Decorating Your SOA Services With Governance Enforcement Contracts

Michael Wheaton

Principal Engineer
SOA Center of Excellence
Sun Microsystems, Inc.

TS-8440

The Goal of This Talk

Discuss the importance of SOA Governance and how to architect a solution for your Java™ Platform, Enterprise Edition (Java™ EE platform) environment.

Agenda

Importance of SOA Governance

Architecting a service governance solution

Transition strategies for Java EE platform

SGF Governance solution

java.sun.com/javaone

Agenda

Importance of SOA Governance

Architecting a service governance solution

Transition strategies for Java EE platform

SGF Governance solution

What Is SOA Governance?

- The ability to organize, enforce, and re-configure service interactions in an SOA

Why is this so important ?

- Key success criteria for SOA implementations

- Provides a measurable feedback loop and validates IT goals and strategies

- Makes the reuse promises of SOA "realistic"

- Accounts for performance characteristics earlier in the architecture and design

Benefits of SOA Governance

Business Benefits

Introduces flexibility for new business models (Chargeback or Gold and Platinum Membership Services)

Enforces consistent SLAs or operational expectations as the organization scales

Manage security and compliance

Reduce cost of operations

Added IT Benefits

Control service and policy proliferation within the enterprise

Manage service lifecycle, dependencies, and interdependencies

Facilitate incorporation of evolving standards

Simplify infrastructure

Promote interoperability

Features of a SOA Governance Solution

Design Time

Artefact

- Lookup and discovery
- Meta-data management
  - Taxonomies
  - Classification
- Versioning
- Lifecycle management SOA system of record change notifications

Run Time

Governance contract

- Interjection
- Enforcement
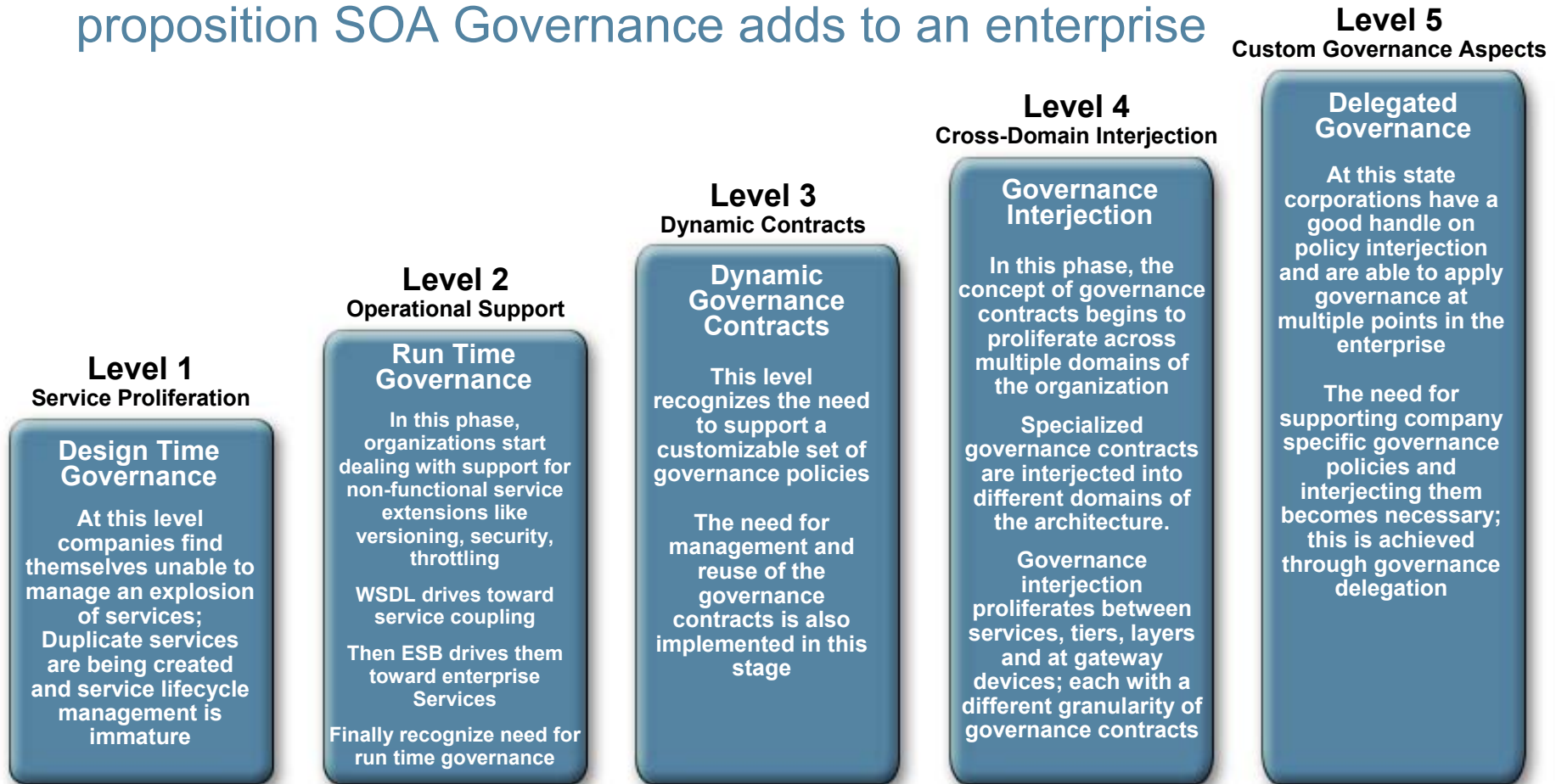- Validation
- Fault compensation
- Dynamic routing

Operations

- Monitoring and measurement
- System management and administration
- BAM and reporting
- Dynamic linking and control

