste di link e la spaziatura orizzontale tra link consecutivi deve essere di almeno 0,5 em.**

Requisiti delle tabelle di dati

- **Identificare le intestazioni di righe e colonne**.
- Nelle tabelle complesse (con più di un livello di intestazione) **associare le celle di dati e le celle di intestazione**

Requisiti delle tabelle di layout

- Il **contenuto della tabella deve essere comprensibile anche quando questa viene letta in modo linearizzato.**
- Usare solo <td> e non utilizzare elementi e attributi delle tabelle dati (es. th, scope, tbody ecc.).

Requisiti dei form

- **Associare in maniera esplicita le etichette ai rispettivi campi**.
- **Posizionare le etichette vicino ai campi (o sopra, o immediatamente a lato)**.
- **Le distanze orizzontale e verticale tra i pulsanti di un modulo deve essere di almeno 0,5em.**.

- **Le dimensioni dei pulsanti devono essere tali da rendere chiaramente leggibile l'etichetta in essi contenuta** (padding di almeno 0,5em ai lati).
- **Si deve poter ingrandire il carattere anche nei campi dei moduli** (su IE5.5, IE6).

Requisiti dei contenuti multimediali

- **Fornire un'alternativa testuale equivalente.**
- **Garantire che siano sempre distinguibili il contenuto informativo e lo sfondo ricorrendo a differenti livelli sonori** (in caso di parlato con sottofondo musicale).

Requisiti di script, applet e plug-in

- **Le pagine devono funzionare anche quando questo tipo di oggetti sono disabilitati o non supportati:** se questo non fosse possibile si deve fornire una spiegazione testuale equivalente.
- **Per la validazione dei dati inseriti in un form è necessaria la verifica lato server:** una verifica preliminare tramite Javascript è utile ma non sufficiente.
- **Funzionalità e informazioni di script e applet devono essere direttamente accessibili**.
- **I gestori di eventi devono essere indipendenti da uno specifico dispositivo di input**: devono funzionare anche con il solo uso della tastiera.
- **Se è previsto un intervallo di tempo predefinito** entro il quale eseguire determinate azioni, **è necessario avvisare l'utente,** e indicare il tempo massimo consentito.

Allegato 7: Liste di controllo per le misure minime di sicurezza

Misure minime da osservare per tutti i trattamenti	
Esiste una procedura di autenticazione che permette l'identificazione univoca dell'utente attraverso opportune credenziali di autenticazione	□
È utilizzata una parola chiave (password), quando prevista dal sistema di autenticazione, composta da almeno otto caratteri	□
Esiste la possibilità di modifica della parola chiave, quando prevista dal sistema di autenticazione, da parte dell'utente al primo utilizzo e, successivamente, almeno ogni sei mesi	□
Esistono meccanismi di disattivazione delle credenziali di autenticazione non utilizzate da almeno sei mesi, salvo quelle preventivamente autorizzate per soli scopi di gestione tecnica	□
I codici di identificazione già impiegati non sono riutilizzati nel tempo assegnandoli ad altri utenti	□
Esistono meccanismi di autorizzazione per la separazione dei privilegi degli incaricati in base a diversi profili autorizzativi	□
Esistono meccanismi di protezione dei dati contro le minacce di intrusione e dell'azione di programmi malevoli (es. cifratura delle password, impiego di firewall o di software antivirus, hardening dei sistemi, ecc.)	□
I programmi sono aggiornati periodicamente per prevenire le vulnerabilità e correggerne difetti (es. patch di sistema, aggiornamenti antivirus, ecc.)	□
Esistono meccanismi di backup e ripristino, con salvataggio dei dati effettuato con frequenza almeno settimanale	□

Misure minime ulteriori da osservare nel caso di trattamenti di dati sensibili e/o giudiziari	
Esiste la possibilità di modifica della parola chiave quando prevista dal sistema di autenticazione, da parte dell'utente al primo utilizzo e, successivamente, almeno ogni tre mesi	□
Esistono meccanismi di ripristino dei dati che permettono la ricostruzione degli stessi, in caso di danneggiamento, in tempi non superiori ai sette giorni	□
Sono utilizzate tecniche di cifratura o codici identificativi, tali da rendere temporaneamente inintelligibili i dati sensibili e/o giudiziari anche a chi è autorizzato ad accedervi e da permettere l'identificazione degli interessati solo in caso di necessità	□

Allegato 8: Clausola "Sicurezza, privacy e riservatezza" per contratti e capitolati tecnici

Quando si scrive un contratto o un capitolato tecnico per l'acquisizione di prodotti o servizi IT è necessario inserire una clausola che preveda la conformità a quanto previsto dal D.Lgs. 196/2003 "Codice in materia di protezione dei dati personali".

Il contenuto di tale clausola deve essere simile a questo:

Sicurezza, privacy e riservatezza

La Ditta aggiudicataria/contraente dovrà garantire all'Amministrazione, che i servizi di supporto informatico e l'eventuale trattamento di dati per conto dell'Amministrazione sono prestati in piena conformità a quanto previsto dal "Codice in materia di protezione dei dati personali", D.Lgs. 196 del 30/06/2003, ed eventuali integrazioni o successive modificazioni e che ai dati trattati sono applicate tutte le misure minime di sicurezza di cui all'Allegato "B" del citato Testo Unico.

Sarà possibile ogni operazione di auditing da parte dell'Amministrazione attinente le procedure adottate dal contraente in materia di riservatezza, protezione di dati e programmi e gli altri obblighi assunti.

L'aggiudicatario/contraente non potrà conservare copia di dati e programmi della Regione Emilia-Romagna, né alcuna documentazione inerente ad essi dopo la conclusione del contratto.

La Ditta aggiudicataria/contraente ha l'obbligo di mantenere riservati i dati e le informazioni di cui venga a conoscenza o in possesso per l'esecuzione del contratto, di non divulgarli in alcun modo, né di farne oggetto di comunicazioni o trasmissioni senza l'espressa autorizzazione dell'Amministrazione.

La Ditta aggiudicataria/contraente risponde nei confronti dell'Amministrazione per eventuali violazioni all'obbligo di riservatezza commesse da propri dipendenti.

Le applicazioni sviluppate nell'ambito del presente capitolato/contratto dovranno soddisfare le indicazioni fornite nel "Disciplinare Tecnico in materia di sicurezza delle applicazioni informatiche nella Giunta della Regione Emilia-Romagna" (determinazione n. 2651/2007).

Inoltre è da prevedere un articolo del capitolato/contratto per la designazione della ditta aggiudicataria/contraente quale responsabile esterno del trattamento di dati personali secondo lo schema scaricabile da Internos nella sezione Privacy/Fac-simili e modulistica.

Allegato 9: Specifiche tecniche per l'utilizzo dei sistemi di autenticazione centralizzata

1 Considerazioni generali

Questo documento ha lo scopo di descrivere le possibili soluzioni centralizzate da adottare per implementare l'autenticazione nelle applicazioni ospitate sulle infrastrutture della Regione Emilia-Romagna:

- Sistema di Identity & Access management (IAM),
- Soluzione custom.

Sistema di Identity & Access management (IAM)

2 Gestione degli utenti di un'applicazione con il sistema di Identity Management

L'applicazione deve delegare al sistema di Identity la gestione degli utenti. L'identity ha bisogno di oggetti chiamati "connettori" che gli permettano di interagire con il repository degli utenti utilizzato dall'applicazione (es. una tabella utenti su un db). In linea generale esistono due tipologie di connettori:

- connettori standard,
- connettori custom.

I primi sono connettori già implementati da Sun. In questo caso non è necessaria alcuna fase di sviluppo e si può passare alla fase di integrazione. Per informazioni circa i connettori standard si rimanda alla documentazione di prodotto e in particolare al documento **Sun Java System Identity Manager 7.1 Resources Reference**.
Nel secondo caso, è invece necessario sviluppare dei metodi **network-enable** (API, Store Procedure, Web Services) che devono essere esposti per l'integrazione con il sistema di IdM.
Tali metodi sono normalmente un sottoinsieme dei seguenti, in funzione delle necessità:

- createUser, crea un account sul target
- deleteUser, cancella un account sul target
- getUser, resituisce la vista di un utente sul target
- getAccountIterator, restituisce un iteratore sull'accoundID degli utenti sul target
- update, aggiorna i dati di un utente sul target (compresa la password)
- enable, abilita l'utente sul target
- disable, disabilita l'utente sul target
- listProfile, effettua la lista di tutti i profili del target
- test, testa il funzionamento del servizio

Dei metodi precedenti dovranno inoltre essere definiti i parametri di input e di output in base al target da integrare (alcuni target potranno avere dei parametri differenti).
Particolare attenzione va al campo password, nel caso in cui nel target che si vuole integrare essa sia gestita.
Qualora infatti si debba gestire la password sul target, deve essere indicato al gruppo di IdM l'algoritmo di cifratura utilizzato.

2.1 Use case per la scelta dei connettori

Di seguito vengono presentati alcuni use case di esempio per la scelta del connettore da implementare.

- In caso di integrazione di target con utenti gestiti su un'unica tabella di DB, in cui non sia possibile sviluppare delle Store Procedure, esiste un connettore specifico "Database Table" che effettuerà delle operazioni di Insert, Update e Delete direttamente sul database.
- In caso di integrazione di target tramite Store Procedure, deve essere utilizzato il connettore "Scripted JDBC" opportunamente customizzato (già utilizzato per le applicazioni Vetrina Sostenibilità, FTPS, ACollab e Atti).
- In caso di integrazione di target con connettori non presenti nel documento IDM_Resource_Reference, sarà possibile svilupparne di nuovi utilizzando il Sun Resource Extension Facility Kit (fornito nella Directory /REF del prodotto) che rappresenta la guida per creare connettori custom. In questa Directory sono presenti anche codici di esempio e tool per la creazione di nuovi connettori. Tale REF Kit rappresenta una modalità di sviluppo del connettore a basso livello.

3 Accesso alle web application

Per poter accedere ad una Web Application protetta da un sistema di Web SSO, l'utente deve prima aver effettuato il Logon al sistema di autenticazione.
Effettuato il Logon Primario, l'utente può accedere alle Web Application esposte.

Ad ogni richiesta di accesso vengono ripetuti i seguenti passi:

1. La richiesta viene intercettata dal Reverse Proxy
2. L'Agent del Reverse Proxy verifica la presenza di una sessione autenticata per l'utente. Per il mantenimento della sessione viene utilizzato un cookie volatile sul client. Nel caso non esista una sessione associata alla richiesta, l'utente viene diretto verso il modulo di autenticazione del sistema di Web SSO centralizzato.
3. Il sistema di Web SSO autentica l'utente e restituisce le informazioni all'Agent che a sua volta trasmette le informazioni necessarie alla web application dell'aderente utilizzando l'header http.

3.1 Requisiti di progettazione delle interfacce

Il passaggio dei parametri tra il reverse proxy e l'applicazione avviene sfruttando i meccanismi standard del Web e cioè gli header del protocollo HTTP, quindi la Web Application deve essere in grado di gestire gli header HTTP.
Nel caso in cui l'applicazione voglia filtrare ulteriormente gli accessi può prelevare gli attributi dell'utente dall'http header ed effettuare la profilazione applicativa.

3.2 Formato parametri

I parametri passati nell'header HTTP alla web application esposta servono ad identificare ed a caratterizzare l'originatore della richiesta.
Il parametro necessariamente presente nell'header HTTP dovrà essere il seguente:

Parametro	Significato
USERNAME	Identificativo utente (identifica l'originatore della richiesta).

Potranno essere aggiunti ulteriori parametri da passare nell'http header. Questi parametri sono prelevati dagli attributi dell'utente.
I parametri elencati saranno presenti nell'header HTTP di ogni richiesta. La modalità di accesso alle

variabili dell'header sono legate al linguaggio utilizzato per la creazione e la gestione delle pagine web. Al termine del presente documento sono riportati esempi di accesso a queste variabili in JSP, PERL, ASP.NET, ASP.

3.3 Vincoli

Sono riportati di seguito i vincoli a cui le Web Application devono attenersi per poter essere protette dal sistema di Web SSO.

3.3.1 Identificazione e Autenticazione dell'utente

La web application esposta non deve richiedere il login agli utenti che accedono, in quanto il processo di identificazione e autenticazione viene già effettuato nella fase di Logon Primario che l'utente effettua sul sistema di Web SSO. In particolare, l'applicazione non deve richiedere l'immissione esplicita da parte dell'utente di un username e di una password, ma può utilizzare le informazioni di identificazione dell'utente contenute nell'header HTTP di ogni richiesta

3.3.2 Autorizzazione dell'utente

Il processo di autorizzazione all'accesso da parte dell'utente alla web application esposta è effettuato dal Reverse Proxy, che abilita o impedisce l'accesso a singole Applicazioni in base a policy prestabilite.
E' facoltà dell'applicazione di estendere il processo di autorizzazione svolto dal Reverse-Proxy, utilizzando le informazioni contenute nell'header HTTP di ogni richiesta per realizzare nuove regole di autorizzazione.

3.3.3 Uso di Cookie

Il meccanismo di Logon e le verifiche effettuate dal Reverse Proxy si basano sullo scambio di *cookie* con la Postazione di Lavoro dell'utente che necessariamente deve permettere l'utilizzo di *cookie*.

3.3.4 Convenzioni sui nomi dei domini

Una web application esposta dalla RER potrà essere accessibile, sia da Internet che dalla rete interna, mediante una URL così strutturata:

https://applicazioni.regione.emilia-romagna.it/<*percorso-applicazione*>/

dove <*percorso-applicazione*> è il *path* (al limite costituito da un singolo nome) scelto per identificare la web application (è ammesso il passaggio di parametri nella URL).

3.3.5 Convenzioni sui nomi delle web application

Qualora un servizio esponga più di una web application, gli URL corrispondenti si differenziano solo relativamente alla componente <*percorso applicazione*>.
Esempio:

https://<dominio>/<percorso applicazione -1>/

https://<dominio>/<percorso applicazione -2>/

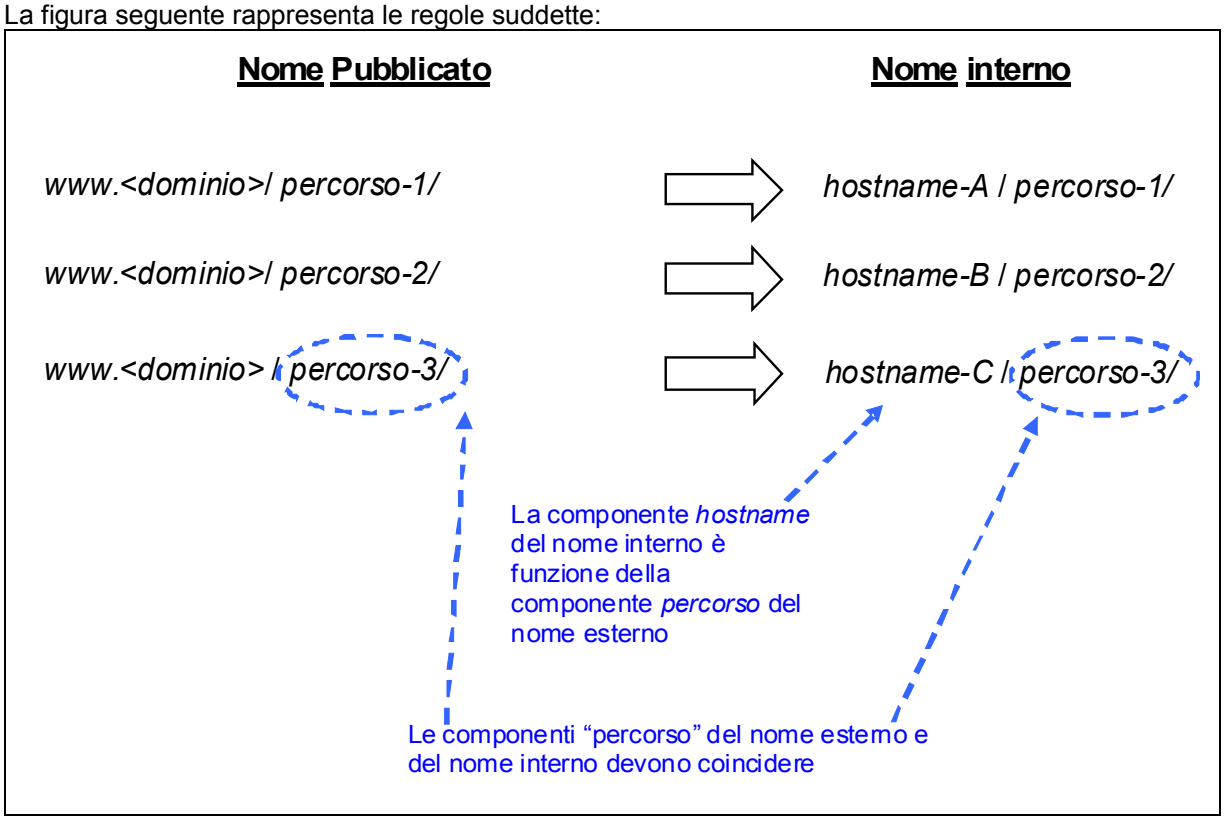
https://<dominio><percorso applicazione -3>/

3.3.6 Regole d'instradamento del Reverse Proxy

La convenzione sui nomi degli URL si riferisce all'esposizione delle web application, ovvero al modo in cui tali applicazioni sono esposte tramite Reverse Proxy e non implica che la medesima convenzione debba necessariamente essere adottata internamente all'applicativo.
Internamente all'applicativo, le web application possono risiedere su uno o più host. La corrispondenza tra il "nome esterno" e il "nome interno" dell'applicazione viene effettuata dal Reverse Proxy tramite le *regole di instradamento*. Tali regole consentono di collocare le proprie web application

sui server della rete interna, svincolandosi dall'*hostname* con cui sono visibili.
Nell'instradamento, il path dell'applicazione (cioè la porzione dell'URL che viene dopo l'*hostname*) del "nome esterno" deve coincidere con il path del "nome interno".

La figura seguente rappresenta le regole suddette:



E' ammesso il passaggio di parametri nell'URL, se previsto dalla web application, mentre non è ammesso utilizzare i parametri per identificare la web application esposta.

Esempio di utilizzo di URL non è ammesso:
https:// www.<dominio>/entrypoint?applicazione=applicazione1

3.3.7 Accessibilità dell'applicazione tramite proxy

Si descrivono di seguito i vincoli che la web application dell'Aderente deve rispettare per poter essere esposta attraverso il Reverse Proxy (requisiti di *proxability*).
La web application da esporre non deve contenere riferimenti assoluti alle proprie risorse, ma solo puntamenti relativi.
In altri termini le eventuali risorse referenziate all'interno dell'applicazione (quali., ad esempio, immagini o link ad altre pagine) devono essere indirizzate tramite **URL relativi**, ovvero URL in cui viene esplicitata solamente la componente *path* senza le componenti *protocollo* ed *hostname*. Oltre agli URL anche **i PATH devono essere relativi**, ovvero non devono iniziare con il carattere "*/*".

Ad esempio:

URL ASSOLUTI (NON UTILIZZABILI)	URL RELATIVI (DA UTILIZZARE)
http(s)://hostname/logo.gif	logo.gif
http(s)://hostname/subdir1/index.html	subdir1/index.html **NOTA :** non è possibule utilizzare URL relativi del tipo */* subdir1/index.html, i quali, pur essendo URL relativi (non vengono infatti indicati protocollo ed hostname), sono comunque PATH assoluti.

Inoltre, ogni singola Web Application deve prevedere un unico punto di ingresso da cui si diramano i diversi sottoservizi.
Non sono quindi consentiti collegamenti a sottoservizi non appartenenti all'albero che ha come radice la URL di ingresso della Web Application: ad esempio, supponendo che l'URL di "ingresso" della Web Application *sia* ***http(s)://hostnameX/directoryY***, non è consentito il collegamento a pagine che non risiedano sotto il path *directoryY* quali ***http(s)://hostnameX/directoryZ/pageJ.jsp*** (sempre che l'applicazione che ha come "ingresso" della WebApplication l'URL ***http(s)://hostnameX/directoryZ*** non sia stata esposta a sua volta).

3.3.8 Sicurezza

Ogni applicazione dovrà accettare solo le richieste pervenute dall'indirizzo IP del reverse proxy. Il controllo dovrà essere effettuato in tutti i casi a livello applicativo e, ove possibile, a livello sistemistico (con un firewall o sul web server). L'applicazione dovrà riconoscere l'utente, tramite i parametri passati nell'http Header, solo nel caso la richiesta gli venga inoltrata dal reverse proxy, in tutti gli altri casi l'applicazione non dovrà permettere l'accesso all'utente.

3.3.9 Gestione del logout applicativo

Nel caso in cui l'applicazione abbia creato una sessione al momento dell'autenticazione dell'utente, il logout dall'applicazione deve invalidare la sessione applicativa.
In ogni caso l'utente deve essere reindirizzato alla lista delle applicazioni al seguente link relativo:

- ·/index.php

3.3.10 Definizione delle regole di autorizzazione

La regola implicita di autorizzazione degli accessi a una web application esposta sul dominio RER prevede la negazione di ogni accesso. Pertanto, ogni abilitazione deve essere espressamente dichiarata al sottosistema di controllo accessi tramite la formulazione di opportune regole.
Ogni singola regola prevede la specifica di:

1. il nome della risorsa oggetto della regola di abilitazione, dove per 'risorsa' si intende una web application o una sua porzione (sottoalbero o singola pagina);
2. l'elenco dei gruppi applicativi (nell'ambito di quelli definiti dalla RER) abilitati ad accedere alla risorsa di cui al punto precedente;
3. le operazioni ammesse (get e post).

Il nome della risorsa (*Resource Name*) è espresso mediante una *regular expression* che può contenere *wildcards*.

Esempi di nomi risorsa:

https:// www.<dominio>/applicazione1/*	Tutta l'applicazione: 'applicazione1'
https://www.<dominio>/applicazione1/home.htm	La sola pagina 'home.htm' dell'applicazione 'applicazione1'
https://www.<dominio>/applicazione1/consult/*	Tutte le risorse all'interno del sottoalbero 'consult' dell'applicazione 'applicazione1'

Ogni volta che un utente accede ad una *web application* esposta sul dominio RER tramite il *Reverse Proxy*, le URL delle pagine chiamate vengono confrontate con questa *regular expression*. In caso di corrispondenza, viene consentito l'accesso solo se il gruppo dell'originatore della richiesta è stato abilitato per la risorsa identificata e se l'operazione (get o post) è ammessa.
Nel caso vi siano applicazioni con redirezione implicita su una pagina di default è necessario aggiungere una nuova regola dedicata. (ad esempio https:// www.<dominio>/applicazione2).

4 Modulo di raccolta informazioni

Concordemente con quanto espresso sopra, al fine di impostare le configurazioni del *Reverse Proxy*, le regole di autorizzazione, ecc., è necessario per il sottosistema di *Access Management* conosca un insieme di informazioni che dovranno essere fornite dai responsabili applicativi, che verranno raccolte mediante il modulo riportato di seguito.
Sulla base dei dati in esso raccolti verranno configurati il *reverse proxy* (interno ed esterno) e le regole di autorizzazione centralizzate del sistema di *Access Management*.

DATO	VALORE
Nome applicativo	
Referente tecnico	
Path relativo	
Path assoluto	
Note	

Indicazioni per la compilazione:

- **Nome applicativo:** nome dell'applicativo per esteso (esempio: Vetrina della sostenibilità)
- **Referente tecnico:** riferimenti della persona indicata come referente tecnico per l'integrazione delle web application (esempio: Marco Rossi, telefono 051-999999, mail xxx@xx.xx)
- **Path relativo:** nome breve dell'applicativo, utilizzato per formare l'URL di accesso alla web application sul portale delle applicazioni[1] (esempio: VetrinaSostenibilita)
- **Path assoluto:** URL completo della web application da mappare sull'Access Management (esempio: http://wwwservizi.regione.emilia-romagna.it/VetrinaSostenibilita).
- **Note:** eventuali note

5 Esempi di utilizzo degli attributi dell'http Header

Esempio di JSP:

```
<html>
<head><title>Recupero Username e Dominio </title></head>
<body>
<%
String userid = request.getHeader("username");
String gruppo = request.getHeader("domain");

out.println("<BR> UserID  ->  " + username);
out.println("<BR> Dominio  ->  " + domain);
```

[1] https://applicazioni.regione.emilia-romagna.it

```
%>
</body>
</html>
```

Esempio di PERL:

```
#!/usr/bin/env perl
#
# Programma di test
#

print "Content-type: text/html\n\n" ;
print "<html><head><title>Recupero Username e Dominio></title></head>\n";

print "<table border=1 bordercolor=black>";
foreach $e (%ENV)
{
	if ($e eq "HTTP_USERNAME" || $e eq "HTTP_DOMAIN")
	{
		print "<tr>";
		print "<td>$e </td><td>$ENV{$e}</td>";
		print "</tr>";
	}
}
print "</table></body></html>\n";
```

Esempio di ASP.NET:

```
string username = string.Format(@"{0}\{1}",
                    Request.Headers["Domain"],
                    Request.Headers["Username"]
                );

Response.Write(username);
```

Esempio di ASP:

```
username = Request.ServerVariables("HTTP_Domain") & "\" &
            Request.ServerVariables("HTTP_Username")

Response.Write username
```

Sistema di autenticazione centralizzata custom

Questo sezione ha lo scopo di descrivere la soluzione custom per implementare l'autenticazione nelle applicazioni ospitate sulle infrastrutture della Regione Emilia-Romagna (in seguito applicazioni *client*). In particolare sono presenti due domini "Active Directory", uno contenente gli account degli utenti regionali[2] (dominio intranet) e l'altro contenente account di utenti esterni alla regione (dominio extranet), e un sistema centralizzato di autenticazione (utilizzabile per entrambi i domini).

Ci sono due possibilità per sfruttare i servizi di questo sistema centralizzato:

- appoggiarsi per il login e la modifica della password all'applicazione web di autenticazione centralizzata disponibile all'url https://wwwservizi.regione.emilia-romagna.it/AutenticazioneCentralizzata (l'integrazione con l'applicazione *client* avviene tramite "browser redirection")
- richiamare i servizi web che implementano i metodi necessari per l'autenticazione disponibili all'url https://wwwservizi.regione.emilia-romagna.it/WebServices/AutenticazioneCentralizzata/Authentication.asmx

NB: in entrambi i casi ci si occupa solo dell'autenticazione dell'utente e non delle autorizzazioni (ad esempio profilature degli utenti, etc…) che rimangono a carico dell'applicazione *client*.
L'autenticazione consiste nella verifica dell'esistenza dell'utente nei domini "Active Directory" e nella verifica dell'appartenenza ad un gruppo di dominio definito a priori per ciascuna applicazione *client*.
Visto che le credenziali d'accesso sono gestite completamente dal sistema di autenticazione centralizzata, è necessario eliminare dal database dell'applicazione *client* ogni campo che contiene password.
Deve invece rimanere ogni campo relativo allo username, al fine di associare gli eventuali profili autorizzativi. Tali campi possono essere al massimo di 40 caratteri e devono contenere sia il dominio che il nome account separati dal backslash (ad esempio EXTRARER\sempronio.tizio per gli utenti esterni e RERSDM\Cognome_N per gli utenti regionali)

Per garantire che il servizio di autenticazione sia usato solo da chi ne ha diritto e per garantire a chi lo usa che sia proprio il servizio di autenticazione della Regione a rispondere, si sfrutta il sistema del *salted hash*. Ad ogni applicazione *client*, i gestori del sistema assegnano un identificativo univoco dell'applicazione (in seguito *IDApplicazione*) ed un codice segreto (il *salt*) noto solo all'applicazione *client* e al sistema di autenticazione. Il chiamante aggiunge al messaggio il *salted hash* (ovvero l'hash della concatenazione tra messaggio e salt). Il ricevente ricalcola il *salted hash* del messaggio e lo confronta con quello ricevuto. Se sono uguali, allora è garantita l'autenticità del mittente e l'integrità del messaggio.
L'algoritmo di hashing scelto è l'MD5.

Riassumendo, per poter sfruttare i servizi del sistema centralizzato è necessario farne richiesta al Servizio SIIR (Sistema informativo-informatico regionale) che provvederà a:

- fornire l'identificativo univoco dell'applicazione *client,*
- fornire il codice segreto dell'applicazione *client,*
- creare i gruppi di dominio (rersdm e extrarer) associati all'applicazione client, illustrando le modalità operative di gestione degli utenti[3] nell'Active Directory.

Nella richiesta è necessario indicare il referente regionale dell'applicazione e, nel caso in cui ci si appoggi per il login all'applicazione web di autenticazione centralizzata, è necessario fornire la url della pagina dell'applicazione client (in seguito pagina definita a priori) a cui l'applicazione di autenticazione centralizzata reindirizza il browser dopo aver autenticato l'utente.

[2] Si intendono "regionali" anche i collaboratori, consulenti, ecc…
[3] Almeno fino a quando non ci sarà la possibilità di crearli in modo indipendente mediante opportuna interfaccia

6 Redirect all'applicativo web generalizzato di autenticazione

Nelle applicazioni client per le quali non importa una propria pagina di autenticazione, è possibile effettuare un redirect all'applicativo web generalizzato di autenticazione disponibile all'url https://wwwservizi.regione.emilia-romagna.it/AutenticazioneCentralizzata/Default.aspx.

Si visualizza una maschera standard di login in cui si deve specificare il tipo di utente (dominio), lo username e la password (alternativamente vi è un apposito link per utenti già autenticati sul dominio, ovvero che hanno fatto logon sulla propria workstation con le credenziali regionali).

I parametri da passare, nella querystring, alla pagina Default.aspx sono:

- *IDApplicazione* (stringa) (obbligatorio)
- *dominio* (stringa) (facoltativo): se valorizzato, si imposta automaticamente nel form il corrispondente campo. Valori possibili: "RERSDM", "EXTRARER" a seconda che l'utente sia regionale o esterno. **NB:** se il dominio è vuoto l'applicativo cerca di estrapolarlo dallo username assumendo la sintassi ***nomedominio\nomeutente***.
- *username* (stringa) (facoltativo): se valorizzato, si imposta automaticamente nel form il corrispondente campo
- *customInfo* (stringa) (facoltativo): verrà restituita dall'applicazione di autenticazione centralizzata alla *pagina definita a priori* così come gli è stata inviata. E' a disposizione dell'applicazione *client* per tenere traccia di qualsiasi tipo di informazione durante la fase di autenticazione (che è appunto esterna all'applicazione *client* stessa). Ad esempio: se l'applicazione *client* vuole che a fronte dell'avvenuta autenticazione, il controllo ritorni alla pagina che era stata richiesta inizialmente dall'utente, si può mettere in c*ustomInfo* l'URL della pagina da richiamare. Naturalmente il redirect a questa pagina è a carico della *pagina definita a priori*.
- *saltedHash* (stringa) (obbligatorio)

L'applicazione di autenticazione generalizzata, dopo aver controllato l'integrità del messaggio, tenta di verificare le credenziali specificate nel form.
Se le credenziali sono scorrette, l'account è scaduto, bloccato o disabilitato, allora si visualizza un messaggio di login non riuscito.
Se le credenziali sono corrette, l'utente è autorizzato ad accedere all'applicazione, non è il primo accesso e la password non è scaduta, allora si eseguirà redirect alla *pagina definita a priori* indicandogli:

- *username* (stringa) dell'utente autenticato (nel formato ***nomedominio\nomeutente*)**.
- *validity* (stringa): data e ora massima di validità della richiesta, ovvero una data/ora oltre la quale non è più attendibile, per la pagina richiamata dall'applicativo centralizzato, l'informazione che l'utente è stato autenticato (ciò per tutelarsi di sniffing dell'url richiamato). Il formato di *validity è* AAAAMMGGhhmmss[4]
- *customInfo* (stringa)
- *saltedHash* (stringa)

La pagina dell'applicazione *client* invocata dal servizio centralizzato riceve i parametri sopra indicati. Dopo aver verificato l'integrità del messaggio e la validità della richiesta[5] potrà operare normalmente (come quando aveva in locale la pagine di richiesta delle credenziali: potrà quindi creare il proprio cookie di autenticazione o token di autenticazione, ecc…).

Se invece le credenziali sono corrette e l'utente è autorizzato ad accedere all'applicazione, ma è il primo accesso oppure la password è scaduta, allora si rimanda al form di modifica password:

[4] AAAA: anno a quattro cifre; MM, GG, hh, mm, ss: rispettivamente mese, giorno, ora, minuti secondi a 2 cifre
[5] Controllare che la data/ora attuale sia inferiore o uguale alla data di validità ricevuta (*validity*)

https://wwwservizi.regione.emilia-romagna.it/AutenticazioneCentralizzata/ModificaPassword.aspx (che è comunque sempre richiamabile, da chiunque e in qualunque momento, per cambiare la password) pre-impostando il dominio e lo username. Dopo la modifica della password si procede come nel caso precedente.

NB: nel caso si dovessero avere difficoltà a calcolare l'hash MD5 è disponibile un web service che lo fa (vedere sotto).

6.1 Modifica della password

È sempre possibile richiamare la pagina di modifica della password (da qualsiasi applicazione o sito, indipendentemente che si tratti di una delle applicazioni che sfruttano l'autenticazione centralizzata. L'url della pagina da richiamare è:

https://wwwservizi.regione.emilia-romagna.it/AutenticazioneCentralizzata/ModificaPassword.aspx

e i parametri da passare sono:

- *dominio* (stringa) (facoltativo): se valorizzato, si imposta automaticamente nel form il corrispondente campo. Valori possibili: "RERSDM", "EXTRARER" a seconda che l'utente sia regionale o esterno. **NB:** se il dominio è vuoto l'applicativo cerca di estrapolarlo dallo username assumendo la sintassi ***nomedominio\nomeutente***.
- *username* (stringa) (facoltativo): se valorizzato, si imposta automaticamente nel form il corrispondente campo
- *tornaA* (stringa) (facoltativo): se valorizzato, nella pagina di modica password viene creato un link con testo "Torna alla pagina precedente" che punta alla pagina indicata nel parametro *tornaA*

7 Chiamata dei servizi web di Autenticazione

Le applicazioni web che vogliono gestire l'autenticazione localmente (senza appoggiarsi all'applicativo centralizzato, su cui non si possono eseguire personalizzazioni di grafica o altro), possono richiamare i seguenti servizi web:

- *Authentication* che espone i metodi *Login*, *LoginIntranet*, *CambioPassword* disponibile all'url https://wwwservizi.regione.emilia-romagna.it/WebServices/AutenticazioneCentralizzata/Authentication.asmx
- *HashHelper* che espone i metodi *ComputeHash*, *ComputeSaltedHash* disponibile all'url https://wwwservizi.regione.emilia-romagna.it/WebServices/AutenticazioneCentralizzata/HashHelper.asmx

NB: le applicazioni *client* che richiamano tali servizi web devono utilizzare il protocollo *https*.

7.1 Authentication, metodo "Login"

Consente di verificare se un certo utente può accedere ad una certa applicazione.
Riceve in input i parametri:

- *IDApplicazione* (stringa)
- *Dominio* (stringa), valori possibili: "RERSDM", "EXTRARER" a seconda che l'utente sia regionale o esterno
- *Username* (stringa)
- *Password* (stringa)
- *SaltedHash* (stringa)

Restituisce in output uno dei seguenti valori:
- NonRiuscito
- Riuscito
- LoginFallito
- PasswordScaduta
- AccountDisabilitato
- PasswordDaCambiare
- AccountBloccato
- AccountScaduto

7.2 Authentication, metodo "LoginIntranet"

Consente di verificare se un utente, autenticato sul dominio, può accedere ad una certa applicazione.
Riceve in input i parametri:
- *IDApplicazione* (stringa)
- *Dominio* (stringa), valori possibili: "RERSDM", "EXTRARER" a seconda del dominio su cui l'utente si è loggato
- *Username* (stringa)
- *SaltedHash* (stringa)

Restituisce in output uno dei seguenti valori:
- NonRiuscito
- Riuscito
- LoginFallito
- PasswordScaduta
- AccountDisabilitato
- PasswordDaCambiare
- AccountBloccato
- AccountScaduto

7.3 Authentication, metodo "CambioPassword"

Consente di modificare la password di un utente.
Riceve in input i parametri:
- *IDApplicazione* (stringa)
- *Dominio* (stringa), valori possibili: "RERSDM", "EXTRARER" a seconda che l'utente sia regionale o esterno
- *Username* (stringa)
- *OldPassword* (stringa)
- *NewPassword* (stringa)
- *SaltedHash* (stringa)

Restituisce in output uno dei seguenti valori:
- NonRiuscito
- Riuscito
- VecchiaPasswordSbagliata
- PasswordNonSoddisfaIRequisiti
- AccountRestriction
- PasswordNonModificabile
- AccountBloccato
- AccountScaduto

7.4 HashHelper, metodo ComputeHash

Consente di calcolare l'hash in formato MD5 di una stringa ricevuta in input.

Riceve in input il parametro
- *value* (stringa)

Restituisce in output l'hash di *value.*

7.5 HashHelper, metodo ComputeSaltedHash

Consente di calcolare l'hash di una sequenza di parametri salati con un codice privato.

Riceve in input il parametro
- *salt* (stringa), il codice privato
- *parametri (*stringa), messaggio da "salare"

Restituisce in output il *salted hash*, ovvero l'hash della stringa ottenuta concatenando i vari parametri e mettendo in coda il *salt.*

8 Libreria di utilità per applicazioni ASP

In caso si sviluppi in linguaggio ASP è possibile sfruttare una "libreria" di funzioni, definite in una pagina ASP, che rendono molto semplici l'utilizzo dei servizi dell'infrastruttura dell'autenticazione centralizzata.
Tali funzioni utilizzano implicitamente i valori delle variabili *RCRER_IDApplicazione* e *RCRER_codicePrivato* che vanno pertanto impostate (ma non dichiarate, sono dichiarate nella libreria) nella specifica applicazione *client* e sono rispettivamente l'identificativo univoco e il codice segreto dell'applicazione *client*.

La pagina ASP *AutenticazioneCentralizzata.asp* (che si trova nella directory virutale "/includes" dei server web presenti in Regione) va inclusa nelle applicazioni che intendono utilizzare le funzioni in essa definite. Le funzioni disponibili sono elencate qui sotto.

8.1 Funzioni utili in caso di redirect all'Autenticazione Centralizzata

VaiALoginCentralizzatoConCustomInfo (customInfo) / VaiALoginCentralizzato

Esegue il redirect all'url restituito da HRefLoginCentralizzato(customInfo).
Vedi "HRefLoginCentralizzato (customInfo)"

Utilizzare una o l'altra a seconda che sia necessario o meno tenere traccia di informazioni.

VaiALoginCentralizzatoConCustomInfo (customInfo)

Esegue il redirect all'url restituito da HRefLoginCentralizzato(customInfo).
vedi "HRefLoginCentralizzato (customInfo)"

HRefLoginCentralizzato (customInfo)

Restituisce l'url su cui ridirezionare l'applicazione client per il login.
Tale url è comprensivo dei parametri che si aspetta il sistema di autenticazione centralizzata, ovvero
- *IDApplicazione*
- *customInfo*
- *saltedHash*

Vedi “Redirect all’applicativo web generalizzato di autenticazione”

MessaggioUtenteNonAutorizzatoConCustomInfo (customInfo) / MessaggioUtenteNonAutorizzato

Restituisce il messaggio da visualizzare nel caso in cui l’utente sia stato autenticato ma non disponga della autorizzazione necessaria per accedere (ad esempio se nel database non è presente alcun utente *username*). La stringa restituita è comprensiva anche del link al *Login*.

Utilizzare una o l’altra a seconda che sia necessario o meno tenere traccia di alcune informazioni.

IsSalaturaOK (byref username)

Si recuperano i parametri che l’applicazione centralizzata ha passato all’applicazione *client*. Si ricalcola il *salted hash* in base ai parametri ricevuti e lo si confronta con quello ricevuto. Se sono uguali, la funzione restituisce true altrimenti si visualizza “Messaggio Corrotto” e la funzione restituisce false.

IsLatenzaOK

Si controlla che la data attuale non sia superiore a *validity*. La funzione restituisce true o false a seconda che la richiesta sia ancora valida o meno. In particolare se la richiesta non è valida, allora si ridireziona il browser alla pagina di login.

IsAccessoOK (byref username)

Restituisce true se la salatura è corretta (*IsSalaturaOK*) e la richiesta è ancora valida (*IsLatenzaOK*). Se una delle due condizioni non è verificata restituisce false.

HashHelper_ComputeHash (value)

Restituisce l’hash di *value*.

HashHelper_ComputeSaltedHash(salt, parametri)

Restituisce il saltedhash della stringa *parametri* salato con il codice privato *salt*

8.2 Funzioni utili in caso di richiamo dei servizi Web

Authentication_Login (domain, username, password)

Calcola il salted hash dei parametri ricevuti (*HashHelper_ComputeSaltedHash*) e richiama il metodo *Login* del servizio web di *Authentication*. Restituisce l’esito del login (vedi “Authentication, metodo “Login””).

Authentication_LoginIntranet (domain, username)

Calcola il salted hash dei parametri ricevuti (*HashHelper_ComputeSaltedHash*) e richiama il metodo *LoginIntranet* del servizio web di *Authentication*. Restituisce l’esito del login (vedi “Authentication, metodo “LoginIntranet””)

IsLoginRiuscito (esitoLogin)

Restituisce true o false se *esitoLogin* è uguale a Riuscito

IsLoginPasswordScaduta (esitoLogin)

Restituisce true o false se *esitoLogin* è uguale a PasswordScaduta

IsLoginPasswordDaCambiare (esitoLogin)

Restituisce true o false se *esitoLogi*n è uguale a PasswordDaCambiare

Authentication_CambioPassword (domain, username, oldPassword, newPassword)

Richiama il metodo *CambioPassword* del servizio web di *Authentication* e restituisce l'esito del cambio password (vedi "Authentication, metodo "CambioPassword"").

