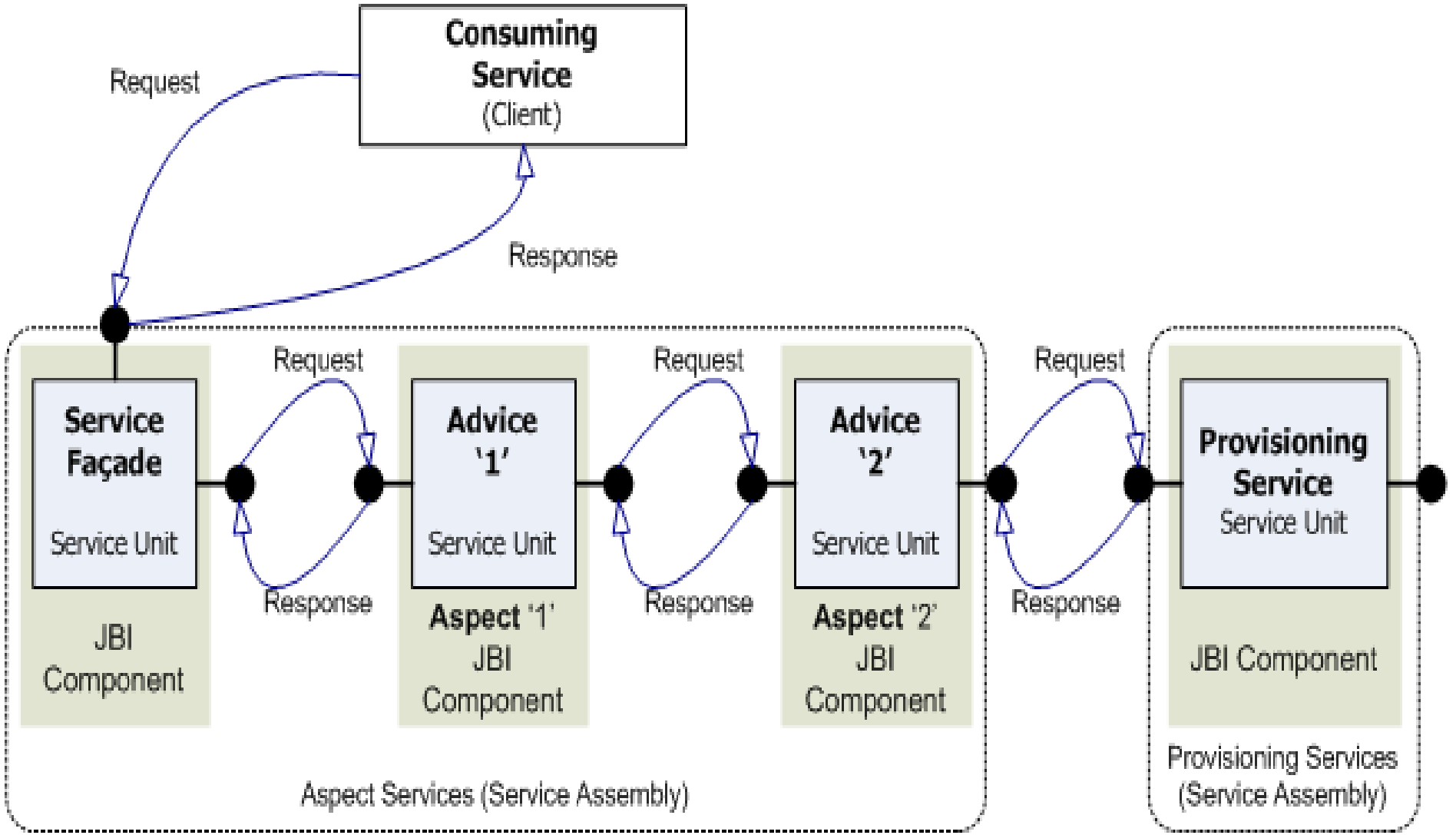
Aspect Orientation for Mashups

Gopalan Suresh Raj
Gopalan.Raj@sun.com
http://blogs.sun.com/gopalan

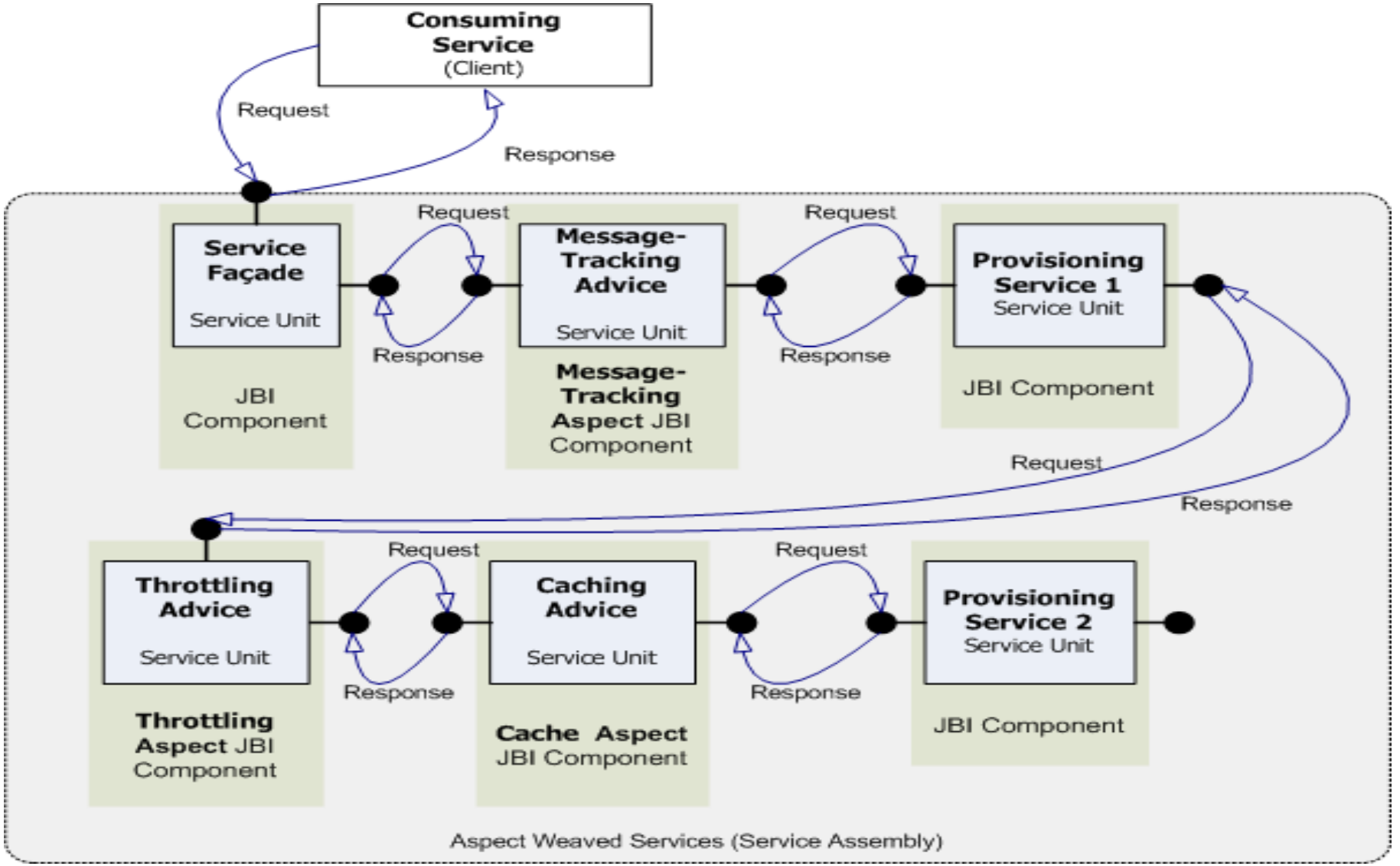
Aspect Orientation for Composite Apps

- Aspects help to encapsulate cross-cutting expressions in one place.
- By applying an Advice, at various points in an application called Join-Points, Aspects can alter the behavior of the non-aspect parts of a software application.
- There are two types of aspect patterns that are addressed:
  - Facade Pattern
  - Aspect-Weaving Pattern

