

Refactoring legacy applications for SOA using Spring Technologies

Oleg Zhurakousky

Sr. Software Engineer

SpringSource, a division of VMware

ozhurakousky@vmware.com

- SOA
 - What SOA “IS NOT”
 - What SOA “IS”
- Three principles of SOA
 - Strategy first programming model
 - Process Life-cycle dynamics
 - Process Collaboration
- Demo
- Q&A

Severely Overused Acronym

SOA knows what you did last summer, and is disappointed that it wasn't SOA. - soafacts.com ;)

SOA Challenges



- Focus of Enterprise Architecture is now on Software
 - Internet, Virtualization, and Cloud have made hardware and network ubiquitous (and abstract)
- Best Practices for Software Architecture are still rooted in solid Application Architecture
 - High Cohesion, Loose Coupling
- Traditional Independent Software Vendors can't sell you solid Application Architecture
 - There is no server for that

SOA "IS NOT"

-
- An expensive suite of software products reflective of your legacy IT investment
 - Static
 - Predictable
 - Definitely NOT a Product you can purchase!

- An architectural pattern reflective of your business
- Agile
 - Data is only a means to an end, it is in the successful interpretation of data that a business lives or dies
- Evolutionary
 - The successful interpretation of data is inexorably linked to the evolving environment in which the business is competing
- A journey
 - Safely and securely nurturing and advancing the IP of a business is indeed akin to "survival of the fittest" within the free market economy

Must withstand the test of time



Foundation and Three principles of SOA

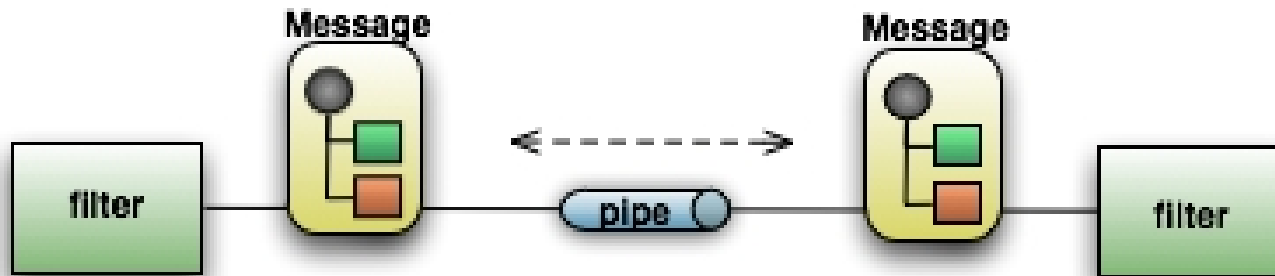


- The foundation of SOA is in *highly cohesive, loosely coupled* processes built on the following 3 principles:
 - ***A strategy-first programming model***
 - ***Process life-cycle dynamics***
 - ***Loosely-coupled process collaboration***

