Architettura

30 Ottobre 2003

Architettura

Versione 1.3

30 Ottobre 2003

A cura di: Fabio Bagatin, Alessandro Osnaghi

Sommario

Capitolo 1

Introduzione

Il progetto PEOPLE è costituito da un'aggregazione di amministrazioni locali che rappresentano altrettanti punti di riferimento nella definizione e gestione di sistemi informatici per l'accesso ai servizi territoriali da parte di cittadini e imprese. Queste stesse amministrazioni locali hanno già realizzato numerose applicazioni innovative per permettere ed allargare l'accesso *online* ai servizi di pubblica amministrazione, adottando modalità e tecnologie che consentono di fruire dei servizi medesimi in modo sempre più snello ed efficace.

Il progetto PEOPLE rappresenta lo sforzo da parte di queste amministrazioni di compiere un ulteriore sostanziale progresso nella realizzazione, dispiegamento ed offerta dei servizi *online* razionalizzando al tempo stesso la gestione dei sistemi informatici ed offrendo agli utenti finali modalità di interazione unificate e facilmente accessibili.

L'obiettivo più generale del progetto PEOPLE è quello di migliorare la disponibilità dei servizi *online* nella pubblica amministrazione attraverso la progressiva costruzione di un sistema federato che permetta significative economie di scala e tragga il massimo vantaggio dalle capacità esistenti e dalla storia recente delle varie realtà locali, promuovendo la condivisione aperta delle risorse e delle esperienze.

1.1 Obiettivi del documento

Questo documento ha lo scopo di descrivere gli elementi fondamentali di un'architettura per sistemi distribuiti e federati che consenta agli attori partecipanti al progetto PEOPLE (i comuni, ma anche i fornitori di soluzioni) di descrivere in termini condivisi i diversi componenti facenti parte del progetto, senza per altro entrate nello specifico delle tecnologie impegnate per la realizzazione.

Si rimanda ai documenti di progetto la definizione di come i diversi componenti dell'architettura sono implementati nelle specifiche realizzazioni.

Allo stesso modo si rimanda ad altri documenti la definizione degli obiettivi del progetto PEOPLE

Capitolo 2

Architettura di riferimento

2.1 Introduzione

L'architettura di riferimento per i domini partecipanti a PEOPLE non può non modellarsi su quella di un tipico sistema distribuito. Infatti, per quanto ogni membro della Comunità mantenga la propria autonomia amministrativa e la responsabilità della gestione dei servizi ai cittadini, al tempo stesso sempre di più il modello di interazione si avvicina a quello di un sistema distribuito in cui i diversi partecipanti offrono e fruiscono servizi in base alla proprie esigenze/ruoli.

Anche se a livello di e-Government il modello architetturale accettato è quello composto da sistemi autonomi (i domini per l'appunto) che interagiscono a livello applicativo con altri domini attraverso la coppia porta applicativa – porta delegata, è importante condividere un modello generale di architettura di sistema che permetta di "descrivere" in maniera più completa i componenti che organicamente compongono tali domini.

A tale scopo si propone un modello architetturale quale quello rappresentato in Figura 1 in cui si evidenziano tre "strati" fondamentali:

1) Uno strato di servizi di front-end rivolti ai cittadini, ad altre amministrazioni e alle imprese, "Front-end Service Layer"
2) Uno strato di servizi virtuali che implementa i cosiddetti "eventi della vita", "Virtual Service Layer"
3) Uno strato di servizi primitivi messi a disposizione dai sistemi informativi dei Comuni, da fornitori di servizi e da altre amministrazioni, "Back-end Service Layer"

A questi strati più tipicamente applicativi si affiancano infrastrutture, per loro natura trasversali, quali quelle necessarie alla

- Amministrazione
- Comunicazione
- Sicurezza

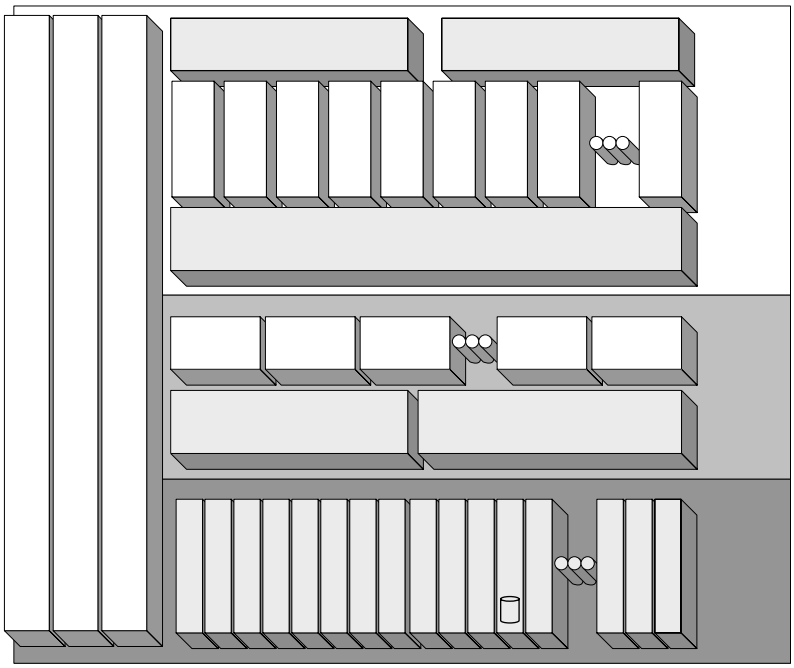


Figura 1 Architettura logica

Questa vista logica tende ad evidenziare i macrocomponenti strutturali che compongono il sistema distribuito di dominio, non implica in nessun modo che ciascun sistema partecipante a PEOPLE debba implementare in toto il modello architetturale.

Anzi è auspicabile che, con l'eccezione dei comuni di più ampie dimensioni, che potrebbero scegliere di realizzare istanze complete dei diversi strati, ogni membro della Comunità implementi tutti e soli gli strati e i componenti che meglio rispondono alle proprie esigenze, capacità, specializzazioni e obblighi di legge (in particolare per quanto riguarda aspetti quali la riservatezza delle informazioni).

Come viene infatti esemplificato nella Figura 2, è ipotizzabile che vengano creati dei domini specifici in cui vengano concentrati, per esempio, i servizi di Front End, attraverso la creazione di uno o più portali che mettano a fattor comune la gestione dei contenuti, i meccanismi di registrazione e autenticazione degli utenti e così via, come pure la creazione di servizi di Comunità che diano vita alla metafora degli eventi della vita comune, come pagare le tasse, abitare … oltre a servizi di back-end forniti da terze parti, quali quelli di billing.

Management Infrastructure

Communication Infrastructure

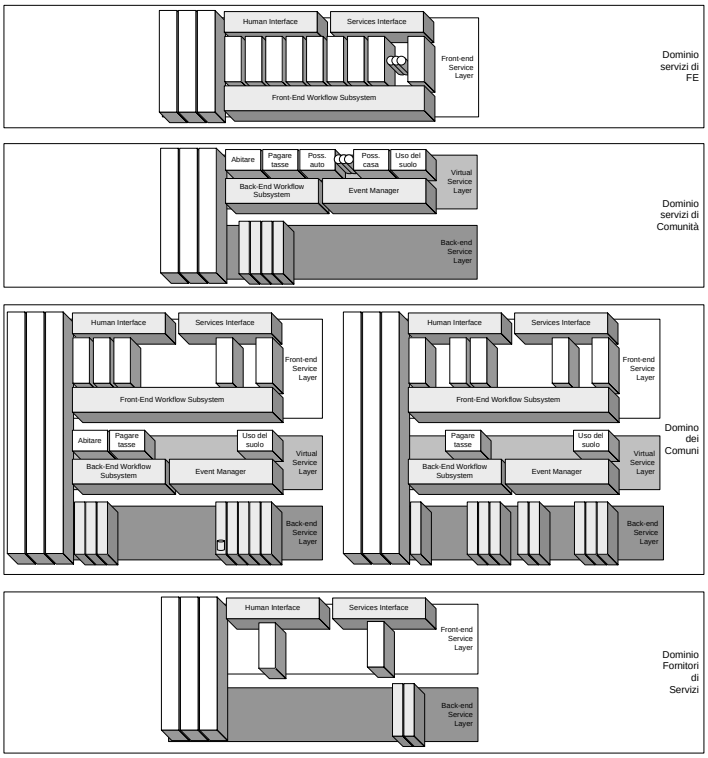


Figura 2 Ripartizione dei servizi

Nel seguito verranno analizzati singolarmente i diversi strati, evidenziandone le peculiarità sia tecnologiche che organizzative.

2.2 Il Front-end Service Layer

Il layer di Front-end, come si può notare dalla Figura 3, è quello che direttamente interagisce con le entità esterne al dominio. Sono ipotizzabili due tipi fondamentali di interazione:

- Un'interazione con "umani"
- Un'interazione con "sistemi informatici"

E' ovvio che l'intermediario tra una persona e PEOPLE sarà comunque e sempre un computer, che sia embedded in un telefono cellulare o un Personal Computer, quello che si intende evidenziare è che nel primo caso il sistema di Front-end dovrà mettere a disposizione delle interfacce utilizzabili da umani, siano esse pagine Web o applicazioni scaricabili su palmare o cellulare, nel secondo caso invece sarà il sistema all'altro capo della conversazione che dovrà preoccuparsi di come mediare l'interazione con i propri utenti umani, qualora ce ne fossero. Questo secondo caso è quello che più normalmente viene identificato come "cooperazione applicativa".

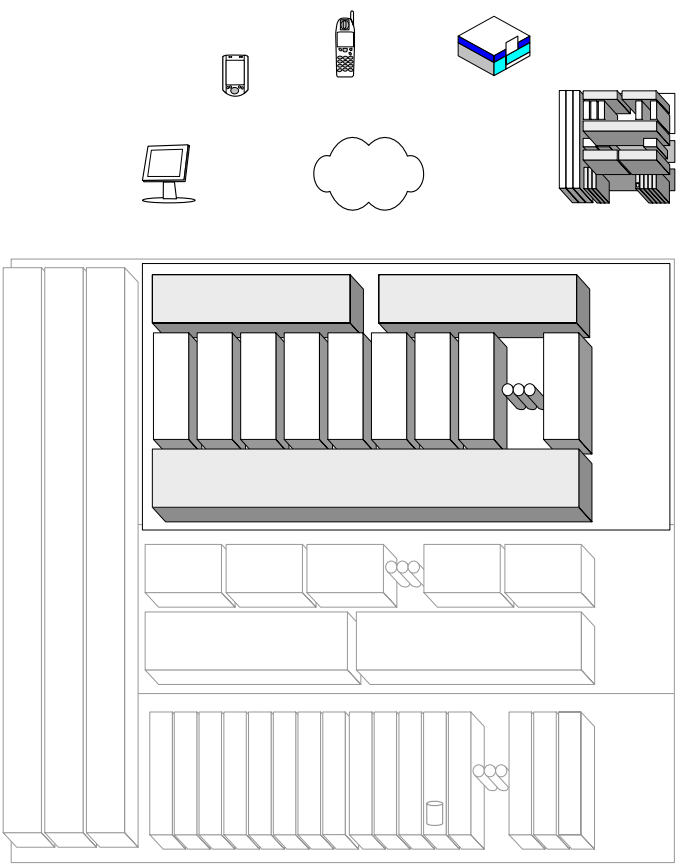


Figura 3 Front-end Layer

Questa duplicità nella modalità di accesso si rispecchia nei componenti di questo strato e in particolare in quelli denominati::

- Human Interface
- Services Interface

La Human Interface è quella che si prenderà cura di creare un'interfaccia fruibile da umani in modalità multicanale, mentre la Service Interface è quella che dovrà creare l'interfaccia dei servizi o, in altre parole, la Porta Applicativa di dominio.

In questo strato saranno presenti quindi una serie di servizi di Front-end, esposti sia in maniera diretta (implementati cioè nello strato stesso) o veicolati da altre entità in modalità di "integrazione lasca" (link) o "integrazione stretta" (attraverso portlet, vedi in seguito).

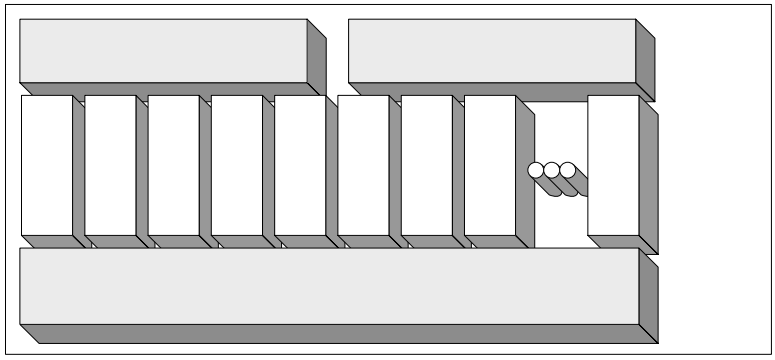


Figura 4 Servizi di Front-ent

La tipologia di servizi andrà da semplici funzionalità informative, quali quelle relative alle notizie, alle informazioni per i cittadini, alle informazioni locali e turistiche, a funzionalità di front-end verso attività di tipo transattive, quali il cambio di indirizzo, piuttosto che la gestione delle pratiche edilizie.

Il front-end non si assume la funzione di esecutore della transazione, ma funge da collettore delle informazioni e da punto di ingresso per il tracciamento dell'avanzamento delle richieste.

Data la complessità di alcuni dei processi coinvolti, questo strato ha la necessità di implementare un Workflow di tipo User-oriented, un workflow cioè che permetta di automatizzare la raccolta delle informazioni e che verifichi la corretta esecuzione dei diversi passi necessari sul lato dell'utente (non cioè sul lato del comune o dell'amministrazione).

Per fare un esempio, nel caso di richiesta di una concessione edilizia, il workflow di front-end si prenderà cura di verificare l'identità dell'intermediario sottomettente la richiesta, che questo sia iscritto e attivo nell'apposito albo di categoria, che la delega sia stata approvata dal destinatario il servizio (il cittadino per cui si richiede la concessione), che siano stati pagati i bolli eventualmente necessari e così via. Una volta fatte le verifiche il workflow potrà inoltrare la richiesta (sintatticamente corretta) verso gli strati sottostanti.

