GREAT INDIAN DEVELOPER SUMMIT

www.developersummit.com

24 / 04 / 2009

Spring Integration

- A cheaper & better alternative

Srini Kumar

http://www.srinikumar.com

Satyam
What Business Demands.

Agenda

- A brief on EAI
- EAI Options / patterns
- SOA Stack / ESB
- Need for a simple cleaner solution
- What is Spring Integration
- Oops did I waste all my money
- Conclusion
- References

Brief on EAI

- Enterprise Application Integration (EAI) is an application of technology defined as the integration of data and services between applications.

- Need for EAI
  - ❑Impossible to have a single application catering to every IT need of an Enterprise.
  - ❑Multiple Applications with different technologies and protocols need to collaborate to help the business run smoothly.

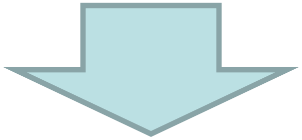
Options / Patterns

based on EAI Patterns from Gregor Hohpe

- File Transfer
  - ❑ Have each application produce files of shared data for others to consume, and consume files that others have produced.
- Shared Database
  - ❑ Have the applications store the data they wish to share in a common database and share tables.
- Remote Procedure Invocation
  - ❑ Have each application expose some of its services so that they can be invoked by others Example: EJB Services, SOAP
- Messaging
  - ❑ Have each application connect to a common messaging system, and exchange data and invoke behavior using messages Example: JMS architecture

Will EAI options suffice

- As applications become old integration become difficult. Integrating each end point will create a Spaghetti
- File Transfer, Shared Database, RPC and Messaging are disparate because not one of them is fit for all approaches
- JMS, REST and SOAP are platform agnostic and they fail when they have FTP end points