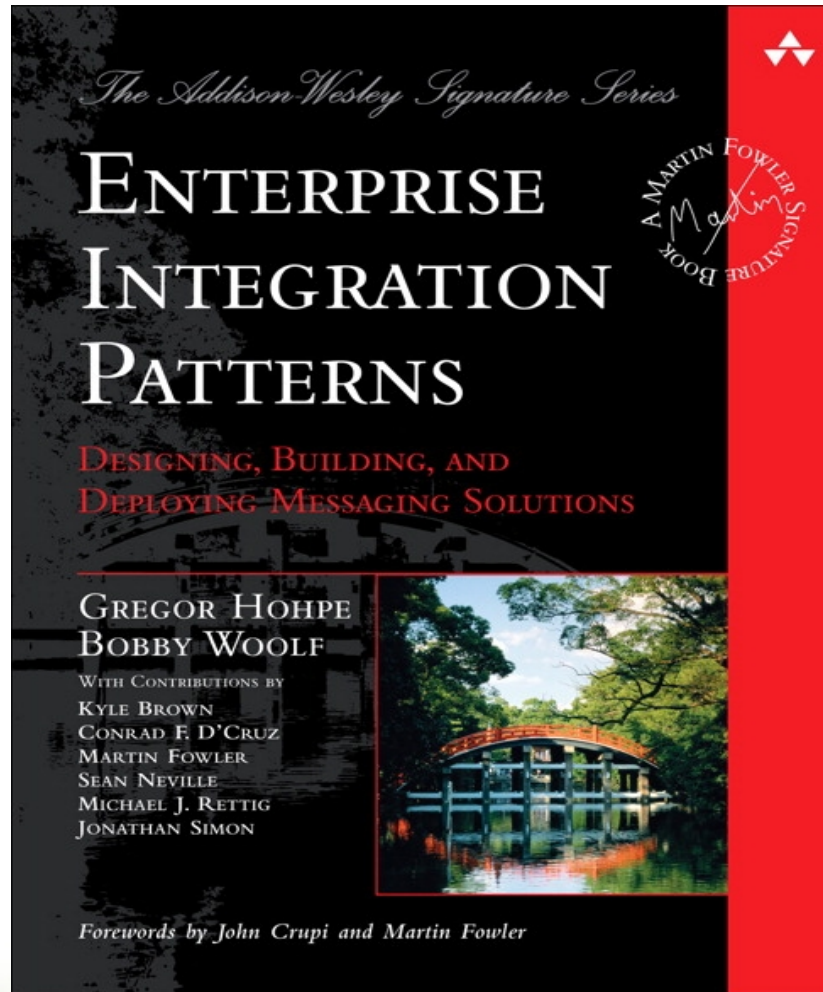
2

Modern Event Driven Architecture

- A business' enterprise architecture is a **living organism**, much like the business it supports
 - A set of composite processes linked together in a linear fashion through the use of “pipes”, “filters” and rule-based pathing



Enterprise Integration Patterns



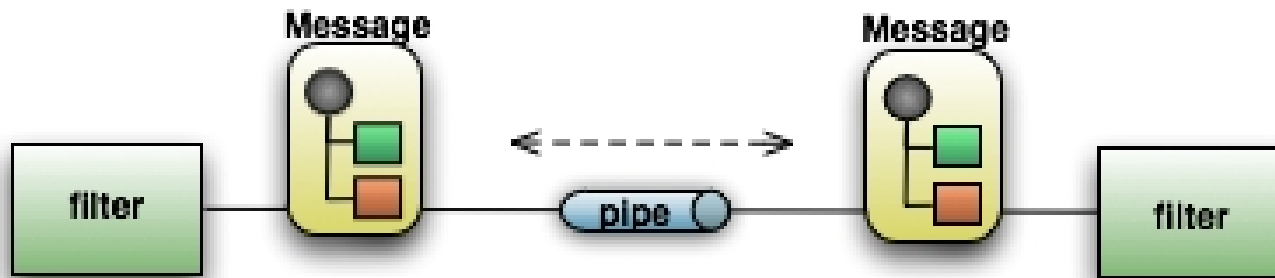


Demo

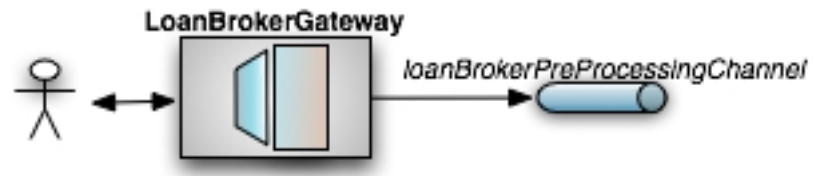
Loan Broker

Pipes and Filters

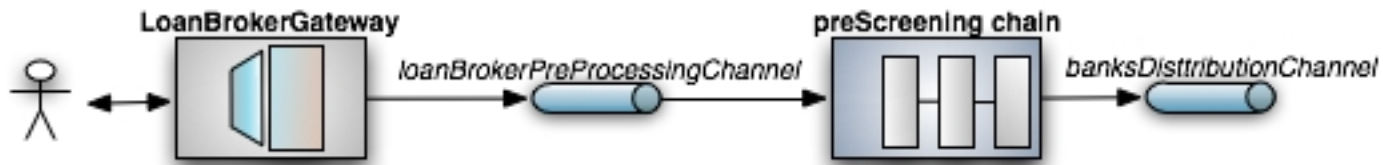
- Pipes (Channels) connect Filters (Endpoints) to exchange messages



