

# Soluzioni Infrastrutturali Open Source per il Sistema Pubblico di Cooperazione Applicativa

Giansalvatore Mecca    Alessandro Pappalardo

Salvatore Raunich    Il Gruppo di Sviluppo ICAR<sup>1</sup>

Dipartimento di Matematica e Informatica,  
Università degli Studi della Basilicata, Potenza, Italy

**Abstract** Il Sistema Pubblico di Cooperazione Applicativa (SPCoop) è stato standardizzato dal CNIPA nel 2005 e finalmente promette di raggiungere il livello del concreto dispiegamento. Questo articolo fornisce una panoramica sullo stato di avanzamento dell'iniziativa. Per cominciare introduce l'architettura di SPCoop e il suo raffinamento prodotto nell'ambito del recente progetto ICAR. Inoltre, confronta due soluzioni open source attualmente disponibili per la realizzazione dell'architettura.

## 1 Introduzione

E' da anni in corso un'iniziativa finalizzata alla standardizzazione dei meccanismi di cooperazione applicativa tra le pubbliche amministrazioni italiane. Già nel 2005 il CNIPA<sup>2</sup> ha pubblicato una serie di standard che vanno sotto il nome di SPC e SPCoop [6], rispettivamente “Sistema Pubblico di Connettività” e “Sistema Pubblico di Cooperazione”. Il primo definisce l'infrastruttura di rete ed i parametri per garantirne la sicurezza, il secondo, invece, definisce un framework per l'interoperabilità fra servizi applicativi. In particolare, SPCoop introduce gli elementi tecnologici che comporranno l'infrastruttura di cooperazione tra le pubbliche amministrazioni italiane, come la *porta di dominio*, la *busta e-Gov*, l'*accordo di servizio*, il *registro dei servizi*. Ad oggi SPCoop rappresenta lo standard nazionale per la cooperazione applicativa in ambiente di e-government.

Due diversi fattori inducono a credere che i tempi siano finalmente maturi per un concreto dispiegamento dell'infrastruttura nelle Pubbliche Amministrazioni. Da una parte, il CNIPA ha avviato una serie di gare per la fornitura di servizi di cooperazione applicativa alle pubbliche amministrazioni centrali. D'altra parte, tra il 2006 e il 2007, il CISIS<sup>3</sup>, ha promosso il progetto denominato ICAR (“Interoperabilità e Cooperazione Applicativa tra le Regioni”), il quale promette di fornire un ulteriore stimolo alla realizzazione di

---

1 Il Gruppo di Sviluppo ICAR dell'Università della Basilicata è formato da Donatello Santoro, Michele Amodeo, Nunzia Laguardia, Rocco Piliero, Michele Santomauro, Rocco Sapienza

2 Centro Nazionale per l'Informatica nella Pubblica Amministrazione – <http://www.cnipa.gov.it>

3 Centro Interregionale per i Sistemi Informatici, geografici e Statistici – <http://www.cisis.it>

iniziative di cooperazione.

Gli obiettivi del progetto ICAR sono principalmente due. Per cominciare, il progetto mira a valorizzare il ruolo delle reti regionali nell'ambito dell'infrastruttura nazionale, di fatto rappresentando lo spunto per l'integrazione nel sistema delle pubbliche amministrazioni locali. Inoltre, ICAR si propone di sviluppare un'implementazione di riferimento completa e di carattere open source per i componenti dello standard SPCoop.

Questo articolo ha come scopo descrivere queste iniziative, ed in particolare: (a) introdurre i concetti tecnici alla base dell'architettura SPCoop; (b) descrivere le iniziative collegate ad ICAR e le principali innovazioni introdotte dal progetto; (c) presentare due diverse soluzioni open source per la realizzazione dell'infrastruttura. Una di queste, il progetto *freESBee*, è attualmente in corso di sviluppo presso l'Università degli Studi della Basilicata.

## 2 L'Architettura SPCoop

SPCoop propone un'architettura di tipo SOA per la pubblica amministrazione in cui le applicazioni possono comunicare esponendo una serie di servizi o consumando servizi erogati da altre applicazioni.

