REGIONE CAMPANIA – LINEE DI INDIRIZZO PER L'IMPLEMENTAZIONE DEL SISTEMA INFORMATIVO SANITARIO REGIONALE

Allegato 1A

Sinfonia - Architettura Generale del sistema applicativo

Versione 1.00

21 Settembre 2018

SOMMARIO

1. Introduzione 4

2. Scopo e Ambito di Applicazione 4

3. Riferimenti 4

4. Termini e definizioni 5

5. Vincoli ed obiettivi architetturali 9

6. Architettura del Software del Sistema Sinfonia 11

6.1. Presentation Tier 11

6.1.1. Presentation Logic della User Interface 11

6.1.2. Presentation Logic dei Web Services 13

6.1.3. Report 15

6.1.4. HL7 CDA 15

6.2. Business Tier 16

6.3. Elaborazioni Batch 16

6.3.1. Presentation Layer 17

6.3.2. Business Layer 17

7. Componenti per la Firma Digitale 19

7.1. CNS, PKCS#11, Wrapper Java, card reader e PC/SC driver 19

7.2. Formato dei documenti prodotti e standard di firma 20

7.3. Controlli di validità su un documento firmato 20

7.4. Il processo di firma digitale 21

8. Gestione Utenti, Identificazione, Autenticazione ed Autorizzazione 24

8.1. Identificazione, autenticazione ed autorizzazione degli utenti 24

8.1.1. Definizione e Profilazione degli Utenti 24

8.1.2. Autorizzazione 24

9. Identificazione ed Autenticazione per i servizi di cooperazione 25

9.1. Il processo complessivo 25

9.1.1. Identificazione ed autenticazione dei Sistemi Fruitori 26

9.1.2. Integrità del messaggio 27

9.1.3. Non ripudio del messaggio 27
9.1.4. Mantenimento delle informazioni della richiesta di servizio 27
9.1.5. Riservatezza del messaggio 28
9.1.6. Firma dei messaggi di risposta 28
9.1.7. Identità dell'utente 28
9.1.8. Identificazione ed Autenticazione della Porta di Dominio 29
9.2. Autorizzazione per i servizi di cooperazione. 29
9.2.1. Autorizzazione del Sistema Fruitore all'uso di Sinfonia 29
9.2.2. Autorizzazione del Sistema Fruitore all'uso del servizio 29
9.2.3. Autorizzazione dell'utente del Sistema Fruitore 30
9.3. Riepilogo dei controlli eseguiti ad ogni richiesta di servizio di cooperazione 30

10.Sicurezza accesso ai dati su DB 32
10.1. Identificazione, autenticazione ed autorizzazione 32
10.2. Cifratura 32
10.3. Disaccoppiamento tra dati sensibili e anagrafici 33

11.Amministrazione Applicativa 34

12.Tracciabilità e Monitoraggio 35
12.1. Ambito del tracciamento 35
12.2. Punti di tracciamento nel software 36
12.3. Procedura di tracciamento 36
12.4. Persistenza 36
12.5. Canali di tracciamento 36
12.6. Record di tracciamento 36
12.7. Procedura di auditing 37
12.8. Record di auditing 37

1. Introduzione

Il presente documento fornisce una panoramica generale dell'organizzazione delle componenti software che compongono il sistema Sinfonia, riportando le diverse "viste" architetturali, che descrivono differenti aspetti del sistema. Esso formalizza le diverse decisioni architetturali e progettuali che sono state prese per le componenti del sistema in oggetto.

2. Scopo e Ambito di Applicazione

Il documento è il risultato delle activity del workflow di Analysis & Design previste dalla metodologia adottata e costituisce l'input per quelle relative al workflow di Implementation & Test; pertanto è destinato a tutti i ruoli coinvolti nelle attività relative a quest'ultimo workflow.

Vengono riportati gli obiettivi ed i vincoli che possono risultare significativi per le restanti viste architetturali (dei Casi d'Uso e Logica).

3. Riferimenti

- Progetto Esecutivo per So.Re.Sa. S.p.A. Società Regionale per la Sanità Regione Campania Rif. Consip ID SIGEF 1607.
- Decreto Dirigenziale n. 131 del 20.06.2018.

4. Termini e definizioni

Modello	Rappresentazione concettuale del sistema ottenuta attraverso l'utilizzo di costrutti linguistici e semantici propri di un linguaggio standardizzato di modellazione (UML).
Package	Elemento del modello che rappresenta un contenitore di altri elementi quali classi, componenti, interfacce, diagrammi, package, ecc.
Componente o Area applicativa	Componente intesa come sistema applicativo oggetto di fornitura o di terze parti.
WEB Tier	Livello architetturale del sistema software dedicato alla interazione con l'utente attraverso tecnologia e protocolli Internet.
EJB Tier	Livello architetturale del sistema costituito da tutti i componenti software relativi all'area della logica applicativa (Business Rules).
Documento HTML	Un documento HTML è un documento SGML che soddisfa i requisiti delle specifiche W3C.
J2EE (Java 2 Enterprise Edition)	Versione enterprise della piattaforma Java.
DAO (Data Access Object)	Pattern che ha lo scopo di disaccoppiare la logica di business dalla logica di accesso ai dati.
HTTP (Hyper Text Transfer Protocol)	Protocollo standard di trasferimento di un ipertesto.
MVC (Model-View-Controller)	Pattern architetturale per lo sviluppo di interfacce grafiche di sistemi software.
SOAP (Simple Object Access Protocol)	Specifica per lo scambio di informazioni strutturate nell'implementazione di Web Services in reti di computer. Il formato dei messaggi è l'XML.
SMTP (Simple Mail Transfer Protocol)	Protocollo standard per la trasmissione via internet di e-mail.

HTML(HyperText Markup Language)	Linguaggio usato per descrivere la struttura dei documenti ipertestuali disponibili nel World Wide Web.
XML (eXtensible Markup Language)	Metalinguaggio standardizzato dal World Wide Web Consortium (W3C).
Stylesheet	Lo Stylesheet (foglio di stile) specifica il formato di presentazione di un documento XML descrivendo sia la trasformazione (opzionale) della struttura del documento di ingresso in un'altra struttura sia come devono essere visualizzati gli elementi della struttura. Il linguaggio per definire un o stylesheet è XSL (eXtensible Stylesheet Language).
XHTML (eXtensible HyperText Markup Language)	Versione aggiornata ed estesa dell'HTML.
JVM (Java Virtual Machine)	Macchina virtuale che esegue programmi in linguaggio Java bytecode.
JS (JavaScript)	Linguaggio di scripting.
DB	DataBase.
Certificato Digitale	Documento elettronico che attesta, con una firma digitale di una certification authority riconosciuta, l'associazione tra una chiave pubblica e l'identità di un soggetto. I certificati digitali aderiscono al formato internazionale ITU-T X.509 secondo quanto descritto dallo standard PKIX “Certificate and CRL Profile”.
Certification Authority (CA)	Entità preposta alla creazione, emissione e garanzia dei certificati digitali, cioè crea ed assegna una determinata coppia di chiavi pubblica e privata e garantisce sull’identità del possessore di tale coppia di chiavi. Gli standard di riferimento per la realizzazione della CA sono RFC2510 ed RFC2511. I certificati devono aderire allo standard X.509 v3 con tutte le estensioni previste in RFC3260. Ulteriori requisiti per i certificati sono specificati dalla RFC3039 che definisce la struttura dei Qualified Certificates come specificato nella European Digital Signature Directive. Responsabilità della CA è anche la

	pubblicazione delle Certificate Revocation Lists (CRLs) secondo lo standard X.509 v2 specificato in RFC2459 e RFC3280.
Registration Authority (RA)	Entità dedicata alla registrazione degli utenti e all'accettazione delle richieste per i certificati. Raccoglie le informazioni dell'utente, esegue la verifica della sua identità, che viene quindi utilizzata per la registrazione dello stesso secondo le policy accordate. Gestisce la revoca dei certificati e le Certificate Revocation Lists (CRLs) e comunica con i protocolli ed i formati specificati in RFC2510bis ed RFC2511bis. Inoltre fa da interfaccia verso il Repository che pubblica i certificati emessi dalla CA e le CRLs. La RA è anche preposta alla scrittura delle smartcard quando è richiesta la generazione centralizzata della coppia di chiavi.
RA Repository	Il luogo dove vengono registrati i certificati, le chiavi, le Certificate Revocation Lists (CRLs).
Certificate Revocation List (CRL)	Lista di certificati digitali revocati perché non sono più validi a causa di molteplici ragioni tra cui compromissione della chiave privata, cambio dei dati personali, scadenza. Le CRL aderiscono al formato internazionale ITU-T X.509 secondo quanto descritto dallo standard PKIX "Certificate and CRL Profile".
PKI (Public Key Infrastructure)	Standardizza l'insieme di tecnologie, infrastrutture, e pratiche di management richieste per abilitare e rendere effettivo l'uso di autenticazione, cifratura e firma elettronica basate su chiave pubblica in applicazioni distribuite (Certificati Digitali, Certification Authority (CA), Registration Authority (RA), Repository e Certificate Revocation List (CRL)).
UDDI (Universal Description Discovery and Integration)	Registry (base dati ordinata ed indicizzata), basato su XML ed indipendente dalla piattaforma hardware, che permette alle aziende la pubblicazione dei propri dati e dei servizi (Web services) offerti su internet.

ESB (Enterprise Service Bus)	Un Enterprise Service Bus (ESB) è un'infrastruttura software che fornisce servizi di supporto ad architetture SOA complesse. Un ESB si basa su sistemi disparati, interconnessi con tecnologie eterogenee, e fornisce in maniera consistente servizi di orchestration, sicurezza, messaggistica, routing intelligente e trasformazioni, agendo come una dorsale attraverso la quale viaggiano servizi software e componenti applicativi.
HL7 (Health Level 7)	Health Level 7 (HL7) è un'associazione non profit internazionale che si occupa di gestire standard per la sanità. HL7 è riferito anche ad alcuni degli specifici standard creati da questa associazione (es. HL7 v2.x, v3.0, CDA, ecc.).
Spring	Spring è un framework open source per lo sviluppo di applicazioni su piattaforma Java. A questo framework sono associati tanti altri progetti, che hanno nomi compositi come Spring Boot, Spring Data, String MVC, Spring Batch, eccetera.
Hibernate	Hibernate è una piattaforma middleware open source per lo sviluppo di applicazioni Java che fornisce un servizio di Object-relational mapping (ORM), ovvero gestisce la persistenza dei dati sul database attraverso la rappresentazione e il mantenimento su database relazionale di un sistema di oggetti Java.