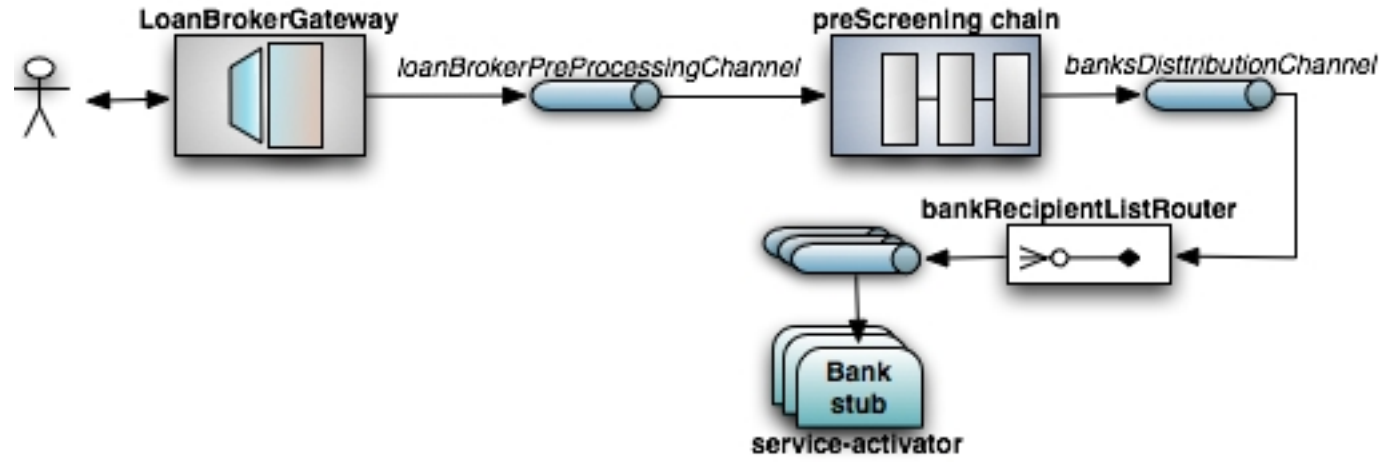
Step 1 - (Messaging Gateway)



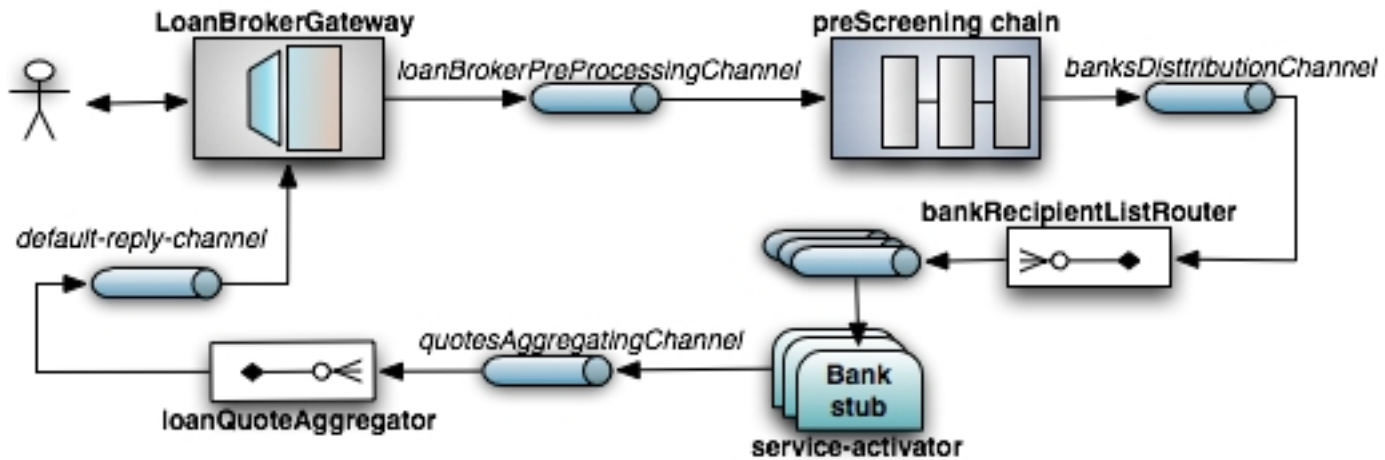
Step 2 - (Composed Message Processor & Content Enricher)



Step 3 - (Recipient List router)



Step 5 - (Aggregator)



- Loan Broker source and detailed explanation is available here:
<http://blog.springsource.com/2010/03/18/eip-loan-broker-reference-implementation-part-1/>
- Also see blog from Gary Russel here:
<http://blog.springsource.com/2010/03/29/using-udp-and-tcp-adapters-in-spring-integration-2-0-m3/>

Q & A