



L'e-government portato ai cittadini anche sul canale televisivo digitale interattivo

Raffaele Andreace – Mario Polito

AGSM Telecomunicazioni Verona

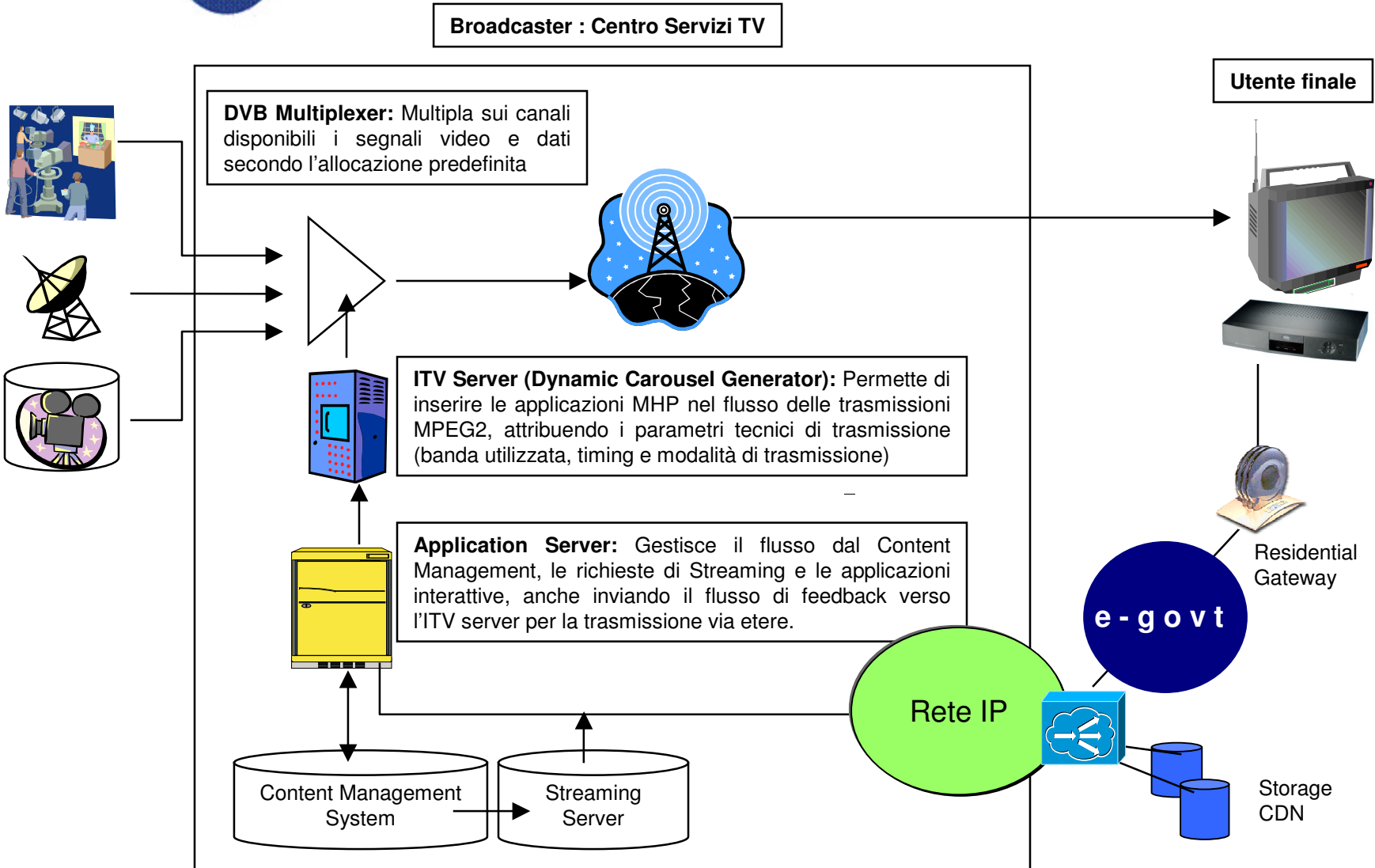
Raffaele Andreace – Direttore Internet Technology Division

Mario Polito – Sr. Application & Technology consultant, Internet Technology Division

Indice

- Scenario di riferimento DTT
- Vincoli tecnologico – funzionali
- Tipologie di servizi T-gov
- Requisiti di Design Time
- Requisiti di Run Time
- La generazione di una applicazione/Servizio
- Una proposta di architettura
- Il iDTT Gateway

Scenario di riferimento DTT



Vincoli Tecnologici e funzionali

- **Schermo:** la *bassa risoluzione* comporta la impossibilità di mostrare testi lunghi;
- **Input device:** il telecomando è per adesso l'unico meccanismo di input standardizzato e riconosciuto. Mentre risulta semplice l'input di numeri e il movimento di un puntatore, *l'input del testo è penalizzato*.
- **Il download** delle applicazioni avviene mediante il canale televisivo e non il canale di ritorno (almeno per l'immediato futuro). Di conseguenza non possono essere interessanti servizi di nicchia o business.