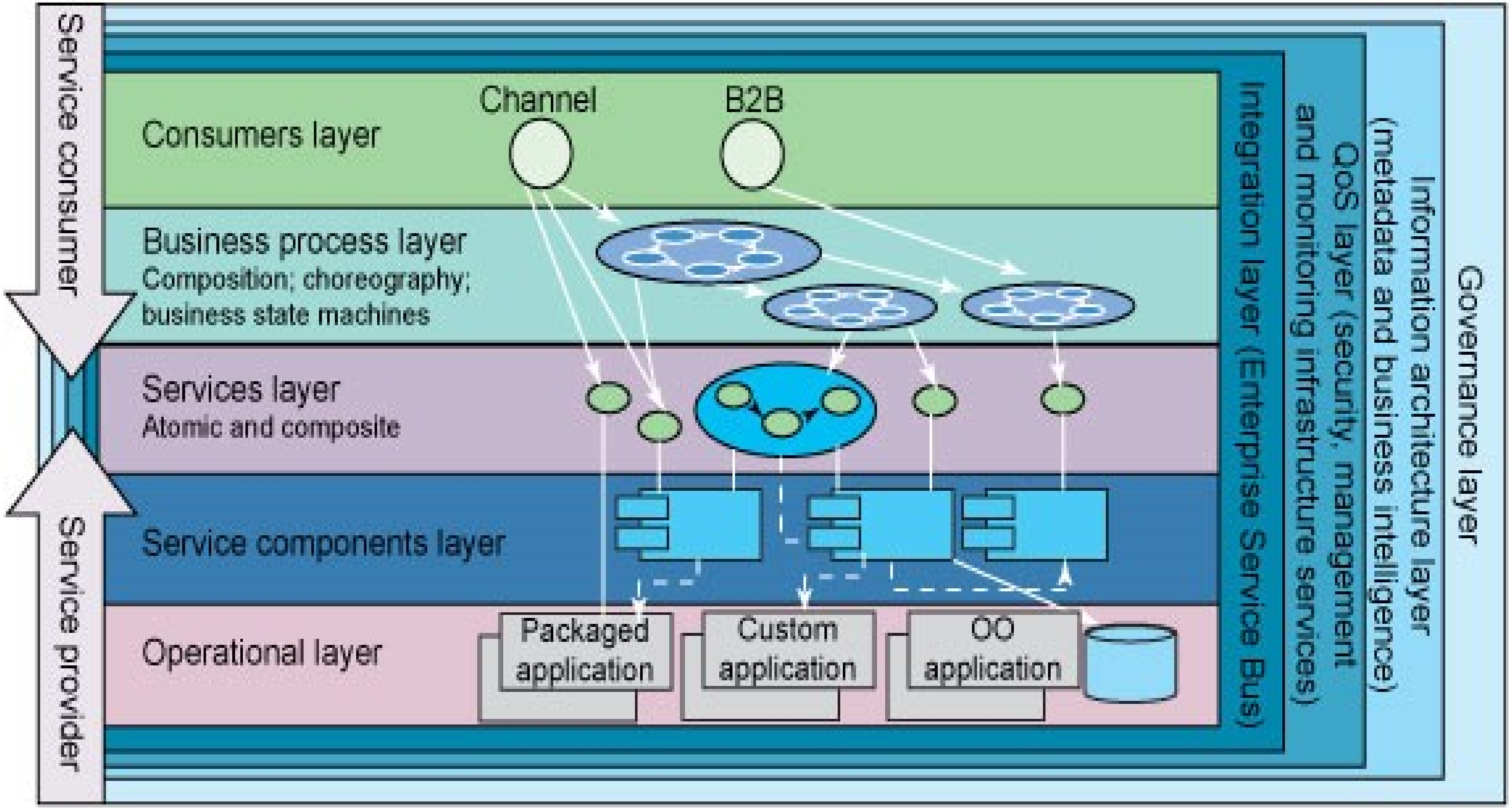
ESB

Middle ware was born to solve all these issues

ESB’s provide

- Flexible Routing
- Messaging - Transformation
- Mediation – using Adapters
- Complex Event Processing
- Consume and provide services – Invocation
- Helps in logging, auditing and authentication

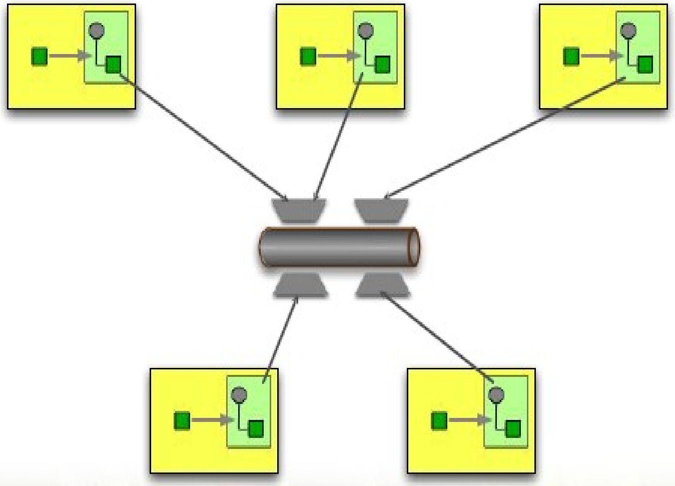
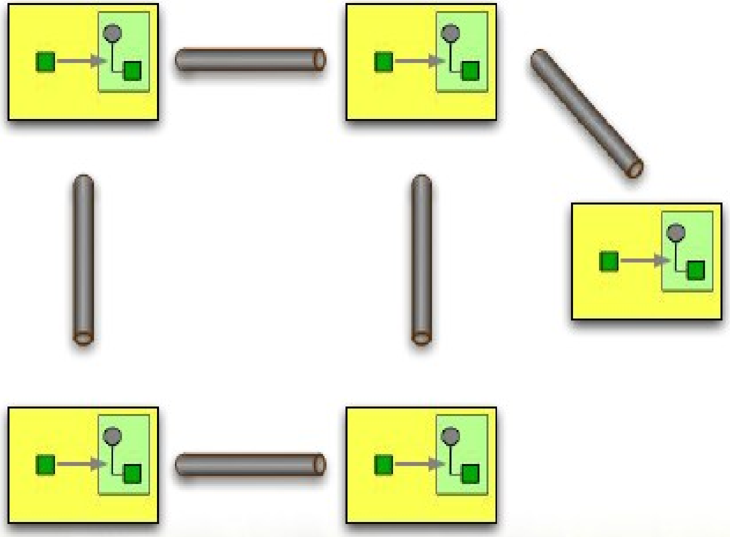
ESB & other products in a Reference Architecture


Service consumer
Service provider
Consumers layer
Channel
B2B
Business process layer
Composition; choreography; business state machines
Services layer
Atomic and composite
Service components layer
Operational layer
Packaged application
Custom application
OO application
Integration layer (Enterprise Service Bus)
QoS layer (security, management and monitoring infrastructure services)
Information architecture layer (metadata and business intelligence)
Governance layer

Is there an Alternative

- Spring Integration the cheaper alternative
  - Don’t need to spend on SOA stack, no installations
  - Can replace heavy ESB solutions
- Spring Integration is better
  - Built on top of Spring Framework; Spring Integration applications are simple Java programs that are configured using Spring's schemas.
  - Benefits from the same dependency injection and aspect runtime available to regular Spring bean
  - With Spring Integration, the application context *is* the bus. The bus exists as long as the context is available.

