Service Virtualization: Separating Business Logic from Policy Enforcement

K. Scott Morrison

VP of Engineering
& Chief Architect
Layer 7 Technologies
www.layer7tech.com

Ron Ten-Hove

Senior Staff Software Engineer
Sun Microsystems, Inc.
www.sun.com

TS-8459

Goal of This Talk

Learn How Service Virtualization Can Help You to Securely Manage Service-Oriented Architectures

Agenda

Service Delivery: Separation of Concerns

➡ Declarative Policy

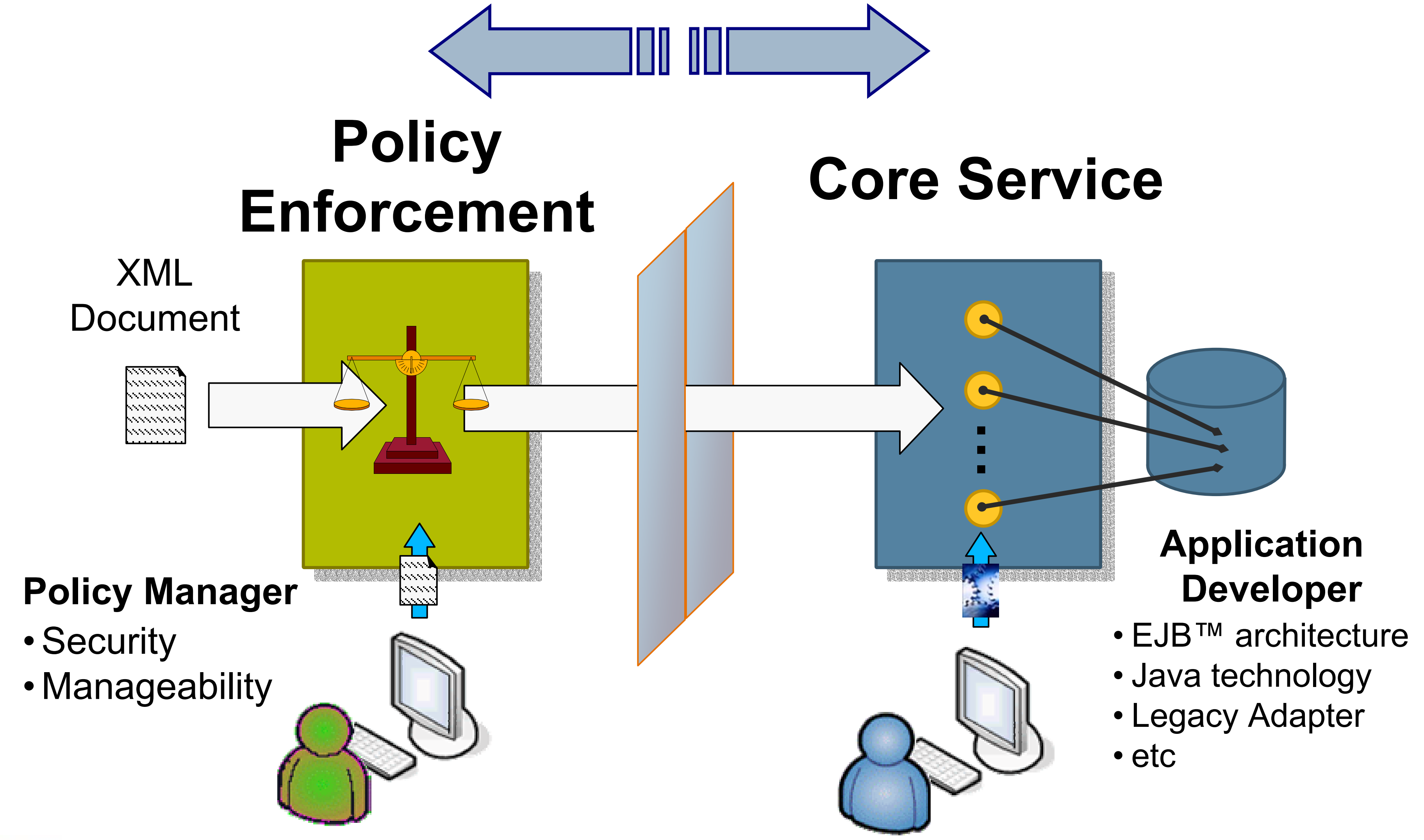
Virtualization and Policy Enforcement

Deployment Strategies

Java™ technology based XML Appliances for Policy Enforcement

Benefits and Costs

Service Delivery: 2 Separate Concerns



Policy Enforcement

Core Service

XML Document

Policy Manager

- Security
- Manageability

Application Developer

- EJB™ architecture
- Java technology
- Legacy Adapter
- etc

These Are Fundamentally Different

Policy

Dynamic and run time

- Security
  - AuthN/AuthR, integrity, confidentiality, key mgmt, audit, etc.
- SLA, QoS
  - Throughput limits, traffic shaping
- Application routing
- Versioning

➡ ***A continuously evolving problem***

Core Service

Static and design time

- Data binding
  - Java application environment to/from XML
- Transport handling
  - HTTP, Java™ Message Service (JMS) handling
- Localized “routing”
  - Mapping of service to local EJB architecture, Java code, legacy adapter, etc.

➡ ***A largely solved problem***

Consider Security, For Example:

- **Remember:** OASIS WS-Security (WSS) is about integrating and accommodating different security models
- Authentication
  - HTTP basic and digest
  - WSS UTP, x509, Kerberos, SAML, REL, etc
- Authorization
  - LDAP, Sun Java™ System Access Manager, MSAD, etc (very long list…)
- Confidentiality and Integrity
  - SSL/TLS, W3C XML encryption, canonicalization and signing

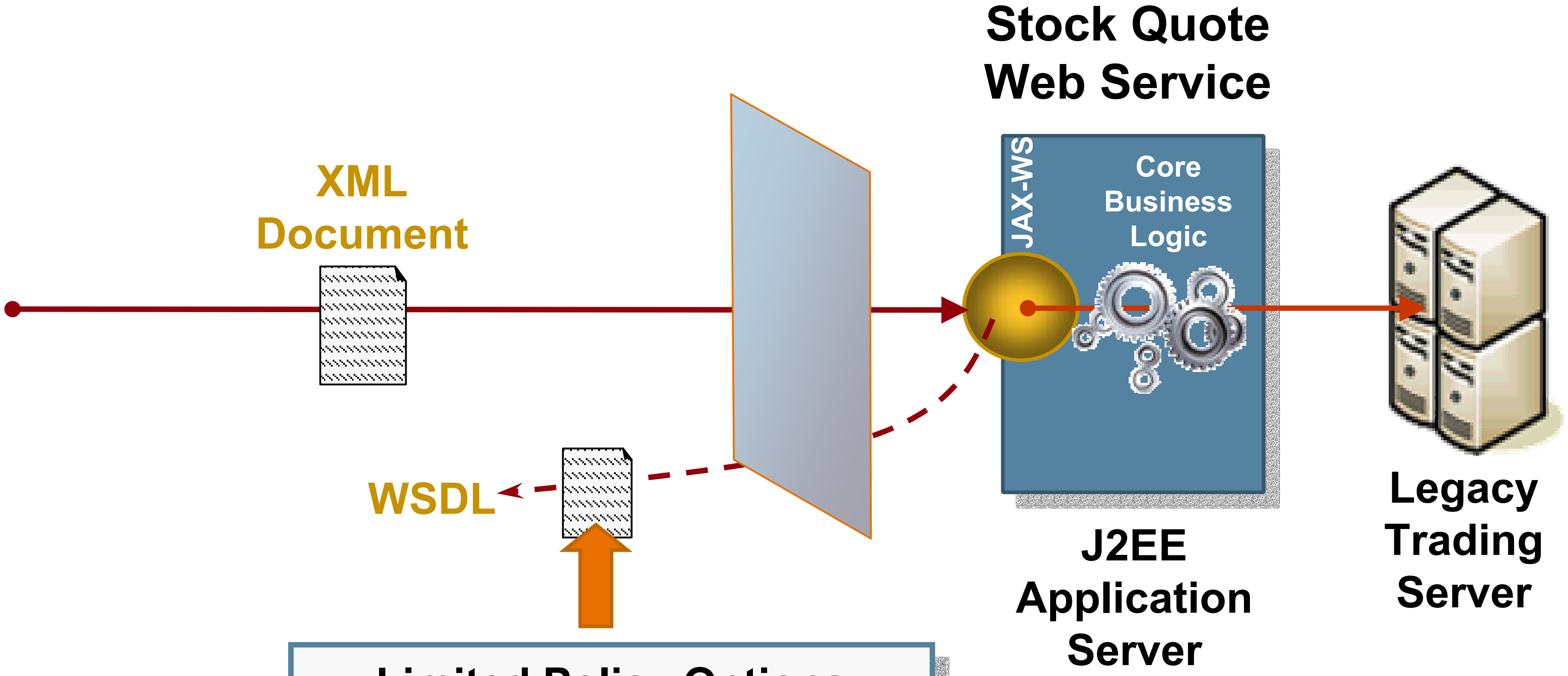
➡ ***A problem of enormous breadth and complexity…***