Di fatto, SPCoop può essere definito come l'*Enterprise Service Bus (ESB)* della pubblica amministrazione italiana. Recentemente il termine Enterprise Service Bus è diventato uno dei più usati nell'ambito del marketing collegato alla fornitura di soluzioni di tipo SOA. Come spesso succede in questi casi, non c'è un completo consenso sul reale significato tecnico di questa nozione [5], dal momento che viene spesso utilizzata con significati completamente diversi. E' possibile però schematizzare dicendo che un Enterprise Service Bus è un'infrastruttura di comunicazione orientata ai servizi di un'organizzazione complessa, che fornisce tipicamente almeno le seguenti funzionalità:

- standardizzazione del formato dei messaggi scambiati tra le applicazioni, supportando eventualmente la trasformazione dei messaggi nativi per adeguarli al formato standard;
- accesso trasparente ai servizi disponibili sul bus, con funzionalità di routing delle richieste, ed eventualmente, astrazione rispetto ai protocolli supportati da fruitori ed erogatori di servizi;
- funzionalità avanzate di middleware, come supporto alla sicurezza, qualità del servizio, orchestrazione e coreografia.

SPCoop standardizza la maggior parte di queste caratteristiche attraverso le nozioni di *porta di dominio*, *busta e-gov* e *accordo di servizio*.

### La Porta di Dominio

In un classico scenario di cooperazione applicativa proposto da SPCoop, due Sistemi Informativi Locali (SIL) appartenenti a reti private diverse, possono comunicare tra di loro instaurando una comunicazione punto-punto tra le

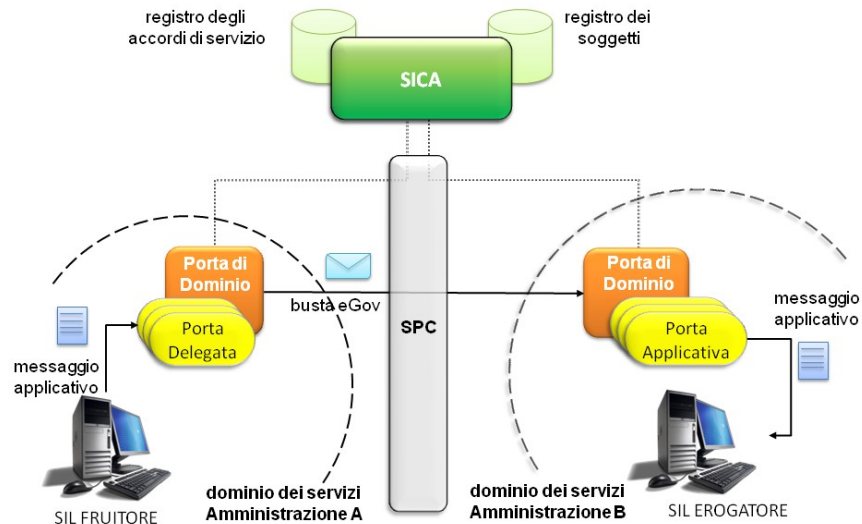


Figura 1: Architettura SPCoop

proprie *porte di dominio*. La porta di dominio [4] rappresenta il gateway attraverso cui ogni amministrazione offre e consuma servizi. E' possibile pensare che sia il punto di ingresso e di uscita unico per il dominio dell'amministrazione, come illustrato in Figura 1.

### La Busta e-gov

Per svolgere i propri compiti, la porta di dominio ha una duplice interfaccia: una interna per comunicare con i servizi applicativi ed una esterna per scambiare messaggi con altre porte di dominio. I messaggi che la porta di dominio scambia con i SIL del proprio dominio sono messaggi applicativi e possono essere basati su protocolli e standard diversi – lo standard SPCoop non vincola questo aspetto, lasciando libertà nello stabilire le interfacce di comunicazione e i protocolli.

Viceversa, i messaggi che una porta di dominio scambia con altre porte di dominio devono rispettare un formato ed un protocollo standard. Questi messaggi, detti *messaggi SPCoop*, devono essere messaggi basati sul formato SOAP 1.1 with attachments, utilizzare il protocollo HTTPS, e prevedere una serie di intestazioni standard che vanno sotto il nome di *busta di e-gov*.

### I Servizi di Registro

Per consentire la corretta identificazione dei servizi, dei soggetti e delle relative porte di dominio, l'architettura prevede che siano disponibili opportuni *Servizi Infrastrutturali per la Cooperazione Applicativa (SICA)*. Il compito principale dei servizi infrastrutturali, che sono centralizzati e indipendenti dalle singole

amministrazioni, è mantenere opportuni registri che vengono interrogati dalle porte di dominio per l'elaborazione dei messaggi applicativi. In particolare, i servizi SICA contengono:

- un registro dei soggetti abilitati alla comunicazione nell'ambito del sistema;
- un registro degli *accordi di servizio* che prescrivono le modalità e i requisiti per lo svolgimento corretto della cooperazione applicativa.

