ORACLE®

SOA Concepts – British Computer Society Advanced Programming Group

James Allerton-Austin
EMEA Fusion Middleware SOA Product Manager

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.
The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Recommended Reading

- The Definitive Guide to SOA: Oracle Service Bus (2nd Edition)
  Authors Davies, J; Schorow, D; Ray, S; Rieber, D

  ISBN: 978-1-43021-057-3

Agenda

- What is SOA?
- Separating concerns
- What is a Service?
- Beyond the Bus Service
- Summary
- Q&A

Service-Oriented Architecture (SOA)

Service-Oriented Architecture

A Business Lead **IT Strategy** that enables the creation of **loosely coupled**, **interoperable business services** that can be easily **shared** within and between enterprises to meet and **adapt** to business need.

What is SOA ?

Key areas

- Run-time re-use
- Discrete Business Services
- Reliable Infrastructure
- Service Lifecycle

Service Component Architecture

Composites, not monolithic applications

- Standard framework for “wiring” services
- Networks of services, provide capability
- Combining code and configuration
- Capability orchestrated from:
  - Service Enabled applications
  - Coded services
  - “Wiring” configuration

Agenda

- What is SOA?
- Separating concerns
- What is a Service?
- Beyond the Bus Service
- Summary
- Q&A

Separating concerns

Layering and separating responsibility

- Service Enablement
- Service Virtualisation
- Mediation
  - Translation
  - Transformation
  - Routing
  - Security
- Canonical Data

That doesn’t make sense to program again.