5. Vincoli ed obiettivi architetturali

Le soluzioni tecnologiche ed architetturali adottate per la definizione del sistema Soresa rappresentano una evoluzione degli attuali sistemi in essere nel segmento sanitario della Regione Campania. L'architettura applicativa di Sinfonia si arricchisce di tutti gli elementi utili al funzionamento del sistema secondo il paradigma del Cloud e in linea con le soluzioni proposte nella progettazione esecutiva offerta.

Rimane ovviamente la necessità di disporre di applicazioni che sono tra loro interoperabili e che permettono l'effettivo passaggio di informazioni tra gli attori coinvolti senza interruzioni.

L'evoluzione del complesso ecosistema Sinfonia verso un modello di cloud computing introduce una semplificazione architetturale, organizzativa e di cooperazione dei sistemi e rafforza d'altro canto le caratteristiche di affidabilità, scalabilità, disponibilità e sicurezza complessiva del sistema.

L'innovazione architetturale che qui viene presentata è agevolata e trae vantaggio dal nuovo assetto architetturale e tecnologico che si determina con il consolidamento complessivo del sistema in ambiente cloud. Tale riconfigurazione tecnologica centralizzata, favorisce, rispetto all'attuale configurazione architetturale distribuita, l'introduzione di ulteriori componenti middleware standard come più dettagliatamente descritto nel seguito.

Sotto il profilo organizzativo, architetturale e di cooperazione applicativa, rispetto all'attuale modello di dispiegamento, tutte le istanze del sistema gestionale Sinfonia vengono consolidate sul sistema Cloud messo a disposizione da Soresa.

I servizi di Sinfonia sono resi fruibili ad un utente finale tramite applicazioni Web. Il sistema Sinfonia supporta inoltre la cooperazione applicativa con sistemi fruitori mediante l'esposizione di servizi applicativi di cooperazione (in modalità web services e SPCoop).

Le caratteristiche generali della soluzione si basano sugli strumenti tecnici ed architetturali che allo stato attuale appaiono più maturi, stabili e con elevato potenziale di crescita e diffusione. Come accennato, ci si è riferiti ai modelli di comunicazione basata sugli standard nazionali per l'interoperabilità e cooperazione (SPCoop), ai servizi per la sicurezza e la privacy basati su smart card e firma digitale, all'infrastruttura RUPAR-SPC ed ai servizi connessi peraltro già presenti nella Regione Campania.

Elementi fondamentali del modello architetturale sono:

- l'adozione della SOA (Services Oriented Architecture) e dei Web services che forniscono un approccio per la definizione, la pubblicazione e l'utilizzo dei servizi applicativi;
- la Porta di Dominio che costituisce l'interfaccia standard di connessione dei servizi applicativi di ogni dominio alla RUPAR e a SPCoop (Sistema Pubblico di Cooperazione).

L'invocazione dei servizi di cooperazione è basata sulla "Busta e-Gov", per quanto riguarda gli aspetti di sicurezza point-to-point, affidabilità della trasmissione e tracciatura delle comunicazioni e sui linguaggi XML e WSDL, HL7 e sullo standard UDDI per la formalizzazione degli "Accordi di servizio".

L'architettura del sistema proposto si basa sui seguenti standard tecnologici di progetto e sviluppo già in uso:

- adozione di browser Internet standard per la visualizzazione dell'interfaccia utente;
- logica di presentazione web basata sull'impiego di Web Server e pagine dinamiche JSP;
- logica di presentazione di tipo programmatico basata su Web Services con l'utilizzo di XML su protocollo SOAP;
- logica applicativa implementata in architettura J2EE.

Si aggiunge l'uso dei seguenti ulteriori standard:

- Il middleware WSO2 EI (Enterprise Integrator) che include le componenti ESB, MB, BPM e BRM per una architettura orientata ai servizi (SOA).
- JPA/Hibernate per la logica di accesso ai dati.
- Spring per la gestione della logica di presentation.

L'accesso ai servizi offerti dal sistema Sinfonia è realizzato mediante tre differenti modalità:

- utilizzo della logica di Web presentation resa disponibile tramite pagine JSP: modalità fruibile da utenti mediante l'utilizzo di un browser Internet;
- Porta Applicativa: realizza l'esposizione di servizi di cooperazione di Sinfonia secondo la modalità SPCoop a favore di sistemi fruitori cooperanti – di norma – appartenenti a domini organizzativi esterni a quello che ospita Sinfonia;
- Web services: realizza l'esposizione di servizi di cooperazione di Sinfonia a favore di sistemi fruitori cooperanti – di norma – appartenenti allo stesso dominio organizzativo che ospita Sinfonia.

6. Architettura del Software del Sistema Sinfonia

Per ogni istanza Sinfonia il software applicativo è distribuito sui seguenti tier:

- Presentation Tier
- Business Tier

Inoltre il sistema Sinfonia ha la responsabilità dell'esecuzione di elaborazioni batch, cioè di processi la cui esecuzione è asincrona rispetto alla richiesta attivata dall'utente mediante la web application e che non richiedono interazione con l'utente.

6.1. Presentation Tier

Il Presentation Tier del Sinfonia ha la responsabilità di presentare i servizi sia in forma programmatica (Web Services) sia come Web Application per i servizi applicativi e per la generazione di report.

I componenti di questo tier sono:

- Presentation Logic della User Interface
- Presentation Logic dei Web Services
- Report
- HL7 Engine

6.1.1. Presentation Logic della User Interface

La Presentation Logic della User Interface implementa l'interfaccia Web del sistema Sinfonia ed ha la responsabilità della presentazione dei servizi applicativi tramite interfaccia Web (XHTML/HTTP) ad un operatore accreditato tramite una Workstation dotata di internet browser html standard.

La sua implementazione si basa sul Web-tier di un Application Server J2EE compliant, costituito da un Web Server e un Web Servlet Container che costituisce il contenitore standard per i componenti di front end della tecnologia J2EE: Servlet e Java Server Pages (JSP).

Questo componente si occupa del dispatching delle richieste HTTP provenienti dal browser client e provvede alla generazione dinamica delle pagine XHTML di risposta. Ha la responsabilità della uniformità dell'interfaccia grafica, della gestione delpage flow e del mantenimento dello stato conversazionale con il client (sessione).

La generazione dinamica delle pagine XHTML di risposta ed il controllo del flusso delle pagine web generate dinamicamente avviene seguendo il modello definito dal pattern MVC (Model View Controller).

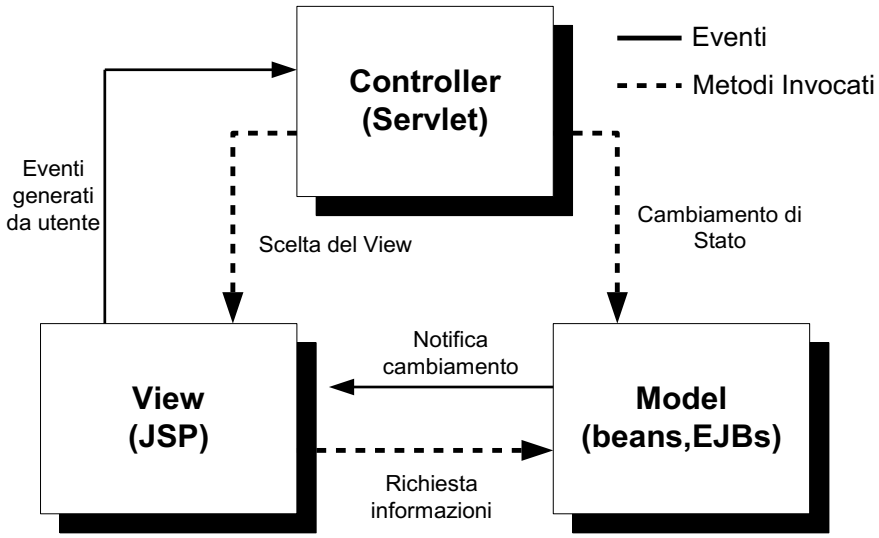

Figura 1 – Il Pattern MVC

In questo modello generale la Servlet (Controller) si comporta da motore web: in base all'input dell'utente, sul quale esegue una verifica di consistenza, decide quale processo di business invocare e seleziona la View successiva.

La View è costituita da una pagina JSP che dinamicamente costruisce il suo contenuto in base ai cambiamenti avvenuti nel Model, vale a dire recupera il risultato della transazione innescata dalla Servlet e lo visualizza formattato all'utente.

Il Model è il sistema contenente la logica di business con cui l'utente interagisce innescando processi atti a conoscerne o alterarne lo stato.

L'uso del pattern MVC permette il completo disaccoppiamento fra la logica di presentation implementata nelle pagine JSP e la logica di controllo implementata nel codice della Servlet.

La specifica implementazione di MVC per i Servizi Applicativi di Sinfonia prevede che:

- il Model renda disponibile il risultato di una transazione lanciata dalla Servlet nell'oggetto HTTPSession (oggetto standard contenente i dati di sessione);
- la JSP raccolga e renda visibile tale risultato all'utente titolare della sessione;

Nell'architettura dei servizi applicativi di Sinfonia il ruolo del Model specificato nel pattern MVC è ricoperto, in linea con quanto dettato dal pattern Business Delegate, da un componente definito EJB Delegate a cui viene delegata la lookup, tramite JNDI, degli EJBs nell'EJB-tier che realizzano il Business Tier, disaccoppiando quindi in maniera completa la logica di controllo e di presentation dalla logica applicativa e di integrazione.

Rientra nella responsabilità della Servlet la gestione della sessione HTTP, ovvero dello stato conversazionale con il client, il controllo di consistenza dei dati inseriti dall'utente nella form e la gestione di situazioni anomale con presentazione all'utente di una pagina di errore. E' previsto l'uso di una Servlet e quindi di uno specifico flusso di pagine per ogni singolo caso d'uso del sistema e le responsabilità della Servlet sono distribuite su più classi di supporto (helper) al fine di rendere più modulare e manutenibile il codice.

