PEOPLE
INNOVARE NELLA PUBBLICA AMMINISTRAZIONE

Ministro per l'Innovazione e le Tecnologie
Piano Nazionale di e-Government

Architettura di Riferimento

PEOPLE 1.0

Draft

Versione documento

Modifiche apportate	Data
Aggiunto capitolo 4.4	3/11/2004

Indice

Indice 0
1 Introduzione 0
2 Reference Implementation 0
3 Architettura generale PEOPLE 1.0 4
3.1 I nodi di FE 4
3.2 Il nodo PEOPLE 4
3.3 VSL 4
3.4 I nodi di BE 4
4 Architettura di dettaglio PEOPLE 1.0 4
4.1 Scenario Setup 4
4.1.1 Istallazione FE 4
4.1.2 Istallazione Nodo People 4
4.1.3 Istallazione BE 4
4.1.4 Configurazione BE 4
4.1.5 Configurazione FE 4
4.2 Scenario Administration 4
4.3 Scenario Runtime 4
4.4 Integrazione con il BE 4
5 Deployment di PEOPLE 1.0 4
5.1 Sistema distribuito 4
5.2 Sistema multipiattaforma 4

1 Introduzione

Questo documento descrive l'architettura generale del progetto PEOPLE 1.0, allo scopo di costituire un riferimento per le attività di realizzazione delle versioni commerciali del framework e dei servizi applicativi ed infrastrutturali.

Questo documento costituisce inoltre la base per la realizzazione dell'implementazione di riferimento del framework 1.0, che consentirà di verificare i principi e le tecniche descritte e verrà utilizzato come strumento per le previste attività di collaudo e certificazione dei servizi e dei framework commerciali.

2 Reference Implementation

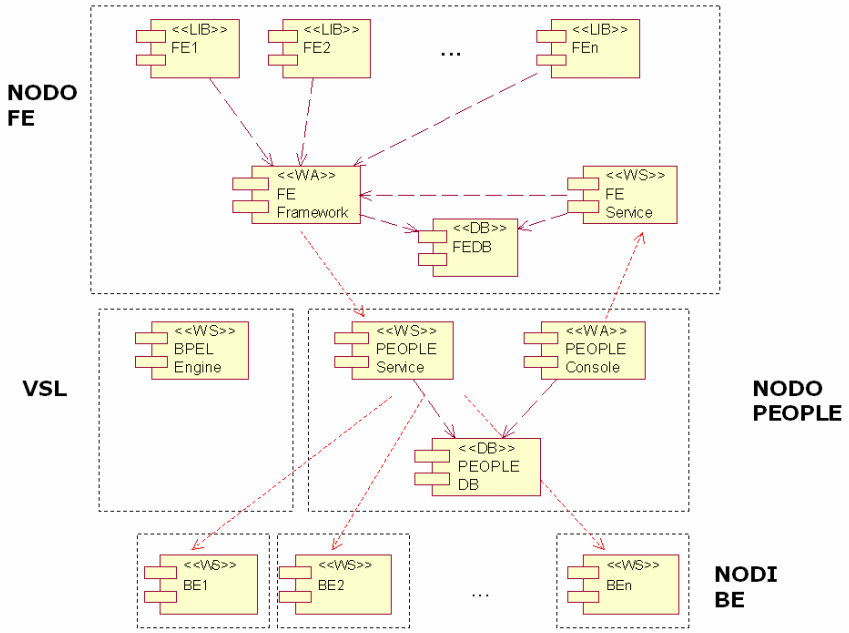
L'implementazione della RI avverrà a partire dal framework 1.0b, componente già realizzato e che implementa in modo più o meno completo molte funzionalità previste per il framework vero e proprio.

Come descritto nei capitoli successivi, il framework 1.0b corrisponde in larga misura ad un preciso componente (il FE Framework) della nuova architettura, e sarà pertanto incorporato mediante refactoring successivi.

Allo scopo di consentire ai vari attori interessati (sviluppatori di servizi e dei framework commerciali) di interagire il più rapidamente possibile con la nuova architettura, la RI verrà rilasciata in varie versioni preliminari successive, definite ogni qual volta si rendano disponibili adeguate funzionalità aggiuntive o modificate.

Con questo approccio iterativo la realizzazione della RI 1.0 avverrà mediante una serie di Preliminary Release, la prima delle quali sarà rilasciata il giorno 22 ottobre 2004.

3 Architettura generale PEOPLE 1.0



La figura mostra l'architettura generale del progetto PEOPLE.

Ogni istallazione PEOPLE è composta da una combinazione di tre differenti tipologie di nodi: i nodi di Front End (FE), il nodo PEOPLE vero e proprio (in cui si trova il kernel del framework), il VSL, i nodi di Back End (BE).