Vincoli Tecnologici e funzionali

- **MHP** è ancora abbastanza *povero di funzionalità*: ci si aspetta che questo vincolo diventi sempre meno importante già dalle prossime release di MHP;
- I *dispositivi* di gestione delle *Smart Card non* sono ancora *attivi* sui STB. Non è possibile quindi realizzare servizi che prevedano firma digitale (stipule di contratti);
- Il **canale di ritorno** è in genere a pagamento. Questo fatto, combinato con la *lentezza dell'input*, implica la necessità di *evitare lunghe transazioni* interattive con il canale di ritorno aperto. Inoltre una lunga serie di *"request-response"* sul canale di ritorno è un modello più adatto alla navigazione Internet che al modello di servizio Televisivo, anche come percezione da parte dell'utenza.

Tipologie di servizi T-govt

MUST per ogni servizio di T-govt

- Il Servizio deve essere adatto alle caratteristiche di usabilità del mezzo e rispettare i vincoli sopra esposti;
- Il servizio deve riutilizzare logiche di business già realizzate per gli altri canali, per esigenze architettoniche e di sincronizzazione dei dati (un utente potrebbe usare differenti canali per accedere agli stessi servizi).

Tipologie di servizi T-govt

Presentazioni statiche

L'utente naviga in una serie di schermate e informazioni a "lento cambiamento". Il canale di ritorno non è utilizzato : dati e applicazioni sono residenti o già' downloaded su STB.

Navigazione dinamica di informazioni

L'utente naviga anche eseguendo richieste, ricerche ecc.. L'interattività è solo apparente in quanto le informazioni vengono aggiornate e trasmesse insieme alla applicazione. Il canale di ritorno non viene utilizzato dati e applicazioni sono residenti o già' downloaded su STB.

Richiesta di servizi mediante compilazione di moduli

L'utente visualizza diverse schermate di input, inserisce le necessarie informazioni e giunge alla conferma di servizio erogato. **Il canale di ritorno deve essere attivato.**

Requisiti “Design-time”

1. E' necessario un linguaggio di specificazione di un servizio, a livello di dati di input da parte dell'utente, in modo che le funzioni competenti in ambito Pa e i Service creators possano accordarsi velocemente e con precisione.
2. Deve essere definito un unico protocollo a basso/medio livello per lo scambio di dati sul canale di ritorno: il protocollo deve quindi essere implementato e disponibile su client, e usato in ogni applicazione che faccia uso del canale di ritorno.
3. A livello di server deve essere disponibile un sistema che recepisca questo protocollo e su cui si possano velocemente istanziare interfacce ad alto livello specifiche per ogni servizio;

Requisiti “Design-time”

4. A livello di “application definition” si devono poter creare in maniera standardizzata le classi per la gestione del canale di ritorno;
5. Si deve disporre di un sistema RAD (rapid application development), information driven, per l’implementazione di flussi di navigazione, risiedenti su client;
6. Si deve disporre di un sistema RAD per il disegno di schermate di input, che possa agevolmente gestire la frammentazione di pagine (non è possibile il caso, tipico di internet, di schermate molto più lunghe dell’altezza dello schermo);

Per RAD si intende semplice, non invasivo, non opaco rispetto all’ambiente MHP sottostante, deve essere “non addictive” [si puo’ farne a meno]

Requisiti “Run-time”

1. Le schermate di input devono contenere campi di testo (riempibili con tastiere virtuali), campi booleani (checkbox), liste a scelta singola (equivalenti ai combobox);
2. Il Front end del sistema (parte della architettura che riceve i dati di input) deve validare i dati in input, con un sistema abbastanza potente per esprimere vincoli complessi;
3. Il Front end del sistema deve rispondere indicando i messaggi di successo o di errore che appariranno direttamente a schermo;
4. Il front End del sistema deve essere in grado di istanziare processi multiformi e facilmente modificabili;

Requisiti “Run-time”

5. Il front end del sistema deve permettere agli amministratori del sistema di capire in ogni momento lo stato di un processo, a tutti i livelli del protocollo;
6. Il Front end del sistema deve interfacciarsi con il back end con diversi protocolli di basso livello (SMTP, FTP, JRMI, Web Service);
7. Deve essere possibile sul client utilizzare dati persistenti, tipicamente di utilizzo comune a vari servizi, che l'utente immette una volta sola. Per esempio i dati personali (vedi personal data cache)
8. In futuro si deve prevedere l'integrazione con il meccanismo delle smart card con certificato di sottoscrizione, in modo da poter utilizzare i servizi che prevedono la stipula (mediante firma digitale) di un contratto.