Facade Pattern



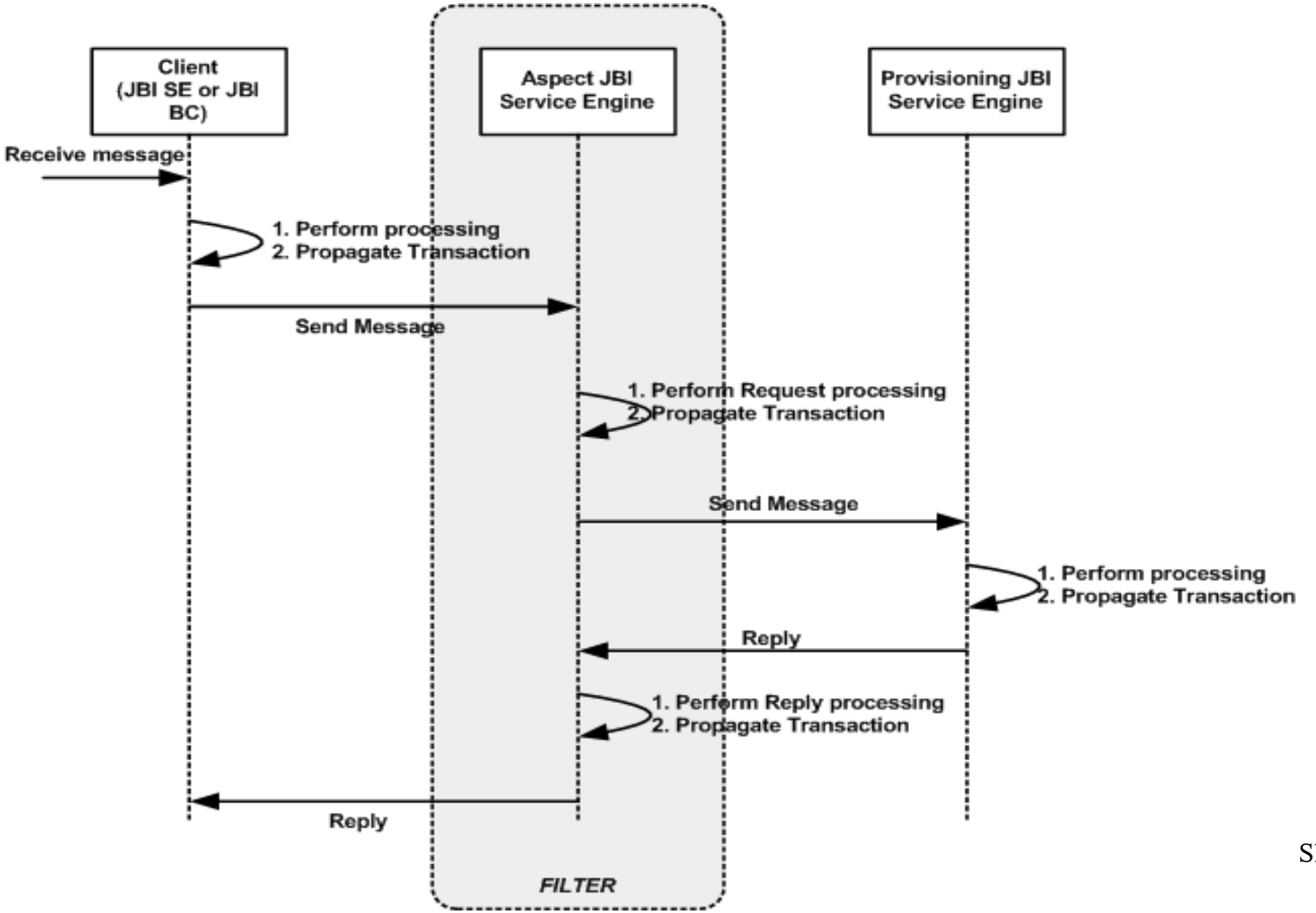
Aspect Weaving Pattern



Aspect Message Exchange Patterns

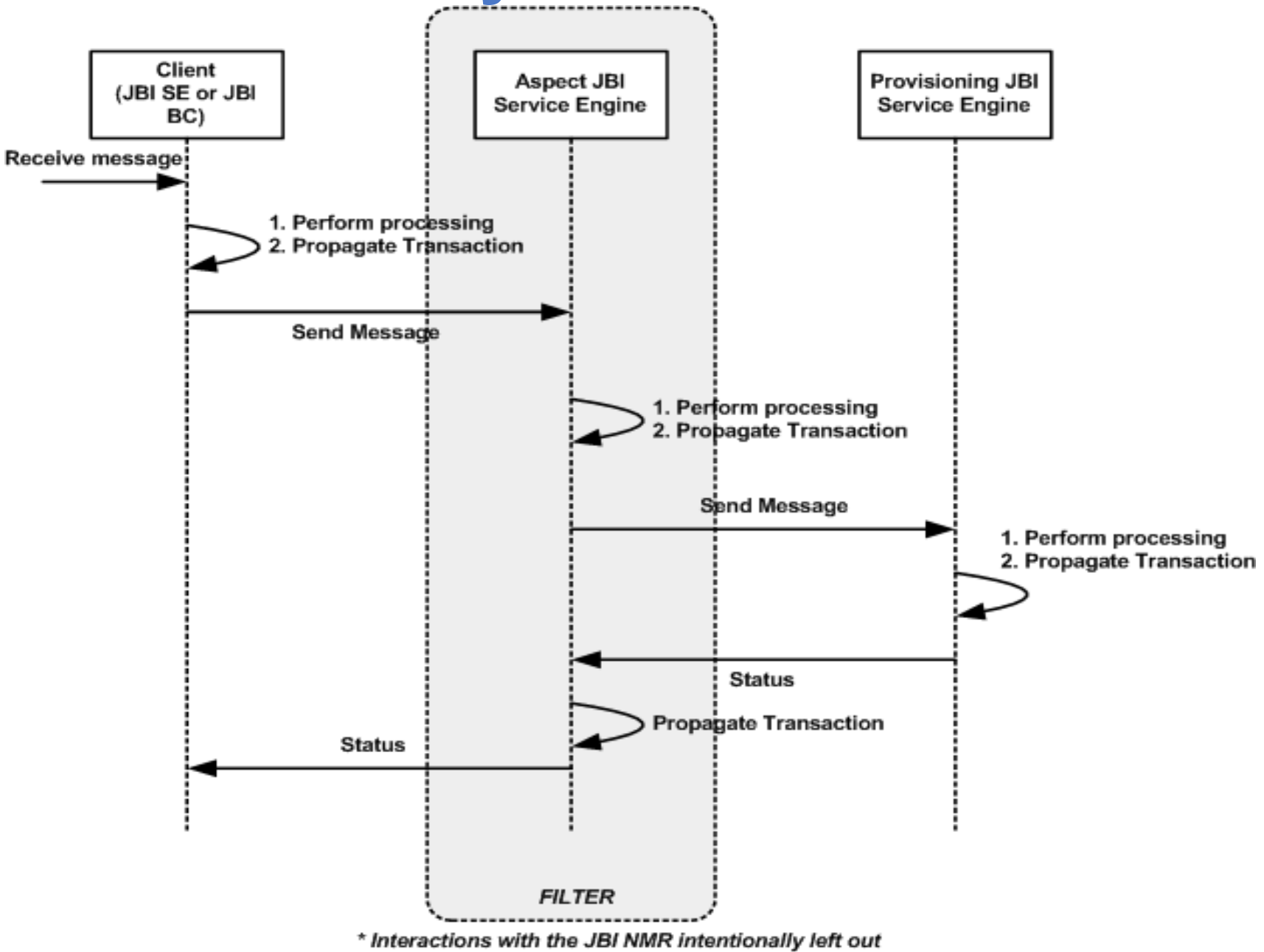
- To facilitate injection of aspect advices to a composite application, we extend the JBI JSR-208 Message Exchange patterns and add a couple of new message exchange patterns based on the pre-existing request/reply and one-way patterns.
- They are:
  - Filter Request/Reply Pattern
  - Filter OneWay Pattern