IsCambioPasswordRiuscito (esitoCambioPassword)

Restituisce true o false se *esitoCambioPassword* è uguale a Riuscito

IsCambioPasswordVecchiaPasswordSbagliata (esitoCambioPassword)

Restituisce true o false se *esitoCambioPassword* è uguale a VecchiaPasswordSbagliata.

IsCambioPasswordPasswordNonSoddisfaRequisiti (esitoCambioPassword)

Restituisce true o false se *esitoCambioPassword* è uguale a PasswordNonSoddisfaRequisiti.

8.3 Esempio di utilizzo delle funzioni in caso di Redirect all'autenticazione centralizzata

Supponiamo che lo scenario dell'applicazione *client* sia il seguente:

- *Default.ASP*: è la prima pagina caricata (per intenderci quella che normalmente conterrebbe il form di login)
- *Login.ASP*: è la pagina che viene richiamata dall'applicazione di autenticazione centralizzata (per intenderci quella che sarebbe l'action del form di login della *default*.ASP)

Entrambe includono la pagina *Global.ASP* che contiene le funzioni di utilità comuni all'applicazione.

Per utilizzare le funzioni descritte nel paragrafo "Funzioni utili in caso di redirect all'Autenticazione Centralizzata", occorre procedere, per ciascuna pagina, nel seguente modo.

Global.ASP

Includere il file /includes/AutenticazioneCentralizzata.asp e valorizzare le variabili *RCRER_IDApplicazione* e *RCRER_codicePrivato*.
In pratica inserire le seguenti istruzioni:

```
<!-- include virtual="/includes/AutenticazioneCentralizzata.ASP" -- >
<%
RCRER_IDApplicazione = "ApplicazioneX"
RCRER_codicePrivato = "swu55kdjfk16grte"
%>
```

Default.asp

Ridirezionare il browser all'applicazione centralizzata richiamando *VaiALoginCentralizzato*.
In pratica le istruzioni della pagina diventerebbero:

```
<!- - include file="global.ASP" - - >
<%
VaiALoginCentralizzato
%>
```

In questo caso particolare il browser richiamerebbe il seguente url:

http://.....?IDApplicazione=ApplicazioneX&customInfo=&saltedHash=jkquF+X6+c29hoJC04lGDw==[6]

NB: nel caso in cui si voglia tenere traccia di alcune informazioni, utilizzare VaiALoginCentralizzatoConCustomInfo passando le informazioni necessarie. Tali informazioni saranno restituite, così come sono state passate, alla pagina Login.ASP (vedi sotto).
Ad esempio, volendo passare il nome di una pagina asp a cui la Login.ASP dovrebbe ridirigersi, si avrebbe:

```
<!- - include file="global.ASP" - - >
<%
VaiALoginCentralizzato "torna_a=menu.asp"
%>
```

e in questo caso particolare il browser richiamerebbe il seguente url:

http://.....?
IDApplicazione=ApplicazioneX&customInfo=torna_a=menu.asp&saltedHash=lFr1E+65zr6uneh7hhXCPg==[7]

Login.asp

Definire la variabile *username* se già non è stata definita nel global.ASP
Verificare che la salatura e che il periodo di validità siano corretti richiamando *IsAccessoOK* passando *username* che è passato per riferimento e impostato dalla chiamata stessa.

In pratica le istruzioni della pagina diventerebbero:

```
<!- - include file="global.ASP" - - >
<%
Dim username
If IsAccessoOK (username) Then
  'qui ci va quello che si farebbe normalmente:
    'creazione del cookie di autenticazione,
      'impostazione delle variabili di sessione,
        'individuazione del profilo, redirect al menu, etc….

          'In particolare se nel database non è presente alcun
  'utente username, è possibile richiamare
    'MessaggioUtenteNonAutorizzato
End If
%>
```

N.B. In questo costrutto *If – Then* non è presente l'*Else* perchè è la funzione *IsAccessoOK* che si occupa di visualizzare un messaggio o ridirezionare alla pagina di login se il messaggio era corrotto o se la richiesta non era più valida.

8.4 Esempio di utilizzo delle funzioni in caso di richiamo dei servizi web

Supponiamo che lo scenario dell'applicazione *client* sia il seguente:

- *Default.ASP*: è la prima pagina caricata, quella che contiene il form di login
- *Login.ASP*: è l'action di *Default.ASP*

[6] È il salted hash dei parametri passati nel querystring (in questo caso della stringa "*ApplicazioneX*" - unico parametro non vuoto) salati con il codice privato dell'applicazione
[7] È il salted hash dei parametri passati nel querystring (in questo caso della stringa "*ApplicazioneXtorna_a=menu.asp*") salati con il codice privato dell'applicazione

Entrambe includono la pagina *Global.ASP* che contiene le funzioni di utilità comuni all'applicazione.

Per utilizzare le funzioni descritte nel paragrafo Funzioni utili in caso di richiamo dei servizi Web, occorre procedere, per ciascuna pagina, nel seguente modo.

Global.asp

Includere il file e valorizzare le variabili *RCRER_IDApplicazione* e *RCRER_codicePrivato*.
In pratica inserire la seguente inclusione di file:

```
<!- - include virtual="/includes/AutenticazioneCentralizzata.ASP" - - >
<%
RCRER_IDApplicazione = "ApplicazioneX"
RCRER_codicePrivato = "swu55kdjfk16grte"
%>
```

Default.ASP

Rimane inalterata.

Login.ASP

Richiamare la funzione *authentication_Login* passando quanto impostato nel form di login.

Se il login è riuscito (*IsLoginRiuscito* è true) e la password è da cambiare (*IsLoginPasswordScaduta* e *IsLoginPasswordDaCambiare* sono true), allora richiamare la pagina per modificare la password.

Diversamente procedere normalmente a seconda che il login sia riuscito (senza necessità di cambiare la password) oppure non riuscito.

In pratica le istruzioni diventerebbero:

```
<!- - include file="global.ASP" - - >
<%
Dim Dominio, Username, Password
Dim esitoLogin
Dominio  = Trim(Request.Form("Dominio"))
Username = Trim(Request.Form("Username"))
Password = Trim(Request.Form("Password"))

esitoLogin = authentication_Login (dominio, username, password)
If IsLoginRiuscito (esitoLogin) Then
     If IsLoginPasswordScaduta(esitoLogin) or
     IsLoginPasswordDaCambiare(esitoLogin) Then
       'qui ci va il redirect alla pagina per modificare la password

   Else
       'qui ci va quello che si farebbe normalmente:
     'creazione del cookie di autenticazione,
     'impostazione delle variabili di sessione,
     'individuazione del profilo, redirect al menu, etc….
End If
Else
  'qui si deve visualizzare il messaggio di login non riuscito
End If
%>
```

9 Classe di utilità per applicazioni ASP.NET

Analogamente a quanto fatto per le pagine ASP è disponibile una classe che rende molto semplici l'utilizzo dei servizi dell'infrastruttura dell'autenticazione centralizzata nel caso di applicazioni web ASP.NET che utilizzano la Forms Authentication. Per quanto riguarda i webservices non è stato definito niente in quanto con .NET già sufficiente quello che fornisce il framework per il "consumo" di servizi web.

La classe è definita nell'assembly RER.Tools il suo nome è:

RER.Tools.AutenticazioneCentralizzataHelper

Tale classe ha due costruttori a cui passare l'IDApplicazione e il codice privato (che rappresentano rispettivamente l'identificativo univoco e il codice segreto dell'applicazione *client*). I due costruttori differiscono solo per il parametro customInfo che può essere specificato o meno.

Questi tre attributi: IDApplicazione, CodicePrivato e CustomInfo sono anche accessibili tramite omonime proprietà pubbliche dell'istanza della classe.

Sono poi definiti i seguenti metodi (da usare per invocare l'applicazione centralizzata):

- **VaiALogin(), VaiALogin(string customInfo)**: che esegue il redirect all'applicazione centralizzata di login.
- **UrlLogin(), UrlLogin(string customInfo)**: che restuisce l'url su cui ridirezionare l'applicazione client per il login.

Sono infine definite queste altre tre proprietà (da usare quando si deve interpretare l'esito del login, nella *pagina definita a priori*, per intendersi):

- **UtenteAutenticato**: invoca le due proprietà sottostanti e se tutto ok, restituisce l'utente autenticato. Se invece la "salatura" non è corretta lancia l'eccezione *RER.Tools.AutenticazioneCentralizzataHelper.MessaggioCorrottoException,* se invece è la "latenza" a non essere accettabile, esegue automaticamente un redirect alla pagina di login.
- **IsSalaturaOk**: si recuperano i parametri che l'applicazione centralizzata ha passato all'applicazione client. Si ricalcola il salted hash in base ai parametri ricevuti e lo si confronta con quello ricevuto. Se sono uguali, la funzione restituisce true altrimenti restituisce false.
- **IsLatenzaOk**: controlla che la data attuale non sia superiore a *validity*. La funzione restituisce true o false a seconda che la richiesta sia ancora valida o meno.

Per comodità la classe espone anche una proprietà *static* che contiene l'url per la pagina di modifica password.

9.1 Esempio di utilizzo

Definire nella classe Global (del global.asax) le due costanti IDApplicazione e CodicePrivato:

```
public class Global : System.Web.HttpApplication
{

        public const string IDApplicazione = "yyyyyyyyyyyyyyy";
        public const string CodicePrivato = "xxxxxxxxxxxxxx";

        // ...
}
```

Nella pagina di Login (quella indicata nella configurazione della Forms Authentication) adattare il seguente codice per l'evento Page_Load nel caso IsPostBack sia falso (il caso in cui è true non si verifica mai, in quanto la pagina di login non ha una interfaccia propria):

```
private void Page_Load(object sender, System.EventArgs e)
{
   if (!IsPostBack)
   {
      AutenticazioneCentralizzataHelper autenticazioneCentralizzataHelper =
         new AutenticazioneCentralizzataHelper(
            Global.IDApplicazione, Global.CodicePrivato);

     if (Request.QueryString["Username"] == null)
         autenticazioneCentralizzataHelper.VaiALogin();
     else
     {
          string username = autenticazioneCentralizzataHelper.UtenteAutenticato;

          // qui mettere il codice per la creazione dell cookie di
          // autenticazione e eventuale gestione dell'autorizzazione
     }
   }
}
```

Allegato 10: Specifiche tecniche per l'utilizzo dei web services del Protocollo informatico

1 Introduzione

Questo documento ha lo scopo di descrivere i WS e relativi metodi messi a disposizione dalla piattaforma e-Grammata di protocollo per permettere l'integrazione SOA con gli applicativi eterogenei regionali. I servizi sono disponibili sulla intranet regionale all'indirizzo:
https://protocollo.ente.regione.emr.it/axisProduzione

Dall'indirizzo:
https://protocollo.ente.regione.emr.it/axisProduzione/servlet/AxisServlet
si possono vedere i rispettivi wsdl.

L'ambiente di collaudo analogamente espone i servizi all'indirizzo:
https://protocollocollaudo.ente.regione.emr.it/axisCollaudo

Sarà cura dell'applicazione chiamante accedere a tali servizi.

2 Web services e-Grammata

Si noti che attualmente i servizi web e-Grammata non includono implicitamente la chiamata ai servizi di autenticazione centralizzata RER, integrazione che verrà implementata quanto prima.

I servizi elencati nel presente documento sono:

1. Protocollazione: consente di effettuare una protocollazione in entrata, in uscita e tra uffici.
2. Protocollazioni allegati: consente di effettuare una protocollazione in entrata, in uscita e tra uffici con allegati.
3. Ricerca documento: consente di effettuare una ricerca per reperire le informazioni di uno o più documenti.
4. Ricerca anagrafica esterna: consente di effettuare una ricerca per reperire le informazioni di una o più anagrafiche esterne a partire dal codice fiscale oppure dal cognome e nome/ ragione sociale.
5. Acquisizione da sportello: consente di effettuare una protocollazione (acquisizione) da sportello.
6. Ricerca fascicoli: consente di effettuare una ricerca dei fascicoli corrispondenti a criteri di ricerca.
7. Ricerca Unità organizzativa (uo) di una postazione: permette di effettuare la ricerca delle informazioni associate ad una postazione conoscendone l'identificativo; viene inoltre restituita anche la postazione "padre".
8. Modifica protocollazione: permette di modificare o annullare i campi di una registrazione; se la registrazione non conteneva allegati inoltre permette di inserire in una soluzione "one shot" uno o più allegati. L'ordine di inserimento rispecchierà l'ordine con cui vengono registrati.
9. Estrai allegati: data una segnatura di registrazione o un identificativo documento permette di estrarre su file system tutti i suoi files allegati.
10. Crea Copia: data una segnatura o un identificativo documento e una unità organizzativa destinataria, permette di creare una copia della registrazione a essa assegnata.

Tutti i Web Services hanno i seguenti dati in input, tutti obbligatori:

1.	Codice ente che richiama il WS, si tratta del codice numerico che identifica la AOO
2.	Username con cui si autentica al sistema di Protocollo l'utente applicativo per cui si invoca il WS
3.	Password con cui si autentica al sistema di Protocollo l'utente applicativo per cui si invoca il WS
4.	File XML con i dati specifici da passare al sistema di Protocollo

Tabella 2.1

Alcuni Web Services hanno come output un file XML (il cui tracciato è quello riportato di volta in volta nella documentazione che già avete) che contiene l'esito del processo e l'eventuale errore. Altre eventuali informazioni di output da restituire al chiamante vengono riportate in un XML che è il primo attachment del messaggio SOAP.

La stringa XMLInput sarà conforme al servizio richiesto e allo specifico DTD corrispondente (vedi tabella).

Nome servizio	**DTD di richiesta**	**DTD di risposta**
WSProtocollazione WSProtocollazioneAllegati	SegnaturaProt.dtd	Risposta.dtd
WSRicercaProtocollo	RicercaProtocollo.dtd	RisultatoRicerca.dtd
WSRicercaAnagrafica	RicercaAnagrafica.dtd	RisultatoRicercaAnagrafica.dtd
WSProtoSportello	SegnaturaAcqSportello.dtd	Risposta.dtd
WSRicercaFascicolo	RicercaFasc.dtd	RisultatoRicercaFasc.dtd
WSRicercaUO	RicercaUO.dtd	RispostaRicercaUO.dtd
WSModificaProtocollazione Allegati	ModificaDoc.xsd	
WSEstraiDocElettronici	EstraiFileElettronici.xsd	RisultatoRicerca.dtd
WSCreaCopia	CreaCopia.xsd	

Tabella 2.2

3 Descrizione di dettaglio dei servizi

3.1 Servizio di protocollazione

WSProtocollazione
WSProtocollazioneAllegati
Il servizio consente di effettuare una protocollazione in entrata, in uscita e tra uffici.
Per quanto riguarda i MITTENTI / DESTINATARI INTERNI è possibile specificare:
l'identificativo dell'unità operativa (ID_UO)
oppure
la 5-upla (Settore – Servizio – UOC – UOS – Postazione).

Per quanto riguarda i MITTENTI / DESTINATARI ESTERNI è possibile indicare:
una persona fisica / giuridica già esistente in archivio
oppure
una persona fisica / giuridica da inserire per la prima volta in archivo.

Nel primo caso è necessario valorizzare l'identificativo anagrafico (ID_ANAG), reperito tramite il WS di Ricerca Anagrafica;
nel secondo caso bisogna specificare il Nome Cognome o Ragione Sociale, l'eventuale matricola e l'eventuale codice fiscale o partita iva della nuova persona fisica / giuridica da inserire.

Le informazioni che vengono gestite dal servizio, descritte nel file SegnaturaProt.dtd sono:

Attributo o elemento	***Descrizione***	***Prot entrata***	***Prot uscita***	***Obbligatorio***
IdUteIn	Identificativo dell'utente collegato	✓	✓	✓
IdUOIn	Identificativo dell'unità operativa collegata	✓	✓	✓
FlgProtGen	Se valorizzato con 'S' specifica se si richiede una	✓	✓	

Attributo o elemento	***Descrizione***	***Prot entrata***	***Prot uscita***	***Obbligatorio***
	protocollazione generale, se 'N' esclude la registrazione di protocollo generale			
TipoRep	Tipo di repertorio	✓	✓	
DtArrivoIn	Data estesa di arrivo del documento	✓		✓
FlgComplIn	Flag di completezza della documentazione	✓	✓	
FlgRsvIn	Flag di riservatezza del documento	✓	✓	
FlgEvdIn	Flag di evidenza del documento	✓	✓	
TipoFisicoIn	Identificativo del tipo fisico	✓	✓	
TipoLogicoIn	Identificativo del tipo logico	✓	✓	
SttpLogicoIn	Identificativo del sottotipo logico	✓	✓	
TxtOggIn	Testo oggetto del documento	✓	✓	✓
NoteIn	Note da allegare alla protocollazione	✓	✓	
RifProvIn	Riferimento della provenienza	✓[8]		
ProtProvIn	Protocollo della provenienza	✓[1]		
DtProvIn	Data della provenienza	✓[1]		
IdIndice	Identificativo univoco dell'indice dell'originale	✓	✓	✓
IdTitolazione	Identificativo univoco della titolazione dell'originale	✓	✓	✓
IdFascicolo	Identificativo univoco del fascicolo dell'originale	✓	✓	✓[9]
NumFasc	Progressivo del fascicolo dell'originale	✓	✓	✓[10]
NumSottofasc	Progressivo del sottofascicolo dell'originale	✓	✓	
AnnoFasc	Anno del fascicolo dell'originale	✓	✓	✓[11]
FlgNoPubblIn	Flag di riservatezza al pubblico	✓	✓	
DtTermNoPubblIn	Data di termine della riservatezza al pubblico	✓	✓	
NrimandaOrigIn	Indica se l'originale di un documento in uscita deve essere inviato all' uo mittente		✓	
EsibDest	Elenco dei **DESTINATARI esterni** con le seguenti informazioni: flgDestCopia (S | N) "N" flag originale/copia flgTpAnag (P | D) "P" flag persona/ditta flgCC (S | N) "N" Copie per conoscenza IdAnag – codice anagrafico Matricola – matricola del cittadino Cognome, Nome, RagioneSociale – desc.anagrafica CodFis – codice fiscale ParIva – partita I.V.A. IndirizziEsibDest …		✓	✓[12]
IndirizziEsibDest	Indirizzo - Indirizzo del singolo destinatario esterno		✓	
Indirizzo	Indirizzo del destinatario con le seguenti		✓	

[8] Se indicato in uscita il servizio restituirà un errore.
[9] Obbligatorio solo in uscita se non è stato indicato NumFasc e AnnoFasc
[10] Obbligatorio solo in uscita se non è stato indicato IdFascicolo
[11] Obbligatorio solo in uscita se non è stato indicato IdFascicolo

[12] Se l'Anagrafico Destinatario è esistente, deve essere valorizzato il relativo identificativo (ID_ANAG); se l'Anagrafico è insesistente, è necessario indicare il Cognome Nome – Ragione Sociale, l'eventuale matricola, l'eventuale codice fiscale – partita iva e non valorizzare l'identificativo (ID_ANAG).

Attributo o elemento	***Descrizione***	***Prot entrata***	***Prot uscita***	***Obbligatorio***
	informazioni: IdTopon – identificativo della toponomastica DesInd – descrizione dell'indirizzo NumCiv – numero civico EspCiv – esponente civico Comune – Descrizione del Comune CAP – C.A.P. Provincia – Codice Provincia Nazione – Codice Stato			
Firm	Elenco dei **MITTENTI esterni** con le seguenti informazioni: flgTpAnag (P | D) "P" IdAnag – codice anagrafico Matricola – matricola del cittadino Cognome, Nome, RagioneSociale – desc.anagrafica CodFis – codice fiscale ParIva – partita I.V.A. IndirizziFirm…	✓		✓[13]
IndirizziFirm	Indirizzo - Indirizzo del singolo destinatario esterno	✓		
Indirizzo	Indirizzo del mittente esterno con le seguenti informazioni: Vedi indirizzi destinatari interni…	✓		
UOProv. IdUO IO	Identificativo dell'unità operativa di provenienza		✓	✓[14]
SettProvIn	Settore della U. O. di provenienza		✓	
ServProvIn	Servizio della U. O. di provenienza		✓	
UOCProvIn	U. O. C. della U. O. di provenienza		✓	
UOSProvIn	U. O. S. della U. O. di provenienza		✓	
PostProvIn	Postazione della U. O. di provenienza		✓	
CopieArrIn	Elenco delle copie con le seguenti informazioni: attributi: flgorig (S | N) "N" #REQUIRED (Obbligatorio) note CDATA #IMPLIED flgCC (S | N) "N" #REQUIRED Copie per conoscenza	✓	✓[15]	✓[16]

[13] Se l'Anagrafico Mittente è esistente, deve essere valorizzato il relativo identificativo (ID_ANAG); se l'Anagrafico è insesistente, è necessario indicare il Cognome Nome – Ragione Sociale, l'eventuale matricola, l'eventuale codice fiscale – partita iva e non valorizzare l'identificativo (ID_ANAG).

[14] È obbligatorio identificare l'unità operativa di provenienza, il servizio accetta sia il suo identificativo (ID_UO) che la cinquina (settore–servizio–U. O. C.–U. O. S.–Postazione), questo solo per la protocollazione in uscita, se viene indicato in entrata il servizio restituirà un errore.

[15] Se il documento è in uscita per la copia contrassegnata con "Originale" non dovrà essere indicato l'assegnatario. Per le altre copie invece è obbligatorio.

[16] In entrata per l'"Originale" è obbligatorio indicare l'assegnatario.

Attributo o elemento	***Descrizione***	***Prot entrata***	***Prot uscita***	***Obbligatorio***
	Uoass –U. O. assegnataria IdInd – identificativo dell'indice (Obbligatorio) AnnoFasc – anno del fascicolo ProgrFasc – progressivo del fascicolo Numsottofasc – numero del sottofascicolo di appartenenza			
UoAss	Elenco delle copie con le seguenti informazioni: attributi: flgorig (S | N) "N" #REQUIRED (Obbligatorio) note CDATA #IMPLIED flgCC (S | N) "N" #REQUIRED Copie per conoscenza IdUo – identificativo della U. O. assegnataria SettIn – settore della U. O. assegnataria ServIn – servizio della U. O. assegnataria UOCIn – U. O. C. della U. O. assegnataria UOSIn – U. O. S. della U. O. assegnataria PostIn – postazione della U. O. assegnataria	✓	✓[17]	✓[18]
TipoProtIN	Tipo protocollazione: E/U	✓	✓	✓
AllegaArrIn	Elenco degli allegati con le seguenti informazioni: Tipo CDATA #REQUIRED Note CDATA #IMPLIED NumeroAttachment CDATA #IMPLIED NomeFile CDATA #IMPLIED	✓	✓	
Documento Elettronico	Documento Primario con le seguenti informazioni: NumeroAttachment CDATA #REQUIRED NomeFile CDATA #REQUIRED	✓	✓	

NB. Il formato dei campi Data è GG/MM/AAAA.

Risposta.

La risposta del WS viene restituita in formato XML secondo il tracciato definito nel file Risposta.dtd

Attributo o elemento	***Descrizione***
Stato	(Codice, Messaggio)
Codice	Codice di ritorno del WS

[17] Se il documento è in uscita per la copia contrassegnata con "Originale" non dovrà essere indicato l'assegnatario. Per le altre copie invece è obbligatorio.
[18] In entrata per l'"Originale" è obbligatorio indicare l'assegnatario.

Attributo o elemento	***Descrizione***
Messaggio	Messaggio associato al codice di ritorno
Avvertimento	Avvertimento nel caso in cui ci siano errori nel salvataggio degli allegati
NumeroProtocollo	Tipo = tipo di protocollo Anno = anno di protocollazione Numero = Numero di protocollo
NuoveAnagrafiche	Elenco di una o più elementi di tipo Anagrafica
Anagrafica	Attributi codice - id_anag dell'anagrafica inserita matricola - matricola del cittadino tipo - (P | D) "P" Nominativo – desc.anagrafica CodiceFiscale – codice fiscale PartitaIva – partita I.V.A. DescrizioneResidenza – descrizione Residenza DataFineValidita – data fine validità

3.2 Servizio di ricerca protocollo

Il servizio consente di effettuare una ricerca sulla banca dati per reperire le informazioni di uno o più documenti.

Le informazioni che vengono gestite dal servizio, descritte nel file RicercaProtocollo.dtd sono:

Attributo o Elemento	***Descrizione***
IdPostazLavoro	ID_UO della postazione di lavoro che richiede la ricerca
RegPrimaria.Sigla	Sigla della registrazione del documento
RegPrimaria.Anno	Anno della registrazione
RegPrimaria.Nro	Numero della registrazione
NumCopia	Numero della copia del documento
FlgTpReg	Flag indicante il tipo di registrazione: E in Entrata, U in Uscita, I tra Uffici, D Differita, T qualsiasi
DtRegDa	Limite inferiore dell'intervallo di data di protocollazione
DtRegA	Limite superiore dell'intervallo di data di protocollazione
RegEmerg.Sigla	Sigla del Registro di Emergenza
RegEmerg.Anno	Anno del Registro di Emergenza
RegEmerg.Numero	Numero del Registro di Emergenza
DtArrivoDa	Limite inferiore dell'intervallo di data di arrivo dei documenti
DtArrivoA	Limite superiore dell'intervallo di data di arrivo dei documenti
UoReg.IdUo	Id della Uo che ha eseguito la registrazione
UoReg.Liv1	Settore della Uo che ha eseguito la registrazione
UoReg.Liv2	Servizio della Uo che ha eseguito la registrazione
UoReg.Liv3	Uoc della Uo che ha eseguito la registrazione
UoReg.Liv4	Uos della Uo che ha eseguito la registrazione
UoReg.Postazione	Postazione della Uo che ha eseguito la registrazione
UoReg.FlgInclSottoUO	Indicazione di includere le sottouo della Uo che ha eseguito la reg. (valori 1/0): se 1 il filtro per U.O. di reg. è soddisfatto anche se la registrazione dei doc. da estrarre è stata effettuata da postazioni delle sotto-UO della UO immessa nel filtro per UO di registrazione

Attributo o Elemento	***Descrizione***
SiglaAttoAutProtDifferita	Sigla dell'atto di protocollazione differita (ES. PG)
AnnoAttoAutProtDifferita	Anno dell'atto di protocollazione differita
NumAttoAutProtDifferita	Numero dell'atto di protocollazione differita
NominativoEsterno.FlgTpNomEst	Indica il tipo di nominativo esterno: E=Esibente(=Mittente), D=Destinatario, F=Firmatario, C=Contraente, T Qualsiasi (da valorizzare solo se la denominazione del nominativo è valorizzata a sua volta)
NominativoEsterno.IdAnag	Identificativo anagrafico di un nominativo (mitt./dest. ecc) esterno
NominativoEsterno.Denominazione	Denominazione di un nominativo (mitt./dest. ecc) esterno (ricerca esatta, non possibile quella per stringa contenuta)
DtProv	Data di protocollazione del documento ricevuto
RifProv	Riferimenti del documento ricevuto
ProtProv	Protocollo del documento ricevuto
UoProv.IdUo	Id. della U.O. mittente
UoProv.Liv1	Settore della U.O. mittente
UoProv.Liv2	Servizio della U.O. mittente
UoProv.Liv3	UOC della U.O. mittente
UoProv.Liv4	UOS della U.O. mittente
UoProv.Postazione	Postazione della U.O. mittente
UoProv.FlgInclSottouo	Indicazione di includere le sottouo della Uo mittente (valori 1/0): se 1 il filtro per U.O. mittente è soddisfatto anche se i mittenti dei doc. da estrarre sono delle sotto-UO della UO mittente
IdTpFis	id. del mezzo di trasmissione
DtRaccomandataDa	Limite inferiore dell'intervallo di ricerca sulla data di raccomandata
DtRaccomandataA	Limite superiore dell'intervallo di ricerca sulla data di raccomandata
IdTpLog	Id. del tipo del documento
IdSttpLog	Id. del sottotipo del documento
IdTpAlleg	Id. del tipo di un allegato del documento
DesAlleg	Descrizione di un allegato del documento
DocPrec.Sigla	Sigla di registrazione del documento precedente
DocPrec.Anno	Anno di registrazione del documento precedente
DocPrec.Nro	Numero di registrazione del documento precedente
Oggetto	Oggetto del documento (o parte di esso)
UoPresso.IdUo	Id. della U.O. presso cui o presso le cui postazioni devono trovarsi i documenti da ricercare
UoPresso.Liv1	Settore della U.O. presso cui o presso le cui postazioni devono trovarsi i documenti da ricercare
UoPresso.Liv2	Servizio della U.O. presso cui o presso le cui postazioni devono trovarsi i documenti da ricercare
UoPresso.Liv3	UOC della U.O. presso cui o presso le cui postazioni devono trovarsi i documenti da ricercare
UoPresso.Liv4	UOS della U.O. presso cui o presso le cui postazioni devono trovarsi i documenti da ricercare
UoPresso.Postazione	Postazione della U.O. presso cui o presso le cui postazioni devono trovarsi i documenti da ricercare
UoPresso.FlgInclSottouo	Se 1 il filtro sulla uo presso cui o presso le cui postazioni devono trovarsi i documenti da ricercare è soddisfatto anche se i documenti si trovano presso sotto-uo di UoPresso.IdUo (o loro postazioni)
Uo1Ass.IdUo	id. della U.O. prima assegnataria dei documenti da ricercare

Attributo o Elemento	***Descrizione***
Uo1Ass.Liv1	Settore della U.O. prima assegnataria dei documenti da ricercare
Uo1Ass.Liv2	Servizio della U.O. prima assegnataria dei documenti da ricercare
Uo1Ass.Liv3	UOC della U.O. prima assegnataria dei documenti da ricercare
Uo1Ass.Liv4	UOS della U.O. prima assegnataria dei documenti da ricercare
Uo1Ass.Postazione	Postazione della U.O. prima assegnataria dei documenti da ricercare
Uo1Ass.FlgInclSottouo	Se 1 il filtro sulla uo di prima assegnazione è soddisfatto anche se i documenti hanno come UO di prima assegnazione una sotto-UO di Uo1Ass.IdUo (o sue postazioni)
FlgClassif	Stato di classificazione: S=Solo copie classificate; N=Copie non ancora classificati; T=qualsiasi stato
FlgFascicolazione	Stato di fascicolazione: S=Solo copie fascicolate; N=Copie non ancora fascicolate; T =qualsiasi stato
IdIndice	Id. della voce di indice del documento
Classif_FascApp.AnnoFasc	Anno del fascicolo di appartenenza
Classif_FascApp.Classif.TitoloRomano	Titolo/Categoria di classificazione (espresso in numero romano)
Classif_FascApp.Classif.Classe	Classe
Classif_FascApp.Classif.Sottoclasse	Sottoclasse
Classif_FascApp.Classif.IdTitolazione	Id della classificazione archivistica
Classif_FascApp.ProgrFasc	Numero progressivo del fascicolo di appartenenza
Classif_FascApp.NumSottofasc	Numero del sottofascicolo
Classif_FascApp.IdFascicolo	Id del fascicolo di appartenenza
NoteCopia	Note della copia

Risposta.

La risposta conterrà 0 o più elementi XML di tipo <Documento>.

La risposta del WS viene restituita in formato XML secondo il tracciato definito nel file RisultatoRicerca.dtd

Attributo o elemento	***Descrizione***
Stato	***(Codice, Messaggio)***
Codice	Codice di ritorno del WS
Messaggio	Messaggio associato al codice di ritorno
Tipo	Tipo di protocollo (PG) o repertorio
Anno	Anno di protocollazione o repertoriazione
Numero	Numero del protocollo o del repertorio
NumeroCopia	Numero della copia
DataArrivo	Data estesa di arrivo del documento
FlgCompl	Flag di completezza della documentazione
FlgRiservato	Flag di riservatezza del documento
FlgEvdidenza	Flag di evidenza del documento
TipoFisico	Identificativo del tipo fisico
TipoLogico	Identificativo del tipo logico
SottoTipoLogico	Identificativo del sottotipo logico
Oggetto	Testo oggetto del documento

Attributo o elemento	***Descrizione***
Note	Note da allegare alla protocollazione
RifProv	Riferimento della provenienza
ProtProv	Protocollo della provenienza
DataProv	Data della provenienza
FlgNoPubbl	Flag di riservatezza al pubblico
DataTermNoPubbl	Data di termine della riservatezza al pubblico
NrimandaOrig	Indica se l'originale di un documento in uscita deve essere inviato all' uo mittente
Nominativo	Elenco degli esibenti o destinatari con le seguenti informazioni anagrafiche: attributi flgDestCopia (S | N) "N" flag originale/copia flgTpAnag (P | D) "P" flag persona/ditta IdAnag = codice anagrafico Cognome, Nome, RagioneSociale = desc.anagrafica CodFis = codice fiscale ParIva = partita I.V.A.
Indirizzo	Elenco degli indirizzi degli esibenti o destinatari con le seguenti informazioni: IdTopon = identificativo della toponomastica DesInd = descrizione dell'indirizzo NumCiv = numero civico CodiceISTAT = codice Istat del Comune Cap = C.A.P. Provincia Nazione Telefono Fax
Firmatario	Firmatario con le seguenti informazioni anagrafiche: attributi flgTpAnag (P | D) "P" IdAnag = codice anagrafico Cognome, Nome, RagioneSociale = desc.anagrafica CodFis = codice fiscale ParIva = partita I.V.A. **Indirizzo…**
UO IdUO	Identificativo dell'unità operativa
Sett	Settore della U. O.
Serv	Servizio della U. O.
UOC	U. O. C. della U. O.
UOS	U. O. S. della U. O.
Post	Postazione della U. O. di provenienza
UOProv	Elemento di tipo **UO =** U.O. di provenienza
UOAss	Elemento di tipo **UO =** U.O. destinataria
Allegato	Elenco degli allegati con le seguenti informazioni: attributi: Tipo = tipo di allegato Note = note all'allegato

3.3 Servizio di ricerca anagrafica

Il servizio consente di effettuare una ricerca sulla banca dati per reperire le informazioni di una o più anagrafiche esterne.
E' possibile ricercare un nominativo specificando il codice fiscale oppure il Cognome Nome / Ragione Sociale:
se nella ricerca è stata specificata il cf, il risultato sarà composto dall'eventuale anagrafica attiva identificata dal cf stesso

se nella ricerca non è stata specificata il cf, il risultato sarà composto dall'insieme delle anagrafiche attive identificate dal Nominativo (ricerca sql in LIKE) o da Codice Fiscale / Partita Iva.

La risposta conterrà 0 o più elementi XML di tipo <Anagrafica>.

Per ogni Anagrafica vengono visualizzati i seguenti dati:
Codice anagrafico, Matricola, Tipo P / D (ovvero Persona o Ditta), Nominativo, Codice fiscale, Partita Iva, Indirizzo di residenza, Data di fine validità della posizione anagrafica.
Le informazioni che vengono gestite dal servizio, descritte nel file RicercaAnagrafica.dtd sono:

Attributo o elemento	***Descrizione***	***Obbligatorio***
RicercaAnagrafica	Root element. Contiene al suo interno un elemento di tipo Matricola oppure di tipo DatiAnag	
Matricola	Matricola del cittadino	
DatiAnag	Contiene al suo interno uno dei seguenti elementi: Codice Nominativo	
Codice	Contiene al suo interno uno dei seguenti elementi: CodiceFiscale: Codice fiscale della persona PartitaIva: Partita Iva della persona	
Nominativo	Nominativo della persona	

Matricola e DatiAnag sono in alternativa: se specificata, la ricerca viene effettuata solo con la matricola, ignorando il contenuto degli altri eventuali campi.

Esempio di ricerca per Matricola

```
<?xml version="1.0" ?>
<!DOCTYPE RicercaAagrafica">
<RicercaAnagrafica versione="2004-01-19" xml:lang="it">
<Matricola>SPGB20571654</Matricola>
</RicercaAnagrafica>
```

Esempio di ricerca per Nominativo e/o Codice Fiscale

```
<?xml version="1.0" ?>
<!DOCTYPE RicercaAagrafica>
<RicercaAnagrafica versione="2004-01-19" xml:lang="it">
<DatiAnag>
  <Codice>
    <CodiceFiscale>RSSLSN76L55D969W</CodiceFiscale>
  </Codice>
  <Nominativo>ROSSI</Nominativo>
</DatiAnag>
</RicercaAnagrafica>
```

Risposta.

La risposta del WS viene restituita in formato XML secondo il tracciato definito nel file RisultatoRicercaAnagrafica.dtd.

Se nella ricerca è stata specificata la Matricola, il risultato sarà composto dall'eventuale anagrafica attiva identificata dalla matricola stessa.

Se nella ricerca non è stata specificata la Matricola, il risultato sarà composto dall'insieme delle anagrafiche attive identificate dal Nominativo (ricerca sql in LIKE) o da CodiceFiscale / PartitaIva.

Attributo o elemento	***Descrizione***
Stato	(Codice, Messaggio)
Codice	Codice di ritorno del WS
Messaggio	Messaggio associato al codice di ritorno
Anagrafica	Codice = codice dell'anagrafica Matricola = Matricola del cittadino Tipo = **P**(persona fisica), **D**(ditta)
Nominativo	Nominativo dell'anagrafica
CodiceFiscale	Codice Fiscale
PartitaIva	Partita Iva
DescrizioneResidenza	Indirizzo di residenza
DataFineValidita	Data fine validità anagrafica

Esempio di risposta del WS

```
<?xml version="1.0" ?>
<RisultatoRicercaAnagrafica>
    <Stato>
        <Codice>0</Codice>
        <Messaggio>Trovati 2 record</Messaggio>
    </Stato>
<Anagrafica codice="9077" matricola="" tipo="P">
        <Nominativo>CRICCA LUIGI</Nominativo>
        <CodiceFiscale></CodiceFiscale>
        <DescrizioneResidenza></DescrizioneResidenza>
        <DataFineValidita></ DataFineValidita >
</Anagrafica>
<Anagrafica codice="100414" matricola="SPGB20571654" tipo="P">
        <Nominativo>GIACOPETTI ALESSANDRA</Nominativo>
        <CodiceFiscale>GCPLSN76L55D969W</CodiceFiscale>
        <DescrizioneResidenza></DescrizioneResidenza>
        <DataFineValidita></ DataFineValidita >
</Anagrafica>
</RisultatoRicercaAnagrafica>
```

3.4 Servizio di acquisizione da sportello

Il servizio consente di effettuare una protocollazione (acquisizione) da sportello.
Per quanto riguarda i MITTENTI ESTERNI è possibile indicare:
una persona fisica / giuridica già esistente in archivio
oppure
una persona fisica / giuridica da inserire per la prima volta in archivo.

Nel primo caso è necessario valorizzare l'identificativo anagrafico (ID_ANAG), reperito tramite il WS di Ricerca Anagrafica;
nel secondo caso bisogna specificare il Nome Cognome o Ragione Sociale, l'eventuale matricola e l'eventuale codice fiscale o partita iva della nuova persona fisica / giuridica da inserire.
Richiesta.

Le informazioni che vengono gestite dal servizio, descritte nel file SegnaturaAcqSportello.dtd sono:

Attributo o elemento	***Descrizione***	***Prot entrata***	***Prot uscita***	***Obbligatorio***
DtArrivoIn	Data estesa di arrivo del documento	✓		✓
TxtOggIn	Testo oggetto del documento	✓	✓	✓
RifProvIn	Riferimento della provenienza	✓		
DtProvIn	Data della provenienza	✓		
Firm	Elenco dei **MITTENTI esterni** con le seguenti informazioni: flgTpAnag (P | D) "P" IdAnag – codice anagrafico Matricola – matricola del cittadino Cognome, Nome, RagioneSociale – desc.anagrafica CodFis – codice fiscale ParIva – partita I.V.A.	✓[19]		

NB. Il formato dei campi Data è GG/MM/AAAA.

Risposta.

La risposta del WS viene restituita in formato XML secondo il tracciato definito nel file Risposta.dtd solo per gli elementi indicati.

Attributo o elemento	***Descrizione***
Stato	(Codice, Messaggio)
Codice	Codice di ritorno del WS
Messaggio	Messaggio associato al codice di ritorno
NumeroProtocollo	Tipo = tipo di protocollo

[19] Se l'Anagrafico Mittente è esistente, deve essere valorizzato il relativo identificativo (ID_ANAG); se l'Anagrafico è insesistente, è necessario indicare il Cognome Nome – Ragione Sociale, l'eventuale matricola, l'eventuale codice fiscale – partita iva e non valorizzare l'identificativo (ID_ANAG).

Attributo o elemento	*Descrizione*
	Anno = anno di protocollazione Numero = Numero di protocollo
DataProtocollazione	Data di protocollazione

3.5 Servizio di ricerca fascicolo

Richiesta.