Add to This Emerging Threat Vectors

- API discovery attacks
  - WSDL, UDDI
- Direct assaults on an API
  - Replay, parameter substitution
- Denial of Service (DoS)
  - Numerous parser-based attacks, such as recursive payload, oversized payloads, coercive parsing, etc.
- Reference substitutions
  - STRs (both inside and outside messages), external entities, Xincludes, etc.
- Content attacks
  - SQL injection, XQuery injection, schema poisoning, virus/trojan/spyware embedded inside attachments and message content, etc.
- Compromise of participants
  - A particular issue for intermediates in a multi-hop transaction

Benefits of Declarative Policy 1

Just basic, application server-based policy



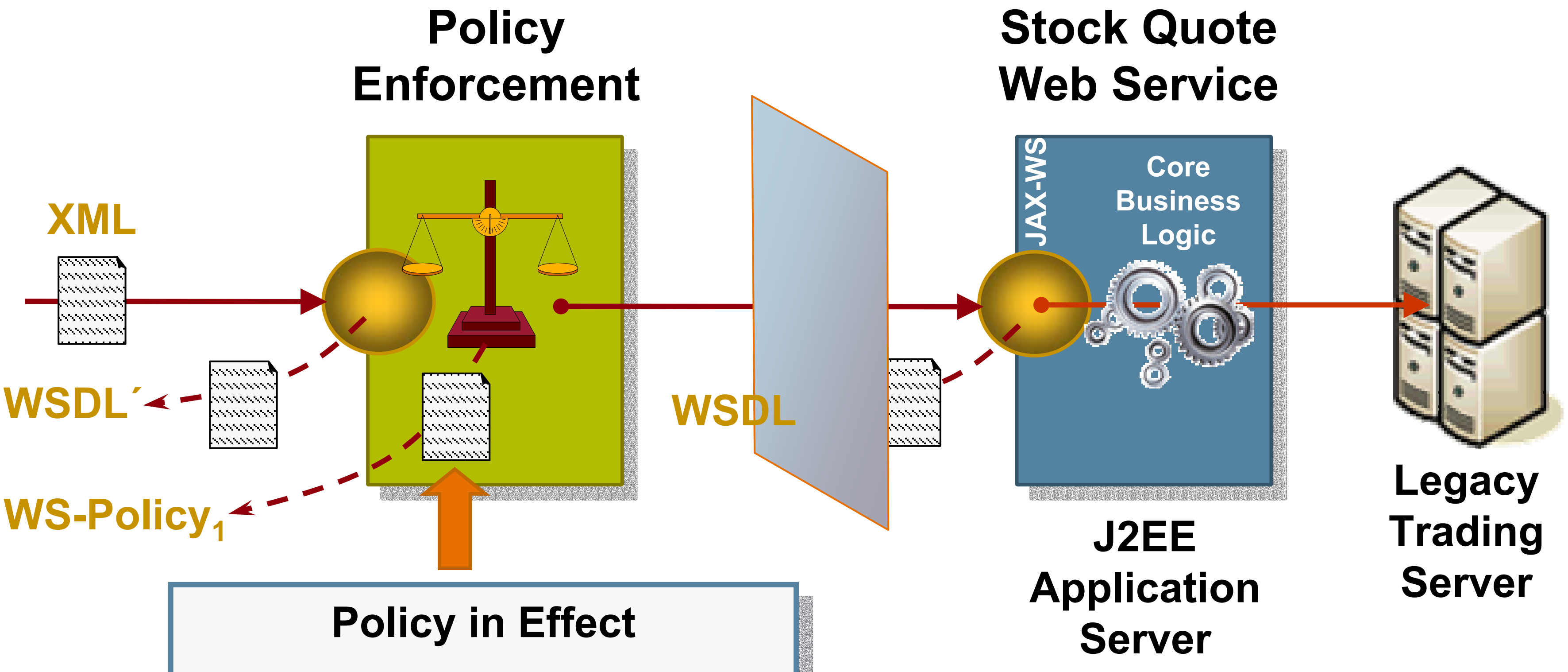
Limited Policy Options

✓Simple HTTP Authentication

✓SSL

Benefits of Declarative Policy 2

Add policy enforcement layer

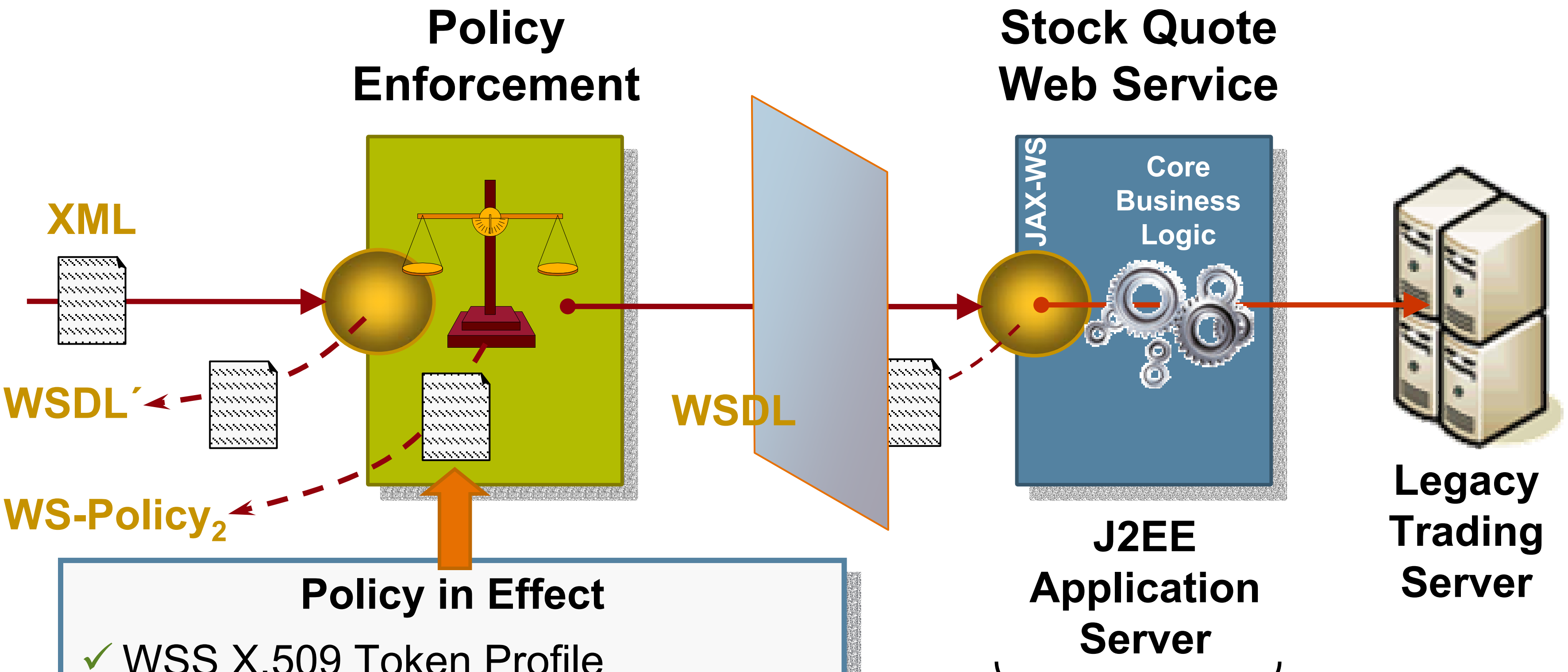


Policy in Effect

✓Anonymous access

✓No confidentiality or integrity

Benefits of Declarative Policy 3



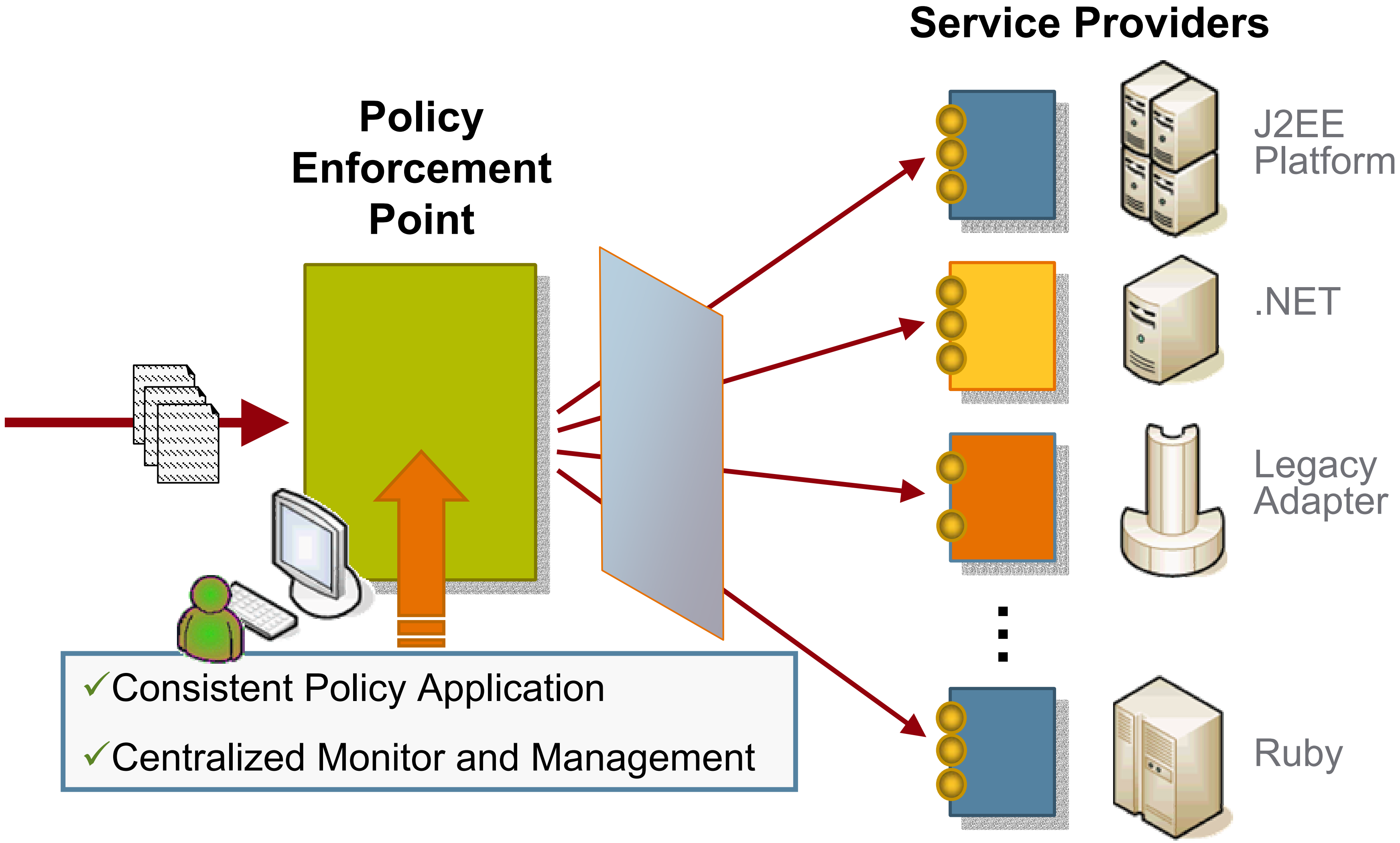
Policy in Effect

- ✓ WSS X.509 Token Profile
- ✓ Encrypted `<SOAP:Body>` content using AES256 and EncryptedKey
- ✓ Timestamps in `<SOAP:Header>`

$\underbrace{\text{J2EE Application Server}}$

No changes to implementation

Here Is When This Really Shines:



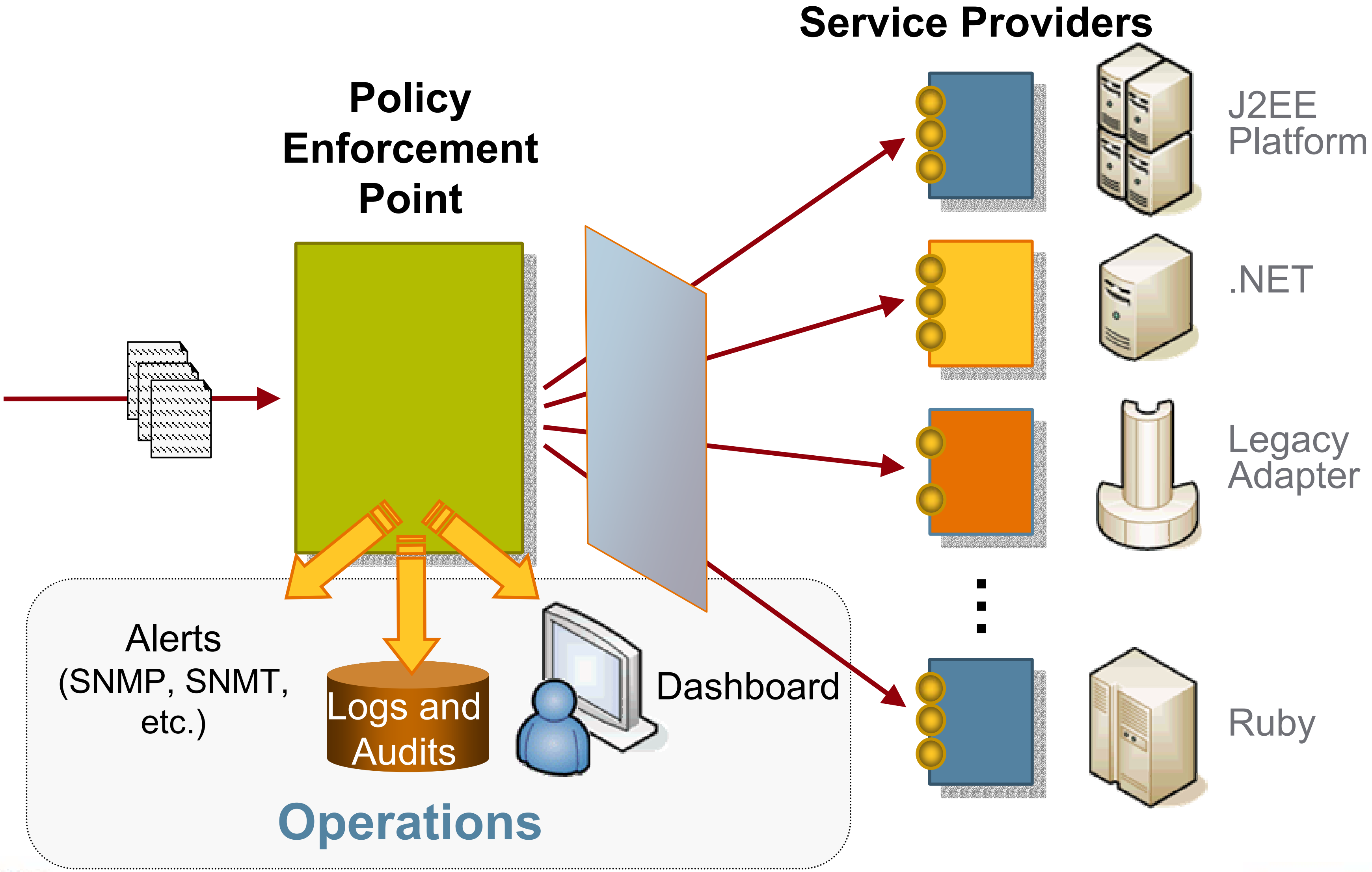
✓Consistent Policy Application

✓Centralized Monitor and Management

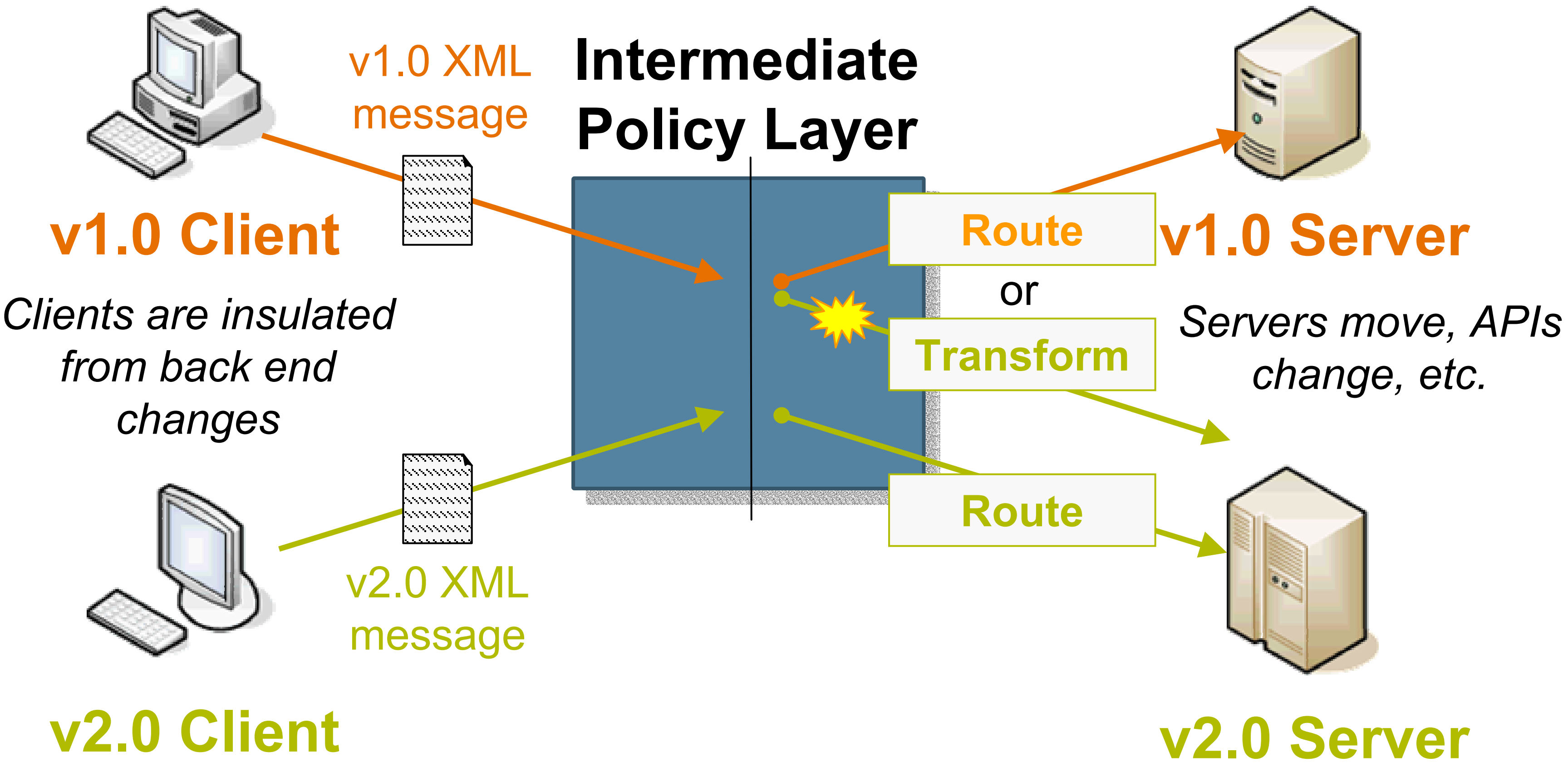
Service Virtualization Is a View of a Service Managed Through Policy

- Policy can be anything applied to a stream of XML
  - Extract credentials, authenticate and authorize
  - Decrypt, validate signatures
  - Schema validate, scan for threats
  - Transform
  - Route, etc
- Policy is declarative
  - Determined at run time
  - Easy to change
- Policy is administrative
  - Not programmatic; it is not implemented in your Java code!
  - Written by a dedicated security administrator
- Policy is effectively an Aspect of a service