Author and Manage

Execute and Enforce

Policies and Contracts

Monitor and Reconfigure
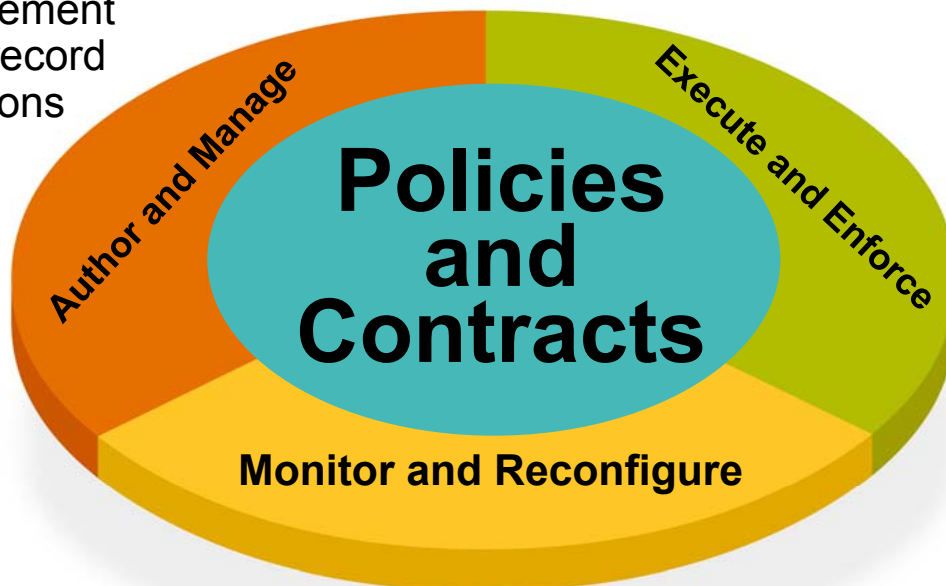
Maturity Milestones for SOA Governance

The maturity milestones describe the evolving value proposition SOA Governance adds to an enterprise

Level 1
Service Proliferation

Design Time Governance

At this level companies find themselves unable to manage an explosion of services; Duplicate services are being created and service lifecycle management is immature

Level 2
Operational Support

Run Time Governance

In this phase, organizations start dealing with support for non-functional service extensions like versioning, security, throttling

WSDL drives toward service coupling

Then ESB drives them toward enterprise Services

Finally recognize need for run time governance

Level 3
Dynamic Contracts

Dynamic Governance Contracts

This level recognizes the need to support a customizable set of governance policies

The need for management and reuse of the governance contracts is also implemented in this stage

Level 4
Cross-Domain Interjection

Governance Interjection

In this phase, the concept of governance contracts begins to proliferate across multiple domains of the organization

Specialized governance contracts are interjected into different domains of the architecture.

Governance interjection proliferates between services, tiers, layers and at gateway devices; each with a different granularity of governance contracts

Level 5
Custom Governance Aspects

Delegated Governance

At this state corporations have a good handle on policy interjection and are able to apply governance at multiple points in the enterprise

The need for supporting company specific governance policies and interjecting them becomes necessary; this is achieved through governance delegation

Agenda

Importance of SOA Governance

Architecting a service governance solution

Transition strategies for Java EE platform

SGF Governance solution

Pattern Considerations

- Consider an intermediary approach

- Use Inversion of Control to assemble pluggable aspects

- Maximize the use of facades and delegation when creating the governance contracts and the services abstractions

- Design solution with plans to decorate any service with our governance contracts

- Leverage concepts of Aspect-oriented programming

java.sun.com/javaone

Design for Separation of Roles

SOA Developer

- Authentication
- Authorization
- Throttling
- Lease
- Chargeback
- Auditing
- Throughput
- Etc.

Develops service implementations and Aspect enforcement methods

Operations Analyst

- Authentication
- Authorization
- Throttling
- Lease
- Chargeback
- Auditing
- Throughput
- Etc.

Platinum   Gold

Create SLA-based governance contracts

Governance Officer

Platinum   Gold

GetQuoteSymbol

Apply business policies to services for consumption

Extend the Definition of a Service

Governed "Service"

Service Consumer → Non-Functional Qualities → Service Interface → Functional Business Logic
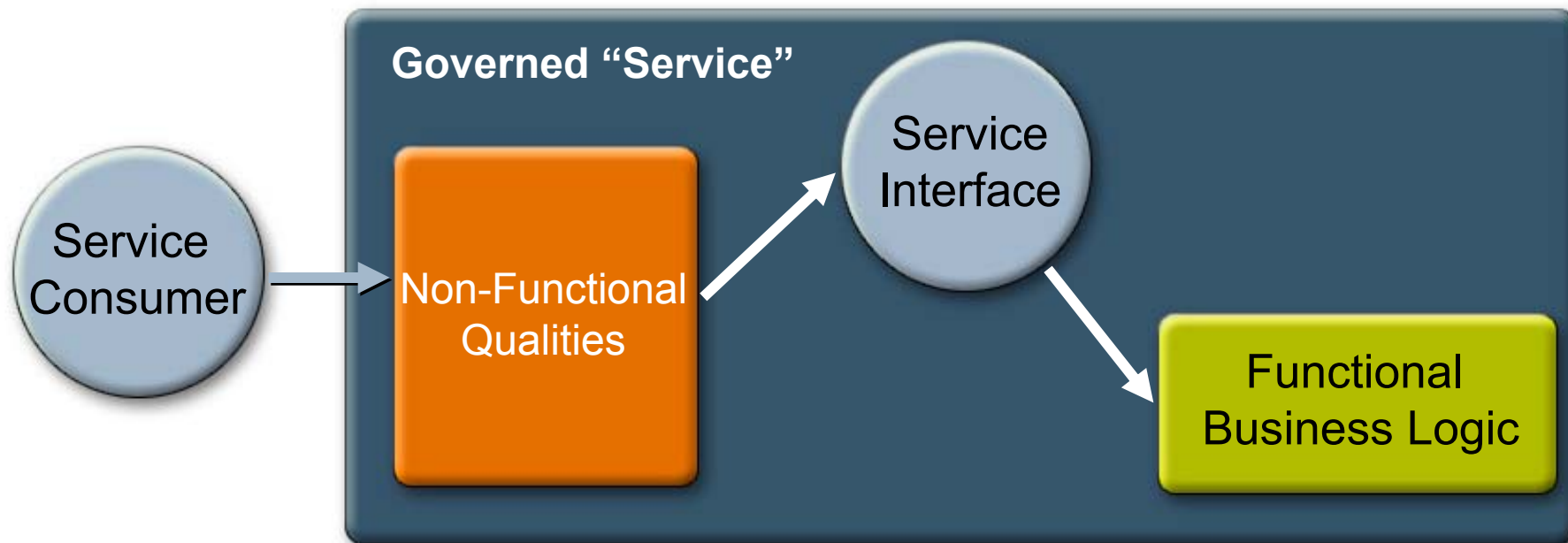
A **service** is a self-contained component **delivering business functionality** combined with an **extendible set of non-functional, policy-driven qualities** (such as security, industry/customer defined service policies, management, monitoring, and lifecycle management) which **responds to requests** through a **well-defined, standard, published interface**