Le informazioni che vengono gestite dal servizio, descritte nel file RicercaFascicoli.dtd sono:

Attributo o elemento	*Descrizione*	*Obbligatorio*
IdUoIn	ID_UO della postazione di lavoro.	✓
OggettoIn	Oggetto del documento (o parte di esso).	
Classif_FascApp.AnnoFasc	Anno del fascicolo di appartenenza.	
Classif_FascApp.Classif.Titolo	Titolo/Categoria di classificazione.	
Classif_FascApp.Classif.Classe	Classe	
Classif_FascApp.Classif.SottoClasse	Sottoclasse	
Classif_FascApp.Classif.Livello4	4° livello	
Classif_FascApp.Classif.Livello5	5° livello	
Classif_FascApp.Classif.IdTitolazione	Id titolazione della classificazione.	
Classif_FascApp.ProgrFasc	Numero del fascicolo.	
Classif_FascApp.NumSottofasc	Numero del sotto fascicolo.	
Classif_FascApp.IdFascicolo	Id fascicolo.	
DesTitolazioneIn	Descrizione della titolazione	
FlgStatoIn	Stato de fascicolo. Valori possibili: NULL = nessun criterio, A = Solo fascicoli aperti, C = Solo fascicoli chiusi, AC = In Archivio corrente, AD = In archivio di deposito, AS = In archivio storico, SC = Scartati.	
DtAperturaDaIn	Dalla data apertura del fascicolo	
DtAperturaAIn	Alla data apertura del fascicolo	
DtChiusuraDaIn	Dalla data chiusura del fascicolo	
DtChiusuraAIn	Alla data chiusura del fascicolo	
IdUOAperturaIn.IdUO	ID UO che ha aperto i fascicoli estratti (viene ignorato se almeno uno dei livelli della UO non è valorizzato)	
IdUOAperturaIn.Liv1	Settore della UO di apertura.	
IdUOAperturaIn.Liv2	Servizio della UO di apertura.	
IdUOAperturaIn.Liv3	Uoc della UO di apertura.	
IdUOAperturaIn.Liv4	Uos della UO di apertura.	
IdUOAperturaIn.Postazione	Postazione della UO di apertura.	
IdUOAperturaIn.FlgInclSottouo	se 1 il filtro precedente è soddisfatto anche se i fascicoli sono stati aperti da postazioni delle sotto-UO di IdUOAperturaIn.	
IdUOPressoIn.IdUO	ID UO presso cui o presso le cui postazioni (solo quelle direttamente collegate) devono trovarsi i fascicoli estratti (viene ignorato se almeno uno dei livelli della UO non è valorizzato)	

IdUOPressoIn.Liv1	Settore della UO di appartenenza.	
IdUOPressoIn.Liv2	Servizio della UO di appartenenza.	
IdUOPressoIn.Liv3	Uoc della UO di appartenenza.	
IdUOPressoIn.Liv4	Uos della UO di appartenenza.	
IdUOPressoIn.Postazione	Postazione della UO di appartenenza.	
IdUOPressoIn.FlgInclSottouo	Se 1 il filtro sulla uo presso cui o presso le cui postazioni devono trovarsi i documenti da ricercare è soddisfatto anche se i documenti si trovano presso sotto-uo di UoPresso.IdUo (o loro postazioni)	
IdTpProcIn	Identificativo del tipo di procedimento associato al fascicolo.	
FlgAnnullamentoIn	filtra sulla validità/annullamento dei fascicoli. Valori possibili: NULL=Nessun criterio, S=Solo annullati, N=Validi	
ChiavexRicercaIn	parola/frase chiave che deve essere contenuta tra le parole/frase chiave del fascicolo.	
Classif_FascRifIn.AnnoFasc	Anno del fascicolo di riferimento.	
Classif_FascRifIn.Classif.Titolo	Titolo/Categoria di classificazione di riferimento.	
Classif_FascRifIn.Classif.Classe	Classe di riferimento.	
Classif_FascRifIn.Classif.SottoClasse	Sottoclasse di riferimento.	
Classif_FascRifIn.Classif.Livello4	4° livello di riferimento.	
Classif_FascRifIn.Classif.Livello5	5° livello di riferimento.	
Classif_FascRifIn.Classif.IdTitolazione	Id titolazione della classificazione di riferimento.	
Classif_FascRifIn.ProgrFasc	Numero del fascicolo di riferimento.	
Classif_FascRifIn.NumSottofasc	Numero del sotto fascicolo di riferimento.	
Classif_FascRifIn.IdFascicolo	Id fascicolo di riferimento.	
AttrOpzIn.Id_attributo	ID attributo opzionale.	
AttrOpzIn.datatype	TIPO attributo opzionale. Valori possibili : VARCHAR2, NUMBER, DATE , DATETIME, CHECK	
AttrOpzIn.operatore	OPERATORE attributo opzionale. Valori possibili : TRA, > , >= , < , <= , =	
AttrOpzIn.valore1	Dal valore.	
AttrOpzIn.valore2	Al valore.	
TpOrderByIn	Indica il criterio in base a cui devono essere ordinati i record estratti Valori possibili: 'C'= per codice del fasc., ovvero x anno, classificazione, progressivo e n.ro di sottofasc; 'DA' = per data di apertura.	
FlgDescOrderByIn	1 = l'ordinamento deve essere discendente. 0 = l'ordinamento deve essere ascendente.	

Esempio ricerca

```
<?xml version="1.0" ?>
<!DOCTYPE RicercaFascicolo PUBLIC ""
"http://172.28.20.145:8088/protorerProd/includes/DTD/RicercaFascicoli.dtd">
        <RicercaFascicolo versione="2005-01-31" xml:lang="it">
              <IdUoIn>2</IdUoIn>
              <OggettoIn>%prova ws%</OggettoIn>
              <Classif_FascApp>
                    <CLASSIF_FASC>
                          <AnnoFasc>2006</AnnoFasc>
                          <Classif>
                                <CLASSIFICAZIONE>
                                      <Titolo>I</Titolo>
                                      <Classe>14</Classe>
                                      <SottoClasse></SottoClasse>
                                      <Livello4></Livello4>
                                      <Livello5></Livello5>
                                      <IdTitolazione></IdTitolazione>
                                </CLASSIFICAZIONE>
                          </Classif>
                          <ProgrFasc></ProgrFasc>
                          <NumSottofasc></NumSottofasc>
                          <IdFascicolo></IdFascicolo>
                    </CLASSIF_FASC>
              </Classif_FascApp>
              <DesTitolazioneIn>%Interventi di carattere%</DesTitolazioneIn>
              <FlgStatoIn></FlgStatoIn>
              <DtAperturaDaIn>22/07/2006</DtAperturaDaIn>
              <DtAperturaAIn>25/07/2006</DtAperturaAIn>
              <DtChiusuraDaIn></DtChiusuraDaIn>
              <DtChiusuraAIn></DtChiusuraAIn>
              <IdUOAperturaIn>
                    <UO>
                          <IdUO></IdUO>
                          <Liv1>99</Liv1>
                          <Liv2>0</Liv2>
                          <Liv3>0</Liv3>
                          <Liv4>0</Liv4>
                          <Postazione>1</Postazione>
                    </UO>
              </IdUOAperturaIn>
              <IdUOPressoIn>
                    <UO>
                          <IdUO></IdUO>
                          <Liv1></Liv1>
                          <Liv2></Liv2>
                          <Liv3></Liv3>
                          <Liv4></Liv4>
                          <Postazione></Postazione>
                    </UO>
              </IdUOPressoIn>
              <IdTpProcIn></IdTpProcIn>
              <FlgAnnullamentoIn></FlgAnnullamentoIn>
              <ChiavexRicercaIn></ChiavexRicercaIn>
```

```
<Classif_FascRifIn>
	<CLASSIF_FASC>
		<AnnoFasc></AnnoFasc>
		<Classif>
			<CLASSIFICAZIONE>
				<Titolo></Titolo>
				<Classe></Classe>
				<SottoClasse></SottoClasse>
				<Livello4></Livello4>
				<Livello5></Livello5>
				<IdTitolazione></IdTitolazione>
			</CLASSIFICAZIONE>
		</Classif>
		<ProgrFasc></ProgrFasc>
		<NumSottofasc></NumSottofasc>
		<IdFascicolo></IdFascicolo>
	</CLASSIF_FASC>
</Classif_FascRifIn>
<AttrOpzIn>
	<AttrOpz>
		<Id_attributo></Id_attributo>
		<datatype></datatype>
		<operatore></operatore>
		<valore1></valore1>
		<valore2></valore2>
	</AttrOpz>
</AttrOpzIn>
<TpOrderByIn></TpOrderByIn>
<FlgDescOrderByIn></FlgDescOrderByIn>
</RicercaFascicolo>
```

Risposta.

La risposta conterrà 0 o più elementi XML.

La risposta del WS viene restituita in formato XML secondo il tracciato definito nel file RisultatoRicercaFasc.dtd.

Attributo o elemento	**Descrizione**
Stato	(Codice, messaggio)
Codice	Codice di ritorno del WS
Messaggio	Messaggio associato al codice di ritorno
ID_FASCICOLO	Id fascicolo.
ANNO_FASC	Anno del fascicolo
NUM_FASC	Numero del Fascicolo
NUM_SOTTOFASC	Numero del sotto fascicolo.
TXT_OGG	Oggetto del fascicolo
TITOLO_CLASS	Titolo della classifica del fascicolo
CLASSE_CLASS	Classe della classifica del fascicolo
SOTTOCLASSE_CLASS	Sottoclasse della classifica del fascicolo
LIVELLO4_CLASS	Livello4 della classifica del fascicolo
LIVELLO5_CLASS	Livello5 della classifica del fascicolo
CODICE	<Anno>.<livelli classif.>.<n.ro progr.>[/<n.ro sottofasc> se >0]
DES_TITOLAZIONE	Descrizione della classificazione del fascicolo
DT_APERTURA	Data di apertura del fascicolo
DT_CHIUSURA	Data di chiusura del fascicolo
DT_ARCH	Data di archiviazione

Attributo o elemento	Descrizione
FLG_ANN	Flag di annullamento. Valori: S/N
DT_VERSAMENTO	Data di versamento in archivio storico
ID_FASCICOLO_RIF	Id. Del fascicolo di riferimento
NUM_SOTTOFASC_RIF	N.ro di sottofascicolo di riferimento
DEC_FASC_RIF	Codice e data di apertura del fascicolo/sottofasc. Di riferimento
MOTIVI_RIF	Motivi del collegamento al fascicolo di riferimento
PAROLE_CHIAVE	Lista di parole chiave del fascicolo

3.6 *Servizio di ricerca UO*

Richiesta.

Il WS effettua la ricerca dei dati (anagrafici e di coordinata archivistica) di una postazione dato in input l'identificativo. Le informazioni che vengono gestite dal servizio, descritte nel file RicercaUO.dtd sono:

Attributo o elemento	*Descrizione*	*Obbligatorio*
IdPostazLavoro	ID_UO della postazione di lavoro.	✓
UserName	UserName	
Matricola	Matricola	

Esempio per RicercaUO.dtd:

```
<?xml version="1.0" ?>
<!DOCTYPE RicercaUO PUBLIC ""
"http://172.28.20.145:8088/protoemilia/includes/DTD/RicercaUO.dtd">

<RicercaUO versione="2007-06-20" xml:lang="it">
       <IdPostazLavoro>2</IdPostazLavoro>

       <UserName>40830</UserName>
       <Matricola>40737</Matricola>

</RicercaUO>
```

Risposta.
La risposta conterrà 0 o più elementi XML di tipo < UOAnag>.

La risposta del WS viene restituita secondo il tracciato definito nel file RispostaRicercaUO.dtd.

Attributo o elemento	*Descrizione*
Stato	(Codice, Messaggio)
Codice	Codice di ritorno del WS
Messaggio	Messaggio associato al codice di ritorno
IdUte	Id utente
DesUte	Descrizione utente

Attributo o elemento	***Descrizione***
IdUO	Id UO
Liv1	Settore
Liv2	Servizio
Liv3	Uoc
Liv4	Uos
Postazione	Postazione
DesUO	Descrizione postazione
IdUO_P	Id UO padre
Liv1_P	Settore padre
Liv2_P	Servizio padre
Liv3_P	Uoc padre
Liv4_P	Uos padre
Postazione_P	Postazione padre
DesUO_P	Descrizione postazione padre
FlgAbil	Flag che informa sulla validità della postazione

3.7 Servizio di modifica protocollazione

Richiesta.
Il WS gestisce la modifica o l'annullo di metadati di protocollo; se la registrazione non conteneva allegati è inoltre possibile inserire (in una sola soluzione) uno o più allegati. L'ordine di inserimento corrisponde all'ordine in cui verranno registrati (il primo è assunto come file principale, il secondo come primo allegato etc.).

Le informazioni che vengono gestite dal servizio, descritte nel file ModificaDoc.xsd:

Attributo o elemento	***Descrizione***	***Obbligatorio***
PostazLavoro	ID_UO della postazione di lavoro.	✓
RegMod	Identificativo (id_doc o tipo, anno e numero) del documento da modificare	
MotivazioniAnn	Motivazioni dell'annullamento dati (da valorizzare solo se modificati dati soggetti ad annullamento)	
DtArrivo	Data e ora di arrivo (dato modificabile solo se registrazione in entrata) Dato soggetto ad annullamento	✓
Oggetto	Oggetto del documento. Dato soggetto ad annullamento	✓
EstremiRegEmerg	Estremi (tipo, anno e numero) del registro di emergenza di un eventuale registrazione di emergenza (modificabile solo se la registrazione non è fatta solo a sportello). Dato soggetto ad annullamento	
AttoAutoProtDifferita	Estremi dell'atto di autorizzazione alla prot. differita (può essere modificato solo se registrazione in entrata e non fatta solo a sportello). Dato soggetto ad annullamento	
DtProv	Data di protocollo del documento ricevuto (dato modificabile solo se registrazione in entrata). Dato soggetto ad annullamento	
RifProv	Riferimenti del documento ricevuto (dato modificabile solo se registrazione in entrata). Dato soggetto ad annullamento	

Attributo o elemento	***Descrizione***	***Obbligatorio***
ProtProv	Estremi di protocollo del documento ricevuto (dato modificabile solo se registrazione in entrata). Dato soggetto ad annullamento	
IdUOProv	Identificativo della U.O. mittente(id uo o i 5 livelli della pianta organica) (dato modificabile solo se registrazione in uscita o tra uffici). Dato soggetto ad annullamento	
MittDestEsterni	Array con i dati dei mittenti (se registrazione in entrata) o destinatari (se registrazione in uscita o tra uffici). Dato soggetto ad annullamento	
FlagMDEAppSost	Flag che stabilisce se i MittDestEsterni indicati devono essere aggiunti a quelli già presenti; valori possibili: 'A' → Append. Per valori diversi da 'A' viene fatta la sostituzione dei nominativi, se presenti in input.	
TpFisico	Mezzo di trasmissione (tabella ptt_tpfis). Dato NON soggetto ad annullamento	
DtRaccomandata	Data di raccomandata. Dato NON soggetto ad annullamento	
NrRaccomandata	Numero di raccomandata (alfanumerico). Dato NON soggetto ad annullamento	
TipoLogico	tipo di documento (tabella ptt_tplog). Dato NON soggetto ad annullamento	
SottoTipoLogico	Sottotipo di documento (tabella ptt_tplog). Considerato solo se TipoLogico è valorizzato . Dato NON soggetto ad annullamento	
FlgTpRegDocPrec	Indica il tipo di registrazione del doc. precedente; valori: P=Prot. Gen., R=Altro. Dato NON soggetto ad annullamento	
EstremiDocPrec	Estremi del documento precedente. Dato NON soggetto ad annullamento	
FlgDocNoPubbl	Flag di riservatezza al pubblico (con valori S/N): Se S il doc. non è visibile al pubblico, se N sì. Dato NON soggetto ad annullamento	
DtTermNoPubbl	Data di termine della riservatezza al pubblico Dato NON soggetto ad annullamento	
FlgDocRsv	Flag di riservatezza (con valori S/N): Se S il doc. è riservato, se N non è riservato. Dato NON soggetto ad annullamento	
DtTermRsvPP	Data di termine della riservatezza data del protocollo particolare. Dato NON soggetto ad annullamento	
FlgEvidenza	Flag di evidenza (con valori S/N): Se S il doc. è in evidenza, se N non lo è. Dato NON soggetto ad annullamento	
Note	Note della registrazione. Dato NON soggetto ad annullamento	
TpRepDef	Se la registrazione è un repertorio provvisorio, questo campo contiene la sigla del repertorio definitivo in cui sarà convertito il documento. Dato NON soggetto ad annullamento	
DocPrimario	Dato del documento primario. Dato soggetto ad annullamento	
Allegati	Array con i dati degli allegati. Dato NON soggetto ad annullamento	

Insieme alla richiesta xml sarà possibile allegare i file da inserire come allegati elettronici al documento primario o agli allegati.
Si ricorda che è possibile inserire un documento elettronico (primario o allegato) qualora non fosse presente, solo se non ci sono già presenti file elettronici legati alla registrazione quindi come nella modifica della registrazione disponibile da E-Grammata non è gestito un versioning né per il file primario tanto meno per gli allegati.

Per quanto riguarda gli allegati valgono le regole che seguono, che riprendono il comportamento della funzionalità di modifica registrazione presente in E-Grammata, il WS assume due linee di comportamento secondo una principale discriminante, la presenza o meno di un file elettronico legato alla registrazione (tra primario e allegati) al momento della modifica.

Nel caso in cui la registrazione <u>non abbia</u> allegati files elettronici:

- L'array degli allegati - passato in input - viene sostituito all'array degli allegati già legati alla registrazione, quindi è possibile aggiungere, cancellare e modificare tutti gli allegati, per quanto riguarda gli attributi TIPO e DESCRIZIONE.

Finchè non c'è l'allegato elettronico tutto ciò che viene inserto nell'array va in sovrascrittura dell'esistente e non in aggiunta.
Se l'array che si passa è vuoto ma sul sistema c'era già qualcosa questo viene cancellato.
<numalleg> in questo caso è insignificante.

Nel caso in cui la registrazione <u>abbia</u> allegati files elettronici:

- Con la modifica non è possibile cancellare allegati già presenti
- Con la modifica non è possibile aggiungere allegati
- Con la modifica è possibile modificare la descrizione ed il tipo di file di allegati già presenti
- <u>Attributi obbligatori</u>: num allegato e tipo allegato
- Si può passare l'array degli allegati da modificare indicando solo gli allegati soggetti a modifica e tralasciando quelli che non subirebbero modifiche.

La prima volta che si passa un file elettronico tra primario e allegati, non si può più modificare l'array degli allegati si possono solo modificare i campi tipo, descrizione e note in questo caso si può definire quale allegato si vuole modificare nei dati indicati allora bisogna usare il numalleg come identificativo del file elettronico di cui si vogliono modificare i dati, se non si specifica il numalleg si deve passare tutto l'array.

Si riporta di seguito un esempio di un caso di una registrazione che non abbia allegati; vi vengono inseriti un allegato senza file elettronico e un allegato con file firmato

Esempio
Allegato 1 senza file elettronico
Allegato 2 con file elettronico firmato

Dopo la modifica in realtà ci saranno tre file elettronici associati alla registrazione nel seguente ordine rispetto all'esempio:

Allegato 1 senza file elettronico
Allegato 2 con file elettronico sbustato
Allegato 3 il file elettronico firmato

Per cui nelle successive richieste di modifica bisogna fare attenzione a quale posizione ha raggiunto il file che si vuole modificare.

NOTA BENE: relativamente all'ordine degli allegati: verranno riportati legati alla registrazione nell'ordine in cui sono stati passati in input al WS.

Risposta.

La risposta conterrà solo un messaggio per segnalare se la modifica e' andata a buon fine o in caso contrario ritornerà una segnalazione di errore

Attributo o elemento	***Descrizione***
Stato	(Codice, Messaggio)

Attributo o elemento	***Descrizione***
Codice	Codice di ritorno del WS
Messaggio	Messaggio associato al codice di ritorno

Esempio:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
    <RispostaModReg>
       <Stato>
           <Codice>0</Codice>
           <Messaggio>Modifica eseguita con successo</Messaggio>
        </Stato>
    </RispostaModReg>
```

3.8 Servizio di estrazione documenti elettronici

Richiesta.

Le informazioni che vengono gestite dal servizio, descritte nel file EstraiFileElettronici.xsd sono:

Attributo o elemento	***Descrizione***	***Obbligatorio***
PostLavoro	ID_UO della postazione di lavoro.	✓
IdDocumento	IdDoc del documento in esame	
RegType	Tripla [TIPO/ANNO/NUMERO].	

Esempio di request.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<Dati xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
       <PostLavoro>
                    <IdUo>2</IdUo>
       </PostLavoro>
       <documento>
              <IdDocumento>59375</IdDocumento>
       </documento>
</Dati>
```

Risposta.

La risposta conterrà 1 elemento XML di tipo < RisultatoRicerca>, lo stato, il codice e il messaggio di ritorno del WS, ed infine 0 o più descrizioni dei file elettronici allegati.

Risposta:

Attributo o elemento	***Descrizione***
Stato	(Codice, Messaggio)
Codice	Codice di ritorno del WS

Messaggio	Messaggio associato al codice di ritorno
IdDocElettronicoOut	Id doc del documento primario
DocElettronicoOut	Nome del file del documento primario
Dimensione	Dimensione del file del documento primario
IdDocEl	Id_doc dell'allegato
AttachNumber	Numero dell'attachment
FileName	Nome del file allegato
Note	Note allegato
Tipo	Tipo allegato
Dimensione	Dimensione Allegato

Esempio

```
<?xml version="1.0" ?>
<RisultatoRicerca>
    <Stato>
        <Codice>0</Codice>
        <Messaggio>Estrazione file Elettronici, trovati: 2</Messaggio>
    </Stato>
    <DocPrimario>
          <IdDocElettronicoOut>4506</IdDocElettronicoOut>
          <DocElettronicoOut>DAVARINICERTIFICATONUOVO.xml.p7m</DocElettronicoOut>
          <Dimensione>3152 Byte</Dimensione>
    </DocPrimario>
    <Allegato>
          <IdDocEl>4536</IdDocEl>
          <AttachNumber>1</AttachNumber>
          <FileName>doc2.txt</FileName>
          <Note>descrizione doc2.txt</Note>
          <Tipo>2</Tipo>
          <Dimensione>24 Byte</Dimensione>
    </Allegato>
</RisultatoRicerca>
```

3.9 Servizio crea copia

Richiesta.
Le informazioni che vengono richieste dal servizio, descritte nel file CreaCopia.xsd, sono:

Attributo o elemento	*Descrizione*	*Obbligatorio*
PostLavoro	ID_UO o 5 Livelli della postazione di lavoro.	✓
PostLavoro.IdUo	IdUo.	
PostLavoro.Livelli	Livello: mediante gli attributi "Nro" e "Codice" per identificare la cinquina.	
Documento	Documento da cui effettuare la/e copia/e.	✓
Documento.IdDocumento	IdDoc del documento.	
Documento.SiglaType	SiglaType: Con i sotto Tag "Tipo", "Anno" e "Numero" identifica il documento.	
Copia	Array delle copia/e che si vuole creare.	✓

Attributo o elemento	***Descrizione***	***Obbligatorio***
Copia.UOAss	ID_UO o 5 Livelli della postazione di lavoro.	✓
Copia.UOAss.IdUo	IdUo	
Copia.UOAss.Livelli	Livelli: Con i sotto Tag “Settore”, “Servizio”, “Uoc”, “Uos” e “Postazione” identifica la cinquina.	
Copia.IdInd	Indice della copia.	
Copia. FlgCopiaConoscenza	Identifica con il valore “S” o “N” se si tratta di una copia per conoscenza.	
Copia. Fascicolo.IdFascicolo	IdFascicolo	
Copia. Fascicolo.FascicoloDett	FascicoloDett: Con i sotto Tag “Anno”, “Numero” e il precedente “IdInd” identifica il fascicolo.	
Copia. Fascicolo. SottoFascicolo	SottoFascicolo.	
Copia. NoteInvio	Note di invio copia.	

Per ogni Tag Copia inserito nell’xml di request verrà inserita una copia nel sistema.
Il Tag “UOAss” è l’unico campo obbligatorio, mentre FlgCopiaConoscenza, indice, fascicolo e note sono facoltativi.
Il “FlgCopiaConoscenza” identifica se si tratta di una “Copia Conoscenza”, qualora non sia specificato la copia sarà di default una normale copia archivistica.

Esempio:

```
<?xml version="1.0" encoding="UTF-8"?>
<Dati xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

	<PostLavoro>
		<!-- <IdUo>2</IdUo> -->
		<Livelli Nro="1" Codice="99"></Livelli>
		<Livelli Nro="2" Codice="00"></Livelli>
		<Livelli Nro="3" Codice="00"></Livelli>
		<Livelli Nro="4" Codice="00"></Livelli>
		<Livelli Nro="5" Codice="01"></Livelli>
	</PostLavoro>

	<Documento>
<!-- <IdDocumento>60293</IdDocumento> -->
		<SiglaType>
			<Tipo>PG</Tipo>
			<Anno>2008</Anno>
			<Numero>190</Numero>
		</SiglaType>
	</Documento>

		<Copia>
			<UOAss>
				<IdUo>2</IdUo>
<!--
				<Livelli>
					<Settore>99</Settore>
					<Servizio>00</Servizio>
					<Uoc>00</Uoc>
					<Uos>00</Uos>
					<Postazione>11</Postazione>
				</Livelli>
-->
			</UOAss>
			<IdInd>4001</IdInd>
			<FlgCopiaConoscenza>N</FlgCopiaConoscenza>
			<Fascicolo>
<!-- <IdFascicolo>13807</IdFascicolo> -->
				<FascicoloDett>
					<Anno>2008</Anno>
					<Numero>3</Numero>
				</FascicoloDett>
				<SottoFascicolo>0</SottoFascicolo>
			</Fascicolo>
			<NoteInvio>nota invio di sp</NoteInvio>
		</Copia>

		<Copia>
			<UOAss>
				<IdUo>2</IdUo>
			</UOAss>
			<FlgCopiaConoscenza>S</FlgCopiaConoscenza>
			<NoteInvio>CONOSCENZA 1</NoteInvio>
		</Copia>

</Dati>
```

Risposta.

La risposta conterrà solo un messaggio per segnalare se la creazione e' andata a buon fine o in caso contrario ritornerà una segnalazione di errore

Attributo o elemento	***Descrizione***
Stato	(Codice, Messaggio)
Codice	Codice di ritorno del WS
Messaggio	Messaggio associato al codice di ritorno

Esempio:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<RispostaCreaCopia>
     <Stato>
          <Codice>0</Codice>
          <Messaggio>Copia creata con successo</Messaggio>
     </Stato>
</RispostaCreaCopia>
```

4 Appendice A: Elenco degli schemi DTD

Di seguito si dettagliano i dtd delle chiamate e delle risposte dei servizi web come da tabella 1.2.

4.1 SegnaturaProt.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY % dataPubblicazione "2003-12-09">
<!ELEMENT SegnaturaGenerica (Dati)>
<!ATTLIST SegnaturaGenerica
  versione NMTOKEN #FIXED "2003-12-09"
  xml:lang NMTOKEN #FIXED "it"
>
<!--
********************** Dati ******************************************
* E' rilevante se la Uo che è loggata (IdUoIn) è diversa dalla Uo mittente (IdUoProvIO) viene fatto un movimento di iter
* per il  documento appena registrato dalla uo loggata alla uo mittente.
*
*
*************************************************************************
-->
<!ELEMENT Dati (IdUteIn, IdUOIn, FlgProtGen?, TipoRep?, DtArrivoIn, FlgComplIn?, FlgRsvIn?, FlgEvdIn?, TipoFisicoIn?, TipoLogicoIn?, SttpLogicoIn?, TxtOggIn, NoteIn?, RifProvIn?, ProtProvIn?, DtProvIn?, IdIndice, IdTitolazione, IdFascicolo, NumFasc, NumSottofasc, AnnoFasc, FlgNoPubblIn?, DtTermNoPubblIn?, NrimandaOrigIn?, EsibDest*, Firm*, UOProv?, CopieArrIn*, DocumentoElettronico?, AllegaArrIn*, RegistroEmergenza?)>
<!ATTLIST Dati
  RimandaOrigIn (S | N) "N"
  TipoProtIN (E | U) "E"
>
<!ELEMENT IdUteIn (#PCDATA)>
```

```
<!ELEMENT IdUOIn (#PCDATA)>
<!ELEMENT FlgProtGen (#PCDATA)>
<!ELEMENT TipoRep (#PCDATA)>
<!ELEMENT DtArrivoIn (#PCDATA)>
<!ELEMENT FlgComplIn (#PCDATA)>
<!ELEMENT FlgRsvIn (#PCDATA)>
<!ELEMENT FlgEvdIn (#PCDATA)>
<!ELEMENT TipoFisicoIn (#PCDATA)>
<!ELEMENT TipoLogicoIn (#PCDATA)>
<!ELEMENT SttpLogicoIn (#PCDATA)>
<!ELEMENT TxtOggIn (#PCDATA)>
<!ELEMENT NoteIn (#PCDATA)>
<!ELEMENT RifProvIn (#PCDATA)>
<!ELEMENT ProtProvIn (#PCDATA)>
<!ELEMENT DtProvIn (#PCDATA)>
<!ELEMENT IdIndice (#PCDATA)>
<!ELEMENT IdTitolazione (#PCDATA)>
<!ELEMENT IdFascicolo (#PCDATA)>
<!ELEMENT NumFasc (#PCDATA)>
<!ELEMENT NumSottofasc (#PCDATA)>
<!ELEMENT AnnoFasc (#PCDATA)>
<!ELEMENT FlgNoPubblIn (#PCDATA)>
<!ELEMENT DtTermNoPubblIn (#PCDATA)>
<!ELEMENT NrimandaOrigIn (#PCDATA)>
<!--
************************** EsibDest
*******************************************************************************
* Regole aggiuntive *
* - un FlgDestCopia contraddistingue i destinatari di copie per conoscenza, può assumere i valori
(S/N),
* in entrata se valorizzato può assumere solo il valore N.
*
*******************************************************************************************************************
********
-->
<!ELEMENT EsibDest (IdAnag?, Matricola?, ((Cognome, Nome) | RagioneSociale), (CodFis |
ParIva)?, IndirizziEsibDest?)>
<!ATTLIST EsibDest
  flgDestCopia (S | N) "N"
  flgTpAnag (P | D) "P"
>
<!ELEMENT IdAnag (#PCDATA)>
<!ELEMENT Matricola (#PCDATA)>
<!ELEMENT Cognome (#PCDATA)>
<!ELEMENT Nome (#PCDATA)>
<!ELEMENT RagioneSociale (#PCDATA)>
<!ELEMENT CodFis (#PCDATA)>
<!ELEMENT ParIva (#PCDATA)>
<!ELEMENT IndirizziEsibDest (Indirizzo)>
<!ELEMENT Indirizzo ((IdTopon | DesInd), NumCiv?, EspCiv?, Comune, CAP?, Provincia?, Nazione?,
Telefono?, Fax?)>
<!ELEMENT IdTopon (#PCDATA)>
<!ELEMENT DesInd (#PCDATA)>
<!ELEMENT NumCiv (#PCDATA)>
<!ELEMENT EspCiv (#PCDATA)>
<!ELEMENT Comune (#PCDATA)>
<!ATTLIST Comune
```

```
  codiceISTAT CDATA #IMPLIED
>
<!ELEMENT CAP (#PCDATA)>
<!ELEMENT Provincia (#PCDATA)>
<!ELEMENT Nazione (#PCDATA)>
<!ELEMENT Telefono (#PCDATA)>
<!ATTLIST Telefono
  note CDATA #IMPLIED
>
<!ELEMENT Fax (#PCDATA)>
<!ATTLIST Fax
  note CDATA #IMPLIED
>
<!--
************************** Firm
*****************************************************************************
*
*
*****************************************************************************************************************************
********
-->
<!ELEMENT Firm (IdAnag?, Matricola?, ((Cognome, Nome) | RagioneSociale), (CodFis | ParIva),
IndirizziFirm?)>
<!ATTLIST Firm
  flgTpAnag (P | D) "P"
>
<!ELEMENT IndirizziFirm (Indirizzo)>
<!ELEMENT UO (IdUo | (SettIn, ServIn, UOCIn, UOSIn, PostIn))>
<!ELEMENT IdUo (#PCDATA)>
<!ELEMENT SettIn (#PCDATA)>
<!ELEMENT ServIn (#PCDATA)>
<!ELEMENT UOCIn (#PCDATA)>
<!ELEMENT UOSIn (#PCDATA)>
<!ELEMENT PostIn (#PCDATA)>
<!--
***************UOProv *****************
*******************************
-->
<!ELEMENT UOProv (UO)>
<!--
*****************CopieArrIn ***************
*******************************
-->
<!ELEMENT CopieArrIn (UoAss?, IdInd, AnnoFasc?, ProgrFasc?, Numsottofasc?)>
<!ATTLIST CopieArrIn
  flgorig CDATA #REQUIRED
  note CDATA #IMPLIED
  flgCC CDATA #REQUIRED
>
<!ELEMENT UoAss (UO)>
<!ELEMENT IdInd (#PCDATA)>
<!ELEMENT ProgrFasc (#PCDATA)>
<!ELEMENT Numsottofasc (#PCDATA)>
<!--
******************AllegaArrIn *************
*******************************
-->
```

```
<!ELEMENT AllegaArrIn (#PCDATA)>
<!ATTLIST AllegaArrIn
  Tipo CDATA #REQUIRED
  Note CDATA #IMPLIED
  NumeroAttachment CDATA #IMPLIED
  NomeFile CDATA #IMPLIED
>
<!ELEMENT DocumentoElettronico (#PCDATA)>
<!ATTLIST DocumentoElettronico
  NumeroAttachment CDATA #REQUIRED
  NomeFile CDATA #REQUIRED
>
<!ELEMENT RegistroEmergenza (#PCDATA)>
<!ATTLIST RegistroEmergenza
  Tipo CDATA #REQUIRED
  Anno CDATA #REQUIRED
  Numero CDATA #REQUIRED
```

4.2 SegnaturaAcqSportello.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY % dataPubblicazione "2003-12-09">
<!ELEMENT SegnaturaGenerica (Dati)>
<!ATTLIST SegnaturaGenerica
  versione NMTOKEN #FIXED "2003-12-09"
  xml:lang NMTOKEN #FIXED "it"
>
<!--
   ********** Dati **********************************************
* E` rilevante se la Uo che e` loggata (IdUoIn) e` diversa dalla Uo mittente (IdUoProvIO) viene fatto un
movimento di iter
* per il  documento appena registrato dalla uo loggata alla uo mittente.
*
*
*****************************************************************************
-->
<!ELEMENT Dati (IdUOIn?,DtArrivoIn, TxtOggIn, RifProvIn?, DtProvIn?, Firm*, UOProv*)>
<!ELEMENT IdUOIn (#PCDATA)>
<!ELEMENT DtArrivoIn (#PCDATA)>
<!ELEMENT TxtOggIn (#PCDATA)>
<!ELEMENT RifProvIn (#PCDATA)>
<!ELEMENT DtProvIn (#PCDATA)>

<!--
************************** Firm ************************
*********************************************************
-->

<!ELEMENT Firm (IdAnag?, ((Cognome, Nome) | RagioneSociale), (CodFis | ParIva))>
<!ATTLIST Firm  flgTpAnag (P | D) "P" >
<!ELEMENT IdAnag (#PCDATA)>
<!ELEMENT Cognome (#PCDATA)>
<!ELEMENT Nome (#PCDATA)>
<!ELEMENT RagioneSociale (#PCDATA)>
<!ELEMENT CodFis (#PCDATA)>
<!ELEMENT ParIva (#PCDATA)>

<!--
```

```
**********UOProv **********
******************************
-->

<!ELEMENT UOProv (UO)>
<!ELEMENT UO (IdUo | (SettIn, ServIn, UOCIn, UOSIn, PostIn))>
<!ELEMENT IdUo (#PCDATA)>
<!ELEMENT SettIn (#PCDATA)>
<!ELEMENT ServIn (#PCDATA)>
<!ELEMENT UOCIn (#PCDATA)>
<!ELEMENT UOSIn (#PCDATA)>
<!ELEMENT PostIn (#PCDATA)>
```

4.3 Risposta.dtd

```
<!ENTITY % dataPubblicazione "2004-01-19">
<!ELEMENT Risposta (Stato, NumeroProtocollo?, NuoveAnagrafiche?)>
<!ATTLIST Risposta
  versione NMTOKEN #FIXED "%dataPubblicazione;"
  xml:lang NMTOKEN #FIXED "it"
>
<!--
********** Stato **********************************************
* Rappresenta il codice di ritorno del servizio corredato di un eventuale
* messaggio di errore
* 0 = Servizio eseguito con successo
*****************************************************************
-->
<!ELEMENT Stato (Codice, Messaggio)>
<!ELEMENT Codice (#PCDATA)>
<!ELEMENT Messaggio (#PCDATA)>
<!--
********** ListaValori ***************************************
* Questo elemento permette di restituire i risultati prodotti dal servizio.
*****************************************************************
-->
<!ELEMENT NumeroProtocollo EMPTY>
<!ATTLIST NumeroProtocollo
  tipo NMTOKEN #REQUIRED
  anno NMTOKEN #REQUIRED
  numero NMTOKEN #REQUIRED
>
<!ELEMENT NuoveAnagrafiche (Anagrafica+)>
<!ELEMENT Anagrafica (Nominativo, (CodiceFiscale | PartitaIva)?, DescrizioneResidenza?,
DataFineValidita?)>
<!ATTLIST Anagrafica
  codice NMTOKEN #REQUIRED
  matricola CDATA ""
  tipo (P | D) "P"
>
```

4.4 RicercaProtocollo.dtd

Esempio 1

```
<!--
```

```
**********                Ricerca documenti                **********
********************************************************************************
-->
<!ENTITY % dataPubblicazione "2005-01-31">
<!ELEMENT RicercaProtocollo (IdPostazLavoro, RegPrimaria?, NumCopia?, FlgTpReg?, DtRegDa?,
DtRegA?, RegEmerg?,
            DtArrivoDa?, DtArrivoA?, UOReg?, SiglaAttoAutProtDiff?,AnnoAttoAutProtDiff?,
NumAttoAutProtDiff?,
            NominativoEsterno?, DtProv?, RifProv?, ProtProv?, UOProv?,
            IdTpFis?, DtRaccomandataDa?, DtRaccomandataA?, NroRaccomandata?, IdTpLog?,
IdSttpLog?,
            IdTpAlleg?, DesAlleg?, DocPrec?, Oggetto?, UOPresso?, UO1Ass?,
            FlgClassif?, FlgFascicolazione?, IdIndice?, Classif_FascApp?,
            NoteCopia?)>
<!ATTLIST RicercaProtocollo
  versione NMTOKEN #FIXED "2005-01-31"
  xml:lang NMTOKEN #FIXED "it"
  FlgTpReg (E | U | I | D | T) "T"
  FlgClassif (S | SC | SI | N | T) "T"
  FlgFascicolazione (S | N | T) "T"
>

<!ELEMENT EstremiReg (Sigla, Anno, Nro)>
<!ELEMENT Sigla (#PCDATA)>
<!ELEMENT Anno (#PCDATA)>
<!ELEMENT Nro (#PCDATA)>

<!ELEMENT UO (IdUO?, Liv1, Liv2, Liv3, Liv4, Postazione)>
<!ATTLIST UO
  FlgInclSottoUO (1 | 0) "0"
>

<!ELEMENT IdUO (#PCDATA)>
<!ELEMENT Liv1 (#PCDATA)>
<!ELEMENT Liv2 (#PCDATA)>
<!ELEMENT Liv3 (#PCDATA)>
<!ELEMENT Liv4 (#PCDATA)>
<!ELEMENT Postazione (#PCDATA)>


<!ELEMENT CLASSIFICAZIONE (Titolo, Classe, SottoClasse, Livello4, Livello5, IdTitolazione?)>
<!ELEMENT Titolo (#PCDATA)>
<!ELEMENT Classe (#PCDATA)>
<!ELEMENT SottoClasse (#PCDATA)>
<!ELEMENT Livello4 (#PCDATA)>
<!ELEMENT Livello5 (#PCDATA)>
<!ELEMENT IdTitolazione (#PCDATA)>

<!ELEMENT CLASSIF_FASC (AnnoFasc?, Classif, ProgrFasc?, NumSottofasc?, IdFascicolo?)>
<!ELEMENT AnnoFasc (#PCDATA)>
<!ELEMENT Classif (CLASSIFICAZIONE)>
<!ELEMENT ProgrFasc (#PCDATA)>
<!ELEMENT NumSottofasc (#PCDATA)>
<!ELEMENT IdFascicolo (#PCDATA)>

<!ELEMENT NOMINATIVO_ESTERNO (IdAnag?, Denominazione)>
<!ATTLIST NOMINATIVO_ESTERNO
```

```
  FlgTpNomEst (E | D | F | C | T) "T"
>
<!ELEMENT IdAnag (#PCDATA)>
<!ELEMENT Denominazione (#PCDATA)>

<!ELEMENT IdPostazLavoro (#PCDATA)>

<!--
********** Filtri di ricerca**********************************************************************
*** i filtri che iniziano per Dt vanno passati in formato DD/MM/YYYY
                                ***
*** nei filtri relativi a campi testo (Oggetto, DesAlleg, NoteCopia) va messo il % per indicare qualsiasi
stringa     ***
*****************************************************************************************************************
*
-->
<!ELEMENT RegPrimaria (EstremiReg)>
<!ELEMENT NumCopia (#PCDATA)>
<!ELEMENT FlgTpReg (#PCDATA)>
<!ELEMENT DtRegDa (#PCDATA)>
<!ELEMENT DtRegA (#PCDATA)>
<!ELEMENT RegEmerg (EstremiReg)>
<!ELEMENT DtArrivoDa (#PCDATA)>
<!ELEMENT DtArrivoA (#PCDATA)>
<!ELEMENT UOReg (UO)>