Mgmt. of Services: Monitoring and Audit



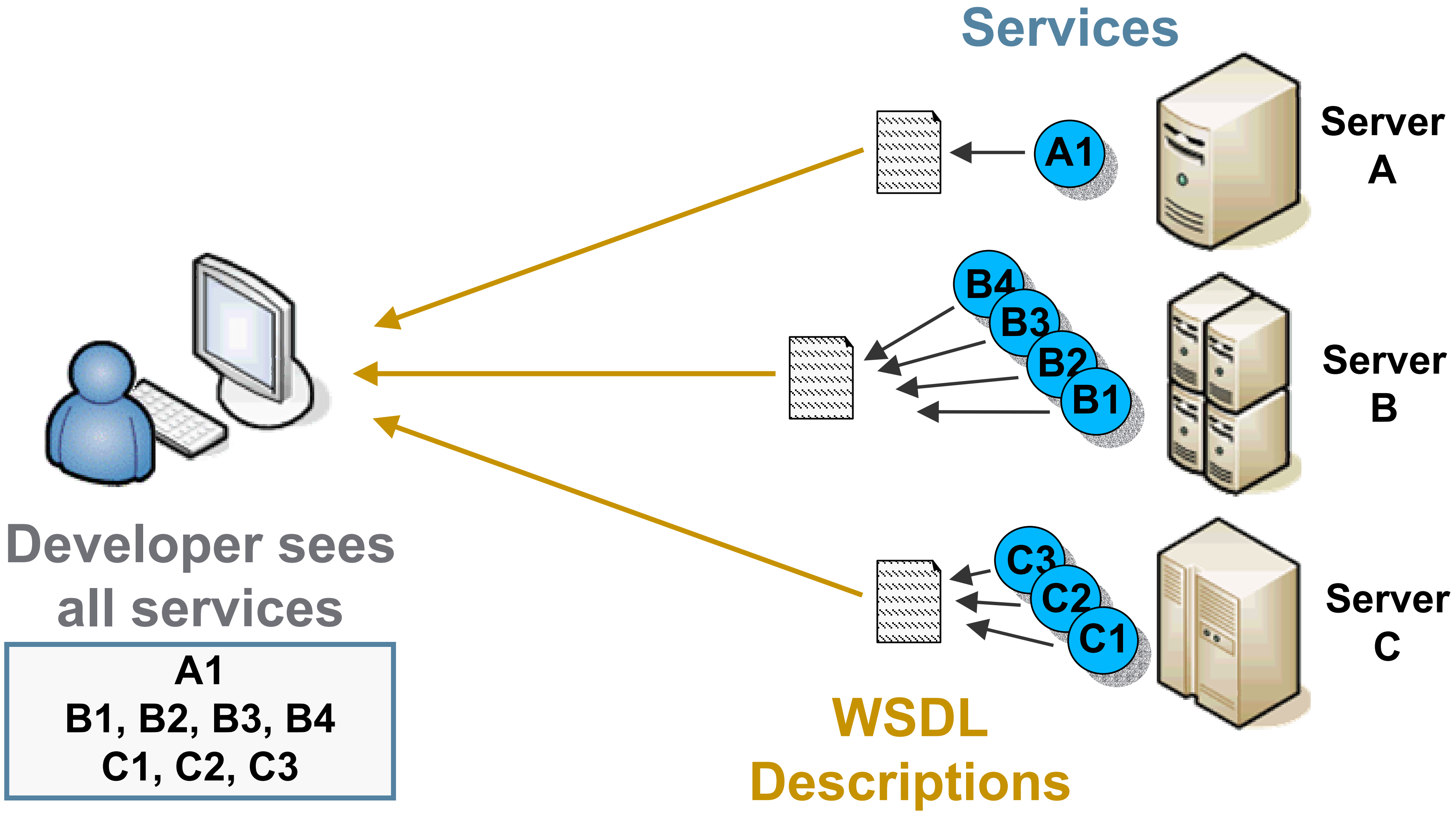
Service Versioning: Dealing with Change



Clients are insulated from back end changes

Servers move, APIs change, etc.