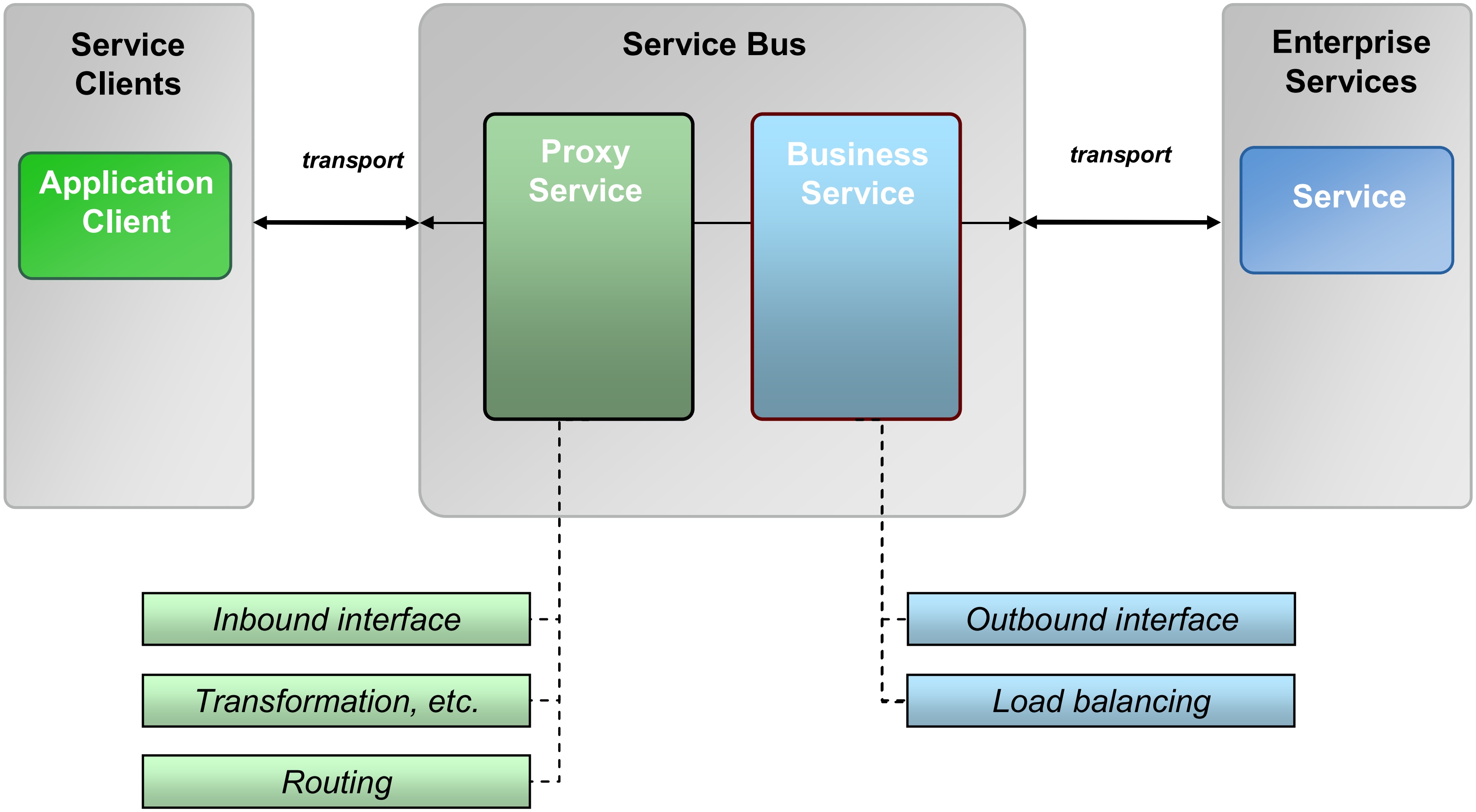
Separating concerns

Catching the Bus

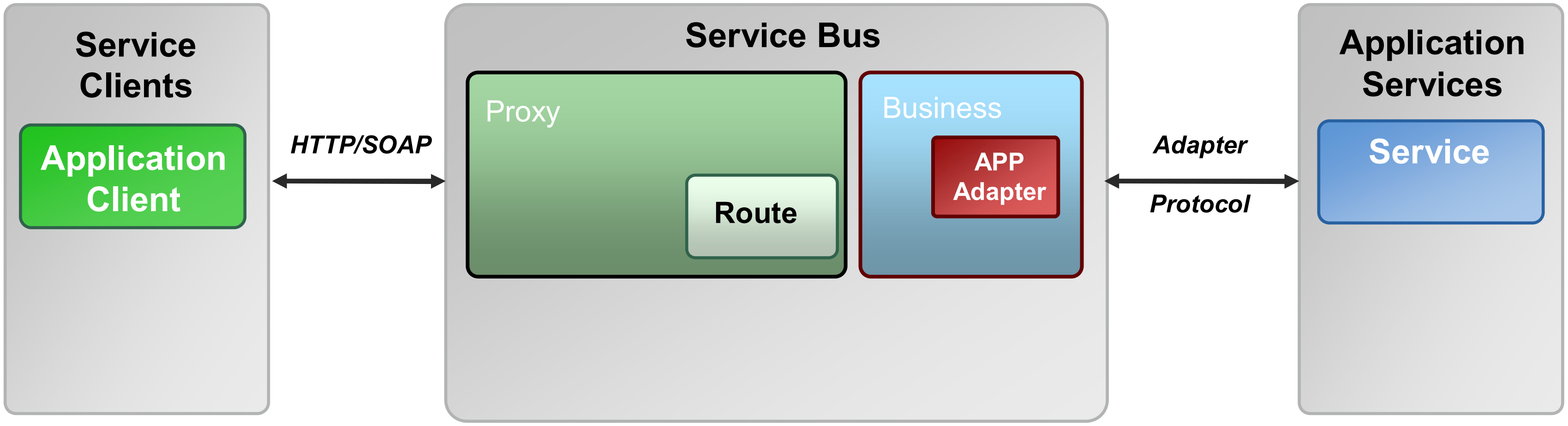
- Delivered with a modern Service Bus
- Features delivered:
  - Service Virtualisation
  - Mediation
  - Transformation
  - Service Brokering
  - Service Aggregation
  - Security
  - Monitoring
  - Load balancing
  - Configuration, rather than coding