I dati manipolati nel Presentation-tier sono modellati da classi di business (Business Object) che rappresentano le entità presenti nel dominio del problema (ad es. assistibile, esenzione ecc.).

Le pagine JSP che prevedono l'immissione dell'input da parte dell'utente (form) fanno uso di codice JavaScript eseguito lato client dal browser per un primo controllo formale sull'input dell'utente al fine di evitare inutile traffico in rete; tale controllo è in ogni caso ripetuto dalla Servlet lato server per verificare comunque l'integrità dei dati ricevuti ed in modo che l'eventuale disabilitazione del supporto JavaScript del browser non sia bloccante per l'applicazione.

0.1.1.1 DAO del Web-tier

Il pattern architetturale DAO qui descritto viene adottato per realizzare l'accesso ai dati nel Web-tier da parte delle componenti di Web presentation.

Tali accessi non sono utilizzati dalla logica di business dell'applicazione ma sono finalizzati esclusivamente al reperimento di informazioni utili per popolare combo box, list box, drop–down list, label, menù etc., quindi informazioni utili alla presentation logic.

6.1.2. Presentation Logic dei Web Services

La Presentation Logic dei Web Services ha la responsabilità della presentazione dei servizi applicativi tramite interfaccia programmatica basata sul web service standard SOAP.

I servizi esposti con tale modalità vengono fruiti da:

- Porta Applicativa della Porta di Dominio per la cooperazione applicativa interdominio standard SPCoop;
- applicativi che non utilizzano SPCoop e che interagiscono con la presentation logic del dominio di riferimento con modalità web service standard SOAP.

La scelta progettuale di utilizzare JAX-WS e JAXB per l'implementazione della Presentation Logic dei Web Services comporta una visione che implica il pensare un web service in termini di RPC basato su SOAP partendo dall'implementazione Java del servizio realizzata tramite POJO (Plain Old Java Object).

Un POJO è una semplice classe Java che non ha un legame diretto con un container o un application server e quindi non implementa interfacce specifiche e non estende classi specifiche. Il POJO contiene metodi che implementano i servizi che vanno esposti sotto forma di web services ed a questo scopo in esso vengono utilizzate le "annotations" (informazioni per il compilatore o per il runtime che non hanno influenza sul codice Java), una caratteristica introdotta nel linguaggio Java a partire da J2SE 5 (1.5).

L'ambiente operativo per l'esecuzione dei POJO e per l'esposizione dei web services è costituito dal framework WSIT (Web Services Interoperability Technology). WSIT è l'implementazione del web services stack Metro, progetto open source supportato da Sun Microsystems prima e da Oracle ora. WSIT implementa gli standard più recenti di SOAP, WSDL e WS-* (con una particolare cura per WS-Security). La principale caratteristica del WSIT (detto anche stack Metro) è la completa aderenza agli standard JSR e JAX-* e la completa e garantita interoperabilità con i Framework Microsoft .NET 3.0 e .NET 3.5.

Lo scheletro di un POJO, con le relative annotazioni necessarie a renderlo un web service viene tipicamente costruito automaticamente dall'IDE che provvede anche alla definizione del relativo WSDL che, all'atto del deploy del servizio, viene pubblicato da una URL specificata. Inoltre l'IDE consente l'aggiunta degli handler per espandere le funzioni infrastrutturali del servizio come ad esempio la gestione trasparente di aspetti come la gestione degli errori (fault), il log e la sicurezza.

L'implementazione dei web services è orientata alla invocazione remota di un metodo Java via SOAP, interoperabile con altre piattaforme che implementano i web services.

Per i dettagli circa identificazione, autenticazione ed autorizzazione si faccia riferimento al paragrafo Identificazione e Autenticazione per i servizi di cooperazione.

Ciascun web service di Sinfonia sarà quindi costituito delle seguenti componenti principali:

- POJO (Plain Old Java Object);
- VO (Value Object).

0.1.1.2 POJO

Per poter esporre un servizio di cooperazione mediante un web service è necessario definire classi Java denominate POJO (Plain Old Java Object), che ne implementano il comportamento. Tali classi di implementazione fanno uso delle *java-annotation* standard, del package *javax.jws.**.

Utilizzando le *java-annotation* il framework di web-services, a run-time, sull'application-server, genera automaticamente per ogni POJO un WSDL.

Sarà realizzata una classe POJO di implementazione per ogni entità.

Ogni metodo della classe POJO implementa una operation del web service, descritta nel WSDL del web service ed esposta sui nodi di Sinfonia. Il nome del metodo deve essere uguale al nome della operation riportato nel documento di architettura dell'area.

0.1.1.3 Value Object (VO)

I metodi dei POJO (come ogni metodo Java) prevedono parametri di input e output.

Ciascun parametro è in alternativa:

- tipo elementare (tipi primitivi e classi standard Java ad es. String e wrapper)
- tipo complesso, realizzati mediante classi denominate Value Object (VO) che contengono soltanto attributi ed i relativi metodi *set* e *get*.

Le classi Value Object saranno collocate in un package e sono costruite a partire dalla definizione dell'input e dell'output di ciascuna operation, secondo quanto specificato nei documenti di architettura di ciascuna area applicativa

6.1.3. Report

Per la realizzazione dei report vengono utilizzate le librerie:

- Java Reporting Component (JRC) Ver. 11.8 o successiva di Business Object per integrare nell'applicazione la generazione dei report;
- Report Designer Crystal Reports 2008 per la fase di design di ogni report.

Le JRC costituiscono un efficace modulo di creazione di report che sfrutta appieno i vantaggi offerti dalla portabilità Java su più sistemi operativi e piattaforme hardware.

0.1.1.4 Integrazione report interattivi

L'integrazione di ogni report all'interno della web application avviene tramite i componenti:

- **Report Engine:** processa la richiesta di report ricevendo l'opportuno file .rpt e rende disponibile il report source risultante al Report Viewer.
- **Report Viewer:** esegue il rendering di un report nel formato di esportazione scelto dall'utente fra PDF e RTF.

La servlet gestisce il caso d'uso di generazione del report raccogliendo i parametri necessari alla generazione del report, gestisce l'eventuale esportazione del report nei formati previsti. Il report generato viene reso visibile all'utente tramite il Report Viewer, il tutto nel pieno rispetto del pattern MVC.

6.1.4. HL7 CDA

Nell'ambito del progetto Sinfonia il trattamento dei dati clinico-sanitari si basa sulla gestione di documenti XML in un tipico approccio document-oriented. La struttura di tali documenti XML è conforme allo standard HL7v3/CDA rev.2 e costituisce il veicolo di trasporto dei dati clinici in documenti (atti sanitari) firmati digitalmente (nel seguito documenti CDA).

CDA (Clinical Document Architecture) è una specifica XML di HL7 riconosciuta dall'ANSI per la rappresentazione standard di dati clinici. HL7 CDA è uno standard diffuso a livello internazionale, di conseguenza, la sintassi e la semantica dei metadati ha una valenza "globale" e la loro implementazione rappresenta un efficace strumento verso l'interoperabilità documentale.

Si rimanda a documenti Specifica dei Requisiti Software delle aree applicative per i tipi di CDA trattati.

6.2. Business Tier

Il Business Tier del sistema Sinfonia è responsabile della realizzazione della logica di business.

L'implementazione di questo strato si basa sull'EJB-tier di un Application Server J2EE compliant, contenente l'EJB Container che costituisce l'ambiente operativo dei componenti della logica di business nella tecnologia J2EE, gli Enterprise Java Beans (EJB).

L'EJB-tier prevede un doppio strato di EJB:

- lo strato di EJB di facciata, organizzati per entità di business, che espongono metodi in corrispondenza uno a uno con la corrispondente richiesta proveniente dal Presentation Tier;
- lo strato di EJB locali (non visibili all'esterno dell'EJB-container) che espongono, ad uso esclusivo degli EJB di facciata, servizi (transazionali o di accesso ai dati in lettura) organizzati per entità di business.

Lo strato di EJB locali si avvale dei servizi di classi helper che provvedono alla implementazione di controlli formali e applicativi ed alla implementazione di trasformazioni o conversioni di formato.

L'accesso al database dedicato da parte degli EJB avviene tramite il componente DAO (Data Access Object) che viene strutturato in modo che visto dagli EJB locali esporrà un suo sottotipo per ogni business entity. Ognuno di tali sottotipi implementerà le interfacce contenenti gli statement elementari relativi ad ogni singola relational table di cui utilizza i dati per i propri scopi. La transazionalità degli accessi al database sarà gestita dal container degli EJB.

Si sceglie di usare Session EJB di tipo Stateless per due ragioni:

- le richieste provenienti dal Presentation Tier sono di tipo stateless;
- gli Stateless Session EJB sono più efficienti e performanti.

6.3. Elaborazioni Batch

Per elaborazione batch si intende un qualsiasi processo la cui esecuzione è asincrona rispetto alla richiesta attivata dall'utente mediante la web application e che durante la sua esecuzione non richieda interazione con l'utente. Pertanto non necessita di un ambiente operativo J2EE essendo sufficiente l'utilizzo di una JVM.

I casi d'uso che prevedono tali elaborazioni consentono all'utente, tramite interfaccia web, di richiedere una elaborazione batch consentendogli di esplicitare, se necessario, gli opportuni parametri di input. Successivamente un operatore di back office analizza le richieste pervenute e avvia l'esecuzione del processo batch. A processo concluso viene memorizzata su database l'avvenuta esecuzione del processo e l'esito (elaborato con successo o elaborato con errore). Lo stato della elaborazione viene reso disponibile all'utente mediante una funzionalità ad hoc.

L'autorizzazione dell'operatore di back office all'avvio di un processo batch è conseguente alla sua identificazione e autenticazione, mediante username e password, al sistema operativo ed al DBMS.

Dal punto di vista architetturale, le componenti che entrano in gioco nella richiesta e nell'esecuzione del batch sono distribuite su due layer:

- **Presentation Layer** che realizza i casi d'uso di richiesta di elaborazione batch;
- **Business Layer** che contiene la logica di start e di esecuzione del batch, che può essere un processo batch oppure un report batch, cioè un report non interattivo.