Service Virtualization



Services

Server A

A1

B4 B3 B2 B1

Server B

C3 C2 C1

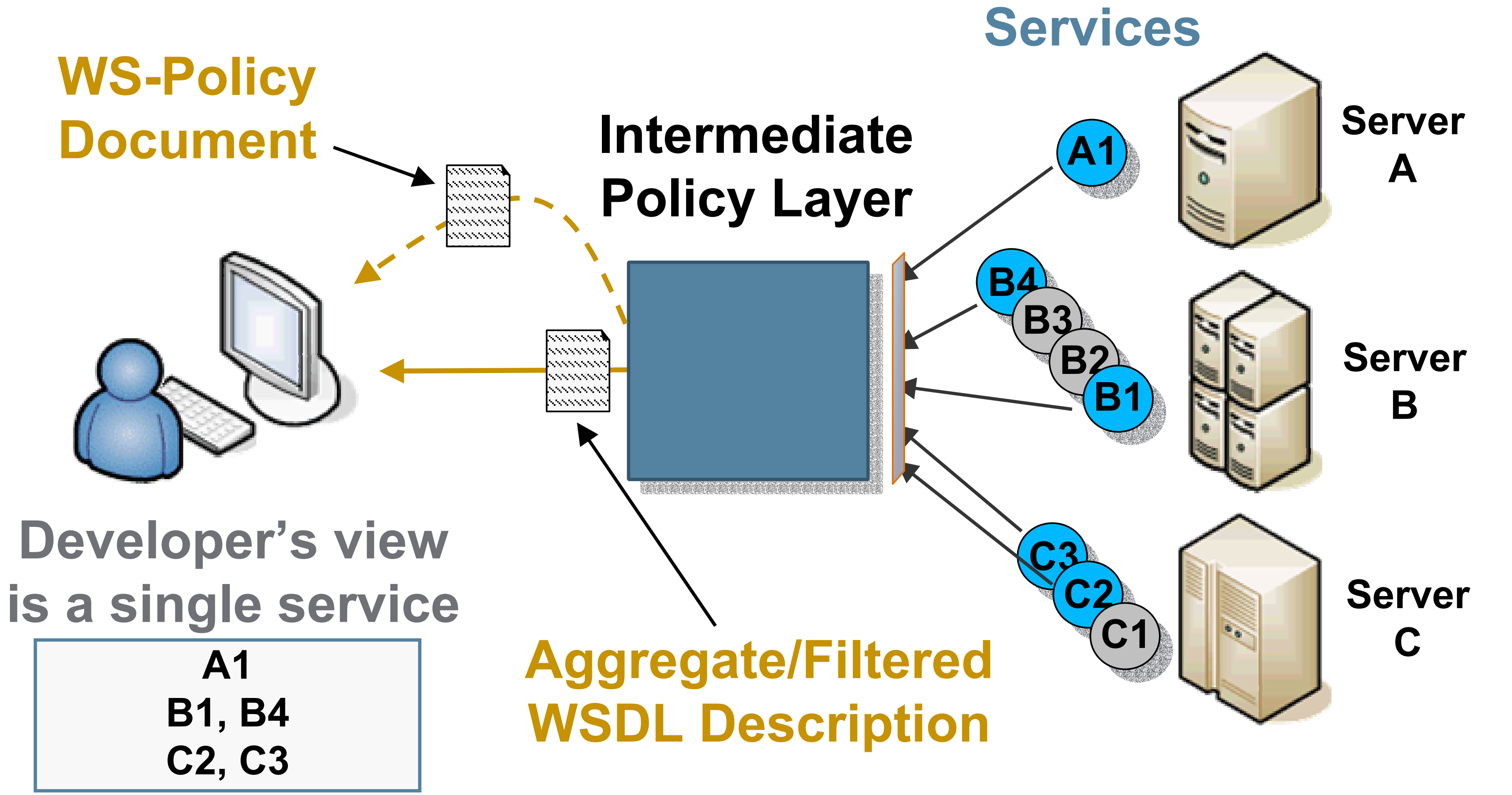
Server C

Developer sees all services

A1
B1, B2, B3, B4
C1, C2, C3

WSDL Descriptions

Service Virtualization



Services

WS-Policy Document

Intermediate Policy Layer

Server A

Server B

Server C

A1