Si deve sottolineare che la complessità dei controlli compiuti dal workflow a questo livello è altamente variabile, dalla semplice verifica della correttezza del codice fiscale alla presenza di N autorizzazioni, funzionalità che non necessariamente richiedono l'acquisizione di un prodotto specifico per il workflow di Front-end. Spesso tali controlli e workflow sono insiti nell'implementazione dell'applicazione di front-end, sottoforma di codice Javascript lato client o di controlli formali lato server. Quello che qui si vuole sottolineare è che questo strato non è di pura informazione o di ridirezione verso altri siti, ma può a tutti gli effetti fungere da piattaforma comune di front-end per diverse entità, svolgendo tutto un insieme di funzionalità di verifica sintattica preliminare all'invocazione dell'attività transattiva.

Un'altra precisazione necessaria è quella relativa alla modalità di interazione con gli altri strati/domini. Come indicato i servizi di Front-end possono essere tanto servizi totalmente autonomi che utilizzanti servizi messi a disposizione da entità esterne allo strato. Queste entità possono esporre tali servizi in modalità di Front-end di tipo Human-interface, e in questo caso si ha una ridirezione verso il sito che fornisce tale funzionalità, in modalità di Front-end di tipo Services-interface, quando cioè il servizio a sua volta diventa un client di un altro servizio allo stesso livello di astrazione, o ancora in modalità di Virtual service e di Back-end service, esponendo in maniera pubblica (tramite cioè porta applicativa) tali servizi.

Cambio Indirizzo

2.2.1 Human-interface e Portlet

Si è accennato precedentemente al fatto che taluni servizi di Front-end di tipo Human-interface possono essere integrati in modalità Portlet. L'architettura a Portlet è una moderna modalità di creazione della interfaccia utente Web in cui si tenta di riutilizzare la metafora del desktop e delle finestre. In Figura 5 è possibile vedere un tipico ambiente a portlet.

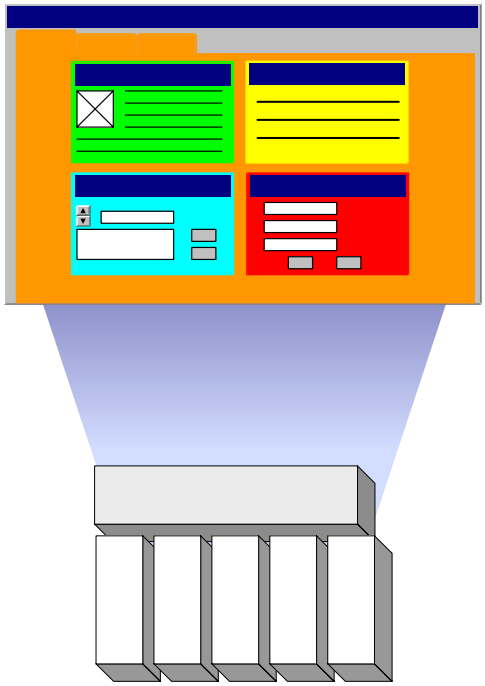
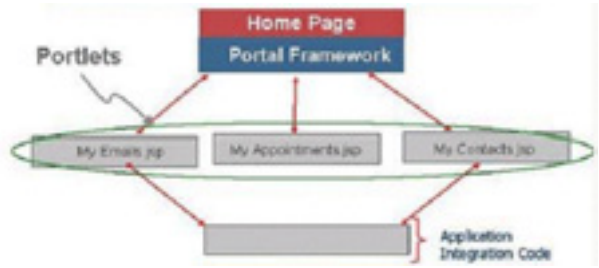


Figura 5 Tipico ambiente a portlet

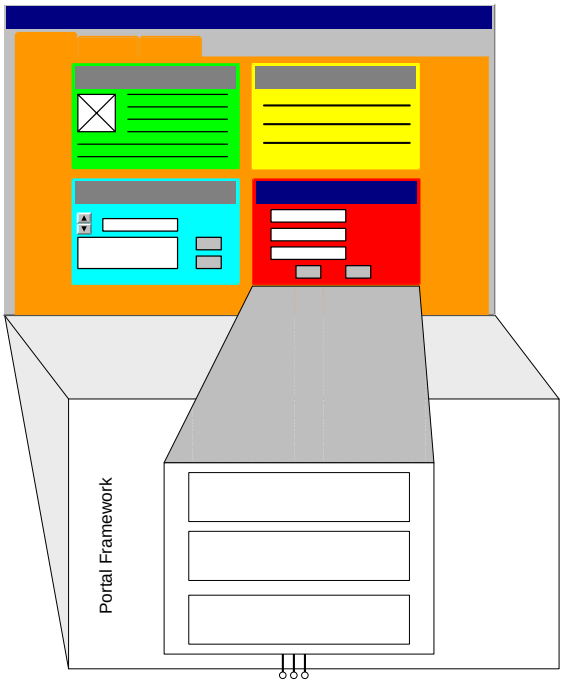
Come si può notare il browser Web è diventato un contenitore di finestre rettangolari che proprio come su un desktop possono avere tre stati: nomale, massimizzata (finestra intera), minimizzata.

Inoltre, attraverso un'interfaccia di personalizzazione è possibile decidere quali portlet si vuole aggiungere alla(e) pagina(e) disponibili.

Questa modalità di aggregazione della presentazione offre una grande flessibilità nella manutenzione dei portali, permettendo l'aggiunta di componenti applicativi senza bisogno di intervento specialistico per l'aggiornamento della grafica.



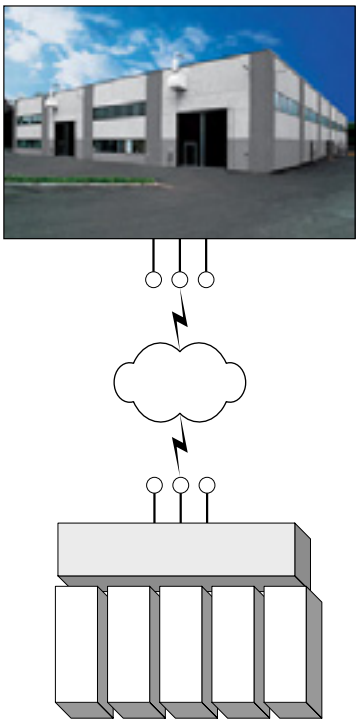
Indipendentemente dalla tecnologia utilizzata per l'implementazione l'architettura a portlet è ormai abbastanza comune: una pagina web funge da contenitore di uno o più portlet, i quali poi si prendono cura di implementare sia l'aspetto della presentazione grafica della propria area, che dell'accesso alle informazioni.



Per quanto riguarda il Front-end è possibile, in un'architettura a portlet, separare in maniera molto netta la componente di presentazione da quella di logica, trasformando la prima in un elemento di front-end, mentre la seconda di back-end. La prima è chiaramente dipendente dalla tecnologia utilizzata per l'implementazione del portale, la seconda può invece essere indipendente. Questa indipendenza è sottolineata dal fatto che ormai la maggior parte dei produttori di ambienti per portali offrono off-the-shelf portlet in grado di interagire in maniera semplice con Web Services, in cui la presentazione può essere facilmente personalizzata attraverso l'inclusione di un foglio di stile (CSS o XSLT).

2.2.2 Service Interface

Come precedentemente accennato il FSL dovrà essere in grado di offrire servizi non solo a "umani" ma anche ad altri comuni non facenti parte di PEOPLE, ad aziende, ad altre amministrazioni.



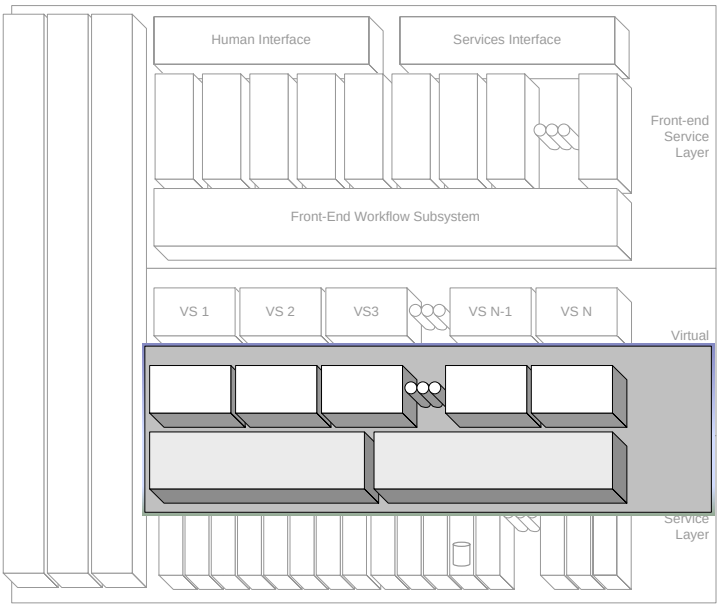
La Human-interface espone per sua natura un'interfaccia non adatta all'interazione tra sistemi, in quanto l'obiettivo è quello di rendere "facile" la navigazione all'utente tra le possibili opzioni, partizionando la complessità in diverse pagine anche a scapito dell'efficienza applicativa.

E' perciò importante che il FSL esponga, per un ben selezionato sott'insieme di funzionalità un'interfaccia di tipo server-to-server, nella modalità dei Web Services, con o meno Busta di Government in base alla validità legale o meno del procedimento.

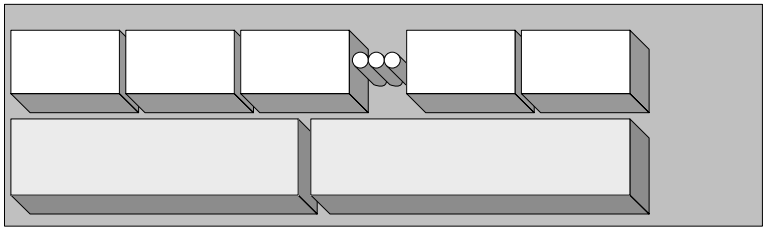
E' importante rilevare che la modalità di interazione Serve-to-Server a questo livello, per quanto a livello tecnico similare, è leggermente diversa da quella richiesta dagli altri strati (Virtual Service Layer e Back-end Service Layer) in quanto a questo livello l'entità interagente non partecipa al workflow degli eventi della vita, ma semplicemente "fa le veci" del suo corrispettivo umano, attivando uno degli eventi della vita e facendo perciò a tutti gli effetti da end-point applicativo.

2.3 Virtual Service Layer

Il Virtual Service Layer ha lo scopo creare un sistema dei servizi virtuale, in cui gli eventi significativi dell'interazione del cittadino con in comuni, secondo la logica degli "Eventi della Vita" sono mappati su servizi generici e indipendenti dalla topologia e peculiarità delle applicazioni di back-end.



Questa indipendenza permette in particolare di gestire interazioni, quali quelle relative al servizio "Abitare", che possono coinvolgere più comuni o amministrazioni diverse, per questa ragione questo strato è tipico del Dominio dei Servizi di Comunità.



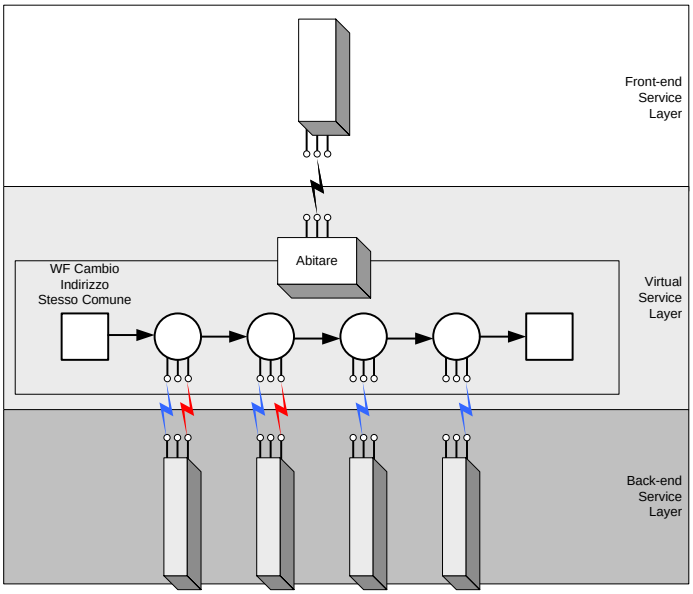
A supporto dei servizi offerti da questo strato sono fondamentali due componenti:

- Il Back-End Workflow Subsytem
- L'Event Manager

Al contrario di quanto avviene nel front-end, dove le interazioni sono normalmente di tipo sincrono con l'utente, i procedimenti attivati dai servizi di questo strato sono prevalentemente di tipo asincrono e di lunga durata (long-lived).

In genere possono essere innescati da una richiesta proveniente dallo strato superiore, o da un evento proviene da un altro servizio o dai sistemi di back-end. Questi eventi danno inizio a, potenzialmente, una serie di passi (step) che devono essere registrati e monitorati, per dare origine a report sull'avanzamento della pratica o, in alcuni casi, a escalation in caso di problemi. *I diversi servizi virtuali devono quindi poter fare affidamento su un Workflow di Back-end affidabile e non demandato all'implementazione del singolo servizio, questo per poter più facilmente, e in maniera dichiarativa,*

gestire l'associazione tra strutture e responsabilità, in modo da poter utilizzare modalità di indirizzamento (routing) dei task di tipo Role Based, Rule Based e Relationship Based.



Allo stesso modo deve essere affidabile e flessibile l'Event Manager, il quale viene innescato dal Workflow manager ogni qual volta questo ha bisogno di interagire con altri servizi (Virtuali o di Back-end), e che a sua volta innesca il Workflow al ricevimento di eventi asincroni provenienti dai servizi stessi. Le modalità di decisione su quali servizi innescare può essere determinata dall'Event Manager in base a più meccanismi di routing delle richieste:

- Sorgente-Destinatario
- Publish & Subscribe
- Content Based Routing

Una delle funzionalità più complesse di questo strato è la gestione della trasazionalità delle operazioni. Non potendo fare affidamento su meccanismi transazionali di tipo 2PC (2 Phase Commit), questo strato si dovrà preoccupare di concordare meccanismi di "compensazione", in modo da poter richiedere l'annullamento di un'operazione richiesta qualora non vada a buon fine con tutti i partecipanti. Questo onde evitare, per esempio, che una persona che ha richiesto un cambio di residenza, venga cancellata dall'anagrafe del comune in cui correntemente risiede e non venga inserita nel comune di destinazione.