6.3.1. Presentation Layer

Il presentation layer è costituito dalle componenti di interfaccia web che consentono all'utente di richiedere l'esecuzione asincrona di un processo batch, fornendo gli opportuni parametri al processo. Tale richiesta, che è a tutti gli effetti un caso d'uso del sistema, viene memorizzata insieme ai parametri di input su database. In un momento successivo la richiesta viene presa in carico da un operatore di back office che avvia il processo batch, ne controlla l'esecuzione, ne analizza l'output e tramite interfaccia web può apporre un flag "visto" alle richieste evase nella tabella delle richieste dove è anche presente un flag di stato che prevede i seguenti valori:

1. "IN ELABORAZIONE" all'atto della richiesta di esecuzione del batch;
2. "ELABORATO CORRETTAMENTE" dopo la corretta esecuzione del batch;
3. "ELABORATO CON ERRORI" se si sono verificati errori nella elaborazione del batch;
4. "ELABORATO CORRETTAMENTE DATI NON TROVATI" se l'elaborazione si è conclusa correttamente ma non sono stati riscontrati dati;
5. "ANNULLATA" quando l'elaborazione viene annullata dall'operatore.

Ciascun processo batch provvede a registrare il proprio cambio di stato e produce un file di log consultabile dall'operatore di back office.

L'operatore che aveva effettuato la richiesta di esecuzione del processo batch può controllarne l'avvenuta esecuzione tramite interfaccia web, e, se previsto, eseguire il download del risultato del processo (report).

Dal punto di vista architetturale il presentation layer è del tutto simile ad un qualsiasi altro caso d'uso. Quindi per i casi d'uso di richiesta elaborazione batch valgono le scelte progettuali e architetturali effettuate per il presentation layer dell'intero sistema.

6.3.2. Business Layer

Il Business Layer delle elaborazioni batch si divide in due categorie diverse sia per il tipo di processo, sia per il risultato prodotto sia per la tecnologia adottata:

- ***Processo Batch***: elaborazione che produce dati che vengono memorizzati sul db;
- ***Report Batch***: elaborazione che produce file report in formato ASCII o pdf.

0.1.1.5 Processo Batch

Un processo batch:

- elabora dati presenti nel database in funzione dei parametri input forniti dall'utente che ha richiesto l'esecuzione del processo batch, anch'essi presenti nel database;
- produce risultati che vengono memorizzati nel database (a titolo esemplificativo rientrano in questa categoria i batch di calcolo delle competenze delle diverse categorie di medici).

Un processo batch è costituito da un programma Java avviato da console, in un momento successivo alla richiesta, da un operatore di back office. E' un processo la cui esecuzione avviene all'interno di una macchina virtuale Java JVM standard J2SE. Per motivi di efficienza, dato il massiccio uso di accessi al database, la JVM di esecuzione può essere quella di una macchina in connessione intranet con Database Server se non una JVM presente sul database server.

0.1.1.6 Report Batch

Un report batch:

- elabora dati presenti nel database;
- l'elaborazione è in funzione dei parametri input forniti dall'utente che ha richiesto l'esecuzione del report batch, anch'essi presenti nel database;
- il layout e la logica di reporting del report sono definiti in un file con estensione .rpt che risiede in un path accessibile dalla macchina in cui risiede la JVM di esecuzione del report batch;
- produce, come risultato della elaborazione un report, cioè un file di dati in formato conforme alle specifiche definite per il report; i dati sono disposti con un layout predefinito in fase di progettazione del report.

Il report batch è costituito da un programma Java avviato da console, in un momento successivo alla richiesta, da un operatore di backoffice, quindi è un processo la cui esecuzione avviene all'interno di una macchina virtuale Java JVM standard J2SE. Per motivi di efficienza, dato il massiccio uso di accessi al database, la JVM di esecuzione può essere quella di una macchina in connessione intranet con Database Server se non una JVM presente sul database server.

Tale programma utilizza le librerie denominate Java Reporting Component Ver. 11.8 (JRC) di Business Object per integrare nell'applicazione la generazione dei report.

Il file con estensione .rpt, che definisce il layout di un particolare report, risiede in un path accessibile dalla macchina in cui risiede la JVM di esecuzione del report batch ed è il prodotto della fase di design del report.

Il design di ogni report viene eseguito con l'ausilio del tool di sviluppo Report Designer Crystal Reports 2008.

7. Componenti per la Firma Digitale

Il processo di firma digitale si basa su PKI (Public Key Infrastructure) che standardizza l'insieme di tecnologie, infrastrutture, e pratiche di management richieste per abilitare e rendere effettivo l'uso di autenticazione, cifratura e firma elettronica basate su chiave pubblica in applicazioni distribuite e garantisce:

- **Autenticazione**, cioè certezza dell'identità di una persona o di un'applicazione
- **Integrità dei dati**, per dimostrare che non vi sono state manipolazioni dei dati durante il trasporto
- **Non ripudio**, per dimostrare l'origine da cui proviene l'informazione.

Oltre all'infrastruttura PKI è necessaria l'emissione, per ogni utente, di una CNS (Carta Nazionale dei Servizi) o token USB.

7.1. CNS, PKCS#11, Wrapper Java, card reader e PC/SC driver

L'interfacciamento fra le applicazioni e la CNS/token è basato sullo standard PKCS#11 implementato tramite librerie software fornite dal produttore della specifica CNS/token utilizzata. Le librerie PKCS#11 comunicano con la CNS tramite una periferica (card reader) collegata ad una porta seriale oppure ad una porta USB del PC, e con i token USB direttamente tramite porta USB. I driver dei card reader vengono forniti dal produttore e sono basati sullo standard PC/SC. Poiché le librerie PKCS#11 sono librerie software a basso livello, la Certification Authority, che tipicamente fornisce sia lettore sia smartcard con credenziali di firma (coppia di chiavi asimmetriche RSA con relativo certificato) conformi alle direttive AIPA, CNIPA, DigitPA e ora AgID fornisce anche librerie che fungono da wrapper per le librerie PKCS#11, implementate con i più diffusi linguaggi di programmazione, incluso Java (come evidenziato in figura).

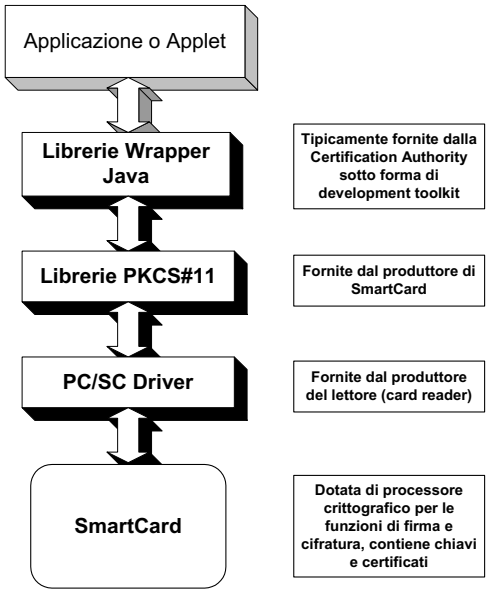

Figura 2 – Gestione CNS

L'eventualità della presenza di più lettori con più CNS inserite viene risolta dallo standard PKCS#11 tramite lo Slot. Lo Slot rappresenta il lettore dal punto di vista logico e pertanto esisteranno tanti slot per quanti lettori di Smartcard vengono rilevati nel sistema.

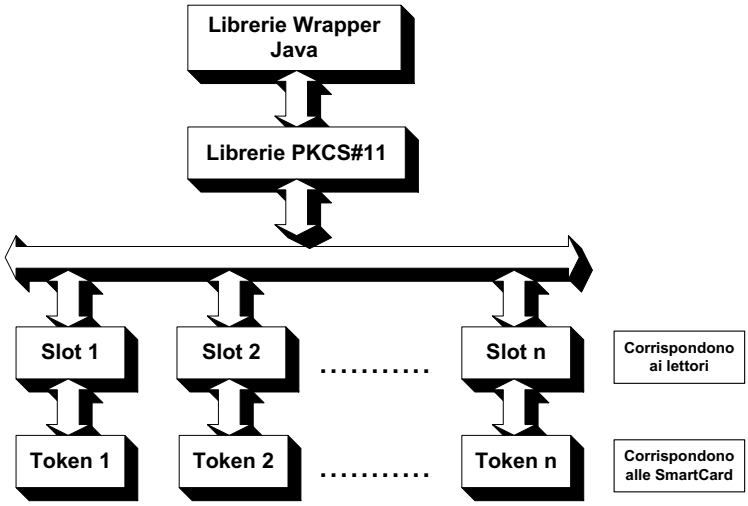

Figura 3 – Gestione di più CNS con Slot

La presenza di una infrastruttura PKI con relativa Certification Authority e di una piattaforma CNS con relativo CardOS, librerie PKCS#11 e Wrapper Java consentono l'implementazione del processo di firma. La fornitura Sinfonia supporta CNS con certificati di autenticazione e di firma rilasciati da CA inserite nel registro dell'AgID.

7.2. Formato dei documenti prodotti e standard di firma

I documenti prodotti da Sinfonia da sottoporre al processo di firma sono dei seguenti formati:

- CDA HL7 come, ad esempio, prescrizioni, erogazioni e referti;
- PDF per qualsiasi altro tipo di documento per cui è necessaria l'apposizione della firma.

E' prevista l'adozione dei seguenti standard di firma riconosciuti con validità legale dal AgID (ex DigitPA e ex CNIPA):

- p7m per il formato HL7 CDA;
- PDF Signature secondo gli standard Adobe Systems Inc. approvati dal AgID (ex DigitPA e ex CNIPA).

Nei documenti SRS di ogni area applicativa verrà inserito l'elenco dei documenti soggetti a firma, il loro formato e la tipologia di firma.

7.3. Controlli di validità su un documento firmato

I controlli effettuati lato server su un documento firmato sono:

a) integrità del documento firmato;

b) validità della firma apposta tramite un certificato digitale X.509v3;

c) integrità e validità del certificato di firma dell'utente.