B4 B3 B2 B1

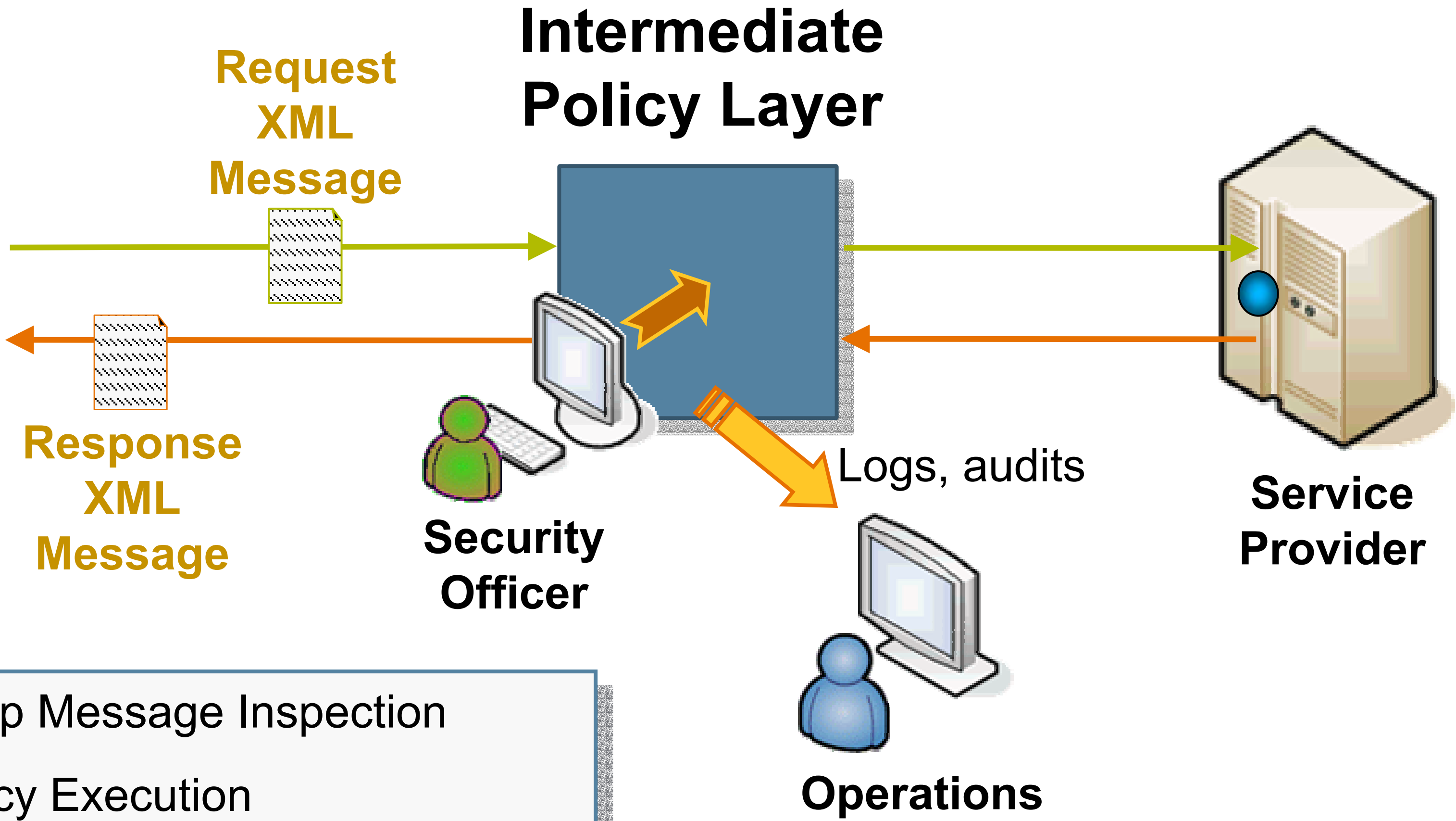
C3 C2 C1

Developer’s view is a single service

A1
B1, B4
C2, C3

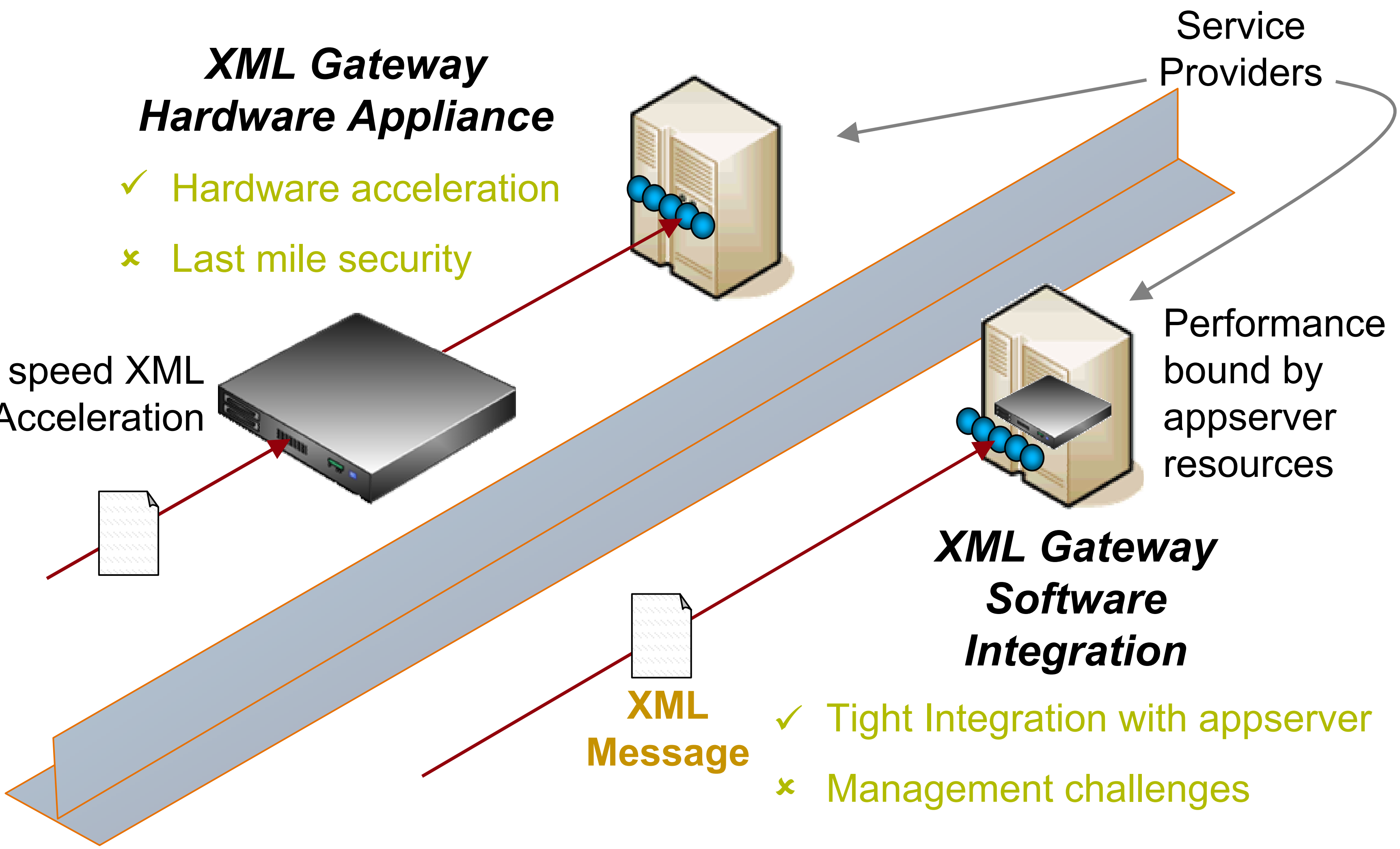
Aggregate/Filtered WSDL Description

How It Works



- ✓Deep Message Inspection
- ✓Policy Execution
  - – Security, Manageability, etc.
- ✓Declarative Policy Authoring