Come si è accennato è fondamentale che a questo livello venga mantenuto un log storico completo, e facilmente interrogabile, dello stato delle richieste provenienti dagli altri strati, del loro stato attuale e di eventuali anomalie.

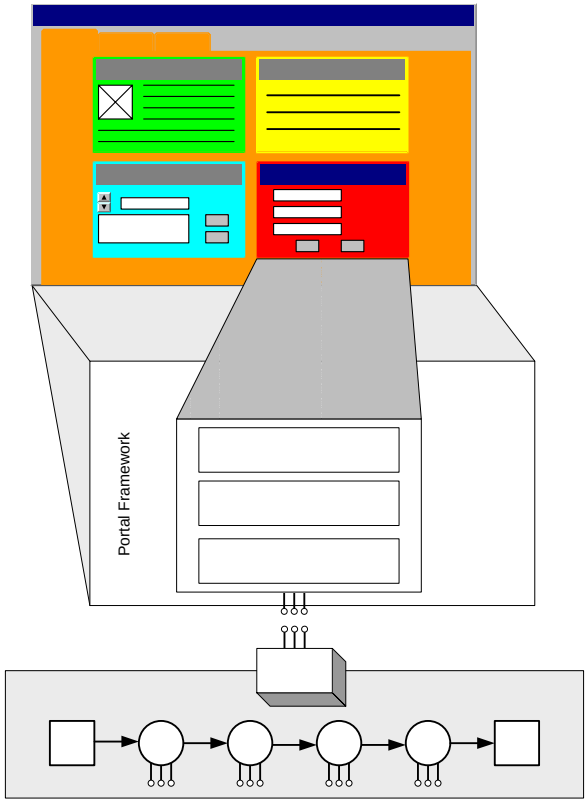
Al contrario, per questioni di riservatezza delle informazioni, questo strato dovrà mantenere informazioni sui cittadini solo per le funzionalità e il tempo necessari all'esecuzione della richiesta.

L'interazione dello strato Virtual Service Layer con gli altri strati è tipicamente di tipo machine-to-machine, le eventuali interazioni con gli utenti umani dovrebbero essere fatte galleggiare a livello dello strato di Front-end.

2.3.1 Rapporto tra Front-end Service Layer e Virtual Service Layer

Esiste uno stretto rapporto tra il Virtual Service Layer e i portlet di presentazione presenti nel Front-end Service Layer. Mentre alcuni di questi ultimi sono totalmente di tipo informativo, e,

come abbiamo già indicato, danno semplicemente accesso al notizie, informazioni per i cittadini, informazioni locali e turistiche, gestite tramite strumenti editoriali, e in linea di massima presenti nello stesso strato, altri saranno a tutti gli effetti “le finestre” di accesso ai servizi esposti dal VSL.



E’ anzi ipotizzabile che per ogni servizio presente nel VSL possano esistere diversi portlet che presentano le diverse sfaccettature del servizio (informativo: la natura del servizio e le diverse possibilità di richiesta, i prerequisiti, ecc. transattivi: sottomissione della richiesta, verifica dell’avanzamento, accesso ai risultati, ecc.).

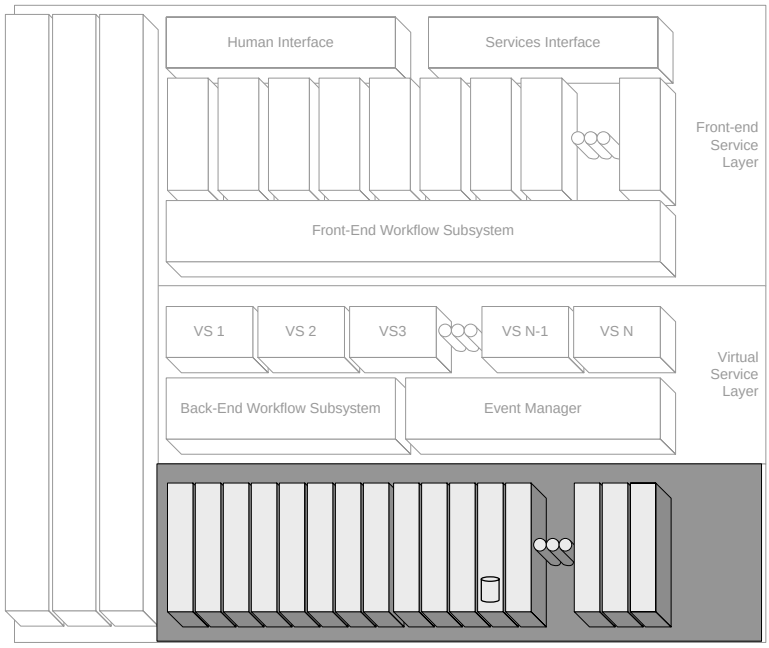
Dato che il FSL non memorizza localmente dati utente, è responsabilità del VSL gestire eventuali ottimizzazioni relative al caching dei suddetti dati, al recupero dello stato dei procedimenti dell’utente e così via.

Anche se, come si è detto, in alcuni casi i diversi strati saranno implementati integralmente nella stessa infrastruttura, è opportuno che i portlet assumano di non essere locali al VSL e utilizzino comunque le funzionalità di Porta Applicativa per interfacciarsi al VSL. Questo permetterà non solo di svincolare ulteriormente la scelta di tecnologie fatte per il FSL da quelle del VSL, ma di permettere un’evoluzione del front-end su dispositivi “smart”, dotati cioè di intelligenza locale, quali telefoni e palmari con JME, Pocket PC, Symbian, Linux ecc., in grado di fungere direttamente da client dei servizi esposti o tramite la service interface del FSL o direttamente con il VSL (si pensi per esempio al caso in cui ai Vigili venga fornita la possibilità di confermare o meno l’effettivo cambio di residenza di un cittadino tramite il palmare che già oggi alcuni utilizzano per rilevare le infrazioni).

Per questioni di ottimizzazione è invece auspicabile che i diversi portlet sfruttino l’ambiente e i servizi del VSL per tutti i contenuti informativi, in questo modo limitando il colloquio con il VSL solo per gli aspetti transattivi.

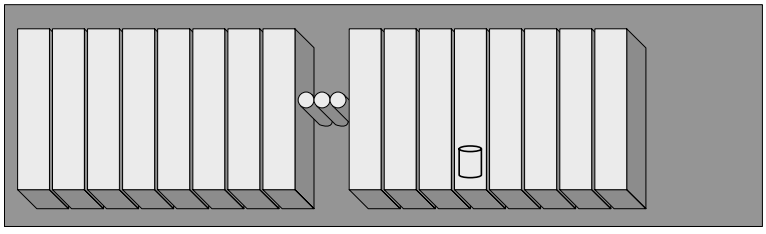
2.4 Back-End Service Layer

Lo strato dei servizi di Back-end è quello dove vengono messi a disposizione le operazioni fornite dai sistemi informativi comunali e da fornitori di servizi specializzati, quali quelli relativi al billing.

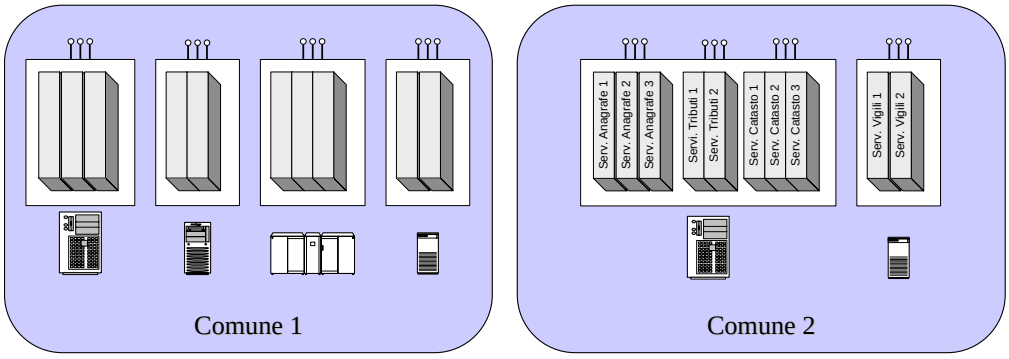


Ogni comune partecipante a PEOPLE metterà a disposizione, tramite le sue porte applicative, secondo i protocolli che verranno definiti, quei servizi che vuole rendere disponibili per la cooperazione a livello di comunità. In questo strato le richieste provenienti dagli strati superiori potranno quindi essere mappate nelle funzionalità supportate dalla specifica applicazione.

Alcuni di questi servizi saranno realmente di tipo transattivo e utilizzati dal Virtual Service Layer, altri di tipo informativo e messi a disposizione per il rendering tramite portlet al Front-End Layer di Comunità, altri ancora saranno esposti direttamente dal sito/portale del Comune, attraverso un proprio strato di Front-End.



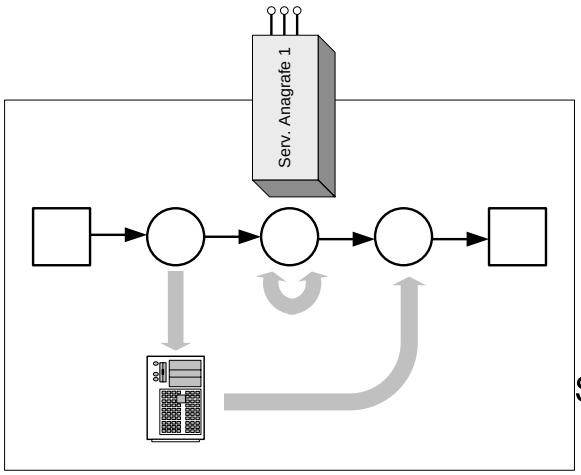
La topologia di implementazione dei servizi dipenderà dai singoli Comuni. In alcuni casi i servizi saranno distribuiti su più sistemi informativi separati e in sedi diverse, in altri casi faranno parte di un unico sistema informativo integrato



2.4.1 Rapporto tra Back-end Service Layer e Virtual Service Layer

Senza i servizi di back-end esposti dai Comuni e dalle altre amministrazioni il VSL è un vaso vuoto. E' fondamentale che il BSL dei Comuni implementi tutte le interfacce definite a livello di VSL, questo principalmente per ridurre la complessità dei workflow a livello del VSL, che in caso contrario, dovrebbero gestire una grande quantità di eccezioni.

Esporre "tutte" le interfacce non significa però ch sia necessario modificare l'implementazione dei sistemi di Back-end in modo da soddisfare in maniera compiuta le richieste provenienti dal VSL, quanto piuttosto creare dei programmi di riempimento (stub) che rispondano alle richieste provenienti dal VSL e che invece di generare una chiamata verso un sistema, producano una richiesta verso un "umano", per esempio inviando un messaggio di posta elettronica, lanciando una stampa o, in alcuni casi, semplicemente restituendo un valore predefinito.



Serv. Anagrafe 1 Serv. Anagrafe 2 Serv. Anagrafe 3 Servi. Tributi 1

Sist. Inf. Anagrafe Sist.

L'interfaccia dei servizi esposti dal BSL tende ad essere composta da operazioni relativamente "atomiche", in modo da permettere di spostare a livello del VSL la strutturazione del processo. Questo non significa però che tutte le richieste provenienti dal VSL debbano avere una corrispondenza uno-a-uno con le funzionalità presenti nei sistemi di Back-office del Comune. La specifica implementazione della porta applicativa del Comune si prenderà cura di accorpare più funzionalità o di virtualizzarne altre in modo da adattarle alla realtà locale. Da un punto di vista dei protocolli di comunicazione tra VSL e BSL si può sicuramente ipotizzare che in molti casi, quando VSL e BSL sono implementati all'interno dello stesso Comune, le richieste provenienti dal VSL non passino attraverso lo strato di WebServices/Porta applicativa – delegata, ma avvengano semplicemente come una chiamata di funzione locale. Tuttavia questa è da considerarsi una eccezione a scopo di ottimizzazione delle prestazioni e difficilmente generalizzabile, in quanto in molti casi, come si è detto, le funzionalità richieste dal VSL

risiedono su sistemi diversi (sia da un punto di vista fisico che di sistemi operativi/applicazioni) all'interno del Comune e in alcuni casi devono essere coinvolte altre amministrazioni locali o centrali (es. Motorizzazione) che difficilmente condivideranno i loro sistemi con quelli del Comune.

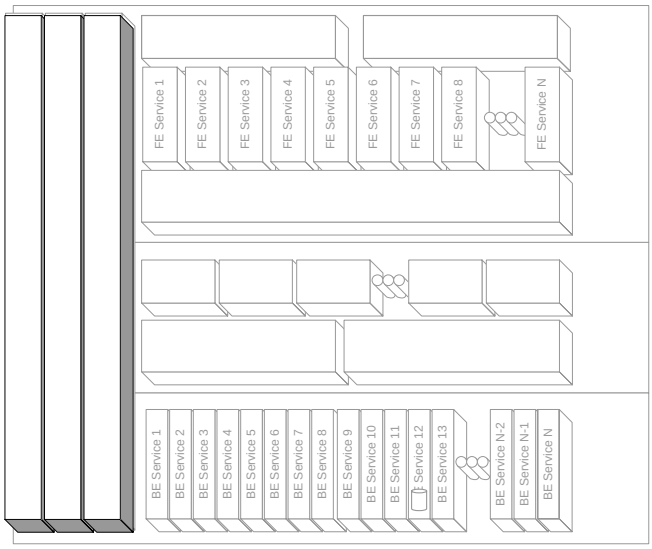
2.5 Servizi Strumentali e Servizi Infrastrutturali

Il modello a strati proposto per l'architettura di PEOPLE risponde a esigenze di organizzazione, di sviluppo, di scalabilità e di gestione e di dispiegamento che caratterizzano la natura e la complessità del Progetto. I differenti strati sono concepiti come funzionalmente disaccoppiati tra loro, soprattutto in previsione della possibilità che, in sede di dispiegamento del Portale PEOPLE presso i Comuni, lo strato di FSL e possa essere collocato anche in domini informatici differenti dal dominio in cui è collocato lo strato di VSL e soprattutto da quello del domino informatico del Comune dove è collocato lo strato di BSL.