The Service Bus

Delivering Service Virtualisation

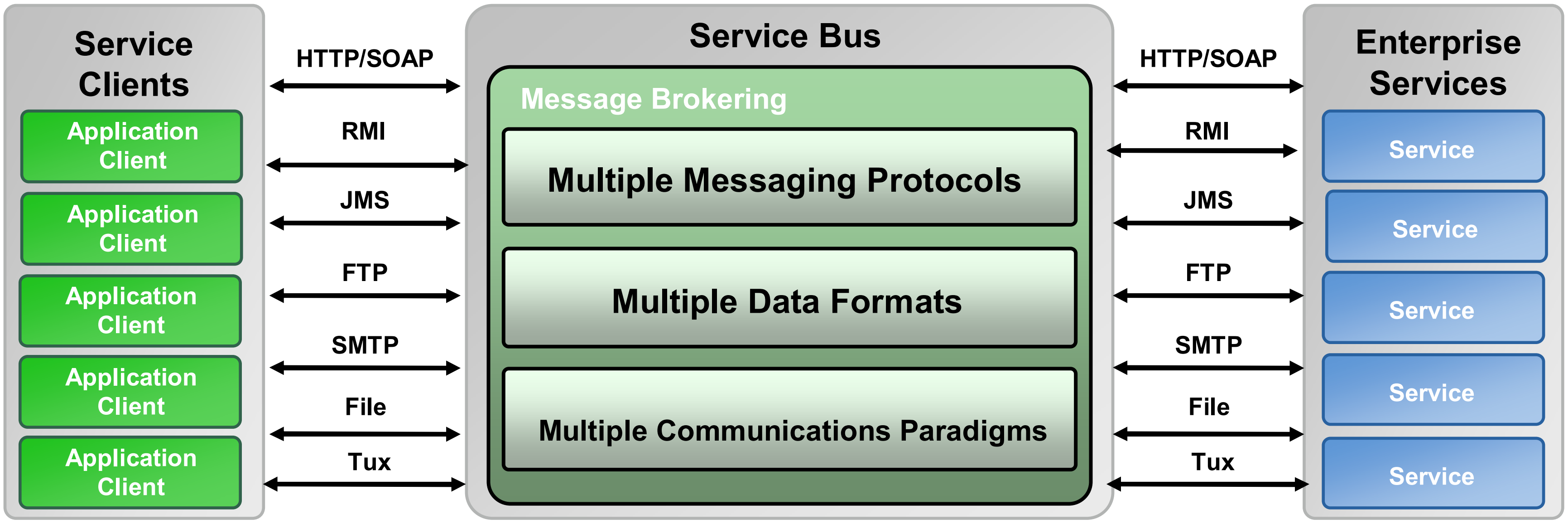


Separation of Concerns
Adapter based Integration



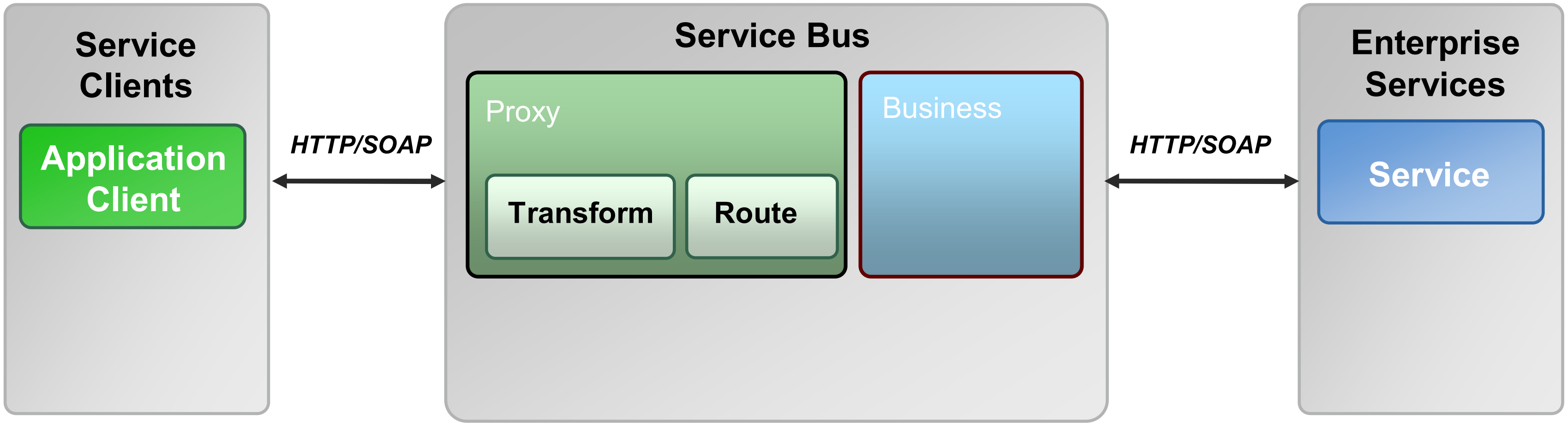
Separation of Concerns

Service Brokering



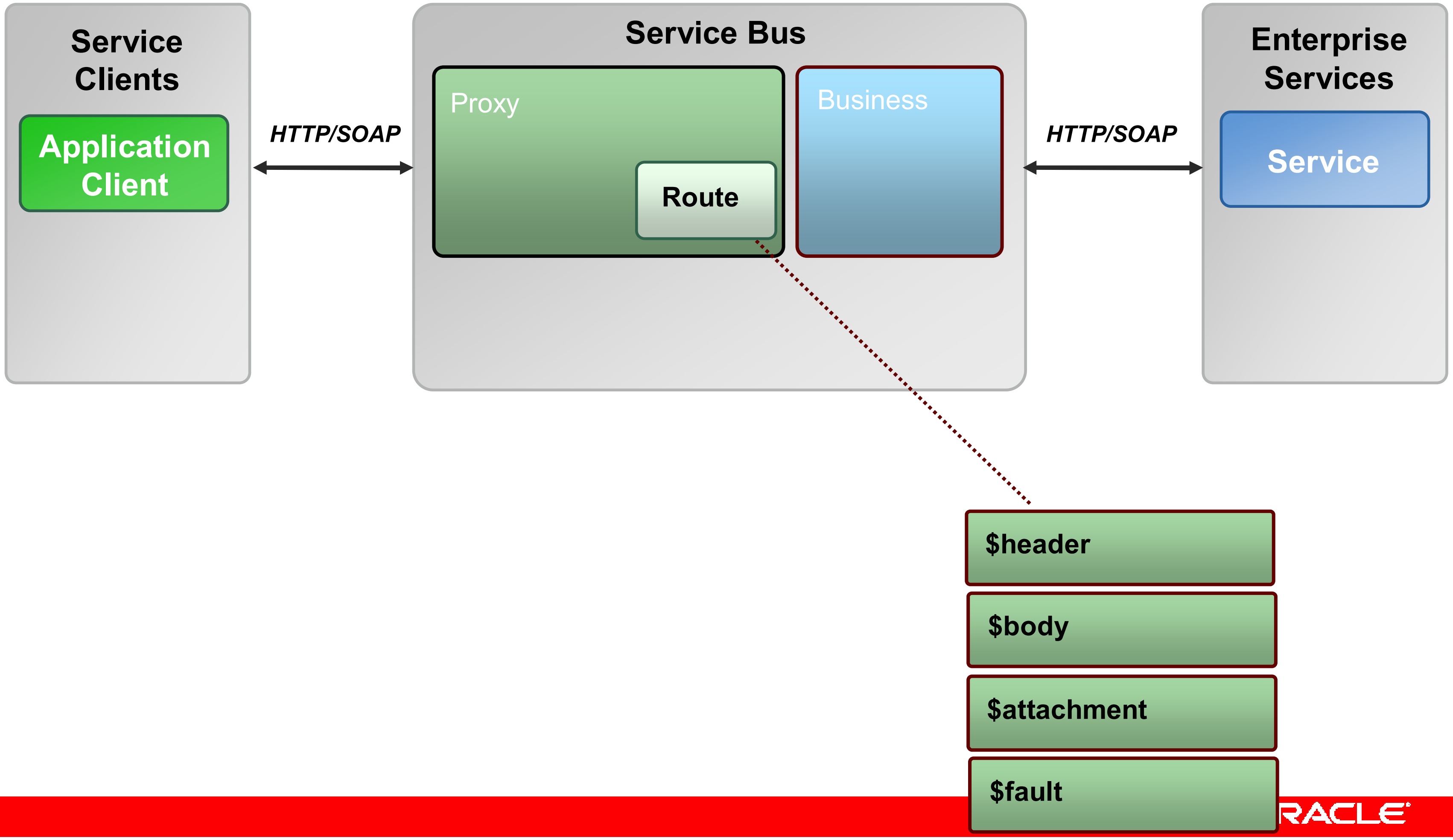
Application

