



Java is a trademark of Sun Microsystems, Inc.

JavaOneSM

SOA deployment challenges in the real world

Sastry Malladi

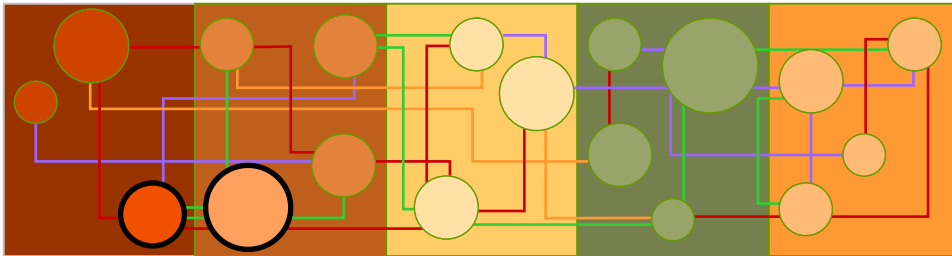
Distinguished Architect.
eBay, Inc.

Agenda

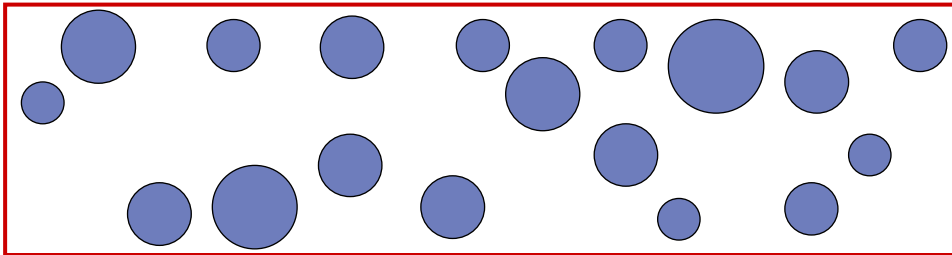
- > Brief recap of SOA concepts and benefits
- > Challenges encountered in large scale SOA deployments
- > How is eBay addressing these challenges
- > The role of SOA governance
- > Summary

What is SOA?

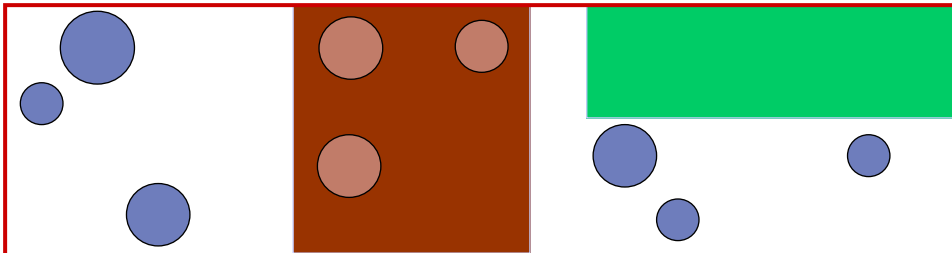
SOA is an *Application Architecture* evolution, not a *Technology* revolution



SOA is an architecture to move from brittle, hardwired, application silos that inhibit change...



... to shared, reusable, business and application services...



... which eliminates application redundancy and complexity, and enable Business Agility, Innovation and Operational Excellence.

SOA - Not just technology

Technology

- SOA Run-time stack
- Integrated tooling
- Infrastructure services
- Registry/Repository
- Routing

SOA

Process

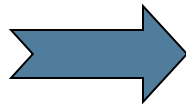
- Life-cycle and governance
Processes for Service Interface Approval, Security, change mgmt, etc.
- PDLC changes

People

- Think SOA
- Promote Reuse
- Biz and IT alignment
- Training & Docs

Some common Misconceptions dispelled

SOA is a new technology



It's a new architecture paradigm applied to existing or new technologies

SOA implies Web Services and SOAP



WS/SOAP is not an exclusive protocol. REST style with JSON, NV is equally popular, if not more.

SOA is an end in itself



SOA is a means to enable business agility

People are going to develop services from ground up



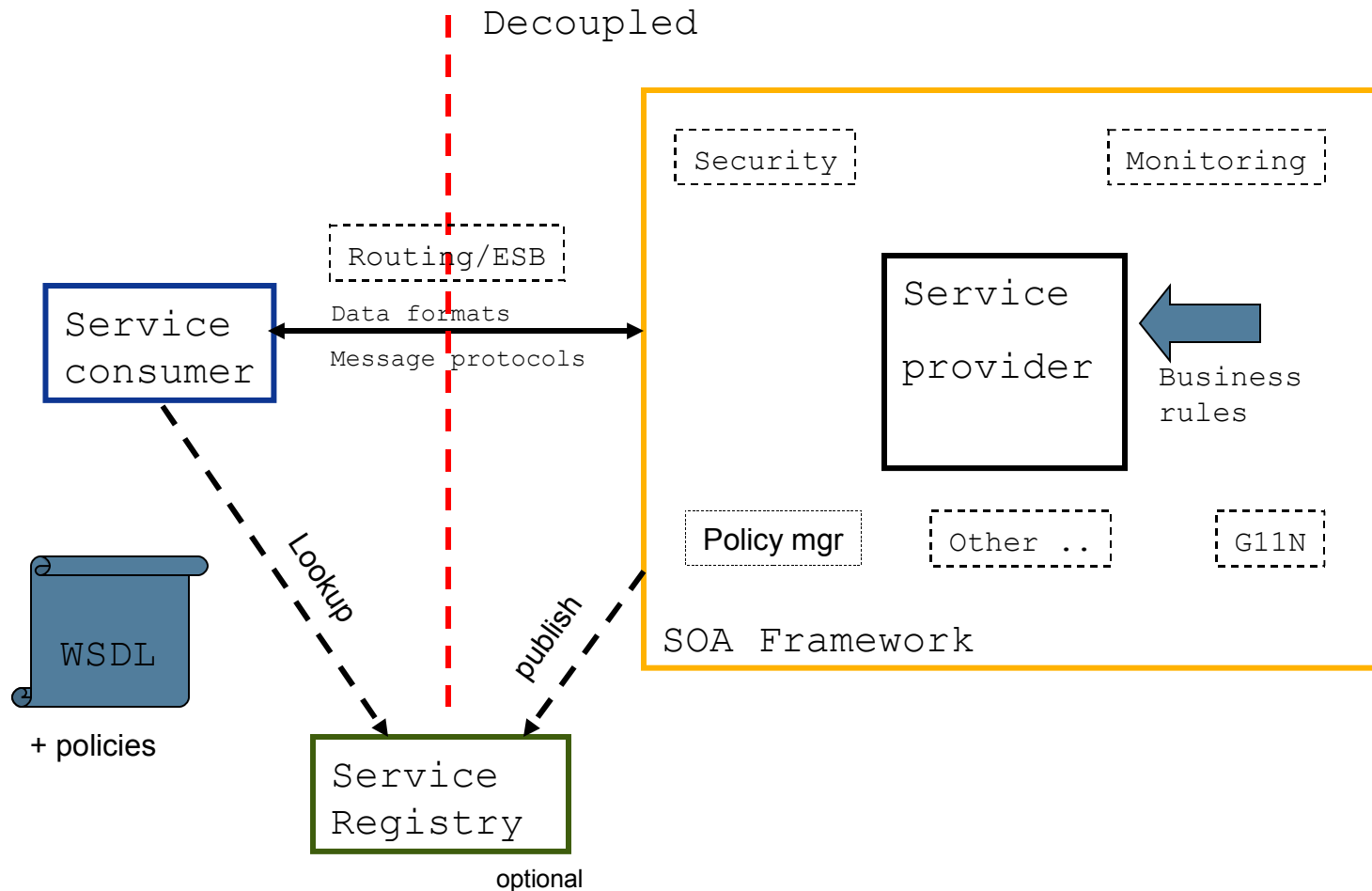
Business always leverage existing functionality, but morph them into services and develop some new services on the way