È necessario quindi assicurare, secondo le necessità, lo scambio dati e l'integrazione applicativa tra i diversi strati dell'architettura che sono comunque immaginati come logicamente interconnessi attraverso una rete.

L'integrazione tra i tre strati della Architettura precedentemente descritti (FSL, VSL, BSL) ed il loro funzionamento sono assicurati grazie due tipologie di servizi di natura trasversale e che permeano tutti gli stati:

- i **Servizi Strumentali** propri del Framework di People;
- i **Servizi Infrastrutturali** propri della Comunità degli Enti People.



I Servizi Strumentali necessari al buon funzionamento del sistema PEOPLE sono:

- I Servizi di Integrazione applicativa
- I Servizi di Gestione
- I Servizi di Sicurezza

L'insieme di questi servizi e gli strumenti tecnologici che danno corpo al modello Architetturale descritto costituiscono il **Framework di PEOPLE** destinato ad ospitare le componenti applicative dei tre strati. Come detto il Framework di PEOPLE, concepito come logicamente unico, in sede di dispiegamento potrà essere collocato presso ogni comune oppure anche presso

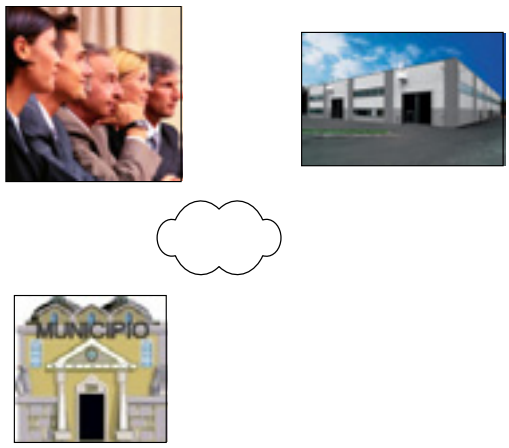
centri servizi per consentire la condivisione del FSL, ed eventualmente anche del VSL, tra i Comuni partecipanti al progetto secondo le autonome scelte di ciascun Ente.

Il funzionamento di PEOPLE richiede la disponibilità di ulteriori **Servizi Infrastrutturali**, la cui particolare natura suggerisce che vengano considerati condivisi tra tutti gli Enti membri della Comunità di PEOPLE, ma esterni al Framework, e quindi esterni al dominio informatico degli Enti PEOPLE. Questi servizi possono essere tipicamente erogati o dai altri soggetti pubblici, ad esempio i gestori delle Reti regionali, oppure anche da soggetti privati specializzati e accreditati. I Servizi Infrastrutturali non saranno quindi progettati e realizzati da PEOPLE, ma saranno acquisiti, in termini di servizi telematici, dai soggetti terzi che si conformeranno ai requisiti funzionali ed economici del Progetto PEOPLE.

Il Framework PEOPLE comprenderà quindi solamente quei componenti necessari alla integrazione dei Servizi Infrastrutturali erogati da terze parti.

2.5.1 Servizi di Integrazione applicativa

I Servizi di Integrazione applicativa hanno il compito di permettere l'interazione tra i componenti esposti dai diversi strati, utilizzando, in maniera indifferenziata e a seconda della necessità, sia meccanismi locali di comunicazione ad alte prestazioni (ad esempio chiamate di funzione piuttosto che Local RPC), quando ci si trovi in una condizione di co-località, che meccanismi di comunicazione basati sul protocollo TCP/IP (più tipicamente Web services). I primi utilizzano i servizi di comunicazione tra processi offerti dai sistemi operativi, mentre gli ultimi richiedono un'infrastruttura di comunicazione. Nel caso di PEOPLE non si prevede una infrastruttura autonoma di comunicazione (intranet), ma i comuni facenti parte di PEOPLE utilizzeranno Internet e, dove applicabile, le reti regionali per comunicare con i Centri servizi di PEOPLE o con gli erogatori dei Servizi Infrastrutturali.



E' ormai dimostrato, infatti, che l'infrastruttura di Internet, anche grazie alle caratteristiche del protocollo IP, ha raggiunto dei livelli di servizio tali da competere con le infrastrutture interne alle aziende, raggiungendo dei livelli di uptime di tutto rispetto. Considerando l'estrema granularità e distribuzione dei comuni partecipanti a PEOPLE, l'ipotesi di creare una rete privata autonoma, con gli stessi livelli di servizio, è economicamente impraticabile. Questo tuttavia pone degli importanti requisiti sulla gestione della comunicazione sicura. Se il livello di affidabilità di Internet è ormai consolidato, non si può dire altrettanto per quanto riguarda la riservatezza delle informazioni e l'integrità del canale. Onde ovviare a questo problema è necessario che le comunicazioni avvengano su un canale sicuro (SSL o IPsec) ma non basta, dato che la de-crittografazione del canale avviene a livello degli edge server ai due estremi del canale virtuale, questi potrebbero essere stati compromessi da un hacker, con il conseguente problema di una possibilità di manipolazione dei dati in transito, per questa ragione i dati stessi dovranno essere firmati in maniera elettronica dalla applicazione sorgente così da garantire che

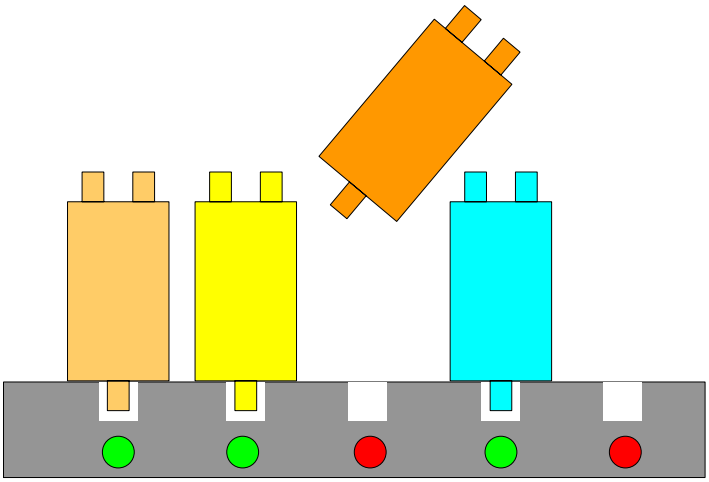
l'applicazione ricevente possa verificarne l'integrità (a meno che anche l'applicazione non sia stata compromessa).

Da questo punto di vista è corretto aspettarsi che tutte le chiamate tra i diversi strati del Framework di PEOPLE, quando non avvengano all'interno dello stesso dominio informatico, e le chiamate verso i Servizi Infrastrutturali, utilizzino sempre e comunque un unico modello di certificazione dell'integrità dei messaggi, per esempio quello proposto da WS-Security e XML-Signature.

2.5.2 Servizi di Gestione

Perché un sistema complesso come quello di PEOPLE possa rispondere in maniera adeguata ai requisiti di affidabilità e disponibilità richiesti, è fondamentale la creazione di Servizi di gestione non semplicemente sistemistici, ma anche applicativi. In particolare sarà necessario poter gestire in maniera dinamica l'aggiunta/rimozione di comuni/fornitori/servizi/utenti.

Questo significa rinunciare al modello monolitico di costruzione/gestione delle applicazioni, per un modello in cui vengono favoriti gli aspetti di configurabilità dinamica delle applicazioni e di esternalizzazione delle interfacce di gestione. Per riprendere un metafora oggi sempre più diffusa, l'infrastruttura di gestione di PEOPLE deve essere simile a quelle delle più diffuse "utility" (elettricità, telefono, gas), e in particolare deve essere considerata la creazione di un sistema di Provisioning generalizzato.



Come abbiamo indicato questo modello modifica radicalmente il modo in cui le applicazioni e i servizi applicativi sono "costruiti", acquisiti ed utilizzati:

- I servizi applicativi devono poter essere registrati e "aggiunti" in maniera dinamica al sistema senza richiedere interruzioni di servizio.
- Devono adattarsi agli standard che verranno definiti per la generazione di log e eventi asincroni (trap) per la tracciatura di problemi e per la produzione di informazioni sull'utilizzo (sia puntuale che statistico) in modo da non richiedere interfacce (sia umane che programmatiche) diverse per la gestione.
- Devono esporre interfacce per la configurazione di nuove entità (comuni e utenti) senza necessitare l'intervento umano.

A questi vincoli sulle applicazioni corrispondono altrettante standardizzazioni e automatismi nei servizi di gestione, così che sia realmente possibile attivare on-demand i servizi stessi e, ove necessario, configurare il sistema di billing o di pagamento, scelti dallo specifico comune, così che l'utilizzo possa essere monitorato ed eventuali addebiti ai cittadini possano essere immediatamente basati sul reale utilizzo.

2.5.3 Servizi di Sicurezza

L'argomento della sicurezza in un ambiente distribuito come quello di PEOPLE è di estrema importanza e complessità. E' fondamentale che PEOPLE possa contare su Servizi di sicurezza che garantiscano tanto la riservatezza che l'integrità delle informazioni trattate, oltre alla rispondenza ai requisiti di legge. Perché ciò avvenga dovranno essere definiti e utilizzati tutti quei meccanismi che permetteranno:

- L'identificazione certa degli attori coinvolti
- L'implementazione di meccanismi di delega
- La certificazione dell'integrità delle informazioni
- La certificazione dell'avvenuta sottomissione/esecuzione di una operazione (non ripudio)
- L'assicurazione che le credenziali degli utenti non vengano compromesse
- La prevenzione da attacchi maliziosi volti a compromettere l'integrità dei servizi
- La prevenzione da attacchi maliziosi volti a compromettere la disponibilità dei servizi
- La prevenzione da virus
- Ecc.

Tuttavia, poiché la maggior parte degli attacchi alla sicurezza fanno leva sul fattore umano, sui suoi errori ed ingenuità, si deve ricordare che la sicurezza è solo in minima parte assicurata da strumenti tecnici, ma necessita anche di educazione e di attitudine alla gestione del rischio. Di conseguenza il termine "Servizi di sicurezza" non indica solo un insieme di servizi tecnologici, ma un insieme di regole e comportamenti organizzativi che deve influenzare il disegno, l'implementazione e la gestione quotidiana di tutte le componenti facenti parte del Framework di PEOPLE. Come accennato precedentemente a proposito dei servizi di comunicazione, i servizi di sicurezza dovranno assicurare la creazione di canali sicuri, fornire gli strumenti per l'autenticazione degli utenti, la crittografia dei messaggi e così via.

Capitolo 3

I Servizi infrastrutturali

3.1 Analisi dei Servizi di Pagamento

3.1.1 Pagamenti con carte di credito

La modalità di pagamento con carte di credito è quella più tradizionalmente utilizzata su Internet. L'utente esegue il pagamento di un servizio inserendo in un modulo i dati della propria carta di credito. La denominazione di "basata sul credito" deriva dal fatto che una banca "acquirer" garantisce il credito per una persona, o per una azienda, nei confronti del circuito emittente la carta, fino ad una certa cifra giornaliera o mensile. In questo modo nel momento in cui il portatore della carta effettua un pagamento non viene controllata la sua reale disponibilità monetaria, il suo conto in banca. Il creditore (nel nostro caso tipicamente la Tesoreria del Comune) può quindi essere immediatamente accreditato del valore dell'operazione.

Dal punto di vista tecnico esistono due modalità in cui è possibile fornire il servizio di pagamento con carta di credito nei confronti di un Comune:

1. Pagamento diretto sul sito del Comune;
2. Ridirezione dell'utente dal sito del Comune al sito di una banca (potrebbe essere proprio la banca che gestisce la Tesoreria del Comune).

Nel caso di pagamento eseguito direttamene sul sito del Comune, questi deve procurarsi e gestire un apposito software, chiamato payment gateway, e deve formulare un accordo con una "banca acquirer" (potenzialmente diversa da quella che gestisce la Tesoreria) che funge da intermediario verso i circuiti internazionali (VISA, AMEX ...). La "banca acquirer" fornirà al Comune i codici identificativi da inserire nel payment gateway e si procurerà di accreditare su un conto corrente della Tesoreria del Comune gli importi delle transazioni.

Oltre al fatto che il Comune deve sostenere il costo per l'acquisizione e la gestione del software del gateway e della linea dedicata verso la banca acquirer, i problemi di questo approccio sono diversi, in primo luogo di tipo tecnologico/organizzativo, infatti le informazioni relative alle carte di credito transitano in chiaro all'interno dei sistemi del Comune. È quindi di estrema importanza assicurarsi che non solo la comunicazione tra la pagina web che riceve i dati della carta di credito e il software del gateway avvengano in maniera sicura, ma che tutto il software tra essi frapposto sia privo di codice che possa dare origine a problemi di sicurezza o "cavalli di Troia" inseriti da qualche malintenzionato, che le informazioni siano memorizzate solo per il tempo necessario alla comunicazione al gateway e poi vengano eliminate. (Meccanismi quali quello di memorizzare le informazioni della carta di credito per evitare che l'utente sia costretto

a inserirle tutte le volte sono estremamente pericolosi).

Nel caso di ridirezione, nel momento in cui il l'utente è invitato a pagare, attraverso la restituzione di uno speciale header HTTP, avviene una "ridirezione" del browser dal sito del Comune a una pagina protetta di una banca, ad esempio la banca che gestisce la Tesoreria del Comune.

È opportuno rendere esplicita tale ridirezione presentando sulla pagina indicazioni che informano l'utente del fatto di trovarsi su una pagina protetta di una banca, e non più sul sito del Comune, cosa di cui egli si può accertare chiedendo la verifica del certificato del server al proprio browser. In questo modo l'utente è "tranquillizzato" dal fatto di avere un rapporto con un ente, la banca, con cui è abituato a fare transazioni monetarie.