La validità del certificato digitale X.509v3 di firma prevede che il certificato:

- sia rilasciato da una CA riconosciuta: verifica che il certificato di root della CA, presente nel certificato X.509v3, sia contenuto nel TrustStore del server;
- sia autentico: verifica della validità della firma apposta sul certificato dalla CA;
- sia temporalmente valido;
- contenga il codice fiscale del titolare della CNS/token all'interno del commonName;
- non sia interdetto all'uso (sospeso o revocato): lo stato interdetto di un certificato è verificato mediante l'analisi della CRL indirizzata dalla URI presente nel certificato digitale X.509v3 stesso.

Nel caso di CRL con validità scaduta, la verifica dello stato di interdizione (certificato sospeso, certificato revocato) è subordinata al parametro di sistema che indica se considerare valida la CRL anche se con validità scaduta.

7.4. Il processo di firma digitale

I documenti SRS delle aree applicative riportano nella descrizione dei casi d'uso, ove applicabile, la esecuzione del processo di firma digitale. I documenti SRS delle aree applicative non riportano il dettaglio del processo di firma digitale che risulta essere descritto nella seguente sezione.

Il processo di firma via web prevede la visualizzazione del documento da firmare e, a seguito della conferma da parte dell'utente, l'apposizione della firma digitale. La responsabilità della visualizzazione del documento è a carico della Servlet di Firma. L'apposizione della firma digitale sul documento è a carico dell'Applet di Firma. La responsabilità dei controlli sul documento firmato è a carico della Servlet di Firma.

Insieme all'applet vengono scaricate dal browser le librerie necessarie alle interazioni con la smartcard e l'applet viene eseguita dalla JVM (Java Virtual Machine) all'interno del browser. Essendo una applet che interagisce con componenti hardware e software installati sulla workstation, l'applet deve essere firmata (signed) affinché venga verificata la provenienza e l'autenticità.

La figura che segue descrive la sequenza delle interazioni tra componenti e attori che "partecipano" a questo processo.

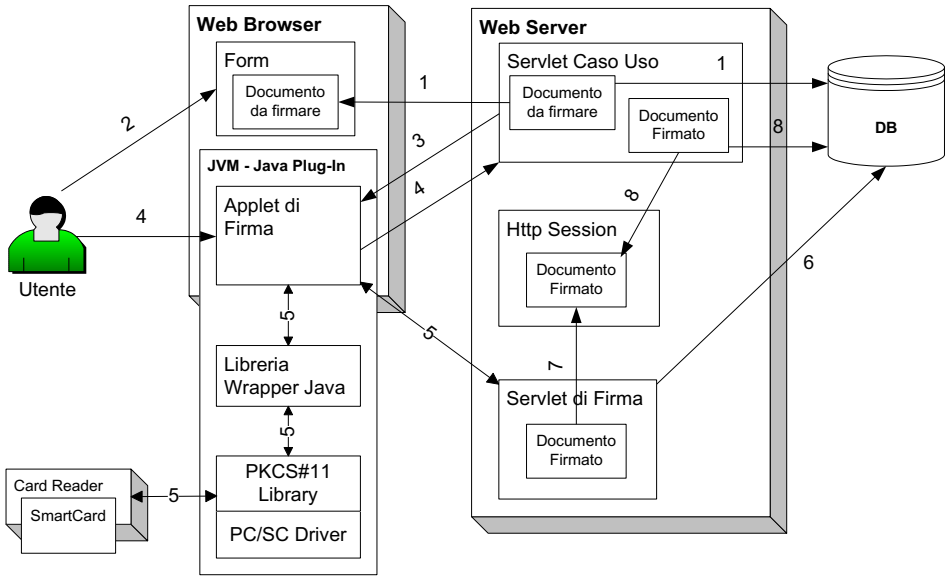

Figura 4 – Processo di Firma

Flusso Principale

1. la Servlet del Caso d'Uso produce il documento conservandone temporaneamente una copia nel DB ed invoca la Servlet di Firma che provvede alla visualizzazione e alla richiesta di conferma;
2. l'utente conferma la volontà di firmare;
3. la Servlet di Firma invia la pagina contenente l'Applet di Firma;
4. l'Applet di Firma scarica il documento dal server, propone – nel caso siano presenti più certificati di firma sulla smartcard - la selezione di uno dei certificati di firma e richiede la digitazione del pin di autenticazione della smartcard e del pin di firma;
5. l'Applet di Firma invoca le funzionalità crittografiche della smartcard, firma il documento e restituisce il documento firmato alla Servlet di Firma;
6. la Servlet di Firma esegue i controlli di validità del certificato di firma e controlla la validità del documento firmato e la sua corrispondenza con il documento da firmare;
7. la Servlet di Firma conserva il documento firmato nel DB;
8. la Servlet del Caso d'Uso recupera il documento firmato dal DB rimuovendo quello temporaneo non firmato.

Flusso Alternativo: *Rifiuto della volontà di firmare*

Se l'utente NON conferma la volontà di firmare il documento, il controllo ritorna Servlet del Caso d'Uso. Il documento non viene né firmato né archiviato. La Servlet del Caso d'Uso deve rimuovere il documento da firmare temporaneamente conservato nel DB.

Flusso Eccezionale: ***Errore di inserimento del PIN***

Se l'utente digita un pin errato, l'Applet di Firma visualizza un messaggio di pin errato. Alla terza digitazione consecutiva di pin errato l'Applet di Firma visualizza un messaggio dove si comunica all'utente che la smartcard è inservibile.

Flusso Eccezionale: ***Esito negativo della verifica di firma.***

La Servlet di Firma invalida la firma se

a) non sono superati i controlli di validità del certificato di firma;

b) non sono superati i controlli di validità del documento firmato;

c) il documento firmato non è identico a quello conservato temporaneamente nel DB prima dell'inizio del processo di firma.

La Servlet di Firma informa dell'esito negativo di verifica di validità della firma.

8. Gestione Utenti, Identificazione, Autenticazione ed Autorizzazione

Il sistema consente di accedere a tutti i servizi offerti, con meccanismi di autenticazione basati su username / password o smart card crittografiche (CIE, CNS, CRS, etc.). Il sistema gestisce, con un'interfaccia, la gestione applicativa di ruoli e profili, che sarà possibile scegliere in fase di accesso. Gli utenti delle singole applicazioni presenti nella fornitura Sinfonia accedono al sistema tramite un unico Front-End Web, la cui implementazione e autenticazione viene demandata al WSO2 Identity Server che interagisce in maniera federata con il Single-Sign On (SSO) regionale. Si rimanda a tale paragrafo per ulteriori approfondimenti in merito.

8.1. Identificazione, autenticazione ed autorizzazione degli utenti

L'identificazione, l'autenticazione e l'autorizzazione costituiscono i passi del processo attraverso il quale una entità accerta la corretta, o presunta, identità digitale di un utente.

Tutte le componenti applicative della fornitura Sinfonia per la fase di identificazione e autenticazione si integrano con il sistema di SSO.

La componente *Gestione Utenti*, invece, fornisce i servizi di amministrazione necessari a definire e profilare utenti nonché i servizi per l'autorizzazione dell'utente all'utilizzo delle singole funzionalità/servizi e alla visibilità di dati sensibili. La componente inoltre ha la responsabilità di produrre la reportistica analitica e riepilogativa relativa all'utilizzo dei servizi, all'assegnazione dei ruoli ed alla distribuzione degli utenti rispetto a ruoli e aziende sanitarie.

8.1.1. Definizione e Profilazione degli Utenti

La definizione e la profilazione di un utente è basata sulla sua identità e sui ruoli, detti Ruoli Istituzionali, che l'utente può assumere all'interno di una o più strutture.

La funzionalità di definizione e profilazione dell'utente fornisce la possibilità, ad un operatore autorizzato tramite interfaccia web, di definire o modificare più corrispondenze fra identità dell'utente, Ruolo Istituzionale e struttura in cui quel ruolo viene ricoperto.

La definizione e profilazione dell'utente in questi termini pone le basi per il processo di autorizzazione basata su ruolo e rende il meccanismo utilizzabile sia nel contesto Regionale che nel contesto aziendale.

8.1.2. Autorizzazione

Il processo di definizione delle autorizzazioni è fondato sui Ruoli Istituzionali e alla attribuzione a ciascun Ruolo Istituzionale di uno o più Ruoli Operativi. Un Ruolo Operativo viene definito come raggruppamento logicamente coerente di servizi ed è gestito solo a livello di backoffice.

Il sistema autorizzerà l'utente alla fruizione di un servizio solo se tale servizio è associato al Ruolo Operativo attribuito al Ruolo Istituzionale ricoperto dall'utente stesso (assegnato nella fase di definizione e profilazione).

Si rimanda alla documentazione delle diverse aree applicative per la semplificazione del modello dei ruoli attualmente previsto.

A livello di backoffice sarà possibile definire o modificare, tramite interfaccia web, il mapping Ruoli Istituzionali – Ruoli Operativi, eliminare un Ruolo Istituzionale, creare un nuovo Ruolo

Istituzionale ed associargli uno o più Ruoli Operativi. I Ruoli Operativi sono predefiniti nel sistema in quanto, per loro stessa natura, legati ai servizi che il sistema offre.

9. Identificazione ed Autenticazione per i servizi di cooperazione

Obiettivo del capitolo è quello di illustrare i meccanismi e le specifiche con cui il sistema Sinfonia implementa i meccanismi di identificazione, autenticazione ed autorizzazione dei sistemi applicativi cooperanti che inoltrano richieste di servizio in modalità web services e in modalità SPCoop.

Nel seguito si intende per Sistema Fruitore un sistema applicativo che fruisce dei servizi di cooperazione di Sinfonia, che funge da Sistema Erogatore. Vengono considerati sistemi fruitori, alla stregua di qualunque altro sistema applicativo cooperante, le stesse componenti applicative ed i sistemi infrastrutturali di Sinfonia per le richieste di servizio che questi inoltrano verso le altre componenti applicative di Sinfonia.

L'identificazione, l'autenticazione e l'autorizzazione del Sistema Fruitore, per ciascun servizio di cooperazione esposto, sono implementati dall'ESB di WSO2.