3.1 I nodi di FE

Un nodo di FE è costituito da un insieme di servizi web esposti da un Comune tramite il progetto PEOPLE.

In questo senso, l'identificativo del Comune può essere utilizzato per identificare ciascun singolo nodo di FE.

Un sistema PEOPLE può contare più nodi di FE che vengono gestiti in modo centralizzato, consentendo così di realizzare un servizio in modalità ASP.

Ciascun nodo FE è costituito da tre componenti infrastrutturali, il **FE Framework**, il **FE Service** ed il **FE DB**, e da tante librerie quanti sono i servizi istallati.

Il FE Framework è un insieme di librerie di supporto per lo sviluppo dei servizi e coincide più o meno con quanto finora noto come *Framework 1.0a*.

E' costituito principalmente da un sistema di workflow che consente di realizzare un servizio PEOPLE specificando i vari step che lo compongono mediante opportune classi e configurazioni.

Ciascun servizio viene sviluppato utilizzando il pattern MVC (vedere il documento "Guida allo sviluppo di un servizio") e viene esposto tramite l'unica web application fornita dal FE Framework.

Ogni servizio di FE necessita per il suo corretto funzionamento di uno o più servizi di back office, sviluppati ad hoc per il particolare Comune che eroga il servizio sulla base di un'interfaccia standard definita dal framework PEOPLE.

Il FE Service è un componente necessario alle interazioni con il kernel del framework, ed espone una serie di web method che vengono invocati dalla PEOPLE Console e mediante la quale il nodo PEOPLE riesce a controllare il singolo servizio.

Il FE DB è il database relativo al nodo di FE. Contiene la configurazione del nodo e dei singoli servizi, ed è condiviso fra FE Service e FE Framwork.

Il nodo FE necessita per il suo corretto funzionamento del web server Tomcat e di un database MySQL, e può essere istallato su server remoti e anche su piattaforme diverse rispetto al nodo PEOPLE e ai suoi servizi di back office.

3.2 Il nodo PEOPLE

Il nodo PEOPLE è il kernel di un'istallazione PEOPLE, e costituisce il centro da cui è possibile gestire l'intero sistema.

Tutti i nodi di FE e BE che costituiscono un sistema PEOPLE devono essere registrati e configurati opportunamente su un nodo PEOPLE perchè possano funzionare correttamente e comunicare tra loro.

Il nodo PEOPLE comunica con tutti gli altri nodi esclusivamente tramite web service, il che garantisce la possibilità di istallare su una piattaforma remota e basata su un sistema operativo diverso senza particolari problemi di interazione.

I componenti che costituiscono un nodo PEOPLE sono i seguenti:

- PEOPLE Console
- PEOPLE Service

La PEOPLE Console è un'applicazione web di amministrazione del sistema, dalla quale è possibile controllare i servizi istallati, configurarli, gestire la mappatura fra i servizi di FE ed i corrispondenti servizi di back office.

Il PEOPLE Service è il componente server del nodo PEOPLE.

Riunisce diverse funzioni infrastrutturali, quali servizi di gestione (es. log(), trace(), …) e di interoperabilità (comunicazione FE-BE).

3.3 VSL

Il Virtual Service Layer è costituito dal componente che contiene la logica di orchestrazione del servizio, e sarà implementato mediante un BPEL Engine.

Il BPEL Engine è un componente di web service orchestration verso il quale i servizi di comunicazione potrebbero dirigere alcune opportune richieste.

Tale componente sarebbe realizzato mediante l'utilizzo di un apposito engine, ma al momento le funzionalità di orchestration non sono ancora state analizzate in modo sufficiente e pertanto la versione 1.0 del framework non conterrà tale componente.

3.4 I nodi di BE

I nodi di BE costituiscono lo strato di integrazione verso i sistemi legacy dei singoli comuni.

Ciascun nodo sarà costituito da un singolo web service che esporrà sostanzialmente un unico web method:

string process(string)

Il metodo accetta come parametro di input un unica stringa in formato XML con tutte le informazioni necessarie all'esecuzione del suo lavoro, e restituisce in output una stringa XML con l'eventuale risposta.

Lo schema del XML sarà definito in collaborazione con il gruppo di modellazione da parte del fornitore di quella tipologia di servizi.

Ciascun nodo sarà realizzato *ad hoc* e sviluppato sulle tecnologie più adatte a semplificare l'integrazione con i sistemi di back office preesistenti

Proprio per questo si è optato per un interfaccia minimale (solo un metodo) sacrificando le possibilità di controllo da parte della PEOPLE Console a favore della semplicità e rapidità di realizzazione.