In sintesi, un accordo di servizio è un documento – in formato XML – che regola l'erogazione di un servizio applicativo da parte di un sistema locale ai fruitori che vogliono accedervi, descrivendo in dettaglio:

- i soggetti coinvolti (l'erogatore e i fruitori autorizzati ad accedere ai servizi);
- i servizi che sono oggetto dell'accordo, con le relative interfacce (sostanzialmente, frammenti di codice WSDL)
- i profili di collaborazione secondo cui i servizi vengono erogati (sincrono, asincrono, one-way);
- le politiche di sicurezza e la qualità del servizio prescritta.

E' previsto che le porte di dominio contattino il registro centrale per ottenere informazioni sui soggetti e sugli accordi di servizio.

### **Il Percorso di un Messaggio SPCoop**

Possiamo a questo punto descrivere sinteticamente il processo secondo cui un SIL appartenente ad un dominio – per esempio quello del Ministero dell'Università – invia un messaggio con una richiesta di erogazione di servizio ad un SIL che appartiene ad un dominio distinto – ad esempio il Ministero dell'Interno.

- per cominciare, il SIL fruitore contatta la propria porta di dominio, chiedendo di inoltrare al sistema erogatore del servizio un messaggio applicativo; il componente della porta di dominio demandato a ricevere messaggi applicativi di richiesta da parte dei SIL si chiama *porta delegata*;
- la porta delegata effettua il processo di *imbustamento* del messaggio applicativo ricevuto, per trasformarlo in una busta e-gov ed instradarla sul bus; in particolare, reperisce l'indirizzo della porta del dominio a cui appartiene il SIL erogatore del servizio richiesto, e inoltra la busta;
- il componente della porta di dominio che riceve messaggi e-gov di richiesta si chiama *porta applicativa*; ha il compito di effettuare lo *sbustamento* della richiesta ricevuta, ed inoltrarla al SIL erogatore;

E' semplice immaginare come il messaggio di risposta – che può essere inviato secondo un profilo di collaborazione sincrono o asincrono – segua un percorso

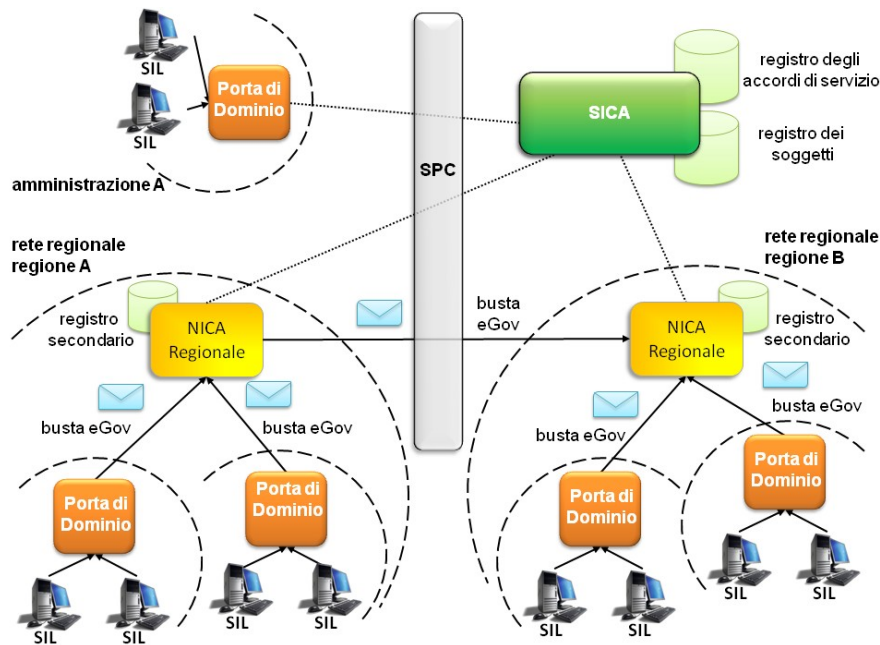


Figura 2: Architettura ICAR

inverso ma simile.

### 3 Il Progetto ICAR

ICAR<sup>4</sup> è un progetto nazionale diretto dal CISIS che coinvolge 16 regioni ed una provincia autonoma. L'obiettivo è la creazione di un'infrastruttura concreta di cooperazione tra le amministrazioni secondo le specifiche fornite dal CNIPA.