At service development time, its consumers/usecases are known



Service development may start with some usecases in mind, but they will evolve over a period of time

SOA model - From 30,000 feet – You already know !



SOA Benefits

> **Loosely coupled**

- Easily assemble and modify business applications
- Increases reusability and sharing; lowers implementation costs
- Increases adaptability; changes — resulting from M&A, 3rd party applications, etc. — are easily integrated

> **Modular**

- Avoids costly “big bang” implementations - Enables incremental development, deployment, and maintenance
- Over time, accelerates deployment of new application functionality; process becomes mostly assembly of existing services versus new development

> **Non-intrusive**

- Allows existing investment in assets to be leveraged

> **Standards-based**

- Interoperability with partner applications
- Platform independence enables the use of software and hardware of our choice
- Reduced threat of vendor lock-ins

SOA Benefits (contd.)

> **Business**

- Business agility
- Reduced IT costs

> **Scalability**

- Effective utilization of resources because of componentization
- Increased scalability, with appropriate deployment topologies

> **Manageability**

- Better manageability of code and functionality
- Easier code rollouts

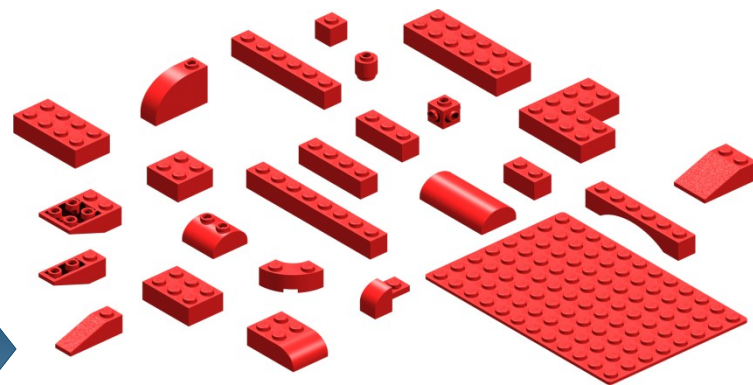
> **Opening up possibilities ...**

- Enabling building creative and interesting applications based on re-usable and encapsulated services
- Developer productivity and Better ownership

SOA Benefits – Enabling possibilities



TODAY



TOMMOROW



FUTURE

Terminology

> Service

- Well defined piece of functionality, with a defined interface, with fully encapsulated business logic and data, that is managed and exported consistently, and which is independent of any data formats and message protocols

> Data Binding/Format

- Also referred to as Wire format
- The format in which the data is represented on the wire (e.g XML, NV, JSON, BinaryXML, ..)

> Message Protocol

- The protocol followed by the client and service when exchanging messages (e.g, SOAP). There need not be any message protocol present (especially when communicating with internal services)

> Transport protocol

- Protocol used at the transport level (e.g HTTP, SMTP)

Agenda

- > Brief recap of SOA concepts and benefits
- > Challenges encountered in large scale SOA deployments
- > How is eBay addressing these challenges
- > SOA governance and its role
- > Summary

General SOA Challenges

> Technical

- Additional latencies due to multi-hop communications
- Debugging/Tracing is harder
- Need for efficient Request/Session level caching
- Increased Security and monitoring challenges
- Multiple, often competing standards

> Operational

- Developer adoption and learning curve
- Governance processes
- Migration of existing Apps
- Updates to existing tools and processes.
- Deployment and rollout

Further challenges in large scale deployments

> Technical

- Supporting internal and external clients that have different protocols/data binding needs, for the same service deployment
- QoS and SLA management
- Integration testing
- High availability and Scalability – high volume and low latency
- Decomposition of existing apps and migration of legacy services

> Operational

- Version and dependency management
- Impact to existing operational tools/environment
- Time to Market pressures
- Need for a strong yet flexible governance processes, especially with lots of services and the higher velocity of changes



Agenda

- > Brief recap of SOA concepts and benefits
- > Challenges encountered in large scale SOA deployments
- > How is eBay addressing these challenges
- > SOA governance and its role
- > Summary

Addressing the challenges

> Technical

- Light weight and high performant homegrown SOA platform
- Unified Testing framework and service virtualization
- Model-driven Service decomposition

> Operational

- Strong and yet flexible governance processes and life cycle management
 - Automated through a service registry/repository
- Incremental service deployment
- Strong operational management tools
- Developer training and incentives for being a good citizen

eBay SOA Platform Overview

> A comprehensive SOA Infrastructure platform

- **Framework** - Highly performant, extensible and light weight framework (overhead < 5 ms)
- **Monitoring** - A built in monitoring subsystem that is customizable
- **Security** – XACML based flexible and extensible authentication and authorization policies
- **Rate Limiting** - For enforcing capacity, budgeting and traffic control
- **Service Registry and Repository** – Governance, life cycle management