Poichè per molti casi di servizi di FE sarebbe sufficiente realizzare il back office con una semplice mail verso un funzionario del comune, tale funzionalità sarà realizzata nel kernel stesso e non coinvolgerà nodi di BE veri e propri.

4 Architettura di dettaglio PEOPLE 1.0

Questo capitolo descrive con maggior dettaglio il design dei componenti di PEOPLE 1.0.

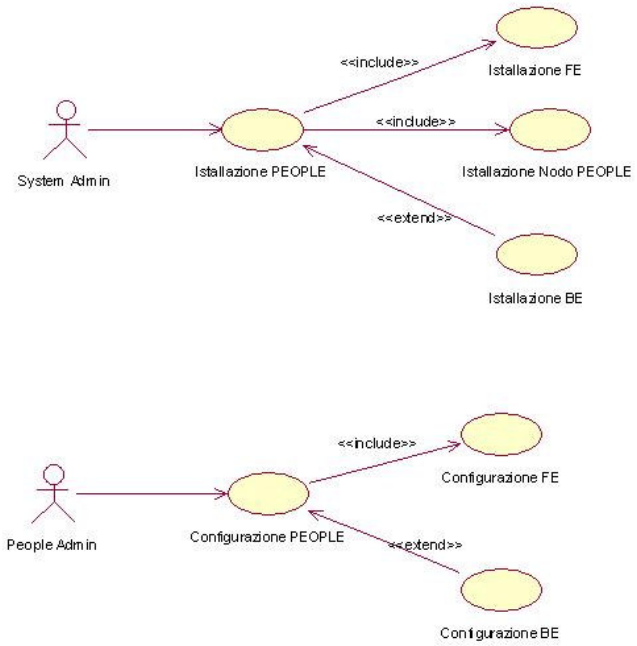
Come già accennato si è data priorità alla costituizione rapida dell'architettura di base del framework, più che alla completezza delle funzionalità.

Pertanto l'architettura che segue potrà essere in seguito migliorata nella quantità e nella qualità delle singole funzioni, pur mantenendo fermi gli elementi di base che la compongono.

Per maggior chiarezza espositiva, in questo capitolo si è ritenuto di descrivere il sistema con un approccio di tipo Use Case based.

Sono stati individuati tre scenari operativi per il framework PEOPLE, e ciascuno sarà oggetto di uno dei paragrafi successivi.

4.1 Scenario Setup



Il diagramma mostra lo scenario relativo ai processi di istallazione e configurazione del sistema.

Le due fasi hanno un significato distinto: l'istallazione è un processo sistemistico e consiste sostanzialmente nell'istallare e configurare i componenti software di base (Tomcat, MySQL), il framework ed i vari servizi.

Una volta completata la fase di istallazione tutte le attività sistemistiche sono terminate, ma il sistema non sarà ancora utilizzabile fino alla sua fase di configurazione.

La fase di configurazione è costituita da una serie di operazioni che un amministratore PEOPLE dovrà espletare sulla PEOPLE Console così da registrare opportunamente tutti i componenti del sistema, rendendoli così in grado di cooperare fra loro.

Esamineremo ora in dettaglio ciascun Use Case.

4.1.1 Istallazione FE

L'istallazione di un nodo di FE comprende i componenti FE Framework, FE Service e FE DB, più attività specifiche relative a ciascun servizio FE.

Si tratta di copiare il codice e configurare i file config.xml relativi a ciascun componente.

I file config.xml contengono informazioni necessarie alla corretta registrazione dei componenti presso il nodo PEOPLE, e vengono utilizzati solo all'atto della configurazione, dato che in seguito tutti i componenti utilizzano le impostazioni provenienti dalla PEOPLE Console.

4.1.2 Istallazione Nodo People

L'istallazione del nodo PEOPLE comprende i componenti PEOPLE Console, PEOPLE Service e PEOPLE DB.

Si tratta di copiare il codice e configurare i file config.xml relativi a ciascun componente.

4.1.3 Istallazione BE

L'istallazione del BE implica l'istallazione di un certo numero di componenti con scopi specifici, che dovranno interagire con il resto del sistema tramite web service.

Relativamente ai componenti di BE, il framework fornisce la sola interfaccia WSDL da implementare.

4.1.4 Configurazione BE

Una volta completate tutte le procedure di istallazione si passa alla fase di configurazione del sistema.

Per prima cosa occorre registrare ciascun componente di BE presso il nodo PEOPLE utilizzando la Console.

La registrazione avviene abbinando un nome logico all'URL del servizio di BE.

Al momento non è prevista per la versione 1.0 un'interazione fra PEOPLE Console e servizio di BE nella fase di configurazione, ma in futuro si potrebbe esaminare almeno la possibilità di effettuare un check di verifica dell'URL inserito.

