

## Lezione 8 modulo 3

Concludiamo questa serie di moduli dedicati all'integrazione fra sistemi informativi dal punto di vista dell'integrazione delle applicazioni tecnologiche parlando dell'integrazione tramite un approccio a servizi. Il tema dei servizi l'abbiamo trattato nel corso già presentando diversi aspetti sia di modellazione dei sistemi sia dal punto di vista anche dell'utilizzo di diversi sistemi a livelli diversi, ad esempio quando abbiamo parlato del cloud computing, quando abbiamo parlato di software and services piuttosto che platform as a service o infrastructure as a service. Quindi è un concetto di tipo generale che abbiamo già introdotto in vari casi. Qui parleremo dei servizi cercando di risolvere un problema che è rimasto, quando abbiamo parlato delle varie infrastrutture che consentono di gestire l'integrazione di applicazioni tramite un approccio messaggi. Il problema che è rimasto da affrontare è un problema di interfacce, cioè che cosa i diversi sistemi stanno comunicando ad altri sistemi e che tipo di servizi forniscono. Abbiamo parlato di formati dei dati, di scambio di dati, di messaggi, ma abbiamo anche detto che all'interno delle piattaforme di middleware abbiamo anche nascosto della logica applicativa che abbiamo poi cercato di far emergere tramite processi, ma quello che è rimasto caratteristico in tutti gli approcci che abbiamo visto finora è il fatto che dobbiamo costruire degli adattatori per consentire alle varie applicazioni di parlare fra di loro tramite queste piattaforme. Con l'approccio servizi portato nel campo della tecnologia di integrazione noi cerchiamo di risolvere questo problema. Sostanzialmente l'ottica in cui ci si mette è quella di fornire servizi. Un'ottica che può essere all'interno dell'organizzazione ma anche come viene sottolineato qui inter organizzazione; quindi è un paradigma che possiamo utilizzare sia all'interno che all'esterno di una certa organizzazione. Riprendendo il nostro discorso precedente che avevamo fatto di avere uno stack per cui abbiamo diverse tecnologie su cui ci andiamo a basare e abbiamo creato un sistema ERP che ci dà certe funzionalità, quello che possiamo fare è aggiungere uno strato e creare una serie di servizi che il nostro sistema ERP espone. Questi vengono rappresentati da questi simbolini che rappresentano delle operazioni che io posso fare utilizzando sistemi ERP e che vengono caratterizzate dal sistema, quindi vengono offerte dal sistema a chi li vuole utilizzare. In un certo senso stiamo cambiando la logica di integrazione rispetto a quello che abbiamo visto prima, perché abbiamo che ogni sistema dichiara quello che può fare e come è possibile accedere per utilizzare i suoi servizi. Questo è stato sviluppato nel tempo utilizzando approcci tecnologici che sono diversi; qui presentiamo i due principali molto sinteticamente, sono l'ottica principale quello di basarsi soprattutto su un protocollo di interazione che è quello di internet, quindi http come base per trasferimento dei messaggi. Quindi abbiamo la fornitura di servizi, che sono servizi in rete. Il primo approccio tecnologico che è stato proposto in questo ambito è quello dei web services in cui sostanzialmente si è cercato di dare una struttura a quest'ottica servizi utilizzando quello che è anche stato definito come un protocollo che serve a scambiare messaggi dedicati all'interazione dei servizi che è il protocollo SOAP, che è un protocollo che è definito sopra http e utilizza XML come linguaggio per rappresentare poi il contenuto dei servizi, il contenuto dei servizi che viene rappresentato secondo i seguenti diciamo punti di vista e che vengono specificati all'interno del linguaggio web services description language come, diciamo, caratteristiche di un certo servizio. Allora, come abbiamo visto finora nello scambio di informazioni fra applicazioni, abbiamo visto soprattutto il fatto che ci siamo scambiati dei messaggi o un'applicazione doveva mandare delle informazioni tra applicazioni con dei messaggi. Questo approccio