Il progetto ICAR è composto da tre task infrastrutturali e da sette task applicativi, strettamente connessi tra loro. I task infrastrutturali definiscono l'infrastruttura di base su cui si sviluppano i vari casi di studio definiti dai task applicativi. Gli obiettivi dichiarati sono i seguenti:

- valorizzare il ruolo delle reti regionali nell'ambito dell'infrastruttura nazionale di cooperazione applicativa, definendo modalità standard per l'integrazione nell'architettura SPCoop delle pubbliche amministrazioni locali;
- sviluppare un'implementazione di riferimento completa e di carattere open source per i componenti dello standard SPCoop, anche allo scopo di approfondire gli aspetti della specifica che non sono completamente

4 <http://www.progettoicar.it>

definiti, esplorando e confrontando varie possibili soluzioni;

- avviare una sperimentazione completa nell'ambito di scenari applicativi concreti che includono la compensazione sanitaria, la mobilità anagrafica, le tasse automobilistiche, definendo tra l'altro schemi di accordi di servizio da utilizzare per la cooperazione in questi domini.

I task infrastrutturali hanno il compito di riconsiderare la specifica SPCoop alla luce dell'integrazione delle reti regionali. Da questo punto di vista, l'introduzione di numerosi enti locali giustifica un ripensamento delle politiche di instradamento dei messaggi sulla rete.

Nello scenario di cooperazione tra la pubblica amministrazione centrale è ragionevole che ciascuna amministrazione costituisca uno o più domini di cooperazione e comunichi in modalità punto-punto con i domini delle altre amministrazioni, condividendo esclusivamente i servizi di registro centrali. Viceversa, pare opportuno, nel caso di cooperazione applicativa tra enti locali, introdurre un gateway regionale per l'accesso all'infrastruttura SPCoop, denominato NICA.

Il NICA (Nodo di Interconnessione per la Cooperazione Applicativa) è un nodo applicativo di servizio per le porte di dominio di una rete regionale, con le seguenti caratteristiche:

- funge da punto di diramazione per tutti i messaggi originati nella rete regionale e per tutti i messaggi destinati alla rete regionale, offrendo servizi di instradamento trasparente delle buste all'interno di SPCoop; in quest'ottica, è possibile pensare che gli enti locali siano integrati nell'infrastruttura attraverso una dorsale di NICA regionali che si occupano di coordinarne ed instradare il traffico, evitando la necessità di stabilire comunicazioni punto-punto tra i numerosi nodi del sistema;
- offre servizi di registro secondari per i nodi della regione, sincronizzando opportunamente i contenuti del registro con il registro centrale SPCoop;
- rappresenta il naturale punto di monitoraggio della qualità del servizio per la rete regionale, oltre che delle politiche di sicurezza;
- offre servizi applicativi a valore aggiunto, come un registro degli eventi per le applicazioni di tipo publish and subscribe.

L'architettura proposta da ICAR è rappresentata in Figura 2.

#### **4 Le Implementazioni Esistenti**

Nel corso degli anni si sono sviluppate varie implementazioni della specifica SPCoop. Alcune di queste – come la porta di dominio della Oracle e quella della Microsoft – sono di carattere commerciale e basate su licenze di tipo closed-source.

Esistono vari progetti open-source sviluppati nel corso degli anni (ad

esempio OpenPDD<sup>5</sup>), ma non sempre aggiornati rispetto alle più recenti versioni dello standard. Tra questi, è particolarmente rilevante il progetto OpenSPCoop<sup>6</sup> [1][2]. OpenSPCoop nasce nel 2004 come tesi di laurea in informatica presso l'Università di Pisa, e, diventato progetto OpenSource per iniziativa della società Link.it, presto attira attorno a sé una comunità di sviluppo ampia ed attiva. La robustezza del progetto è confermata dalla sua adozione da parte della Regione Toscana nell'ambito dell'infrastruttura CART (Cooperazione Applicativa nella Regione Toscana)<sup>7</sup>, attualmente dispiegata su tutto il territorio della regione.

In sintesi, OpenSPCoop è una implementazione completa delle specifiche SPCoop, sia per quanto riguarda le funzionalità della porta di dominio, che per quanto riguarda il NICA e i servizi di registro. Dalla versione 1.0 fornisce anche un'applicazione per la gestione di un registro di eventi e l'offerta di soluzioni di tipo publish&subscribe.