In questa prima fase si è preferito alleggerire al massimo gli oneri per lo sviluppo del BE, e pertanto si suggerisce di verificare l'URL invocando il servizio con un normale browser.

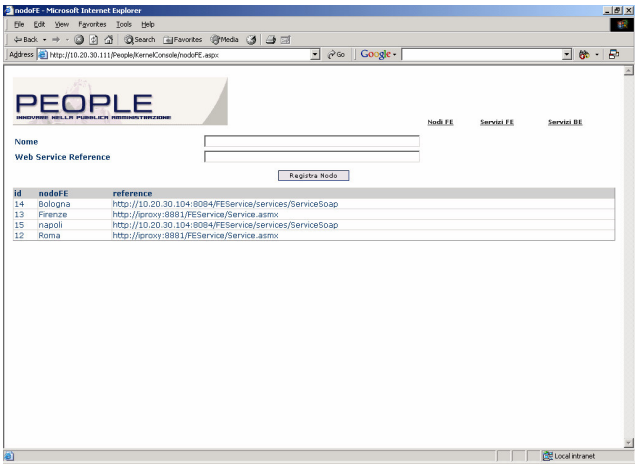
4.1.5 Configurazione FE

La configurazione del FE avviene eseguendo una successione di passi:

- registrazione del nodo di FE sulla Console
- registrazione di ciascun servizio
- configurazione di ciascun servizio di FE
- configurazione dei mapping necessari per ciascun servizio di FE

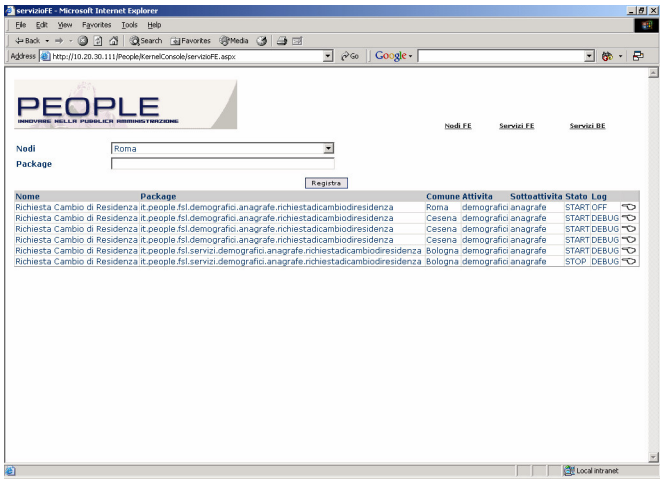
La **registrazione del nodo di FE sulla Console** avviene abbinando un nome logico (tipicamente il nome del Comune) all'URL del FE Service del nodo.

La figura mostra la schermata relativa.



Contestualmente la Console invia al nodo di FE l'URL del proprio PEOPLE Service, così che possa essere invocato quando necessario.

La fase successiva consiste nella registrazione di ciascun servizio presente sul nodo di FE, fornendo la coppia comune-package mediante la pagina mostrata in figura.



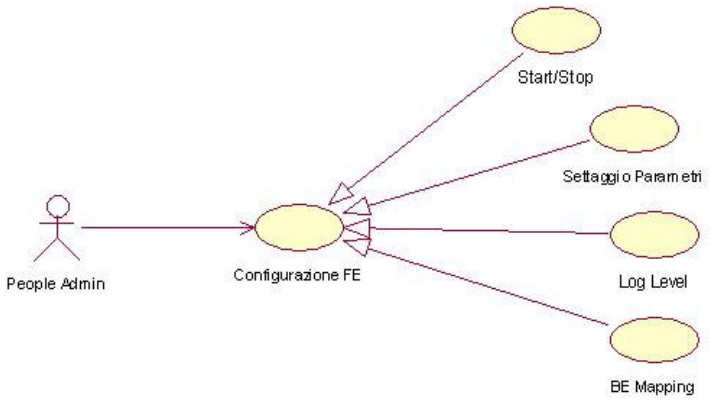
All'atto della registrazione il FE invia alla Console la configurazione del servizio registrato prelevandola dal relativo file config.xml.

Le informazioni comprendono dati sul servizio (attività, sottoattività) sulla sua attuale configurazione (log, stato, parametri operativi) e sulle sue dipendenze verso servizi di BE.

Vedremo nei capitoli successivi (Administration e Runtime) come le dipendenze svolgano un ruolo fondamentale nell'assicurare l'integrità del sistema stesso.

In futuro potrà essere prevista la possibilità di registrare più servizi in un unico momento, ad esempio fornendo al sistema un file di configurazione con i dati necessari.

