Swordfish
SOA Runtime Framework

Aleksander Pohl

Katedra Informatyki, Akademia Górniczo-Hutnicza

4 maja 2009

Agenda


SWORDFISH

*„The goal of the Swordfish project is to provide an **extensible SOA runtime framework** based on the proven Eclipse Equinox runtime technology. The framework is designed to be complemented by additional open source components such as service registry, a messaging system, a process engine, etc. to form a comprehensive open source SOA runtime environment based on both established and emerging open standards."*

Swordfish

- Allows application developers and system integrators to build their own ESB
- Built upon
  - Apache ServiceMix
  - Apache CXF
  - Eclipse Equinox

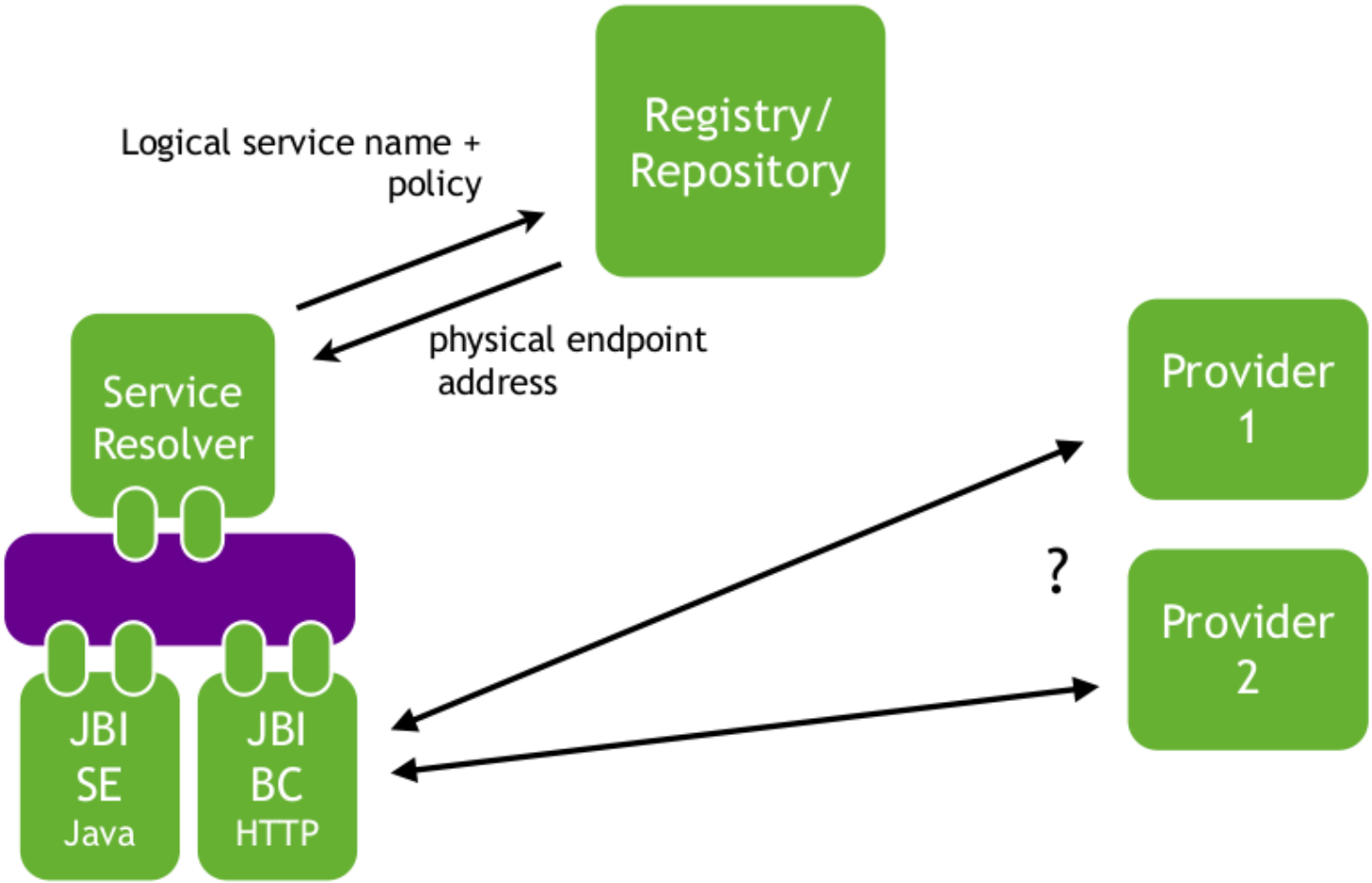
Features

- Integrates dynamic Service Registry
- Based on Distributed ESB pattern
- Remote configuration mechanism
- Extensive monitoring capabilities
- OSGi as a component model
- JBI 1.0/2.0 for messaging abstraction

Service registry

- Consumer refers to provider by logical identifier
- Logical identifier signifies the service's interface
- Consumers Policy covers its non-functional capabilities and requirements
- Service Registry selects matching provider and calculates an effective Policy
- Service Registry will be complemented by a Service Repository that allows all SOA-related artifacts to be stored in reliable and traceable manner