Gli autori descrivono OpenSPCoop come un *Web Service mediator* [1], ovvero come un sistema orientato alla trasformazione di messaggi SOAP. Dal punto di vista tecnico, si tratta di sistema di code JMS per la gestione e la trasformazione dei messaggi, che richiede per essere eseguito un contenitore per applicazioni J2EE di fascia alta. Le istruzioni sul sito del progetto fanno riferimento a JBoss.

## 5 Il Progetto freESBee

Nonostante OpenSPCoop rappresenti di fatto una soluzione di riferimento più che robusta per il dispiegamento dell'architettura SPCoop, ci sono vari elementi che suggeriscono l'opportunità di una implementazione alternativa.

Per cominciare, tutte le implementazioni attualmente disponibili dello standard rappresentano sostanzialmente soluzioni ad hoc. Al di fuori delle specifiche funzionali prodotte dal CNIPA – essenzialmente un elenco di casi d'uso – non esiste, attualmente, una concettualizzazione condivisa dell'architettura di una porta di dominio SPCoop.

Peraltro, le specifiche SPCoop – pur molto dettagliate da alcuni punti di vista – lasciano viceversa altri aspetti all'interpretazione dello sviluppatore. Per esempio, non sono standardizzate le interfacce dei servizi che le porte di dominio devono utilizzare per interrogare il SICA, né quelle da utilizzare per la sincronizzazione tra registro primario e secondari. Da questo punto di vista, poter contare su più di una implementazione open source è un requisito importante per verificare la concreta interoperabilità tra le soluzioni adottate, e mettere in luce gli aspetti della specifica che richiedono un maggior livello di approfondimento.

Da queste considerazioni nasce il progetto *freESBee*,<sup>8</sup> condotto

---

5 <http://www.openpdd.org>

6 <http://www.openspcoop.org>

7 <http://www.cart.rete.toscana.it>

8 <http://icar.unibas.it>

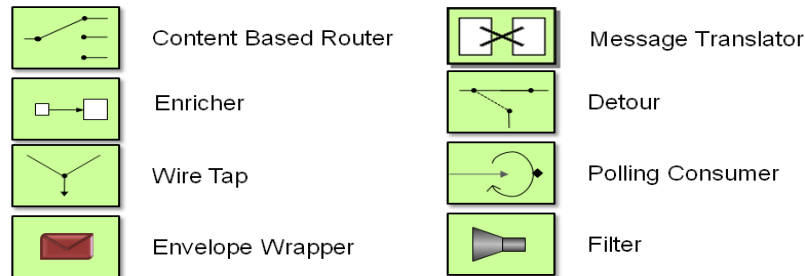


Figura 3: Enterprise Integration Patterns

dall'Università degli Studi della Basilicata in collaborazione con l'Ufficio Società Informazione della Regione Basilicata. freESBee è una implementazione open source della specifica SPCoop e ICAR pensata per essere alternativa ma interoperabile con OpenSPCoop. Il punto di partenza nello sviluppo di freESBee è stato la rielaborazione dell'architettura della porta di dominio utilizzando un modello concettuale diffuso nei sistemi di messaggistica: gli *Enterprise Integration Patterns*.

## 5.1 Enterprise Integration Patterns

Gli *Enterprise Integration Patterns (EIP)* [3] sono un'applicazione del concetto noto di design pattern ai sistemi di messaggistica. In sintesi, consentono di descrivere le elaborazioni e i percorsi subiti da un messaggio in termini di componenti elementari come “canale”, “endpoint”, “router”, “message enricher” ecc. Alcuni dei principali pattern sono illustrati in Figura 3 con la loro notazione grafica. I nomi sono per lo più sufficientemente esplicativi della relativa funzione; un elenco completo di descrizioni è disponibile sul sito [www.enterpriseintegrationpatterns.com](http://www.enterpriseintegrationpatterns.com).