Filter Request/Reply Pattern



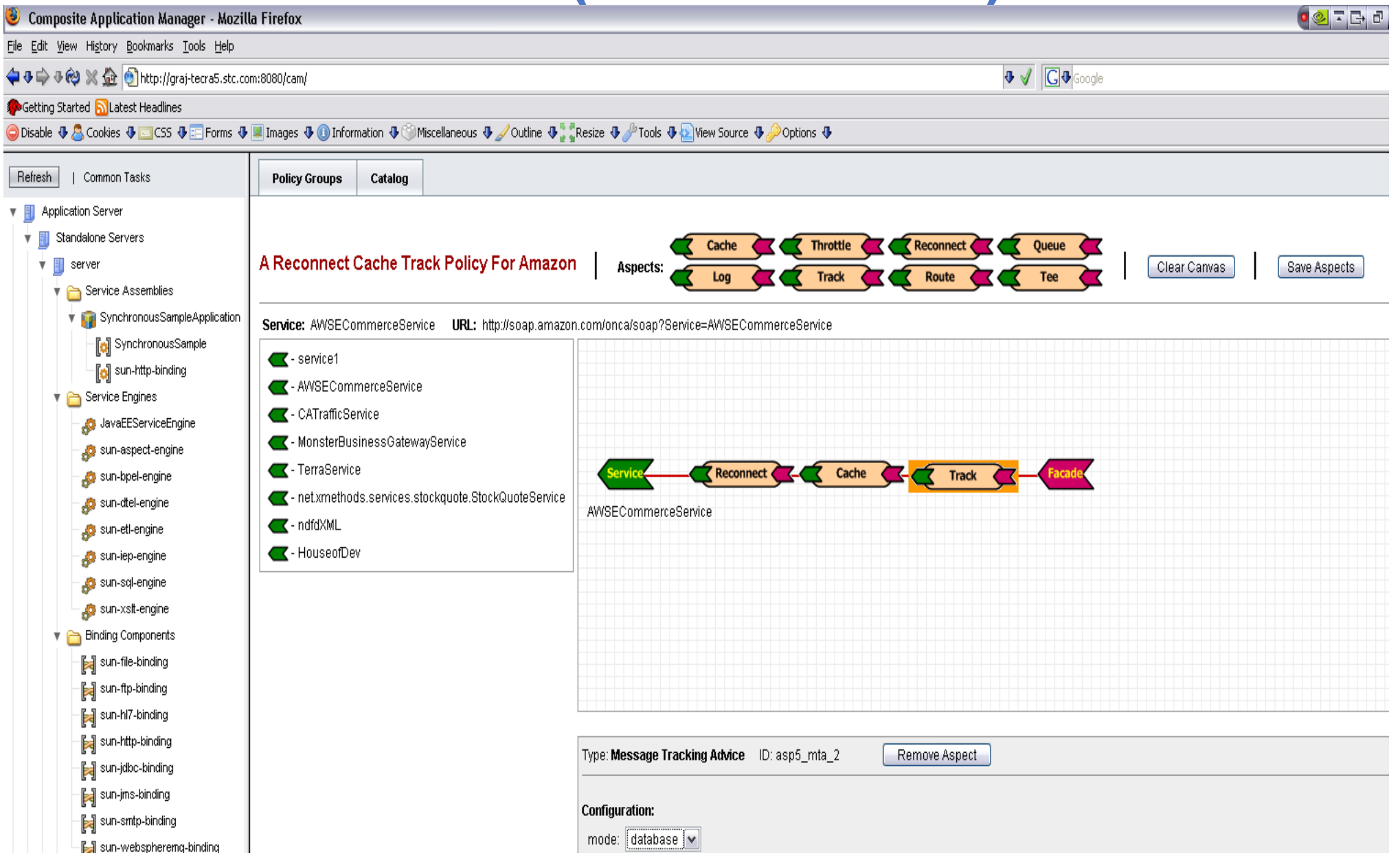
** Interactions with the JBI NMR intentionally left out*

Filter One-Way Pattern



** Interactions with the JBI NMR intentionally left out*

Facade Pattern (screen-shot)



Caching Aspect

- There are a couple of patterns that the Cache aspect supports:
  - Write-Through Cache (a.k.a. Transparent Cache) pattern
    - Cache is placed between the client and the service that is being invoked
  - Cache-Aside pattern
    - Manages caching of certain key elements and their values contained within the messages that flow through the write-through cache and use it in building features into any composite app.

Throttling Aspect

- This aspect reduces the load on services based on policies which determine:
  - When to delay access
  - When to refuse access
- This pattern ensures the Throttling policy acts between the client and the service that is being invoked.