How different than ESB

ESB	Spring
Service bus exists between disparate systems. Becomes spaghetti over time	Spring Integration has multiple light weight messaging buses between disparate systems. This helps integration of new systems not affecting the entire architecture and help in easy maintenance as well.
	

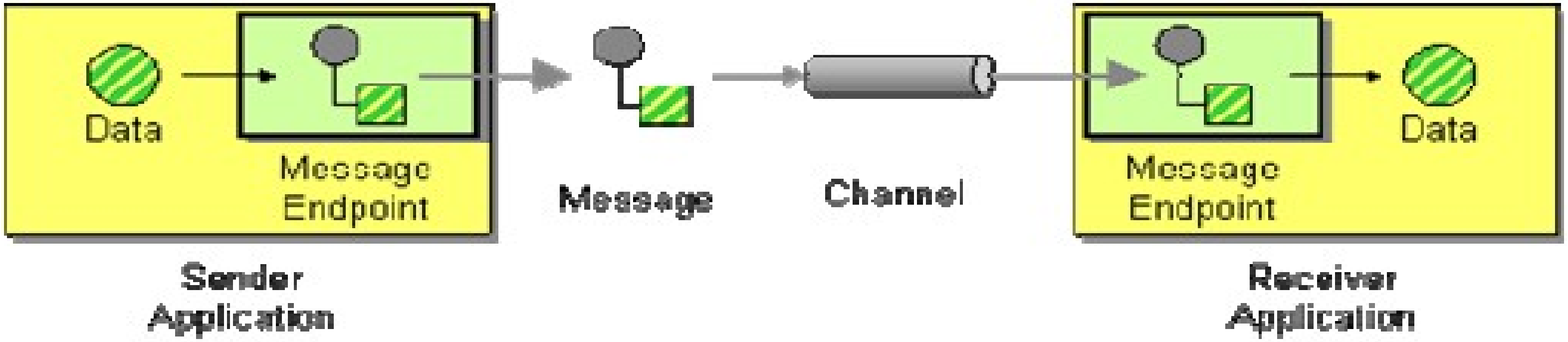
Spring Integration or ESB still confused

Comparisons…

- ESB are highly robust and configuration based
- Spring is focused on Integration, not on ESB
- ESB are lot heavier and focuses on SOA deployment rather than integration
- ESB should be used for complex routing
- ESB has lot more integration options as of now

Spring Integration - Terminology

- Spring integration deals with notion of messages traveling thro channels
- A Message Channel represents the "pipe" of a pipes-and-filters
- Channel Adapter connects a Message Channel to some other system.
- The message starts its life at an endpoint, typically greeted by an adapter. Endpoint connects the code to the messaging system.



- As the message moves through the processing pipeline, it may be transformed, routed to other channels, split up, responded to, or aborted and sent on a dead message channel, all inside the bus.

Spring Integration - Advantages

- Ease of adoption
  - Programming model familiar to existing spring users.
  - Easy Integration of messaging to existing application.
  - Integrates seamlessly with Spring Security and Spring Web Services.
- Lightweight
  - Does not require install of heavyweight software.
  - Quick to start and stop as part of the spring Application Context.
- Non Intrusive
  - Framework decouples components from messaging infrastructure.
- Testable
  - Simple unit testing of POJO components.
  - Quick and simple integration testing.

Spring Integration
- Need improvements

- Still new, needs more time to mature
- There are several vendors and solutions, Spring needs to try that out where adapters are well established, Still adapters such as SFTP are not supported
- Persistent channel implementation combined with Spring Integration has a few limitations. Need to write your own logic for now.

What else to expect

- Messaging gateway for Spring MVC
- Integration with Spring batch
- Spring security for end points
- Adapters for Spring application events
- Different adapters such as SFTP are being worked on

References

- *Enterprise Integration Patterns (the book)* – by Gregor Hohpe and Bobby Woolf (Addison Wesley 2004) – website: http://www.eaipatterns.com
- http://www.springframework.org/spring-integration
- Getting started with Spring Integration by Joshua Long - http://www.infoq.com/articles/Spring-Integration-Joshua-Long
- http://www.ibm.com/developerworks/library/ar-archtemp/
- http://www.chariotsolutions.com/slides/pdfs/ete2008-intro-to-

Thank you

Any Q&A