java.sun.com/javaone

Building the Governance Contract

- A Governance contract specifies the **service agreement** between consumers and producers

- It is composed of any number of non-functional **aspects**

- Aspects are enforced by **clauses** that associate enforcement values

- Ex. Throughput must be >= 10 transactions/sec

Governance Contract

Clause (Valid till 1/1/08)

Aspect (Lease)

Aspect (Monitoring)

Aspect (Chargeback)

Aspects Building Blocks

- Each Aspect represents an implementation strategy for a systemic quality

- Aspects are reusable and re-combinable

- Aspects may associate service offerings with specific types of policies—such as security

- Aspects can be **universal** (security), **industry specific** (HIPAA compliance), or **unique for an enterprise**

- Aspect groups provide a design type classification system for organizing Aspects and Aspect enforcement methods
  - Security, availability
  - Accountability, management
  - Performance

Common Aspects

- Throttling
  - Impose throughput limits
- Lease enforcement
  - Enforce service expiration dates
- Chargeback
  - Chargeback for service usage
  - Bill the service consumer for throughput

- Monitoring
  - Simple logging of service metrics or complex monitoring model
  - Link to reporting and alerting tools
  - Incorporate third-party enterprise tools
- Version enforcement
  - Advanced routing
  - Translate message formats

java.sun.com/javaone

Aspect Enforcement Methods

- Executable code that enforces the non-functional characteristics
  - Enforce policy
  - Accept/reject messages based on dynamic criteria
  - Record metrics

- Transform the overall governance contract context
  - Enrich message content
  - Message translation/transformation
  - Chained together to leverage previous Aspects (if authorized)

- Support delegated governance
  - Delegate enforcement to existing implementations
  - Non-functional abstraction that can evolve over time
    - Standards implementation—authentication (evolving implementations)
    - Monitoring routines—extended for SOX or new compliance issue
  - Support extensibility with existing or future Aspects definitions

Evaluate Contract Standards

- Consider the set of WS-policy specs as the governance contract interface standard