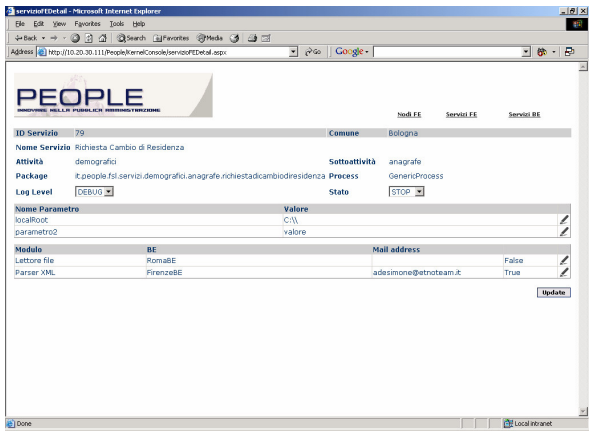
4.2 Scenario Administration



Lo scenario amministrazione consiste nell'esercizio delle funzionalità di controllo sui singoli servizi di FE da parte di un amministratore PEOPLE utilizzando la PEOPLE Console.

L'amministratore potrà selezionare il nodo di FE da gestire (ciascuna istallazione PEOPLE può infatti avere più nodi di FE, corrispondenti a più comuni) e quindi selezionare il servizio interessato.

Accede pertanto alla maschera, già vista in precedenza, con tutte le informazioni relative al singolo servizio.



Da questa pagina potrà effettuare lo **start e stop** del servizio, il che comporterà la disattivazione del servizio sul FE (la sua invocazione mostrerà una pagina di "Servizio non disponibile").

Sarà possibile **settare i livelli di log** minimi da registrare sul FE, che saranno DEBUG, WARN,INFO,ERROR, FATAL e OFF (nessun log andrà registrato).

Tali livelli di log vanno intesi come crescenti, il che significa che se l'amministratore setterà il livello a ERROR, solo i log ERROR e FATAL dovranno effettivamente essere registrati.

Lo scopo di questa funzione è abbastanza evidente: in un sistema con centinaia di servizi per decine di nodi il controllo sul livello di log è cruciale quanto il log stesso.

I log verranno conservati sul FE DB e trasferiti sul nodo PEOPLE solo a richiesta.

Il **settaggio dei parametri** consente di modificare eventuali parametri operativi del singolo servizio di FE agendo semplicemente dalla PEOPLE Console.

La funzionalità più importante, sia in fase di configurazione che in quella di amminstrazione, è costituita dal **mapping dei servizi di BE**.

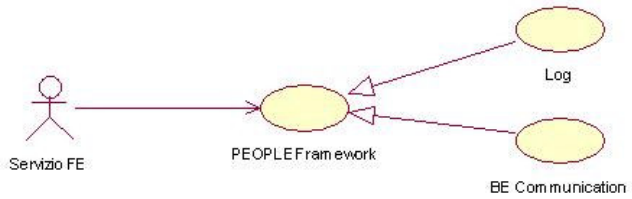
Come accennato nel paragrafo relativo alla fase di configurazione, ciascun servizio di FE al momento della registrazione "dichiara" le sue dipendenze verso componenti di BE, fornendo un identificativo logico (una stringa) che sarà utilizzata per richiedere il servizio al momento opportuno.

Dalla PEOPLE Console sarà possibile mappare tali dipendenze in due modi: indicando una casella e-mail verso cui inviare la richiesta, oppure indicando il servizio di BE (fra quelli registrati sul nodo PEOPLE) che sarà utilizzato allo scopo.

Tale mapping resterà ignoto al servizio di FE, il quale continuerà ad inviare le sue richieste utilizzando l'identificativo logico.

Nel capitolo successivo vedremo in dettaglio cosa succederà al momento della richiesta.

4.3 Scenario Runtime



Al momento il sistema PEOPLE offre ai servizi di FE due funzionalità: il log ed il supporto nella comunicazione con il BE.

Il **log** avviene mediante l'invocazione da parte del servizio di un apposita classe del FE Framework che utilizza un modello di log simile a quello del noto framework Log4J.

L'aspetto fondamentale è che il log verrà registrato (sul FE DB) solo se il livello sarà uguale o superiore a quanto richiesto da PEOPLE.

In altre parole, qualora l'amministratore di PEOPLE abbia impostato un certo servizio sul livello FATAL, i log di livello DEBUG vengono semplicemente ignorati dal sistema.

Il log registrato sul FE DB può essere richiesto dalla PEOPLE Console, con un apposito web method, e quindi inviato al nodo PEOPLE per eventuali analisi.

La **comunicazione fra FE e BE** avviene sfruttando alcuni meccanismi preesistenti nel framework 1.0a, a cui sono state aggiunte funzionalità di controllo e configurazione da parte del sistema PEOPLE.