qui rimane, vengono definiti dei parametri che sono dei parametri di ingresso e di uscita, quindi io avrò un servizio definito come un'operazione che potrò fare quindi, ad esempio, invio l'ordine che avrà dei parametri di ingresso, potrà avere dei parametri di uscita, ad esempio tipico avere un parametro che è la conferma della ricezione e poi vengono definiti anche dei parametri per la gestione degli errori, che sono necessari ovviamente sempre nello scambio di messaggi. Quello che è diverso rispetto a quello che abbiamo visto finora è che mentre prima ci si concentrava solo diciamo sulla struttura di questi messaggi noi andiamo a definire all'interno del linguaggio tutte le operazioni che un certo sistema ERP offre all'esterno e che possono essere utilizzate da altri sistemi. Queste vengono poi raggruppate anche all'interno di cosiddette porte che vengono utilizzate appunto per definire insieme di operazioni collegate



fra di loro. Questo consente di definire, utilizzando una struttura diciamo e un linguaggio definiti e priori dei servizi che elencando le loro caratteristiche e i messaggi che devono essere scambiati per poterli invocare. Un altro tema che è stato trattato nell'ambito dei web services è il tema di come poter fare a trovare dei servizi, si parla di Registry nel senso che il servizio ovviamente così definito è un servizio abbastanza elementare che io posso utilizzare ovviamente io potrò avere servizi che sono vicini all'ottica applicativa quindi all'ordine oppure servizi anche molto più fini tipo converti la valuta da una valuta a un'altra quindi una singola operazione. Si è parlato ovviamente a lungo di progettazione di servizi e questo è un tema diciamo che non tratteremo nel corso ovviamente il tema della progettazione è nascosto in tutto quello che abbiamo visto finora perché quando abbiamo parlato di definire i dati e scambi dei dati formati adattatori e processi di interazione eccetera e di scambio dei dati all'interno del middleware noi dovremo sempre definire quali sono i messaggi scambiati qual è la loro granularità e quindi il livello di dettaglio per potere poi realizzare le applicazioni. In questo caso siccome il punto di partenza è proprio quello di rendere disponibili i servizi, il Registry è utile per poter avere il catalogo di tutti i servizi che sono definiti all'interno di un insieme di servizi condivisi da parte di più moduli applicativi tecnologici che si scambiano delle informazioni. Il tutto viene inserito in un paradigma di Service Oriented Architecture di cui abbiamo parlato in cui sostanzialmente il fornitore del servizio dichiara la propria disponibilità a fornire certi servizi, ci saranno dei potenziali utilizzatori dei servizi che potranno richiamarli e poi ci sarà effettivamente l'utilizzo dei servizi quindi concetto di binding per cui io avrò l'utilizzatore che richiede il servizio direttamente al fornitore del servizio. Questo paradigma ovviamente ha una struttura di definizione servizi abbastanza complessa, abbastanza complessa perché richiede innanzitutto un protocollo basato su XML per la gestione dei messaggi che chiaramente è legato anche a garanzie di consegna dei messaggi che però appesantisce le interazioni tra diversi sistemi e poi tutta una strutturazione delle varie operazioni all'interno diciamo nella definizione appunto di un web service come insieme di operazioni con certe caratteristiche. Quello che si sta diffondendo spesso nella realizzazione dei sistemi attualmente per consentire la comunicazione fra i sistemi sempre con un'ottica servizi è quello che viene chiamato lo stile rest che sostanzialmente utilizza http come protocollo di base quindi non utilizza un protocollo aggiuntivo per controllare la interazione con i servizi che vengono forniti e poi vengono definite all'interno dei parametri di un normale messaggio scambiato utilizzando http quelli che sono i dati che vengono scambiati e quindi questa struttura che è basata sulla struttura dei parametri dei messaggi che posso scambiare con http viene