- Policy snippets easily attach to our services at:
  - Ports, operations, messages
  - Endpoints
  - Protocols

```
<wsp:Policy>
  <wsp:ExactlyOne>
    <wsp:All>
      <Assertion> ... </Assertion>
      ...
      <Assertion> ... </Assertion>
    </wsp:All>
    ...
    <wsp:All>
      <Assertion> ... </Assertion>
      ...
      <Assertion> ... </Assertion>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```
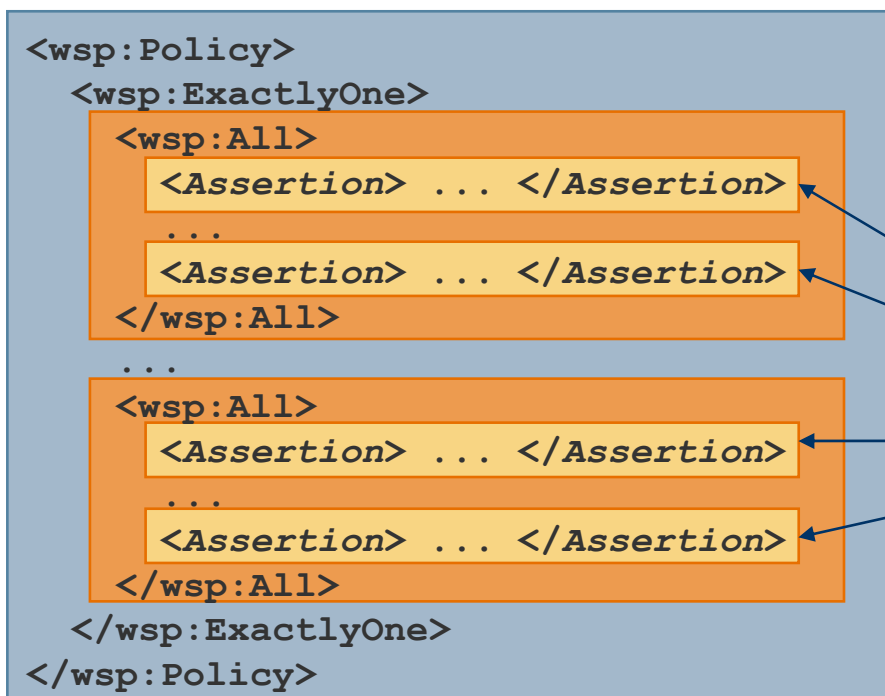
Flexible policy expressions collection of alternatives (**clauses**)

Policy assertion domain-specific behavior **(represents an aggregation of Governance statements)**

Manage and Apply Contracts

- Manage contracts and classify policy via UDDI
    - Reusable policy expressions can be registered as distinct models

- Attaching contracts to services—WSDL
    - Policy references can be made
        - Via global attributes
        - Or to policy expressions defined within the WSDL document
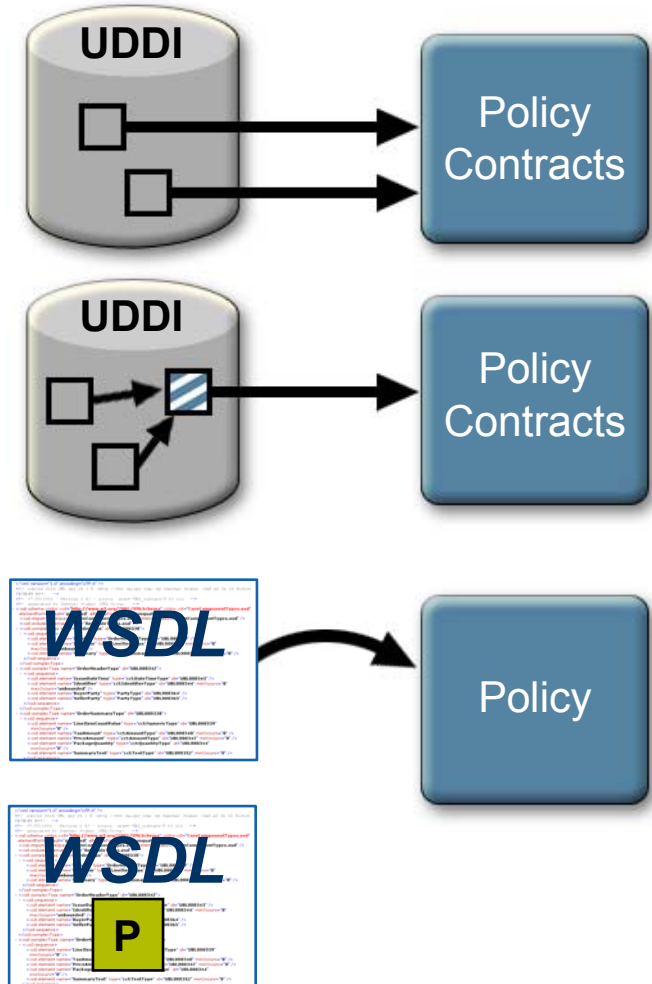
Governance Contract Overview

Standard Interface/WS-Policy Framework

Governance Contract

Clause (Valid till 1/1/08)

Aspect (Lease)
Pre-Built Implementation

Aspect (Monitoring)
Pre-Built and Customized

Aspect (Chargeback)
Delegated (Custom Aspect)

- Aspect can be enforced before, after, or both
- Can combine and reuse any combination of Aspects
- Can be used in combination with any service

Delegate Governance

Aspect Implementation Method
Invoke
(Customized Implementation)

Aspect Implementation Method
Invoke

Chargeback Implementation

java.sun.com/javaone

Governed Services

Governed Service Offering

Functional Business Logic

Service Facade

Non-Functional Attributes

Governance Contract

Service façade abstracts
- Service implementation
- Includes one or more versions
- Request routing
- Binding/connectivity

Governance contract
- Aspects
- Aspect enforcement methods
- Aspect groups
- Clauses: Values and policies

Service Governance Lifecycles
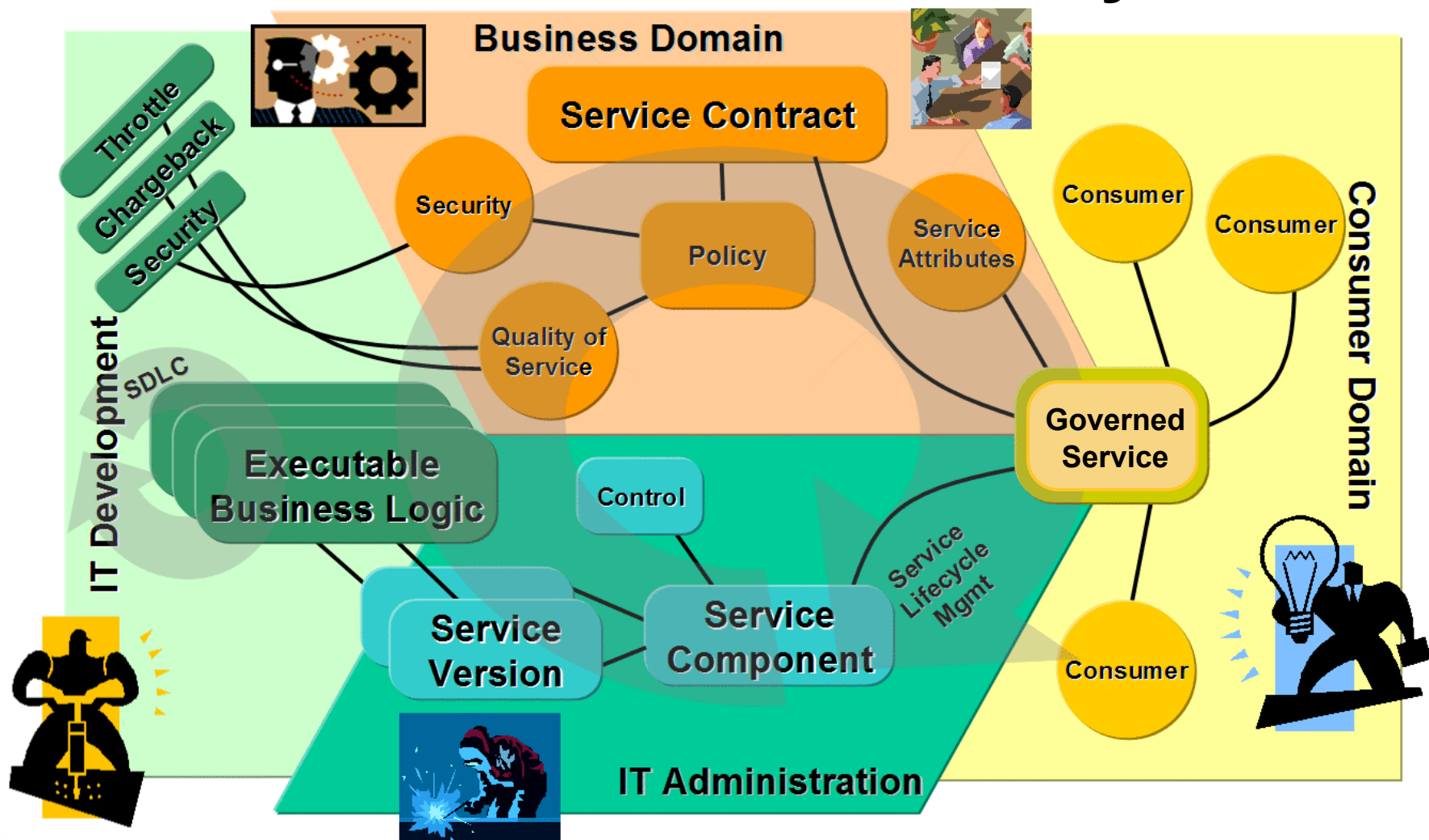