Separation of Concerns
Message transformation



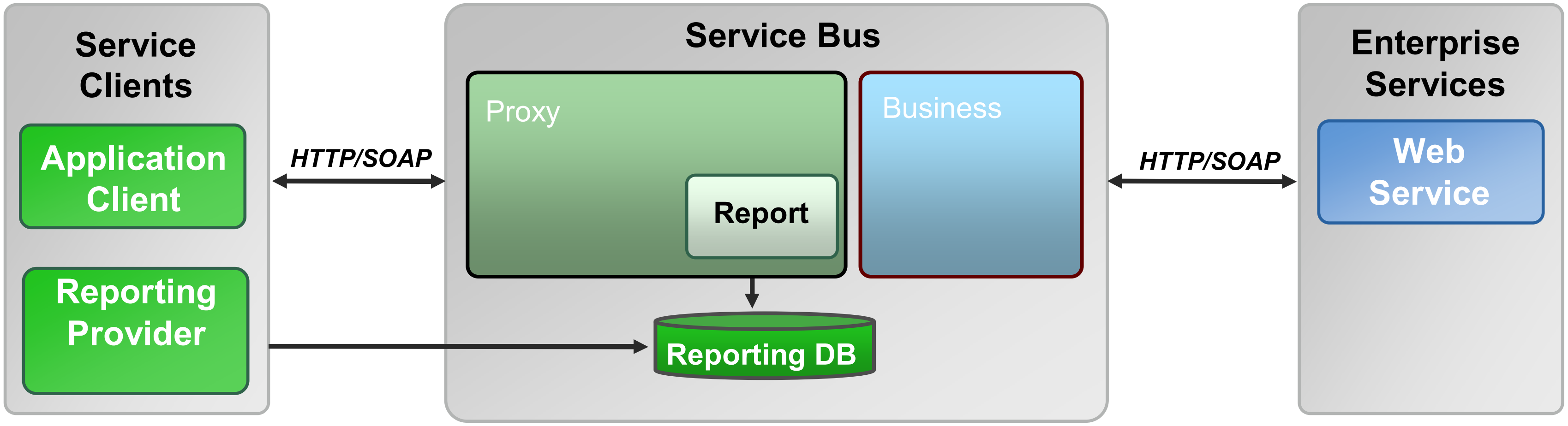
Separation of Concerns

Mediation and Routing

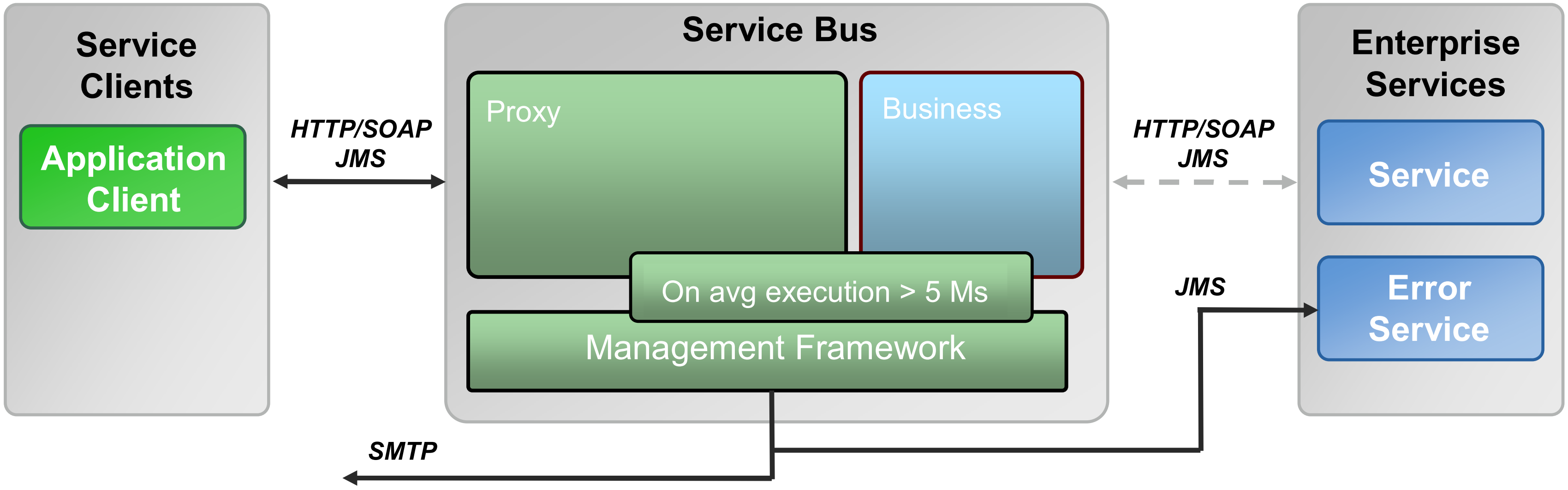


Separation of concerns

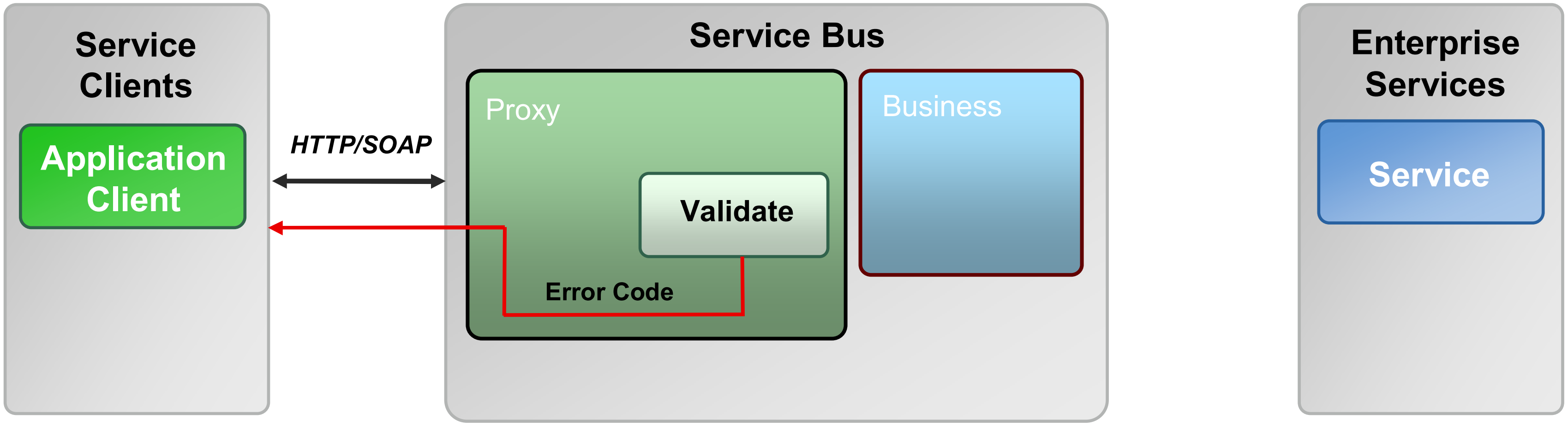
Monitoring Service Health



Separation of Concerns
Alerting SLA violations