> Tooling

- Developer tools (eclipse plugins) for service/consumer development, types management and error management
- Operational tools (management, monitoring and alerting)

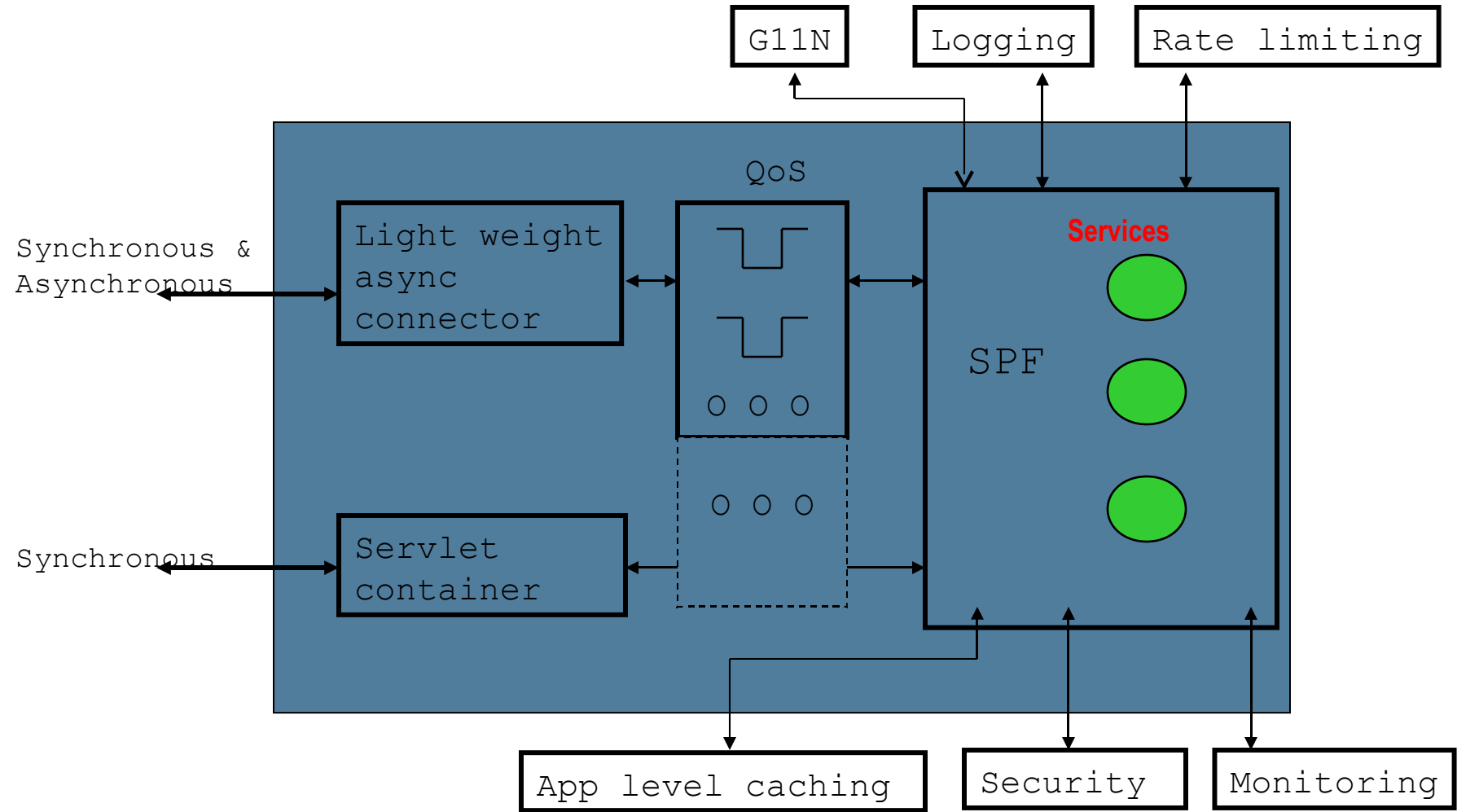
SOA Framework

- > A Pipeline based high performance architecture
- > Request and Response decoupling
- > Protocol and data binding agnostic service
 - The same service instance can be invoked using multiple protocols and data formats
 - No message normalizations or conversations
- > Pluggable data formats
 - Out of the box support for SOAP, REST, JSON, Binary XML
 - Streaming and attachment support
 - WSDL with SOAP and HTTP bindings
- > Pluggable transports, including local binding
- > Built in G11N support

SOA Framework (contd.)