SOA Governance Framework

- Every service request passes through a series of Aspect enforcement methods as defined in the contracts

- Each method enforces a single clause

- Enforcement methods can be added without code change

- Use the same bindings as service implementations

Service Governance Framework

Service Consumers

Service Bindings

Governance Aspects

Security | Lease | Versioning | Throttling | Monitoring | Chargeback | Client-specific | Industry-specific | Compliance

Service Consumer → HTTP
Service Consumer → JMS, SMTP
Service Consumer → RMI, etc.

Implementation Bindings

Service Implementations

HTTP ↔ Service A (HTTP)
JMS ↔ Service B (JMS)
EJB ↔ Service C (EJB)
POJO ↔ Service D (POJO)
etc. ↔ Service X (etc.)

Contract A:
Authentication: required
Lease: enforced
Versioning: validate only

Contract B:
Throttling: service (20 TPS)
Chargeback: per call
SOX compliance: …

JMS = Java Message Service (JMS)
EJB = Enterprise JavaBeans™ (EJB™)

java.sun.com/javaone

Decorate Governance—SOA Layers
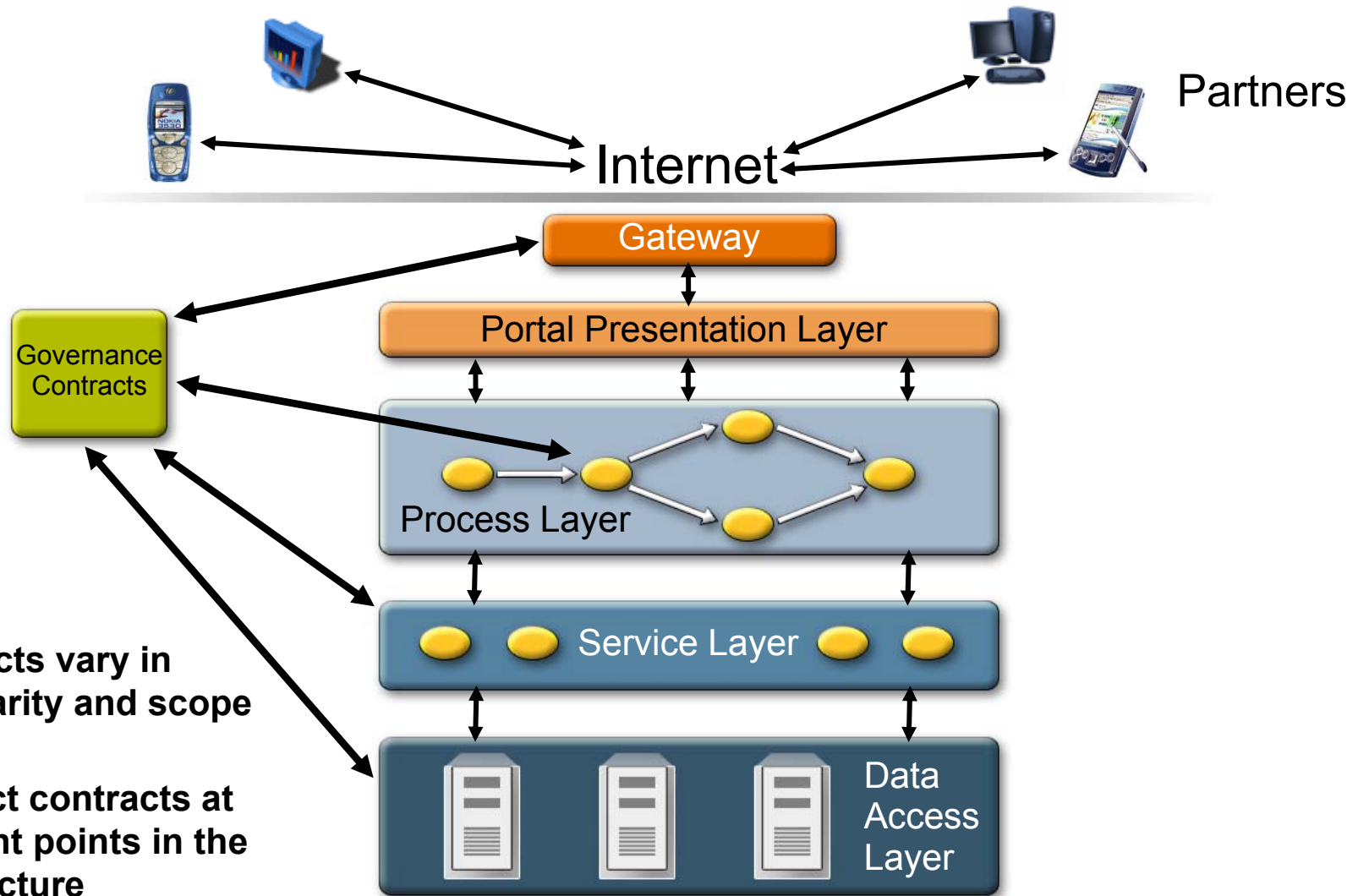
Partners

Internet

Gateway

Governance Contracts

Portal Presentation Layer

Process Layer

Service Layer

Data Access Layer

Contracts vary in granularity and scope

Interject contracts at different points in the architecture

Agenda

Importance of SOA Governance

Architecting a service governance solution

Transition strategies for Java EE platform

SGF Governance solution

Transition Strategies for SOA Governance

- ## Service gateway model
  - ### Relegate governance to edge devices

- ## Service governance intermediaries
  - ### Vary the granularity of the governance contract and apply governance at multiple enforcements point in the architecture

- ## Standards-based Java Business Integration (JBI) solution
  - ### Position your enterprise for a pluggable approach to governance; where governance contracts aggregate the necessary pluggable binding engines

Service Gateway Model