<!ELEMENT SiglaAttoAutProtDiff (#PCDATA)>
<!ELEMENT AnnoAttoAutProtDiff (#PCDATA)>
<!ELEMENT NumAttoAutProtDiff (#PCDATA)>

<!ELEMENT NominativoEsterno (NOMINATIVO_ESTERNO)>
<!ELEMENT DtProv (#PCDATA)>
<!ELEMENT RifProv (#PCDATA)>
<!ELEMENT ProtProv (#PCDATA)>
<!ELEMENT UOProv (UO)>

<!ELEMENT IdTpFis (#PCDATA)>
<!ELEMENT DtRaccomandataDa (#PCDATA)>
<!ELEMENT DtRaccomandataA (#PCDATA)>
<!ELEMENT NroRaccomandata (#PCDATA)>
<!ELEMENT IdTpLog (#PCDATA)>
<!ELEMENT IdSttpLog (#PCDATA)>

<!ELEMENT IdTpAlleg (#PCDATA)>
<!ELEMENT DesAlleg (#PCDATA)>
<!ELEMENT DocPrec (EstremiReg)>

<!ELEMENT Oggetto (#PCDATA)>
<!ELEMENT UOPresso (UO)>
<!ELEMENT UO1Ass (UO)>

<!ELEMENT FlgClassif (#PCDATA)>
<!ELEMENT FlgFascicolazione (#PCDATA)>
<!ELEMENT IdIndice (#PCDATA)>
<!ELEMENT Classif_FascApp (CLASSIF_FASC)>

<!ELEMENT NoteCopia (#PCDATA)>
```

Esempio 2

```
<!--
**********                    Ricerca documenti                    **********
********************************************************************************
-->
<!ENTITY % dataPubblicazione "2005-01-31">
<!ELEMENT RicercaProtocollo (IdPostazLavoro, RegPrimaria?, NumCopia?, FlgTpReg?, DtRegDa?,
DtRegA?, RegEmerg?,
            DtArrivoDa?, DtArrivoA?, UOReg?, AnnoAttoAutProtDiff?, NumAttoAutProtDiff?,
            NominativoEsterno?, DtProv?, RifProv?, ProtProv?, UOProv?,
            IdTpFis?, DtRaccomandataDa?, DtRaccomandataA?, NroRaccomandata?, IdTpLog?,
IdSttpLog?,
            IdTpAlleg?, DesAlleg?, DocPrec?, Oggetto?, UOPresso?, UO1Ass?,
            FlgClassif?, FlgFascicolazione?, IdIndice?, Classif_FascApp?,
            NoteCopia?)>
<!ATTLIST RicercaProtocollo
  versione NMTOKEN #FIXED "2005-01-31"
  xml:lang NMTOKEN #FIXED "it"
  FlgTpReg (E | U | I | D | T) "T"
  FlgClassif (S | N | T) "T"
  FlgFascicolazione (S | N | T) "T"
>

<!ELEMENT EstremiReg (Sigla, Anno, Nro)>
<!ELEMENT Sigla (#PCDATA)>
<!ELEMENT Anno (#PCDATA)>
<!ELEMENT Nro (#PCDATA)>

<!ELEMENT UO (IdUO?, Liv1, Liv2, Liv3, Liv4, Postazione)>
<!ATTLIST UO
  FlgInclSottoUO (1 | 0) "0"
>

<!ELEMENT IdUO (#PCDATA)>
<!ELEMENT Liv1 (#PCDATA)>
<!ELEMENT Liv2 (#PCDATA)>
<!ELEMENT Liv3 (#PCDATA)>
<!ELEMENT Liv4 (#PCDATA)>
<!ELEMENT Postazione (#PCDATA)>


<!ELEMENT CLASSIFICAZIONE (TitoloRomano, Classe, SottoClasse, IdTitolazione?)>
<!ELEMENT TitoloRomano (#PCDATA)>
<!ELEMENT Classe (#PCDATA)>
<!ELEMENT SottoClasse (#PCDATA)>
<!ELEMENT IdTitolazione (#PCDATA)>

<!ELEMENT CLASSIF_FASC (AnnoFasc?, Classif, ProgrFasc?, NumSottofasc?, IdFascicolo?)>
<!ELEMENT AnnoFasc (#PCDATA)>
<!ELEMENT Classif (CLASSIFICAZIONE)>
<!ELEMENT ProgrFasc (#PCDATA)>
<!ELEMENT NumSottofasc (#PCDATA)>
<!ELEMENT IdFascicolo (#PCDATA)>

<!ELEMENT NOMINATIVO_ESTERNO (IdAnag?, Denominazione)>
<!ATTLIST NOMINATIVO_ESTERNO
  FlgTpNomEst (E | D | F | C | T) "T"
```

```
>
<!ELEMENT IdAnag (#PCDATA)>
<!ELEMENT Denominazione (#PCDATA)>

<!ELEMENT IdPostazLavoro (#PCDATA)>

<!--
********** Filtri di ricerca**********************************************************************************
*** i filtri che iniziano per Dt vanno passati in formato DD/MM/YYYY
                                    ***
*** nei filtri relativi a campi testo (Oggetto, DesAlleg, NoteCopia) va messo il % per indicare qualsiasi
stringa     ***
*****************************************************************************************************************************
*
-->
<!ELEMENT RegPrimaria (EstremiReg)>
<!ELEMENT NumCopia (#PCDATA)>
<!ELEMENT FlgTpReg (#PCDATA)>
<!ELEMENT DtRegDa (#PCDATA)>
<!ELEMENT DtRegA (#PCDATA)>
<!ELEMENT RegEmerg (EstremiReg)>
<!ELEMENT DtArrivoDa (#PCDATA)>
<!ELEMENT DtArrivoA (#PCDATA)>
<!ELEMENT UOReg (UO)>
<!ELEMENT AnnoAttoAutProtDiff (#PCDATA)>
<!ELEMENT NumAttoAutProtDiff (#PCDATA)>

<!ELEMENT NominativoEsterno (NOMINATIVO_ESTERNO)>
<!ELEMENT DtProv (#PCDATA)>
<!ELEMENT RifProv (#PCDATA)>
<!ELEMENT ProtProv (#PCDATA)>
<!ELEMENT UOProv (UO)>

<!ELEMENT IdTpFis (#PCDATA)>
<!ELEMENT DtRaccomandataDa (#PCDATA)>
<!ELEMENT DtRaccomandataA (#PCDATA)>
<!ELEMENT NroRaccomandata (#PCDATA)>
<!ELEMENT IdTpLog (#PCDATA)>
<!ELEMENT IdSttpLog (#PCDATA)>

<!ELEMENT IdTpAlleg (#PCDATA)>
<!ELEMENT DesAlleg (#PCDATA)>
<!ELEMENT DocPrec (EstremiReg)>

<!ELEMENT Oggetto (#PCDATA)>
<!ELEMENT UOPresso (UO)>
<!ELEMENT UO1Ass (UO)>

<!ELEMENT FlgClassif (#PCDATA)>
<!ELEMENT FlgFascicolazione (#PCDATA)>
<!ELEMENT IdIndice (#PCDATA)>
<!ELEMENT Classif_FascApp (CLASSIF_FASC)>
<!ELEMENT NoteCopia (#PCDATA)>
```

Esempio 3

```
<!--
```

```
**********                Ricerca documenti                **********
*******************************************************************************
-->
<!ENTITY % dataPubblicazione "2005-01-31">
<!ELEMENT RicercaProtocollo (IdPostazLavoro, RegPrimaria?, NumCopia?, FlgTpReg?, DtRegDa?,
DtRegA?, RegEmerg?,
            DtArrivoDa?, DtArrivoA?, UOReg?, AnnoAttoAutProtDiff?, NumAttoAutProtDiff?,
            NominativoEsterno?, DtProv?, RifProv?, ProtProv?, UOProv?,
            IdTpFis?, DtRaccomandataDa?, DtRaccomandataA?, NroRaccomandata?, IdTpLog?,
IdSttpLog?,
            IdTpAlleg?, DesAlleg?, DocPrec?, Oggetto?, UOPresso?, UO1Ass?,
            FlgClassif?, FlgFascicolazione?, IdIndice?, Classif_FascApp?,
            NoteCopia?)>
<!ATTLIST RicercaProtocollo
  versione NMTOKEN #FIXED "2005-01-31"
  xml:lang NMTOKEN #FIXED "it"
  FlgTpReg (E | U | I | D | T) "T"
  FlgClassif (S | N | T) "T"
  FlgFascicolazione (S | N | T) "T"
>

<!ELEMENT EstremiReg (Sigla, Anno, Nro)>
<!ELEMENT Sigla (#PCDATA)>
<!ELEMENT Anno (#PCDATA)>
<!ELEMENT Nro (#PCDATA)>

<!ELEMENT UO (IdUO?, Liv1, Liv2, Liv3, Liv4, Postazione)>
<!ATTLIST UO
  FlgInclSottoUO (1 | 0) "si"
>

<!ELEMENT IdUO (#PCDATA)>
<!ELEMENT Liv1 (#PCDATA)>
<!ELEMENT Liv2 (#PCDATA)>
<!ELEMENT Liv3 (#PCDATA)>
<!ELEMENT Liv4 (#PCDATA)>
<!ELEMENT Postazione (#PCDATA)>


<!ELEMENT CLASSIFICAZIONE (TitoloRomano, Classe, SottoClasse, IdTitolazione?)>
<!ELEMENT TitoloRomano (#PCDATA)>
<!ELEMENT Classe (#PCDATA)>
<!ELEMENT SottoClasse (#PCDATA)>
<!ELEMENT IdTitolazione (#PCDATA)>

<!ELEMENT CLASSIF_FASC (AnnoFasc?, Classif, ProgrFasc?, NumSottofasc?, IdFascicolo?)>
<!ELEMENT AnnoFasc (#PCDATA)>
<!ELEMENT Classif (CLASSIFICAZIONE)>
<!ELEMENT ProgrFasc (#PCDATA)>
<!ELEMENT NumSottofasc (#PCDATA)>
<!ELEMENT IdFascicolo (#PCDATA)>

<!ELEMENT NOMINATIVO_ESTERNO(IdAnag?, Denominazione)>
<!ATTLIST NOMINATIVO_ESTERNO
  FlgTpNomEst (E | D | F | C | T) "T"
>
<!ELEMENT IdAnag (#PCDATA)>
```

```
<!ELEMENT Denominazione (#PCDATA)>

<!ELEMENT IdPostazLavoro(#PCDATA)>

<!--
********************** Filtri di ricerca*****************************************************************************
*** i filtri che iniziano per Dt vanno passati in formato DD/MM/YYYY
                                    ***
*** nei filtri relativi a campi testo (Oggetto, DesAlleg, NoteCopia) va messo il % per indicare qualsiasi
stringa     ***
*************************************************************************************************************************
*
-->
<!ELEMENT RegPrimaria (EstremiReg)>
<!ELEMENT NumCopia (#PCDATA)>
<!ELEMENT FlgTpReg (#PCDATA)>
<!ELEMENT DtRegDa (#PCDATA)>
<!ELEMENT DtRegA (#PCDATA)>
<!ELEMENT RegEmerg (EstremiReg)>
<!ELEMENT DtArrivoDa (#PCDATA)>
<!ELEMENT DtArrivoA (#PCDATA)>
<!ELEMENT UOReg (UO)>
<!ELEMENT AnnoAttoAutProtDiff (#PCDATA)>
<!ELEMENT NumAttoAutProtDiff (#PCDATA)>

<!ELEMENT NominativoEsterno(NOMINATIVO_ESTERNO)>
<!ELEMENT DtProv(#PCDATA)>
<!ELEMENT RifProv (#PCDATA)>
<!ELEMENT ProtProv(#PCDATA)>
<!ELEMENT UOProv (UO)>

<!ELEMENT IdTpFis (#PCDATA)>
<!ELEMENT DtRaccomandataDa (#PCDATA)>
<!ELEMENT DtRaccomandataA (#PCDATA)>
<!ELEMENT NroRaccomandata (#PCDATA)>
<!ELEMENT IdTpLog (#PCDATA)>
<!ELEMENT IdSttpLog (#PCDATA)>

<!ELEMENT IdTpAlleg (#PCDATA)>
<!ELEMENT DesAlleg (#PCDATA)>
<!ELEMENT DocPrec (EstremiReg)>

<!ELEMENT Oggetto (#PCDATA)>
<!ELEMENT UOPresso(UO)>
<!ELEMENT UO1Ass(UO)>

<!ELEMENT FlgClassif (#PCDATA)>
<!ELEMENT FlgFascicolazione (#PCDATA)>
<!ELEMENT IdIndice (#PCDATA)>
<!ELEMENT Classif_FascApp (CLASSIF_FASC)>

<!ELEMENT NoteCopia (#PCDATA)>
```

4.5 ***RisultatoRicerca.dtd***

```
<!ENTITY % dataPubblicazione "2005-01-31">
<!ELEMENT RisultatoRicerca (Stato, Documento*)>
```

```
<!ATTLIST RisultatoRicerca
  versione NMTOKEN #FIXED "%dataPubblicazione;"
  xml:lang NMTOKEN #FIXED "it"
>


<!--
********************** Stato ******************************************
* Rappresenta il codice di ritorno del servizio corredato di un eventuale
* messaggio di errore
* 0 = Servizio eseguito con successo
***************************************************************************
!-->
<!ELEMENT Stato (Codice, Messaggio)>
<!ELEMENT Codice (#PCDATA)>
<!ELEMENT Messaggio (#PCDATA)>

<!--
********************** Documento ***************************************
*** Questo elemento permette di restituire i risultati della ricerca  ***
***************************************************************************
!-->

<!ELEMENT EstremiReg (Sigla, Anno, Nro)>
<!ELEMENT Sigla (#PCDATA)>
<!ELEMENT Anno (#PCDATA)>
<!ELEMENT Nro (#PCDATA)>


<!ELEMENT Documento (IdDoc, NumCopia, RegPrimaria, DtReg, RegSecondaria, RegEmergenza,
DtArrivo, Oggetto,
            Mittenti, Destinatari, UoAssegnataria, IdIndice, DesIndice,
            AnnoFasc, Titolo, Classe, SottoClasse, Livello4, Livello5, ProgrFasc,
            IdFascicolo, NumSottofasc, OggettoFasc, DtAperturaFasc, DtChiusuraFasc,
            OggettoSottofasc, DtAperturaSottofasc, DtChiusuraSottofasc,
            NroAlleg, RifProv, ProtProv, DtProv, DocPrec)>
<!ATTLIST Documento
  TipoReg (In entrata |In uscita | Tra uffici)
  FlgPresoIncarico (S | N)
  FlgEvd (S | N)
  FlgRsv (S | N)
  FlgScartato (S | N)
  FlgRegAnnullata (S | N)
  FlgCopiaAnnullata (S | N)
>
<!ELEMENT IdDoc (#PCDATA)>
<!ELEMENT NumCopia (#PCDATA)>
<!ELEMENT RegPrimaria (EstremiReg)>
<!ELEMENT DtReg (#PCDATA)>
<!ELEMENT RegSecondaria (EstremiReg)>
<!ELEMENT RegEmergenza (EstremiReg)>
<!ELEMENT DtArrivo (#PCDATA)>
<!ELEMENT Oggetto (#PCDATA)>
<!ELEMENT Mittenti (#PCDATA)>
<!ELEMENT Destinatari (#PCDATA)>
<!ELEMENT UoAssegnataria (#PCDATA)>
<!ELEMENT IdIndice (#PCDATA)>
```

```
<!ELEMENT DesIndice (#PCDATA)>
<!ELEMENT AnnoFasc (#PCDATA)>
<!ELEMENT Titolo (#PCDATA)>
<!ELEMENT Classe (#PCDATA)>
<!ELEMENT SottoClasse (#PCDATA)>
<!ELEMENT Livello4 (#PCDATA)>
<!ELEMENT Livello5 (#PCDATA)>
<!ELEMENT ProgrFasc (#PCDATA)>
<!ELEMENT IdFascicolo (#PCDATA)>
<!ELEMENT NumSottofasc (#PCDATA)>
<!ELEMENT OggettoFasc (#PCDATA)>
<!ELEMENT DtAperturaFasc (#PCDATA)>
<!ELEMENT DtChiusuraFasc (#PCDATA)>
<!ELEMENT OggettoSottofasc (#PCDATA)>
<!ELEMENT DtAperturaSottofasc (#PCDATA)>
<!ELEMENT DtChiusuraSottofasc (#PCDATA)>
<!ELEMENT NroAlleg (#PCDATA)>
<!ELEMENT RifProv (#PCDATA)>
<!ELEMENT ProtProv (#PCDATA)>
<!ELEMENT DtProv (#PCDATA)>
<!ELEMENT DocPrec (EstremiReg)>
```

4.6 ***Ricerca Anagrafica.dtd***

```
<!--
********** Ricerca Anagrafica **********

La ricerca avviene in base ad una stringa rappresentata
dal tag Nominativo.
************************************************************
-->
<!ENTITY % dataPubblicazione "2004-01-19">
<!ELEMENT RicercaAnagrafica (Matricola | DatiAnag)>
<!ATTLIST RicercaAnagrafica
  versione NMTOKEN #FIXED "2004-01-19"
  xml:lang NMTOKEN #FIXED "it"
>
<!--
********** Parametri di ricerca**********
************************************************************
-->
<!ELEMENT DatiAnag (Codice?, Nominativo?)>
<!ELEMENT Codice (CodiceFiscale? | PartitaIva?)>
<!ELEMENT Nominativo (#PCDATA)>
<!ELEMENT Matricola (#PCDATA)>
<!ELEMENT CodiceFiscale (#PCDATA)>
<!ELEMENT PartitaIva (#PCDATA)>
```

4.7 ***RisultatoRicerca Anagrafica.dtd***

```
<!ENTITY % dataPubblicazione "2004-01-19">
<!ELEMENT RisultatoRicercaAnagrafica (Stato, Anagrafica*)>
<!ATTLIST RisultatoRicerca
  versione NMTOKEN #FIXED "%dataPubblicazione;"
  xml:lang NMTOKEN #FIXED "it"
>
<!--
```

```
********** Stato **********
* Rappresenta il codice di ritorno del servizio corredato di un eventuale
* messaggio di errore
* 0 = Servizio eseguito con successo
**************************************************************************
-->
<!ELEMENT Stato (Codice, Messaggio)>
<!ELEMENT Codice (#PCDATA)>
<!ELEMENT Messaggio (#PCDATA)>
<!--
********** Anagrafica **********
* Questo elemento permette di restituire i risultati prodotti dal servizio.
**************************************************************************
-->
<!ELEMENT Anagrafica (Nominativo, (CodiceFiscale | PartitaIva)?, DescrizioneResidenza?,
DataFineValidita?)>
<!ATTLIST Anagrafica
  codice NMTOKEN #REQUIRED
  matricola CDATA ""
  tipo (P | D) "P"
>
<!ELEMENT Nominativo (#PCDATA)>
<!ELEMENT CodiceFiscale (#PCDATA)>
<!ELEMENT PartitaIva (#PCDATA)>
<!ELEMENT DescrizioneResidenza (#PCDATA)>
<!ELEMENT DataFineValidita (#PCDATA)>
```

4.8 ***RicercaFascicoli.dtd***

```
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) -->
<!ENTITY % dataPubblicazione "2004-01-19">
<!--
**********Ricerca fascicoli**********
-->


<!ENTITY % dataPubblicazione "2005-01-31">
<!ELEMENT RicercaFascicolo (IdUoIn,OggettoIn?,Classif_FascAppIn?,DesTitolazioneIn?,
FlgStatoIn?,DtAperturaDaIn?,DtAperturaAIn?,DtChiusuraDaIn?,DtChiusuraAIn?,IdUOAperturaIn?,IdU
OPressoIn?,IdTpProcIn?,FlgAnnullamentoIn?,ChiavexRicercaIn?,
Classif_FascRifIn?,AttrOpzIn?,TpOrderByIn?,FlgDescOrderByIn?)>
<!ATTLIST RicercaFascicolo
  versione NMTOKEN #FIXED "2005-01-31"
  xml:lang NMTOKEN #FIXED "it"
>


<!ELEMENT UO (IdUO?, Liv1, Liv2, Liv3, Liv4, Postazione)>
<!ATTLIST UO
  FlgInclSottoUO (1 | 0) "0"
>

<!ELEMENT IdUO (#PCDATA)>
<!ELEMENT Liv1 (#PCDATA)>
<!ELEMENT Liv2 (#PCDATA)>
<!ELEMENT Liv3 (#PCDATA)>
<!ELEMENT Liv4 (#PCDATA)>
```

```
<!ELEMENT Postazione (#PCDATA)>


<!ELEMENT CLASSIFICAZIONE (Titolo, Classe?, SottoClasse?, Livello4?, Livello5?, IdTitolazione?)>
<!ELEMENT Titolo (#PCDATA)>
<!ELEMENT Classe (#PCDATA)>
<!ELEMENT SottoClasse (#PCDATA)>
<!ELEMENT Livello4 (#PCDATA)>
<!ELEMENT Livello5 (#PCDATA)>
<!ELEMENT IdTitolazione (#PCDATA)>


<!ELEMENT CLASSIF_FASC (AnnoFasc?, Classif, ProgrFasc?, NumSottofasc?, IdFascicolo?)>
<!ELEMENT AnnoFasc (#PCDATA)>
<!ELEMENT Classif (CLASSIFICAZIONE)>
<!ELEMENT ProgrFasc (#PCDATA)>
<!ELEMENT NumSottofasc (#PCDATA)>
<!ELEMENT IdFascicolo (#PCDATA)>


<!ELEMENT AttrOpz (Id_attributo, datatype, operatore, valore1,valore2?)>
<!ELEMENT Id_attributo (#PCDATA)>
<!ELEMENT datatype (#PCDATA)>
<!ELEMENT operatore (#PCDATA)>
<!ELEMENT valore1 (#PCDATA)>
<!ELEMENT valore2 (#PCDATA)>



<!ELEMENT IdUoIn (#PCDATA)>

<!--
********************** Filtri di ricerca*******************************
*** i filtri che iniziano per Dt vanno passati in formato DD/MM/YYYY *** nei filtri relativi a campi testo
(Oggetto, DesAlleg, NoteCopia) va messo il % per indicare qualsiasi
stringa**********************************************
-->
<!ELEMENT OggettoIn (#PCDATA)>
<!ELEMENT Classif_FascApp (CLASSIF_FASC)>
<!ELEMENT DesTitolazioneIn (#PCDATA)>

<!ELEMENT FlgStatoIn (#PCDATA)>
<!ELEMENT DtAperturaDaIn (#PCDATA)>
<!ELEMENT DtAperturaAIn (#PCDATA)>
<!ELEMENT DtChiusuraDaIn (#PCDATA)>
<!ELEMENT DtChiusuraAIn (#PCDATA)>
<!ELEMENT IdUOAperturaIn (UO)>
<!ELEMENT IdUOPressoIn (UO)>


<!ELEMENT IdTpProcIn (#PCDATA)>
<!ELEMENT FlgAnnullamentoIn (#PCDATA)>
<!ELEMENT ChiavexRicercaIn (#PCDATA)>
<!ELEMENT Classif_FascRifIn (CLASSIF_FASC)>

<!ELEMENT AttrOpzIn (AttrOpz)>
```

```
<!ELEMENT TpOrderByIn (#PCDATA)>

<!ELEMENT FlgDescOrderByIn (#PCDATA)>
```

4.9 RisultatoRicercaFasc.dtd

```
<!ENTITY % dataPubblicazione "2005-01-31">
<!ELEMENT RisultatoRicerca (Stato, Fascicolo*)>
<!ATTLIST RisultatoRicerca
  versione NMTOKEN #FIXED "%dataPubblicazione;"
  xml:lang NMTOKEN #FIXED "it"
>


<!--
********** Stato ********************************************
* Rappresenta il codice di ritorno del servizio corredato di un eventuale
* messaggio di errore
* 0 = Servizio eseguito con successo
*****************************************************************************
!-->
<!ELEMENT Stato (Codice, Messaggio)>
<!ELEMENT Codice (#PCDATA)>
<!ELEMENT Messaggio (#PCDATA)>

<!--
********** Fascicolo ***************************************
*** Questo elemento permette di restituire i risultati della ricerca  ***
*
*ID_FASCICOLO   Id. del fascicolo (sempre valorizzato)
*ANNO_FASC  Anno del fascicolo
*NUM_FASC   Numero del Fascicolo
*NUM_SOTTOFASC  N.ro di sottofascicolo (sempre valorizzato: per un fascicolo è 0)
*TXT_OGG oggetto del fascicolo
*DT_APERTURAdata di apertura del fascicolo
*DT_CHIUSURA data di chiusura del fascicolo
*TITOLO_CLASS    TITOLO DELLA CLASSIFICA DEL FASCICOLO
*CLASSE_CLASS    classe DELLA CLASSIFICA DEL FASCICOLO
*SOTTOCLASSE_CLASS sottoclasse DELLA CLASSIFICA DEL FASCICOLO
*LIVELLO4_CLASS livello4 DELLA CLASSIFICA DEL FASCICOLO
*LIVELLO5_CLASS livello5 DELLA CLASSIFICA DEL FASCICOLO
*CODICE  è il codice con cui va mostrato il fascicolo: <anno>.<livelli classif.>.<n.ro progr.>[/<n.ro
sottofasc> se >0 ]
*DES_TITOLAZIONE      descrizione della classificazione del fascicolo
*DT_APERTURAdata di apertura del fascicolo
*DT_CHIUSURA data di chiusura del fascicolo
*DT_ARCH data di archiviazione
*FLG_ANN Flag di annullamento. Valori: S/N
*DT_VERSAMENTO  Data di versamento in archivio storico
*ID_FASCICOLO_RIF     Id. del fascicolo di riferimento
*NUM_SOTTOFASC_RIF N.ro di sottofascicolo di riferimento
*DEC_FASC_RIF         Codice e data di apertura del fascicolo/sottofasc. di riferimento
*MOTIVI_RIF           Motivi del collegamento al fascicolo di riferimento
*PAROLE_CHIAVE        Lista di parole chiave del fascicolo
*****************************************************************************
!-->
```

```
<!ELEMENT
Fascicolo(ID_FASCICOLO,NUM_SOTTOFASC,CODICE,DES_TITOLAZIONE,DT_APERTURA,DT_C
HIUSURA,


DT_ARCH,TXT_OGG,FLG_ANN,DT_VERSAMENTO,ID_FASCICOLO_RIF,NUM_SOTTOFASC_RIF,
DEC_FASC_RIF,


MOTIVI_RIF,PAROLE_CHIAVE,ANNO_FASC,NUM_FASC,TITOLO_CLASS,CLASSE_CLASS,SOTT
OCLASSE_CLASS,LIVELLO4_CLASS,LIVELLO5_CLASS)>

<!ELEMENT ID_FASCICOLO (#PCDATA)>
<!ELEMENT NUM_SOTTOFASC (#PCDATA)>
<!ELEMENT CODICE (#PCDAT)>
<!ELEMENT DES_TITOLAZIONE (#PCDATA)>
<!ELEMENT DT_APERTURA (#PCDATA)>
<!ELEMENT DT_CHIUSURA (#PCDATA)>
<!ELEMENT DT_ARCH (#PCDATA)>
<!ELEMENT TXT_OGG (#PCDATA)>
<!ELEMENT FLG_ANN (#PCDATA)>
<!ELEMENT DT_VERSAMENTO (#PCDATA)>
<!ELEMENT ID_FASCICOLO_RIF (#PCDATA)>
<!ELEMENT NUM_SOTTOFASC_RIF (#PCDATA)>
<!ELEMENT DEC_FASC_RIF (#PCDATA)>
<!ELEMENT MOTIVI_RIF (#PCDATA)>
<!ELEMENT PAROLE_CHIAVE (#PCDATA)>
<!ELEMENT ANNO_FASC (#PCDATA)>
<!ELEMENT NUM_FASC (#PCDATA)>
<!ELEMENT TITOLO_CLASS (#PCDATA)>
<!ELEMENT CLASSE_CLASS (#PCDATA)>
<!ELEMENT SOTTOCLASSE_CLASS (#PCDATA)>
<!ELEMENT LIVELLO4_CLASS (#PCDATA)>
<!ELEMENT LIVELLO5_CLASS (#PCDATA)>
```

4.10 ***RicercaUO.dtd***

```
<!ENTITY % dataPubblicazione "2007-06-20">
<!ELEMENT RicercaUO (IdPostazLavoro, ((UserName?),(Matricola?))   )>
<!ATTLIST RicercaUO
  versione NMTOKEN #FIXED "2007-06-20"
  xml:lang NMTOKEN #FIXED "it"
>
<!ELEMENT IdPostazLavoro (#PCDATA)>
<!ELEMENT UserName (#PCDATA)>
<!ELEMENT Matricola (#PCDATA)>
```

4.11 ***RispostaRicercaUO.dtd***

```
<!--
<!ENTITY % dataPubblicazione "2007-06-20">
<!ELEMENT RispostaRicercaUO (IdPostazLavoro, ((UserName?),(Matricola?))   )>
<!ATTLIST RicercaUO
  versione NMTOKEN #FIXED "2007-06-20"
```

```
  xml:lang NMTOKEN #FIXED "it"

<!ELEMENT Stato (Codice, Messaggio)>
<!ELEMENT Codice (#PCDATA)>
<!ELEMENT Messaggio (#PCDATA)>

<!ELEMENT IdUte (#PCDATA)>
<!ELEMENT DesUte (#PCDATA)>
<!ELEMENT IdUO (#PCDATA)>
<!ELEMENT Liv1 (#PCDATA)>
<!ELEMENT Liv2 (#PCDATA)>
<!ELEMENT Liv3 (#PCDATA)>
<!ELEMENT Liv4 (#PCDATA)>
<!ELEMENT Postazione (#PCDATA)>
<!ELEMENT IdUO_P (#PCDATA)>
<!ELEMENT Liv1_P (#PCDATA)>
<!ELEMENT Liv2_P (#PCDATA)>
<!ELEMENT Liv3_P (#PCDATA)>
<!ELEMENT Liv4_P (#PCDATA)>
<!ELEMENT Postazione_P (#PCDATA)>
<!ELEMENT DesUO_P (#PCDATA)>
<!ELEMENT FlgAbil (#PCDATA)>
```

4.12 ***ModificaDoc.xsd***

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- edited with XMLSPY v5 U (http://www.xmlspy.com) by  () -->
<!-- edited with XMLSpy v2006 sp1 U (http://www.altova.com) by Admin (EMBRACE) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attrib-
uteFormDefault="unqualified">
  <xs:complexType name="LivelliUOType">
          <xs:annotation>
                  <xs:documentation>Livelli di una UO</xs:documentation>
          </xs:annotation>
          <xs:attribute name="Nro" use="required">
                  <xs:annotation>
                          <xs:documentation>1 è il livello più alto, 5 è una
postazione</xs:documentation>
                  </xs:annotation>
                  <xs:simpleType>
                          <xs:restriction base="xs:integer">
                                  <xs:minInclusive value="1"/>
                                  <xs:maxInclusive value="5"/>
                          </xs:restriction>
                  </xs:simpleType>
          </xs:attribute>
          <xs:attribute name="Codice" use="required">
                  <xs:annotation>
                          <xs:documentation>Identifica il livello  corrispondente in seno alla pianta
organica</xs:documentation>
                  </xs:annotation>
                  <xs:simpleType>
                          <xs:restriction base="xs:integer">
                                  <xs:minInclusive value="0"/>
                                  <xs:maxInclusive value="9999"/>
                          </xs:restriction>
                  </xs:simpleType>
```

```
            </xs:attribute>
    </xs:complexType>
    <xs:complexType name="UOType">
            <xs:annotation>
                    <xs:documentation>Identificativo di una postazione o i suoi 5
livelli</xs:documentation>
            </xs:annotation>
            <xs:choice>
                    <xs:element name="IdUo">
                            <xs:simpleType>
                                    <xs:restriction base="xs:integer">
                                            <xs:minInclusive value="1"/>
                                            <xs:maxInclusive value="99999999"/>
                                    </xs:restriction>
                            </xs:simpleType>
                    </xs:element>
                    <xs:element name="Livelli" type="LivelliUOType" maxOccurs="5"/>
            </xs:choice>
    </xs:complexType>
    <xs:complexType name="SiglaType">
            <xs:annotation>
                    <xs:documentation>Tipo, anno e numero del documento</xs:documentation>
            </xs:annotation>
            <xs:sequence>
                    <xs:element name="Tipo">
                            <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                            <xs:maxLength value="5"/>
                                    </xs:restriction>
                            </xs:simpleType>
                    </xs:element>
                    <xs:element name="Anno">
                            <xs:simpleType>
                                    <xs:restriction base="xs:integer">
                                            <xs:minInclusive value="1900"/>
                                            <xs:maxInclusive value="2100"/>
                                    </xs:restriction>
                            </xs:simpleType>
                    </xs:element>
                    <xs:element name="Numero">
                            <xs:simpleType>
                                    <xs:restriction base="xs:integer">
                                            <xs:minInclusive value="1"/>
                                            <xs:maxInclusive value="9999999"/>
                                    </xs:restriction>
                            </xs:simpleType>
                    </xs:element>
            </xs:sequence>
    </xs:complexType>
    <xs:complexType name="RegType">
            <xs:annotation>
                    <xs:documentation>Id del documento oppure sigla, anno e
numero</xs:documentation>
            </xs:annotation>
            <xs:choice>
                    <xs:element name="IdDocumento">
                            <xs:simpleType>
```

```
                                        <xs:restriction base="xs:integer">
                                                <xs:minInclusive value="1"/>
                                                <xs:maxInclusive value="99999999"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="SiglaType" type="SiglaType"/>
                </xs:choice>
        </xs:complexType>
        <xs:complexType name="OggDiTabDiSistemaType">
                <xs:annotation>
                        <xs:documentation>Rappresenta un oggetto censito in una tabella di sistema del
sistema di protocollo</xs:documentation>
                </xs:annotation>
                <xs:choice>
                        <xs:element name="CodId">
                                <xs:annotation>
                                        <xs:documentation>Identificativo del dato di sistema nella banca dati
del sistema di Protocollo</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Decodifica_Nome">
                                <xs:annotation>
                                        <xs:documentation>Decodifica del dato di sistema nella banca dati
del sistema di Protocollo</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:choice>
        </xs:complexType>
        <xs:complexType name="ComuneItalianoType">
                <xs:annotation>
                        <xs:documentation>Codice Istat o Denominazione del comune</xs:documentation>
                </xs:annotation>
                <xs:choice>
                        <xs:element name="CodISTAT">
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:pattern value="[0-9]{6}"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="Denominazione" type="xs:string"/>
                </xs:choice>
        </xs:complexType>
        <xs:complexType name="ToponimoType">
                <xs:annotation>
                        <xs:documentation>Estremi della via</xs:documentation>
                </xs:annotation>
                <xs:choice>
                        <xs:element name="IdToponomastico" type="xs:string">
                                <xs:annotation>
                                        <xs:documentation>Identificato nel viario comunale (se trattasi di
inidirizzo nel territorio comunale)</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="DesToponimo" type="xs:string">
                                <xs:annotation>
```

```
                                        <xs:documentation>Descrizione (incluso il tipo: Via, Vicolo, Piazza
ecc)</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:choice>
        </xs:complexType>
        <xs:complexType name="IndirizzoSoggType">
                <xs:annotation>
                        <xs:documentation>Indirizzo di un soggetto (sede legale, residenza
ecc.)</xs:documentation>
                </xs:annotation>
                <xs:all>
                        <xs:element name="IndirizzoLocalita" type="ToponimoType"/>
                        <xs:element name="Civico" minOccurs="0">
                                <xs:simpleType>
                                        <xs:restriction base="xs:integer">
                                                <xs:minInclusive value="1"/>
                                                <xs:maxInclusive value="99999"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="EsponenteCivico" minOccurs="0">
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:maxLength value="3"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="Interno" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>N.ro dell'interno del civico</xs:documentation>
                                </xs:annotation>
                                <xs:simpleType>
                                        <xs:restriction base="xs:integer">
                                                <xs:maxInclusive value="99999"/>
                                                <xs:minInclusive value="1"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="EsponenteInterno" minOccurs="0">
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:maxLength value="1"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="Scala" minOccurs="0">
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:maxLength value="1"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="Piano" minOccurs="0">
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:maxLength value="2"/>
```

```
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="CAP" minOccurs="0">
                    <xs:simpleType>
                        <xs:restriction base="xs:integer">
                            <xs:minInclusive value="1"/>
                            <xs:maxInclusive value="99999"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="Frazione" minOccurs="0">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="25"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="ComuneCitta">
                    <xs:complexType>
                        <xs:choice>
                            <xs:element name="Comune" type="ComuneItalianoType"/>
                            <xs:sequence>
                                <xs:element name="StatoEstero" type="OggDiTab-
DiSistemaType"/>
                            </xs:sequence>
                        </xs:choice>
                    </xs:complexType>
                </xs:element>
            </xs:all>
        </xs:complexType>
        <xs:complexType name="IndirizzoType">
            <xs:annotation>
                <xs:documentation>Indirizzo del mittente/destinario</xs:documentation>
            </xs:annotation>
        </xs:complexType>
        <xs:complexType name="SoggettoEsternoType">
            <xs:annotation>
                <xs:documentation>Soggetto (persona fisica o giuridica) esterno
all'AOO</xs:documentation>
            </xs:annotation>
            <xs:sequence>
                <xs:element name="IdInAnagrafeProt" type="xs:positiveInteger" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>Codice identificativo del soggetto nell'anagrafe
soggetti del Protocollo.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="FlagFisica">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:length value="1"/>
                            <xs:pattern value="P|D"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="Denominazione_Cognome">
```

```
                        <xs:annotation>
                                <xs:documentation>Denominazione (se persona giuridica) o
cognome (se persona fisica)</xs:documentation>
                        </xs:annotation>
                        <xs:simpleType>
                                <xs:restriction base="xs:string">
                                        <xs:maxLength value="150"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
                <xs:element name="Nome" minOccurs="0">
                        <xs:annotation>
                                <xs:documentation>Nome (se valorizzato si tratta di persona
fisica)</xs:documentation>
                        </xs:annotation>
                        <xs:simpleType>
                                <xs:restriction base="xs:string">
                                        <xs:maxLength value="40"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
                <xs:element name="CodiceFiscale" minOccurs="0">
                        <xs:simpleType>
                                <xs:restriction base="xs:string">
                                        <xs:maxLength value="16"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
                <xs:element name="PartitaIva" minOccurs="0">
                        <xs:simpleType>
                                <xs:restriction base="xs:string">
                                        <xs:maxLength value="11"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
                <xs:element name="Sesso" minOccurs="0">
                        <xs:simpleType>
                                <xs:restriction base="xs:string">
                                        <xs:pattern value="M|F"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
                <xs:element name="DataNascitaCostituzione" type="xs:date" minOccurs="0">
                        <xs:annotation>
                                <xs:documentation>Data di nascita /
costituzione</xs:documentation>
                        </xs:annotation>
                </xs:element>
                <xs:element name="ComuneNascita" type="ComuneItalianoType" minOccurs="0"/>
                <xs:element name="StatoCittadinanza" type="OggDiTabDiSistemaType"
minOccurs="0"/>
                <xs:element name="ResidenzaSedeLegale" type="IndirizzoSoggType"
minOccurs="0"/>
                <xs:element name="Recapito" type="IndirizzoSoggType" minOccurs="0"/>
                <xs:element name="PerConoscenza" fixed="1" minOccurs="0">
                        <xs:annotation>
```

```
                                        <xs:documentation>Indica che il destinatario è solo per
conoscenza</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="MittDestEsternoType">
                <xs:annotation>
                        <xs:documentation>Dati relativi al Mittente\Destinatario esterno</xs:documentation>
                </xs:annotation>
                <xs:complexContent>
                        <xs:extension base="SoggettoEsternoType"/>
                </xs:complexContent>
        </xs:complexType>
        <xs:complexType name="VersioneElettronicaType">
                <xs:annotation>
                        <xs:documentation>Contiene le informazioni di come trattare/salvare i file in attach
in input al Web Service</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="NroAttachmentAssociato" type="xs:integer">
                                <xs:annotation>
                                        <xs:documentation>Indica quali degli attach in input al Web Service
rappresenta la versione elettronica del documento primario o di un allegato (1 è il primo attach, 2 il
secondo ecc)</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="NomeFile" type="xs:string">
                                <xs:annotation>
                                        <xs:documentation>Nome del file: si tenga conto che l'estensione
del nome del file viene utilizzata per ricavare il formato (in particolare solo se questa è pari a p7m si
ritiene un file firmato digitalmente)</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="AttivaVerificaFirma" fixed="1" minOccurs="0"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="AllegatoType">
                <xs:annotation>
                        <xs:documentation>Contiene i dati di un allegato</xs:documentation>
                </xs:annotation>
                <xs:sequence>
                        <xs:element name="TipoDocAllegato" type="OggDiTabDiSistemaType">
                                <xs:annotation>
                                        <xs:documentation>Tipo dell'allegato (possibile solo uno dei valori in
apposita tabella PTT_TPALLEG)</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="DesAllegato" minOccurs="0">
                                <xs:annotation>
                                        <xs:documentation>Descrizione / oggetto
dell'allegato</xs:documentation>
                                </xs:annotation>
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:maxLength value="250"/>
                                        </xs:restriction>
```

```
                        </xs:simpleType>
                    </xs:element>
                    <xs:element name="VersioneElettronica" type="VersioneElettronicaType"
minOccurs="0">
                        <xs:annotation>
                            <xs:documentation>Contiene le informazioni su come
trattare/salvare l'eventuale attach in input al Web Service che corrispinde all'allegato (nome con cui
salvarlo, se la firma eventuale va verificata)</xs:documentation>
                        </xs:annotation>
                    </xs:element>
            </xs:sequence>
    </xs:complexType>
    <xs:element name="Dati">
            <xs:annotation>
                    <xs:documentation>ROOT element dello schema</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                    <xs:sequence>
                        <xs:element name="PostLavoro" type="UOType">
                            <xs:annotation>
                                <xs:documentation>Identificativo (id_uo o livelli della pianta
organica) della postazione che sta eseguendo l'operazione</xs:documentation>
                            </xs:annotation>
                        </xs:element>
                        <xs:element name="RegMod" type="RegType">
                            <xs:annotation>
                                <xs:documentation>Identificativo (id_doc o tipo, anno e
numero) del documento da modificare</xs:documentation>
                            </xs:annotation>
                        </xs:element>
                        <xs:element name="MotivazioniAnn" minOccurs="0">
                            <xs:annotation>
                                <xs:documentation>Motivazioni dell'annullamento dati (da
valorizzare solo se modificati dati soggetti ad annullamento)</xs:documentation>
                            </xs:annotation>
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:maxLength value="250"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                        <xs:element name="DtArrivo" type="xs:dateTime" minOccurs="0">
                            <xs:annotation>
                                <xs:documentation>Data e ora di arrivo (dato modificabile
solo se registrazione in entrata)  Dato soggetto ad annullamento</xs:documentation>
                            </xs:annotation>
                        </xs:element>
                        <xs:element name="Oggetto" minOccurs="0">
                            <xs:annotation>
                                <xs:documentation>Oggetto del documento       Dato
soggetto ad annullamento</xs:documentation>
                            </xs:annotation>
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:maxLength value="500"/>
                                </xs:restriction>
                            </xs:simpleType>
```

```
                    </xs:element>
                    <xs:element name="EstremiRegEmerg" type="SiglaType" minOccurs="0">
                        <xs:annotation>
                            <xs:documentation>Estremi (tipo, anno e numero) del
registro di emergenza di un eventuale registrazione di emergenza (modificabile solo se la
registrazione non è fatta solo a sportello)  Dato soggetto ad
annullamento</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element name="AttoAutProtDifferita" type="SiglaType" minOccurs="0">
                        <xs:annotation>
                            <xs:documentation>Estremi dell'atto di autorizzazione alla
prot. differita (può essere modificato solo se registrazione in entrata e non fatta solo a sportello)
Dato soggetto ad annullamento</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element name="DtProv" type="xs:dateTime" minOccurs="0">
                        <xs:annotation>
                            <xs:documentation>Data di protocollo del documento
ricevuto (dato modificabile solo se registrazione in entrata)  Dato soggetto ad
annullamento</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element name="RifProv" minOccurs="0">
                        <xs:annotation>
                            <xs:documentation>Riferimenti del documento ricevuto
(dato modificabile solo se registrazione in entrata)  Dato soggetto ad
annullamento</xs:documentation>
                        </xs:annotation>
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:maxLength value="12"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:element>
                    <xs:element name="ProtProv" minOccurs="0">
                        <xs:annotation>
                            <xs:documentation>Estremi di protocollo del documento
ricevuto (dato modificabile solo se registrazione in entrata)  Dato soggetto ad annullamento
</xs:documentation>
                        </xs:annotation>
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:maxLength value="20"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:element>
                    <xs:element name="IdUOProv" type="UOType" minOccurs="0">
                        <xs:annotation>
                            <xs:documentation>Identificativo della U.O. mittente(id uo o
i 5 livelli della pianta organica) (dato modificabile solo se registrazione in uscita o tra uffici)
Dato soggetto ad annullamento  </xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element name="MittDestEsterni" type="MittDestEsternoType"
minOccurs="0" maxOccurs="unbounded">
                        <xs:annotation>
```

```
                                        <xs:documentation>Array con i dati dei mittenti (se
registrazione in entrata) o destinatari (se registrazione in uscita o tra uffici)          Dato soggetto ad
annullamento  </xs:documentation>
                                        </xs:annotation>
                                </xs:element>
                                <xs:element name="FlagMDEAppSost" minOccurs="0">
                                        <xs:annotation>
                                                <xs:documentation>Flag che informa se i mitt/desti sono da
appendere o sostituire a quelli esistenti</xs:documentation>
                                        </xs:annotation>
                                        <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                        <xs:length value="1"/>
                                                </xs:restriction>
                                        </xs:simpleType>
                                </xs:element>
                                <xs:element name="TpFisico" type="OggDiTabDiSistemaType" minOc-
curs="0">
                                        <xs:annotation>
                                                <xs:documentation>Mezzo di trasmissione (tabella ptt_tpfis)
Dato NON soggetto ad annullamento  </xs:documentation>
                                        </xs:annotation>
                                </xs:element>
                                <xs:element name="DtRaccomandata" type="xs:date" minOccurs="0">
                                        <xs:annotation>
                                                <xs:documentation>Data di raccomandata        Dato NON
soggetto ad annullamento  </xs:documentation>
                                        </xs:annotation>
                                </xs:element>
                                <xs:element name="NroRaccomandata" minOccurs="0">
                                        <xs:annotation>
                                                <xs:documentation>Numero di raccomandata (alfanumerico)
Dato NON soggetto ad annullamento  </xs:documentation>
                                        </xs:annotation>
                                        <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                        <xs:maxLength value="30"/>
                                                </xs:restriction>
                                        </xs:simpleType>
                                </xs:element>
                                <xs:element name="TipoLogico" type="OggDiTabDiSistemaType" minOc-
curs="0">
                                        <xs:annotation>
                                                <xs:documentation>tipo di documento (tabella ptt_tplog)
Dato NON soggetto ad annullamento  </xs:documentation>
                                        </xs:annotation>
                                </xs:element>
                                <xs:element name="SottoTipoLogico" type="OggDiTabDiSistemaType"
minOccurs="0">
                                        <xs:annotation>
                                                <xs:documentation>Sottotipo di documento (tabella
ptt_tplog). Considerato solo se TipoLogico è valorizzato        Dato NON soggetto ad annullamento  </
xs:documentation>
                                        </xs:annotation>
                                </xs:element>
                                <xs:element name="FlgTpRegDocPrec" minOccurs="0">
                                        <xs:annotation>
```

```
                                        <xs:documentation>Indica il tipo di registrazione del doc.
precedente; valori: P=Prot. Gen., R=Altro        Dato NON soggetto ad annullamento
</xs:documentation>
                                    </xs:annotation>
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:enumeration value="P"/>
                                            <xs:enumeration value="R"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                                <xs:element name="EstremiDocPrec" type="SiglaType" minOccurs="0">
                                    <xs:annotation>
                                        <xs:documentation>Estremi del documento precedente.
Dato NON soggetto ad annullamento       </xs:documentation>
                                    </xs:annotation>
                                </xs:element>
                                <xs:element name="FlgDocNoPubbl" minOccurs="0">
                                    <xs:annotation>
                                        <xs:documentation>Flag di riservatezza al pubblico (con
valori S/N): Se S il doc. non è visibile al pubblico, se N sì                Dato NON soggetto ad
annullamento  </xs:documentation>
                                    </xs:annotation>
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:length value="1"/>
                                            <xs:pattern value="S|N"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                                <xs:element name="DtTermNoPubbl" type="xs:date" minOccurs="0">
                                    <xs:annotation>
                                        <xs:documentation>Data di termine della riservatezza al
pubblico        Dato NON soggetto ad annullamento  </xs:documentation>
                                    </xs:annotation>
                                </xs:element>
                                <xs:element name="FlgDocRsv" minOccurs="0">
                                    <xs:annotation>
                                        <xs:documentation>Flag di riservatezza (con valori S/N): Se
S il doc. è riservato, se N non è riservato                    Dato NON soggetto ad annullamento
</xs:documentation>
                                    </xs:annotation>
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:length value="1"/>
                                            <xs:pattern value="S|N"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                                <xs:element name="DtTermRsvPP" type="xs:date" minOccurs="0">
                                    <xs:annotation>
                                        <xs:documentation>Data di termine della riservatezza data
del protocollo particolare          Dato NON soggetto ad annullamento  </xs:documentation>
                                    </xs:annotation>
                                </xs:element>
                                <xs:element name="FlgEvidenza" minOccurs="0">
                                    <xs:annotation>
```

```
                                        <xs:documentation>Flag di evidenza (con valori S/N): Se S il
doc. è in evidenza, se N non lo è          Dato NON soggetto ad annullamento
  </xs:documentation>
                                    </xs:annotation>
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:length value="1"/>
                                            <xs:pattern value="S|N"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                                <xs:element name="Note" minOccurs="0">
                                    <xs:annotation>
                                        <xs:documentation>Note della registrazione          Dato NON
soggetto ad annullamento   </xs:documentation>
                                    </xs:annotation>
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:maxLength value="250"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                                <xs:element name="TpRepDef" minOccurs="0">
                                    <xs:annotation>
                                        <xs:documentation>Se la registrazione è un repertorio
provvisorio, questo campo contiene la sigla del repertorio definitivo in cui sarà convertito il documento
Dato NON soggetto ad annullamento  </xs:documentation>
                                    </xs:annotation>
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:maxLength value="5"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                                <xs:element name="DocPrimario" type="VersioneElettronicaType"
minOccurs="0">
                                    <xs:annotation>
                                        <xs:documentation>Dato del documento primario Dato
soggetto ad annullamento </xs:documentation>
                                    </xs:annotation>
                                </xs:element>
                                <xs:element name="Allegati" type="AllegatiType" minOccurs="0" maxOc-
curs="unbounded">
                                    <xs:annotation>
                                        <xs:documentation>Dati di un allegato                Dato NON
soggetto ad annullamento </xs:documentation>
                                    </xs:annotation>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:complexType name="AllegatiType">
                        <xs:annotation>
                            <xs:documentation>Num Allegato e metadati relativi all'allegato</xs:documentation>
                        </xs:annotation>
                        <xs:sequence>
                            <xs:element name="NumAlleg" minOccurs="0">
```

```
                        <xs:annotation>
                                <xs:documentation>Numero dell'allegato della registrazione, è
obbligatorio solo se non si vuole specificare l'allegato da modificare, senza intervenire sui
restanti</xs:documentation>
                        </xs:annotation>
                </xs:element>
                <xs:element name="Allegato" type="AllegatoType"/>
        </xs:sequence>
  </xs:complexType>
</xs:schema>
```

