

# Integrazione di sistemi e messaggistica asincrona

Francesco Malvezzi

Università di Modena e Reggio nell'Emilia

9 novembre 2011

# Introduzione

Questo talk introduce le strategie di integrazione di sistemi come catalogate da Martin Fowler<sup>1</sup>:

- scambio di file;
- database condivisi;
- RPC (remote procedure call);
- messaggistica asincrona.

(con qualche preferenza per la messaggistica asincrona)

---

<sup>1</sup> *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*, Gregor Hohpe e Bobby Woolf

# La sfida dell'integrazione dei sistemi

Integrare sistemi diversi è difficile perché

- i sistemi devono essere disaccoppiati;
- le modifiche devono essere poco invasive;
- i risultati dell'integrazione devono essere sempre aggiornati.

D'altra parte integrare sistemi è una necessità, perché l'alternativa, cioè sviluppare sistemi unitari, non è spesso percorribile.

# Scambio di file

Ogni applicazione coinvolta produce uno scarico dei dati. I dati provenienti da diverse applicazioni vengono importati, filtrati e processati per essere importati nelle applicazioni che li necessitano.

Vantaggi:

- semplice culturalmente per gli sviluppatori;
- praticamente ogni servizio può produrre file;
- non introduce nessun accoppiamento nei sistemi;

Svantaggi:

- rischi di dissonanza cognitiva;
- mai aggiornato.

# Database condiviso

Tutte le applicazioni condividono una base di dati. Ciascuna applicazione si integra con le altre leggendo le tabelle dati altrui.

Vantaggi:

- facile grazie alla diffusione dei DB relazionali;
- elimina dissonanza cognitiva;
- in tempo reale.

Svantaggi:

- difficile incapsulamento dei dati.

# RPC (remote procedure call)

Ogni applicazione espone delle funzionalità che possono essere chiamate come servizi (in apparenza locali).

Esempi: Java RMI, SOAP, REST

Vantaggi:

- semantica molto ricca perché sono esportate delle funzioni, non i semplici dati;
- forte incapsulamento dei dati;
- in tempo reale.

Svantaggi:

- performance;
- dipendente dalla disponibilità di tutti gli attori;
- forte accoppiamento dei sistemi.

# Messaggistica asincrona - 1

Il collante è un sistema di code dove le applicazioni pubblicano solo le variazioni dei dati.

Chi è interessato a certe variazioni si abbona a certe code e riceve notifica.

Il sistema è distribuito, per cui all'assenza di rete o di uno degli attori non blocca l'intero sistema.

La messaggistica asincrona (come la mail) svincola il richiedente dall'attesa della risposta: durante la elaborazione il richiedente è libero.

## Messaggistica asincrona - 2

È possibile interporre in maniera trasparente degli elementi per disaccoppiare i sistemi (enterprise integration pattern):

- routing: il mittente non è obbligato a conoscere l'indirizzo del destinatario;
- traduzione: il formato dati può essere trasformato durante il trasferimenti per adattarsi al ricevente;
- trasformazioni complesse: ad esempio un messaggio può essere suddiviso, o più messaggi raggruppati o a un messaggio possono essere aggiunti dati ulteriori (ottenuti come richieste sincrone o asincrone ad altri servizi).



# Messaggistica asincrona - vantaggi e svantaggi

## Vantaggi:

- debole accoppiamento tra le applicazioni;
- resilienza alla mancanza di un attore;
- miglioramento delle prestazioni;
- tempo reale.

## Svantaggi:

- culturalmente ostile.

# Prodotti per la messaggistica asincrona

Come minimo serve un applicativo per gestire le code, ad esempio:

- **activemq** <http://activemq.apache.org>;
- **rabbitmq** <http://www.rabbitmq.com>.

che è il componente che accetta i messaggi in coda, ne cura la persistenza, li rende disponibili ai client che li richiedono in accesso concorrente.

Può essere utile una libreria di trasporto e trasformazioni (ad esempio **camel** <http://camel.apache.org>), cioè uno strumento per gestire ad alto livello gli enterprise integration pattern.

È possibile fondere i due elementi precedenti in un Enterprise Service Bus (ad esempio **servicemix** <http://servicemix.apache.org>).

# Il correlatore di UniMORE - cosa vogliamo ottenere

Il lavoro del sistema di gestione delle identità di UniMORE è la popolazione di un directory service OpenLDAP a partire dalla basi dati autoritative dell'ateneo.

Ad esempio: il record ldap di uno studente è composto da dati provenienti da:

- ESSE3 (anagrafica e dati dei corsi frequentati);
- servizio posta (indirizzo mail reale e se e' attivo);
- database delle password (passwd cifrata e dati posix\_account e shadow\_account).

I dati dei dipendenti (utenti strutturati i cui dati provengono dal settore risorse umane) e degli esterni (utenti inseriti autonomamente dalla strutture) hanno struttura simile.

La stessa persona può ricoprire più ruoli contemporaneamente (esterno e studente, ad esempio) e può avere fino a due entry ldap con due dn differenti.