Requisiti “Run-time”

9. Personal data cache (cache dati personale) : apribile mediante PIN 6 cifre (necessita create/add/modify/delete/reset ...) e' una shortcut di sicurezza
10. Identificazione personale tramite smart-card CNS/CIE con certificato di autenticazione : abilitazioni a servizi ad autenticazione forte (non basta id STB, che e' condiviso da piu' persone) : nel periodo di interregno e' possibile utilizzare schemi alternativi e.g. HTTPS “unlock(PIN((DI+CF+paPIN o password))”

Dove **unlock PIN** (apre la personal cache di dati punto 9)

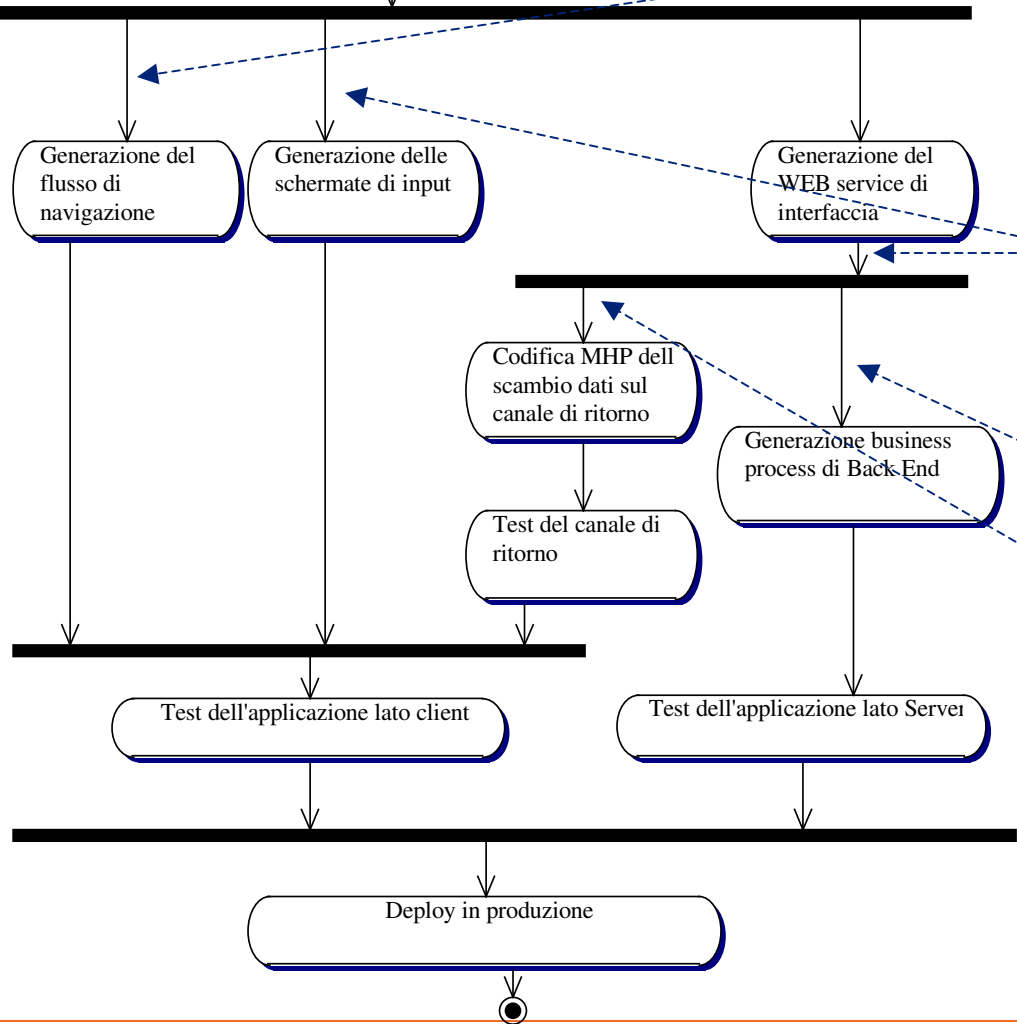
DI sono i dati identificativi (per persona in STB) apribili via unlock PIN (cache)