4.13 CreaCopia.xsd

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:complexType name="LivelliUOType">
        <xs:annotation>
                <xs:documentation>Livelli di una UO</xs:documentation>
        </xs:annotation>
        <xs:attribute name="Nro" use="required">
                <xs:annotation>
                        <xs:documentation>1 è il livello più alto, 5 è una
postazione</xs:documentation>
                </xs:annotation>
                <xs:simpleType>
                        <xs:restriction base="xs:integer">
                                <xs:maxInclusive value="5"/>
                                <xs:minInclusive value="1"/>
                        </xs:restriction>
                </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="Codice" use="required">
                <xs:annotation>
                        <xs:documentation>Identifica il livello  corrispondente in seno alla pianta
organica</xs:documentation>
                </xs:annotation>
                <xs:simpleType>
                        <xs:restriction base="xs:integer">
                                <xs:minInclusive value="0"/>
                                <xs:maxInclusive value="9999"/>
                        </xs:restriction>
                </xs:simpleType>
        </xs:attribute>
  </xs:complexType>
  <xs:complexType name="UOType">
        <xs:annotation>
                <xs:documentation>Identificativo di una postazione o i suoi 5
livelli</xs:documentation>
        </xs:annotation>
        <xs:choice>
                <xs:element name="IdUo">
                        <xs:simpleType>
                                <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="1"/>
                                        <xs:maxInclusive value="99999999"/>
                                </xs:restriction>
                        </xs:simpleType>
```

```
                </xs:element>
                <xs:element name="Livelli" type="LivelliUOType" maxOccurs="5"/>
        </xs:choice>
</xs:complexType>

        <xs:complexType name="LivelliUOAssType">
        <xs:annotation>
                <xs:documentation>Livelli di una UO per TAG UOAss</xs:documentation>
        </xs:annotation>
        <xs:sequence>
                <xs:element name="Settore" minOccurs="1" maxOccurs="1">
                        <xs:simpleType>
                                <xs:restriction base="xs:integer" >
                                        <xs:minInclusive value="00"/>
                                        <xs:maxInclusive value="9999"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
                <xs:element name="Servizio" minOccurs="1" maxOccurs="1">
                        <xs:simpleType>
                                <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="00"/>
                                        <xs:maxInclusive value="9999"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
                <xs:element name="Uoc" minOccurs="1" maxOccurs="1">
                        <xs:simpleType>
                                <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="00"/>
                                        <xs:maxInclusive value="9999"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
                <xs:element name="Uos"  minOccurs="1" maxOccurs="1">
                        <xs:simpleType>
                                <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="00"/>
                                        <xs:maxInclusive value="9999"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
                <xs:element name="Postazione" minOccurs="1" maxOccurs="1">
                        <xs:simpleType>
                                <xs:restriction base="xs:integer" >
                                        <xs:minInclusive value="00"/>
                                        <xs:maxInclusive value="9999"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
        </xs:sequence>
</xs:complexType>


<xs:complexType name="UOAssType">
        <xs:annotation>
```

```
                <xs:documentation>Identificativo di una postazione o i suoi 5
livelli</xs:documentation>
            </xs:annotation>
            <xs:choice>
                <xs:element name="IdUo">
                    <xs:simpleType>
                        <xs:restriction base="xs:integer">
                            <xs:minInclusive value="1"/>
                            <xs:maxInclusive value="99999999"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="Livelli" type="LivelliUOAssType" maxOccurs="1"/>
            </xs:choice>
    </xs:complexType>

    <xs:complexType name="SiglaType">
            <xs:annotation>
                <xs:documentation>Tipo, anno e numero del documento</xs:documentation>
            </xs:annotation>
            <xs:sequence>
                <xs:element name="Tipo">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="5"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="Anno">
                    <xs:simpleType>
                        <xs:restriction base="xs:integer">
                            <xs:minInclusive value="1900"/>
                            <xs:maxInclusive value="2100"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="Numero">
                    <xs:simpleType>
                        <xs:restriction base="xs:integer">
                            <xs:minInclusive value="1"/>
                            <xs:maxInclusive value="9999999"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
            </xs:sequence>
    </xs:complexType>


    <xs:complexType name="FascicoloDettType">
            <xs:annotation>
                <xs:documentation>Identificativo, anno, numero e sottofascicolo del
documento</xs:documentation>
            </xs:annotation>
            <xs:sequence>
                <xs:element name="Anno"  minOccurs="1" maxOccurs="1">
                    <xs:simpleType>
                        <xs:restriction base="xs:integer">
```

```
                                        <xs:minInclusive value="1900"/>
                                        <xs:maxInclusive value="2100"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
                <xs:element name="Numero" minOccurs="1" maxOccurs="1">
                        <xs:simpleType>
                                <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="1"/>
                                        <xs:maxInclusive value="9999999"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
        </xs:sequence>
  </xs:complexType>

  <xs:complexType name="FascicoloType">
        <xs:annotation>
                <xs:documentation>Identificativo, anno, numero e sottofascicolo del
documento</xs:documentation>
        </xs:annotation>
        <xs:sequence>
                <xs:choice>
                        <xs:element name="IdFascicolo">
                                <xs:simpleType>
                                        <xs:restriction base="xs:integer">
                                                <xs:minInclusive value="1"/>
                                                <xs:maxInclusive value="9999999"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="FascicoloDett" type="FascicoloDettType" maxOc-
curs="1"/>
                </xs:choice>
                <xs:element name="SottoFascicolo" minOccurs="0">
                        <xs:simpleType>
                                <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="0"/>
                                        <xs:maxInclusive value="9999"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
        </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RegType">
        <xs:annotation>
                <xs:documentation>Id del documento oppure sigla, anno e
numero</xs:documentation>
        </xs:annotation>
        <xs:choice>
                <xs:element name="IdDocumento">
                        <xs:simpleType>
                                <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="1"/>
                                        <xs:maxInclusive value="99999999"/>
                                </xs:restriction>
                        </xs:simpleType>
```

```
                </xs:element>
                <xs:element name="SiglaType" type="SiglaType" minOccurs="0"/>
                <xs:element name="Copia" default="0">
                        <xs:simpleType>
                                <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="0"/>
                                        <xs:maxInclusive value="9999999"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
        </xs:choice>
</xs:complexType>
<xs:complexType name="CopiaType">
        <xs:annotation>
                <xs:documentation>Identificativo assegantrio della copia</xs:documentation>
        </xs:annotation>
        <xs:sequence>
                <xs:element name="UOAss" type="UOAssType"/>
                <xs:element name="IdInd" minOccurs="0">
                        <xs:simpleType>
                                <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="1"/>
                                        <xs:maxInclusive value="9999999"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
                <xs:element name="FlgCopiaConoscenza" minOccurs="0">
                        <xs:simpleType>
                                <xs:restriction base="xs:string">
                                        <xs:maxLength value="1"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
                <xs:element name="Fascicolo" type="FascicoloType" minOccurs="0"/>
                <xs:element name="NoteInvio" minOccurs="0">
                        <xs:simpleType>
                                <xs:restriction base="xs:string">
                                        <xs:maxLength value="1000"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:element>
        </xs:sequence>
</xs:complexType>
<xs:element name="Dati">
        <xs:annotation>
                <xs:documentation>ROOT element dello schema</xs:documentation>
        </xs:annotation>
        <xs:complexType>
                <xs:sequence>
                        <xs:element name="PostLavoro" type="UOType">
                                <xs:annotation>
                                        <xs:documentation>Identificativo (id_uo o livelli della pianta
organica) della postazione che sta eseguendo l'operazione</xs:documentation>
                                </xs:annotation>
                        </xs:element>
                        <xs:element name="Documento" type="RegType">
                                <xs:annotation>
```

```
                                        <xs:documentation>Identificativo (id_doc o tipo, anno e
numero) del documento da modificare</xs:documentation>
                                    </xs:annotation>
                                </xs:element>
                                <xs:element name="Copia" type="CopiaType" minOccurs="1"
maxOccurs="unbounded">
                                    <xs:annotation>
                                        <xs:documentation>Elenco delle copie da creare
</xs:documentation>
                                    </xs:annotation>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:schema>
```

5 Appendice B: Codici di errore

Elenco dei codici di errore:

Codice	Significato
0	Nessun errore
Valori <>0	Errore da decodificare nel messaggio restituito dal WS

Allegato 11: Specifiche tecniche per l'utilizzo dei web services del sistema di gestione documentale

1 Introduzione

Questo documento ha lo scopo di descrivere i WS e relativi metodi messi a disposizione dalla piattaforma Hummingbird per permettere l'integrazione SOA con gli applicativi eterogenei regionali. Il servizio, denominato DMPantarei, e i suoi metodi, sono disponibili sulla intranet regionale all'indirizzo: http://sin28srv/pantarei/dmpantarei.asmx.

In collaudo DMPantarei è disponibile all'indirizzo:
http://sin25srv/pantarei/dmpantarei.asmx.

Sarà cura dell'applicazione chiamante accedere al servizio e relativi metodi.

L'interfaccia del Web Services "DMPantarei" è completamente specificata dal file WSDL.

2 Il web service DMPantarei e i suoi metodi

I metodi elencati nel presente documento sono:

1. Login: Esegue l'autenticazione nel sistema documentale centrale ed effettua il recupero del token di sicurezza - DST.
2. Import: Esegue l'inserimento (e la profilazione) di un nuovo documento nel sistema documentale centrale.
3. Export: Effettua il recupero di un documento (file e metadati) dal sistema documentale centrale.
4. Modify: Esegue la modifica di un documento memorizzato nel sistema documentale centrale.
5. Delete: Esegue la cancellazione di un documento memorizzato nel sistema documentale centrale.
6. getDocuments: Esegue la ricerca di documenti memorizzati nel sistema documentale centrale.

Vista la variabilità delle informazioni di profilazione di un documento è stato deciso di inserire i parametri in un file XML, il cui schema è denominato Doc_info.xsd (per dettagli v. Par. 3 e Appendice A : schema di Doc_Info.xsd).

Le tipologie di parametri che il Web Services tratta sono di seguito elencate:

a. informazioni di sistema: sono le informazioni utili per effettuare le operazioni di riconoscimento nel sistema documentale (ad es. la library di riferimento, lo username dell'utente, il DST, ecc.);
b. <file_XML>: è il file che contiene i dati di profilazione (e di ricerca) del documento oggetto della transazione;
c. <file_binario>: è il file elettronico che rappresenta il documento vero e proprio cui ci si riferisce (solo per import, export, modify, getDocuments) e a cui corrispondono i dati di profilazione presenti nel file XML precedentemente menzionato.

Un altro aspetto importante riguarda la modalità di invio dei file allegati ad un messaggio SOAP; è stato deciso di utilizzare le due tecnologie maggiormente utilizzate e consigliate dal W3C (World Wide Web Consortium). Il Web Service "DMPantarei" è in grado di trattare messaggi SOAP con allegati utilizzando sia la specifica MIME che la specifica DIME. I chiamanti del servizio possono pertanto scegliere come modalità di invio dei file o la tecnologia MIME o DIME.

Quando si inserisce un nuovo documento si hanno due casistiche, di seguito elencate:

- inserimento di un file firmato (formato P7M) che consiste nell'invio di tre file, rigorosamente nell'ordine di seguito riportato:
- file XML di profilazione,

- file elettronico (il documento vero e proprio),
- file P7M.
- inserimento di un file generico, che consiste nell'invio di due file, rigorosamente nell'ordine di seguito riportato:
- file XML di profilazione,
- file elettronico (il documento vero e proprio).

I documenti firmati elettronicamente (formato P7M – le buste in pratica) verranno inseriti come "attachment" (secondo la terminologia di Hummingbird DM) del file elettronico vero e proprio. Gli attachment (in Hummingbird DM) non possiedono un proprio profilo di classificazione, ma si appoggiano al documento principale (di cui costituiscono l'attachment); questo nel gergo Hmb si traduce nel concetto di related (v. Par. 5).

3 Descrizione di dettaglio dei metodi del servizio

3.1 Metodo di autenticazione (login)

Per quanto riguarda il login al sistema documentale centrale, i parametri che dovranno essere forniti dagli applicativi chiamanti sono i seguenti:

- **library_name**: è il nome della libreria a cui ci si vuol connettere (i valori ammissibili sono memorizzati nella tabella REMOTE_LIBRARIES del database di Hummingbird DM);
- **username**: è il nome utente relativo a chi sta effettuando la sessione di lavoro (i valori ammissibili sono memorizzati nella tabella PEOPLE);
- **password**: è la password dell'utente che sta effettuando la sessione di lavoro.

Gli applicativi chiamanti forniranno delle credenziali predefinite al Web Services, sarà pertanto creato un utente arbitrario in Hummingbird DM (ad esempio "ut_protocollatore") che effettuerà tutte le operazioni di interfacciamento tra gli applicativi chiamanti e l'EDMS centrale.
I dati restituiti dal metodo di login sono di seguito riportati:

- **library_name**: è il nome della libreria in cui è stata effettuata l'operazione di accesso;
- **DST (Document Security Token):** è il token generato dal DM Server che tiene traccia delle informazioni relative all'utente collegato e che mantiene le informazioni relative alle policy di sicurezza dell'utente (dovrà essere memorizzato localmente negli applicativi di gestione del protocollo);
- **error_number**: è l'identificativo dell'errore (0 in caso di esito positivo altrimenti il codice di ritorno generato dal DM Server).;
- **error_description**: è la descrizione dell'errore (0 in caso di esito positivo altrimenti la descrizione dell'errore ritornata dal DM Server).

Questo metodo si occuperà di verificare le credenziali dell'utente e in caso di correttezza delle credenziali fornite, dovrà restituire il token DST[20] che costituisce la "chiave" di accesso al sistema documentale per le successive operazioni. Inoltre, il metodo di login restituirà anche delle informazioni di controllo. Gli applicativi di gestione del protocollo dovranno invocare questo metodo come prima chiamata e memorizzare al loro interno il token DST che utilizzeranno sempre ad ogni successiva chiamata; il token DST ha valore solo durante una sessione corrente di lavoro. Di seguito si riporta la tabella di dettaglio del metodo in oggetto.

Parametri input

Nome	Tipo	Descrizione
strLibraryName	String	Il nome della libreria in cui si sta compiendo l'operazione di ricerca.
strUserName	String	Il nome dell'utente che effettua la sessione di lavoro (è un utente predefinito).

[20] Il DST memorizza le informazioni in forma crittografata.

strPassword	String	Password utente

Parametri output

Nome	Tipo	Descrizione
strLibraryName	String	Il nome della libreria in cui.è stata effettuata l'autenticazione
strDST	String	Il DST token (generato dal DM Server) che tiene traccia delle informazioni relative all'utente collegato.
lngErrNumber	Long	Identificativo di errore (0 in caso positivo, altrimenti il codice di errore generato dal DM server).
strErrString	String	La descrizione dell'errore (0 in caso positivo, altrimenti la descrizione dell'errore generata dal DM server).

Tabella 2.1.1a, 2.1.1b – Interfaccia del metodo di login

3.2 Metodo di inserimento documenti (Import)

Per quanto riguarda l'inserimento di un nuovo documento nell'EDMS centrale, i parametri che dovranno essere forniti dagli applicativi di gestione del protocollo sono i seguenti:

- **library_name**: è il nome della libreria in cui si vuol inserire il documento in oggetto (i valori ammissibili sono memorizzati nella tabella REMOTE_LIBRARIES);
- **username**: è l'identificativo dell'utente predefinito (ad esempio "ut_protocollatore") che sta effettuando l'inserimento (i valori ammissibili sono memorizzati nella tabella PEOPLE);
- **DST (Document Security Token**): è il token generato dal DM Server (nella fase di login) che tiene traccia delle informazioni relative all'utente collegato e che mantiene le informazioni relative alle policy di sicurezza dell'utente (memorizzato localmente negli applicativi di gestione del protocollo in una precedente fase di login);
- **<file_XML>:** è il file che contiene i dati di profilazione del documento, sotto forma di coppie [nome,valore], deve essere specificato anche il form utilizzato per la profilazione del documento. Per ulteriori dettagli sulla struttura del file XML consultare il capitolo 4;
- **<file_binario>:** è il file elettronico che si sta inserendo nel sistema e a cui corrispondono i dati di profilazione presenti nel file XML precedentemente menzionato.

I dati restituiti dal metodo di inserimento sono di seguito riportati:

- **library_name**: è il nome della libreria in cui è stato inserito il documento;
- **docnumber**: è l'identificativo univoco del documento che è stato inserito (questo dato dovrà essere memorizzato localmente negli applicativi di gestione del protocollo);
- **error_number**: è l'identificativo dell'errore (0 in caso di esito positivo altrimenti il codice di ritorno generato dal DM Server). Per ulteriori dettagli sulla gestione degli errori, consultare l'Appendice C, intitolata *"Elenco dei codici di errore"*;
- **error_description**: è la descrizione dell'errore (0 in caso di esito positivo altrimenti la descrizione dell'errore ritornata dal DM Server).

Questo metodo si occuperà di inserire un nuovo documento all'interno dell'EDMS centrale, esso dovrà pertanto inviare sia i dati di profilazione del documento oggetto della transazione che il file stesso (o i file, nel caso si invii anche un P7M); inoltre, il metodo restituirà anche delle informazioni di controllo. Il metodo di inserimento gestirà anche la creazione (o l'aggiornamento) dei Fascicoli; per inserire un Fascicolo dovranno essere fornite le seguenti informazioni:

- COD_ENTE: riferimento all'Ente,
- COD_TITOLARIO: riferimento al Titolario di riferimento del Fascicolo,
- ANNO_FASCICOLO: anno del Fascicolo,
- NUM_FASCICOLO: numero del Fascicolo,
- DES_FASCICOLO: descrizione del Fascicolo.

Il metodo di inserimento gestirà anche la creazione delle Aree Tematiche, dovranno essere fornite le seguenti informazioni:

- COD_AREA: codice dell'Area Tematica,
- DES_AREA: descrizione associata all'Area Tematica,
- COD_ENTE: riferimento all'Ente,
- COD_AOO: riferimento alla AOO.

Nel caso esistessero già dei valori identici per i Fascicoli e le Aree Tematiche, il Web Service effettuerà l'update delle informazioni presenti nel database di Hummingbird DM; in particolare l'aggiornamento si riferisce solo ai campi "DES_FASCICOLO" e "DES_AREA".
Le informazioni di profilazione di un documento (e quindi anche di un Fascicolo) sono fornite nel file XML utilizzato per la profilazione e per la ricerca,
Gli applicativi di gestione del protocollo dovranno memorizzare localmente il valore corrispondente al DOCNUMBER; mediante tale valore sarà possibile effettuare le eventuali e successive operazioni di accesso al documento. Uno dei parametri di input è costituito dal token DST precedentemente ottenuto mediante l'invocazione del metodo di login; questa informazione dovrà sempre essere passata ad ogni eventuale richiesta ulteriore, in quanto il token DST mantiene traccia delle informazioni relative all'utente collegato.

Di seguito si riporta la tabella di dettaglio del metodo in oggetto.

Parametri input

Nome	Tipo	Descrizione
strLibraryName	String	Il nome della libreria in cui si sta compiendo l'operazione di ricerca.
strUserName	String	Il nome dell'utente che effettua la sessione di lavoro (è un utente predefinito).
strDST	String	Il DST token (generato dal DM Server) che tiene traccia delle informazioni relative all'utente collegato.
objDocInfo	Attachment DIME/MIME	Il file XML che contiene il nome del form di profilazione e le coppie [nome,valore] necessarie per la classificazione del documento.
objDocument	Attachment DIME/MIME	Il file fisico oggetto della transazione

Parametri output

Nome	Tipo	Descrizione
strLibraryName	String	Il nome della libreria in cui si sta compiendo l'operazione di ricerca.
lngDocID	Long	L'identificativo univoco associato al documento memorizzato nel sistema documentale.
lngErrNumber	Long	Identificativo di errore (0 in caso positivo, altrimenti il codice di errore generato dal DM server).
strErrString	String	La descrizione dell'errore (0 in caso positivo, altrimenti la descrizione dell'errore generata dal DM server).

Tabelle 2.2.1a,2.2.1b – Interfaccia del metodo di importazione documenti nel DM

3.3 Metodo di estrazione documenti (export)

Il recupero di un documento può avvenire in due modalità: mediante invio del docnumber o mediante una ricerca per parametri. Nel primo caso, l'invio del docnumber individua univocamente un documento all'interno dell'EDMS centrale, invece, nel caso di ricerca per parametri, è possibile che il risultato sia una collezione di documenti. In questo caso il metodo ritornerà un messaggio di errore ad indicare che la query effettuata non è in grado di individuare univocamente un documento; pertanto, sarà cura degli applicativi di gestione del protocollo eseguire una query che dia come risultato un unico documento.
Di seguito si riportano i parametri che dovranno essere forniti dagli applicativi di gestione del protocollo:

- **library_name**: è il nome della libreria da cui si vuole recuperare il documento (i valori ammissibili sono memorizzati nella tabella REMOTE_LIBRARIES);
- **docnumber**: è l'identificativo univoco del documento archiviato all'interno dell'EDMS centrale (questo dato è stato memorizzato localmente negli applicativi di gestione del protocollo in una precedente fase di inserimento). Se il parametro docnumber viene inviato con valore "-1", significa che tale dato non è significativo e per identificare il documento che si vuole recuperare verrà utilizzato il contenuto del file XML (passato come parametro più avanti);
- **username**: è l'identificativo dell'utente predefinito (ad esempio "ut_protocollatore") che sta effettuando l'inserimento (i valori ammissibili sono memorizzati nella tabella PEOPLE);
- **DST** (Document Security Token): è il token generato dal DM Server (nella fase di login) che tiene traccia delle informazioni relative all'utente collegato e che mantiene le informazioni relative alle policy di sicurezza dell'utente (memorizzato localmente negli applicativi di gestione del protocollo in una precedente fase di login);
- **<file_XML>** : è il file che contiene il nome del form di ricerca e le coppie [nome,valore] necessarie per identificare il documento in oggetto (nel caso il docnumber non sia significativo). Per ulteriori dettagli sulla struttura del file XML consultare il capitolo 4.

I dati restituiti dal metodo di recupero sono di seguito riportati:

- **library_name**: è il nome della libreria da cui è stato prelevato il documento oggetto della transazione;
- **<file_XML>**: è il file che contiene i dati di profilazione del documento recuperato, per ulteriori dettagli sulla struttura del file XML consultare il capitolo 4;
- **<file_binario>**: è il file elettronico prelevato dal sistema e a cui corrispondono i dati di profilazione presenti nel file XML precedentemente menzionato;
- **error_number**: è l'identificativo dell'errore (0 in caso di esito positivo altrimenti il codice di ritorno generato dal DM Server). Per ulteriori dettagli sulla gestione degli errori, consultare l'Appendice C, intitolata "*Elenco dei codici di errore*";
- **error_description**: è la descrizione dell'errore (0 in caso di esito positivo altrimenti la descrizione dell'errore ritornata dal DM Server);
- **error_file_path**: qualora non sia possibile accedere al file elettronico, viene restituito il suo percorso fisico di memorizzazione nel Document Server per fini di debug o eventuali comunicazioni all'amministratore locale del proprio Ente di appartenenza.

Riassumendo, il recupero di un documento memorizzato all'interno dell'EDMS centrale potrà avvenire in due modalità:

- mediante l'invio del docnumber (che identifica in modo univoco un documento),
- mediante il passaggio di coppie del tipo [nome,valore] (contenute nel file XML passato come parametro).

La tabella di dettaglio dei parametri di input e output:

Parametri input

Nome	Tipo	Descrizione
strLibraryName	String	Il nome della libreria in cui si sta compiendo l'operazione di ricerca.
lngDocID	Long	L'identificativo univoco associato al documento memorizzato nel sistema documentale.
strUserName	String	Il nome dell'utente che effettua la sessione di lavoro (è un utente predefinito).
strDST	String	Il DST token (generato dal DM Server) che tiene traccia delle informazioni relative all'utente collegato.
intExportType	String	Indica il tipo di Export da effettuare (solo profilo "0" o profilo e documento "1").
objSearchInfo	Attachment DIME/MIME	Il file XML che contiene il nome del form di ricerca e le coppie [nome,valore] necessarie per identificare il documento oggetto della transazione, nel caso si invii il valore "–1" nel parametro lngDocId.

Parametri output

Nome	Tipo	Descrizione
strLibraryName	String	Il nome della libreria in cui si sta compiendo l'operazione di ricerca.
objDocInfo	Attachment DIME/MIME	Il file XML che contiene le informazioni di profilazione del documento recuperato.
objDocument	Attachment DIME/MIME	Il file fisico oggetto della transazione
lngErrNumber	Long	Identificativo di errore (0 in caso positivo, altrimenti il codice di errore generato dal DM server).
strErrString	String	La descrizione dell'errore (0 in caso positivo, altrimenti la descrizione dell'errore generata dal DM server).
strErrorFilePath	String	Ritorna il percorso fisico del file all'interno del Document Server nel caso si verifichi un errore di accesso (ad esempio nel caso che il file fosse corrotto).

Tabelle 2.3.1a,2.3.1b – Interfaccia del metodo di estrazione documenti

Il file XML di **output** a struttura fissa restituisce i seguenti dati del profilo (campi della tabella profile):

Dato di Profilo	Tag XML
Identificativo del Documento	DOCNUM
Titolo	DOCNAME
Codice Ente	COD_ENTE
Tipo Documento	TYPE_ID
Applicazione	APP_ID
Stato Pantarei	STATO_PANTAREI
Codice Area Tematica	COD_AREA
Descrizione Area Tematica	DES_AREA
Codice AOO	COD_AOO
Codice Titolario	COD_TITOLARIO
Anno Protocollo	ANNO_PG
Numero Protocollo	NUM_PG
Oggetto Protocollo	OGGETTO_PG
Registro Protocollo	REGISTRO_PG
Anno Fascicolo	ANNO_FASCICOLO
Numero Fascicolo	NUM_FASCICOLO
Descrizione Fascicolo	DES_FASCICOLO
Autore del Documento	AUTHOR_ID
Editore del Documento	TYPIST_ID

Tabelle 2.3.2 parametri di profilazione di output

3.4 Metodo modifica documenti (modify)

Anche la modifica di un documento, così come per il recupero, può avvenire in due modalità: mediante invio del docnumber o mediante una ricerca per parametri. Nel primo caso, l'invio del docnumber individua univocamente un documento all'interno dell'EDMS centrale, invece, nel caso di individuazione del documento da modificare mediante la ricerca per parametri, è possibile che il risultato sia una collezione di documenti. In questo caso il metodo ritornerà un messaggio di errore ad indicare che la query effettuata non è in grado di individuare univocamente un documento da modificare; pertanto, sarà cura degli applicativi di gestione del protocollo eseguire una query che dia come risultato un unico documento.
Di seguito si riportano i parametri che dovranno essere forniti dagli applicativi di gestione del protocollo:

- **library_name**: è il nome della libreria in cui è memorizzato il documento oggetto della transazione (i valori ammissibili sono memorizzati nella tabella REMOTE_LIBRARIES);
- **docnumber**: è l'identificativo univoco del documento che si vuol modificare (memorizzato localmente negli applicativi di gestione del protocollo in una precedente fase di inserimento). Se il parametro docnumber viene inviato con valore "-1", significa che tale dato non è significativo e per identificare il documento che si vuole modificare verrà utilizzato il contenuto del file XML (passato come parametro più avanti);
- **username**: è l'identificativo dell'utente predefinito (ad esempio "ut_protocollatore") che sta effettuando l'operazione di modifica (i valori ammissibili sono memorizzati nella tabella PEOPLE);
- **DST** (Document Security Token): è il token generato dal DM Server che tiene traccia delle informazioni relative all'utente collegato e che mantiene le informazioni relative alle policy di sicurezza dell'utente (memorizzato localmente negli applicativi di gestione del protocollo in una precedente fase di login);
- **modification_type**: identifica il tipo di modifica che si vuole effettuare, ovvero la modifica del solo profilo di classificazione (valore 1 = profile), la sostituzione del file elettronico (valore 2 = document), oppure, la modifica di entrambe le cose, ovvero il profilo e il file (valore 3 = all);
- **<file_XML>**: è il file che contiene i nuovi dati di profilazione del documento oggetto della transazione. Contiene anche il nome del form di ricerca e le coppie [nome,valore] neces-

sarie per identificare il documento oggetto della modifica (nel caso il docnumber non sia significativo). Per ulteriori dettagli sulla struttura del file XML consultare il capitolo 4;

- **<file_binario>**: è il file elettronico che si vuol sostituire nel sistema e a cui corrispondono i dati di profilazione presenti nel file XML precedentemente menzionato.

I dati restituiti dal metodo di modifica sono di seguito riportati:

- **library_name**: è il nome della libreria in cui è stata effettuata l'operazione di modifica;
- **docnumber**: è l'identificativo univoco del documento modificato;
- **error_number**: è l'identificativo dell'errore (0 in caso di esito positivo altrimenti il codice di ritorno generato dal DM Server). Per ulteriori dettagli sulla gestione degli errori, consultare l'Appendice C, intitolata "*Elenco dei codici di errore*";
- **error_description**: è la descrizione dell'errore (0 in caso di esito positivo altrimenti la descrizione dell'errore ritornata dal DM Server);
- **error_file_path**: qualora non sia possibile accedere al file elettronico, viene restituito il suo percorso fisico di memorizzazione nel Document Server per fini di debug o eventuali comunicazioni all'amministratore locale del proprio Ente di appartenenza.

Riassumendo, l'identificazione del documento da modificare potrà avvenire in due modalità:

- mediante l'invio del docnumber (che identifica in modo univoco un documento),
- mediante il passaggio di coppie del tipo [nome,valore] (contenute nel file XML passato come parametro).

Come per l'inserimento, anche il metodo di modifica gestirà la creazione (o l'aggiornamento) dei Fascicoli; per inserire un Fascicolo dovranno essere fornite le seguenti informazioni:

- COD_ENTE: riferimento all'Ente,
- COD_TITOLARIO: riferimento al Titolario di riferimento del Fascicolo,
- ANNO_FASCICOLO: anno del Fascicolo,
- NUM_FASCICOLO: numero del Fascicolo,
- DES_FASCICOLO: descrizone del Fascicolo.

Il metodo di modifica gestirà anche la creazione delle Aree Tematiche, dovranno essere fornite le seguenti informazioni:

- COD_AREA: codice dell'Area Tematica,
- DES_AREA: descrizione associata all'Area Tematica,
- COD_ENTE: riferimento all'Ente,
- COD_AOO: riferimento alla AOO.

Nel caso esistessero già dei valori identici per i Fascicoli e le Aree Tematiche, il Web Services effettuerà l'update delle informazioni presenti nel database di Hummingbird DM; in particolare l'aggiornamento si riferisce ai campi "DES_FASCICOLO" e "DES_AREA". Di seguito si riporta la tabella di dettaglio del metodo in oggetto.

Parametri input

Nome	Tipo	Descrizione
strLibraryName	String	Il nome della libreria in cui si sta compiendo l'operazione di ricerca.
lngDocID	Long	L'identificativo univoco associato al documento memorizzato nel sistema documentale.
strUserName	String	Il nome dell'utente che effettua la sessione di lavoro (è un utente predefinito).
strDST	String	Il DST token (generato dal DM Server) che tiene traccia delle informazioni relative all'utente collegato.

intModificationType	Integer	Identifica il tipo di modifica che si vuole effettuare; i valori ammissibili per questo parametro sono i seguenti: 1 = profile (modifica del solo profilo di classificazione) 2 = document (modifica del solo file elettronico) 3 = all (modifica completa, sia il profilo che il file)
objDocInfo	Attachment DIME/MIME	Il file XML che contiene le nuove informazioni di profilazione del documento oggetto della modifica. Il file XML contiene anche il nome del form di ricerca e le coppie [nome,valore] necessarie per identificare il documento oggetto della transazione, nel caso si invii il valore "–1" nel parametro lngDocId.

Parametri output

Nome	Tipo	Descrizione
strLibraryName	String	Il nome della libreria in cui si sta compiendo l'operazione di ricerca.
lngDocID	Long	L'identificativo univoco associato al documento modificato
lngErrNumber	Long	Identificativo di errore (0 in caso positivo, altrimenti il codice di errore generato dal DM server).
strErrString	String	La descrizione dell'errore (0 in caso positivo, altrimenti la descrizione dell'errore generata dal DM server).
strErrorFilePath	String	Ritorna il percorso fisico del file all'interno del Document Server nel caso si verifichi un errore di accesso (ad esempio nel caso che il file fosse corrotto).

Tabelle 2.3.1a, 2.3.1b – Interfaccia del metodo di estrazione documenti

3.5 Metodo di cancellazione documenti (delete)

Anche la cancellazione di un documento, come il recupero e la modifica, può avvenire in due modalità: mediante invio del docnumber o mediante una ricerca per parametri. Nel primo caso, l'invio del docnumber individua univocamente un documento all'interno dell'EDMS centrale, invece, nel caso di individuazione del documento da cancellare mediante la ricerca per parametri, è possibile che il risultato sia una collezione di documenti. In questo caso il metodo ritornerà un messaggio di errore ad indicare che la query effettuata non è in grado di individuare univocamente un documento; pertanto, sarà cura degli applicativi di gestione del protocollo eseguire una query che dia come risultato un unico documento. Di seguito si riportano i parametri che dovranno essere forniti dagli applicativi di gestione del protocollo:

- **library_name**: è il nome della libreria in cui è presente il documento oggetto dell'operazione di cancellazione (i valori ammissibili sono memorizzati nella tabella REMOTE_LIBRARIES);

- **docnumber**: è l'identificativo univoco del documento da cancellare (memorizzato localmente negli applicativi di gestione del protocollo in una precedente fase di inserimento). Se il parametro docnumber viene inviato con valore "-1", significa che tale dato non è significativo e per identificare il documento che si vuole cancellare verrà utilizzato il contenuto del file XML (passato come parametro più avanti);
- **username**: è l'identificativo dell'utente predefinito (ad esempio "ut_protocollatore") che sta effettuando l'operazione di cancellazione (i valori ammissibili sono memorizzati nella tabella PEOPLE);
- **DST** (Document Security Token): è il token generato dal DM Server che tiene traccia delle informazioni relative all'utente collegato e che mantiene le informazioni relative alle policy di sicurezza dell'utente (memorizzato localmente negli applicativi di gestione del protocollo in una precedente fase di login);
- **deletion_type**: consente di specificare il tipo di cancellazione da effettuare, ovvero la cancellazione del solo file elettronico (valore 1 = delete content) o la cancellazione del file e dei metadati (valore 2 = delete all);
- **<file_XML>** : è il file che contiene il nome del form di ricerca e le coppie [nome,valore] necessarie per identificare il documento in oggetto (nel caso il docnumber non sia significativo). Per ulteriori dettagli sulla struttura del file XML consultare il capitolo 4.

I dati restituiti dal metodo di cancellazione sono di seguito riportati:

- **library_name**: è il nome della libreria in cui è stato cancellato il documento oggetto della transazione;
- **docnumber**: è l'identificativo univoco del documento cancellato;
- **error_number**: è l'identificativo dell'errore (0 in caso di esito positivo altrimenti il codice di ritorno generato dal DM Server). Per ulteriori dettagli sulla gestione degli errori, consultare l'Appendice C, intitolata "*Elenco dei codici di errore*";
- **error_description**: è la descrizione dell'errore (0 in caso di esito positivo altrimenti la descrizione dell'errore ritornata dal DM Server);
- **error_file_path**: qualora non sia possibile accedere al file elettronico, viene restituito il suo percorso fisico di memorizzazione nel Document Server per fini di debug o eventuali comunicazioni all'amministratore locale del proprio Ente di appartenenza.

Riassumendo, l'identificazione del documento da cancellare potrà avvenire in due modalità:

- mediante l'invio del docnumber (che identifica in modo univoco un documento),
- mediante il passaggio di coppie del tipo [nome,valore] (contenute nel file XML passato come parametro).

Di seguito si riporta la tabella di dettaglio del metodo in oggetto.

Parametri input

Nome	Tipo	Descrizione
strLibraryName	String	Il nome della libreria in cui si sta compiendo l'operazione di ricerca.
lngDocID	Long	L'identificativo univoco associato al documento memorizzato nel sistema documentale.
strUserName	String	Il nome dell'utente che effettua la sessione di lavoro (è un utente predefinito).
strDST	String	Il DST token (generato dal DM Server) che tiene traccia delle informazioni relative all'utente collegato.

intDeleteType	Integer	Identifica il tipo di modifica che si vuole effettuare; i valori ammissibili per questo parametro sono i seguenti: 1 = content (cancella solo file elettronico) 2 = document (cancella file e metadati)
objSearchInfo	Attachment DIME/MIME	Il file XML che contiene il nome del form di ricerca e le coppie [nome,valore] necessarie per identificare il documento oggetto della transazione, nel caso si invii il valore "–1" nel parametro lngDocId.

Parametri output

Nome	Tipo	Descrizione
strLibraryName	String	Il nome della libreria in cui si sta compiendo l'operazione di ricerca.
lngDocID	Long	L'identificativo univoco associato al documento modificato
lngErrNumber	Long	Identificativo di errore (0 in caso positivo, altrimenti il codice di errore generato dal DM server).
strErrString	String	La descrizione dell'errore (0 in caso positivo, altrimenti la descrizione dell'errore generata dal DM server).
strErrorFilePath	String	Ritorna il percorso fisico del file all'interno del Document Server nel caso si verifichi un errore di accesso (ad esempio nel caso che il file fosse corrotto).

Tabelle 2.5.1a, 2.5.1b – Interfaccia del metodo di cancellazione delete

3.6 Metodo di ricerca (getDocuments)

Il Web Service renderà disponibile un metodo per l'interrogazione dell'EDMS centrale al fine di recuperare i documenti che soddisfano determinati criteri di ricerca; il metodo, in particolare, permette di calcolare i documenti a cui ha accesso un determinato gruppo o un utente e di reperire per ogni documento il profilo, le ACL ed i diritti del gruppo o dell'utente sul documento in questione.
Di seguito si riportano i parametri che dovranno essere forniti dagli applicativi che vogliono utilizzare questo metodo:

- **library_name**: è il nome della libreria in cui si vuole effettuare la ricerca (i valori ammissibili sono memorizzati nella tabella REMOTE_LIBRARIES);
- **username**: è l'identificativo dell'utente predefinito (ad esempio "ut_protocollatore") che sta effettuando l'operazione di ricerca (i valori ammissibili sono memorizzati nella tabella PEOPLE);
- **DST** (Document Security Token): è il token generato dal DM Server che tiene traccia delle informazioni relative all'utente collegato e che mantiene le informazioni relative alle policy di sicurezza dell'utente (memorizzato localmente negli applicativi di gestione del protocollo in una precedente fase di login);
- **<file_XML>** : è il file che contiene il nome del form di ricerca e le coppie [nome,valore] necessarie per ricercare i documenti. Per ulteriori dettagli sulla struttura del file XML consultare il capitolo 4.

I dati restituiti dal presente metodo sono di seguito riportati:

- **library_name**: è il nome della libreria in cui è stata effettuata la ricerca;
- **error_number**: è l'identificativo dell'errore (0 in caso di esito positivo altrimenti il codice di ritorno generato dal DM Server). Per ulteriori dettagli sulla gestione degli errori, consultare l'Appendice C, intitolata "*Elenco dei codici di errore*";
- **error_description**: è la descrizione dell'errore (0 in caso di esito positivo altrimenti la descrizione dell'errore ritornata dal DM Server);
- **<file_XML>**: è il file che contiene i risultati della ricerca effettuata, di cui di seguito sarà descritto il dettaglio;

Di seguito si riporta la tabella di dettaglio del metodo in oggetto.

Parametri input

Nome	Tipo	Descrizione
strLibraryName	String	Il nome della libreria in cui si sta compiendo l'operazione di ricerca.
strUserName	String	Il nome dell'utente che effettua la sessione di lavoro (è un utente predefinito).
strDST	String	Il DST token (generato dal DM Server) che tiene traccia delle informazioni relative all'utente collegato.
objSearchInfo	Attachment DIME/MIME	E' il file che contiene il nome del form di ricerca e le coppie [nome,valore] utilizzate per la ricerca; a fronte di N coppie con stesso "nome" il metodo interpreta il formato come un OR logico tra i "valori"

Parametri output

Nome	Tipo	Descrizione
strLibraryName	String	Il nome della libreria in cui si sta compiendo l'operazione di ricerca.
lngErrNumber	Long	Identificativo di errore (0 in caso positivo, altrimenti il codice di errore generato dal DM server).
strErrString	String	La descrizione dell'errore (0 in caso positivo, altrimenti la descrizione dell'errore generata dal DM server).
ObjDocInfo	Attachment DIME/MIME	Il file XML che contiene i risultati della ricerca.

Tabelle 2.6.1a,2.6.1b – Interfaccia del metodo di ricerca documenti

Il file XML di **output** a struttura fissa restituisce i seguenti dati del profilo (campi della tabella profile):

Dato di Profilo	Tag XML
Identificativo Documento	DOCNUM
Titolo	DOCNAME
Codice Ente	COD_ENTE
Tipo Documento	TYPE_ID
Applicazione	APP_ID
Stato Pantarei	STATO_PANTAREI
Codice Area Tematica	COD_AREA

Descrizione Area Tematica	DES_AREA
Codice AOO	COD_AOO
Codice Titolario	COD_TITOLARIO
Anno Protocollo	ANNO_PG
Numero Protocollo	NUM_PG
Oggetto Protocollo	OGGETTO_PG
Registro Protocollo	REGISTRO_PG
Anno Fascicolo	ANNO_FASCICOLO
Numero Fascicolo	NUM_FASCICOLO
Descrizione Fascicolo	DES_FASCICOLO
Autore del Documento	AUTHOR_ID
Editore del Documento	TYPIST_ID
Estensione del file	DEFAULT_EXTENSION

Tabelle 2.6.2 a,2.6.3b – Parametri di output

Nel file XML di output del metodo GetDocuments viene restituito un TAG (RIGHTS) che riassume i diritti totali dell'Utente o Gruppo (parametro di input presente nel file XML della richiesta) sul singolo documento trovato.
Il tag <RIGHTS> è dello stesso tipo di SECURITY_INFOS:

```
<RIGHTS>
<SECURITY_INFO ACLPersonGroup="{Utente o Gruppo di Input}" ACLMask="0" />
</RIGHTS>
```

In questo tag sono presenti i diritti reali dell'Utente/Gruppo sul Documento che il Web Service calcola come segue:

1. Per un Gruppo è esattamente il diritto esplicito del Gruppo nelle ACL del Documento
2. Per un Utente è l'unione dei diritti espliciti sull'Utente (se presenti) e di quelli espliciti su tutti i Gruppi a cui l'Utente appartiene

Resta inteso che se l'utente o uno dei suoi gruppi di accesso ha diritti espliciti di "Revoca" su un documento, lo stesso non apparirà nella lista dei risultati.

4 Note sul file xml di profilazione e ricerca

Il file XML che contiene le informazioni di profilazione e di ricerca (DOC_INFO.xsd) è specificato da uno schema XML. Tale file viene utilizzato in tutte le operazioni ad eccezione del login. Lo schema vede la composizione del file da due elementi principali a partire dall'elemento radice (DOC_INFO), SEARCH_INFOS (elementi per l'interrogazione ai fini di ricerca di un documento) e DOC_DESCR (elementi per la profilazione di un documento).
Quando il file XML viene utilizzato per la profilazione di un documento (il metodo "Import") sarà presente solo l'elemento etichettato "DOC_DESCR" (e i suoi figli). Invece, nel caso di recupero o cancellazione di un documento (i metodi "Export" e "Delete") sarà presente solo l'elemento etichettato "SEARCH_INFOS" (e i suoi figli). Nel caso di modifica (il metodo "Modify") possono essere presenti entrambi gli elementi ("SEARCH_INFOS" e "DOC_DESCR") in quanto il file XML può essere usato sia per individuare il documento da modificare e sia per specificare i nuovi valori dei parametri di classificazione. Di seguito si riporta una tabella riepilogativa di quanto detto sopra.

	SEARCH_INFO	DOC_DESCR
Login		
Import		**X**
Export	**X**	
Modify	**X**	**X**
Delete	**X**	
getDocuments		**X**

Tabella 3.1: Struttura del file XML di profilazione e ricerca rispetto

ai metodi del Web Services

Di seguito si schematizza la sintesi dei vari elementi del file da utilizzare per profilazione e ricerca (per il dettaglio di ogni singolo metodo si veda l'appendice A).

Elemento	**Descrizione**	
DOC_INFO	Sequenza di due elementi opzionali (SEARCH_INFOS e DOC_DESCR).	
SEARCH_INFOS	Sequenza di un elemento (SEARCH_FORM_NAME) e di una sequenza di elementi SEARCH_INFO.	
SEARCH_FORM_NAME	**Attributi**	**Tipo**
	name	String
SEARCH_INFO	**Attributi**	**Tipo**
	name	String
	value	String
DOC_DESCR	Sequenza di due elementi (PROFILE_INFOS e SECURITY_INFOS e related).	
PROFILE_INFOS	Sequenza di un elemento (PROFILE_FORM_NAME) e di una sequenza di elementi PROFILE_INFO.	
PROFILE_FORM_NAME	**Attributi**	**Tipo**
	name	String
PROFILE_INFO	**Attributi**	**Tipo**
	name	String
	value	String
SECURITY_INFOS	Sequenza di elementi di tipo SECURITY_INFO.	
SECURITY_INFO	**Attributi**	**Tipo**
	ACLPersonGroup	String
	ACLMask	enumeration [-1,0,1,2,3] Definisce gli attributi di sicurezza di un documento, può assumere i seguenti valori: -1 = Revoca accesso 0 = Accesso completo 1 = Accesso normale, 2 = Sola lettura 3 = Visualizza profilo
RELATED	Sequenza di uno o più elementi ITEM.	

ITEM	**Attributi**	**Tipo**
	Number	Integer
	op	enumeration [0,1] Specifica se si sta eliminando o aggiungendo una relazione di related, può assumere i seguenti valori: 0 = eliminazione 1 = aggiunta

5 Concetto di documenti related

Un altro aspetto da considerare è quello relativo alla gestione dei documenti "related". In Hummingbird DM, a partire dalla versione 5, è possibile creare dei riferimenti logici tra un insieme di documenti, mediante il concetto di documento related (secondo la terminologia del prodotto).
In questo modo, a partire da un documento è possibile conoscere tutti i documenti a cui è esso è correlato, mediante la scheda etichettata "Related Items" presente nell'applicazione WebTop di Hummingbird DM (o nelle applicazioni Windows denominate "DM Extensions"). La relazione che si instaura tra due documenti related è di tipo bidirezionale (non gerarchica). Per gestire i documenti related è stata introdotta (a partire dalla versione 5 di Hummingbird DM) una tabella denominata "RELATED" in cui sono riportate tutte le relazioni tra i vari documenti correlati. La tabella tiene traccia delle relazioni in modo semplice e lineare: esistono due campi, denominati rispettivamente "ITEM1" e "ITEM2" in cui sono riportati gli identificativi univoci (docnumber) dei documenti oggetto della relazione.
Per quanto riguarda il progetto DOCAREA, verranno inseriti come "related" tutti i file allegati a un documento protocollato. In fase di inserimento di un documento allegato deve essere passato al Web Services anche la chiave univoca (docnumber) che identifica il file protocollato a cui l'allegato si riferisce. I file allegati (di un documento protocollato) vengono quindi inseriti uno per volta. Questa modalità di inserimento degli allegati permette di avere maggiore controllo sulla singola transazione; per ciascuna chiamata del Web Services si ha un messaggio di ritorno che comunica l'esito della transazione effettuata. Per la gestione dei documenti related è stata inserita un'apposita sezione nel file XML di profilazione (e di ricerca); è stato pertanto aggiunto un elemento denominato "RELATED" che al suo interno ha un numero arbitrario di elementi denominati "ITEM". Ad ogni elemento "ITEM" corrisponde un documento, specificato mediante l'attributo denominato "number"; inoltre, mediante l'attributo denominato "op" si può specificare se si sta aggiungendo una relazione di related (valore 1) o se si sta eliminando la relazione di related (valore 0). Di seguito si riporta un esempio di sezione "RELATED":

```
<RELATED>
  <ITEM number="21" op="1"/>
  <ITEM number="20" op="0"/>
</RELATED>
```

6 Appendice A – Schema del file Doc_info

6.1 Doc_info.xsd

Di seguito lo schema xsd per il file Doc_Info

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
```

```
project name: DOCAREA
file name: doc_info.xsd
version: 1.4
date: 24 Marzo 2004
author: Paolo Diomede - paolo.diomede@hummingbird.com
company: Hummingbird S.p.A. - Global Professional Services
note: schema XML per la profilazione e la ricerca di documenti tramite l'interfaccia basata sui Web
Services
-->
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema" elementFormDefault="qualified">
  <xsd:element name="DOC_INFO">
          <xsd:annotation>
                  <xsd:documentation>Elemento radice</xsd:documentation>
          </xsd:annotation>
          <xsd:complexType>
                  <xsd:sequence>
                          <xsd:element name="SEARCH_INFOS" minOccurs="0">
                                  <xsd:annotation>
                                          <xsd:documentation>
Contiene le informazioni che permettono di identificare un documento mediante una interrogazione,
nel caso non si usi il docnumber.
                                          </xsd:documentation>
                                  </xsd:annotation>
                                  <xsd:complexType>
                                          <xsd:sequence>
                                                  <xsd:element name="SEARCH_FORM_NAME"
nullable="true">
                                                          <xsd:annotation>
                                                                  <xsd:documentation>
E` il nome del form utilizzato per eseguire le ricerche nel sistema documentale.
                                                                  </xsd:documentation>
                                                          </xsd:annotation>
                                                          <xsd:complexType>
                                                                  <xsd:attribute name="name"
type="xsd:string" use="required"/>
                                                          </xsd:complexType>
                                                  </xsd:element>
                                                  <xsd:sequence maxOccurs="unbounded">
                                                          <xsd:element name="SEARCH_INFO"
nullable="true"
                                                          maxOccurs="unbounded">
                                                                  <xsd:annotation>
                                                                          <xsd:documentation>
Sono le coppie [Nome,Valore] che descrivono i parametri di ricerca.
                                                                          </xsd:documentation>
                                                                  </xsd:annotation>
                                                                  <xsd:complexType>
                                                                          <xsd:attribute name="name"
type="xsd:string" use="required"/>
                                                                          <xsd:attribute name="value"
type="xsd:string" use="required"/>
                                                                  </xsd:complexType>
                                                          </xsd:element>
                                                  </xsd:sequence>
                                          </xsd:sequence>
                                  </xsd:complexType>
                          </xsd:element>
```

```
                        <xsd:element name="DOC_DESCR" minOccurs="0">
                                <xsd:annotation>
                                        <xsd:documentation>
Contiene le informazioni di profilazione di un documento.
                                        </xsd:documentation>
                                        <xsd:documentation>
Contiene le informazioni di profilazione e di sicurezza
                                        </xsd:documentation>
                                </xsd:annotation>
                                <xsd:complexType>
                                        <xsd:sequence>
                                                <xsd:element name="PROFILE_INFOS"
minOccurs="0">
                                                        <xsd:annotation>
                                                                <xsd:documentation>
Contiene la definizione degli attributi di profilazione di un documento.
                                                                </xsd:documentation>
                                                        </xsd:annotation>
                                                        <xsd:complexType>
                                                                <xsd:sequence>
                                                                        <xsd:element
name="PROFILE_FORM_NAME" nullable="true">
                                                                                <xsd:annotation>

  <xsd:documentation>
E` il nome del form utilizzato per  eseguire la profilazione di un documento  nel sistema documentale.

  </xsd:documentation>
                                                                                </xsd:annotation>
                                                                                <xsd:complexType>

  <xsd:attribute name="name" type="xsd:string"

  use="required"/>

  </xsd:complexType>
                                                                        </xsd:element>
                                                                        <xsd:sequence
maxOccurs="unbounded">
                                                                                <xsd:element
name="PROFILE_INFO" nullable="true"

  maxOccurs="unbounded">

  <xsd:annotation>

  <xsd:documentation>
Sono le coppie [Nome,Valore] che descrivono gli attributi di profilazione di un documento.

  </xsd:documentation>

  </xsd:annotation>

  <xsd:complexType>
```

```
<xsd:attribute name="name" type="xsd:string"

use="required"/>

<xsd:attribute name="value" type="xsd:string"

use="required"/>

</xsd:complexType>
                                                                    </xsd:element>
                                                                </xsd:sequence>
                                                            </xsd:sequence>
                                                        </xsd:complexType>
                                                    </xsd:element>
                                                    <xsd:element name="SECURITY_INFOS"
minOccurs="0">
                                                        <xsd:annotation>
                                                            <xsd:documentation>
Contiene le informazioni relative alla sicurezza applicata al documento.
                                                            </xsd:documentation>
                                                        </xsd:annotation>
                                                        <xsd:complexType>
                                                            <xsd:sequence
maxOccurs="unbounded">
                                                                <xsd:element
name="SECURITY_INFO" nullable="true"
                                                                maxOccurs="unbounded">
                                                                    <xsd:annotation>


  <xsd:documentation>
  Definisce gli attributi di sicurezza di un documento, con i seguenti valori: 0 = =    Norrmal Access, 2 =
Read Only, 3 = View Profile

  </xsd:documentation>
                                                                    </xsd:annotation>
                                                                    <xsd:complexType>


  <xsd:attribute name="ACLPersonGroup" type="xsd:string"

  use="required"/>

  <xsd:attribute name="ACLMask" use="required">

  <xsd:simpleType>

  <xsd:restriction base="xsd:NMTOKEN">

          <xsd:enumeration value="-1"/>

          <xsd:enumeration value="0"/>

          <xsd:enumeration value="1"/>

          <xsd:enumeration value="2"/>
```

```
            <xsd:enumeration value="3"/>
  </xsd:restriction>
  </xsd:simpleType>
  </xsd:attribute>
  </xsd:complexType>
                                                                    </xsd:element>
                                                                </xsd:sequence>
                                                            </xsd:complexType>
                                                        </xsd:element>
                                                        <xsd:element name="RELATED" minOccurs="0">
                                                            <xsd:annotation>
                                                                <xsd:documentation>
Contiene le informazioni sui documenti "related".
                                                                </xsd:documentation>
                                                            </xsd:annotation>
                                                            <xsd:complexType>
                                                                <xsd:sequence>
                                                                    <xsd:element name="ITEM"
maxOccurs="unbounded">
                                                                        <xsd:annotation>
  <xsd:documentation>
Elenco di documenti related, viene specificato se si tratta di un inserimento   di una rimozione di una
relazione, con i seguenti valori per l'attributo "op": 0 = rimozione, 1 = aggiunta.
  </xsd:documentation>
                                                                        </xsd:annotation>
                                                                        <xsd:complexType>
  <xsd:attribute name="number" type="xsd:integer"
  use="required"/>
  <xsd:attribute name="op" use="optional">
  <xsd:simpleType>
  <xsd:restriction base="xsd:NMTOKEN">
            <xsd:enumeration value="0"/>
            <xsd:enumeration value="1"/>
  </xsd:restriction>
  </xsd:simpleType>
  </xsd:attribute>
```

```
  </xsd:complexType>
                                                                    </xsd:element>
                                                                </xsd:sequence>
                                                            </xsd:complexType>
                                                        </xsd:element>
                                                    </xsd:sequence>
                                                    <xsd:attribute name="isRelated" type="xsd:string"
use="default" value="no"/>
                                        </xsd:complexType>
                                    </xsd:element>
                                </xsd:sequence>
                            </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

7 Appendice B: esempi di file Soap per le chiamate dei metodi DMPantarei

Di seguito si dettagliano i files soap per l'invocazione dei metodi di DMPantarei e i files di profilazione associati (schema Doc_info.xsd).

7.1 Login

Esempio di File soap per la chiamata Login:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Login xmlns="http://tempuri.org/">
      <strLibraryName>docarea</strLibraryName>
      <strUserName>ut_proto_ai</strUserName>
      <strPassword>zzzzzz</strPassword>
    </Login>
  </soap:Body>
</soap:Envelope>
```

7.2 Import

SoapImport.xml

Esempio di file soap per la chiamata import:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
        <Import xmlns="http://tempuri.org/">
              <strLibraryName>DOCAREA</strLibraryName>
              <strUserName>UT_PROTO_AI</strUserName>


<strDST>b63599643a31e63ba804b6eeacf7954acfeb4c58ebad7b3d8b5a014e2c471bb9113ea8c3171
5242fcc594a777eea30c749a31fe2f02e8751f0b2fa17f4a15f1d9a42e85a854f82ed6f3d9aff4789ccae36
962e7b0a3820c8c70d835f519c1ea15878a6fb2613c6096c0c5d43c07ce8a783acf6a195b952817dcf6b
```

```
a0ad9be5de</strDST>
        </Import>
  </soap:Body>
</soap:Envelope>
```

SoapImportProfile.xml

Un esempio di istanza del file Doc_info.xsd per l'import è il seguente:

```
<?xml version="1.0" encoding="UTF-8"?>

<DOC_INFO xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="doc_info_v1.4.1.xsd" >
  <SEARCH_INFOS>
        <SEARCH_FORM_NAME name="search_full" />
  </SEARCH_INFOS>
  <DOC_DESCR>
        <PROFILE_INFOS>
              <PROFILE_FORM_NAME name="DEF_DOC_GENERICO" />
              <PROFILE_INFO value="NomeProva"  name="DOCNAME" />
              <PROFILE_INFO value="PROTO_EMR " name="COD_ENTE" />
              <PROFILE_INFO value="DOCUMENTO" name="TYPE_ID" />
              <PROFILE_INFO value="1" name="STATO_PANTAREI" />
        </PROFILE_INFOS>
        <SECURITY_INFOS>
              <SECURITY_INFO ACLPersonGroup="UT_PROTO_AI" ACLMask="2" />
        </SECURITY_INFOS>
  </DOC_DESCR>
</DOC_INFO>
```

7.3 Export

SoapExport.xml

Esempio di file soap per la chiamata export:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
        <Export xmlns="http://tempuri.org/">
              <strLibraryName>pantarei</strLibraryName>
              <lngDocID>62</lngDocID>
              <strUserName>UT_PROTO_BO</strUserName>


<strDST>b63599643a31e63ba804b6eeacf7954ab238e63559207b6839c5a50e9775e99161cea5641b
28beb7e9b7473bbe0edceeef1fd09eaa664119c1578d1f720ddcd5222752bc2159c60983a7bae090e43
371e81447e7a58b6eadbd571dfad7791327d8909bee78ae282669b07355fe32f371c8421143bd971b15
f32e9a88bc96f1c1</strDST>
        </Export>
  </soap:Body>
</soap:Envelope>
```

SoapExportProfile.xml

Un esempio di istanza del file Doc_info.xsd per l'export è il seguente:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<DOC_INFO xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="doc_info_v1.4.1.xsd" >

  <SEARCH_INFOS>
          <SEARCH_FORM_NAME name="SEARCH_FULL"/>
          <SEARCH_INFO name="DOCNUM" value="61"/>
          <PROFILE_INFO name="ANNO_PG" value="2004”/>
          <PROFILE_INFO name="NUM_PG" value="131”/>
  </SEARCH_INFOS>
</DOC_INFO>
```

7.4 Modify

SoapModify.xml

Esempio di file soap per la chiamatagetDocuments:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
          <Modify xmlns="http://tempuri.org/">
                  <strLibraryName>docarea</strLibraryName>
                  <lngDocID>134</lngDocID>
                  <strUserName>UT_PROTO_EMR</strUserName>


<strDST>b63599643a31e63ba804b6eeacf7954acfeb4c58ebad7b3d8b5a014e2c471bb9717c74907e1
003ce9d36c310af0e89c927715a16c0f7ad83c7223b7c328edfa7ce89cb09ffa034f0ff167f7a2ca5fe9c23f
c6b5110c899e0d18cf0daa2bbd1f85251854e7e4c9891beb40d9d81229bc822fc2a32fe8ad259b8a335a
aa1046424</strDST>
                  <intModificationType>1</intModificationType>
          </Modify>
  </soap:Body>
</soap:Envelope>
```

SoapmodifyProfile.xml

Un esempio di istanza del file Doc_info.xsd per il metodo modify è il seguente:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<DOC_INFO>
  <SEARCH_INFOS>
          <SEARCH_FORM_NAME name="search_full"/>
  </SEARCH_INFOS>
  <DOC_DESCR>
          <PROFILE_INFOS>
                  <PROFILE_FORM_NAME name="def_proto_pantarei"/>
                  <PROFILE_INFO value="ProtocolloRBX.doc" name="DOCNAME"/>
                  <PROFILE_INFO value="EMR" name="COD_ENTE"/>
                  <PROFILE_INFO value="DOCUMENTO" name="TYPE_ID"/>
                  <PROFILE_INFO value="4" name="STATO_PANTAREI"/>
                  <PROFILE_INFO value="93" name="COD_AREA"/>
                  <PROFILE_INFO value="Trasmissioni di delibere di giunta alle commissioni
consiliari e all ufficio di Presidenza del Consiglio come quando fuori piove" name="DES_AREA"/>
                  <PROFILE_INFO value="AOO_EMR" name="COD_AOO"/>
                  <PROFILE_INFO value="2005" name="ANNO_PG"/>
```

```
                <PROFILE_INFO value="758" name="NUM_PG"/>
                <PROFILE_INFO value="prova hmb" name="OGGETTO_PG"/>
                <PROFILE_INFO value="PG" name="REGISTRO_PG"/>
        </PROFILE_INFOS>
        <SECURITY_INFOS>
                <SECURITY_INFO ACLPersonGroup="EMR" ACLMask="2"/>
        </SECURITY_INFOS>
  </DOC_DESCR>
</DOC_INFO>
```

7.5 ***Delete***

SoapDelete.xml

Esempio di file soap per la chiamata Delete:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
        <Delete xmlns="http://tempuri.org/">
                <strLibraryName>docarea</strLibraryName>
                <lngDocID>20</lngDocID>
                <strUserName>administrator</strUserName>


<strDST>b63599643a31e63ba804b6eeacf7954ab238e63559207b6839c5a50e9775e99161cea5641b
28beb7e9b7473bbe0edceeef1fd09eaa664119c1578d1f720ddcd5222752bc2159c60983a7bae090e43
371e81447e7a58b6eadbd571dfad7791327d8909bee78ae282669b07355fe32f371c8421143bd971b15
f32e9a88bc96f1c1</strDST>
                <intDeleteType>2</intDeleteType>
        </Delete>
  </soap:Body>
</soap:Envelope>
```

SoapDeleteProfile.xml

Un esempio di istanza del file Doc_info.xsd per il metodo deleteè il seguente:

```
<?xml version="1.0" encoding="UTF-8"?>

<DOC_INFO xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="doc_info_v1.4.1.xsd" >
  <DOC_DESCR>
        <SEARCH_INFOS>
                <SEARCH_FORM_NAME name="search_full"/>
                <SEARCH_INFO name="DOCNUM" value="20"/>
        </SEARCH_INFOS>
  </DOC_DESCR>
</DOC_INFO>
```

7.6 ***getDocuments***

SoapGet.xml

Esempio di file soap per la chiamatagetDocuments:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
        <Import xmlns="http://tempuri.org/">
              <strLibraryName>docarea</strLibraryName>
              <strUserName>administrator</strUserName>


<strDST>b63599643a31e63ba804b6eeacf7954ab238e63559207b6839c5a50e9775e99161cea5641b
28beb7e9b7473bbe0edceeef1fd09eaa664119c1578d1f720ddcd5222752bc2159c60983a7bae090e43
371e81447e7a58b6eadbd571dfad7791327d8909bee78ae282669b07355fe32f371c8421143bd971b15
f32e9a88bc96f1c1</strDST>
        </Import>
  </soap:Body>
</soap:Envelope>
```

SoapGetDocumentsProfile.xml

Un esempio di istanza del file Doc_info.xsd per la ricerca è il seguente:

```
<?xml version="1.0" encoding="UTF-8"?>

<DOC_INFO xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="doc_info_v1.4.1.xsd" >

  <DOC_DESCR>
        <USERGROUPS>
              <USERGROUP name="group" value="MANDATI"/>

        </USERGROUPS>
        <PROFILE_INFOS>
              <PROFILE_FORM_NAME name="search_full"/>
              <PROFILE_INFO name="type_id" value="documento" />
              <PROFILE_INFO name="type_id" value="fattura" />
              <PROFILE_INFO name="anno_pg" value="2005" />
              <PROFILE_INFO name="stato_pantarei" value="5" />
              <PROFILE_INFO name="num_pg" value="10 TO 100" />
        </PROFILE_INFOS>
  </DOC_DESCR>
</DOC_INFO>
```

8 Appendice C– Elenco dei codici di errore

In questa appendice sono riportati tutti i codici di errore forniti nei messaggi di risposta del Web Services “DMPantarei”. In base a queste informazioni e all’interfaccia WSDL fornita è possibile implementare l’integrazione degli applicativi di gestione del protocollo con il sistema documentale centrale (Hummingbird DM).

Codice Errore	Descrizione
0	Esecuzione OK
-1	Errore nel codice del Web Services.
-1	Errore nel codice del Web Services.
-2	Errore nei Trustees: presenza di Gruppi o Utenti sconosciuti.
-3	Errore nei Trustees: diritti da assegnare non riconosciuti (0,1,2,3 ammissibili).
-4	DST non valido.

-10	Errore nella Login: il file XML di Login non è ben formato.
-11	Errore nella Login: LibraryName non corretta.
-12	Errore nella Login: UserId non corretta.
-13	Errore nella Login: Password non corretta.
-14	Errore nella Login: DM Server non disponibile.
-20	Errore nell' Import: il file XML di Import non è ben formato.
-21	Errore nell' Import: numero Allegati non corretto.
-22	Errore nell' Import: Profile Form Name non corretto.
-30	Errore nella Modifica: il file XML di Modifica non è ben formato.
-31	Errore nella Modifica: verificare gli Allegati della richiesta.
-32	Errore nella Modifica: nessun documento trovato.
-33	Errore nella Modifica: documento trovato non univoco.
-34	Errore nella Modifica: form di Ricerca non presente.
-35	Errore nella Modifica: l'utente non possiede diritti di modifica sul documento.
-36	Errore nella Modifica: manca l'allegato della richiesta.
-40	Errore nella Export: il file XML di Export non è ben formato
-41	Errore nella Export: nessun documento trovato.
-42	Errore nella Export: documento trovato non univoco.
-43	Errore nella Export: form di Ricerca non presente.
-50	Errore nella Cancellazione: il file XML di Export non è ben formato.
-51	Errore nella Cancellazione: nessun documento trovato.
-52	Errore nella Cancellazione: documento trovato non univoco.
-53	Errore nella cancellazione: form non presente
<<0	Errori gestiti da DM Server legati a non correttezza o completezza dei dati forniti per eseguire le operazioni richieste. Nel campo "error_description" (il parametro "strErrString") presente in ogni metodo del Web Services sarà riportata la descrizione dell'errore.

Allegato 12: Specifiche tecniche per l'utilizzo dei servizi dell'infrastruttura di firma digitale

1 Introduzione

Questo documento ha lo scopo di descrivere le possibili soluzioni da adottare per implementare funzionalità di firma digitale nelle applicazioni.

Le funzionalità disponibili sono:

- Firma di documenti (richiede thin client e smart card)
- Verifica della firma di documenti (con e senza thin client)
- Cifra documenti (richiede thin client)
- Decifra documenti (richiede thin client e smart card)

Tali funzionalità, oltre ad essere rese disponibili come applicazione web usufruibile via browser, sono disponibili anche come web services per l'integrazione con altre applicazioni.

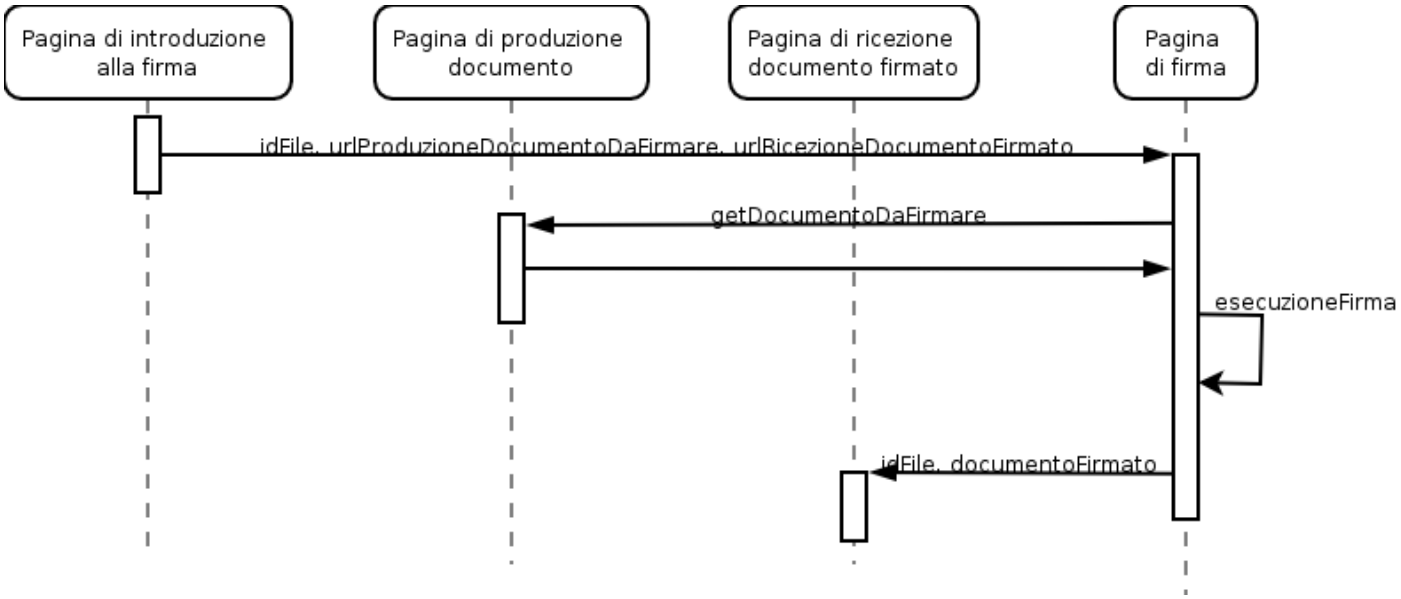
La funzione di firma di documenti è usufruibile anche sotto forma di pagina web da integrare con altre applicazioni al fine di realizzare la firma web dei file stessi, ritornando all'applicazione chiamante il file firmato. (richiede thin client e smart card).

NB: alcune funzionalità, come indicato, richiedono che sul pc client che esegue l'applicazione siano installati il lettore smart card, i driver necessari e la libreria "thin client" per l'interfacciamento. Si tratta di una configurazione standard per le postazioni di lavoro regionali dotate di lettore smart card.

2 Integrazione tramite chiamata a pagina web

Nel presente paragrafo si descriverà la metodologia di funzionamento della pagina di integrazione per la firma di documenti prodotti da applicazioni esterne.

L'immagine seguente evidenzia la sequenza di operazioni che vengono condotte al fine di firmare un documento prodotto da un'applicazione:



In sintesi i passi per pervenire alla firma di un proprio documento un'applicazione deve:

1. dopo aver portato l'utente a produrre il documento, in una pagina apposita redirigere l'utente

verso la pagina di firma, proponendosi di passare a tale pagina tre parametri:

a. **idFile**: un identificativo del file, che lo riferisca univocamente, in modo che all'invio del corrispondente file firmato all'applicazione essa sia in grado di ricondurlo al file originale
b. **urlProduzioneDocumentoDaFirmare**: l'url che la pagina di firma richiamerà che dovrà produrre il file da firmare. Meccanismi di autenticazione e di sicurezza devono essere messi in campo a carico dell'applicazione di partenza.
c. **urlRicezioneDocumentoFirmato**: l'url a cui la pagina di firma farà il post del documento firmato

2. una volta attivata, la pagina di firma, in modo automatico, permette all'utente di scaricare il file firmato dall'url passato al punto precedente e di firmarlo.
3. una volta compiuta la firma, la pagina di firma redireziona la navigazione verso la pagina di ricezione del documento firmato, indicata sempre al passo 1, postando il file stesso.

Oltre al documento firmato verrà inviato:

- l'identificativo del file (ricevuto al punto 1)
- un codice sul risultato dell'operazione
- l'url da cui è stato scaricato il file da firmare (ricevuto al punto 1).

2.1 Dettagli tecnici

La pagina di firma di integrazione può essere referenziata come:

- In produzione: https://firmadigitale.ente.regione.emr.it/FD/jsp/firmaURL.jsp
- In test: https://firmadigitaletest.ente.regione.emr.it/FD/jsp/firmaURL.jsp

I parametri da passare (in GET o in POST) alla pagina all'atto della chiamata sono:

- **fileid**: l'idFile menzionato precedentemente.
- **fileurl**: url da cui ottenere il file da firmare, corrispondente a urlProduzioneDocumentoDa-Firmare.
- **filepost**: url a cui spedire il file firmato oltre che pagina cui redirezionare l'utente dopo la firma. Corrisponde a urlRicezioneDocumentoFirmato.

La pagina di firma, al termine dell'operazione di firma, esegue un POST all'url corrispondente a "filepost" con i seguenti parametri:

- **signeddata**: il valore è il contenuto del file firmato codificato in base64.
- **fileid**: il valore è l'identificativo del file corrispondente al fileid ricevuto nella chiamata del-la pagina
- **returncode**: il valore è il codice di ritorno dell'operazione: vale -1 se c'è stato un errore, è vuoto altrimenti
- **fileurl**: il valore è l'url da cui è stato scaricato il file da firmare corrispondente al parame-tro fileurl passato alla chiamata.

Sostanzialmente quindi un'applicazione che necessita di integrarsi con la firma digitale attraverso questo meccanismo deve realizzare tre pagine logiche:

- una pagina di produzione del file che rediriga la navigazione dell'utente verso la pagina di firma
- una pagina che produca in output il file da firmare
- una pagina che sappia ricevere l'output dell'operazione di firma e sappia trattarlo di con-seguenza.

Tecnicamente, le tre pagine menzionate possono essere realizzate con un'unica pagina fisica: l'importante è che rispettino i parametri di chiamata e sappiano interpretare i parametri di ritorno.

3 Integrazione tramite chiamata dei web services

Le applicazioni che necessitano l'integrazione di funzionalità di firma possono richiamare i seguenti

servizi web:

- CallCifra: consente di cifrare un file restituendo un file cifrato secondo lo standard PKCS#7.
- CallDecifra: consente di decifrare un file cifrato secondo lo standard PKCS#7.
- CallFirma: consente la firma digitale di un file, restituendo il file firmato secondo lo standard PKCS#7.
- CallFirmaXml: consente di firmare un file XML nel formato XMLSignature, secondo lo standard: XML-Signature Syntax and Processing. Consente inoltre di verificare l'integrità di un file XML firmato.
- CallSearchLDAP: consente di cercare su un LDAP i certificati di CA o i certificati utente.
- CallTimeStamp: consente di eseguire una richiesta di marcatura temporale. Consente di verificare l'integrità di una marca temporale e di controllare la credibilità e la validità del certificato del firmatario.
- CallVerifica: consente la verifica della firma apposta ad un file restituendo l'esito della verifica, i certificati dei firmatari e i dati originari recuperati dalla verifica.

I web service sono disponibili ai seguenti indirizzi:

- Ambiente di test: https://firmadigitaletest.ente.regione.emr.it/axis/
- Ambiente di produzione: https://firmadigitale.ente.regione.emr.it/axis/

Nel capitolo 4 è riportata la documentazione sintetica delle classi di interfaccia dei webservices richiamabili dalle applicazioni. La documentazione completa della libreria è consultabile all'indirizzo https://firmadigitale.ente.regione.emr.it/doc/ (raggiungibile solo dall'interno della rete dell'ente) oppure può essere fornita mandando apposita richiesta al SIIR.

Nel capitolo 5 sono proposti alcuni esempi di utilizzo dei web services.

NB: le applicazioni *client* che richiamano tali servizi web devono utilizzare il protocollo *https*.

4 Documentazione tecnica sintetica dei web-services

4.1 Class CallCifra