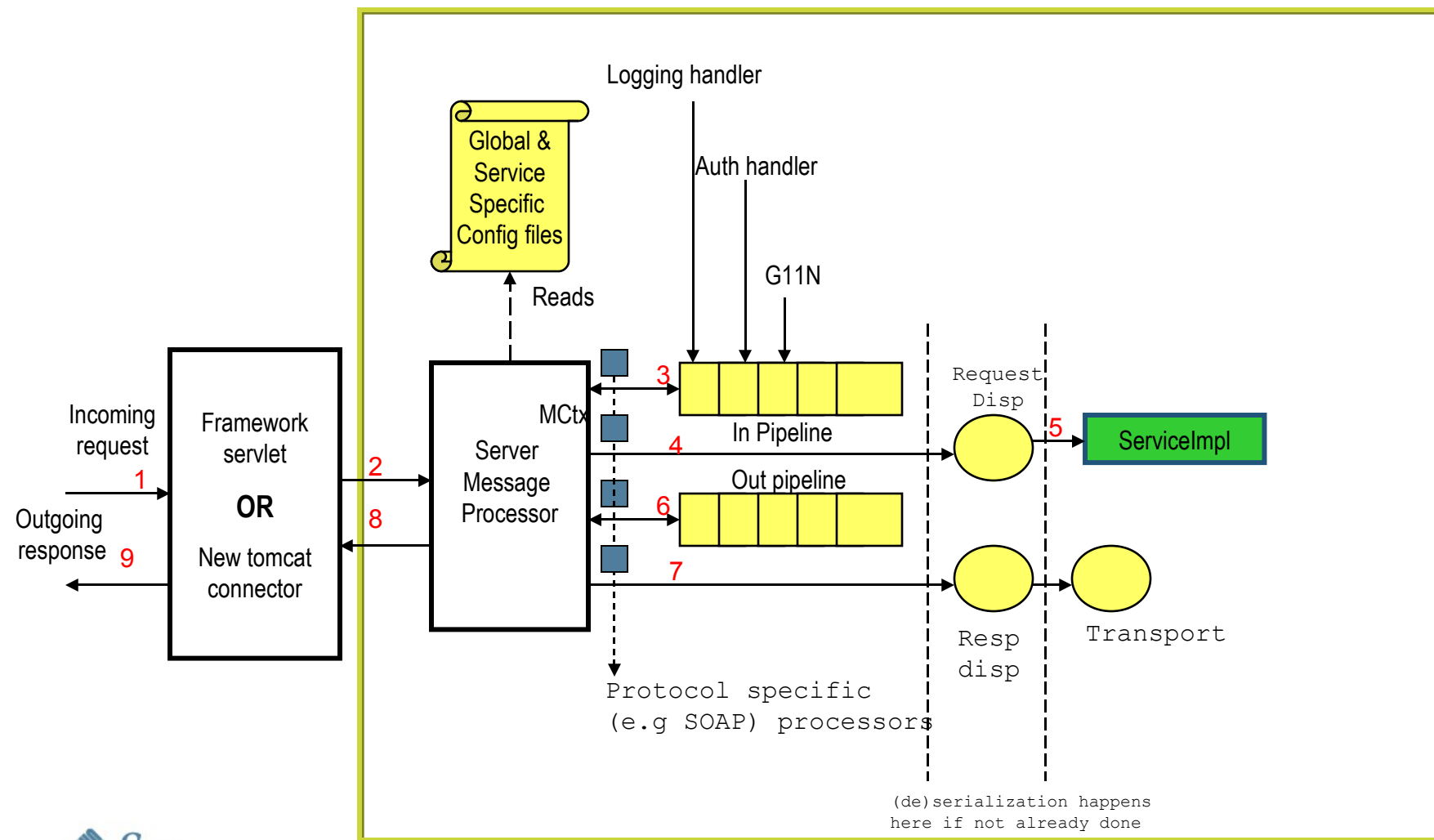
- > Version enforcement
- > Service chaining
- > Error management
- > Synchronous and Asynchronous invocation and execution
 - SEDA (Apache MINA) based tomcat connector
- > QoS
 - Priority and SLA based processing
- > Service Mark-up/Down facilities
- > Efficient Request/Session level caching