CF codice fiscale (certificato SIATEL) (in cache)

pa PIN o password e' un codice (temporaneo) utilizzato dalla PAL per l'accesso ai propri servizi (potrebbe non essere specifico per il canale DTT) consegnato previa identificazione utente (simile CNS/CIE)



Flusso e dati specificati



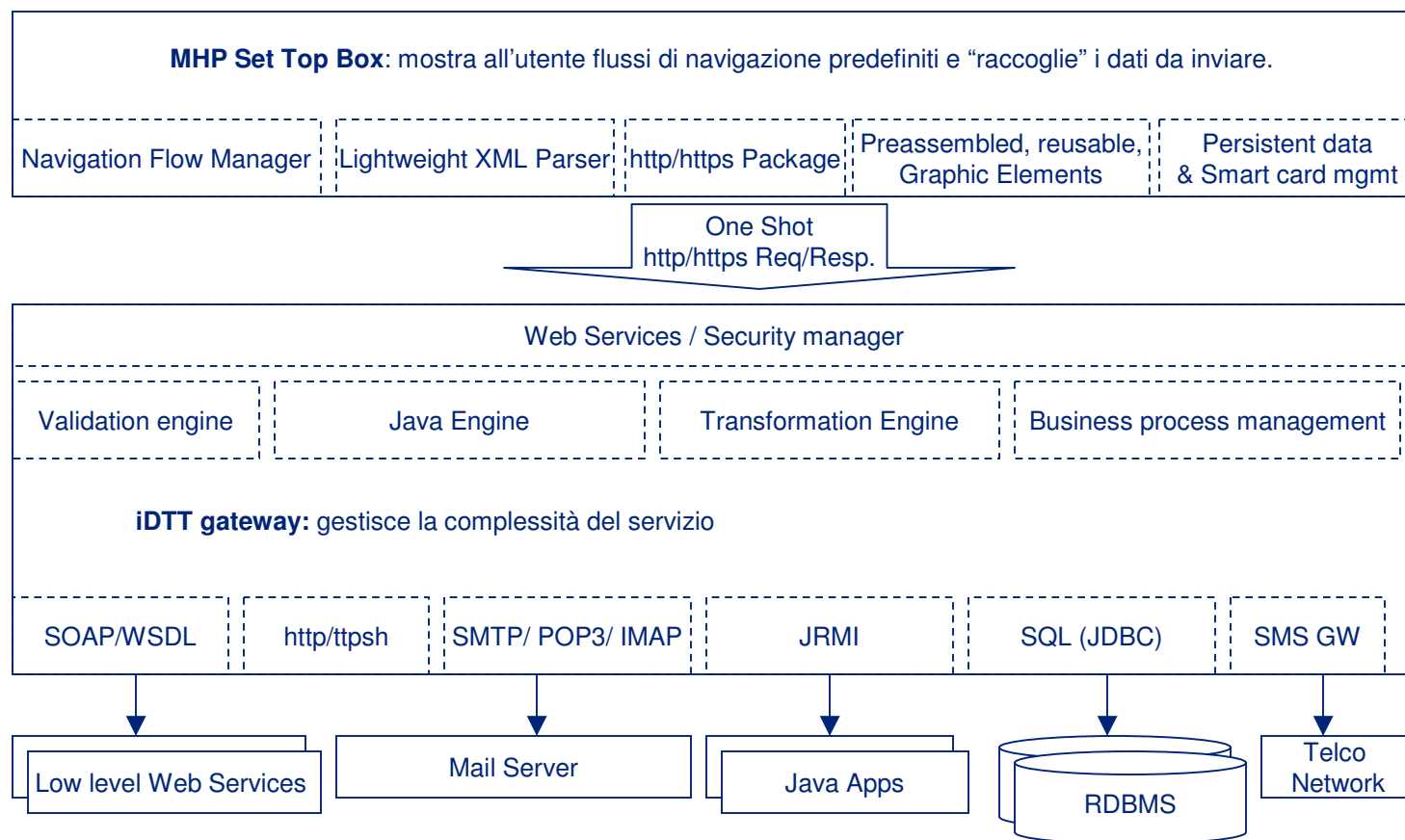
XML con grammatica (schema xsd) condivisi tra Service designer e app creator. Non esiste uno standard W3C. La grammatica DEVE essere autoconsistente.