L'amministratore PEOPLE può impostare per un determinato servizio un mapping sul BE o verso un servizio di invio dati tramite mail.

La comunicazione verso un servizio di BE avviene tramite l'invocazione (trasparente al singolo servizio) del web method process() presente sul PEOPLE Service.

I servizi di comunicazione di PEOPLE Service provvederanno a risolvere opportunamente il mapping verso il servizio di BE e ad inviare i dati.

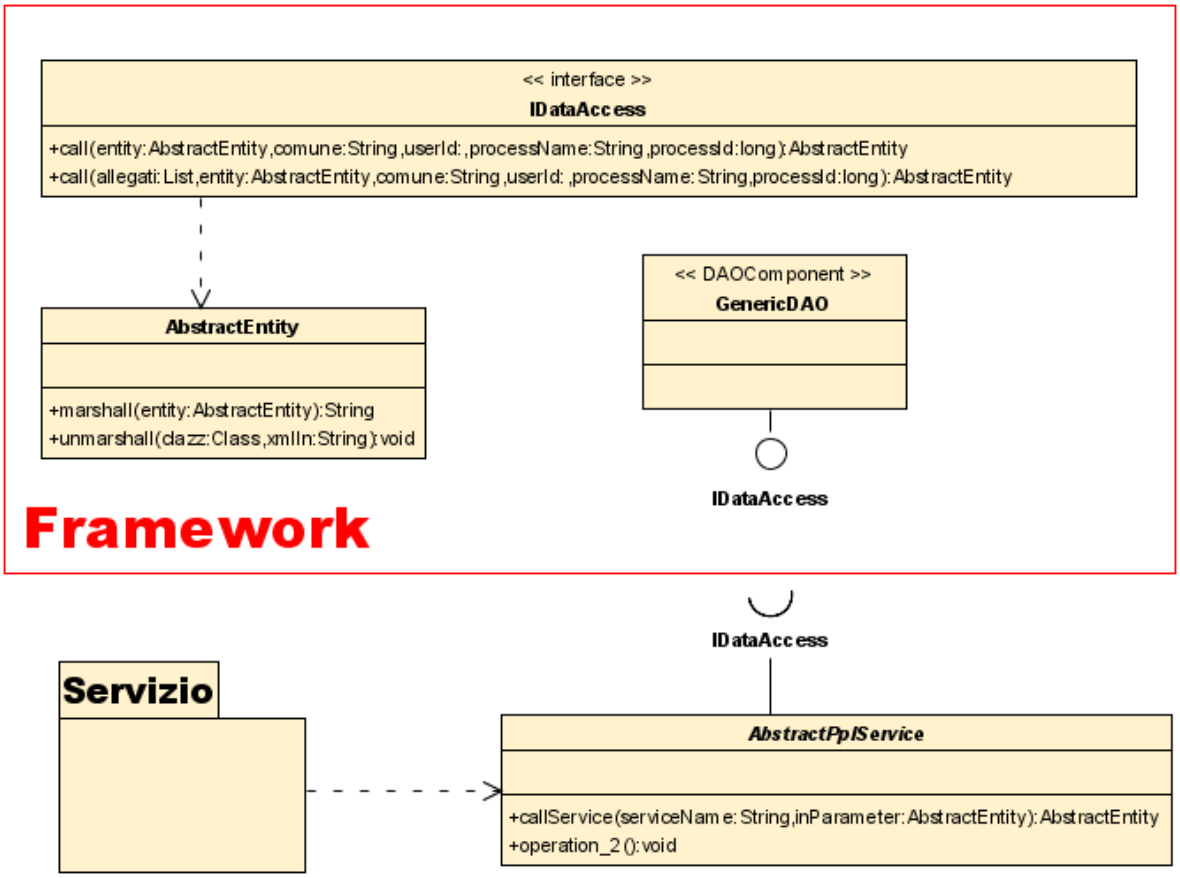
E' inoltre prevista la realizzazione di una funzionalità di "shortcut", consistente nell'invio asincrono della richiesta in tutti quei casi in cui sia noto a priori che non ci sarà risposta.

In questo caso il FE Framework provvederà a storicizzare la richiesta, liberare il servizio di FE e provvedere in un secodo tempo alla spedizione.

Il mapping verso un indirizzo di posta viene risolto direttamente sul FE, dato che il FE Framework in questo caso è in grado di inviare (così come accadeva nella versione 1.0a) direttamente la mail.

4.4 Integrazione con il BE

Questo paragrafo descrive le modalità di interazione fra i servizi di FE, il FE Framework ed i servizi di BE.