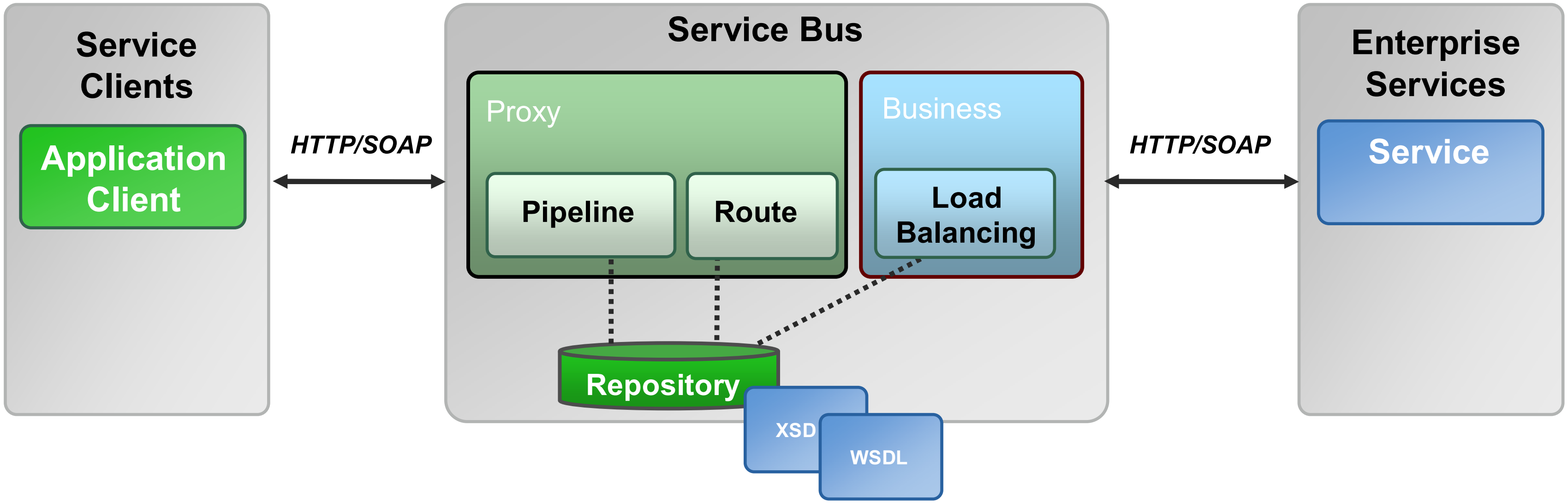


Separation of Concerns
Error Handling



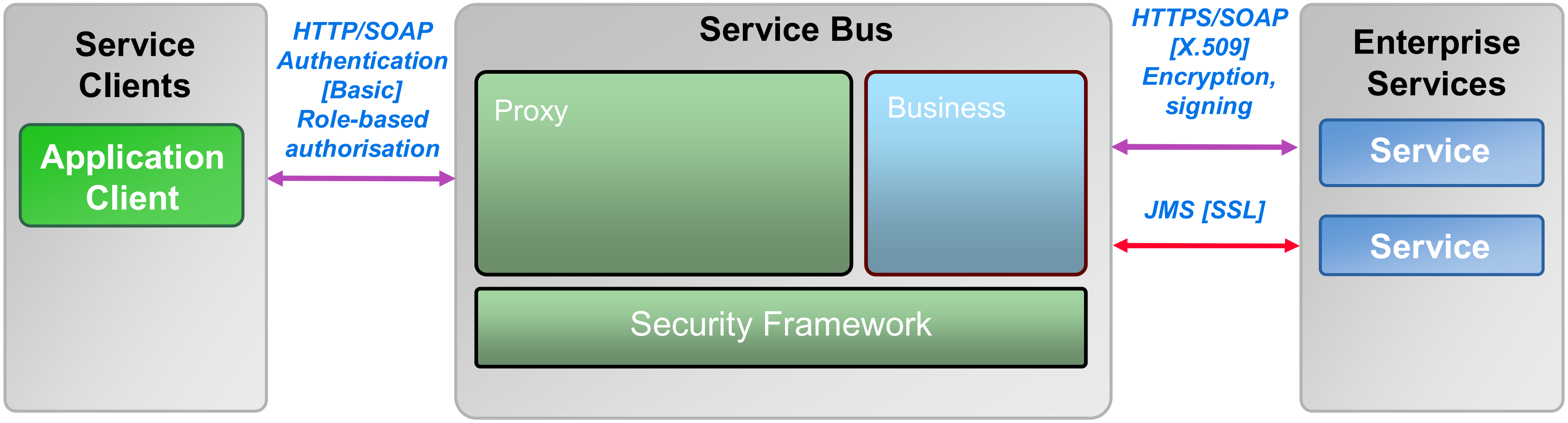
Separation of Concerns

Load balancing and dynamic service resolution



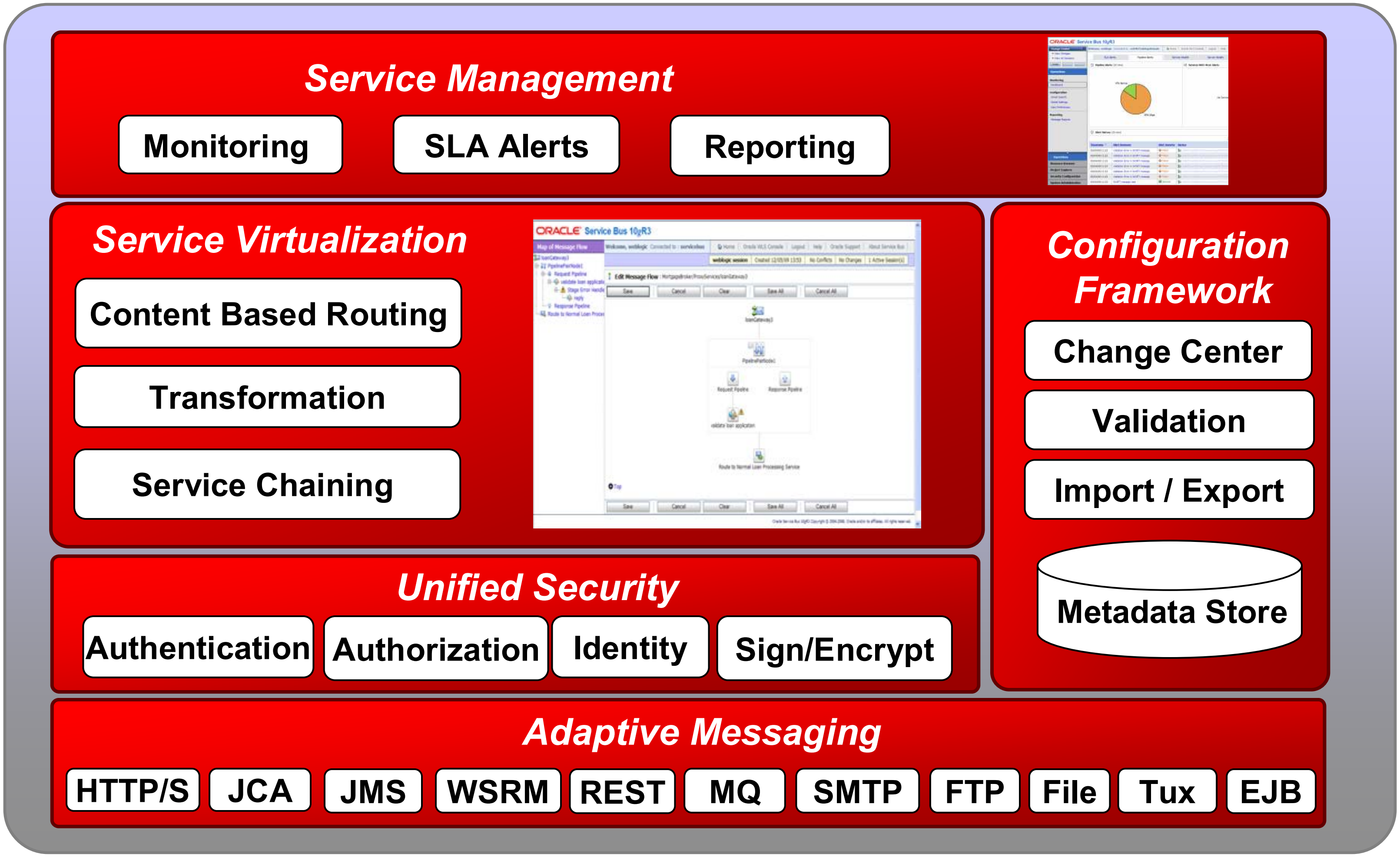
Separation of Concerns

Securing heterogeneous services



Oracle Service Bus

Key Ingredients for Enterprise Service Bus



Service Management

Monitoring | SLA Alerts | Reporting

Service Virtualization

Content Based Routing

Transformation

Service Chaining

Configuration Framework

Change Center

Validation

Import / Export

Metadata Store

Unified Security

Authentication | Authorization | Identity | Sign/Encrypt

Adaptive Messaging

HTTP/S | JCA | JMS | WSRM | REST | MQ | SMTP | FTP | File | Tux | EJB

Agenda

- What is SOA?
- Separating concerns
- What is a Service?
- Beyond the Bus Service
- Summary
- Q&A

What is a Service ?