Nel momento della ridirezione vengono inseriti negli header HTTP (o in POST) dei parametri crittografati, così che arrivando sul sito della banca siano pre-valorizzati:

- La causale dell'operazione
- L'importo dell'operazione
- L'ente richiedente l'addebito

Dopo che l'utente ha inserito i dati della propria carta di credito e la transazione viene approvata, avverrà una nuova ridirezione verso una pagina predefinita del sito originale del Comune, dove l'effetto della transazione verrà registrato. A valle della transazione avviene comunque una comunicazione server-to-server o via e-mail da parte della banca che ulteriormente conferma il risultato dell'operazione. In generale sarebbe utile disporre anche di una console di amministrazione che permetta di controllare lo stato delle richieste.

Quest'ultima modalità solleva il Comune dal dover acquisire e gestire il payment gateway e anche dalla responsabilità della gestione sicura dei dati delle carte di credito, che è a totale carico della banca riindirizzata per il pagamento.

Un problema di questo approccio, per altro ormai ampiamente collaudato nel contesto e-commerce, è dato dal fatto che, per quanto i meccanismi messi in opera dalle diverse banche siano molto simili, non sono standardizzati, con la conseguenza che l'aggiunta di banche diverse comporterebbe un lavoro di integrazione sempre differente. Data l'assoluta scalabilità economica di una soluzione di questo tipo (non richiedendo che minimi investimenti di integrazione al Comune) uno degli obiettivi di PEOPLE sarà perciò la standardizzazione di un protocollo comune, che le diverse banche, in particolare almeno le banche che gestiscono le Tesorerie dei Comuni PEOPLE, dovrebbero impegnarsi ad implementare.

Poiché, per entrambe le modalità, su ogni transazione viene applicata una "tariffa" sia da parte del circuito internazionale che da parte della banca acquirer, questa modalità di pagamento è economicamente poco praticabile per transazioni di valore inferiore ai 5-10 Euro.

3.1.2 Pagamenti basati sul debito

In questa modalità di pagamento con la banca non avviene alcuna transazione economica se non nel momento in cui l'importo viene effettivamente addebitato sul conto dell'utente. Pertanto non è data alcuna garanzia che, in quel momento, l'utente abbia effettivamente la disponibilità economica necessaria. Questo meccanismo è quello comunemente utilizzato per gli assegni bancari, i bonifici e il Bancomat (in cui la verifica viene però effettuata immediatamente).

Prescindendo dal caso del Bancomat utilizzato online (cosa non percorribile dato che l'utente non ha idea del codice memorizzato nella banda magnetica, mentre per l'inserimento del PIN è richiesto dell'hardware dedicato) questo meccanismo richiede una gestione asincrona in cui, per esempio per i bonifici o pagamenti tramite sportello bancomat:

- Vengono raccolti tutti i dati relativi all'operazione richiesta che, in base alla sua

tipologia e costì, può essere o meno eseguita immediatamente a discrezione del Comune

- Viene generato un codice di operazione da associare alla richiesta
- Viene generato un documento che l'utente può stampare o memorizzare, contenente il codice dell'operazione quale causale, l'importo, gli estremi bancari del Comune e quant'altro
- L'utente effettua l'operazione di bonifico presso la propria banca
- La banca comunica in forma elettronica i pagamenti effettuati
- Il sistema di back-end consolida le operazioni decidendo eventuali azioni in caso di assenza o ritardo del pagamento

Nel caso dei RID e dei MAV la comunicazione tra Comune e banche è già probabilmente attiva, si dovrà prevedere, come nel caso precedente la sincronizzazione batch tra il sistema di back-end e la comunicazione da parte delle banche, tuttavia perché RID e MAV funzionino l'utente deve aver preventivamente autorizzato la banca ad effettuare movimenti da parte del Comune.

I problemi di questo approccio sono molteplici, quello comune a tutti è l'asincronicità della sincronizzazione tra l'operazione richiesta e la comunicazione dell'effettivo addebito all'utente.

Nel caso del pagamento tramite bancomat o bonifico bancario non ci sono ulteriori complicazioni, se non nella diffusione dell'applicazione di pagamento presso un numero di istituti bancari tali da non creare eccessivi disagi agli utenti.

Nel caso invece di RID e MAV possono sorgere problemi legati all'effettiva esistenza del conto indicato dall'utente e dell'autorizzazione di quest'ultimo ai pagamenti.

Anche se con minore priorità rispetto al modello precedente, PEOPLE dovrebbe, appoggiandosi agli standard bancari esistenti, definire un insieme di protocolli che permettano di interfacciarsi con le banche utilizzando un solo modello di interfacce applicative.

3.1.3 Pagamenti basati su gettoni

In questa categoria rientrano quei sistemi che utilizzano la crittografia per creare gettoni (token), banconote digitali che possono poi essere spese in maniera totalmente elettronica. Il modello è quello in cui esistono tre attori:

- Il borsellino elettronico
- Il broker (banca)
- La Tesoreria del Comune

Inizialmente l'utente acquista secondo diversi canali (bonifico, acquisto con carta di credito, acquisto di una scheda di ricarica o altro) un codice che, collegandosi con il sito del broker gli permetterà di acquistare un certo numero di banconote digitali. Il broker crea questi gettoni firmando un messaggio che specifica il numero di serie ed il valore delle singole banconote, e li restituisce all'utente. In maniera trasparente all'utente questi vengono inseriti nel suo borsellino elettronico. Nel momento in cui l'utente deve effettuare un pagamento trasferisce una banconota elettronica alla Tesoreria del Comune che ne controlla la validità, decifrando la banconota con la chiave pubblica della banca per controllarne la firma, quindi la invia alla banca che, ne controlla il numero di serie per assicurarsi che non sia già stata spesa e quindi informa la Tesoreria della validità della banconota e gli accredita il valore.

Esistono diversi tipi di borsellini elettronici, sia sottoforma di applicazioni da installare sul PC dell'utente che accessibili via web. Il sistema è sicuro e scalabile. Il problema è la non standardizzazione dei protocolli. Oggi in Italia esistono già diversi sistemi di questo tipo (Carta Facile, Carta Kalibra, OmniPay, Skio, Tell Internet Card, solo per citarne alcuni) e tutti incompatibili tra di loro. Anche se la soluzione è di sicuro interesse e l'abitudine all'acquisto di carte prepagate è diffusissimo tra tutte le fasce di utenza, una standardizzazione è necessaria. PEOPLE potrebbe decidere di acquisire una propria soluzione proprietaria (l'ennesima), rispondente alle norme sulla firma digitale, da condividere a livello dei partecipanti l'attività, o di selezionare tramite gara la soluzione di un singolo fornitore.

Un utilizzo di questa modalità potrebbe offrire una soluzione al pagamento dell'imposta di bollo per i servizi che lo richiedono, tramite l'acquisto di "carte o marche da bollo virtuali" da utilizzare per la compilare le richieste..

3.1.4 Pagamento per bollettazione (billing)

Come si è accennato precedentemente è fondamentale che l'architettura di PEOPLE segua il più possibile un modello "utility" e come tale implementi un modello flessibile sia di acquisizione e fornitura di un servizio, che di associazione tra un servizio e il suo costo. Le società di telecomunicazioni sono state le prime a dover affrontare il problema di offrire ai propri abbonati sempre più servizi a valore aggiunto, con il duplice scopo di soddisfare le loro esigenze e trovare nuove fonti di reddito che non fossero il semplice traffico telefonico. Da soluzioni iniziali monolitiche, il cui la rete, i diversi servizi e la contabilizzazione facevano parte di un'unica grande applicazione, si è passati ad un modello che si basa su quattro componenti fondamentali:

- Servizio
- Provisioning
- Mediation
- Billing

L'idea fondamentale è quella che, mentre l'implementazione tecnologica di un nuovo servizio può essere proprietaria e di complessità qualunque, l'aggiunta del nuovo servizio applicativo deve invece essere fatta rispettando un insieme molto preciso di interfacce per la:

- Creazione/cancellazione di nuovi utenti
- Abilitazione/disabilitazione di funzionalità del servizio
- Generazione di record di consumo
- Generazione di allarmi

In questo modo una volta messo in esercizio sarà possibile interagire (da un punto di vista amministrativo) con il servizio senza bisogno di utilizzare interfacce proprietarie.

Il servizio del Framework che si prende cura della gestione tecnologica degli utenti nei confronti dei servizi è quello del "provisioning". Questo registra l'esistenza di nuovi servizi e delle informazioni che questi necessitano, così che quando il sistema di "accoglienza" del Comune inserisce le informazioni di un nuovo utente, precisando i servizi a cui questo è iscritto, il sistema di "provisioning" si prende cura di propagarle in maniera adeguata, dopo di che i diversi servizi si possono considerare attivati.

La contabilizzazione e bollettazione di un'unica tipologia di servizio è relativamente semplice: è sufficiente inserire i dati di consumo e stabilire eventuali fasce di prezzo per arrivare a creare la bolletta. Diversamente, quando si hanno molti servizi, integrarne la valorizzazione all'interno del

servizio stesso crea degli enormi problemi nel caso in cui si devono applicare politiche diverse, per esempio per tipologia di utenti, o semplicemente quando cambiano le aliquote fiscali.

Il servizio del Framework che si occupa invece di raccogliere i dati relativi all'utilizzo dei diversi servizi di rete, li uniforma, applica delle regole e politiche tariffarie e li converte in un formato comprensibile per il sistema di "bollettazione" e li comunica a quest'ultimo, è quello di "mediation". In questo modo i dati di consumo sono aggregati e convertiti in un singolo punto e diventa molto più semplice stabilire delle regole che si applicano in maniera trasparente e coerente a tutti i servizi. Il servizio di mediation si prende anche cura di distribuire gli accrediti tra i diversi fornitori di servizi (per esempio offerti da altri domini) i sistemi di billing e, nel caso di sistemi prepagati, ai sistemi che scaricano il valore dalla tessera.

L'accoppiata provisioning/mediation permette di ridurre il lavoro manuale di configurazione e attivazione dei servizi, accelera la messa in esercizio di questi ultimi e permette di introdurre dei meccanismi flessibili di bollettazione oltre a un controllo più efficace degli addebiti.

Questi servizi del Framework sono strumentali anche al servizio di billing. Questo come abbiamo visto, riceve le informazioni di tariffazione dal/dai sistemi di mediation, le informazioni vengono aggregate per la creazione della bolletta e, quindi, addebitate all'abbonato secondo le procedure concordate. Il sistema di billing è la parte più tipicamente amministrativa del sistema, ha tutte le informazioni relative alle modalità di pagamento dell'abbonato, se paga tramite addebito sul conto corrente, se riceve bollette bimestrali piuttosto che semestrali, l'indirizzo a cui inviare la bolletta ecc.

In genere il billing può essere delegato a società specializzate, dato che la gestione amministrativa è assolutamente indipendente da ciò che effettivamente viene erogato come servizio.

Da questa discussione si comprende che mentre per le relazione interne tra servizi di provisioning e mediation PEOPLE ha una totale autonomia di definizione delle interfacce, dovrà invece sforzarsi non solo di standardizzare le interfacce verso i sistemi di billing, ma di ottenere un consenso sul formato da parte di un certo numero di organismi che potrebbero diventare erogatori del servizio. Una volta standardizzata l'interfaccia sarà possibile scegliere soluzioni diverse per l'effettivo pagamento; alcuni Comuni potrebbero scegliere di fare emettere bollette periodiche sui conti bancari dell'utente, altri di addebitare sulle cedole della nettezza urbana, altri ancora potrebbero stipulare accordi con le società elettriche o del gas ecc.

3.2 Il Servizio di Autenticazione

In un sistema distribuito come quello di PEOPLE l'aspetto dell'autenticazione degli utenti riveste una particolare importanza. Da un lato, infatti, si vorrebbe poter estendere al maggior numero di cittadini possibile l'accesso ai servizi, dall'altro si vorrebbe poter garantire la sicura autenticazione degli stessi, in modo da poter garantire un corretto funzionamento del sistema sia da un punto di vista operativo che legale. L'obiettivo finale del meccanismo di autenticazione di PEOPLE è in definitiva quello di associare, in maniera certa, un richiedente un servizio a un Codice Fiscale certificato.

Per il riconoscimento sicuro degli utenti PEOPLE non prevede l'emissione di una propria carta servizi, ma accetterà tutte le carte emesse da amministrazioni pubbliche aventi valore legale. Naturalmente ciò implica la disponibilità presso gli utenti dei relativi lettori e del software opportuno. Ma mentre è ragionevole attendersi che con la disponibilità di dispositivi fisici quali la CIE (carta d'identità elettronica), le carte di firma e la CNS il problema del riconoscimento certo degli utenti problema potrà essere definitivamente risolto, vista la relativa limitata diffusione di tali dispositivi è necessario che PEOPLE fornisca ed utilizzi anche altri meccanismi quali:

- Modalità di **registrazione debole**
- Modalità di **registrazione forte**
- Accesso con tramite i certificati di autenticazione delle carte di **firma digitale,** per gli intermediari i professionisti e le aziende

In particolare:

- Per i servizi accessibili con registrazione debole (in pratica serve solo creare un profilo utente) saranno utilizzate coppie "Username e Password".
- Per i servizi accessibili con registrazione forte sarà utilizzata una terna "Codice Fiscale, Password, PIN" assegnata previo riconoscimento sicuro.
- Laddove praticabile potranno essere utilizzati anche certificati digitali di autenticazione

Nel caso dei dispositivi fisici precedentemente citati, e dei certificati digitali, la prova di autenticità è data dal fatto di possedere un oggetto (la smart card o il certificato) e di sapere come "aprirlo" perché possa essere utilizzato, utilizzando il PIN contenuto nella smart-card stessa o la password utilizzata per aprire il secure-storage sul computer. Nel caso invece di meccanismi quali username/password/PIN è necessario che sia il meccanismo di autenticazione di PEOPLE che verifichi la corretta conoscenza dei "codici segreti" da parte dell'utente.

Come tali codici siano stati recapitati agli utenti in maniera tale da garantire la corretta associazione è al di fuori di questa discussione, a tale proposito si può trarre spunto dai meccanismi utilizzati dal Ministero delle Finanze per la dichiarazione dei redditi o dalla POSTE.