- Overview
  - The gateway is a good starting place to manage Governance issues like security and identity

- Pros
  - Isolates governance issues for better manageability
  - Addresses generic high-impact governance issues
  - Can be applied to domains, tiers, or layers of SOA

- Cons
  - Governance contracts quickly get complex as the need to support specific or specialize governance capabilities is needed
  - Hard to extend the specialized aspects

Service Intermediary Model

- Overview
  - Rather then applying governance policies generically to a domain, this model allows for tailored governance of specific services

- Pros
  - Provide specific governance contracts for individual services, operations, or endpoints
  - Governance contract is a small fragment of a policy necessary at an edge device
  - Very extendable with support for delegated implementations of governance aspects

- Cons
  - More complex to implement governance at a services level
  - Should not be used as a general purpose governance solution (i.e., logging every client IP access)

Standards-Oriented Model

- ## Overview
  - This is a visionary approach that leverages standards like SCA/JBI/WCF and policy to build pluggable governance infrastructure

- ## Pros
  - This is the most extendable approach; for instance, new Aspect engines can be plugged into a JBI bus
  - Easy to build service contracts through groupings of policy enforcement
  - Cross-vendor solution of the bus and the policy engines

- ## Cons
  - Requires the maturity and standards support for frameworks like JBI, or WCF

java.sun.com/javaone

Dynamic Policy Model
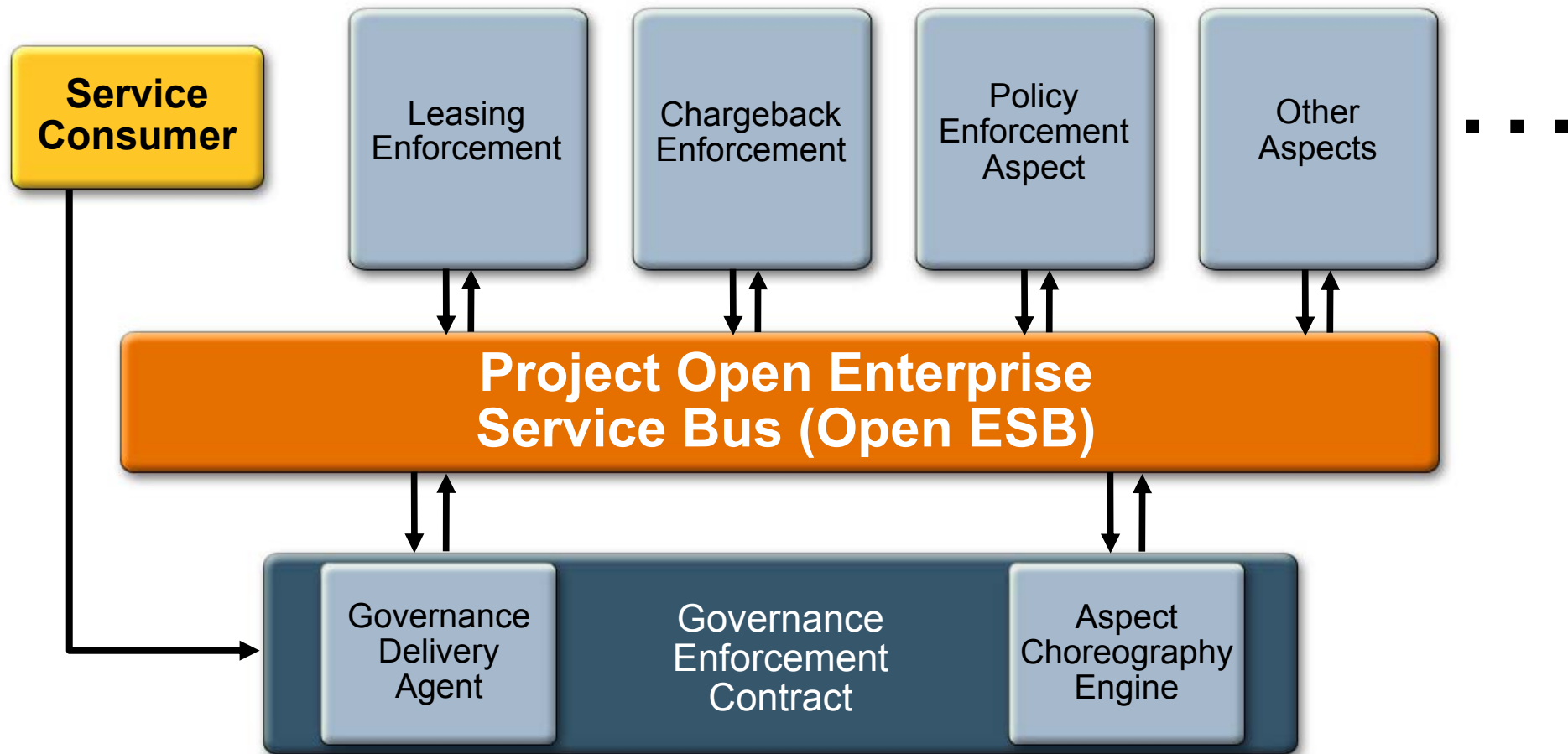
Open ESB

A True Open SOA Community

Open ESB 2.0 Beta 2—Available Now!

JBI-based SOA integration platform

- Open standard, open source, interoperable

- Build composite applications leveraging existing applications and web services

- An extensible platform with pluggable architecture

- Integrated runtime with GlassFish V2