Il framework 1.0b aveva correttamente impostato, il design relativo a questa funzionalità, ma per poter completare l'integrazione con i restanti componenti nella versione PEOPLE 1.0 e per assicurare la corrispondenza con tutti i requisiti funzionali è stato necessario introdurre alcune modifiche.

Dal punto di vista architetturale lo sviluppo dei servizi non ha subito modifiche: il bounduary object AbstractPplProcess che espone i metodi per l'invocazione dei Web

Service è rimasto invariato, tutte le modifiche sono state apportate all'interno del framework stesso.

Gli interventi principali effettuati sono:

- la chiamata al servizio di BE avviene ora indicando il nome logico del servizio
- gli oggetti passati in input e ricevuti in output dalla chiamata devono essere realizzati ad hoc
- riprogettazione del metodo di gestione degli allegati ai processi
- eliminazione dell'utilizzo del package sun.*

L'identificazione del servizio di BE avviene ora con il nome logico dichiarato nel file config.xml così da consentire ai servizi di gestione di poter effettuare il mapping sul servizio di BE concreto. La precedente soluzione, basata sul nome della classe Java era ovviamente adatta al solo mondo Java (e quindi non adatta a chiamate multipiattaforma) e non consentiva in modo naturale tale mapping.

Il meccanismo di passaggio dei parametri attuale consiste nello spedire l'intero oggetto ProcessData: questo tipo di approccio semplifica lo sviluppo ma pone problemi architetturali e tecnici, tra cui una improbabile aderenza agli standard che verranno dettati dal gruppo di modellazione. Inoltre la comunicazione era inefficiente, dato che sostanzialmente ogni chiamata al BE trasferiva tutti i dati inseriti dall'utente, sia che servissero o meno al servizio invocato.

Il nuovo approccio implica che per ogni richiesta ad un servizio di BE il fornitore del servizio di FE debba realizzare un Bean per l'input ed uno per l'output, entrambi derivati dalla classe preesistente AbstractEntity e contenenti tutti e soli i campi necessari al servizio.

Così come nella versione 1.0b sarà il framework a generare il relativo XML, e quindi l'aderenza ai modelli previsti dal gruppo di modellazione per i dati sarà realizzata non operando su file XSD, ma impostando coerentemente la struttura interna dei Bean.

Questo tipo di approccio può sembrare complesso (dal punto di vista dell'aderenza alle spevifiche del gruppo di modellazione), ma è l'unico che consenta ai fornitori di non implementare manualmente il marshalling/unmarshalling dei dati.

Poichè è il framework a realizzare materialmente la conversione Bean/XML (e viceversa) l'unico modo di ottenere un particolare XML consiste nell'implementare oggetti con la

struttura interna desiderata. Questa tecnica, nella sua versione originaria, presentava dei limiti nella gestione di file allegati, che venivano riprodotti con un XML eccessivamente verboso e del tutto inefficiente (ogni byte del file diveniva un elemento <byte>###</byte>),
Il problema è stato risolto presentando il contenuto di ogni file come un singolo tag <data> codificato in BASE64, un formato molto efficiente e veloce per il trasferimento di file binari rappresentati solo con caratteri stampabili.
L'utilizzo dei package sun.* è stato eliminato poichè sconsigliato dalla stessa Sun: si tratta di un package non documentato e non portabile su altre JVM.

4.5 Il log

Al fine di monitorare le attività svolte dai servizi del framework sono state implementate delle funzionalità di logging analoghe a quelle fornite dal progetto log4j utilizzato in precedenza nel framework.
Gli oggetti implementati ricalcano esattamente le segnature di quelli definiti dalla specifica log4j in modo da ridurre al minimo l'impatto del cambiamento nell'utilizzo delle funzionalità di logging dall'interno dei singoli servizi.
Per i servizi che utilizzano le API log4j, sarà dunque necessario sostituire gli import relativi con quelli delle API implementate per il Logger di People.
Come per log4j, un'istanza di un Logger per un determinato servizio potrà essere ottenuta chiamando il metodo statico

Logger.getLogger(Class serviceClass)

che accetta la classe del servizio completa di indicazione del package nel quale è contenuta.
Le informazioni di log ritenute necessarie e che vengono registrate su un determinato supporto sono:

- Il livello minimo di log richiesto dal servizio
- Il codice identificativo del comune richiedente
- Il codice identificativo del servizio
- Il messaggio informativo

Tali informazioni sono passate in modo trasparente al Logger grazie ad un componente JavaBean. Lo sviluppatore si preoccupa solamente di chiamare uno dei metodi di scrittura del Logger passandogli un'opportuno messaggio sottoforma di stringa..

