



AscotWeb - mediatore

Versione 1.0.1

21 dicembre 2015

Storia delle Modifiche

Versione	Data	Descrizione
1.0	19/05/2016	Prima versione
1.1	19/05/2016	Aggiunti i riferimenti alle implementazioni specifiche da Ascot

Riferimenti

Numero	Titolo	Prodotto da	Versione	Data
1	Enterprise Integration Patterns	http://www.enterpriseintegrationpatterns.com/patterns/messaging/		
2	Apache Camel	http://camel.apache.org/		
3	Java Message Service	https://docs.spring.io/spring/docs/current/spring-framework-reference/html/jms.html		

Acronimi

Sigla	Descrizione
JMS	Java Message Service
EIP	Enterprise Integration Pattern

Sommario

1. Introduzione	5
2. Il componente mediatore	6
3. Interfacce esposte verso sistemi terzi.....	9

1. Introduzione

Gli obiettivi del presente documento sono quelli di descrivere l'architettura del componente mediatore e dei paradigmi di base utilizzati per la sua progettazione.

2. Il componente mediatore

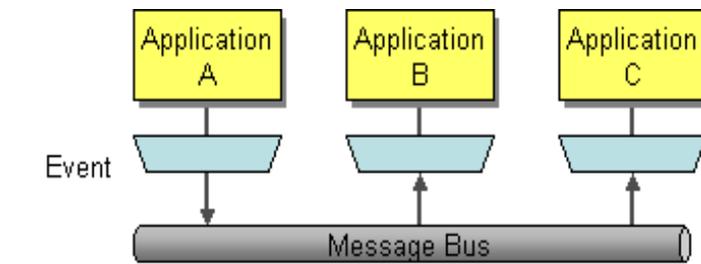
Il componente mediatore nasce con l'esigenza di semplificare l'integrazione tra ASCOT ed applicazioni/sistemi interni o esterni. La parola "integrazione" in ambito ICT significa trovare una risposta al seguente quesito: "Come far comunicare tra loro sistemi di natura eterogenea".

A questo scopo sono stati formalizzati gli Enterprise Integration Pattern che sono soluzioni concettuali che aiutano ad affrontare le problematiche di integrazione.

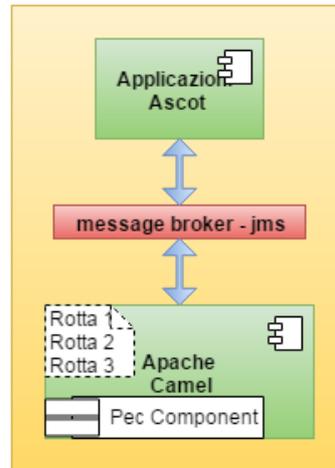
Una panoramica sui pattern di integrazione è disponibile al seguente indirizzo: <http://www.enterpriseintegrationpatterns.com/patterns/messaging/toc.html>

Questi pattern trovano implementazione nei framework di integrazione come Apache Camel o negli Enterprise Service Bus come Mule o Apache Synapse.

Il pattern architetturale adottato per il componente mediatore è quello basato sul Messaging. Questo pattern stabilisce che la comunicazione tra Ascot e i componenti interni o esterni avvenga tramite una infrastruttura di comunicazione comune e tramite scambio asincrono di messaggi attuato attraverso un sistema di code.



Il sistema di code è implementato con tecnologia Oracle Advanced Queue e l'accesso per la lettura/scrittura dei messaggi è attuato con dei broker che usano il protocollo JMS **Java Message Service**.



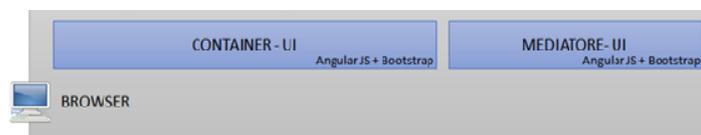
L'instradamento dei messaggi presenti in coda è basato sul concetto di rotta ed è affidato al framework di integrazione open source **Apache Camel**. Le rotte sono meccanismi atti a collegare logicamente componenti che dialogano attraverso i rispettivi endpoint.

Durante la lettura dalla coda Camel esamina le caratteristiche di ogni messaggio individuandone l'opportuna rotta, le rotte possono seguire diverse destinazioni come la PEC, i servizi ANPR, l'indirizzo di un ESB, il SERVICE LAYER, etc.

Ogni tipo di rotta ha la sua corrispondenza a livello di messaggio e, a fronte di richieste di nuove integrazioni devono essere progettate sia la struttura del messaggio che l'implementazione della rotta.

I componenti vengono scritti con tecnologia Java nella **versione 1.8**. Tale scelta è motivata dalla volontà di integrare all'interno della soluzione le caratteristiche di elevata leggibilità, compattezza, robustezza, scalabilità e parallelismo in ambiente multi-core proprie del linguaggio.

Oltre a questo saranno presenti dei componenti per il monitoraggio degli eventi presenti in coda (container) e per la configurazione del mediatore.



L'elemento **MEDIATORE** consente di soddisfare efficacemente i requisiti di integrazione che possono emergere all'interno di una struttura organizzativa complessa, mantenendo un elevato grado di:

- **MODULARITA'**: il mediatore, in virtù della sua strutturazione basata su interscambio di messaggi, è caratterizzato da una intrinseca modularità e dal ridotto accoppiamento fra le singole applicazioni o componenti che

interagiscono fra loro mediante esso. La natura modulare della soluzione e il disaccoppiamento delle componenti (non è necessario, infatti, che ciascuna componente o attore sia a conoscenza della semantica che caratterizza ciascuno dei messaggi ove non di interesse), consente un'agevole integrazione di nuove funzionalità e componenti senza che tali modifiche incidano significativamente sull'operatività generale della soluzione;

- **INDIPENDENZA DELLE COMPONENTI:** la natura asincrona del processo di comunicazione adottato dalla soluzione proposta consente a ciascuna applicazione o componente di operare in modo del tutto indipendente, anche per quanto concerne eventuali flussi da e verso servizi o sistemi esterni rispetto alla soluzione (e.g.: invio di un messaggio di PEC). Anche in presenza di un grave malfunzionamento imputabile ad una delle applicazioni o componenti, le altre componenti architettoniche sono in grado di operare in modo autonomo e garantire la fruibilità dei servizi da esse esposti. La persistenza dei messaggi consente, inoltre, il recupero e la gestione degli eventuali messaggi ignorati dalla componente non operativa;
- **SCALABILITA':** il MEDIATORE è caratterizzato da un elevato grado di scalabilità, garantito a due differenti livelli, al fine di gestire la variazione dei carichi computazionali che possono incidere su di esso.
- **AFFIDABILITA':** il MEDIATORE garantisce un elevato grado di affidabilità, al pari di quel che concerne la scalabilità, sia a livello di componenti architettoniche sia a livello di singoli attori in esecuzione. Nello specifico ciascun elemento dell'architettura, operando in modo indipendente e mediante comunicazione asincrona, non incide in modo bloccante, in caso di malfunzionamento o sovraccarico computazionale (con conseguente degrado delle performance), sul comportamento degli altri moduli del sistema.

3. Interfacce esposte verso sistemi terzi

E' possibile accedere alla coda JMS per inserire nuovi messaggi ed implementare nuove rotte/componenti/funzionalità.