## Il correlatore di UniMORE - i componenti

- raccogliitore (scansiona le basi dati autoritative e raccoglie i dati. La chiave di correlazione è il codice fiscale);
- correlatore (riceve i dati dal raccogliitore e produce i record ldap dell'utente);
- delta (confronta i record ldap precedente con la situazione in linea su OpenLDAP per generare un file differenza ldif)
- read\_diff (legge i file differenza per popolare la coda degli utenti aggiunti, cancellati, degli utenti che hanno modificato ad esempio la mail o il numero di telefono)

Questi componenti sono collegati tra di loro in pipeline su un unico gestore di code ActiveMQ.

## Il correlatore di UniMORE - un nuovo utente

Questo è un caso interessante!

Se arriva al raccoglitore il CF di un nuovo utente, il raccoglitore trova tutti i dati salvo i dati della username (uidnumber, shadow\_entry).

Il correlatore riceve dati incompleti e lancia un'eccezione, che genera un messaggio che viene instradato verso il server della gestione delle username. Il messaggio contiene i dettagli per creare il record di username.

Il server delle username processa il messaggio e invia un messaggio che contiene il codice fiscale dell'utente al raccoglitore.

Questo messaggio si colloca nella coda a massima priorità del raccoglitore che lo processa: stavolta trova tutti i dati. L'utente entra in ldap quasi istantaneamente.

## Il correlatore di UniMORE - un nuovo utente

Questo è un caso interessante!

Se arriva al raccoglitore il CF di un nuovo utente, il raccoglitore trova tutti i dati salvo i dati della username (uidnumber, shadow\_entry).

Il correlatore riceve dati incompleti e lancia un'eccezione, che genera un messaggio che viene instradato verso il server della gestione delle username. Il messaggio contiene i dettagli per creare il record di username.

Il server delle username processa il messaggio e invia un messaggio che contiene il codice fiscale dell'utente al raccoglitore.

Questo messaggio si colloca nella coda a massima priorità del raccoglitore che lo processa: stavolta trova tutti i dati. L'utente entra in ldap quasi istantaneamente.

## Il correlatore di UniMORE - un nuovo utente

Questo è un caso interessante!

Se arriva al raccoglitore il CF di un nuovo utente, il raccoglitore trova tutti i dati salvo i dati della username (uidnumber, shadow\_entry).

Il correlatore riceve dati incompleti e lancia un'eccezione, che genera un messaggio che viene instradato verso il server della gestione delle username. Il messaggio contiene i dettagli per creare il record di username.

Il server delle username processa il messaggio e invia un messaggio che contiene il codice fiscale dell'utente al raccoglitore.

Questo messaggio si colloca nella coda a massima priorità del raccoglitore che lo processa: stavolta trova tutti i dati. L'utente entra in ldap quasi istantaneamente.

## Il correlatore di UniMORE - un nuovo utente

Questo è un caso interessante!

Se arriva al raccoglitore il CF di un nuovo utente, il raccoglitore trova tutti i dati salvo i dati della username (uidnumber, shadow\_entry).

Il correlatore riceve dati incompleti e lancia un'eccezione, che genera un messaggio che viene instradato verso il server della gestione delle username. Il messaggio contiene i dettagli per creare il record di username.

Il server delle username processa il messaggio e invia un messaggio che contiene il codice fiscale dell'utente al raccoglitore.

Questo messaggio si colloca nella coda a massima priorità del raccoglitore che lo processa: stavolta trova tutti i dati. L'utente entra in ldap quasi istantaneamente.



# Un rimpiazzo per OpenLDAP syncrepl

Dopo che il server ldap è stato popolato con il delta, si possono propagare le modifiche con il protocollo proprietario `syncrepl`, però:

- funziona solo tra server OpenLDAP (no Apache DS, no RedHat Directory Service);
- personalmente trovo che abbia molte idiosincrasie

```
curl http://www.openldap.org/software/release/changes.html  
| grep -i syncrepl | grep Fixed | wc -l
```

Soluzione:

Usare la messaggistica asincrona per propagare le differenze.

# Un rimpiazzo per OpenLDAP syncrepl

Dopo che il server ldap è stato popolato con il delta, si possono propagare le modifiche con il protocollo proprietario `syncrepl`, però:

- funziona solo tra server OpenLDAP (no Apache DS, no RedHat Directory Service);
- personalmente trovo che abbia molte idiosincrasie

```
curl http://www.openldap.org/software/release/changes.html  
| grep -i syncrepl | grep Fixed | wc -l
```

Soluzione:

Usare la messaggistica asincrona per propagare le differenze.

## Componenti:

- output del correla;
- filter che capisca quali record propagare (certe variazioni non interessano tutte le repliche);
- modifica del body del messaggio per rimuovere gli attributi che non devono essere rilasciati;
- sistema di messaggistica con code di trasmissione tra master e slave con alcune code etichettate con priorità differente (le code più lente servono per gli inventari);
- (eventualmente) modifica dei messaggi sullo slave;
- applicazioni degli Idif (delta).

Da notare:

- impossibile senza la messaggistica asincrona;
- riutilizzo dei componenti del correlatore per un uso diverso;
- composto da (pochi) elementi ciascuno molto semplice.

- **Enterprise Integration Patterns:**  
<http://www.eaipatterns.com/>, ISBN-13: 978-0321200686