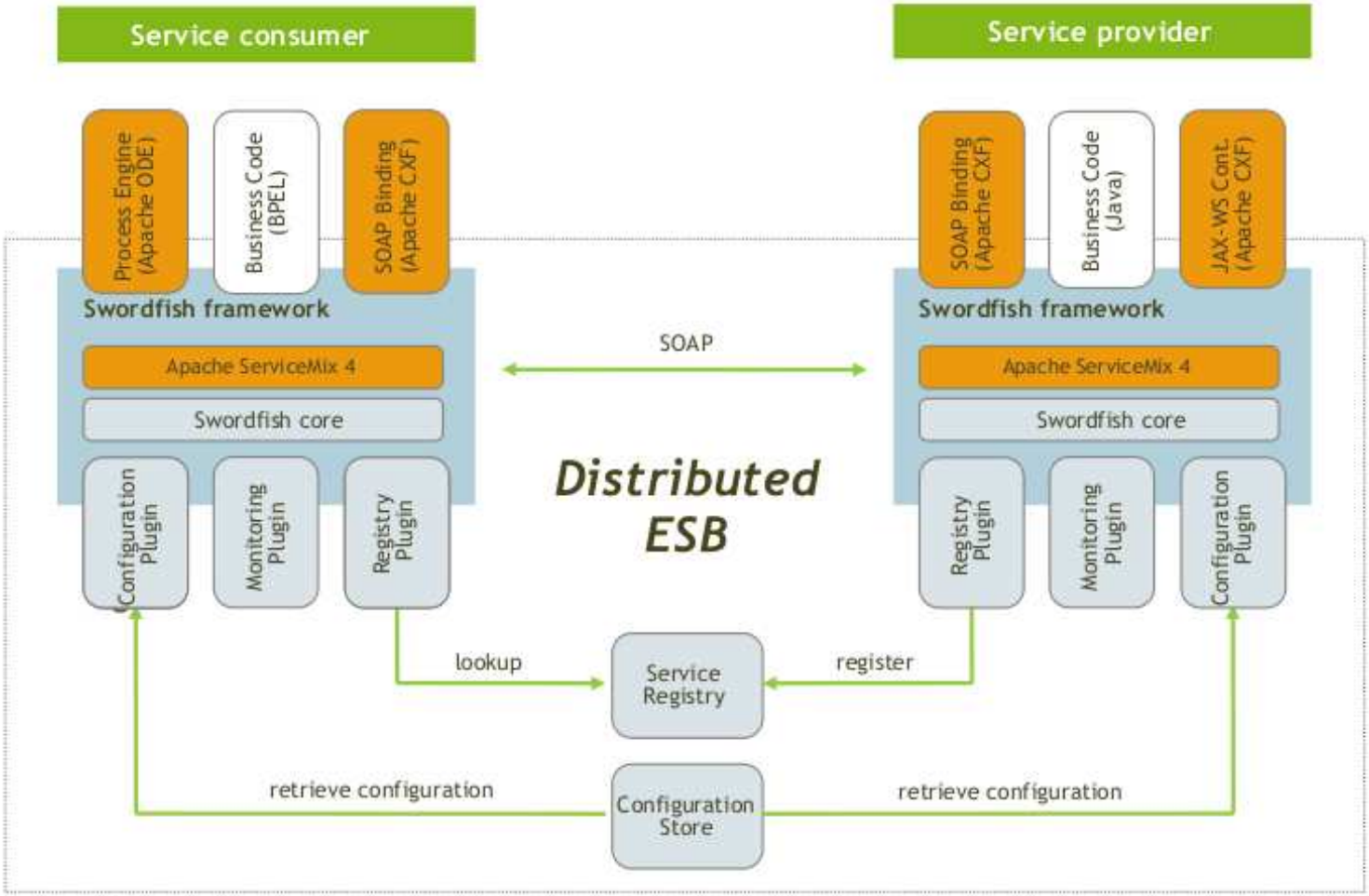
Service registry



Distributed ESB

- No central components are needed after the initial communication set up
- The whole system is much more scalable
- Increased administration and management complexity are compensated for by
  - Remote configuration mechanism
  - Extensive monitoring capabilities

Distributed ESB



OSGi and JBI

- OSGi provides
  - component model
  - deployment mechanism for modules
  - class loading system
- Subset of the JBI
  - messaging abstraction
  - message routing between components
  - JBI 1.0 component model dropped in favor of OSGi