9.1. Il processo complessivo

Di seguito è illustrato il processo complessivo di interazione tra sistemi cooperanti relativi al processo di identificazione, autenticazione e autorizzazione nel caso di invocazione dei servizi esposti dal Sistema Erogatore Sinfonia attraverso ESB:

1. Il middleware ESB, attraverso proxy service, riceve un messaggio SOAP contenente il certificato X.509v3 del Sistema Fruitore in conformità alla specifica "X.509 Certificate Token Profile" fornito da WS-Security. L'integrità del messaggio è garantita dalla firma digitale apposta con certificato X.509v3.
2. Il middleware ESB verifica l'integrità sintattica del messaggio verificando la rispondenza all'xsd. In particolare viene estratto dal messaggio di input il token X.509v3, rappresentante il Sistema Fruitore. Successivamente si applicano tutti i controlli necessari a verificare la validità del certificato. Ottenuta la validazione del certificato, il middleware ESB accerta la validità dell'identità del Sistema Fruitore interfacciandosi con i servizi esposti dalla propria IdP. In particolare il middleware ESB verifica che il common-name del certificato X.509v3 sia presente ed abilitato nell'Anagrafe dei Sistemi Fruitori autorizzati ad interagire con il Sistema Erogatore.
3. Il middleware ESB, superati tutti i controlli del passo precedente, provvede a reinoltrare la request verso il Sistema Erogatore aggiungendo il CN dell'integratore negli attributiAutorizzativi e inserendo una propria security.
4. Il Sistema Erogatore riceve il messaggio SOAP contenente il certificato X.509v3 del middleware ESB in conformità alla specifica "X.509 Certificate Token Profile" fornito da WS-Security. L'integrità del messaggio è garantita dalla firma digitale apposta con certificato X.509v3.

5. Il Sistema Erogatore verifica l'integrità sintattica del messaggio verificando la rispondenza all'xsd. In particolare viene estratto dal messaggio di input il token X.509v3, rappresentante il Sistema Fruitore. Successivamente si applicano tutti i controlli necessari a verificare la validità del certificato. Ottenuta la validazione del certificato, il Sistema Erogatore accerta la validità dell'identità del Sistema Fruitore interfacciandosi con i servizi esposti dalla propria Anagrafica Sistemi Fruitori. In particolare il Sistema Erogatore verifica che il common-name del certificato X.509v3 sia presente ed abilitato nell'Anagrafe dei Sistemi Fruitori autorizzati ad interagire con il Sistema Erogatore. In questo caso il middleware figura come Sistema Fruitore.
6. Il Sistema Erogatore, superati tutti i controlli del passo precedente ed in conformità con quanto definito nei successivi paragrafi relativi all'autorizzazione per i servizi di cooperazione, provvede ad erogare il servizio richiesto.
7. Il Sistema Erogatore accerta la conformità della richiesta applicativa rispetto alle eventuali politiche di sicurezza aggiuntive specifiche del servizio applicativo richiesto.
8. Il messaggio SOAP di response inviato al middleware ESB, e poi di conseguenza al Sistema Fruitore soddisferà, analogamente al messaggio SOAP di request, la specifica "X.509 Certificate Token Profile" fornito WS-Security.

9.1.1. Identificazione ed autenticazione dei Sistemi Fruitori

L'identificazione e l'autenticazione del Sistema Fruitore è basata sull'utilizzo del certificato X.509v3 di autenticazione del Sistema Fruitore, secondo lo standard Web Services Security *X.509 Certificate Token Profile* (http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-x509TokenProfile.pdf).

L'identificazione del sistema fruitore deve individuare in modo univoco e certo lo specifico sistema che sta invocando il servizio esposto. Ciò significa che due installazioni distinte di uno stesso prodotto software, anche nello stesso dominio organizzativo ovvero anche sullo stesso sistema server fisico, sono identificate tramite due identità differenti e, quindi, tramite due distinti certificati di autenticazione X.509v3.

Ne consegue che ogni sistema fruitore, sia interdominio che intradominio, deve essere dotato di un certificato X.509v3 il cui commonname deve essere censito nell'anagrafe dei certificati X.509v3 associati ai sistemi fruitori autorizzati ad interagire con il sistema erogatore.

Come da standard WS-Security, il certificato di autenticazione X.509v3 sarà utilizzato per la firma di parti del messaggio SOAP. Nel caso specifico si è scelto di firmare:

- il tag <Timestamp>, previsto nell'header del messaggio SOAP;
- il tag <To>, previsto nell'header del messaggio SOAP;
- il tag <Action>, previsto nell'header del messaggio SOAP;
- il tag <MessageID>, previsto nell'header del messaggio SOAP;
- il tag <ReplyTo>, previsto nell'header del messaggio SOAP;
- il tag <AttributiAutorizzativi>, presente nell'header del messaggio SOAP;
- il tag del contenuto applicativo, primo ed unico figlio del tag <Body>.

9.1.2. Integrità del messaggio

L'integrità del messaggio SOAP associato all'invocazione di un web service assicura che i messaggi non siano intercettati e alterati durante lo scambio fra Sistema Fruitore e Sistema Erogatore.

In Sinfonia è garantita l'integrità delle parti fondamentali del messaggio, sottoponendo a processo di firma:

- il tag <Timestamp>, previsto nell'header del messaggio SOAP;

- il tag <To>, previsto nell'header del messaggio SOAP;

- il tag <Action>, previsto nell'header del messaggio SOAP;

- il tag <MessageID>, previsto nell'header del messaggio SOAP;

- il tag <ReplyTo>, previsto nell'header del messaggio SOAP;

- il tag <AttributiAutorizzativi>, presente nell'header del messaggio SOAP;

- il tag del contenuto applicativo, primo ed unico figlio del tag <Body>.

9.1.3. Non ripudio del messaggio

Il non ripudio di un messaggio trasmesso dal Sistema Fruitore al Sistema Erogatore è garantito dall'autenticazione che una firma è in grado di offrire.

Infatti, l'univocità della firma digitale applicata ad un messaggio impedisce che il proprietario della firma disconosca le informazioni contenute nel messaggio firmato.

In Sinfonia, il non ripudio del messaggio è garantito dall'applicazione della firma digitale da parte del Sistema Fruitore al messaggio SOAP-Request per:

- il tag <Timestamp>, previsto nell'header del messaggio SOAP;

- il tag <To>, previsto nell'header del messaggio SOAP;

- il tag <Action>, previsto nell'header del messaggio SOAP;

- il tag <MessageID>, previsto nell'header del messaggio SOAP;

- il tag <ReplyTo>, previsto nell'header del messaggio SOAP;

- il tag <AttributiAutorizzativi>, presente nell'header del messaggio SOAP;

- il tag del contenuto applicativo, primo ed unico figlio del tag <Body>.

9.1.4. Mantenimento delle informazioni della richiesta di servizio

Allo scopo di mantenere le informazioni relative alla richiesta del servizio, in maniera funzionale a dimostrare a terzi la legittimità dell'operato del sistema Sinfonia, solo per i servizi critici (ad esempio quelli che trattano dati sensibili), saranno memorizzati su una apposita tabella persistente i seguenti dati:

- Nome del servizio (prelevato dagli attributi autorizzativi)
- Codice identità utente (prelevato dagli attributi autorizzativi)
- Ruolo istituzionale (prelevato dagli attributi autorizzativi)

- Data e ora richiesta
- Identificativo del sistema fruitore (CommonName del certificato)
- Messaggio SOAP-Request, comprensivo di Header.

La memorizzazione su questa tabella può essere abilitata/disabilitata in qualsiasi momento dall'amministratore di sistema intervenendo su un flag di abilitazione/disabilitazione definito per ogni servizio.

In considerazione delle ripercussioni che tale scelta può avere sul sistema in termini di occupazione dei volumi e di ulteriore carico transazionale, la definizione dei servizi per i quali saranno mantenute le suddette informazioni sarà effettuata congiuntamente con il committente.

9.1.5. Riservatezza del messaggio

La riservatezza del messaggio SOAP deve garantire che i dati trasmessi non siano alterati durante lo scambio e non siano interpretabili da alcuno con l'eccezione di chi ha il permesso di accedervi.

Lo strumento per garantire la riservatezza del messaggio è l'utilizzo di SSL (Secure Socket Layer), che permette di creare un canale protetto per lo scambio di dati tra due Sistemi.

Tutti i servizi di Sinfonia esposti come web services standard sono fruibili su protocollo SOAP su HTTPS.

9.1.6. Firma dei messaggi di risposta

Per garantire integrità, non ripudio e riservatezza dei messaggi di risposta, nelle SOAP-Response saranno firmati, utilizzando il certificato X509v3 della singola istanza logica di Sinfonia e del middleware ESB i tag:

- <Timestamp>
- il tag <To>, previsto nell'header del messaggio SOAP;
- il tag <Action>, previsto nell'header del messaggio SOAP;
- il tag <MessageID>, previsto nell'header del messaggio SOAP;
- il tag <RelatesTo>, previsto nell'header del messaggio SOAP;
- il tag del contenuto applicativo, primo ed unico figlio del tag <Body>.

9.1.7. Identità dell'utente

L'individuazione dell'identità dell'utente del sistema fruitore è una responsabilità del sistema fruitore. Ciò comporta che:

- il sistema erogatore non dispone dell'elenco delle identità degli utenti finali dei sistemi fruitori;
- il sistema erogatore si fida, cioè prende atto, dell'identificazione, dell'autenticazione dell'utente eseguita dal sistema fruitore e dell'autorizzazione all'invocazione del servizio esposto.

L'identità dell'utente è rappresentata nella richiesta di servizio tramite un identificativo significativo per il sistema fruitore che consenta allo stesso sistema fruitore, in caso di necessità, di risalire all'identità reale dell'utente finale. E' raccomandabile l'utilizzo del codice fiscale.

L'identità dell'utente è presente in ogni richiesta di servizio, negli attributi autorizzativi presenti nell'header del messaggio (tag <IdentificativoUtente>), e deve essere utilizzata dal sistema Sinfonia per fini di tracciamento delle operazioni e, quando necessario, per finalità legate allo specifico servizio applicativo richiesto.