```
java.lang.Object
  └ com.engiweb.axis.ejbcryptoserver.cifra.CallCifra
```

```
public class CallCifra
extends java.lang.Object
```

Classe di interfaccia SOAP che permette di cifrare un documento. Questo servizio permette di cifrare un documento secondo lo standard PKCS#7 utilizzando le chiavi pubbliche di uno o più certificati.

Method Summary	
byte[]	**cifra**(byte[] toEncrypt, byte[] cert) Cifra il documento toEncrypt utilizzando il certificato cert.
byte[]	**cifraCerts**(byte[] toEncrypt, byte[][] certs) Cifra i documento toEncrypt utilizzando la lista di certificati certs.
CryptoResult[]	**cifraCertsMassiva**(byte[][] toEncrypt, byte[][] certs) Cifra un array di dati toEncrypts utilizzando l'array di certificati passati come parametro certs.
CryptoResult[]	**cifraCertsMassivaUrl**(java.lang.String[] urlToEncrypt, java.lang.String[] urlEncrypted, byte[][] certs) Cifra i dati individuato dall'array di URL urlsToEncrypt utilizzando l'array di certificati passati come parametro certs.
void	**cifraCertsUrl**(java.lang.String urlToEncrypt, java.lang.String urlEncrypted, byte[][] certs) Cifra il documento individuato dall'URL urlToEncrypt utilizzando i certfiicati passati come parametro.
CryptoResult[]	**cifraMassiva**(byte[][] toEncrypt, byte[] cert) Cifra un array di dati toEncrypts utilizzando il certificato passato come parametro cert.
CryptoResult[]	**cifraMassivaUrl**(java.lang.String[] urlToEncrypt, java.lang.String[] urlEncrypted, byte[] cert) Cifra i dati individuato dall'array di URL urlsToEncrypt utilizzando il certificato passato come parametro cert.
byte[]	**cifraP12**(byte[] toEncrypt, java.lang.String nomeP12) Cifra del documento toEncrypt utilizzando un PKCS#12.
byte[]	**cifraP12Default**(byte[] toEncrypt) Cifra del documento toEncrypt utilizzando il PKCS#12 di default.
CryptoResult[]	**cifraP12DefaultMassiva**(byte[][] toEncrypts) Cifra un array di dati toEncrypts utilizzando il PKCS#12 di default.
CryptoResult[]	**cifraP12DefaultMassivaUrl**(java.lang.String[] urlsToEncrypt, java.lang.String[] urlsEncrypted) Cifra i dati individuato dall'array di URL urlsToEncrypt utilizzando il pkcs#12 di default.

void	**cifraP12DefaultUrl**(java.lang.String urlToEncrypt, java.lang.String urlEncrypted) Cifra i dati individuati dall'URL toEncrypt utilizzando il PKCS#12 di default.
CryptoResult[]	**cifraP12Massiva**(byte[][] toEncrypts, java.lang.String nomeP12) Cifra un array di dati toEncrypts utilizzando il PKCS#12 passato come parametro.
CryptoResult[]	**cifraP12MassivaUrl**(java.lang.String[] urlsToEncrypt, java.lang.String[] urlsEncrypted, java.lang.String nomeP12) Cifra i dati individuato dall'array di URL urlsToEncrypt utilizzando il pkcs#12 passato come parametro.
byte[]	**cifraP12Pwd**(byte[] toEncrypt, java.lang.String nomeP12, java.lang.String pwdP12) Cifra del documento toEncrypt utilizzando un PKCS#12.
void	**cifraP12Url**(java.lang.String urlToEncrypt, java.lang.String urlEncrypted, java.lang.String nomeP12) Cifra i dati individuati dall'URL toEncrypt utilizzando un PKCS#12 passato come parametro.
byte[]	**cifraSC**(byte[] toEncrypt, int slot) Cifra il documento toEncrypt utilizzando il dispositivo pkcs#11 corrispondente allo slot slot.
byte[]	**cifraSCAlias**(byte[] toEncrypt, byte[] alias, int slot) Cifra il documento toEncrypt utilizzando il dispositivo pkcs#11 corrispondente allo slot slot e con alias alias.
byte[]	**cifraSCDefault**(byte[] toEncrypt) Cifra il documento toEncrypt utilizzando la SmartCard sullo slot 0.
CryptoResult[]	**cifraSCDefaultMassiva**(byte[][] toEncrypts) Cifra i dati contenuti nell'array toEncrypts utilizzando il dispositivo pkcs#11 di default (slot 0).
CryptoResult[]	**cifraSCDefaultMassivaUrl**(java.lang.String[] urlToEncrypt, java.lang.String[] urlEncrypted) Cifra i dati individuato dall'array di URL urlToEncrypt utilizzando il dispositivo pkcs#11 di default (slot 0).
void	**cifraSCDefaultUrl**(java.lang.String urlToEncrypt, java.lang.String urlEncrypted) Cifra il documento individuato dall'URL urlToEncrypt utilizzando il dispositivo pkcs#11 di default (slot 0).
CryptoResult[]	**cifraSCMassiva**(byte[][] toEncrypts, byte[] alias, int slot) Cifra i dati contenuti nell'array toEncrypts utilizzando il dispositivo pkcs#11 corrispondente allo slot slot e con alias alias.
CryptoResult[]	**cifraSCMassivaUrl**(java.lang.String[] urlToEncrypt, java.lang.String[] urlEncrypted, byte[] alias, int slot) Cifra i dati individuato dall'array di URL urlToEncrypt utilizzando il dispositivo pkcs#11 corrispondente allo slot slot e con alias alias.
void	**cifraSCUrl**(java.lang.String urlToEncrypt, java.lang.String urlEncrypted, byte[] alias, int slot) Cifra il documento individuato dall'URL urlToEncrypt utilizzando il dispositivo pkcs#11 corrispondente allo slot slot e con alias alias.
void	**cifraUrl**(java.lang.String urlToEncrypt, java.lang.String urlEncrypted, byte[] cert) Cifra il documento individuato dall'URL urlToEncrypt utilizzando il certfiicato passato come parametro.

AliasAndCert[]	**getAliasAndCert**(int slot) Restituisce un array di oggetti AliasAndCert contentente tutti gli alias e tutti i certificati di cifra associati, presenti nel dispositivo pkcs#11 corrispondente allo slot slot.

4.2 *Class CallDecifra*

```
java.lang.Object
   └ com.engiweb.axis.ejbcryptoserver.decifra.CallDecifra
```

```
public class CallDecifra
extends java.lang.Object
```

Classe di interfaccia SOAP che permette di decifrare un documento cifrato secondo lo standard PKCS#7 utilizzando la chiave privata contenuta in un file PKCS#12, ovvero utilizzando le chiavi presenti su un dispositivo PKCS#11.

Method Summary

byte[]	**decifraP12**(byte[] toDecrypt, java.lang.String nomeP12) Decifra il documento toDecrypt utilizzando un PKCS#12.
byte[]	**decifraP12Default**(byte[] toDecrypt) Decifra il documento toDecrypt utilizzando il PKCS#12 di default che è il primo della lista.
CryptoResult[]	**decifraP12DefaultMassiva**(byte[][] toDecrypt) Decifra i dati contenuti nell'array toDecrypt utilizzando il PKCS#12 di default.
CryptoResult[]	**decifraP12DefaultMassivaUrl**(java.lang.String[] urlsToDecrypt, java.lang.String[] urlsDecrypted) * Decifra i documenti individuati dll'array di URL urlsToDecrypt utilizzando il PKCS#12 di default.
void	**decifraP12DefaultUrl**(java.lang.String urlToDecrypt, java.lang.String urlDecrypted) Decifra i documenti individuato dall'URL urlToDecrypt utilizzando il PKCS#12 di default.
CryptoResult[]	**decifraP12Massiva**(byte[][] toDecrypt, java.lang.String nomeP12) Decifra i dati contenuti nell'array toDecrypt utilizzando il PKCS#12 passato come parametro.
CryptoResult[]	**decifraP12MassivaUrl**(java.lang.String[] urlsToDecrypt, java.lang.String[] urlsDecrypted, java.lang.String nomeP12) Decifra i documenti individuati dll'array di URL urlsToDecrypt utilizzando il PKCS#12 passato come parametro.
byte[]	**decifraP12Pwd**(byte[] toDecrypt, java.lang.String nomeP12, java.lang.String pwdP12) Decifra il documento toDecrypt utilizzando un PKCS#12.
void	**decifraP12Url**(java.lang.String urlToDecrypt, java.lang.String urlDecrypted, java.lang.String nomeP12) Decifra i documenti individuato dall'URL urlToDecrypt utilizzando il PKCS#12 passato come parametro.
byte[]	**decifraSC**(byte[] toDecrypt, int slot)

	Decifra il documento toDecrypt utilizzando il dispositivo pkcs#11 corrispondente allo slot slot.
byte[]	**decifraSCDefault**(byte[] toDecrypt) Decifra il documento toDecrypt utilizzando la SmartCard sullo slot 0.
CryptoResult[]	**decifraSCDefaultMassiva**(byte[][] toDecrypt) Decifra i dati contenuti nell'array toDecrypt utilizzando il PKCS#11 di default (slot 0).
CryptoResult[]	**decifraSCDefaultMassivaUrl**(java.lang.String[] urlToDecrypt, java.lang.String[] urlDecrypted) Decifra i documenti individuati dagli URL urlToDecrypt utilizzando il PKCS#11 sullo slot di default (slot 0).
void	**decifraSCDefaultUrl**(java.lang.String urlToDecrypt, java.lang.String urlDecrypted) Decifra il documento individuato dall'URL urlToDecrypt utilizzando il dispositivo pkcs#11 di default (slot 0).
CryptoResult[]	**decifraSCMassiva**(byte[][] toDecrypt, int slot) Decifra i dati contenuti nell'array toDecrypt utilizzando il PKCS#11 sullo slot slot.
CryptoResult[]	**decifraSCMassivaUrl**(java.lang.String[] urlToDecrypt, java.lang.String[] urlDecrypted, int slot) Decifra i documenti individuati dagli URL urlToDecrypt utilizzando il PKCS#11 sullo slot slot.
void	**decifraSCUrl**(java.lang.String urlToDecrypt, java.lang.String urlDecrypted, int slot) Decifra il documento individuato dall'URL urlToDecrypt utilizzando il dispositivo pkcs#11 corrispondente allo slot slot.

4.3 Class CallFirma

```
java.lang.Object
  └ com.engiweb.axis.ejbcryptoserver.firma.CallFirma
```

```
public class CallFirma
extends java.lang.Object
```

Classe di interfaccia SOAP che permette di firmare un documento utilizzando la chiave privata estratta da un file PKCS#12 ovvero utilizzando un dispositivo PKCS#11 .
Permette di generare un documento firmato secondo lo standard PKCS#7.

Method Summary	
byte[]	**firmaP12**(byte[] toSign, java.lang.String nomeP12) Firma il documento toSign utilizzando un PKCS#12 passato come parametro.
byte[]	**firmaP12Default**(byte[] toSign) Firma il documento toSign utilizzando il PKCS#12 di default .
CryptoResult[]	**firmaP12DefaultMassiva**(byte[][] toSign) Firma l'array di dati toSign utilizzando il PKCS#12 di default L'amministratore dovrà precedentemente inizializzare i pkcs#12.
void	**firmaP12DefaultUrl**(java.lang.String urlToSign, java.lang.String urlSigned) Firma il documento individuato dall'URL urlToSign utilizzando il PKCS#12 di

	default.
byte[]	**firmaP12Explicit**(byte[] toSign, java.lang.String nomeP12) Firma esplicita del documento toSign utilizzando un PKCS#12 passato come parametro.
byte[]	**firmaP12ExplicitDefault**(byte[] toSign) Firma esplicita del documento toSign utilizzando il PKCS#12 di default.
CryptoResult[]	**firmaP12ExplicitDefaultMassiva**(byte[][] toSign) Firma esplicita dell'array di dati toSign utilizzando il PKCS#12 di default L'amministratore dovrà precedentemente inizializzare i pkcs#12.
CryptoResult[]	**firmaP12ExplicitDefaultMassivaUrl**(java.lang.String[] urlsToSign, java.lang.String[] urlsSigned) Firma esplicita dei documenti individuati dall'array di URL urlsToSign utilizzando il PKCS#12 di default.
void	**firmaP12ExplicitDefaultUrl**(java.lang.String urlToSign, java.lang.String urlSigned) Firma esplicita del documento individuato dall'URL urlToSign utilizzando il PKCS#12 di default.
CryptoResult[]	**firmaP12ExplicitMassiva**(byte[][] toSign, java.lang.String nomeP12) Firma esplicita dei documenti toSign utilizzando un PKCS#12 passato come parametro.
CryptoResult[]	**firmaP12ExplicitMassivaUrl**(java.lang.String[] urlsToSign, java.lang.String[] urlsSigned, java.lang.String nomeP12) Firma esplicita dei documenti individuati dall'array di URL urlsToSign utilizzando il PKCS#12 passato come parametro.
byte[]	**firmaP12ExplicitPwd**(byte[] toSign, java.lang.String nomeP12, java.lang.String pwdP12) Firma esplicita del documento toSign utilizzando un PKCS#12 passato come parametro.
void	**firmaP12ExplicitUrl**(java.lang.String urlToSign, java.lang.String urlSigned, java.lang.String nomeP12) Firma esplicita del documento individuato dall'URL urlToSign utilizzando il PKCS#12 passato come parametro.
CryptoResult[]	**firmaP12Massiva**(byte[][] toSign, java.lang.String nomeP12) Firma l'array di dati toSign utilizzando un PKCS#12 passato come parametro.
CryptoResult[]	**firmaP12MassivaUrl**(java.lang.String[] urlsToSign, java.lang.String[] urlsSigned) Firma i documenti individuati dall'array di URL urlsToSign utilizzando il PKCS#12 di default.
CryptoResult[]	**firmaP12MassivaUrl**(java.lang.String[] urlsToSign, java.lang.String[] urlsSigned, java.lang.String nomeP12) Firma i documenti individuati dall'array di URL urlsToSign utilizzando il PKCS#12 passato come parametro.
byte[]	**firmaP12Pwd**(byte[] toSign, java.lang.String nomeP12, java.lang.String pwdP12) Firma il documento toSign utilizzando un PKCS#12 passato come parametro.
void	**firmaP12Url**(java.lang.String urlToSign, java.lang.String urlSigned, java.lang.String nomeP12) Firma il documento individuato dall'URL urlToSign utilizzando un PKCS#12 passato come parametro.
byte[]	**firmaSC**(byte[] toSign, int slot)

	Firma il documento toSign utilizzando il dispositivo pkcs#11 corrispondente allo slot slot.
byte[]	**firmaSCAlias**(byte[] toSign, byte[] alias, int slot) Firma il documento toSign utilizzando il dispositivo pkcs#11 corrispondente allo slot slot.
byte[]	**firmaSCDefault**(byte[] toSign) Firma il documento toSign utilizzando la SmartCard sullo slot 0.
CryptoResult[]	**firmaSCDefaultMassiva**(byte[][] toSign) Firma i documenti toSign utilizzando il certificato del PKCS#11 sullo slot 0.
void	**firmaSCDefaultUrl**(java.lang.String urlToSign, java.lang.String urlSigned) Firma il documento recuperato dall'URL urlToSign utilizzando il dispositivo pkcs#11 corrispondente allo slot 0 di default.
byte[]	**firmaSCExplicit**(byte[] toSign, int slot) Firma esplicita del documento toSign utilizzando il dispositivo pkcs#11 corrispondente allo slot slot.
byte[]	**firmaSCExplicitAlias**(byte[] toSign, byte[] alias, int slot) Firma esplicita del documento toSign utilizzando il dispositivo pkcs#11 corrispondente allo slot slot.
byte[]	**firmaSCExplicitDefault**(byte[] toSign) Firma esplicita del documento toSign utilizzando la SmartCard sullo slot 0.
CryptoResult[]	**firmaSCExplicitDefaultMassiva**(byte[][] toSign) Firma esplicita dei documenti toSign utilizzando il certificato del PKCS#11 di default (slot 0).
CryptoResult[]	**firmaSCExplicitDefaultMassivaUrl**(java.lang.String[] urlsToSign, java.lang.String[] urlsSigned) Firma esplicita dei documenti recuperati dall'array di URL urlToSign utilizzando il dispositivo pkcs#11 di default (slot 0).
void	**firmaSCExplicitDefaultUrl**(java.lang.String urlToSign, java.lang.String urlSigned) Firma esplicita del documento recuperato dall'URL urlToSign utilizzando il dispositivo pkcs#11 di default (slot 0).
CryptoResult[]	**firmaSCExplicitMassiva**(byte[][] toSign, byte[] alias, int slot) Firma dei documenti toSign utilizzando il dispositivo pkcs#11 corrispondente allo slot slot.
CryptoResult[]	**firmaSCExplicitMassivaUrl**(java.lang.String[] urlsToSign, java.lang.String[] urlsSigned, byte[] alias, int slot) Firma esplicita dei documenti recuperato dall'array di URL urlToSign utilizzando il dispositivo pkcs#11 corrispondente allo slot slot e alias alias.
void	**firmaSCExplicitUrl**(java.lang.String urlToSign, java.lang.String urlSigned, byte[] alias, int slot) Firma esplicita del documento recuperato dall'URL urlToSign utilizzando il dispositivo pkcs#11 corrispondente allo slot slot.
CryptoResult[]	**firmaSCMassiva**(byte[][] toSign, byte[] alias, int slot) Firma i documenti toSign utilizzando il dispositivo pkcs#11 corrispondente allo slot slot.
CryptoResult[]	**firmaSCMassiva**(java.lang.String[] urlsToSign, java.lang.String[] urlsSigned, byte[] alias, int slot) Firma i documenti recuperati dall'array di URL urlToSign utilizzando il

	dispositivo pkcs#11 corrispondente allo slot slot e alias alias.
CryptoResult[]	**firmaSCMassivaUrl**(java.lang.String[] urlsToSign, java.lang.String[] urlsSigned) Firma i documenti recuperato dall'array di URL urlToSign utilizzando il dispositivo pkcs#11 di default (slot 0).
void	**firmaSCUrl**(java.lang.String urlToSign, java.lang.String urlSigned, byte[] alias, int slot) Firma il documento recuperato dall'URL urlToSign utilizzando il dispositivo pkcs#11 corrispondente allo slot slot.
AliasAndCert[]	**getAliasAndCert**(int slot) Restituisce un array di oggetti AliasAndCert contentente tutti gli alias e tutti i certificati di cifra associati, presenti nel dispositivo pkcs#11 corrispondente allo slot slot.

4.4 *Class CallFirmaXml*

```
java.lang.Object
  └ com.engiweb.axis.ejbcryptoserver.xml.CallFirmaXml
```

```
public class CallFirmaXml
extends java.lang.Object
```

Classe di interfaccia SOAP che permette di firmare un documento XML utilizzando la chiave privata estratta da un file PKCS#12. Permette di generare un documento XML firmato. Permette anche di eseguire la verifica strutturale di un documento XML firmato.

Method Summary	
byte[]	**firmaXml**(byte[] xml) Esegue la firma di un documento XML, restituendo il documento XML firmato utilizzando il PKCS#12 di default.
byte[]	**firmaXml**(byte[] xml, java.lang.String nomeP12) Esegue la firma di un documento XML, restituendo il documento XML firmato.
byte[]	**firmaXmlPwd**(byte[] xml, java.lang.String nomeP12, java.lang.String pwdP12) Esegue la firma di un documento XML, restituendo il documento XML firmato.
EsitoVerifica	**verificaXml**(byte[] xmlSigned) Esegue la verifica strutturale del documento XML firmato.

4.5 *Class CallSearchLDAP*

```
java.lang.Object
  └ com.engiweb.axis.ejbcryptoserver.ldap.CallSearchLDAP
public class CallSearchLDAP
extends java.lang.Object
```

Classe di interfaccia SOAP che permette di ricercare su un LDAP i certificati di CA o i certificati utente.

Method Summary	
java.lang.String[]	**fetchBaseDN**(java.lang.String ldapUrl, int ldapPort, int version) Restituisce un array di stringhe contenenti tutti i baseDN (ad es. c=IT) trovati per il server ldap richiesto.

PropertiesCert[]	**searchAllCACertificate**(java.lang.String ldapURL, int ldapPort, java.lang.String stPoint, int version) Ricerca su LDAP i certificati di CA.
PropertiesCert[]	**searchCACertificate**(java.lang.String ldapURL, int ldapPort, java.lang.String stPoint, java.lang.String cn, int version) Ricerca su LDAP i certificati di CA.
PropertiesCert[]	**searchUserCertificate**(java.lang.String ldapURL, int ldapPort, java.lang.String stPoint, java.lang.String cn, int version) Ricerca su LDAP i certificati utente dato un certo CN.

4.6 Class CallTimeStamp

```
java.lang.Object
  └ com.engiweb.axis.tsabaltimore.CallTimeStamp
```

```
public class CallTimeStamp
extends java.lang.Object
```

Classe di interfaccia SOAP che permette di eseguire una richiesta di marcatura temporale con la TSA BALTIMORE 2.03. Permette di verificare il token della marca temporale.

Method Summary	
byte[]	**createRequest**(byte[] data, java.lang.String tsPolicyId) Costruisce un token di richiesta di TimeStamp standard con i dati passati come parametro.
byte[]	**createRequestUrl**(java.lang.String urlData, java.lang.String tsPolicyId) Costruisce un token di richiesta di TimeStamp standard con i dati recuperati dall'URl passato come parametro urlData.
byte[]	**makeRequestServlet**(byte[] data, java.lang.String tssUrl, java.lang.String tsPolicyId, java.lang.String nomeP12) Esegue una richiesta di TimeStamp tramite una servlet.
byte[]	**makeRequestServlet**(java.lang.String urlData, java.lang.String tssUrl, java.lang.String tsPolicyId, java.lang.String nomeP12) Esegue una richiesta di TimeStamp tramite una servlet.
byte[]	**makeRequestServletDefault**(byte[] data, java.lang.String tssUrl, java.lang.String tsPolicyId) Esegue una richiesta di TimeStamp tramite una servlet.
CryptoResult[]	**makeRequestServletDefaultMassiva**(byte[][] urlData, java.lang.String tssUrl, java.lang.String tsPolicyId)
CryptoResult[]	**makeRequestServletDefaultMassivaUrl**(java.lang.String[] urlData, java.lang.String tssUrl, java.lang.String tsPolicyId)
CryptoResult[]	**makeRequestServletMassiva**(byte[][] urlData, java.lang.String tssUrl, java.lang.String tsPolicyId, java.lang.String nomeP12)
CryptoResult[]	**makeRequestServletMassivaUrl**(java.lang.String[] urlData, java.lang.String tssUrl, java.lang.String tsPolicyId, java.lang.String nomeP12)

byte[]	**makeRequestServletPwd**(byte[] data, java.lang.String tssUrl, java.lang.String tsPolicyId, java.lang.String nomeP12, java.lang.String pwdP12) Esegue una richiesta di TimeStamp tramite una servlet.
byte[]	**makeRequestServletRetry**(byte[] data, java.lang.String tssUrl, java.lang.String tsPolicyId, java.lang.String nomeP12, int retry, int delay, int timeOut) Dato il documento data al quale si vuole apporre la marca temporale viene generata una richiesta standard di TimeStamp che viene poi inviata alla TSA.
byte[]	**makeRequestServletSC**(byte[] data, java.lang.String tssUrl, java.lang.String tsPolicyId, byte[] alias, int slot) Esegue una richiesta di TimeStamp tramite una servlet.
byte[]	**makeRequestServletUrl**(java.lang.String urlData, java.lang.String tssUrl, java.lang.String tsPolicyId) Esegue una richiesta di TimeStamp tramite una servlet.
byte[]	**makeRequestStandard**(byte[] data, java.lang.String tssUrl, int port, java.lang.String tsPolicyId) Effettua una richiesta di TimeStamp standard.
CryptoResult[]	**makeRequestStandardMassiva**(byte[][] data, java.lang.String tssUrl, int port, java.lang.String tsPolicyId)
CryptoResult[]	**makeRequestStandardMassivaUrl**(java.lang.String[] urlData, java.lang.String tssUrl, int port, java.lang.String tsPolicyId)
byte[]	**makeRequestStandardUrl**(java.lang.String urlData, java.lang.String tssUrl, int port, java.lang.String tsPolicyId) Effettua una richiesta di TimeStamp standard.
byte[]	**sendTimeStampRequest**(byte[] asn1Request, java.lang.String tssUrl, int port) Invia il token di richiesta di timeStamp alla TSA.
byte[]	**sendTimeStampRequestServlet**(byte[] asn1Request, java.lang.String tssUrl, java.lang.String nomeP12) Invia il token di richiesta di timeStamp alla TSA, tramite una servlet.
TokenDetails	**verifyToken**(byte[] tokenToVerify) Verifica la correttezza delle marche temporati.
TokenDetails	**verifyTokenAndDoc**(byte[] fileToVerify, byte[] tokenToVerify) Verifica la correttezza delle marche temporati.
TokenDetails	**verifyTokenAndDocCRL**(byte[] fileToVerify, byte[] tokenToVerify, boolean download) Verifica la correttezza delle marche temporati.
TokenDetails	**verifyTokenAndDocCRLUrl**(java.lang.String urlFileToVerify, byte[] tokenToVerify, boolean download) Verifica la correttezza delle marche temporati.
TokenDetails	**verifyTokenAndDocTrust**(byte[] fileToVerify, byte[] tokenToVerify) Verifica la correttezza delle marche temporati.
TokenDetails	**verifyTokenAndDocTrust**(java.lang.String urlFileToVerify, byte[] tokenToVerify) Verifica la correttezza delle marche temporati.
TokenDetails	**verifyTokenAndDocUrl**(java.lang.String urlFileToVerify, byte[] tokenToVerify) Verifica la correttezza delle marche temporati.

TokenDetails	**verifyTokenCRL**(byte[] tokenToVerify, boolean download) Verifica la correttezza delle marche temporati.
TokenDetails	**verifyTokenTrust**(byte[] tokenToVerify) Verifica la correttezza delle marche temporati.

4.7 Class CallVerifica

```
java.lang.Object
  └ com.engiweb.axis.ejbcryptoserver.verifica.CallVerifica
```

```
public class CallVerifica
extends java.lang.Object
```

Classe di interfaccia SOAP che permette di effettuare ogni tipo di verifica su un documento firmato secondo lo standard PKCS#7. E' possibile eseguire la verifica strutturale del documento e la verifica della credibilità e delle informazioni di revoca dei certificati dei firmatari.

Method Summary

DetailsCrl	**downloadCRLFromCert**(byte[] cert) Esegue il download delle informazioni di revoca dal certificato passato come parametro e le salva nel CertStore.
DetailsCrl	**downloadCRLFromLdap**(java.lang.String ldapURL, int ldapPort, java.lang.String stPointDefault, java.lang.String stPointIssuerDN, java.lang.String caDN) Esegue il download delle informazioni di revoca dato l'URL dell'LDAP dal quale scaricarle e il CN del certificato di CA con il quale eseguire la ricerca e le salva all'interno del CertStore.
DetailsCrl	**downloadCRLFromUri**(java.lang.String url) Esegue il download delle informazioni di revoca dall'URL passato come parametro e le salva nel CertStore.
DetailsCrl	**getCRLDetails**(byte[] crl) Restituisce un oggetto contenente le proprietà principali di una CRL data la crl.
void	**setDownloadCRL**(boolean download) Attiva e disattiva il download delle CRL Se non viene chiamato si controllano le CRL nel CertStore.
void	**verificaCert**(int typeVerify, byte[] cert) Esegue la verifica del certificato cert.
EsitoVerificaCert	**verificaCert**(int typeVerify, byte[] cert, boolean download) Esegue la verifica del certificato cert.
EsitoVerificaCert	**verificaCert**(int typeVerify, byte[] cert, boolean download, java.lang.String date) Esegue la verifica del certificato cert.
void	**verificaCert**(int typeVerify, byte[] cert, java.lang.String date) Esegue la verifica del certificato cert.
void	**verificaCertList**(int typeVerify, byte[] cert, byte[][] certsCA) Esegue la verifica del certificato cert rispetto alla lista di certificati di CA del CertStore passata come parametro certsCA.
void	**verificaCertTrust**(byte[] cert, byte[] certCA)

	Controlla se il certificato cert è sicuro rispetto al certificato di CA certCA.
void	**verificaCRLCertFromLdap**(byte[] cert, java.lang.String ldapURL, int ldapPort, java.lang.String stPointDefault) Verifica la validità del certificato scaricando le CRL dall'URL dell'LDAP indicato come parametro.
EsitoVerifica	**verificaP7**(byte[] toVerify) Esegue la verifica d'integrità del documento toVerify (PKCS#7 implicit)
EsitoVerifica	**verificaP7Explicit**(byte[] toVerify, byte[] originalData) Verifica esplicita del documento firmato toVerify (PKCS#7 explicit), rispetto ai dati originari originalData.
EsitoVerifica[]	**verificaP7ExplicitMassiva**(byte[][] toVerify, byte[][] originalData) Esegue la verifica d'integrità dei dati dell'array toVerify (PKCS#7 IMPLICIT)
EsitoVerifica[]	**verificaP7ExplicitMassivaUrl**(java.lang.String[] urlToVerify, java.lang.String[] originalData) Esegue la verifica d'integrità dei dati dell'array toVerify (PKCS#7 IMPLICIT)
EsitoVerifica	**verificaP7ExplicitUrl**(java.lang.String urlToVerify, java.lang.String urlOriginalData) Esegue la verifica d'integrità esplicita del documento indicato dall'URL urlToVerify (PKCS#7 EXPLICIT) rispetto ai dati originari individuati dall'URL urlOriginalData..
EsitoVerifica[]	**verificaP7Massiva**(byte[][] toVerify) Esegue la verifica d'integrità dei dati dell'array toVerify (PKCS#7 IMPLICIT)
EsitoVerifica[]	**verificaP7MassivaUrl**(java.lang.String[] urlToVerify, java.lang.String[] urlVerified) Esegue la verifica d'integrità dei dati dell'array toVerify (PKCS#7 IMPLICIT)
EsitoVerifica	**verificaP7PDF**(byte[] pdfToVerify) Esegue la verifica d'integrità del PDF firmato pdfToVerify
EsitoVerifica	**verificaP7PDFUrl**(java.lang.String urlPdfToVerify) Esegue la verifica d'integrità del PDF firmato indicato dall'URL urlPdfToVerify.
EsitoVerifica	**verificaP7Url**(java.lang.String urlToVerify, java.lang.String urlVerified) Esegue la verifica d'integrità del documento indicato dall'URL urlToVerify (PKCS#7 IMPLICIT).

5 Esempi di utilizzo dei web services

5.1 Esempio di utilizzo del web service di verifica (Java)

Di seguito un esempio di codice Java per richiamare il web service di Verifica.

```
package test;

import com.engiweb.axis.cryptoserver.verifica.CallVerifica;
import com.engiweb.axis.cryptoserver.verifica.CallVerificaServiceLocator;
import com.engiweb.axis.cryptoserver.verifica.EsitoVerifica;
import com.engiweb.axis.cryptoserver.utility.PropertiesCert;

import java.io.FileInputStream;
```

```
import java.io.File;
import java.net.URL;

public class Test {
  String host = "http://stefania-v:8080/axis/services/";

  // Variabili indicanti le possibili verifiche

  /** Controlla la validità temporale del certificato */
  public static final int V_DATE = 1;

  /** Controlla il trust del certificato */
  public static final int V_TRUST = 2;

  /** Controlla le crl del certificato */
  public static final int V_CRL = 4;

  /** Controlla la validità temporale e il trust del certificato */
  public static final int V_DATE_TRUST = 3;

  /** Controlla la validità temporale,il trust e le crl del certificato */
  public static final int V_DATE_TRUST_CRL = 7;

  public Test() throws Exception {
          verifica();
  }

  public static void main(String[] args) throws Exception {
          Test test1 = new Test();
  }

  public void verifica() throws Exception {
          byte[] toVerify = null;

          // Recuperare il file da verificare
          try {
                  File file = new File("firmato.p7m");
                  FileInputStream fi = new FileInputStream(file);

                  Long lenght = new Long(file.length());
                  int len = lenght.intValue();
                  toVerify = new byte[len];

                  fi.read(toVerify);
          } catch (Exception e) {
                  return;
          }

          try {
                  // Recupero la classe di interfaccia SOAP con l'EJB di Verifica del
                  // CryptoServer
                  CallVerifica verify = (new CallVerificaServiceLocator())
                                  .getWSCryptoServerAxisVerifica(new URL(host
                                                  + "WSCryptoServerAxisVerifica?wsdl"));

                  byte[] byteVerified = null;
```

```
                    // Verifica integrità
                    EsitoVerifica result = verify.verificaP7(toVerify);

                    // Recupero il numero dei firmatari
                    int numCert = result.getNumCertificate();
                    System.out.println("Recupero firmatari: " + numCert);

                    PropertiesCert prop = null;
                    for (int i = 0; i < numCert; i++) {

                            // recupero i dati del certiticato del firmatario
                            prop = result.getCertificates()[i];

                            System.out.println("firmatario:\n" + prop.getSubjectDN());
                            boolean isVerified = result.getVerifiedSigners()[i];
                            System.out.println("Verifica: " + isVerified);

                            // Se la verifica d'integrità è OK controllo il certificato
                            if (isVerified) {
                                    // recupero il certificato
                                    byte[] cert = prop.getCertBytes();

                                    // Verifica di trust e crl
                                    verify.verificaCert(this.V_DATE_TRUST_CRL, cert);

                                    System.out.println("Verifica eseguita correttamente");
                            }
                            System.out.println("*****************************\n");
                    }
            } catch (Exception e) {
                    System.out.println("Errore nella verifica: " + e.getMessage());
                    return;
            }

    }
}
```

Allegato 13: Specifiche tecniche per l'utilizzo dei web services di consultazione dei dati di personale e strutture

1 Introduzione

Questo documento ha lo scopo di descrivere l'utilizzo dei componenti in RER.Tools.Organigramma, namespace contenente le classi che permettono l'accesso ai dati dell'organigramma regionale. Tali dati risiedono attualmente sul mainframe, ma è stata predisposta una replica da DB2 a SQL Server. Le classi di RER.Tools.Organigramma accedono ai dati presenti su SQL Server.
La configurazione dei parametri d'accesso al DB è eseguita al livello di machine.config dei singoli web server della filiera .NET, pertanto ogni applicazione installata su tali web server può utilizzare tali classi senza fare altro che aggiungere il riferimento alla DLL corrispondente.

1.1 Web services

Una sintesi delle funzioni disponibili tramite la class library RER.Tools.Organigramma è fruibile anche attraverso un web service. Si veda il capitolo 3.2 ("Specifiche per l'accesso al web service") per i riferimenti e le specifiche.

1.2 Descrizione generale

Le entità che riguardano l'organigramma e su cui è basata la libreria delle classi si possono sintetizzare in:

- Unità funzionali: strutture apicali (direzioni/agenzie/isistuti/ecc.), servizi, posizioni di responsabilità, …
- Persone: i dipendenti regionali o comunque chi ha un rapporto di lavoro con la Regione
- Caratteristiche: si tratta di "attributi" che possono essere associati alle unità funzionali (per esempio, tutte le direzioni generali hanno la caratteristica "DIG", i servizi hanno la caratteristica "SER", …)

I servizi forniti sono essenzialmente i seguenti:

- dettaglio delle informazioni relative ad una unità funzionale;
- elenco delle unità funzionali che soddisfano certi parametri di ricerca;
- dettaglio delle informazioni relative ad una persona;
- risposte a domande tipo "una persona appartiene a un struttura?", oppure "c'è un legame di parentela tra due unità funzionali?", …

1.3 Storicizzazione

L'organigramma regionale è una struttura che muta nel tempo, perciò qualsiasi estrazione di dati da tale struttura deve sempre essere relativa ad un ben preciso istante. Pertanto tutte le chiamate di metodi hanno sempre come parametro la *data di riferimento*, anche se per comodità sono stati aggiunti gli overload in cui non è necessario passare la data di riferimento che, in questo caso, è fissata con la data corrente.

NB: nel DB del mainframe, le "date fine" contengono il valore "31/12/2049" per indicare che non c'è una data di fine. Invece, all'interno di questa libreria di classi, tali valori sono stati sostituiti con il valore DateTime.MaxValue.

2 Classi principali

Le classi principali implementate sono le seguenti:

- **UnitaFunzionale**: contiene i dati di un'unità funzionale
- **Persona**: contiene i dati di una persona

Per quanto riguarda UnitaFunzionale e Persona, si sono implementate anche le corrispondenti classi **UnitaFunzionaleCollection** e **PersonaCollection**. Infine esiste anche la classe **Caratteristica** e la

relativa **CaratteristicaCollection.**

2.1 UnitaFunzionale

Di seguito si riporta la definizione della classe (senza implementazione). Si veda anche il capitolo per come creare istanze di questa classe.

```
[Serializable]
public class UnitaFunzionale
{
        public DateTime DataRif;

        public int Livello;
        public string Cod;
        public DateTime DataInizio;
        public DateTime DataFine;
        public string Descrizione;
        public string CodUnitaPadre;
        public int MatricolaResponsabile;

        public static string NullCod;
        public static int NullLivello;

        public PersonaCollection ElencaPersoneAssegnate();
        public UnitaFunzionaleCollection ElencaDiscendentiDiretti();
        public CaratteristicaCollection ElencaCaratteristiche();
        public bool HaLeCaratteristiche(string[] caratteristiche,
        bool richiediTutteLeCaratteristiche);
        public bool HaLaCaratteristica(string codCaratteristica);

        public bool DiscendeDa(string codUnitaAntenato, int livelloMax);
        public bool DiscendeDa(string codUnitaAntenato);

        public int CalcolaLivelliParentela(string codUnitaAntenato, int livelloMax);
        public int CalcolaLivelliParentela(string codUnitaAntenato);

} // end class UnitaFunzionale
```

Descrizione attributi

I primi attributi rappresentano il dettaglio dei dati dell'unità funzionale. Il campo Livello rappresenta a che livello, nella gerarchia delle unità funzionali si trova quella specifica unità funzionale, in verità questa informazione non è sempre presente, in tal caso sarà indicato. Nel caso non sia presente, l'attributo Livello è valorizzato con il valore dell'attributo statico NullLivello.
Analogamente è stato definito l'attributo statico NullCod, utile in tutti i casi in si vuole indicare il concetto di "nessuna unità funzionale" (per esempio, l'attributo CodUnitaPadre vale NullCod nel caso in cui l'unità funzionale non abbia nessun padre, oppure: se tra i parametri di un metodo di ricerca si può indicare il codice di un'unità funzionale, ma in verità non si vuole fare nessuna scelta, si può utilizzare UnitaFunzionale.NullCod).

Descrizione metodi

Relativamente ai metodi, non c'è molto da dire, i nomi sono "parlanti", solo qualche precisazione:

1. essendo metodi di "istanza" non accettano parametri. Quindi, per esempio, la chiamata al metodo "ElencaPersoneAssegnate" significa implicitamente "elenca le persone assegnate all'unità funzionale in data DataRif".

2. il parametro LivelloMax (dei metodi DiscendeDa e CalcolaLivelliParentela) funziona come un limitatore, ovverosia “DiscendaDa” limita il controllo alle unita funzionali entro Livello Max livelli.
3. CalcolaLivelliParentela calcola la distanza tra l’unità funzionale e un possibile antenato (codUnitaAntenato), se non c’è parentela (oppure è superiore a LivelloMax) il metodo ritorna il valore UnitaFunzionale.NullLivello.

2.2 Persona

Di seguito si riporta la definizione della classe (senza implementazione). Si veda anche il capitolo per come creare istanze di questa classe.