Una volta che un utente ha ricevuto per qualche via i dispositivi fisici o le informazioni segrete, è responsabilità dei Comuni partecipanti a PEOPLE autenticare gli utenti stessi per dare accesso ai servizi.

In un sistema locale tale funzionalità potrebbe non essere un problema: l'accoppiata username e l'hash della password potrebbero essere memorizzati localmente in database locale o in servizio di directory LDAP, in un sistema distribuito come PEOPLE invece tale soluzione non sarebbe accettabile perché:

- L'utente sarebbe costretto a ri-registrarsi nei diversi comuni a cui vuole avere accesso (per esempio quello della residenza principale, quello della scuola dei figli), con conseguente ripetizione del meccanismo di assegnazione della password
- L'utente potrebbe impostare/cambiare la password in momenti differenti nei diversi comuni, con il conseguente problema di perdita di sincronizzazione delle password e la potenziale necessità di riassegnazione della password in maniera certificata
- Al momento del login la password dell'utente diventerebbe visibile nei sistemi informativi del Comune con conseguente possibilità di furto da parte di malintenzionati che avessero compromesso il sistema

Inoltre la necessità di supportare tutti i diversi tipi di autenticazione obbligherebbe i Comuni a:

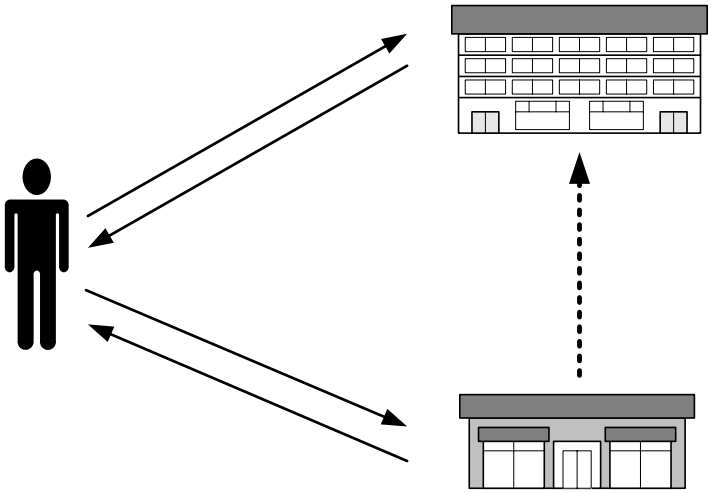
- Farsi carico delle licenze e gestire i software atti a interagire con i dispositivi fisici
- Farsi carico di verificare che i certificati presentati siano ancora validi, tramite verifica della CRL delle Certification authorities

Data la complessità e la criticità dei sistemi coinvolti PEOPLE utilizzerà un modello di autenticazione basato sul modello degli Authentication Server.

3.2.1 Il modello

Nel modello "Autentication Server" un attore principale (nel nostro caso il Comune) attribuisce la fiducia a una terza parte (l'authentication server) di riconoscere una persona (l'utente) e di fornirgli in maniera sicura delle credenziali (token) che verranno accettate dall'attore principale.

Il modello non è differente da quello utilizzato nella vita comune, in cui si accetta l'identità di una persona per il fatto che questa è in possesso di un documento (per esempio il passaporto) emesso da un'autorità riconosciuta (la questura) e che tale documento appare integro, non manipolato (vedi figura).



Da un punto di vista tecnologico, il caso specifico di PEOPLE, questa interazione avviene utilizzando i seguenti meccanismi:

- Ridirezione del browser dell'utente (supportata dal protocollo HTTP)
- Utilizzo delle intestazioni della richiesta HTTP (o Javascript al caricamento della pagina)
- Chiavi simmetriche
- Timestamp

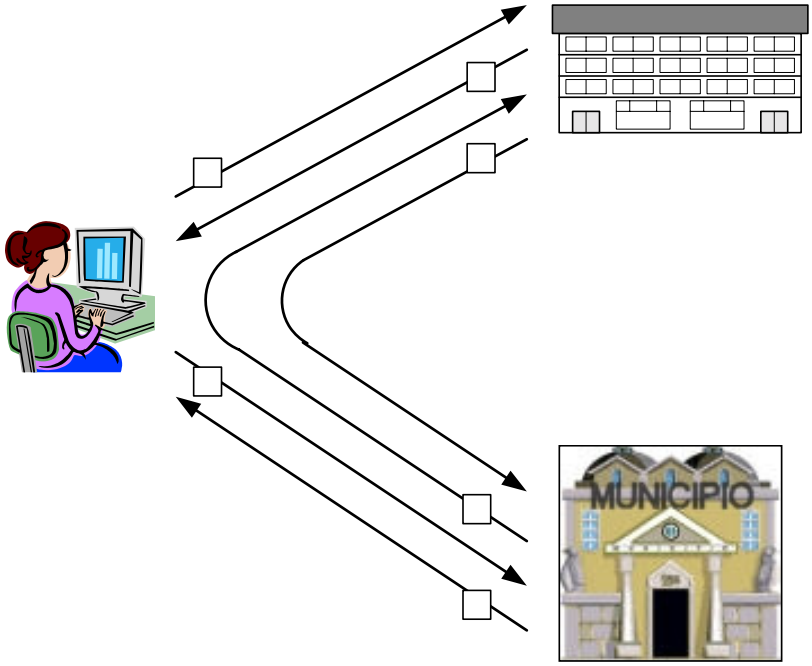
Prerequisito all'operazione è l'accordo tra il FSL e il server di autenticazione, in particolare per la registrazione presso quest'ultimo di:

- chiave simmetrica con cui crittografare i messaggi
- gli stili (CSS o XSLT) con cui presentare la pagina di login all'utente.
- la pagina di logout

In questi meccanismi, per ragioni di velocità e di spazio, viene in genere utilizzata una chiave simmetrica, nulla impedisce che venga utilizzato un meccanismo basato su chiavi asimmetriche.

3.2.2 Meccanismo di Login

L'interazione avviene secondo la modalità evidenziata nella seguente figura):



1) L'utente si connette con il FSL del comune e ad un certo punto accede ad un servizio che richiede autenticazione
2) Il FSL scopre che l'utente non è autenticato, di conseguenza risponde ridirigendo (attraverso l'HEADER HTTP) il browser dell'utente all'indirizzo del servizio di autenticazione utilizzando il protocollo HTTPS, così da garantire la riservatezza delle informazioni scambiate. Tra i parametri passati nell'header ci sono: 4
    a. La pagina del FSL a cui indirizzare l'utente in caso di successo (una pagina a cui viene passato come parametro l'indirizzo della pagina richiesta dall'utente)
    b. La pagina del FSL a cui indirizzare l'utente in caso di insuccesso (pagina di accesso negato)
    c. L'identificativo del FSL
    d. Un timestamp
3) Il server di autenticazione, utilizzando l'identificativo del FSL visualizza una pagina personalizzata di login all'utente, proponendogli di utilizzare uno dei meccanismi di autenticazione disponibili:
    a. CIE
    b. CNS
    c. Usernane/password
    d. …
4) L'utente fornisce le proprie credenziali
5) Il server di autenticazione valida le credenziali dell'utente e: 1
    a. In caso di successo utilizza l'identificativo del FSL per recuperare la chiave con cui crittografare il messaggio di ritorno. Prepara il messaggio, contenente
        - Il proprio identificativo (in chiaro)
        - Il codice fiscale dell'utente, l'identificativo del FSL, ll timestamp dell'operazione, il proprio codice di log dell'operazione crittografati

Inserisce il messaggio nell'HEADER e ridirige il browser dell'utente verso la pagina richiesta. In questo passaggio "scarica" anche un proprio cookie contenente in maniera crittografata (con una propria chiave privata) le stesse informazioni, così da poterle ritrovare in caso l'utente tornasse nuovamente ad autenticarsi entro un certo lasso di tempo.

b. In caso di fallimento dell'autenticazione utilizza l'identificativo del FSL per recuperare la chiave con cui crittografare il messaggio di ritorno. Prepara il messaggio, contenente

- Il proprio identificativo (in chiaro)
- La ragione del fallimento, ll timestamp dell'operazione, il proprio codice di log dell'operazione crittografati

Ridirige il browser dell'utente verso la pagina di errore

6) Il FSL recupera le informazioni dagli HEADER del messaggio e, dopo aver verificato dal timestamp la "freschezza" dell'informazione, utilizzerà il codice fiscale per controllare l'autorizzazione all'accesso della pagina richiesta. In questo caso restituirà la pagina all'utente, eventualmente scaricando un proprio "cookie" di sessione.

3.2.3 Meccanismo di Single Sign On

E' evidente che per tutti i servizi esposti da uno stesso FSL e per tutti i FSL serviti dal uno specifico authentication server l'utente non avrà più bisogno di reinserire le proprie credenziali, in quanto qualunque servizio richiedesse l'autenticazione non farebbe altro che ridirigere il browser dell'utente verso l'authentication server, il quale ritroverebbe il proprio cookie, e scoprirebbe dal timestamp se la sessione di autenticazione dell'utente è ancora valida. In questo caso non avrà bisogno di richiedere nuovamente le credenziali e potrà preparare l'header di ritorno.

Per l'utente l'operazione sarà completamente trasparente, se non per il minimo ritardo introdotto dall'operazione e dall'effetto visibile nella ridirezione nella casella degli indirizzi del browser.

3.2.4 Federazione tra gli authentication server

Si è evidenziato al punto precedente che il meccanismo di Single Sign On funziona in maniera trasparente esclusivamente per tutti i FSL serviti da uno specifico authentication server, in quanto il meccanismo si basa sul recupero di uno specifico cookie (o HEADER) generato dall'authentication server. In presenza di più entità eroganti il meccanismo di autenticazione questa è solo un minimo disturbo per l'utente, in quanto è difficile che l'utente si aspetti di essere comunque "loggato" passando da un Comune ad un altro. Quello che però è fondamentale che non debba utilizzare credenziali differenti. Pertanto mentre per quanto riguarda l'accesso tramite smart card la cosa è automatica, per l'accesso tramite username/password è fondamentale che le diverse entità fornenti il servizio di autenticazione creino un meccanismo di replica delle informazioni contenute nei server LDAP, creando in questo modo un modello federato.

3.2.5 Meccanismo di Log-out e termine di sessione

Per ragioni di sicurezza i diversi FSL dovrebbero impostare un limite alla durata di una sessione autenticata, in particolare se inattiva, mentre il meccanismo di autenticazione dovrebbe supportare la possibilità per l'utente di fare log-out da PEOPLE. Come abbiamo indicato precedentemente l'authentication server tiene traccia in un proprio cookie (ed eventualmente nel

proprio back-end) di tutti i FSL che hanno richiesto (recentemente) l'autenticazione dell'utente. Qualora l'utente richiedesse il log-out dal servizio di PEOPLE prima dello scadere della sessione di autenticazione, il FSL dovrebbe ridirigere il browser dell'utente verso una specifica pagina messa a disposizione dall'authentication server. Questo, a sua volta richiamerà, con lo stesso meccanismo, tutte le pagine di logout di tutti i FSL ritenuti ancora attivi, in questo modo sconnettendo a tutti gli effetti l'utente da tutti i predetti FSL.

3.3 Gestione distribuita dei puntatori ai procedimenti

Come detto nell'introduzione il progetto PEOPLE è costituito da un'aggregazione di amministrazioni locali che rappresentano altrettanti punti di riferimento nella definizione e gestione di sistemi informatici per l'accesso ai servizi territoriali da parte di cittadini e imprese. Questa aggregazione non significa però unificazione, in quanto ogni amministrazione ha, per legge, il dovere di custodire in autonomia e riservatezza le informazioni relative ai cittadini che amministra. Di conseguenza PEOPLE non è un sistema centralizzato, in particolare per quanto riguarda i dati personali.

Ne consegue che un cittadino che avesse avuto la necessità di avviare procedimenti in comuni differenti, per esempio nel caso in cui avesse proprietà in diverse città, figli che vanno a scuola in comuni diversi da quello di residenza e così via, potrebbe avere difficoltà a ricostruire la storia dei propri procedimenti. Oggi, con le pratiche cartacee, gli basterebbe aprire un cassetto per ritrovare i documenti, le denuncie e così via, domani su PEOPLE, con i dati posseduti da amministrazioni diverse sarebbe costretto a navigare da un comune all'altro tra quelli con cui ha avuto relazioni, per ricostruire la situazione.

Una possibile soluzione tecnologica potrebbe essere quella di accentrare tutta la documentazione in un unico punto, ma sappiamo che questo non è percorribile da un punto di vista normativo, amministrativo e gestionale. Un'altra potrebbe fare affidamento sulla posta elettronica: ogni volta che un cittadino avvia una pratica in meniera elettronica, e questa viene protocollata riceve un messaggio di posta elettronica con le informazioni di protocollazione. A questo punto è sua responsabilità stamparla o comunque non perderla.

L'approccio che invece viene proposto a livello architetturale è quello di utilizzare un servizio condiviso a livello architetturale (ed eventualmente federato a livello fisico) di memorizzazione di "puntatori a procedimenti".

Quando un cittadino avvia un procedimento, il sistema, oltre ad inviare tutte le ricevute del caso, crea un codice identificativo univoco del procedimento e lo associa al Codice Identificativo dell'utente, indipendentemente dal FSL, o dal comune, attraverso il quale il cittadino sta interagendo con PEOPLE.

Questo codice univoco non porta con sè informazioni sulla tipologia di procedimento, o altro dato personale, ma solo un codice univoco, l'identificativo del FSL e del comune a cui è stato "sottomesso" e un generico stato ("sottomesso", "letto", "protocollato", "respinto", "accettato" ecc.).

Questa assenza di dati sensibili permette la memorizzazione e la replicazione del codice senza conseguenze per la privacy del cittadino.

Quando il cittadino tornerà su PEOPLE, una volta autenticato, potrà richiedere la visualizzazione di tutti i propri procedimenti. In automatico potrebbe partire una richiesta di aggiornamento dello stato per quei procedimenti che risultano ancora in corso (o che per lo meno lo erano l'ultima volta che è stato fatto l'aggiornamento).