9.1.8. Identificazione ed Autenticazione della Porta di Dominio

L'identificazione e l'autenticazione della Porta di Domino mittente è a carico della componente PDD Scatel 3 allocata sulla Porta di Dominio destinataria del messaggio. I meccanismi sono quelli messi a disposizione dalla stessa PDD Scatel 3.

9.2. Autorizzazione per i servizi di cooperazione.

Ogni invocazione di un servizio applicativo esposto dal sistema erogatore Sinfonia è soggetta ad un processo autorizzativo la cui finalità è verificare che il servizio esposto sia invocabile dall'utente del Sistema Fruitore.

Il processo autorizzativo, sia nel caso intradominio sia nel caso interdominio, per ciascun servizio di cooperazione esposto da Sinfonia, è responsabilità di Sinfonia e non è una responsabilità delle porte di dominio.

Tutti gli attributi autorizzativi necessari al processo di autorizzazione devono essere contenuti nell'header del messaggio SOAP-Request.

Il processo autorizzativo è disaccoppiato dalla logica applicativa che implementa il web service e la anticipa temporalmente verificando il match tra gli attributi autorizzativi e il nome del servizio richiesto. Ciò non toglie che la logica applicativa possa estendere il processo autorizzativo implementando un controllo degli accessi basato sullo specifico contenuto applicativo richiesto e/o su altre politiche di natura meramente applicativa.

Pertanto il processo autorizzativo agisce su diversi livelli al fine di garantire una maggiore granularità delle autorizzazioni sia a livello di singolo utente che di singolo Sistema Fruitore.

9.2.1. Autorizzazione del Sistema Fruitore all'uso di Sinfonia

Il middleware ESB dopo aver autenticato e identificato il Sistema Fruitore verifica che quest'ultimo sia abilitato all'invocazione dei web services esposti dal Sistema Erogatore. Il controllo si sostanzia nel verificare che il Sistema Fruitore oltre ad essere censito nell'Anagrafe Sistemi Fruitori abbia l'abilitazione ad interrogare i web services esposti dal Sistema Erogatore. Tale controllo consente di disabilitare l'erogazione di tutti i web service del Sistema Erogatore ad un determinato Sistema Fruitore che sia già censito nell'Anagrafe Sistemi Fruitori e che abbia un certificato X.509v3 valido.

9.2.2. Autorizzazione del Sistema Fruitore all'uso del servizio

Il Sistema Erogatore, dopo che middleware ESB abbia autenticato, identificato e autorizzato il Sistema Fruitore ad interagire con i web service del Sistema Erogatore, verifica che questo possa accedere allo specifico servizio invocato. Ogni sistema fruitore appartiene ad una classe di sistemi (ad esempio CUP, RIS, LIS etc..) e per ogni classe sono definiti i servizi cui la classe è abilitata mediante un caso d'uso di GUIAA.

In particolare il Sistema Erogatore verificherà che la classe a cui appartiene il Sistema Fruitore sia abilitata al servizio.

9.2.3. Autorizzazione dell'utente del Sistema Fruitore

Tale autorizzazione si avvale di una struttura dati denominata <AttributiAutorizzativi>, presente nell'header del messaggio SOAP, che contiene un gruppo fisso minimo di attributi. La struttura dati <AttributiAutorizzativi> potrà essere estesa per rispondere a nuove necessità quali ad esempio:

- definire ulteriori attributi obbligatori
- definire ulteriori attributi da analizzare durante il processo autorizzativo di specifici servizi esposti.

La struttura <AttributiAutorizzativi>, nella sua forma minima risulta così definita:

<AttributiAutorizzativi>

<IdentificativoServizio/>

<IdentificativoUtente/>

<RuoloIstituzionale/>

</AttributiAutorizzativi>

Ove:

IdentificativoServizio	Nome del servizio invocato.
IdentificativoUtente	Identificatore dell'utente finale la cui attività ha determinato l'invocazione del servizio esposto.
RuoloIstituzionale	Ruolo istituzionale dell'utente finale.

La struttura è firmata con il certificato X.509v3 del sistema fruitore per garantire l'integrità e il non ripudio delle informazioni sulla cui base si attua il processo autorizzativo.

In particolare il processo autorizzativo verifica che il ruolo istituzionale posseduto dall'utente, così come asserito dal sistema fruitore, possa invocare il servizio. In altri termini il Sistema Erogatore verifica che la coppia servizio – ruolo operativo(i) sia abilitata, dove ruolo operativo(i) sia uno dei ruoli operativi definiti a partire dal ruolo istituzionale dichiarato dal Sistema Fruitore. Il processo di risoluzione del ruolo Istituzionale in più ruoli Operativi è a carico del Sistema Erogatore.

L'identità dell'utente finale, così come asserita dal sistema fruitore, non interviene nel processo autorizzativo generale del servizio esposto. L'identità dell'utente è utilizzata, insieme alle informazioni della richiesta di servizio sottomessa, al minimo per scopi di tracciabilità.

L'identità dell'utente finale, così come asserita dal sistema fruitore, potrà tuttavia essere utilizzata per eseguire controlli autorizzativi complementari per uno specifico servizio esposto.

9.3. Riepilogo dei controlli eseguiti ad ogni richiesta di servizio di cooperazione

Il sistema Sinfonia provvederà ad ogni richiesta di servizio di cooperazione a:

- controllare la validità sintattica del messaggio SOAP-Request rispetto allo schema xsd del web service invocato (inclusi tutti i tag xml necessari all'espletamento del processo autorizzativo);
- controllare che la data/ora di creazione del messaggio non sia successiva alla data/ora di sistema (considerando un margine di tolleranza configurabile nella differenza di orario fra client e server);
- controllare che la data/ora di scadenza del messaggio non sia precedente alla data/ora di sistema (considerando una determinata soglia di tolleranza configurabile);
- verificare che siano stati firmati tutti i tag soggetti a firma;
- verificare la validità delle firme apposte;
- controllare che il certificato sia integro;
- controllare che il certificato non sia scaduto;
- controllare che la CA che ha emesso il certificato sia presente nel TrustStore presente sul file system del server;
- verificare che la CRL referenziata dal certificato sia presente sul file system del server;
- verificare che il certificato non sia revocato o sospeso (nel caso di CRL con validità scaduta, la verifica dello stato di interdizione è subordinato al parametro di sistema che indica se considerare valida la CRL anche se con validità scaduta);
- controllare l'identità del sistema fruitore: il commonname del certificato X.509v3 del sistema fruitore deve essere presente nell'anagrafe dei sistemi fruitori;
- controllare che il sistema fruitore sia abilitato all'invocazione di servizi esposti dal sistema erogatore;
- controllare che il sistema fruitore sia autorizzato all'utilizzo del servizio richiesto;
- controllare la presenza delle informazioni minime necessarie per il processo autorizzativo: presenza obbligatoria del tag <AttributiviAutorizzativi>, presenza e valorizzazione dei suoi tag figli;
- controllare che il servizio definito negli attributi autorizzativi sia identico al servizio presente nel Body del messaggio;
- controllare che il ruolo istituzionale definito negli attributi autorizzativi sia autorizzato all'utilizzo del servizio richiesto;
- effettuare gli ulteriori controlli applicativi peculiari del servizio.

10.Sicurezza accesso ai dati su DB

Poiché il database contiene i dati, la sua protezione è un aspetto di centrale importanza. In generale è sempre opportuno adottare ulteriori meccanismi di sicurezza esterni al DB, ma comunque aggiuntivi a quelli tipici del RDBMS. Oracle protegge i dati lì dove vengono conservati, all'interno del database, garantendone la protezione a un livello estremamente elevato. L'RDBMS Oracle offre numerose funzionalità di sicurezza, dall'autenticazione utente alla gestione del privilegio ed al controllo dell'accesso.

10.1. Identificazione, autenticazione ed autorizzazione

Gli utenti accederanno al database per il tramite dell'application server che si connetterà al database mediante un pool di connessioni ed utilizzando una specifica utenza. Pertanto il DBMS si *limiterà* ad identificare ed autenticare l'application server, delegando l'identificazione e l'autenticazione dell'utente ai meccanismi già descritti nel precedente paragrafo. All'RDBMS il rimane il compito di verificare le autorizzazioni (Grant) di accesso ai dati in base al ruolo istituzionale dell'utente. Il database del Sistema Sinfonia gestisce il sistema delle autorizzazioni verificando i privilegi assegnati dal database administrator ai ruoli istituzionali. Quindi, dopo la fase di identificazione ed autenticazione, l'utente può eseguire operazioni su un oggetto del database solo se è stato espressamente autorizzato dall'amministratore. Le autorizzazioni (vale a dire ruoli e privilegi) stabiliscono a quali tipi di dati un utente può accedere e che tipi di operazioni può effettuare su tali oggetti. L'utente può eseguire un'operazione su una risorsa o un oggetto di un database, ad esempio una tabella o una vista, solo se è stato autorizzato a compiere tale operazione dall'amministratore. Senza la concessione esplicita di privilegi, l'utente non può accedere ad alcuna informazione del database. Per garantire la sicurezza e la privacy dei dati, è necessario accordare all'utente solo i privilegi di cui necessita per svolgere le proprie funzioni di lavoro, senza concedergli permessi più ampi. Si tratta del cosiddetto "principio del privilegio minimo". Il database Sinfonia gestisce le autorizzazioni mediante privilegi e ruoli.

Riassumendo, il sistema Sinfonia è provvisto di un doppio sistema di profilazione dell'utente: uno al livello applicativo (solo l'utente autorizzato ad una specifica funzione potrà utilizzare la relativa funzionalità dell'applicazione) e il secondo all'interno del database (pur autorizzato al livello applicativo, vengono accordati all'utente i particolari privilegi sui dati necessari allo svolgimento del suo ruolo). Quindi, qualora un utente del sistema riuscisse anche ad aggirare il sistema di autorizzazione dell'applicativo, non riuscirebbe ad utilizzare le funzionalità in quanto non avrebbe le necessarie autorizzazioni al livello di database. Tale sistema garantisce quindi una profilazione dell'utente robusta a garanzia del principio di necessità nel trattamento dei dati che costituisce la precondizione di qualsiasi Sistema Informativo per la garanzia dei dati personali.