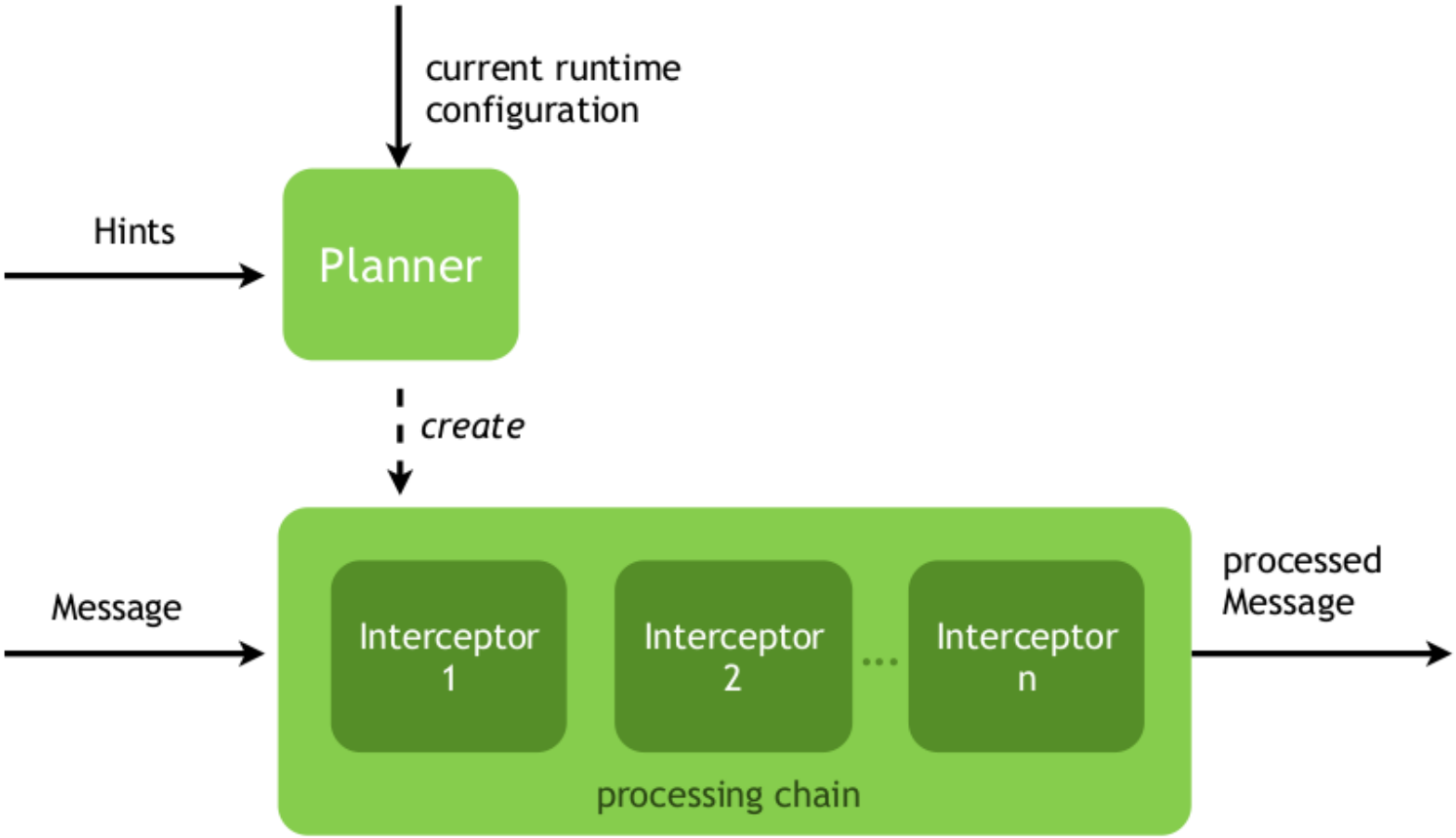
Extending Swordfish

- General interceptor API
- Service resolver API
- Event API
- Configuration API

Interceptor API

- Swordfish hooks into the Normalized Message Router in Servicemix, intercepts all messages flowing back and forth and feeds them into an interceptor chain
- Interceptors in that chain can do (almost) anything with a message: read and modify payload, read and modify properties, re-route etc.
- The interceptor chain is dynamically built from all registered instances of Interceptor
- A component called Planner is responsible for building the chain
- A default implementation of Planner (DefaultPlanner) is part of the Swordfish core

Planner


current runtime configuration
Hints
Planner
create
Message
Interceptor 1
Interceptor 2
Interceptor n
processed Message
processing chain

Service resolver API

- The Service Resolver API builds upon the general interceptor API
- A special kind of interceptor (ServiceResolverInterceptor) is provided as part of the Swordfish core
- The ServiceResolverInterceptor is reponsible for translating the (logical) service interface name carried inside a message exchange into a physical endpoint address where the message exchange will be ultimately sent to
- The ServiceResolverInterceptor collaborates with a number of other (replaceable) components in order to perform its task

Event API

- The Event system is the basis for Swordfish's monitoring capabilities
- The Event API builds upon the OSGi EventAdmin Service