Model within the App Server

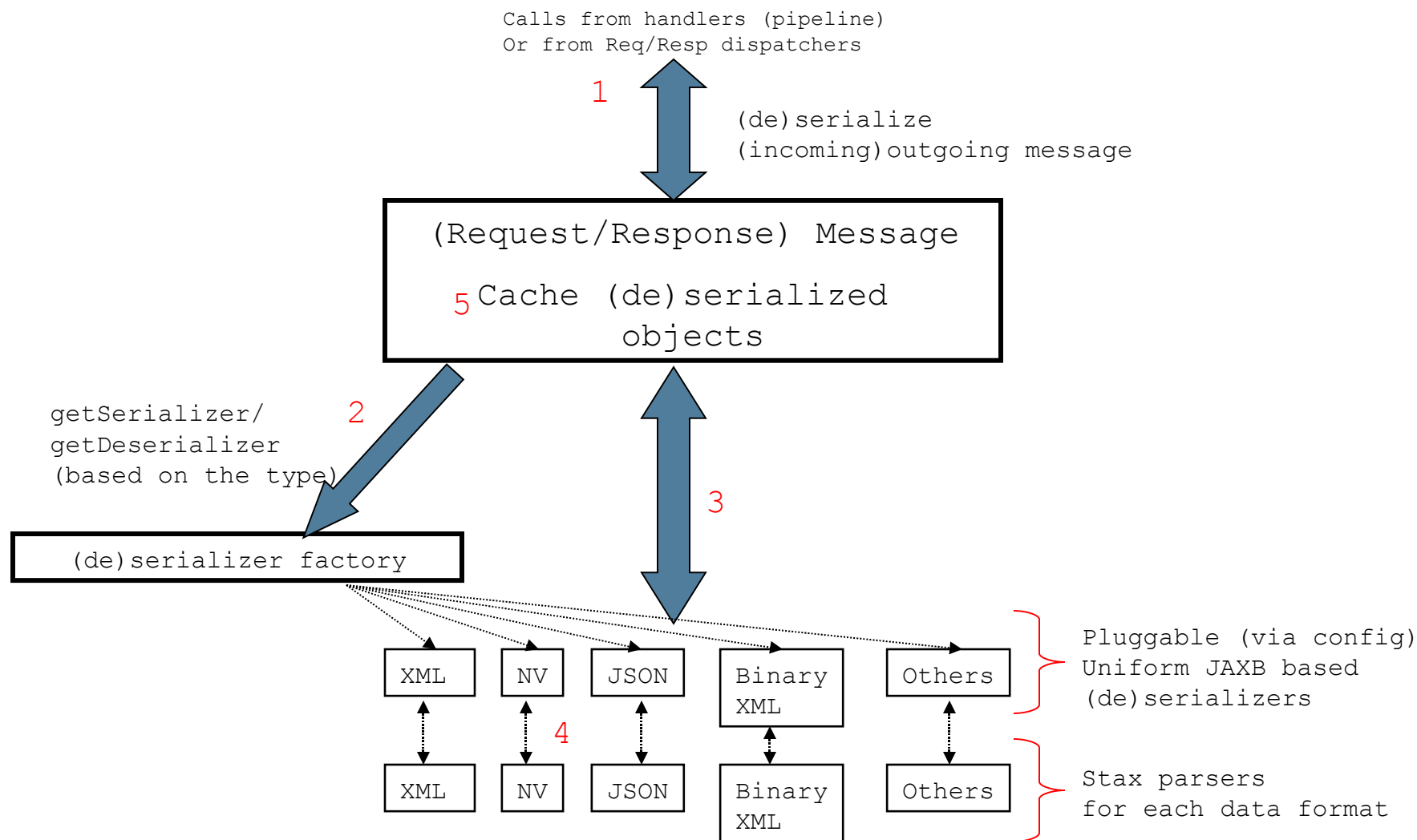


Pipeline architecture – provider framework

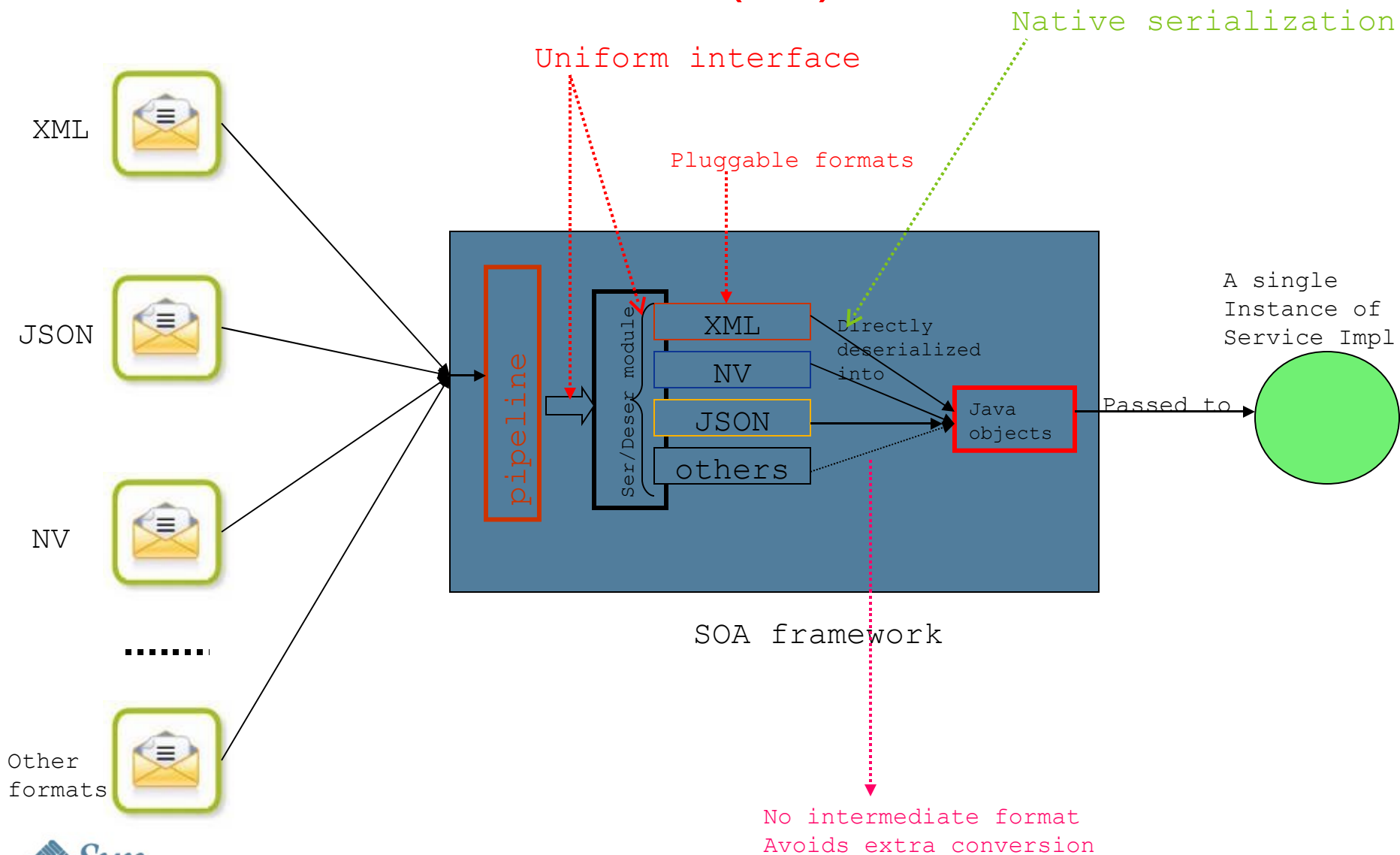
SPF



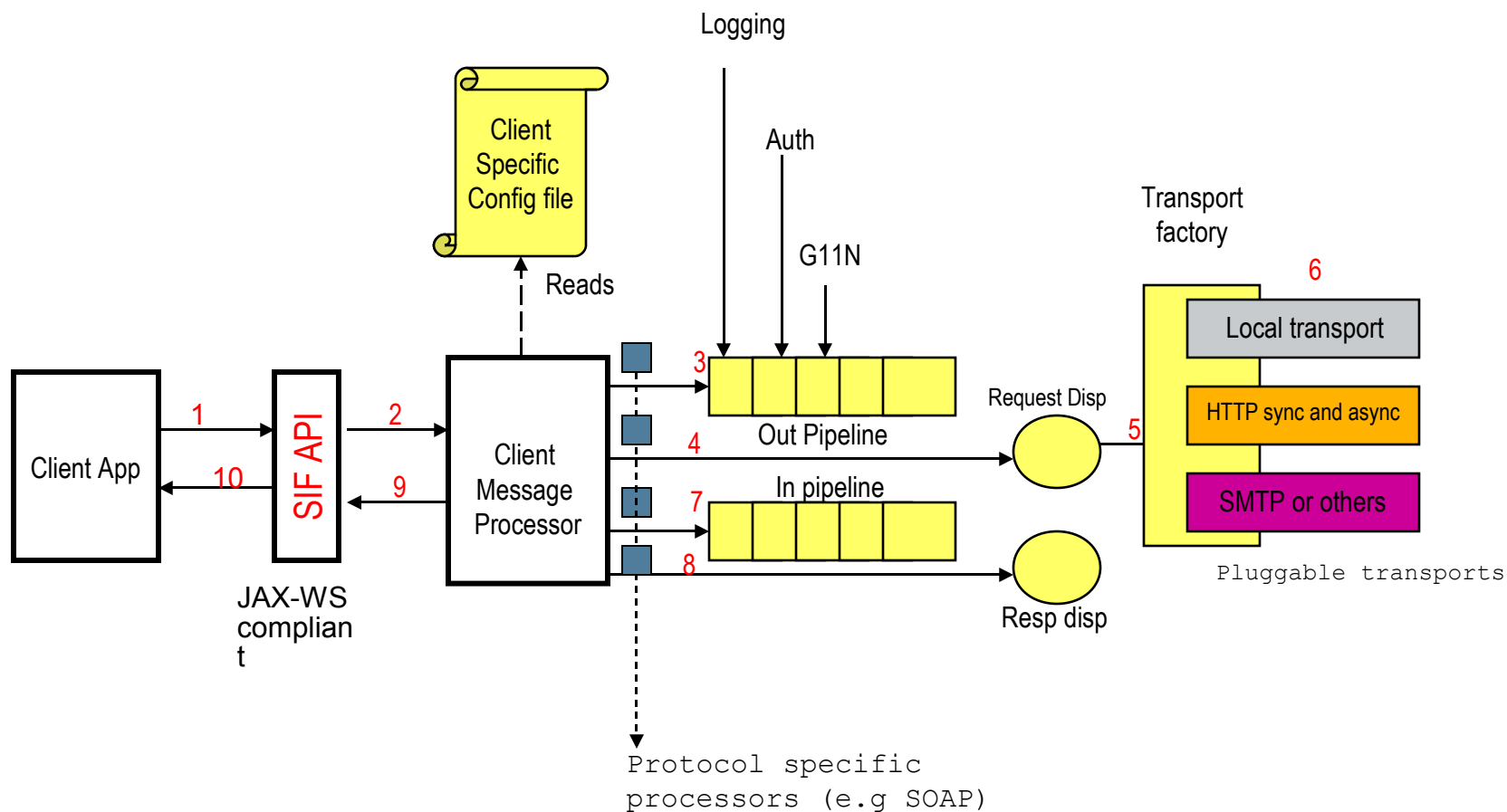
Pluggable data formats using JAXB



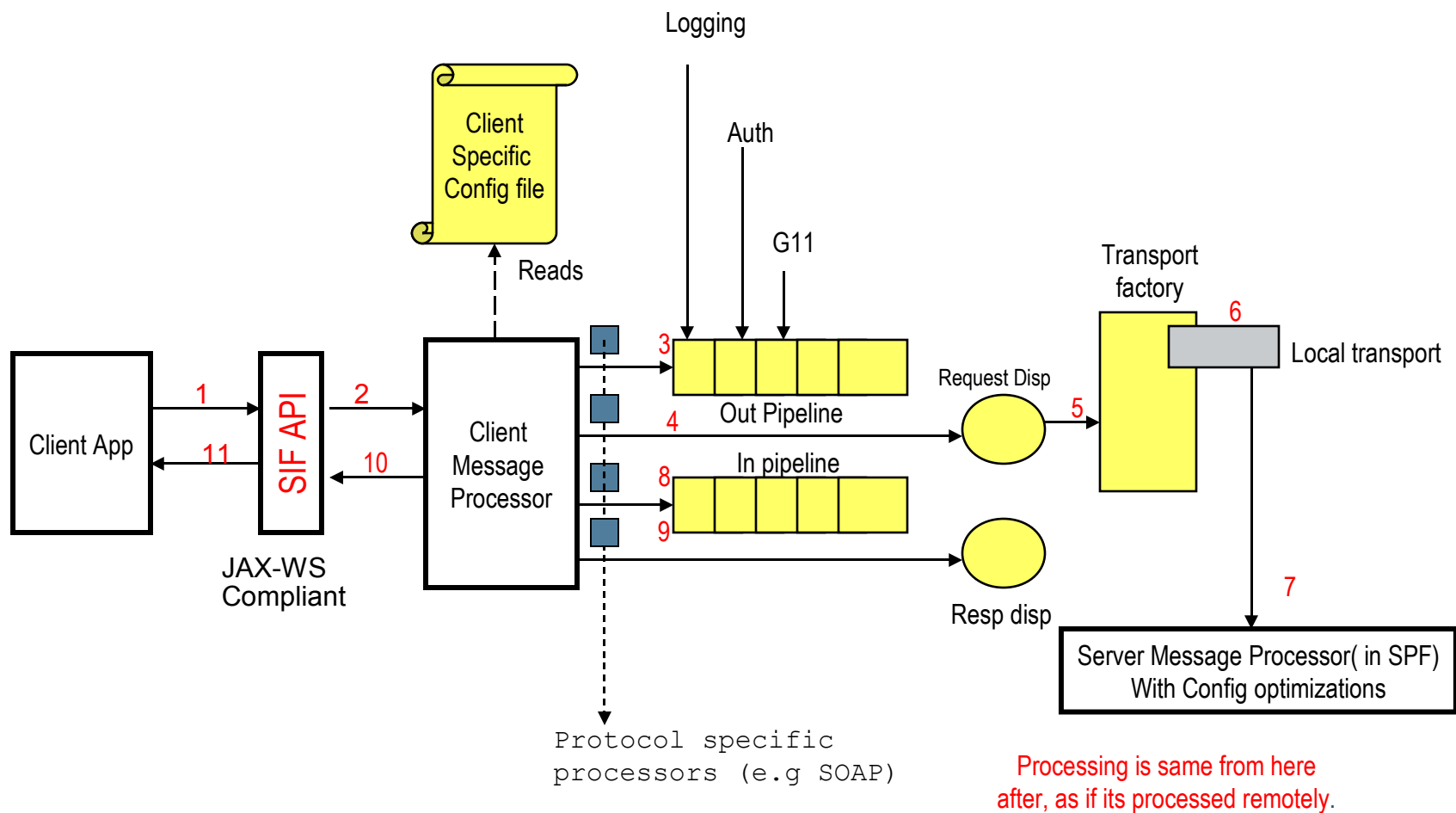
Native and uniform (de)serialization



Service Invocation Framework



Local Binding



Service Layers and Taxonomy

> Defining a finite set of service layers and their relationships

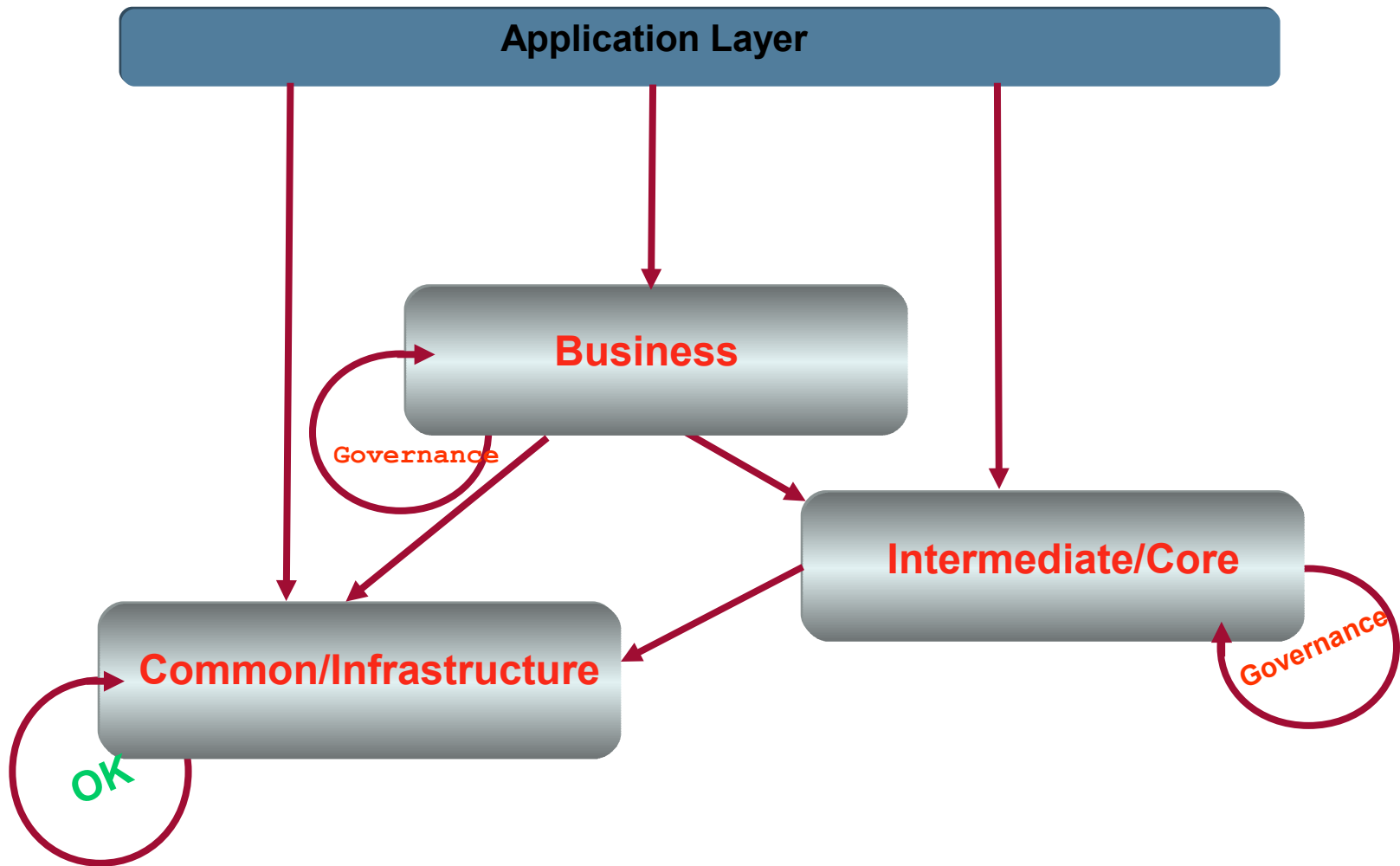
> Why

- One of the fundamental points about services is composability, meaning that services can be composed of other services. This translates to services calling other services creating different dependency relationships
- Without a proper layering and dependency patterns
 - The dependencies can become circular across layers
 - The management and rollout of those services becomes complex, if not impossible. This discourages “re-use” of existing services and encourages the good-old “copy-paste” again
 - Lack of visibility and clear ownership and guiding principles on what functionality can be implemented in any Service