- Integrated tooling through NetBeans™ software 6.0

- Available in Java EE platform SDK

- Community-based JBI component development

- **JBI**—An open standard for SOA based integration platform

- Rich set of Service Engines including BPEL, IEP, XSLT, Java EE platform, Aspects, WLM, Data Mashups, Encoder

- Exhaustive list of Binding components including, Http, Java DataBase Connectivity (JDBC™), JMS, MQ, SAP, Email, CICS, IMS, SAP and many more

- **Free** to download, and deploy

http://open-esb.org

Agenda

Importance of SOA Governance

Architecting a service governance solution

Transition strategies for Java EE platform

SGF Governance solution

java.sun.com/javaone

Sun SOA Governance Solution

Installation package—Software components

- Service Governance Framework (SGF)
- Governance modeller and management console
- Joint effort between Sun and Accenture

Documentation

- Installation guide
- User guide
- API reference, source code

Professional Services

- Training
- Architecture and design in a service governance framework implementation

java.sun.com/javaone

SGF Architecture
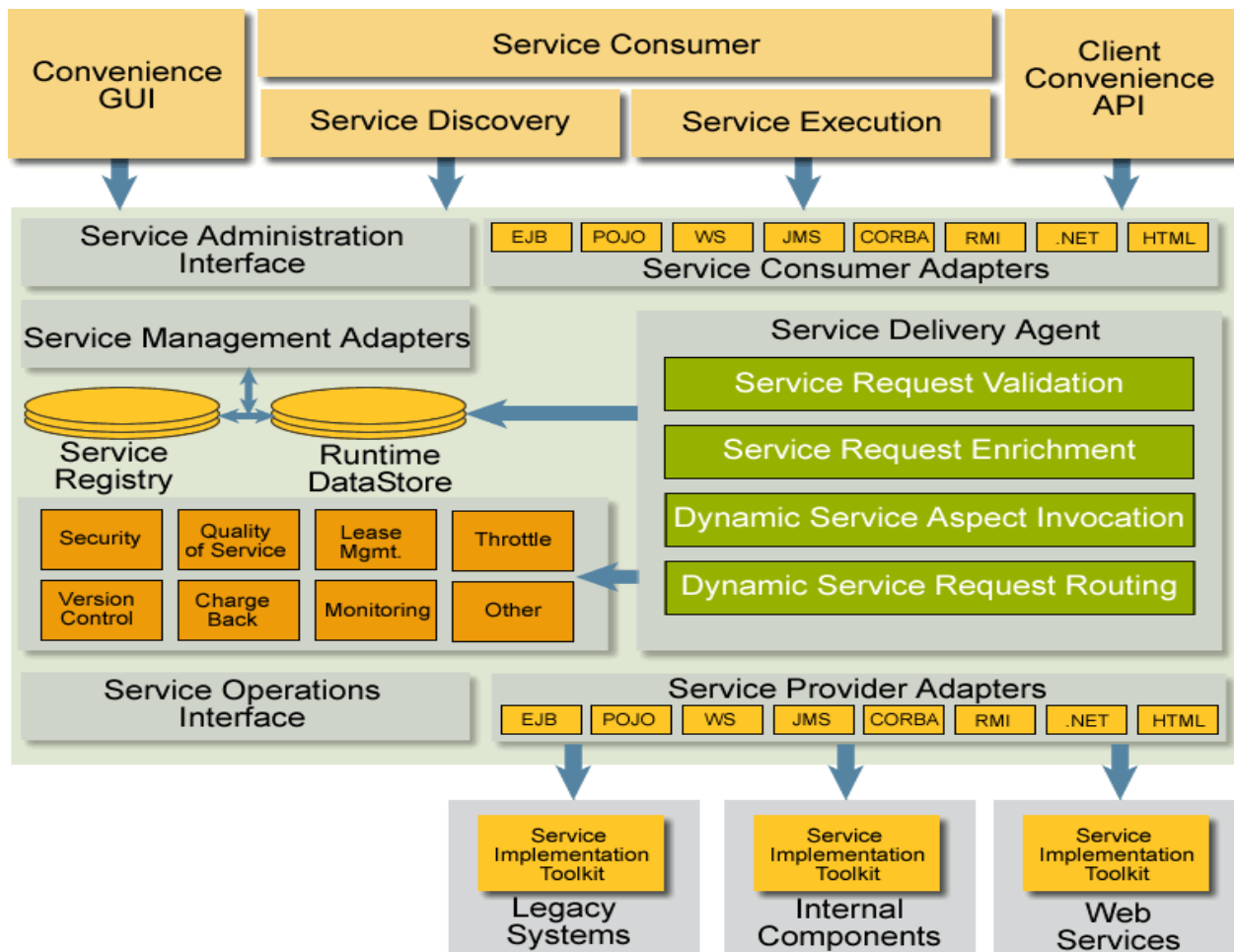
Supports Separation of Roles

Import Service to Be Governed

Define Governance Aspects/Contracts

Support for Custom Aspect Development

- SGF supports delegated Governance through Aspect implementation methods

- Each Aspect in SGF has an associated **Aspect Enforcement Method**

- We can used pre-built Aspects or create our own

- To create a new Aspect we will need to:
  - Design the Aspect functionality
  - Implement the Aspect Invoke Method
  - Build and deploy the new Aspect
  - Configure the Aspect into a governance contract
  - Apply the contract to a service offering and test

Custom Aspect Implementations

- Here is the Aspect invoke method for the lease Aspect:
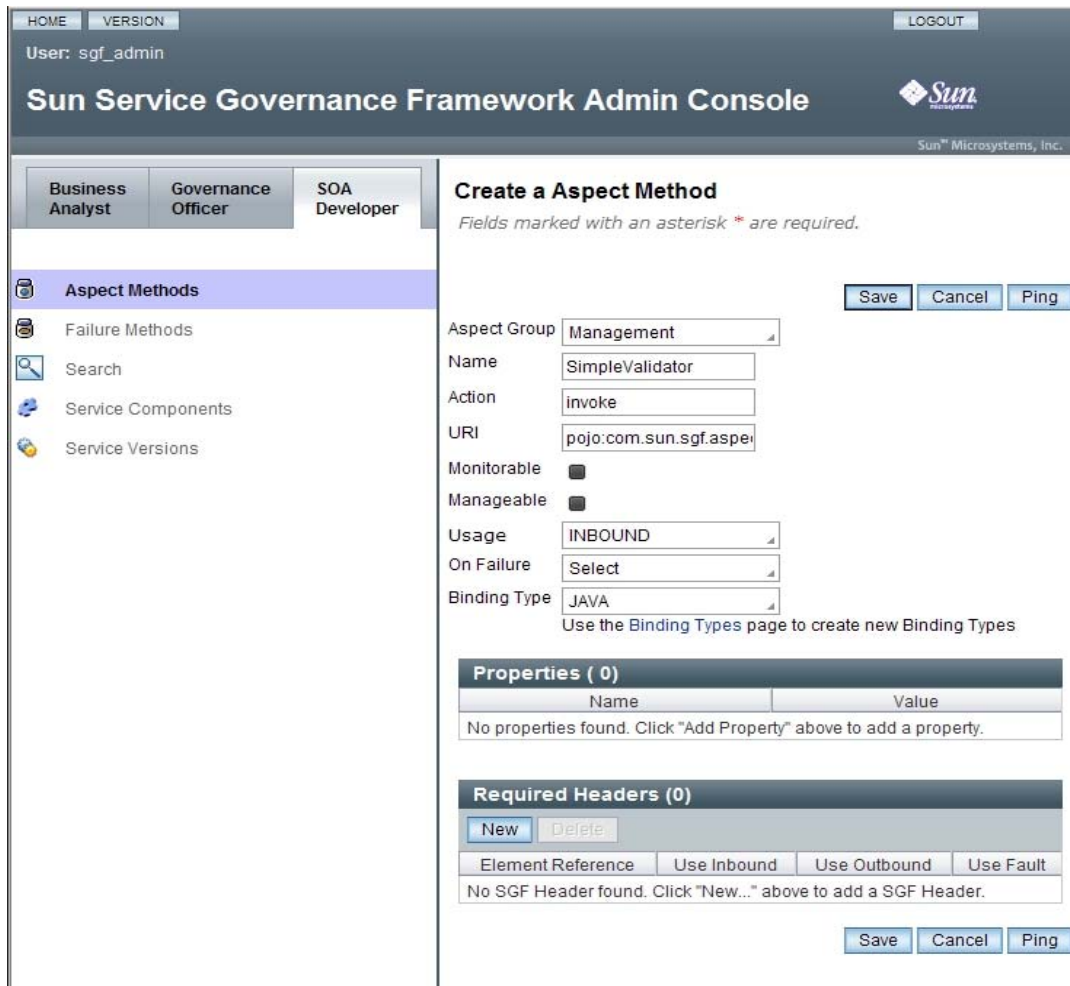
```
252   * (NonrecoverableSGFException with type CONFIGURATION) will be