I procedimenti potrebbero essere mostrati sotto forma di segnalibri (bookmark). Selezionandone uno il cittadino sarà portato verso una pagina di interrogazione attraverso la quale potrà chiedere di visualizzare in dettaglio gli stati, di ritrovare i documenti sottomessi o di visualizzare eventuale documentazione prodotta dalle amministrazioni e associata alla richiesta.

Com'è intuibile a valle di questa interrogazione dovranno essere disponibili opportuni servizi standardizzati di BSL in grado di interrogare i propri sistemi di protocollazione e document-management.

Allo stesso modo è intuibile che il semplice possesso del codice di procedimento non è sufficiente per dare l'accesso alle informazioni, sarà comunque responsabilità dei BSL decidere se visualizzarle o meno in base al Codice Fiscale del cittadino.

3.4 Caratteristiche della Infrastruttura di Gestione

Come si è accennato nella sezione introduttiva relativa alla Infrastruttura di Gestione (2.5.2), perché un sistema complesso come quello di PEOPLE possa rispondere in maniera adeguata ai requisiti di affidabilità e disponibilità richiesti, è fondamentale la creazione di una infrastruttura di gestione non semplicemente sistemistica, ma anche applicativa, in particolare devono essere gestite le funzionalità di:

1. Registrazione delle applicazioni
2. Configurazione delle applicazioni
3. Tracciatura dei processi
4. Tracciatura degli eventi (economici) contabilizzabili
5. Tracciatura dei problemi
6. Raccolta di informazioni relative alle prestazioni

Il rapporto tra le applicazioni (componenti applicativi presenti a livello di FSL, VSL e, in alcuni casi, BSL) e l'infrastruttura di gestione è di tipo server-server, o se si vuole di client-server bidirezionale. Sia l'infrastruttura che le applicazioni devono cioè essere in grado di esporre delle interfacce a servizi (server) e di fruire dei servizi messi a disposizione da altri (client).

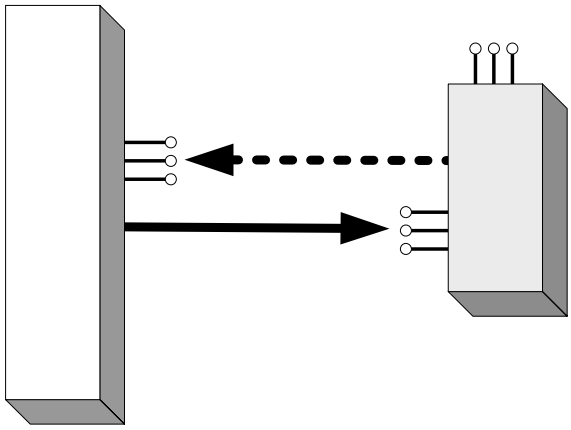
Questo significa che nel momento in cui un nuovo modulo è aggiunto nei diversi strati applicativi deve registrarsi presso un servizio offerto in maniera strandard dall'infrastruttura di gestione (well known service), informandolo della propria classe di applicazione, dei messaggi di configurazione in grado di ricevere, degli eventi in grado di generare e di quali eventi (generati dal sistema o da altre applicazioni) sia interessati a ricevere.

Questa registrazione dinamica è necessaria per poter assicurare un funzionamento evolutivo e non-stop dell'intero sistema.

Una volta che un modulo si è registrato, il sistema di gestione potrà interagire con esso inviandogli messaggi per avviarlo o interromperlo, impostare parametri di configurazione (per esempio il livello di log dell'intero modulo o di eventuali specifici sottositemi), aggiungere utenti, comuni o quant'altro sia necessario.

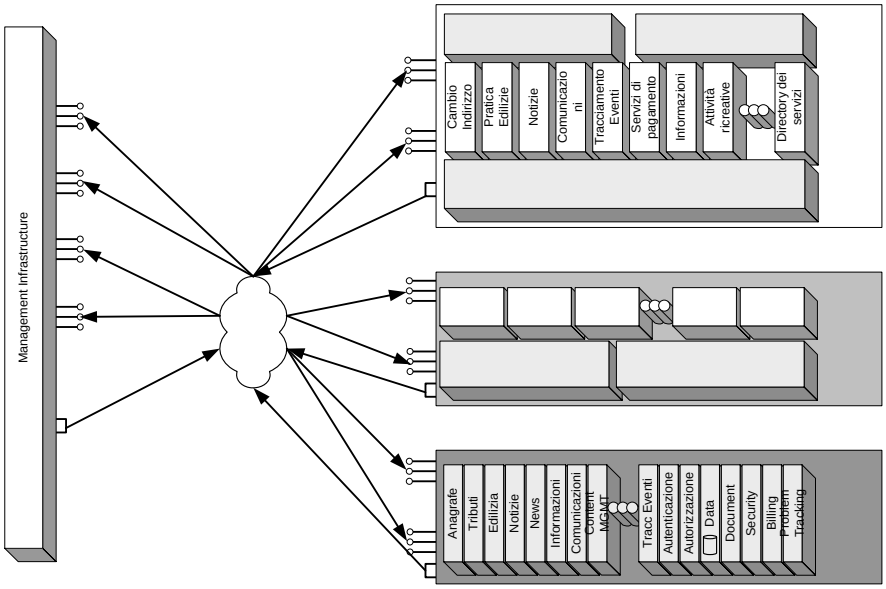
Il modulo, una volta avviato dal sistema di gestione, potrà iniziare ad utilizzare le interfacce offerte dall'infrastruttura per generare informazioni di consumo, errori o tracciatura degli eventi.

Quando l'infrastruttura di gestione rileverà la presenza di eventi a cui il modulo si è sottoscritto, glieli comunicherà utilizzando l'apposità interfaccia esposta dal modulo stesso.



3.4.1 Modalità di interazione tra Infrastruttura di Gestione e i Service Layers

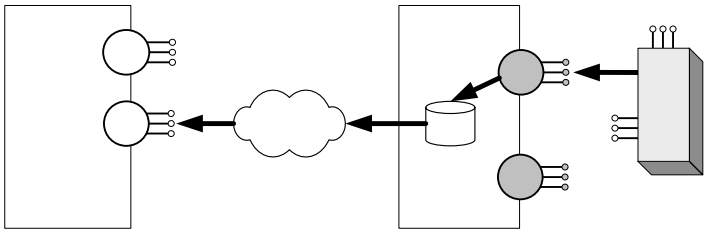
Onde assicurare l'indipendenza tra le tecnologie utilizzate per l'implementazione dell'infrastruttura di gestione e quelle utilizzate per i diversi strati applicativi (FSL, VSL e BSL) l'interazione dovrebbe avvenire attraverso l'utilizzo di WebServices. In questo modo si superebbe anche la necessità di co-località dei servizi, potendo quindi ipotizzare un'infrastruttura di gestione che gestisca in outsourcing il monitoraggio e l'intervento a livello logico di sistemi implementati in comuni differenti.



Mentre la scelta di implementare un'interfaccia a WebServices tra i Service Layers e l'infrastruttura di gestione ha indubbi vantaggi, porta con sè tuttavia anche dei potenziali significativi svantaggi in termini di affidabilità e prestazioni. Perchè infatti il colloquio tra i servizi possa avvenire, tutti gli strati intermedi tra l'applicazione e il servizio (Rete, interfacce di rete, stack TCP/IP, stack HTTP ecc.) devono funzionare correttamente, inoltre le operazioni di impacchettamento (marshaling), tramissione e spacchettamento (unmarshaling) dei messaggi possono avere un impatto significativo sul numero di messaggi trasferibili al secondo, rendendo operazioni quali il tracing di un'applicazione praticamente inattuabile.

Per questa ragione è corretto ipotizzare che le vere API (interfacce di programmazione) messe a disposizione dei moduli applicativi non siano direttamente quelle offerte dai Servizi Web

dell'infrastruttura, quanto un insieme di API native all'ambiente (implementate cioè coerentemente con i linguaggi e le tecnologie con coi vengono implementati i Service Layers) che, agendo da Proxy verso le interfacce dell'infrastruttura di gestione, si prendano cura di mascherare problematiche quali la co-località, la necessità di memorizzare o accodare localmente le informazioni per inviarle a blocchi (bulk messaging) nel momento in cui la rete non fosse disponibile o il servizio remoto smaltisse i messaggi remoti in maniera non sufficientemente efficace.



Ovviamente a livello dell'infrastruttura di gestione dovranno essere presenti servizi in grado sia di ricevere messaggi semplici che composti.

Questa modalità implementativa offre anche il vantaggio di poter meglio integrarsi con i meccanismi nativi di gestione degli errori/eccezioni esistenti negli specifici ambienti, riducendone i tempi di apprendimento delle interfacce e la possibilità di errori.

3.4.2 Registrazione delle applicazioni

Come indicato precedentemente il meccanismo di registrazione delle applicazioni presso l'infrastruttura di gestione dovrà seguire un meccanismo dinamico "attraverso la rete". Nella realtà dei fatti (come precedentemente indicato) è lo specifico FSL, VSL o BSL che funge da intermediario tra le interfacce native e l'infrastruttura di gestione, perciò sarà responsabilità dei primi assicurarsi che l'applicazione sia autorizzata ad inviare eventi e a ricevere comandi. Lo scambio di messaggi tra i Service Layers e l'infrastruttura di gestione potrebbe essere crittografato o semplicemente firmato, in base alla criticità e riservatezza delle informazioni scambiate (cosa particolarmente importante nel caso di informazioni di consumo che potrebbero dare origine a bollettazione), utilizzando chiavi simmetriche scambiate in maniera periodica tra le parti.

Come detto durante il precesso di registrazioni l'applicazione fornirà informazioni relative alla propria classe di appartenenza, ad eventuali estensioni rispetto a un set di classi di base, in termini di attributi e comandi supportati, oltre a indicazione di quali eventi potrebbero essere di interesse all'applicazione stessa.

Il meccanismo di estensione utilizzerà la natura autodescrivente dei messaggi XML per ridurre al minimo la tipologia delle interfacce utilizzate.

3.4.3 Servizi esposti dai moduli applicativi

I servizi messi a disposizione dalle applicazione potranno essere utilizzati dall'infrastruttura di gestione per un insieme di operazioni, definiti al minimo come:

1. Start e Stop del servizio
2. Interfacce per il provisioning (vedi in seguito)
3. Abilitazione della modalità di tracing per l'intera applicazione o uno dei componenti della stessa
4. Abilitazione della modalità di logging e definizione del livello
5. Impostazione di specifici attributi esposti dall'applicazione

6. Recupero di informazioni statistiche (es. numero di invocazioni del servizio nelle 24h, numero di utenti che hanno utilizzato il servizio dallo start, data e ora di start ecc.)
7. Esecuzione di uno dei comandi messi a disposizione dall'applicazione e descritti a livello di registrazione
8. Delivery di un evento sottoscritto

3.4.4 Servizi esposti dall'Infrastruttura

Allo stesso modo l'infrastruttura metterà a disposizione un insieme minimo di interfacce per:

1. Registrare le applicazioni
2. Interfacce per il Provisioning
3. Interfacce per la Mediation
4. Ricevere eventi generici
5. Ricevere messaggi di errore
6. Ricevere messaggi di tracing
7. Ricevere messaggi di log
8. Ricevere messaggi di consumo
9. Ricevere messaggi periodici di carico e di stato

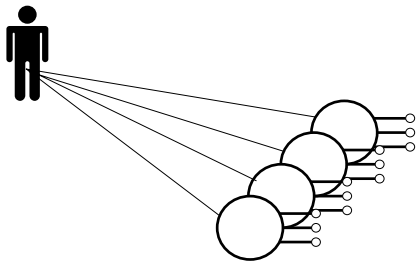
E' responsabilità dell'infrastruttura di gestione definire come organizzare i messaggi in arrivo dalle applicazioni e come eventualmente interfacciarsi tramite protocolli standard (quali SNMP) con strumenti di mercato per il monitoring delle infrastrutture.

3.4.5 Mediation

I dati memorizzati dall'infrastruttura di gestione potranno essere utilizzati da diverse applicazioni, sia per generare report che per specifiche operatività. In particolare le informazioni relative all'utilizzo di specifici servizi (dati di consumo) potranno essere utilizzati da sistemi di mediation esterni per dare origine a bollettazione o al calcolo dei tributi dovuti (per esempio bolli), anche solo quando questo serva per un puro riscontro amministrativo. A questo scopo l'infrastruttura di gestione dovrà definire un insieme di interfacce che permettano l'interrogazione dei dati di consumo, garantendo la riservatezza degli stessi e l'accesso da parte di terzi (comuni) solo per la porzione a loro pertinente.

3.4.6 Provisioning

Come indicato precedentemente, uno degli obiettivi dell'architettura di PEOPLE è quello di permettere l'evoluzione dinamica dei servizi; perchè questo possa avvenire si devono indirizzare due aspetti, uno legato all'inserimento di nuovi servizi applicativi all'interno dell'infrastruttura (installazione e attivazione), l'altro legato al rapporto tra gli utenti e i servizi. Questa ultimo è comunemente indirizzato dai sistemi di provisionig. Tramite questo sistema, che l'Infrastruttura di Gestione dovrebbe a sua volta esporre come servizio, gli amministratori di sistema o applicazioni autorizzate in carico ai comuni, dovrebbero essere in grado di configurare associazioni tra utenti (identificati dal Codice Fiscale/Identificativo) e i servizi (sottoscrizioni), inoltre, per ciascun servizio a cui l'utente è "sottoscritto" dovrebbe essere possibile anche definire uno o più ruoli autorizzati.



Questo aspetto è importante non solo per gli aspetti legati alla sicurezza, ma anche per gli aspetti di "contabilizzazione" del servizio, in quanto, in base al ruolo, i componenti applicativi possono decidere o meno di erogare il servizio in maniera completa o parziale, di contabilizzare o meno i consumi e così via.

Un altro aspetto che deve essere indirizzato dal sistema di Provisioning è quello delle interdipendenze tra componenti. Spesso i servizi sono di tipo "composto", per cui l'attivazione di un'utenza in un servizio deve essere "propagata" ad altri servizi da cui questo dipende. Il sistema di provisioning dovrebbe essere in grado di permettere la definizione di tali dipendenze, automatizzando il processo di attivazione.