> What

- Properly defining different layers of services
- Clearly establishing rules on what functionality can be implemented in which layers
- Defining allowed access pattern rules between layers

Our specific taxonomy (for example sake only)



Integrated Tooling

> Developer tooling well integrated with Eclipse IDE

- Integration with the Service registry and governance processes.
 - Dependencies are checked at design/development time itself
 - Versioning and backward compatibilities are enforced
- Testing : Generation of tests and a mechanism to enforce successful running of the same

> Deployment tools

- Dependency checking and rollout in that order
- Rollback support
- Local Vs Remote binding
- Life-cycle management and change notifications

> Operational tooling

- Management and monitoring dashboard
- Alerting and SLA management



Agenda

- > Brief recap of SOA concepts and benefits
- > Challenges encountered in large scale SOA deployments
- > How is eBay addressing these challenges
- > SOA governance and its role
- > Summary

What is SOA governance

- > A set of activities related to exercising control over design, development and life-cycle of Service providers and Service consumers, for consistency manageability and promoting re-use (subset of IT governance)
- > Design time
 - Interface and types design review and approval
 - Enforcement of consistency guidelines for contract (interface, policies and behavior)
 - Enforcement of dependency and layering guidelines
- > Runtime
 - Deployment policies
 - Security/caching/monitoring/availability policies
 - Reconciliation between what it is (runtime) and what it should be (design)
- > Change management
 - Dependency and impact analysis
 - Maintaining backwards compatibility for minor versions
 - Making sure existing clients don't break.
- > Many industry vendors provide this product
 - Not a very mature space however

SOA governance at eBay

> Service Repository

- Metadata management and life cycle management of services, data types and other assets
- Automated Governance processes
- Dependency and Impact analysis
- Closed loop governance

> Service Registry

- A subset of the repository (without the governance process), light weight access, supports UDDI access
- Used for routing purposes