Concrete Infrastructure



XML Gateway Hardware Appliance

- ✓ Hardware acceleration
- × Last mile security

Service Providers

Wire speed XML Acceleration

Performance bound by appserver resources

XML Message

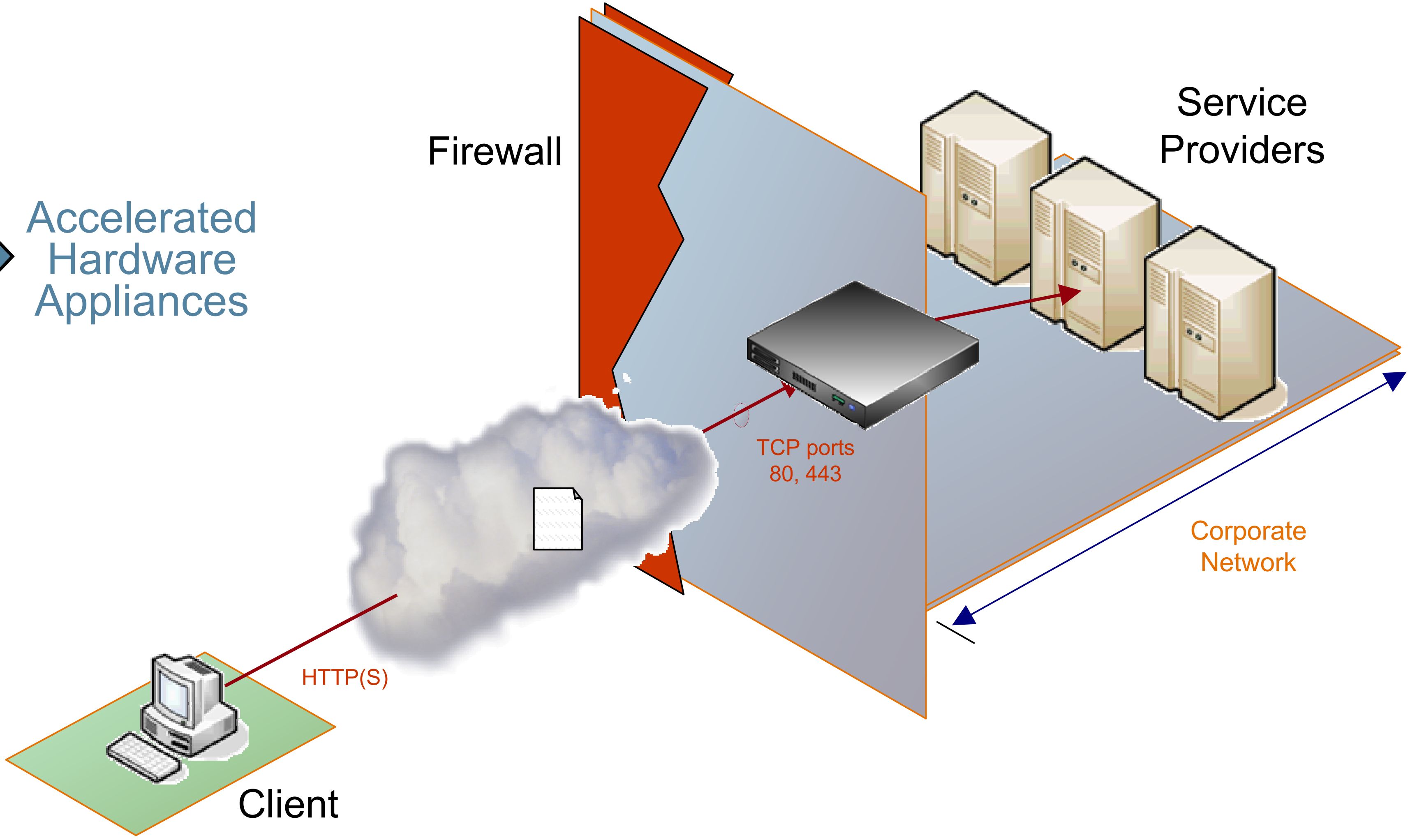
XML Gateway Software Integration

- ✓ Tight Integration with appserver
- × Management challenges

Patterns of Virtualization 1

At the edge of the network

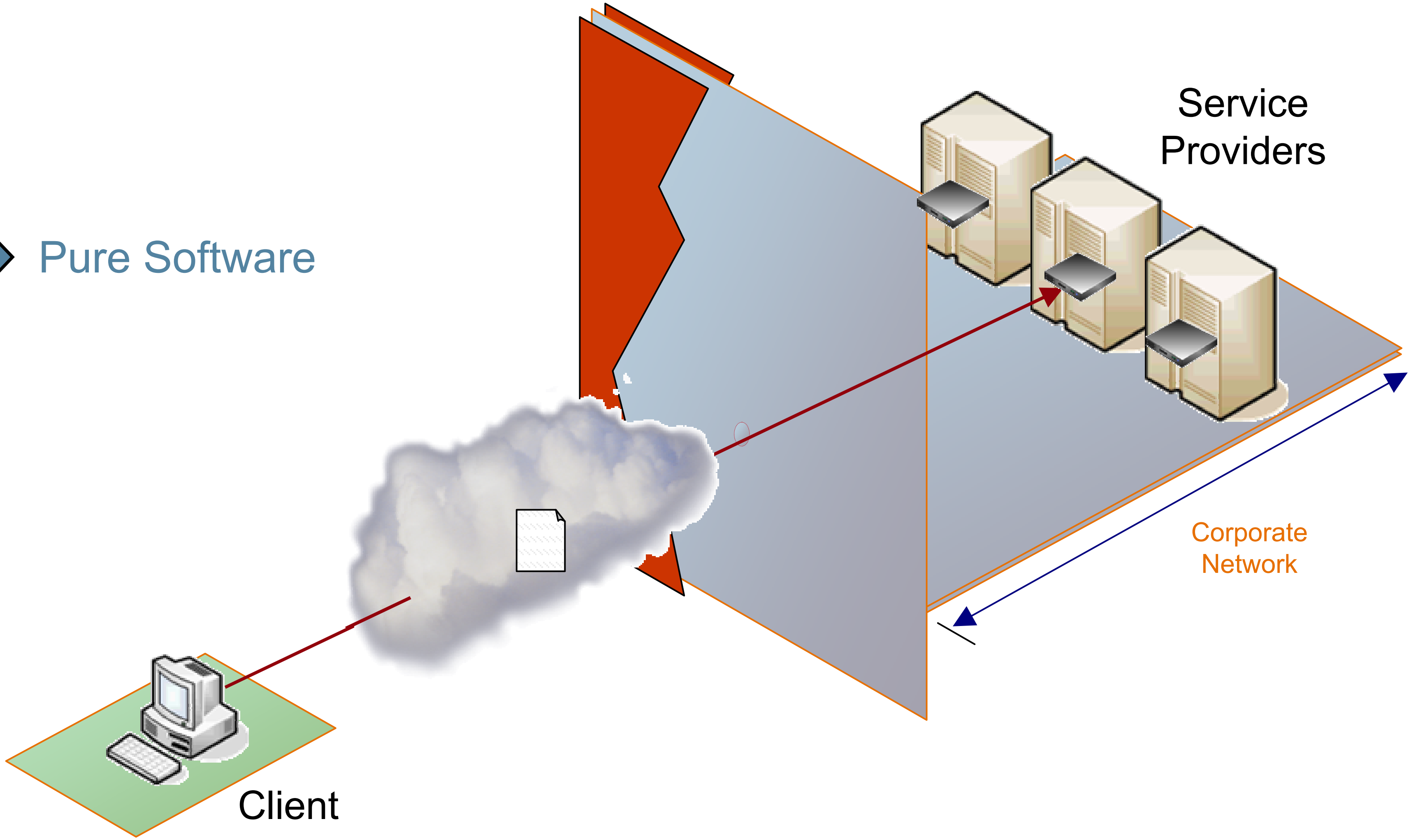
➡ Accelerated Hardware Appliances



Patterns of Virtualization 2

Co-located within the service provider

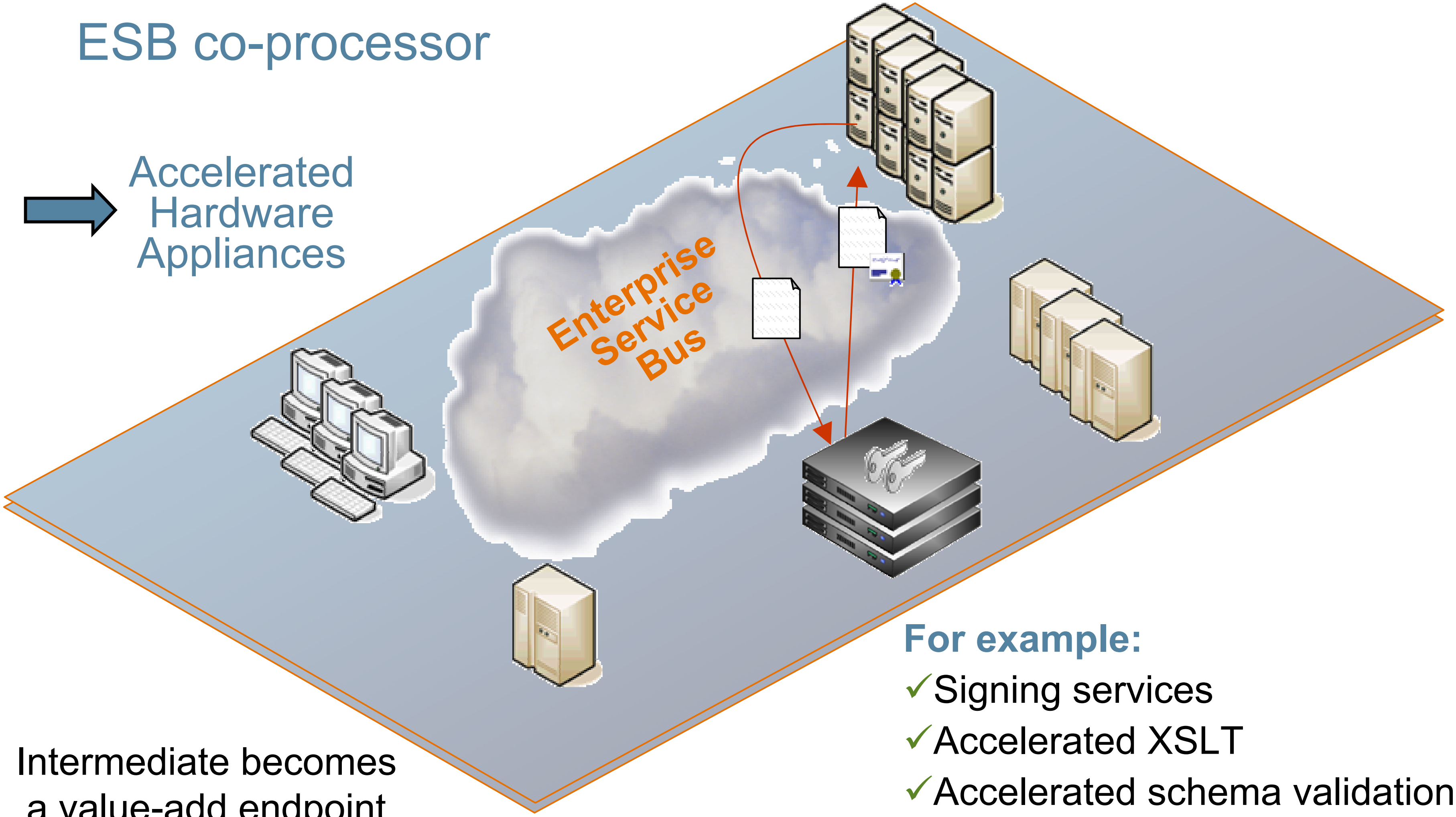
- Pure Software



Patterns of Virtualization 3

ESB co-processor

⇨ Accelerated Hardware Appliances



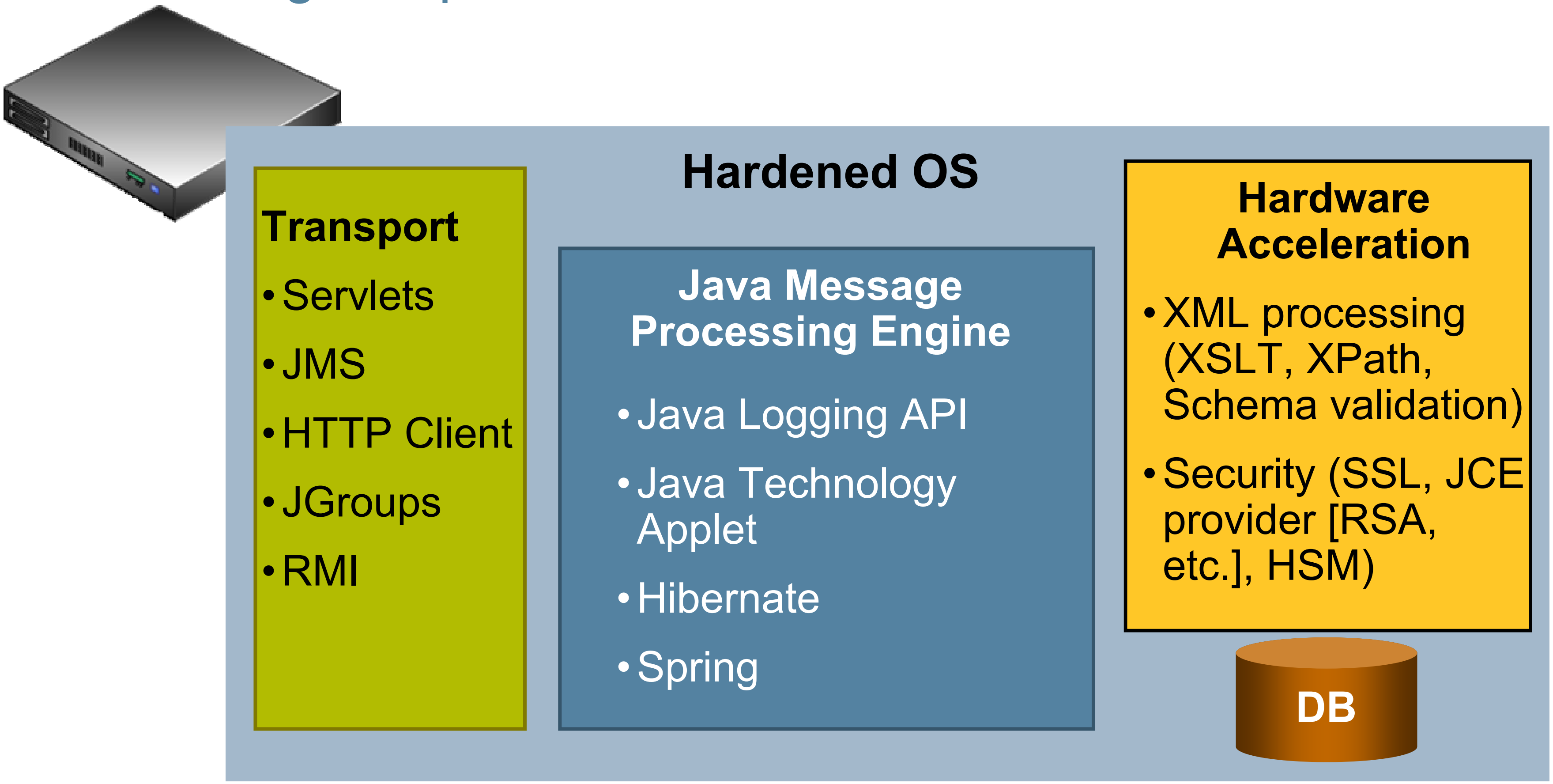
Intermediate becomes a value-add endpoint

For example:

✓Signing services
✓Accelerated XSLT
✓Accelerated schema validation
✓Document threat detection

Java™ Technology Based Virtualization Infrastructure

This is by no means exhaustive, but is just the more interesting components



Hardened OS

Transport

- Servlets
- JMS
- HTTP Client
- JGroups
- RMI

Java Message Processing Engine

- Java Logging API
- Java Technology Applet
- Hibernate
- Spring

Hardware Acceleration

- XML processing (XSLT, XPath, Schema validation)
- Security (SSL, JCE provider [RSA, etc.], HSM)

DB

Benefits and Costs

- Benefits
  - Centralization
  - Consistency
  - Manageability
- Costs
  - Separation from application
  - Scaling and fault tolerance demands

Summary

- **Service Virtualization** is really about creating new, managed views of services
- Management and security is best handled at a **Policy Enforcement Point** (PEP) that is separate from your code
  - This ensures policy is decoupled from the application
- Sun and Layer 7 Technologies have partnered to offer such infrastructure for security and management of services
  - And this is based on Java technology

Q&A

Ron Ten-Hove, Sun Microsystems

K. Scott Morrison, Layer 7 Technologies

Service Virtualization: Separating Business Logic from Policy Enforcement

K. Scott Morrison

VP of Engineering
& Chief Architect
Layer 7 Technologies
www.layer7tech.com

Ron Ten-Hove

Senior Staff Software Engineer
Sun Microsystems, Inc.
www.sun.com

TS-8459