Nella letteratura sui pattern, gli EIP rappresentano un caso particolarmente fortunato. E' noto infatti che molti sistemi di pattern hanno l'ambizione di costituire un “linguaggio completo” (*pattern language*) che consenta di descrivere tutta l'architettura di un sistema complesso esclusivamente in termini di pattern. E' altresì noto che, con riferimento a sistemi software generici, questa ambizione viene considerata velleitaria. Viceversa, nei sistemi di messaggistica – vista la dinamica relativamente semplice delle interazioni, che consistono nel trasferimento di messaggi da un endpoint ad un altro – è quasi sempre possibile descrivere l'intero sistema utilizzando esclusivamente pattern di tipo EIP. In quest'ottica, i pattern portano un notevole valore aggiunto al ciclo di sviluppo, in quanto:

- consentono di concettualizzare in modo elegante l'organizzazione del sistema, e di conseguenza di documentarla e comunicarla in modo molto efficace;



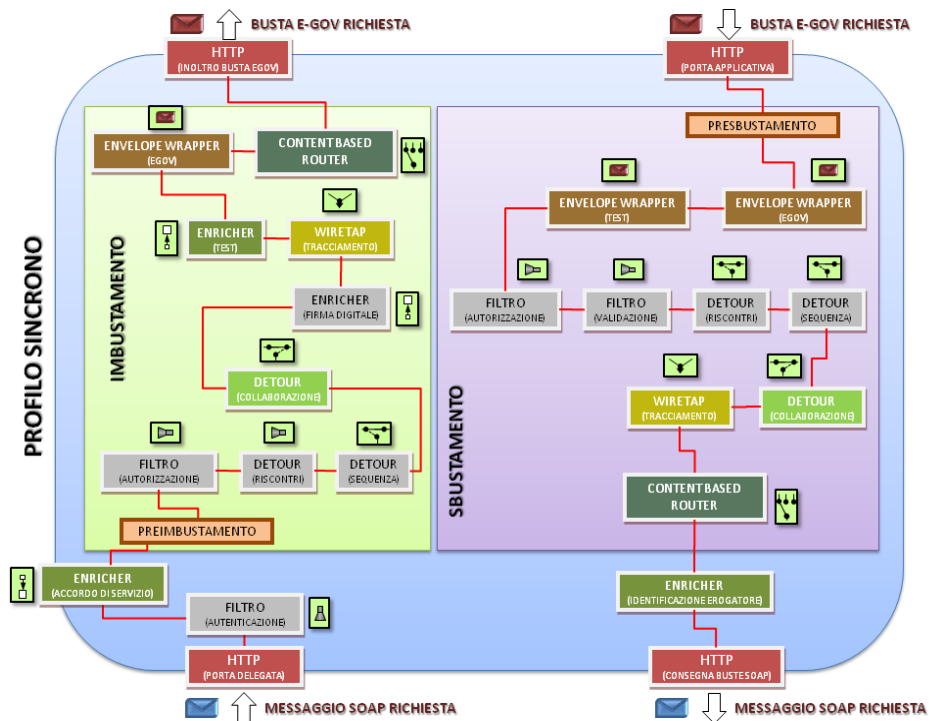


Figura 4: schema concettuale in termini di pattern per un profilo di collaborazione sincrono

- suggeriscono immediatamente un'architettura altamente modulare per l'implementazione e per il test del sistema, costruita fornendo implementazioni opportune per i singoli pattern utilizzati;
- facilitano notevolmente la manutenzione e l'evoluzione del sistema, consentendo facilmente di modificare i percorsi attraversati dai messaggi e le elaborazioni effettuate semplicemente modificando la composizione dei pattern.

E' di particolare interesse, in quest'ottica, il fatto che esistono numerose librerie open source per il supporto allo sviluppo di pattern EIP. Queste librerie da una parte forniscono componenti pronti e facilmente configurabili per implementare istanze dei principali pattern, dall'altra offrono un'infrastruttura pronta per comporre le istanze in modo da realizzare il sistema nel suo complesso. Di fatto, queste librerie supportano realmente un modello di sviluppo orientato all'utilizzo di librerie di componenti, utilizzato in altre discipline dell'ingegneria, come l'ingegneria elettronica, e spesso invocato nella letteratura sull'ingegneria del software, ma mai completamente realizzato.

## 5.2 Architettura di freESBee

In Figura 4 è illustrata l'architettura di freESBee in termini di EIP, con particolare riferimento ad un profilo di collaborazione sincrono. Nello scenario rappresentato distinguiamo le funzionalità della porta di dominio che invia il messaggio, a sinistra, da quella che riceve il messaggio, raffigurata a destra.

La porta delegata riceve dal SIL un messaggio che viene intercettato da un componente di tipo *interceptor* che si occupa di filtrare il messaggio e verificare le autorizzazioni. Se il messaggio SOAP possiede le autorizzazioni necessarie, il messaggio viene arricchito da un componente di tipo *enricher*, con una serie di informazioni dipendenti dall'accordo di servizio.