Chiaramente sia le applicazioni che l'infrastruttura dovranno esporre delle interfacce standardizzate che permettano da un lato la descrizione e l'interrogazione delle relazioni tra utenti e servizi, e tra servizi e servizi, e dall'altro l'attivazione degli stessi.

Come accennato, per quanto molte attività del servizio di provisioning abbiano relazione con gli aspetti di sicurezza, non ci si aspetta che le applicazioni interroghino in tempo reale il sistema di provisioning per ottenere informazioni sugli accessi, al contrario l'infrastruttura di sicurezza è uno dei servizi dipendenti interfacciati dal sistema di provisioning, nel momento in cui un utente viene configurato in un'applicazione, sarà responsabilità dell'applicazione che configura le dipendenze prendersi cura anche della configurazione del sistema di sicurezza e impostare gli attributi di accesso per quell'utente in maniera adeguata.

3.5 Modello Workflow e Event Manager per VSL

Nel capitolo relative al VSL (2.3) si è visto come i servizi "atomici" esposti dal BSL possono essere aggregati all'interno di procedimenti (servizi virtuali) che coinvolgono uffici diversi all'interno dello stesso comune, amministrazioni e sistemi informativi diversi, dando ai cittadini una visione "integrata" di una realtà frammentata dal punto di vista amministrativo.

E' corretto attendersi che i processi automatizzati a questo livello abbiano una granularità piuttosto ampia, non vadano cioè a sostituire i processi amministrativi presenti all'interno dei back-end, ma fungano a tutti gli effetti da "ponte" tra i confini di attività amministrative associabili, in maniera lasca, a uffici o amministrazioni diverse, dando vita ad un passaggio di informazioni automatizzato, più che ad una vera e propria re-implementazione dei processi di back-end.

Per creare questa visione in maniera affidabile e scalabile all'interno di un'architettura a servizi quale quella di PEOPLE sono sono stati individuati due componenti fondamentali, il primo, denominato Back-end Workfllow Subsystem ha lo scopo di automatizzare i diversi "passi" di un procedimento, il secondo, Event Manager, la gestione della comunicazione affidabile e asincrona anche in modalità publish and subscribe.

3.5.1 Il Back-end Workflow Subsystem

Tradizionalmente le soluzioni per l'integrazione di applicazioni distribuite sono state di tipo proprietario e molto costose, adeguate alla fascia alta del mercato dell'integrazione. Al contrario PEOPLE ha bisogno di costruire un sistema distribuito basato su un'achitettura a servizi accoppiati in maniera lasca (loosely-coupled service-oriented architecture), in modo da trarre vantaggio dall'efficenza indotta dall'automatizzazione dei processi minimizzando i costi, i tempi e le risorse necessarie a costruire e manutenere un sistema integrato di gestione degli eventi della vita del cittadino.

Mentre il workflow del FSL e quello eventualmente presente a livello di BSL possono fare affidamento su un modello di programmazione sincrono di tipo request-reply, i servizi esposti dal Back-end, e automatizzati a livello di VSL, sono fondamentalmente dei Servizi Web (Web Services) di tipo asincrono che introducono un modello di programmazione di tipo conversazionale, basato sulla interazione asincrona tra Web Services accoppiati in maniera lasca. Questa attività di controllo del workflow tra componenti viene spesso definita orchestrazione o coreografia, proprio per indicare che i servizi da automatizzare sono essi stessi autonomi, con propri stati e comportamenti che non vengono gestiti dal workflow, ma che sono semplicemente attivati da questo o che con questo interagiscono.

La logica da implementare del processi di VSL può tuttavia essere anche piuttosto articolata e andare da una semplice conversazione a due a una transazione complessa, multi-step e non lineare.

Nella scelta di un formalismo per la descrizione di questi processi è fondamentale poter usare un modello che oltre ad essere condiviso nell'industria (se non adirittura standard) permetta di essere eseguibile, così da poter essere direttamente utilizzato per l'implementazione del workflow indipendentemente dalla tecnologica utilizzata.

In questo contesto è di sicuro interesse il ruolo che il Business Process Execution Language for Web Services (BPEL4WS o BPEL) sta assumendo nell'industria.

Anche se non ufficialmente uno standard de jure, in realtà sta diventando rapidamente la specifica principale per la standardizzazione a livello di logica e automatizzazione di processi tra Web Services. Il BPEL è stato inizialmente creato dalla collaborazione tra aziende quali IBM, BEA, Microsoft, SAP, Sieble e ha rapidamente guadagnato il supporto di altri importanti produttori (Oracle, SUN, Tibco, Vitria, See Beyond e altri) fino alla sottomissione a OASIS (Organization for the Advancement of Structured Information Standards) in aprile 2003, con lo scopo di ottenere un'accettazione ancora più ampia e la standardizzazione attraverso un processo aperto.

Il supporto da parte dei principali attori dell'informatica mondiale ha fatto sì che molti analisti abbiano già identificato nel BPEL lo standard indisputato che ha soppiantato a buon diritto altre attività in corso di standardizzazione.

Le specifiche del BPEL definiscono la sintassi e la semantica del linguaggio, che contiene un'ampia scelta di costrutti per la gestione del processo. E' possibile prendere decisioni di flusso, avviare processi paralleli, gestire sottoprocessi, effettuare il join di più processi e così via.

Il linguaggio rappresenta una convergenza tra il Web Service Flow Language (WSFL) di IBM e l'XLANG di Microsoft e, come caratteristico di altri linguaggi e specifiche nel mondo dei Web Services, è definito in XML.

Il BPEL4WS è un linguaggio con una grammatica basata su XML di tipo "eseguibile", può essere, cioè, direttamente interpretato da un'applicazione di workflow (ad esempio il BPWSAJ di IBM, scaricabile gratuitamente dal sito http://alphaworks.ibm.com/tech/bpws4j, o in implementazioni commerciali vericali quali quelle di Collabra o di Choreo, o ancora in prodotti di EAI quali quelli di Vitria, See Beyond e altri).

BPEL4WS utilizza in maniera importante WSDL: ogni processo di BPEL è esposto come un servizio Web utilizzando WSDL, i tipi di dati di WSDL sono utilizzati per descriver lo scambio

di informazioni anche all'interno del processo (in modalità simile a quella delle variabili) ed infine il WSDL è utilizzato per interagire con i servizi esterni.

BPEL è poi compatibilie con le specifiche WS-Coordination e WS-Transaction.

Le attività utilizzate in BPEL per descrivere i processi sono di due tipi: primitive e strutturali.

Le attività primitive sono di basso livello e rappresentano il "vero" lavoro fatto dal processo:

- <invoke> - invoca un'operazione su un Web Service
- <receive> - rimane in attesa di una richiesta
- <reply> - genera una risposta
- <wait> - aspetta una determinata quantità di tempo
- <assign> - copia dei valori da un contenitore ad un altro
- <throw> - solleva un'eccezione
- <catch> - cattura un'eccezione
- <terminate> - interrompe l'istanza del processo
- <empty> - esegue un'azione nulla

Le attività strutturali vengono invece utilizzate per combinare le attività primitive e organizzare il flusso di controllo:

- <sequence> - esecuzione sequenziale di attività primitive
- <flow> - esecuzione parallela di attività primitive
- <switch> - seleziona un'attività all'interno di un certo set
- <while> - definisce una iterazione (loop)
- <pick> - seleziona uno tra i diversi percorsi disponibili
- <scope> - raggruppa un insieme di attività all'interno di una singola transazione

L'utilizzo del BPEL nella descrizione degli eventi della vita (i processi di VSL) e l'utilizzo di un motore di esecuzione in grado di utilizzare il BPEL (o di convertirlo in un proprio linguaggio interno con esso compatibile) può semplificare notevolmente la descrizione dei processi e la loro portabilità.

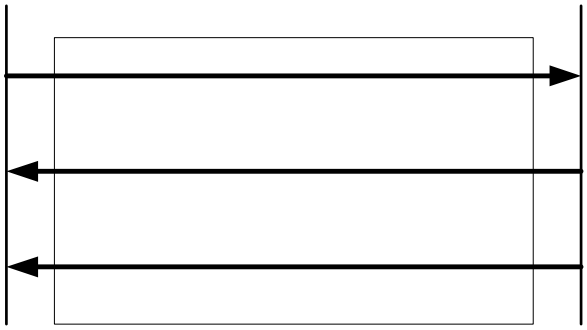
3.5.2 Interazione del Back-end Workflow Subsystem e Event Manager

Come indicato nel paragrafo precedente ogni servizio esposto dal Workflow subsystem viene esposto dal "motore" di esecuzione del BPEL sottoforma di Servizio Web, questa è la modalità ideale di interazione tra il FSL e il VSL per attivare i procedimenti e tra il BSL e VSL per gli avanzamenti. Nel modello del BPEL le entità con cui l'infrastruttura comunica sono definiti "partner" e rispondono a tre tipologie fondamentali:

- Invoked Partners – Altri servizi Web che sono richiamati dal processo
- Client Partners – Applicazioni client che richiamano servizi Web esposti dal processo
- Service Partners – Applicazioni che sono in grado sia di richiamare servizi del processo (client) che di essere richiamati da questo (server)

Nella maggior parte dei casi le applicazioni presenti a livello del FSL sono fondamentalmente dei client, in quanto l'interazione è principalmente attivata da un attore umano che dà inizio ad un procedimento o che ne viene a verificare l'avanzamento. Nel caso invece del BSL possiamo

sicuramente parlare di applicazioni che agiscono da Service, in quanto nella maggio parte dei casi sono attivate dal VSL (Invoked partner) ma spesso forniscono rispose e risultati solo in maniera asicrona a distanza di ore o giorni.



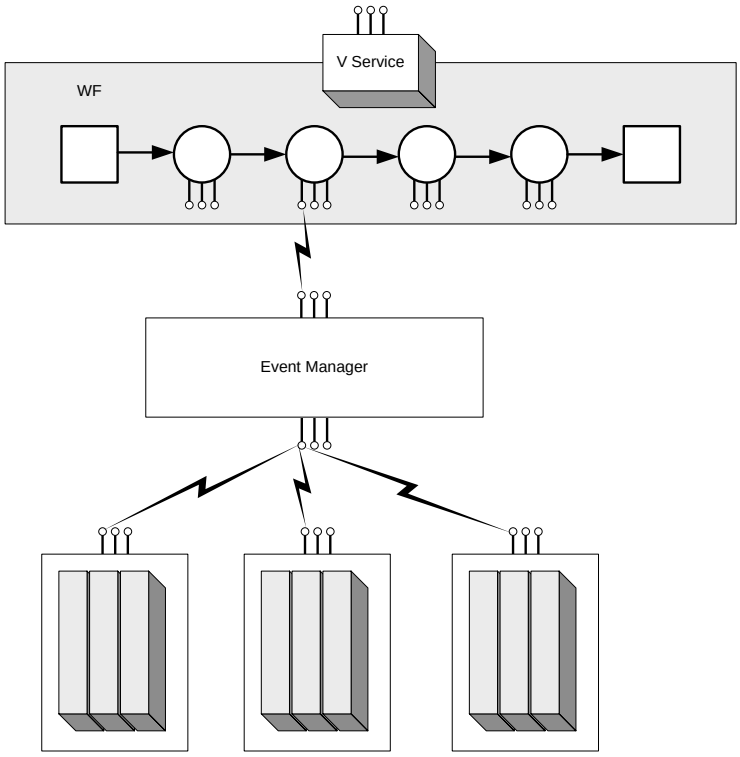
Questo modello asincrono di conversazione, perfettamente supportato dal BPEL non fa alcuna assunzione sul trasporto, sul modo cioè con cui i messaggi vengono recapitati alle parti coinvolte, quest'ultimo può infatti essere implementato utilizzando sia trasporti sincroni quali l'HTTP che asincroni quali JMS, SMTP o protocolli di queing proprietari.

Nel caso di PEOPLE, è necessario considerare che, data la diversa tipologia e l'ampia distribuzione degli enti coinvolti, è quanto meno velleitario assumere la continua disponibilità dei servizi di back-end e, di conseguenza, per poter ottenere affidabilità, scalabilità e flessibilità, le interazioni tra i Web Services devono essere implementate utilizzando un meccanismo in grado di gestire messaggi in maniera asincrona.

Inoltre le caratteristiche del linguaggio BPEL sono orientate alla pura automazione dei processi di workflow tra servizi Web, in cui ogni singola attività di <invoke> o di <receive> ha uno e un solo partner invocato o che invoca. Questo non significa che un'attività non possa iterare tra più partner e inviare più messaggi, ma solo che ogni singolo messaggio va ad un singolo partner. Concetti quali quello di publish-and-subscribe o di multicasting sono estranei al BPEL che, così come il trasporto, possono essere considerati accessori o ortogonali alla descrizione del servizio di workflow.

Tuttavia una gestione di notifica a più entità che hanno registrato un interesse in un determinato evento è sicuramente importante, così come la possibilità di esternalizzare la decisione del trasporto e dell'end point (il comune a cui richiedere il servizio) alla specifica attività. Il BPEL ha la capacità di prendere decisioni in base al contenuto dei messaggi e di fare una sorta di content-based-routing, tuttavia le correlazioni devono essere espresse all'interno di un documento WSDL e come tale associate allo specifico workflow.

Per questa ragione l'architettura di PEOPLE ipotizza la presenza di un Event Manager che affianchi il Back-end Workflow nella virtualizzazione del trasporto, della modalità di interazione e del routing, rendendo i processi indipedendenti dalla topologia e peculiarità dell'installazione.



L'Event Manager dovrà quindi fungere da trasporto intelligente tra il VSL e il BSL (raramente anche con il FSL) prendendosi carico di complementare quest'ultimo in tutta una serie di operazioni e decisioni che poco hanno a che fare con l'automatizzazione del processo: per esempio rendendo trasparente il numero di messaggi inviati per ottenere una risposta, come nel caso di una ricerca distribuita su più Comuni, o nella conversione tra versioni diverse di messaggi.