10.2. Cifratura

Alcuni requisiti di legge richiedono particolari misure quando dati personali o identificativi siano abbinati a informazioni di tipo sensibile. Ad esempio, l'accesso al nome dell'assistito in quanto tale può non richiedere particolari precauzioni, ma la combinazione del nome o del dato identificativo con informazioni di tipo sensibile può richiedere ulteriori misure di sicurezza come la crittografia.

I meccanismi di cripting dei dati sensibili al livello fisico nel database difendono da eventuali attacchi alla sicurezza che dovessero sopraggiungere dall'esterno del database stesso (ad es. qualcuno che riuscisse ad accedere direttamente al livello di Sistema Operativo ai datafile del database bypassando tutti i meccanismi di sicurezza messi a disposizione da Oracle).

Per questa eventualità il sistema RDBMS si avvale della feature **"Oracle Transparent Data Encription**" mediante il quale si può implementare in maniera trasparente il processo di cifratura dei dati sensibili direttamente nel motore RDBMS. Tale meccanismo consente di applicare la cifratura in maniera selettiva su specifiche colonne oppure a livello di intero tablespace per proteggere tabelle, indici e altri dati con algoritmi di cifratura robusti (3DES o AES fino a 256 bits) e senza la complessità della gestione di chiavi di cifratura.

Inoltre, anche la cifratura all'interno della Base Dati, se pur un valido meccanismo di sicurezza, non risolve il problema del furto dei supporti contenenti i backup. Per risolvere questo problema, il sistema, mediante la option Oracle Advanced Security, consentirà la cifratura dei backup direttamente sul supporto di salvataggio rendendo indecifrabili le informazioni in essi contenute in caso di accessi fraudolenti ai supporti sui quali vengono salvati i backup.

Il sistema implementerà la cifratura dei canali di comunicazione tra il database server e gli application server mediante gli algoritmi di cifratura (SSL/TSL) messi a disposizione da Oracle così come descritto nel successivo paragrafo 5.6.3 Sicurezza di Rete.

Il sistema implementerà, inoltre, il mascheramento dinamico dei dati sensibili mediante l'utilizzo della feature "Oracle Data Redaction" inclusa nella option Oracle Advanced Security. Sarà possibile creare policy che specificano le condizioni che devono essere soddisfatte prima che i dati vengano mascherati e restituiti all'utente. Durante la definizione di tali policy, si potrà specificare quali colonne mascherare, il tipo di protezione che deve essere applicato (totale, parziale, random ecc.) ed i ruoli istituzionali ai quali il dato viene mascherato.

In alternativa, laddove l'utilizzo della feature "Oracle Data Redaction" non consenta di soddisfare pienamente il requisito funzionale del mascheramento dati, si procederà al mascheramento dinamico dei dati sensibili in maniera applicativa.

10.3. Disaccoppiamento tra dati sensibili e anagrafici

Oltre al tradizionale meccanismo di ruoli e privilegi ed ai meccanismi di cripting dei dati sensibili si è ritenuto opportuno adottare, nella quasi totalità dei casi, anche il meccanismo di disaccoppiamento logico dei dati.

Tale meccanismo è ottenuto disaccoppiando le tabelle contenenti dati sensibili da quelle contenenti dati identificativi e correlandole tra loro mediante l'utilizzo di "codici non parlanti" (con tale terminologia ci si riferisce a codici non esplicativi della semantica del dato o a codici possano ricondurre immediatamente alla semantica del dato) oppure, utilizzando sempre codici non parlanti nei casi in cui i dati identificativi dell'utente e le informazioni sensibili siano contenute nella stessa tabella (es. codice fiscale dell'assistito e i codici esenzione sono contenuti nella stessa tabella, ma questi ultimi sono codificati con una codifica del tutto interna al sistema e non conosciuta dall'operatore). Solo in rare eccezioni, a causa della grossa mole dei dati e dell'elevata attività transazionale correlata, si è deciso di non applicare il disaccoppiamento per non impattare pesantemente sulle performance del sistema e si utilizzano quindi soltanto i meccanismi di ruoli e privilegi e di cripting dei dati.

11. Amministrazione Applicativa

L'area applicativa "Amministrazione Applicativa" si occupa della gestione e valorizzazione dei parametri di configurazione per tutte le aree applicative del sistema. L'area implementa componenti e metodi richiamabili dalle diverse altre aree applicative per la lettura dei valori dei parametri di configurazione. La lettura dei valori dei parametri può avvenire a diversi livelli del software:

- nello strato WEB
- nello strato EJB
- nei batch che girano nella JVM di Oracle

12.Tracciabilità e Monitoraggio

La componente Tracciabilità e Monitoraggio fornisce servizi trasversali a tutte le aree applicative, con l'obiettivo di raccogliere e successivamente analizzare informazioni riguardanti gli utenti che accedono al sistema, i servizi da essi richiesti, data ed ora della richiesta, modalità con cui accedono al sistema e fruiscono dei servizi, l'esito del servizio richiesto.

La componente ha la responsabilità di conservare traccia degli eventi che si verificano nell'ambito dei servizi implementati di Sinfonia. Le informazioni raccolte, opportunamente aggregate ed elaborate, permettono di:

- misurare i carichi di lavoro del sistema;
- monitorare il livello delle prestazioni (per ogni servizio il tempo medio di esecuzione);
- elencare le situazioni anomale;
- produrre rapporti sui servizi erogati relativamente ad aree applicative;
- monitorare le modifiche ai dati del database.

Questa componente non ha responsabilità di monitoraggio sistemistico che viene delegato alle componenti di ambiente e di sistema. Più precisamente il monitoraggio sistemistico, in cui si colloca il monitoraggio tecnico dei malfunzionamenti del software (eccezioni, messaggistica di errore, ecc), è delegato all'attività di tipo sistemistico di monitoraggio e controllo del funzionamento del sistema.

12.1. Ambito del tracciamento

Sono oggetto di tracciamento i seguenti eventi:

- utilizzo del singolo caso d'uso, query, report;
- singola funzione elementare del caso d'uso laddove applicabile, per es. per gli use case di tipo CRUD;
- attraversamento di ciascuna pagina di un caso d'uso, query o report;
- utilizzo del singolo web service sia per web services di consultazione sia di tipo transazionale;
- operazioni di CUD sulle persistenze del sistema.

Relativamente agli use case di consultazione, alle query e ai report si provvederà a tracciare i dati riguardanti i filtri di ricerca/consultazione. Non verrà effettuato alcun tracciamento del risultato.

Per quanto riguarda il tracciamento delle operazioni CUD sulle persistenze, si provvederà a tracciare, per ogni tabella sottoposta ad attività di tracing quanto segue:

- nome della tabella soggetta a tracciamento;
- tipo di operazione (insert, update, delete);

- istanze della tabella prima della modifica oppure in caso di inserimento di un nuovo record l'intera istanza inserita; nel caso di cancellazione l'istanza cancellata;
- istanza successiva alla modifica, nel caso di modifica;
- id dell'utente che ha effettuato la modifica;
- data ed ora della modifica.

12.2. Punti di tracciamento nel software

I punti di tracciamento per use case, query, report e web services sono allocati sul layer web.

12.3. Procedura di tracciamento

La strategia di tracciamento prevede due passi:

1. log su files (sul file system dell'application server) mediante utilizzo della libreria log4J;
2. procedura batch notturna che legge i files, inserisce i record di tracciamento in una apposita tabella del DB ed effettua il backup dei files.

12.4. Persistenza

Tutte le informazioni di tracciamento sono registrate in una apposita tabella del DB. Questa tabella è il punto di raccolta di tutte le informazioni dei canali di tracciamento. Le informazioni di auditing sono memorizzate su un'altra tabella del DB.

12.5. Canali di tracciamento

Per canale di tracciamento si intende il flusso di informazioni riguardanti una componente del sistema.

Data l'organizzazione del sistema Sinfonia in aree applicative, si definisce un canale di tracciamento per ogni area applicativa. Ad un canale di tracciamento corrisponderà uno specifico file che conterrà tutte le informazioni tracciate dal sistema secondo quanto definito dalle regole **Ambito del Tracciamento** e **Record di Tracciamento**. Dunque ogni componente software che contiene punti di tracciamento (ad esempio servlet del caso d'uso o servlethelper del caso d'uso) utilizzerà il canale di tracciamento dell'area applicativa di appartenenza.

12.6. Record di tracciamento

Il singolo record di tracciamento deve contenere, per ogni evento, i seguenti dati:

- il momento esatto in cui si è verificato l'evento (data, ora, minuti e secondi);
- l'identificativo dell'operatore che lo ha generato;
- l'identificativo della sessione http gestito dal web server (jboss) in corso;
- l'identificativo della transazione in corso, determinato dal software applicativo;

- la modalità di autenticazione (forte o debole);
- il nome del caso d'uso, query, report o web service;
- il nome della pagina attraversata;
- il nome dell'operazione (RICERCA, INSERIMENTO, CANCELLAZIONE, MODIFICA, DETTAGLIO);
- il flag inizio esecuzione operazione;
- il flag fine esecuzione operazione;
- il flag esito operazione;
- i valori della richiesta, nel caso di consultazione;
- l'eventuale stack-trace dell'eccezione.

12.7. Procedura di auditing

Il meccanismo di auditing prevede l'attivazione di trigger sul DBMS ad ogni operazione di inserimento, modifica e cancellazione sulle tabelle del DB di interesse. I trigger provvederanno ad inserire i dati definiti nella regola **Record di auditing** in una apposita tabella di auditing. Tali trigger prelevano la UserId dell'utente dalla tabella degli username. La tabella degli username viene valorizzata dalle transazioni applicative attraverso il metodo *pre()* degli EJB di tipo transazionale, invocato prima di ogni transazione. Alla fine della transazione viene poi invocato il metodo *post()* che provvede a cancellare la UserId dalla tabella degli username.

12.8. Record di auditing

Il singolo record di auditing deve contenere, per ogni evento, i seguenti dati:

- il momento esatto in cui si è verificato l'evento (data, ora, minuti e secondi);
- l'identificativo dell'operatore che lo ha generato (UserId);
- il nome della tabella impattata;
- il tipo di operazione (INSERT, UPDATE, DELETE);
- stringa contenente il valore del record prima della modifica;
- stringa contenente il valore del record dopo la modifica.