Retry (a.k.a.) Auto-Reconnect

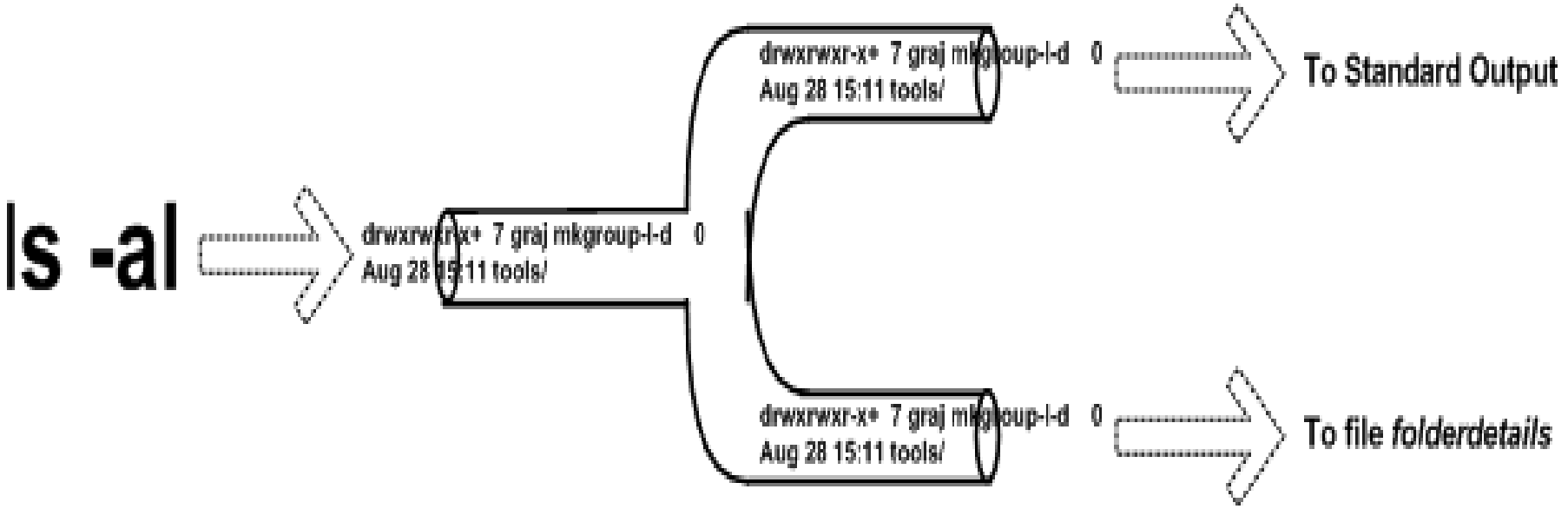
- In a request/reply scenario, if the service provider is unavailable, one has to block the client until the service is up again and a response is received.
- The Auto-Reconnect Aspect would keep trying to make the call to the destination every configurable seconds for another configurable amount of time until the service provider becomes available again or the maximum number of retries is reached based on the configuration.

Queuing Aspect

- In a one-way web service request scenario, if the service provider is unavailable, one can use a Queuing Aspect to queue the requests until the service provider becomes available again.
- The Queuing Aspect would keep trying to make the call to the destination every configurable seconds for another configurable amount of time until the service provider becomes available again or the maximum number of retries is reached based on the configuration.

Tee Aspect

- In some scenarios, you want to redirect the same message to two different components.
- The Tee Aspect Service Engine core introduces a T-junction that facilitates message redirection to two different components at the same time.



Content-Based Routing Aspect

- The Content-Based Routing Aspect is used to forward a message to the correct destination based on the content of the message payload.
- A typical scenario for this may be that on receipt of a purchase order, the message needs to be routed to the appropriate order management system, based on the type of item that has been ordered (which will be obtained by parsing an XPath expression in the message payload).
- The routing will happen based on a configurable Rule Set which can be configured from the Web Console.

Logging Aspect

- The Logging aspect can be used to control logging output. The logging Level objects are ordered and are specified by ordered integers. Enabling logging at a given level also enables logging at all higher levels.
- The levels in descending order are:
  - SEVERE (highest value)
  - WARNING
  - INFO
  - CONFIG
  - FINE
  - FINER
  - FINEST (lowest value)
- In addition there is a level OFF a level ALL

Message Tracking Aspect

- The Message Tracking aspect can be used to track messages flow through the system.
- Message Tracking will happen based on a configurable Rule Set which can be configured from the Web Console.

Resources

- My Blogs
  - http://blogs.sun.com/gopalan
- Aspect Orientation Developer Wiki
  - http://www.glassfishwiki.org/jbiwiki/Wiki.jsp?page=AOSD
- Open ESB website
  - http://open-esb.org

Aspect Orientation for Mashups

Gopalan Suresh Raj
Gopalan.Raj@sun.com
http://blogs.sun.com/gopalan