Come per log4j, sono previsti 5 principali tipi di logging, quindi 5 principali metodi per la stampa delle informazioni sul supporto sottostante (DB o file di testo):

- Logger.debug(String messaggio)
- Logger.info(String messaggio)
- Logger.warn(String messaggio)
- Logger.error(String messaggio)
- Logger.fatal(String messaggio)

Come per log4j, un messaggio di log da stampare utilizzando il metodo warn() non verrà effettivamente stampato se il livello di log richiesto dal servizio è ERROR o FATAL. In generale la chiamata ad un metodo di log di livello X si concluderà con esito positivo, e quindi con la stampa sul supporto di log, solo se il livello di log richiesto dal servizio per il quale si sta loggando è minore o uguale a X.
Inoltre, il generico servizio può richiedere un livello di log pari ad ALL o ad OFF. Nel primo caso qualsiasi metodo del Logger per la scrittura effettua l'operazione di log senza effettuare controlli di livello. Nel caso in cui il livello sia OFF, il Logger del servizio non esegue nessuna stampa.
Un'interfaccia Appender, analoga a quella di log4j, è stata implementata per gestire la scrittura dei log su di un determinato supporto di memorizzazione. L'implementazione di default fornisce un'Appender che realizza le caratteristiche principali di un JDBCAppender in log4j. Le informazioni di log vengono così scritte nella base di dati dei nodi di frontend (fedb), in particolare, nella tabella log_messages.
Una seconda implementazione dell'interfaccia Appender realizza la scrittura dei log di un servizio su un determinato file di testo che prende il nome del package e della classe del servizio stesso.
I file di log di tutti i servizi vengono memorizzati in una cartella individuata da un percorso configurabile nel deployment descriptor (web.xml) della web application del framework. Lo sviluppatore del singolo servizio potrà modificare questa 'variabile d'ambiente' per decidere dove andranno memorizzati i log di tutti i servizi.
La possibilità di poter effettuare dei log su file di testo anziché DB fornisce un'ottimo strumento da utilizzare in fase di sviluppo dei singoli servizi. Il metodo di istanzazione di un

Logger su file di testo prende in tal caso un'ulteriore parametro 'media' di tipo java.lang.String:

```
Logger.getLogger(serviceClass, "file");
```

Un'ultima funzionalità che verrà implementata successivamente riguarda la possibilità di visualizzare, tramite Web Services, le informazioni di log direttamente sulla PeopleConsole prelevandole dalla base di dati. Vi saranno inoltre delle funzionalità per l'individuazione mirata di determinati fasi di log e per la cancellazione di altre.

5 Deployment di PEOPLE 1.0

Dal punto di vista del deployment, l'architettura proposta vede una struttura composta da:

- un nodo PEOPLE
- uno o più nodi FE
- zero o più nodi di BE

Il nodo PEOPLE è il centro di controllo ed il fulcro dell'architettura, consentendo fra l'altro il collegamento fra i componenti di FE e di BE.

In futuro potrebbe consentire la cooperazione applicativa, realizzando la comunicazione fra diversi sistemi PEOPLE.

I nodi FE sono tanti quanti sono i comuni interessati dall'istallazione, dato che per varie ragioni si è ritenuto di mappare in questo modo i componenti software sulle amministrazioni relative.

I nodi di BE realizzano l'interfacciamento con il back office dei singoli comuni e sono in linea teorica non indispensabili, dato che si potrebbe pensare di risolvere tutte le chiamate tramite mail.

Più probabilmente, ciascuna istallazione di PEOPLE avrà molti servizi di BE.

5.1 Sistema distribuito

L'architettura proposta pone il concetto di nodo come l'unità di deployment, e ciascun nodo può essere istallato su un server localizzato differentemente.

Tale possibilità risulta di particolare interesse per le istallazioni di tipo ASP, in cui un centro di grandi dimensioni (es. un capoluogo di provincia o di regione) potrebbe istallare il nodo PEOPLE e vari nodi di FE per vari comuni "affiliati".

Ciascun comune potrebbe poi istallare i necessari nodi di BE localmente.

Questo tipo di istallazione ha implicazioni sulla sicurezza che potranno essere risolte mediante VPN o SSL.

5.2 Sistema multipiattaforma

Quanto detto nel paragrafo precedente circa la possibilità di istallazioni distribuite del sistema, può essere enfatizzato sottolineando che i singoli nodi (poichè comunicano esclusivamente tramite web service) potranno essere istallati anche su piattaforme differenti (Linux o Microsoft based) od utilizzare linguaggi differenti (Java, .net oppure, per

i soli BE, qualsiasi cosa sia in grado di realizzare la necessaria interfaccia basata su web service).

Questa possibilità risulta particolarmente interessante proprio per le istallazioni di tipo ASP, dato che consente ai singoli comuni di esporre i web service di BE utilizzando qualsiasi tecnologia idonea preesistente.