A questo punto il messaggio è pronto per essere trasformato in una busta e-gov, conforme alle specifiche SPCoop, dal modulo di imbustamento. Nello specifico, il modulo di imbustamento, prima di creare la busta e-gov, esegue le operazioni elementari relative al tracciamento e alla correlazione del messaggio.

Dal lato erogatore, invece, la porta applicativa riceve il messaggio in forma di busta e-gov che, attraversando il modulo di sbustamento, subisce una trasformazione inversa rispetto a quella vista nella fase di imbustamento. A questo punto il messaggio può essere consegnato al servizio corrispondente che si preoccuperà di analizzare la richiesta ed elaborare la risposta.

Sul sito <http://icar.unibas.it> sono disponibili tutti i diagrammi dei pattern relativi a freESBee, con la descrizione dei singoli componenti.

Sulla base di questa architettura, freESBee è stato implementato utilizzando la libreria Apache Camel<sup>9</sup>, una delle principali implementazioni degli EIP. E' possibile configurare freESBee non solo come porta di dominio, ma anche come NICA. In questo caso freESBee sarà in grado di effettuare le operazioni di routing dei messaggi ricevuti dalle porte di dominio sottostanti. Le tabelle di routing dei vari NICA vengono mantenute aggiornate tramite procedure di sincronizzazione automatiche.

Per configurare freESBee è disponibile un'applicazione web (freESBeeWeb) che consente di gestire le principali funzionalità come ad esempio la gestione di soggetti, delle porte applicative e delle porte delegate. Nel rispetto della filosofia che sta alla base dell'architettura di freESBee basata sullo scambio di messaggi, freESBee espone una serie di web services per la sua configurazione dall'esterno che vengono invocati da freESBeeWeb. Lo scambio di questi messaggi avviene in maniera sicura rispettando le specifiche di WS-Security che rappresenta lo standard della sicurezza nel contesto dei web services.

Nella Tabella 5 è riportato un confronto tra la versione 0.5beta di freESBee e la versione 1.0 di OpenSPCoop. E' opportuno notare che il confronto è in qualche maniera improprio. OpenSPCoop è un progetto di notevole maturità (nel 2008 raggiunge il suo quarto anno di vita), e vanta una base ampia di

<sup>9</sup> <http://activemq.apache.org/camel/>

	OpenSPCoop 1.0	freESBee 0.5beta
Piattaforma Tecnologica	Apache Axis + JMS	Apache Camel
Server Applicativo	Server J2EE di fascia alta (istruzioni per JBoss)	Qualsiasi server applicativo J2EE (inclusi Tomcat e Jetty)
Protocolli supportati	SOAP + Integration manager	tutti i protocolli supportati da Camel (SOAP, REST, HTTP, HTTPS, JMS, JDBC, FILE, RMI, Corba ecc.)
Linee di codice (stub esclusi)	92.157	7.163

*Tabella 5: Confronto tra OpenSPCoop e freESBee*

installazioni e di sviluppatori. freESBee è un progetto molto più giovane e ancora nelle fasi iniziali. Peraltro, al di là dell'implementazione dei requisiti funzionali della porta del dominio, OpenSPCoop 1.0 offre un numero maggiore di funzionalità aggiuntive rispetto alla versione attuale di freESBee; ad esempio, OpenSPCoop implementa varie interfacce per l'accesso al registro dei servizi; oltre ad una implementazione relazionale, come richiesto nella specifica viene fornita un'interfaccia UDDI. Viceversa, freESBee allo stato attuale prevede esclusivamente una implementazione relazionale.

Nonostante questo fatto, crediamo che il confronto possa essere utile a mettere in luce alcuni contributi che freESBee può portare alla comunità di sviluppo SPCoop. Questi contributi sono essenzialmente legati all'approccio basato sui pattern e alla scelta della piattaforma tecnologica adottata.

Per cominciare, queste scelte hanno consentito di snellire significativamente i requisiti per l'installazione dell'applicazione: freESBee può essere installato all'interno di qualsiasi contenitore, anche di fascia bassa, come Tomcat.

Allo stesso tempo, Camel garantisce il supporto per una grande varietà di protocolli per lo scambio di messaggi e, grazie alla sua natura modulare, aggiungere il supporto per nuovi protocolli risulta essere di facile implementazione. Questo dovrebbe consentire di semplificare la comunicazione tra i sistemi informativi locali e la porta di dominio. Viceversa, OpenSPCoop si propone essenzialmente come un sistema di mediazione tra Web services, e quindi è essenzialmente orientato a SOAP. Tra i vari moduli è previsto un modulo chiamato Integration Manager, essenzialmente una libreria per semplificare l'invio dei messaggi da parte del SIL.