- Swordfish core generates Events of type TrackingEvent for each MessageExchange flowing through the NMR
- Other pre-defined Event types include OperationEvent and ConfigurationEvent

Configuration API

- The Configuration API builds upon the OSGi ConfigurationAdmin service
- A ConfigurationAgent receives configurations from a ConfigurationSource (possibly a remote one) and provides them to other components through the OSGi Configuration Admin service
- A component that implements ConfigurationConsumer receives updated configurations as they become available

Releases

	Galileo June 2009 Release 0.9	Eclipse Summit Europe November 2009 Release 1.0	EclipseCon March 2010 Release 1.1	Eclipse Release Summer 2010 Release 2.0
ESB	Swordfish framework basic plugins basic tool support	Integration with open source security framework	Test support	Full enterprise ESB
Process Orchestration	BPEL process engine	Integration with open source BPM suite (Spagic)		
Registry & Repository	Basic runtime registry	Advanced runtime registry/service locator basic repository	Service repository w/service lifecycle management	Full enterprise service repository
Management	JMX-based management	Plugin for integration into Hyperic HQ		
Data Integration		Plugins for open source ETL tools (e.g. Talend)	Plugins for open source EDI tools (e.g. Smooks)	
Service/Business Activity Monitoring		Service activity reporting	Complex event processing	Full business activity monitoring

Resources

- http://www.eclipse.org/swordfish
- http://wiki.eclipse.org/Swordfish