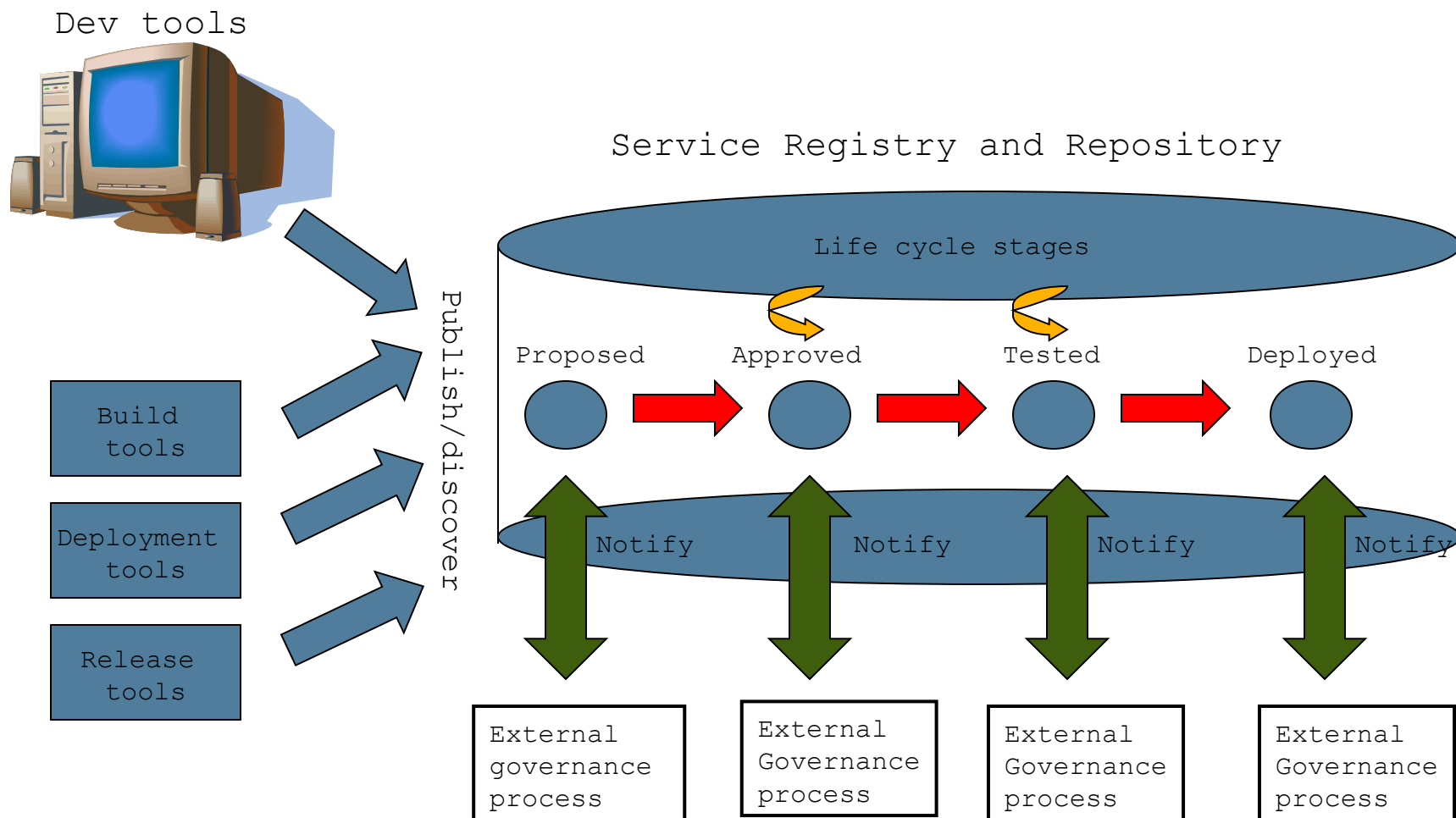
> Implementing SOA governance is crucial for success of our large scale deployments

Design time governance

> Defined a light weight, yet flexible process

- Both services and data types are managed and life cycled
- An automated workflow logic that transitions the process between these states.
- Process is slightly different for different types of services.
- Formal interface (WSDL) and policy (Security, Rate Limiting, Availability) approvals
- Automated XQuery based assertions to validate WSDL and policies
- Integrated with our SOA eclipse plugins for seamless governance process
- Integrated with all our existing processes (e.g, project management, Ops Capacity approvals, Security approvals), so there is a single starting point for the governance
- Consumer governance to track and approve service consumers

Governance process - Sample



Runtime governance

- > Monitor dependencies and policies at runtime
- > Feed the runtime data into a reconciliation process that compares what it is Vs what it should be
- > Address any discrepancies (closed loop governance)

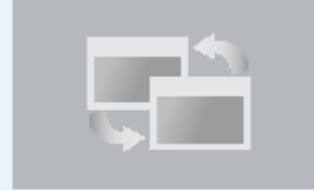
Agenda

- > Brief recap of SOA concepts and benefits
- > Challenges encountered in large scale SOA deployments
- > How is eBay addressing these challenges
- > SOA governance and its role
- > Summary

Summary

- > Moving to SOA model certainly has its benefits, but brings along bigger challenges
- > Solving the technology part is relatively easy. Operational aspects are more challenging and need to address them from the beginning
- > Needs for internal/external services, different velocity changes etc., need to be considered from the get go.
- > Up front design and modeling of your contract/interface, including granularity is very important
- > Service layering, dependency and version management must be well thought
- > Invest upfront on governance, Testing tools, and developer training

VJET brings what developers love about Java™ and JavaScript™ together



**Hear straight from the architects of VJET and you may
walk away with a Sony PLAYSTATION®3**

TS-5047

VJET—bringing the best of Java and JavaScript together
Yitao Yao, Justin Early

Date: Wednesday, June 03
Time: 1:30 - 2:30 PM

Don't miss our other eBay experts on Java, SOA, and scalability

TS-4476

SOA Deployment Challenges in the Real World
Sastry Malladi

Date: Tuesday, June 02
Time: 3:20 - 4:20 PM

BOF-4682

Performance Comparisons of Dynamic Languages
on the Java Virtual Machine
Michael Galpin

Date: Wednesday, June 03
Time: 6:45 - 7:35 PM

BOF-5273

SOA Error and Fault Management
Bhaven Avalani

Date: Wednesday, June 03
Time: 8:45 - 9:35 PM

TS-4407

Best Practices for Large-Scale Web Sites:
Lessons from eBay
Randy Shoup

Date: Thursday, June 04
Time: 2:50 - 3:50 PM

TS-4620

Robust and Scalable Concurrent Programming:
Lessons from the Trenches
Sangjin Lee, Debashis Saha, Mahesh Somani

Date: Friday, June 05
Time: 12:10 - 1:10 PM

Visit us at booth # 733 to get a demo of VJET &
enter to win a Flip Video™ camcorder.



JavaOneSM

Thank You

Sastry Malladi

Distinguished Architect, eBay

smalladi@ebay.com