W3C XSD che contiene la definizione della grammatica dello scambio dati tipico del servizio. Lo STESSO XSD viene usato run time per la validazione dei dati di input

W3C WSDL (Web Service Definition Language)

Framework di classi Java che implementano il protocollo di basso/medio livello

Una proposta di architettura per T-govt



iDTT Gateway (in sviluppo a Verona)

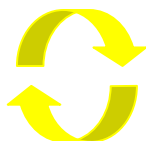
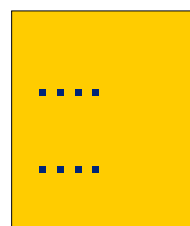
Scopo

- Abilitare da subito le applicazioni portalizzate di e-govt semplificando l'interazione MHP
- Costituire uno standard "de-facto" per l'e-govt su DTT per l'interscambio di pacchetti applicativi MHP/Applicazione specifica di e-govt
- Accelerare l'utilizzo delle funzionalita' proprie del DTT a reale utilita' del cittadino per le applicazioni di e-govt

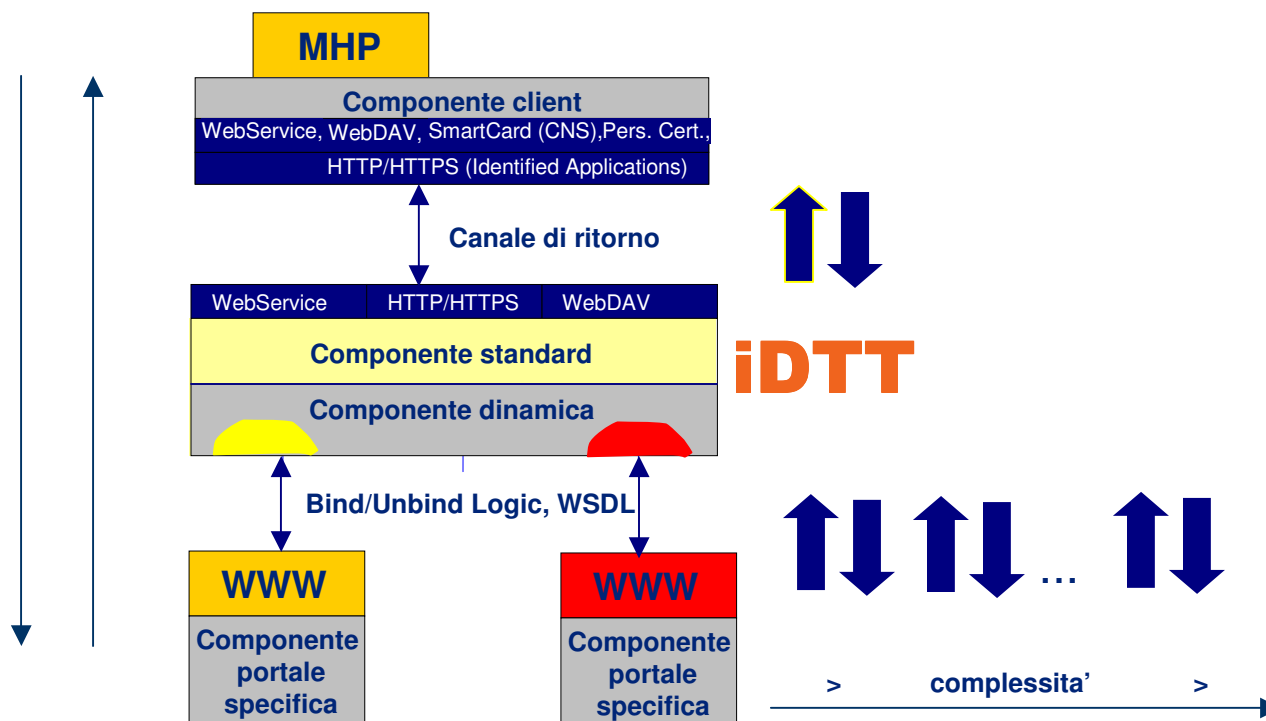
iDTT Gateway : Internet Digital Terrestrial Television Gateway

iDTT Gateway (in sviluppo a Verona)

Inoltre informazioni di sessione + applicative



Transazione end-2-end



iDTT Gateway

Informazioni ulteriori

- Si occupa di gestire la complessità di una applicazione di T-govt a partire da una serie di dati immessi incondizionatamente (per esempio l'autenticazione dell'utente viene controllata su iDTT a valle dell'immissione di TUTTI i dati del servizio e del loro invio a iDTT);
- Gestisce tutte le funzioni di business logic (validazione, processi, ecc.);
- Implementa la comunicazione con tutti i servizi di back end e tutti i protocolli di trasporto interessanti;
- Fornisce i necessari servizi di sicurezza;

Principali Componenti tecnologici

- Web Service manager;
- Java Application manager;
- XSD engine (validatore);
- XSLT engine (trasformazione dei dati);
- Process management system (TBV W3C Web services choreography);