```
[Serializable]
public class Persona
{
        public DateTime DataRif;

        public int Matricola;
        public string Nome;
        public string Cognome;
        public string CodiceFiscale;
        public string Sesso;
        public DateTime DataNascita;
        public string CodCatastoNascita;
        public string Email;
        public string StatoCivile;
        public string CognomeAcquisito;
        public string ResidenzaIndirizzo;
        public string ResidenzaCap;
        public string ResidenzaCodComune;
        public string ResidenzaFrazione;
        public string CodUnitaAppartenenza;
        public string CodUnitaResponsabilita;
        public DateTime DataAssunzione;
        public DateTime DataCessazione;

        public static int NullMatricola;
        public static DateTime NullDataNascita;
        public static DateTime NullDataAssunzione;
        public static DateTime NullDataCessazione;

        public bool IsAssegnataAUnitaFunzionale(string codUnita);
        public bool IsResponsabileUnitaFunzionale(string codUnita);
}
```

Descrizione attributi

I nomi degli attributi dovrebbero essere sufficienti a spiegare il contenuto. Si noti solo che sono presenti degli attributi statici che contengono valori “particolari”: quelli con prefisso “Null”. Come per l’unità funzionale, sono stati definiti per gestire i casi di “mancanza di dati”. Per esempio se la persona non ha data di cessazione allora il corrispondente campo avrà come valore Persona.NullDataCessazione. Stesso discorso, ma questa volta riferito all’unità funzionale, per l’attributo CodUnitaResponsabilita, nel caso la persona non sia responsabile di nessuna struttura, allora l’attributo è valorizzato on UnitaFunzionale.NullCod.

Descrizione metodi

Come per l’unità funzionale, relativamente ai metodi, non c’è molto da dire, i nomi sono “parlanti”. Vale

sempre la precisazione che, essendo metodi di “istanza”, perciò la chiamata al metodo “IsAssegnataAUnitaFunzionale” significa implicitamente “la persona è assegnata alla struttura (passata come parametro) in data DataRif?”.

3 Classi "creazionali" e di utilità

Finora si è vista la struttura delle classi fondamentali, in verità esse rappresentano la base per costruire le informazioni restituite dai metodi delle seguenti classi che sono puramente “statiche”, in quanto implementano solo “servizi”: di creazione delle classi basi e ricerca vera a propria nell’organigramma.

Un paio di precisazioni:

1. di ogni metodo, per comodità, sono stati definiti diversi overload che permettono di evitare di indicare certi parametri, i quali assumono quindi dei valori di default, in questo documento è indicato solo la versione “completa”;
2. riguardo al parametro *dataRif* (la data riferimento), come già accennato nel paragrafo 1.3, se non indicato si assume la data corrente (tale fatto non sarà ulteriormente precisato nei dettagli dei singoli metodi).

3.1 UnitaFunzionaleManager

```
UnitaFunzionale Crea(
        string codUnita,
        DateTime dataRif)
```

Crea un’istanza dell’unità funzionale

```
UnitaFunzionaleCollection Elenca(
        DateTime dataRif,
        string[] caratteristiche,
        bool richiediTutteLeCaratteristiche)
```

Elenca le unità funzionali aventi le caratteristiche specificate (tutte o almeno una a seconda del valore di 'richiediTutteLeCaratteristiche’.
NB: Il valore dell’attributo Livello delle unità funzionali presenti nella collection in questo caso non è valorizzato.

```
UnitaFunzionaleCollection ElencaDiscendentiDiretti(
        string codUnitaRadice,
        DateTime dataRif,
        string[] caratteristiche,
        bool richiediTutteLeCaratteristiche)
```

Elenca le unità funzionali discendenti direttamente dall'unità funzionale codUnitaRadice (le “figlie”) aventi le caratteristiche specificate (tutte o almeno una a seconda del valore di 'richiediTutteLeCaratteristiche’

```
UnitaFunzionaleCollection ElencaDiscendentiInProfondita(
        string codUnitaRadice,
        DateTime dataRif,
        string[] caratteristiche,
        bool richiediTutteLeCaratteristiche)
```

Elenca le unità funzionali discendenti direttamente o indirettamente dall'unità funzionale codUnitaRadice aventi le caratteristiche specificate (tutte o almeno una a seconda del valore di

'richiediTutteLeCaratteristiche'

```
UnitaFunzionaleCollection ElencaDiscendenti(
        string codUnitaRadice,
        int livelloMax,
        DateTime dataRif,
        string[] caratteristiche,
        bool richiediTutteLeCaratteristiche)
```

Elenca le unità funzionali discendenti fino al livello livelloMax dall'unità funzionale codUnitaRadice aventi le caratteristiche specificate (tutte o almeno una a seconda del valore di 'richiediTutteLeCaratteristiche'

```
PersonaCollection ElencaPersone(
        string codUnita,
        DateTime dataRif)
```

Elenca le persone assegnate all'unità funzionale

```
bool DiscendeDa(
        string codUnita,
        string codUnitaAntenato,
        DateTime dataRif,
        int livelloMax)
```

Calcola se c'è un legame di parentela con non più di 'livelloMax' livelli tra l'unita codUnita e codUnitaAntenato (codUnita deve discendere da codUnitaAntento). Assegnare UnitaFunzionale.NullLivello a 'livelloMax' per eseguire la ricerca su tutti i livelli possibili

```
bool FigliaDi(
        string codUnita,
        string codUnitaPadre,
        DateTime dataRif)
```

Calcola se un'unità funzionale codUnita discende direttamente da codUnitraPadre

```
int CalcolaLivelliParentela(
        string codUnita,
        string codUnitaAntenato,
        DateTime dataRif,
        int livelloMax)
```

Calcola la 'distanza', in 'livelli di parentela', tra l'unità funzionale codUnita e un suo antenato (codUnitaAntenato). Assegnare UnitaFunzionale.NullLivello a 'livelloMax' per eseguire la ricerca su tutti i livelli possibili. Se le due unità non sono parenti o la distanza è maggiore di 'livelloMax' restituisce UnitaFunzionale.NullLivello

```
CaratteristicaCollection ElencaCaratteristiche(
        string codUnita,
        DateTime dataRif)
```

Elenca le caratteristiche assegnate all'unità funzionale

```
bool HaLeCaratteristiche(
        string codUnita,
        DateTime dataRif,
        string[] caratteristiche,
        bool richiediTutteLeCaratteristiche)
```

Calcola se l'unità funzionale ha le caratteristiche specificate (tutte o almeno una a seconda del valore di 'richiediTutteLeCaratteristiche'

```
bool HaLaCaratteristica(
        string codUnita,
        DateTime dataRif,
        string caratteristica)
```

Calcola se l'unità funzionale ha la caratteristica specificata

```
UnitaFunzionale CercaAntenatoConCaratteristica(
        string codUnita,
        DateTime dataRif,
        string caratteristica,
        int distanzaMax)
```

Restituisce, se esiste, la prima unita funzionale (tra gli antenati, scorrendoli a ritroso, fino a 'distanzaMax' livelli) che ha le caratteristiche specificate (tutte o almeno una a seconda del valore di 'richiediTutteLeCaratteristiche'). Assegnare UnitaFunzionale.NullLivello a 'distanzaMax' per eseguire la ricerca fino alla radice. Se non esiste nessuna unita funzionale che ha i requisiti restituisce *null*

3.2 Persona Manager

```
Persona Crea(
        int matricola,
        DateTime dataRif)
```

Crea un'istanza della persona

```
bool IsResponsabileUnitaFunzionale(
        int matricola,
        string codUnita,
        DateTime dataRif)
```

Calcola se la persona è il responsabile dell'unità funzionale

```
bool IsAssegnataAUnitaFunzionale(
        int matricola,
        string codUnita,
        DateTime dataRif)
```

Calcola se la persona è assegnata all'unità funzionale

4 Gestione delle gerarchie

La classe **UnitaFunzionaleGerarchia** (e la sua controparte **UnitaFunzionaleGerarchiaManager**) contiene i dati di una (porzione) di gerarchia dell'organigramma ed è stata introdotta per gestire efficientemente e efficacemente[21] i risultati di ricerche sulle unità funzionali per cui si vuole un output "gerarchico" (utile per una navigazione o rappresentazione dei dati tramite un "albero")

[21] Di fatto, la chiamata a questo metodo, esegue un solo accesso al DB, per costruire l'intera gerarchia

La classe contiene sono un sottoinsieme minimale degli attributi delle unità funzionali:

```
public class UnitaFunzionaleGerarchia
{
        public string Cod;
        public string Descrizione;
        public UnitaFunzionaleGerarchiaCollection Discendenti;
}
```

Per creare una gerarchia utilizzare il seguente metodo statico della classe UnitaFunzionaleGerarchiaManager:

```
UnitaFunzionaleGerarchia Crea(
        string codUnitaRadice,
        int livelloMax,
        DateTime dataRif)
```

Crea la gerarchia completa dei discendenti dell'unità con non più di 'livelloMax' livelli di parentela. Assegnare UnitaFunzionale.NullLivello a 'livelloMax' per creare la gerarchia su tutti i livelli possibili

5 Esempi d'uso

```
foreach(UnitaFunzionale u in UnitaFunzionaleManager.Elenca())
  Console.WriteLine("{0} | {1} | {2} | {3}", u.Cod, u.Descrizione, u.CodUnitaPadre,
u.MatricolaResponsabile);

foreach(UnitaFunzionale u in UnitaFunzionaleManager.Elenca(new DateTime(2006,08,21), "SER"))
  Console.WriteLine("{0} | {1} | {2} | {3}", u.Cod, u.Descrizione, u.CodUnitaPadre,
u.MatricolaResponsabile);

foreach(Persona p in UnitaFunzionaleManager.ElencaPersone("00000315", new
DateTime(2005,12,31)))
  Console.WriteLine("{0} | {1} | {2} | {3}", p.Matricola, p.Cognome, p.Nome, p.CodiceFiscale);

foreach(UnitaFunzionale u in UnitaFunzionaleManager.Elenca(new DateTime(2005,12,31), new
string[] {"ENT", "IST", "DIG"}, false))
  Console.WriteLine("{0} | {1} | {2} | {3}", u.Cod, u.Descrizione, u.CodUnitaPadre,
u.MatricolaResponsabile);

foreach(UnitaFunzionale u in UnitaFunzionaleManager.Elenca(new DateTime(2005,12,31), new
string[] {"DIG", "BUD", "ORG"}, true))
  Console.WriteLine("{0} | {1} | {2} | {3}", u.Cod, u.Descrizione, u.CodUnitaPadre,
u.MatricolaResponsabile);

foreach(UnitaFunzionale u in UnitaFunzionaleManager.Elenca(new DateTime(2005,12,31), new
string[] {"SER"}, true))
  Console.WriteLine("{0} | {1} | {2} | {3}", u.Cod, u.Descrizione, u.CodUnitaPadre,
u.MatricolaResponsabile);

Console.WriteLine(PersonaManager.IsResponsabileUnitaFunzionale(999999, "00000315", new
DateTime(2005,12,31) ));
Console.WriteLine(PersonaManager.IsAssegnataAUnitaFunzionale(999999, "00000315", new
DateTime(2005,12,31) ));
Console.WriteLine(PersonaManager.IsAssegnataAUnitaFunzionale(999999, "00000315", new
DateTime(2005,12,31) ));
```

```
Console.WriteLine(UnitaFunzionaleManager.DiscendeDa("00000315", "D0000022", new
DateTime(2005,12,31), 1 ));
Console.WriteLine(UnitaFunzionaleManager.DiscendeDa("00000315", "D0000022", new
DateTime(2000,12,31) ));

foreach(UnitaFunzionale u in UnitaFunzionaleManager.ElencaDiscendenti("D0000022", 3, "SER"))
  Console.WriteLine("{0} | {1} | {2} | {3}", u.Cod, u.Descrizione, u.CodUnitaPadre,
u.MatricolaResponsabile);


Persona p = PersonaManager.Crea(999999);
Console.WriteLine(p.Cognome + " " + p.IsResponsabileUnitaFunzionale("00000435"));

UnitaFunzionale u = UnitaFunzionaleManager.Crea("00000315", new DateTime(2005,12,31));
Console.WriteLine("****** PERSONA ASSEGNATE: **********");
foreach(Persona p in u.ElencaPersoneAssegnate())
  Console.WriteLine("{0} | {1} | {2} | {3}", p.Matricola, p.Cognome, p.Nome, p.CodiceFiscale);
Console.WriteLine("****** DISCENDENTI DIRETTI: **********");
foreach(UnitaFunzionale u1 in u.ElencaDiscendentiDiretti())
  Console.WriteLine("{0} | {1} | {2} | {3}", u1.Cod, u1.Descrizione, u1.CodUnitaPadre,
u1.MatricolaResponsabile);
Console.WriteLine("****** CARATTERISTICHE: **********");
foreach(Caratteristica c in u.ElencaCaratteristiche())
  Console.WriteLine("{0} | {1}", c.Cod, c.Descrizione);
Console.WriteLine(u.HaLeCaratteristiche(new string[] {"SER", "DPS"}, false));
Console.WriteLine(u.DiscendeDa("D0000022"));
Console.WriteLine(u.CalcolaLivelliParentela("G0000001"));

Console.WriteLine(UnitaFunzionaleManager.CercaAntenatoConCaratteristica("00000315", new
DateTime(2005,12,31), "DIG").Cod);
```

6 Specifiche per l'accesso al web service

L' "end-point" a cui fare riferimento per l'utilizzo del web service è il seguente:
http://intraservizi.regione.emilia-romagna.it/WebServices/Organigramma/Main.asmx

Il file WSDL corrispondente è scaricabile dal seguente indirizzo:
http://intraservizi.regione.emilia-romagna.it/WebServices/Organigramma/Main.asmx?WSDL

6.1 "Schemi" XML

Le strutture dati utilizzate dal servizio sono prima di tutto schematizzate all'interno del file WSDL, comunque si riportano di seguito, per ulteriore chiarezza, i frammenti XML corrispondenti a tali strutture dati.

Persona

```
<Persona>
  <DataRif></DataRif>
  <Matricola></Matricola>
  <Nome></Nome>
  <Cognome></Cognome>
  <CodiceFiscale></CodiceFiscale>
  <CodUnitaAppartenenza></CodUnitaAppartenenza>
  <CodUnitaResponsabilita></CodUnitaResponsabilita>
  <DataAssunzione></DataAssunzione>
  <DataCessazione></DataCessazione>
</Persona>
```

Unità funzionale

```
<UnitaFunzionale>
  <DataRif></DataRif>
  <Livello></Livello>
  <Cod></Cod>
  <DataInizio></DataInizio>
  <DataFine></DataFine>
  <Descrizione></Descrizione>
  <CodUnitaPadre></CodUnitaPadre>
  <MatricolaResponsabile></MatricolaResponsabile>
</UnitaFunzionale>
```

Unità funzionale gerarchia

```
<UnitaFunzionaleGerarchia>
  <Cod></Cod>
  <Descrizione></Descrizione>
  <Discendenti>
        <UnitaFunzionaleGerarchia>
              <Cod></Cod>
              <Descrizione></Descrizione>
              <Discendenti>
                    ...
              </Discendenti>
        </UnitaFunzionaleGerarchia>
        <UnitaFunzionaleGerarchia>
              <Cod></Cod>
              <Descrizione></Descrizione>
              <Discendenti>
                    ...
              </Discendenti>
        </UnitaFunzionaleGerarchia>
        <UnitaFunzionaleGerarchia>
              <Cod></Cod>
              <Descrizione></Descrizione>
              <Discendenti>
                    ...
              </Discendenti>
        </UnitaFunzionaleGerarchia>

        ...

  </Discendenti>
</UnitaFunzionaleGerarchia>
```

6.2 Dettaglio delle funzioni

Di seguito si riportano le specifiche delle singole operazioni esposte tramite il web service.

Unità Funzionale

```
UnitaFunzionale DettaglioUnitaFunzionale(
        string codUnita,
        DateTime dataRif
        )
```

Carica i dati base di una unità funzionale

```
PersonaCollection ElencaPersoneUnitaFunzionale(
        string codUnita,
        DateTime dataRif
        )
```

Elenca le persone assegnate all'unità funzionale

```
CaratteristicaCollection ElencaCaratteristicheUnitaFunzionale(
        string codUnita,
        DateTime dataRif
        )
```

Elenca le caratteristiche assegnate all'unità funzionale

```
bool UnitaFunzionaleHaLeCaratteristiche(
        string codUnita,
        DateTime dataRif,
        string[] caratteristiche,
        bool richiediTutteLeCaratteristiche)
```

Calcola se l'unità funzionale ha le caratteristiche specificate (tutte o almeno una a seconda del valore di 'richiediTutteLeCaratteristiche')

```
UnitaFunzionale UnitaFunzionaleCercaAntenatoConCaratteristiche(
        string codUnita,
        DateTime dataRif,
        string[] caratteristiche,
        bool richiediTutteLeCaratteristiche,
        int distanzaMax)
```

Restituisce, se esiste, la prima unita funzionale (tra gli antenati, scorrendoli a ritroso, fino a 'distanzaMax' livelli) che ha le caratteristiche specificate (tutte o almeno una a seconda del valore di 'richiediTutteLeCaratteristiche'). Assegnare -1 a 'distanzaMax' per eseguire la ricerca fino alla radice

```
bool UnitaFunzionaleDiscendeDa(
        string codUnita,
        string codUnitaAntenato,
        DateTime dataRif,
        int livelloMax
        )
```

Calcola se c'è un legame di parentela con non più di 'livelloMax' livelli. Assegnare -1 a 'livelloMax' per eseguire la ricerca su tutti i livelli possibili

```
bool UnitaFunzionaleFigliaDi(
        string codUnita,
        string codUnitaPadre,
        DateTime dataRif
        )
```

Calcola se un'unità funzionale discende direttamente da un'altra

```
int UnitaFunzionaleCalcolaLivelliParentela(
        string codUnita,
        string codUnitaAntenato,
        DateTime dataRif,
        int livelloMax
        )
```

Calcola la 'distanza', in 'livelli di parentela', tra una unità funzionale e un suo antenato. Assegnare -1 a 'livelloMax' per eseguire la ricerca su tutti i livelli possibili. Se le due unità non sono parenti o la distanza è maggiore di 'livelloMax' restituisce -1

Gerarchia

```
public UnitaFunzionaleGerarchia UnitaFunzionaleCreaGerarchia(
        string codUnitaRadice,
        int livelloMax,
        DateTime dataRif
        )
```

Crea la gerarchia completa dei discendenti dell'unità con non più di 'livelloMax' livelli di parentela. Assegnare -1 a 'livelloMax' per creare la gerarchia su tutti i livelli possibili

Elenchi

```
UnitaFunzionaleCollection ElencaUnitaFunzionali(
        DateTime dataRif
        )
```

Elenca le unità funzionali

```
UnitaFunzionaleCollection ElencaUnitaFunzionaliAventiCaratteri-
        stiche(
        DateTime dataRif,
        string[] caratteristiche,
        bool richiediTutteLeCaratteristiche
        )
```

Elenca le unità funzionali aventi le caratteristiche specificate (tutte o almeno una a seconda del valore di 'richiediTutteLeCaratteristiche')

```
UnitaFunzionaleCollection ElencaDiscendentiDirettiUnitaFunziona-
        le(
        string codUnitaRadice,
        DateTime dataRif
        )
```

Elenca le unità funzionali discendenti direttamente dall'unità funzionale (le “figlie”)

```
UnitaFunzionaleCollection ElencaDiscendentiDirettiUnitaFunziona-
        leAventiCaratteristiche(
        string codUnitaRadice,
        DateTime dataRif,
        string[] caratteristiche,
        bool richiediTutteLeCaratteristiche
        )
```

Elenca le unità funzionali discendenti direttamente dall'unità funzionale (le “figlie”) aventi le caratteristiche specificate (tutte o almeno una a seconda del valore di 'richiediTutteLeCaratteristiche')

```
UnitaFunzionaleCollection ElencaDiscendentiUnitaFunzionale(
        string codUnitaRadice,
        int livelloMax,
        DateTime dataRif
        )
```

Elenca le unità funzionali discendenti dell'unità funzionale con non più di 'livelloMax' livelli di parentela. Assegnare -1 a 'livelloMax' per eseguire la ricerca su tutti i livelli possibili

```
UnitaFunzionaleCollection ElencaDiscendentiUnitaFunzionaleAventi-
        Caratteristiche(
        string codUnitaRadice,
        int livelloMax,
        DateTime dataRif,
        string[] caratteristiche,
        bool richiediTutteLeCaratteristiche
        )
```

Elenca le unità funzionali discendenti dell'unità funzionale con non più di 'livelloMax' livelli di parentela, aventi le caratteristiche specificate (tutte o almeno una a seconda del valore di 'richiediTutteLeCaratteristiche'). Assegnare -1 a 'livelloMax' per eseguire la ricerca su tutti i livelli possibili

```
UnitaFunzionaleGerarchia UnitaFunzionaleCreaGerarchia(
        string codUnitaRadice,
        int livelloMax,
        DateTime dataRif
        )
```

Crea la gerarchia completa dei discendenti dell'unità con non più di 'livelloMax' livelli di parentela. Assegnare -1 a 'livelloMax' per eseguire la ricerca su tutti i livelli possibili

Persona

```
Persona DettaglioPersona(
        int matricola,
        DateTime dataRif
        )
```

Carica i dati base di una persona

```
bool PersonaIsResponsabileUnitaFunzionale(
        int matricola,
        string codUnita,
        DateTime dataRif
        )
```

Calcola se la persona è il responsabile dell'unità funzionale

```
bool PersonaIsAssegnataAUnitaFunzionale(
        int matricola,
        string codUnita,
        DateTime dataRif
        )
```

Calcola se la persona è assegnata all'unità funzionale

Allegato 14: Schede tecniche: applicativa e sistemi

SCHEDA TECNICA APPLICAZIONE <nome applicazione>

Dati generali dell'applicazione	
Nome Applicazione	<nome applicazione>
Descrizione sintetica	<sintetica descrizione dell'applicazione>
Tipologia dell'utenza	<Intranet – Extranet - Internet>
Modalità di interazione operatore/applicazione	<applicazione WEB - applicazione client/server - batch - web services misto >
Distribuzione geografica degli accessi all'applicazione	< comunale – provinciale – regionale – nazionale - internazionale>
Modalità di interazione con altri sistemi/servizi	<indicare se l'applicazione si interfaccia con altri sistemi e con quale modalità >
Filiera tecnologica	<java - microsoft - open source>
Stack tecnologico	<indicare sistema operativo, web server, application server, database server >
Lista componenti SW	<indicare eventuali componenti sw utilizzate dall'applicativo>
Protocolli di trasmissione dati	< HTTP - telnet – ftp>
Sistema di cifratura dati	<indicare se viene usato un sistema di cifratura dei dati e descriverne brevemente le caratteristiche>
Sistema di cifratura delle trasmissioni	< SSH – IPSEC – HTTPS – NO Cifratura>
Aspetti relativi alla criticità dell'applicazione	<indicare eventuali vincoli temporali nelle fasi di aggiornamento/elaborazione dei dati>
Modalità di gestione dei files generati e/o utilizzati dall'applicazione	<indicare se vengono memorizzati su file system, su database o sul sistema di gestione documentale>

Documentazione applicativa (da allegare alle schede in formato elettronico)	
Descrizione dell'architettura sw dell'applicazione	<descrivere l'architettura sw dell'applicazione>
Descrizione del sistema di sicurezza	<effettuare l'analisi dei rischi e descrivere il sistema di sicurezza da implementare (v. Disciplinare tecnico in materia di sicurezza delle applicazioni informatiche nella Giunta della Regione Emilia-Romagna", approvato con Determinazione n. 2651 del 2007). Allegare anche la check

	list compilata e motivata>
Organizzazione, struttura e semantica della base dati	<descrivere la struttura della base dati: schema concettuale e schema relazionale>
Descrizione flussi dati	<descrivere eventualmente i flussi dei dati previsti dall'applicazione>
Specifiche tecnico/funzionali	<descrivere in maniera dettagliata ogni funzionalità prevista inserendo lo schema di navigazione e/o il prototipo dell'interfaccia/pagine ed eventuali integrazioni con altri sistemi>
Manuale utente [1]	<descrivere il funzionamento dell'applicazione per il suo utilizzo da parte dell'utente finale>
Manuale di installazione, gestione. Descrizione della gestione del versioning dei sorgenti [1]	<descrivere le procedure per installare, configurare e gestire l'applicazione; indicare come viene gestito il versioning dei sorgenti >
Piano di Backup/Restore e Disaster Recovery	<indicare eventuali requisiti particolari di gestione del backup/restore dei dati ed eventuale necessità di un piano di Disaster Recovery>
Verbale di collaudo [1]	<descrivere i casi di test e l'esito relativi al collaudo dell'applicazione >
Accessibilità [1]	<descrivere la rispondenza dei requisiti di accessibilità, allegando la check list compilata>

Amministrazione degli Utenti	
Sistema di autenticazione	< UserID/password – Smart card – PKI - …>
Meccanismo di autenticazione	<indicare se applicativo o se centralizzato (AD, IAM) >
Amministrazione e consegna delle credenziali di autenticazione	<verificare, conformemente alle procedure dell'Ente, la consegna delle credenziali di autenticazione ed il loro inserimento nel sistema di gestione dell'Ente>
Livelli di profilazione utenti	<descrivere le modalità di implementazione (applicativo, db, AD, IAM) e le tipologie di profili (amministratore, operatore, etc..)>

Specifiche prestazionali	
Massimo numero indicativo di utenti	<indicare il numero massimo indicativo di utenti >
Massimo numero indicativo di utenti contemporanei	<indicare il numero massimo indicativo di utenti contemporanei >
Dimensione DB	<indicare la dimensione iniziale del DB e la percentuale di crescita annuale>
Movimentazione dei dati	<transazionale/batch e relativa distribuzione temporale >

[1] **(1)** Da allegare alla richiesta di rilascio in produzione

SLA applicativo	
Orario di servizio	< 8/24 – 24/24 - ..>
Massimo tempo di disservizio	< hh / gg >

Manutenibilità Software Applicativo (compilare nel caso di prodotti con licenza)	
License Key	
Software Subscription Key	
Software Subscription Scadenza	< gg/mm/aaaa >
Tipologia Licenza	<limitata/illimitata>
Escalation Path per problemi Software	< telefono – e-mail – web application >

FTPS (compilare solo nel caso serva attivare il servizio per l'invio del codice)				
Cognome	**Nome**	**e-mail**	**telefono**	**Interno/ esterno**

Contatti	
Struttura regionale committente	<indicare la struttura regionale che ha commissionato l'applicazione>
Referente utente regionale dell'applicazione	<indicare cognome, nome, tel., mail>
Referente tecnico regionale dell'applicazione	<indicare cognome, nome, tel., mail>
Referente tecnico del fornitore dell'applicazione	<indicare azienda, cognome, nome, tel., mail>

SCHEDA TECNICA DEI SISTEMI PER L'APPLICAZIONE <nomeapplicazione> [1]

Dati Generali	
Nome Sistema	<nome sistema>
Appartenza Dominio Regione	<SI/NO>
Sistema dedicato	<SI/NO>
IP Address	<indirizzo IP >
Funzionalità – Servizio	<Sintetica descrizione del servizio fornito dal sistema>
Dislocazione Fisica	<Edificio, stanza, rack, ecc.>
Inventario	< numero inventario >
Torretta	< numero torretta e porta relativo al cablaggio di rete >
Lista del Software installato	<Lista componenti S.O. e applicative e relative versioni >
System Owner	< Amministratore del sistema >

Manutenzione Hardware	
Serial Number	< numero di serie >
Model Type	< marca, tipo e modello >
Fornitore Contratto	< Generalità del fornitore/produttore del sistema >
Tipologia Contratto (8x7, 24 x7, 4 ore)	< Formula contrattuale per l'assistenza hardware in caso di problemi >
Numero di Contratto / Account	< Riferimento che consentirà di aprire una chiamata per problemi hardware con il fornitore>
Escalation Path per problemi Hardware	<Nr. Telefono / Sito Internet / e_mail address>

Amministrazione degli Utenti			
Utenti O.S	**Username**	**Gruppo**	**Ruolo/Funzione**
Utenti servizio	**Username**	**Gruppo**	**Ruolo/Funzione**

Documentazione da allegare

[1] **(1)** da compilare unicamente in caso di porting da sistemi hardware pre-esistenti o nuovi progetti che prevedono l'acquisizione di hardware.

Architettura - Diagramma fisico del network con, componenti di interconnessioni e.g. switch e router (e area di gestione) e relative porte	< Documento che descrive gli elementi di infrastruttura del sistema dal punto di vista del networking) >
Architettura - Diagramma logico dell'infrastruttura con flussi, DMZ, indirizzi IP	< Documento che descrive il disegno logico delle applicazioni installate nell'ambito dell'infrastruttura complessiva >
Custom script che girano automaticamente, e non, sul sistema	< Documento che descrive le funzionalità degli script >
Soluzione di Backup	< Documento che descrive la soluzione adottata >
Piano di ripristino	< Documento che descrive l'eventuale piano di ripristino >
Standby hardware	<Elencare eventuali componenti hardware per emergenze>

Profilo Hardware	
Modello Macchina	
Modello CPU	
Clock CPU	
Memoria fisica	
Configurazione array dischi	
Configurazione volume logici	

Sistema Operativo	
Nome e Version del Sistema Operativo	
Patch o Service Pack installati	

Interfacce di rete							
Nome	**IP**	**Subnet**	**Speed**	**Duplex**	**Switch**	**Port**	**Vlan**
eth1c0	down	<subnet>	Fast Ethernet	Full Duplex	<switch nome>	<switch port>	<vlan no.>
eth2c0	xxx.xxx.xxx.xxx	xx	Fast Ethernet	Full Duplex	<switch nome>	x	x

DNS		
Hostname	**IP**	**Ruolo**

<Non Presente>		<Primario/Secondario>

Accesso Remoto	
Telnet (se consentito)	<ip1>
SSH	<ip1>

Software Installati (per ogni software installato compilare almeno le prime 4 voci)	
Versione	
Build Number	
Patch installati	
Directory di installazione	
Licenza, specificare se locale o centrale	
Eventuale Indirizzo IP della Management	
Lista delle features che non sono standard ma utilizzate	
Clustering	

Manutenibilità Sistema Operativo e Servizi (compilare per ogni servizio installato)	
License Key	
Software Subscription Key	
Software Subscription Scadenza	
Tipologia Licenza	<limitata/illimitata>
Escalation Path per problemi Software	

Allegato 15: Servizi e strumenti web

1 Forum

Il forum è uno strumento di *community*, cioè è un modo per mettere in contatto persone che hanno idee e interessi comuni, e creare le cosiddette Comunità Virtuali. Un forum è costituito da pagine che mostrano (e permettono) discussioni degli utenti su vari temi. Ogni forum deve essere "animato" e governato, non vive di vita propria e richiede una presenza costante.

Per partecipare a un forum bisogna collegarsi al sito desiderato e compilare un form per inviare il proprio messaggio o leggere i messaggi inviati dagli altri membri; non è necessario alcun software né scaricare alcunché sul proprio pc.

Normalmente la discussione è regolata da un moderatore, che è il primo a ricevere i messaggi e può decidere se pubblicarli, accedendo ad un ambiente dedicato.

Il Servizio SIIR ha sviluppato un sistema di gestione dei Forum che rispecchia tutti gli standard di Internet, e che ha le seguenti caratteristiche:

- può essere un unico forum oppure implementare più "stanze" di discussione (inizialmente si può partire con un forum unico e poi casomai successivamente suddividerlo in più stanze a seconda degli argomenti);
- è possibile configurare un forum in modo che abbia una scadenza; dopo la data di scadenza non sarà più possibile inserire contributi, ma solo consultare quelli esistenti;
- per l'invio di un contributo è obbligatorio l'inserimento dell'oggetto e del testo del messaggio (l'oggetto nel caso di risposte ed altri contributi contiene automaticamente l'oggetto del messaggio originale, preceduto da "RE:"); nel caso sia necessario per altre finalità raccogliere dati aggiuntivi (es. nome, cognome, ente, ecc.) è possibile farlo, ma deve essere specificato nell'informativa sul trattamento dei dati personali il motivo per cui tali dati vengono raccolti;
- volendo, si può dare la possibilità di allegare un file al messaggio;
- l'elenco dei contributi viene visualizzato in ordine temporale, con in cima i più recenti, indicando il nome del mittente, l'oggetto e la data di invio; le risposte vengono visualizzate sotto al messaggio a cui si riferiscono, indentate in base ai comuni sistemi di gestione dei *thread;*
- se ci sono uno o più moderatori, essi possono essere indifferentemente interni o esterni alla Regione;
- il forum può essere personalizzato nella grafica per assumere un aspetto coerente con il sito in cui viene inserito.

2 Newsletter

La Newsletter, è sorta di notiziario telematico che viene inviato via mail agli iscritti ad una

lista di distribuzione; di solito si tratta di un elenco di novità ed eventi di particolare interesse che spesso rimandano al sito web per approfondimenti.

Il Servizio SIIR ha sviluppato un sistema di gestione delle newsletter e delle relative liste di iscritti, conforme alla normativa sulla privacy, che consente, tramite un'interfaccia web semplice e intuitiva, di gestire da un lato la lista degli iscritti (con la possibilità di raggrupparli in sottocategorie) e dall'altro di comporre la propria newsletter (inserendo direttamente il testo oppure scegliendo un documento da allegare) e di inviarla agli iscritti (tutti o solo determinate categorie). E' possibile precompilare in automatico il form per l'invio della newsletter indicando un URL dove reperire il testo HTML e la versione TXT dell'ultima newsletter pubblicata.

L'iscrizione alla newsletter può essere effettuata soltanto dal diretto interessato attraverso la compilazione di un form sul sito web di riferimento. Per verificare l'esistenza dell'indirizzo e-mail e l'effettiva volontà dell'interessato ad iscriversi, il sistema invia una mail con richiesta di conferma dell'iscrizione all'indirizzo specificato; perché l'iscrizione vada a buon fine, l'interessato deve dare conferma alla richiesta entro 48 ore. Analogo sistema viene utilizzato per la cancellazione di un indirizzo dal Servizio.

Per utilizzare il sistema di gestione newsletter è necessario stabilire le seguenti informazioni:

- uno o più referenti che avranno accesso al sistema sia per la gestione degli iscritti che per l'invio della newsletter;
- un indirizzo di posta elettronica (e relativa denominazione) da utilizzare come mittente per l'invio delle newsletter; deve essere un indirizzo regionale: attualmente non è possibile inviare la newsletter con mittenti diversi da quelli del dominio regionale;
- per consentire l'iscrizione al servizio il solo dato necessario da raccogliere è l'indirizzo e-mail; nel caso sia necessario per altre finalità raccogliere dati aggiuntivi (es. nome, cognome, ente, ecc.) è possibile farlo, ma deve essere specificato nell'informativa sul trattamento dei dati personali il motivo per cui tali dati vengono raccolti;
- il testo dell'informativa ai sensi della privacy;
- se si possiede già una lista di indirizzi è possibile importarli nella lista di distribuzione (semprechè i dati siano stati raccolti in conformità con quanto previsto dal D.Lgs. 196/2003 "Codice in materia di protezione dei dati personali").

La realizzazione della newsletter (composizione, redazione, impaginazione, grafica) è a cura del richiedente; è possibile implementare la composizione e redazione della newsletter all'interno del sistema wcm regionale.

3 Iscrizione a convegni

Il Servizio SIIR ha sviluppato un sistema di gestione delle iscrizioni online a convegni o ad altri eventi che consente di realizzare le pagine di iscrizione, di gestire le richieste pervenute, con le relative notifiche via email, e di monitorare l'andamento delle adesioni.

Questo servizio infatti consente ai gestori di personalizzare la pagina di iscrizione con logo

ed opportuno css grafico, nonché definire titolo, testo descrittivo, dati da richiedere agli iscritti, ed eventuale allegato da scaricare (ad esempio il programma dell'evento) o link ad una pagina di approfondimento su un altro sito. In qualsiasi momento il gestore può monitorare il numero di persone iscritte al convegno, consultarne i dati di riferimento ed esportare i dati degli iscritti in locale in formato .xls.

Per attivare un form on-line di iscrizione è necessario:

- individuare uno o più referenti che avranno accesso al sistema per la gestione degli iscritti e per la composizione della pagina di iscrizione,
- stabilire l'indirizzo di posta elettronica e il nome visualizzato che dovrà comparire nelle e-mail di richiesta di conferma dell'iscrizione,
- decidere se e a quale indirizzo i gestori dell'evento desiderano ricevere segnalazione di ogni iscrizione confermata,

stabilire se si vuole fornire agli iscritti la possibilità di inviare ai gestori un breve testo (es. una domanda da porre ai relatori) in fase di iscrizione.

4 Sondaggi e questionari

Il Servizio SIIR ha acquistato un abbonamento annuale al servizio on-line EuroVoxBox che permette di realizzare sondaggi e questionari di ogni tipo nel rispetto della normativa sull'accessibilità e conforme alle norme sulla sicurezza previste dal D.Lgs. 196/2003 "Codice in materia di protezione dei dati personali".

EuroVoxBox offre la possibilità di costruirsi da soli i propri sondaggi e di controllare on line l'intero processo, partendo dalla preparazione del sondaggio, fino all'analisi dei dati ed alla loro presentazione.

La composizione del sondaggio può essere realizzata in piena autonomia tramite una apposita area riservata di accesso al sistema, oppure fornendo al Servizio SIIR le indicazioni per la costruzione del sondaggio, che sarà poi effettuata dai tecnici.

Ogni sondaggio può essere compilato in forma anonima oppure fornendo a ciascun rispondente delle credenziali di accesso personali, in modo da renderlo identificabile.

Il sondaggio può essere somministrato facilmente tramite un link su un sito che rimanda alla pagina di ingresso del sondaggio.

In qualsiasi momento si può monitorare quante persone hanno avuto accesso al sondaggio e quanti ne hanno completato la compilazione. Inoltre, possono essere visualizzati dei semplici risultati sintetici in tabelle di frequenza e grafici.

I risultati finali possono comunque essere analizzati con qualunque strumento statistico disponibile sul mercato oppure rivolgendosi al Servizio competente in materia di statistica.

5 Groupware

Il Groupware è un ambiente di lavoro condiviso per gruppi di collaborazione su Internet.

Questo strumento di collaborazione è l'ideale per gruppi di persone che, a distanza, lavorano allo stesso progetto, sviluppando documentazione ed interagendo variamente attraverso le usuali forme di comunicazione disponibili sul web.

La Regione Emilia-Romagna attualmente utilizza "Acollab" come strumento di groupware. ACollab è un prodotto open source, di fatto accessibile e di facile utilizzo, conforme alle norme sulla sicurezza previste dal D.Lgs. 196/2003 “Codice in materia di protezione dei dati personali”.

L’interfaccia, semplice ed intuitiva, gestisce la collaborazione di gruppo nelle seguenti forme:

- condivisione dei documenti di progetto nelle 2 forme di “bozza” di lavoro (in cui modificare ed archiviare le revisioni dei file di progetto) e di “documento finale” (che consolidano le bozze nella versione finale pubblicata dei documenti),
- partecipazione a forum, per intervenire nelle discussioni interne al gruppo di lavoro,
- comunicazione in tempo reale (*chat*) con gli utenti collegati al gruppo di lavoro,
- messaggistica (e-mail) interna al gruppo di lavoro,
- agenda degli eventi rilevanti per il gruppo di lavoro.

Ogni gruppo è identificato da un nome ed eventualmente un logo; uno dei membri del gruppo assume le funzioni di Amministratore, che consistono principalmente nel definire i profili utente, costruire la struttura delle cartelle dei documenti, inserire news ed eventi.

Altri strumenti di groupware, con funzionalità più avanzate, sono in fase di valutazione.

REGIONE EMILIA-ROMAGNA
Atti amministrativi

GIUNTA REGIONALE

Grazia Cesari, Responsabile del SERVIZIO SISTEMA INFORMATIVO - INFORMATICO REGIONALE esprime, ai sensi della deliberazione della Giunta Regionale n. 2416/2008, parere di regolarità amministrativa in merito all'atto con numero di proposta DPG/2009/4810

data 15/05/2009

IN FEDE

Grazia Cesari