253   * thrown if the ServiceOffering does not have a lease defined
254   * and associated with the Service Offering.
255   */
256   public MessageContext invoke(MessageContext context)
257     throws SGFException {
258       ServiceOffering serviceOffering = context.getServiceOffering();
259       SOAPMessage soapMessage = context.getSoapMessage();
260       AspectSOAPUtil soapUtil = new AspectSOAPUtil();
261       SortedSet<GovernanceClause> inboundClauseList = null;
262       XPath xpath=null;
```

- From the **MessageContext** object, we can get the **ServiceOffering**, which contains key information about the SGF Governance contract

Integrate Aspect Implementation in SGF

- Bind the custom implementation with the SGF framework

- Configure the Aspect as a pojo with a Java technology binding

- Add the Aspect to a Governance contract

- Then you can deploy the Governance contract with any service

java.sun.com/javaone

Generates Governed Services

The SGF Community

The Java Technology Developer Community's Watercooler

Over **200,000*** members (2x growth in last year)

Over **2,500** Projects, 110 JUGs, 22 Communities

Home of JDK™ Project Glassfish, Project Looking Glass, Project Peabody

---

Tools, platform, technology, education, games, ...

Community-building infrastructure

Key Java™ leaders participate

*Source: Sun 2/06—See website for latest stats

Summary

- SOA Governance can be realized today and your enterprise to enhance the benefits of SOA

- A SOA Governance solution should account for all six levels of the governance maturity model

- Governance contracts can be constructed to support governance at multiple layers of the architecture

- Sun has a service governance solution and is working to make it an open source project on Java.net

java.sun.com/javaone

Q&A

Michael.Wheaton@sun.com

Decorating Your SOA Services With Governance Enforcement Contracts

Michael Wheaton

Principal Engineer
SOA Center of Excellence
Sun Microsystems, Inc.

TS-8440