definita a livello di progettazione servizio e ovviamente potrò avere come ho visto prima nel testo libero ad esempio quando parlavamo di indirizzi fatti con una serie di caratteri in una stringa oppure potranno avere una struttura più strutturata quindi definendo con precisione quali sono i loro elementi e qual è l'eventuale struttura gerarchica degli elementi. Questo attualmente viene fatto utilizzando sempre XML quindi do una struttura che se vogliamo è analoga a quella che veniva definita in WSDL per i parametri utilizzando XML o altri linguaggi tipo Json che consentono di rappresentare le informazioni sempre secondo un'ottica di elementi di informazione all'interno di una struttura che è sostanzialmente gerarchica perché hanno caratteristiche diciamo di maggiore leggibilità e quindi anche sono più leggeri dal punto di vista della loro gestione. Quindi l'ottica servizi sia che sia in un'ottica WSDL sia che siano un'ottica rest che d'altra parte è uno stile non è una specifica tecnologia perché si basa su tecnologie disponibili mi consente di definire per sistemi anche molto complessi ad esempio i maggiori fornitori di ERP hanno, ad esempio SAP, hanno convertito tutti i loro servizi in servizi esposti secondo una logica di interfaccia servizi, questa tecnologia mi consente di definire quali sono i servizi forniti e quindi può questi possono essere utilizzati all'interno di un'ottica di integrazione di applicazione. Una volta definiti i servizi rimane il fatto che noi vogliamo supportare poi lo scambio diciamo di messaggi fra questi servizi, ovviamente li possiamo chiamare direttamente una volta che abbiamo definito un indirizzo a cui chiamare il nostro servizio possiamo utilizzare http e possiamo chiamarlo direttamente. In realtà la situazione diventa un po' più complessa perché noi dobbiamo in generale consentire lo scambio di messaggi fra diversi sistemi e quindi ci si ripropongono alcuni dei problemi tecnologici che avevamo menzionato prima, vogliamo garantire la consegna dei messaggi vogliamo garantire aspetti di sicurezza abbiamo detto che abbiamo bisogno di un registry per trovare i



servizi da utilizzare e quindi ci saranno delle regole per l'identificazione dei servizi e così via e questo viene rappresentato, qui vediamo una rappresentazione seguendo quella che la notazione che abbiamo visto per le architetture in bot viene utilizzato un elemento che è un elemento architetturale tecnologico chiamato Enterprise Service Bus. Qual è il significato di questa figura? noi vediamo sostanzialmente quello che avevamo visto nella rappresentazione di architetture utilizzando il modello architetturale di bot in cui abbiamo dei front end che ci consentono da 1 a 3 di interagire con la partecipante B, 4 e 5 con la partecipante I. Abbiamo una serie di moduli applicativi o di database ma vedrete che anche i data base li presenteremo come servizi e abbiamo quindi la nostra architettura in termini di front end e di back end e gli elementi tecnologici all'interno del back end. Finora per collegare tutti questi moduli abbiamo sempre utilizzato una notazione che ci diceva quale modulo colloquiava con un altro modulo quindi avevamo utilizzato collegamenti sostanzialmente. In un'ottica di fornire un supporto all'integrazione di applicazioni ovviamente questa è un'ottica di integrazione di tipo punto a punto. Quello che possiamo fare anche in questo caso è definire un'architettura a servizi anche all'interno di un'organizzazione, quindi definiremo i nostri servizi di front end con cui i vari moduli applicativi back end potranno interagire, i vari servizi per i vari moduli di back end e anche per i moduli tecnologici. L'interazione poi sarà tramite appunto un modulo di integrazione che quindi è anch'esso ovviamente visto all'interno della famiglia dei middleware di integrazione che consente di istradare i messaggi secondo la logica di interazione che viene definita all'interno diciamo delle varie applicazioni che sono sviluppate all'interno di ciascun modulo. Abbiamo all'interno del