Più in generale, l'approccio a componenti adottato per lo sviluppo di freESBee ha consentito di realizzare le funzionalità relative alla porta di dominio con un numero di linee di codice decisamente limitato, favorendo quindi i tempi di sviluppo e soprattutto la manutenzione.

L'aspetto più interessante dello sviluppo di freESBee, però, resta a nostro avviso l'opportunità di verificare l'interoperabilità tra le due soluzioni. I test di

interoperabilità tra le due piattaforme condotti fino a questo punto hanno messo in luce alcuni aspetti significativi della specifica SPCoop che richiedono maggiore approfondimento. Tra questi, il problema della correlazione dei messaggi nei profili di collaborazione asincroni, e la comunicazione con i servizi di registro. E' nostra convinzione che solo con due implementazioni indipendenti e realmente interoperabili tra di loro sarà possibile individuare ed eliminare le ambiguità e le debolezze della specifica SPCoop.

## 6 Opportunità di Ricerca

La disponibilità di componenti open source per la realizzazione dell'infrastruttura è ovviamente un notevole passo avanti verso il dispiegamento concreto del sistema pubblico di cooperazione applicativa italiano. D'altro canto, questo consente esclusivamente di realizzare il service bus, ma lascia aperto il problema dell'integrazione dei servizi applicativi locali.

E' noto nella letteratura sui sistemi SOA [7] che la realizzazione del service bus dovrebbe essere il punto di arrivo di una strategia orientata ai servizi, e non il punto di partenza. Il passo decisamente più delicato resta infatti quello di ripensare i processi organizzativi e riorganizzare i sistemi informativi esistenti per orientarli ai servizi.

Alla luce di queste considerazioni, è opportuno porsi due problemi:

- una volta che l'infrastruttura SPCoop sia stata realizzata e dispiegata, anche solo in parte, come fare per sperimentarla e sottoporla ad una seria verifica di qualità senza dover attendere i tempi (presumibilmente lunghi) necessari per adeguare i sistemi informativi locali ?
- come fare per accelerare il processo di adeguamento all'infrastruttura di sistemi informativi locali di tipo "legacy", e quindi poco aperti e per nulla predisposti all'interazione basata su servizi ?

Questi due aspetti rappresentano a nostro avviso interessanti opportunità di ricerca e promettenti direzioni di studio futuro.

## Ringraziamenti

Salvatore Raunich è parzialmente finanziato dal Fondo Sociale Europeo.

## Bibliografia

- [1] A. Corradini & T. Flagella. OpenSPCoop: un Progetto Open Source per la Cooperazione Applicativa nella Pubblica Amministrazione. In *Atti del Convegno Italiano AICA*. 2007.
- [2] A. Corradini, T. Flagella & A. Poli. Aspetti di Interoperabilità della Specifica SPCoop nell'Implementazione OpenSPCoop 1.0. In *Atti del Congresso su Pubblica Amministrazione Aperta e Libera (PAAL)*. 2007.
- [3] B. Hohpe & G. Woolf. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley, 2004.

- [4] Wikipedia - Porta di Dominio. [http://it.wikipedia.org/wiki/Porta\\_di\\_dominio](http://it.wikipedia.org/wiki/Porta_di_dominio).
- [5] M. Richards. The Role of the Enterprise Service Bus. <http://www.infoq.com/presentations/Enterprise-Service-Bus>. 2008
- [6] Centro Nazionale per l'Informatica nella Pubblica Amministrazione. Requisiti e specifiche funzionali del SPCoop. [http://www.cnipa.gov.it/site/it-it/In\\_primo\\_piano/Sistema\\_Pubblico\\_di\\_Connettivita\\_\(SPC\)/Servizi\\_di\\_interoperabilita\\_evoluta\\_e\\_cooperazione\\_applicativa/](http://www.cnipa.gov.it/site/it-it/In_primo_piano/Sistema_Pubblico_di_Connettivita_(SPC)/Servizi_di_interoperabilita_evoluta_e_cooperazione_applicativa/). 2005
- [7] B. Woolf. ESB-oriented architecture: The wrong approach to adopting SOA. <http://www.ibm.com/developerworks/webservices/library/ws-soa-esbarch/>. 2007