registry anche la possibilità di cercare servizi in modo da instradare eventuali messaggi a servizi secondo un'ottica di ricerca dei servizi che vogliamo utilizzare che ci porta su un'invocazione dinamica dei servizi e quindi non ha collegamenti fissi ma possibilmente anche a collegamenti di tipo variabile. Quando parliamo di servizi possiamo anche utilizzare un'infrastruttura servizi disponibili ad esempio il cloud. Vediamo qui sempre utilizzando la notazione di bot un altro esempio, noi possiamo avere messo in cloud alcuni dei nostri servizi, ovviamente abbiamo indicato appunto questo con la tipica nuvoletta che ci dice che appunto abbiamo i nostri servizi non all'interno del nostro confine diciamo dell'organizzazione dal punto di vista dell'infrastruttura, ma abbiamo un'infrastruttura in cloud e possiamo chiamare dei servizi di back end che abbiamo messo in cloud, di IBMS oppure direttamente; ovviamente questo è un caso in cui avremmo utilizzato come infrastruttura sostanzialmente un'infrastruttura di piattaforma ma potremo anche chiamare direttamente dei servizi; ad esempio potremmo avere il nostro ERP usato come servizio da parte del fornitore esterno utilizzato come servizio in cloud che viene chiamato dalle nostre applicazioni che vengono utilizzate internamente. Per concludere c'è un aspetto da sottolineare che abbiamo cominciato a toccare dicendo che possiamo utilizzare ad esempio servizi erp in rete da parte di altri fornitori. In realtà noi stiamo andando verso l'interazione servizi in generale quindi quando parliamo di interazione fra diverse organizzazioni noi possiamo utilizzare una logica servizi quindi quello che abbiamo visto non è solo per gestire le interazioni fra moduli all'interno di un sistema informativo composto da diverse applicazioni tecnologiche all'interno di una singola azienda oppure dei suoi fornitori se vogliamo utilizzare i servizi all'esterno, ma abbiamo la possibilità di comporre diversi servizi da parte di diversi partecipanti. Ovviamente l'ottica può rimanere quella di un processo di composizione di servizi che vengono composti secondo una logica di un processo di interazione quindi sequenze di messaggi consentite e una logica di avanzamento del processo, utilizzando una tecnologia servizi, quindi utilizzando l'interfaccia, tipo quella che abbiamo visto in stile rest che possono essere utilizzate per chiamare servizi non solo all'interno della propria organizzazione dei propri fornitori ma anche fornito interni ma anche fornitori di servizi esterni. Anche questa è una ottica di integrazione e una ottica d'integrazione che si sta diffondendo molto soprattutto quando si parla di API (Application Programming Interface) che in realtà spesso sono proprio servizi che si possono chiamare secondo una certa struttura definita che dice quali sono i formati dei messaggi che si vogliono inviare e ricevere e che consentono di chiamare servizi da parte di fornitori esterni per comporre le proprie applicazioni. Quindi in questo modulo abbiamo visto come l'approccio servizi che abbiamo definito in generale nel nostro corso come un approccio che può essere utilizzato per l'interazione con vari tipi di sistemi, come anche uno specifico approccio tecnologico per cui



dobbiamo definire l'interfaccia dei servizi lo possiamo fare utilizzando WSDL oppure un approccio di tipo rest per una chiamata diretta dei servizi; abbiamo visto che dal punto di vista tecnologico abbiamo una tecnologia che è quella dell'Enterprise Service Bus che ci dà quelle funzionalità di gestione della ricerca dei messaggi della consegna dei messaggi dell'interazione fra sistemi tramite appunto un approccio Bus a cui tutti si collegano per fornire i servizi che vengono forniti e per ricevere diciamo messaggi che guardano servizi forniti da altri partecipanti e che ci consente di realizzare le diverse interazioni all'interno di un'architettura che sono state indicate in questo caso nell'ambito di una architettura di tipo tecnologico funzionale rappresentata con l'annotazione di bot.