- Service Definition
  - Determines the service boundaries.
  - Defines service contract.
  - Identifies service classification and scope for the service.

Benefits of Service Boundaries

Programming best practice

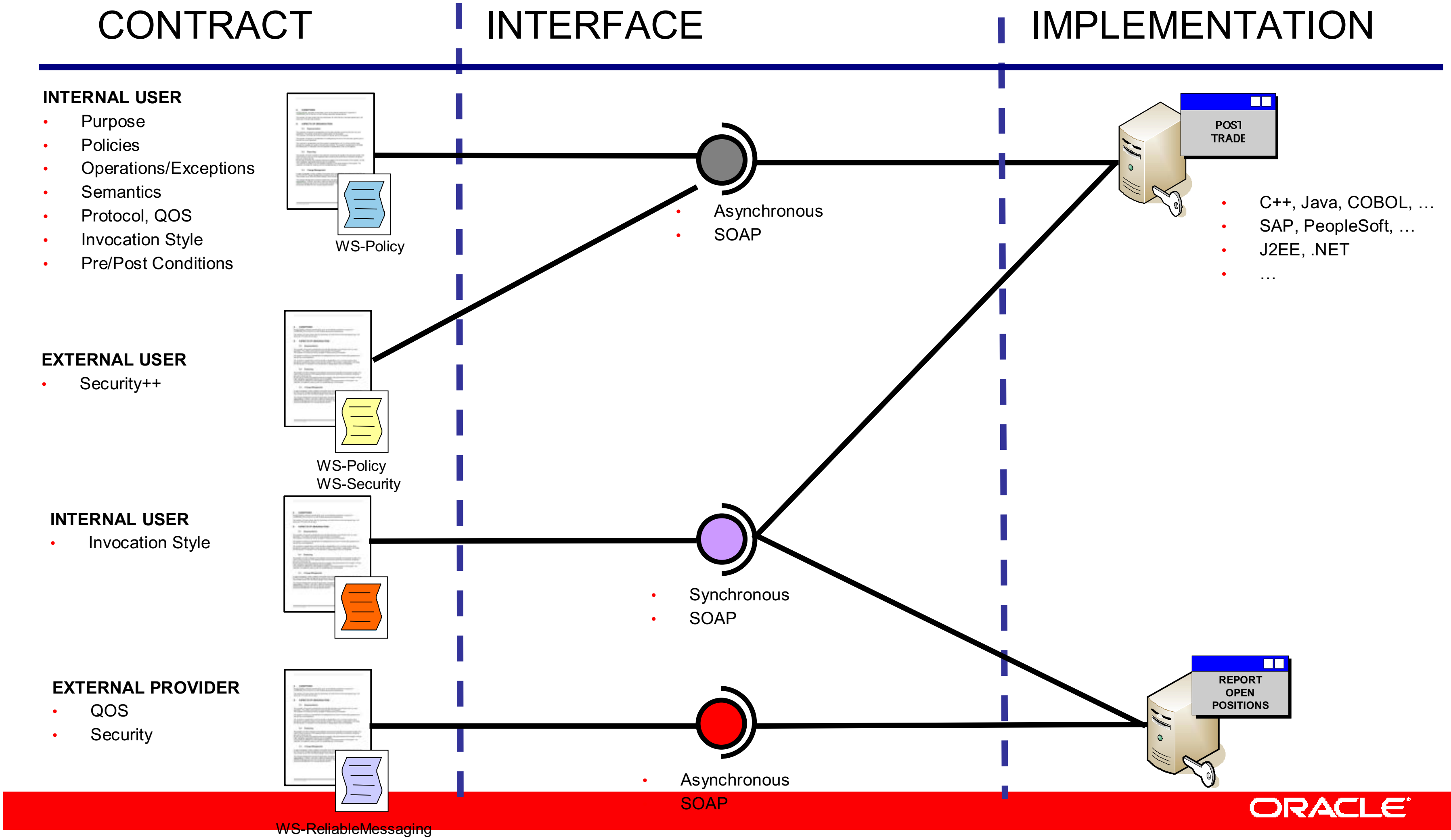
- Isolation
  - Enables services to physically scale or technologically evolve independently
  - Enables parallel development
- Encapsulation
  - Prevents unwanted side-effects since the service is self-contained
  - Enables the service implementation to evolve without impacting users
- Standardised interfaces
  - Enables quick composition of services within a process or service bus
  - Enables standardised service and business activity monitoring
- Managed dependencies
  - Enables effective impact analysis since dependencies can be registered, tracked, and managed
- Reduced system complexity
  - Prevents “hidden” dependencies that would otherwise complicate the delivery of requirements.

Service Design

- Service Design
  - Defines the service interface including message schemas
  - Determines the implementation approach
    - Java, .Net, composition, etc.
  - Determines the underlying resources used by the service
  - Covers both functional and non-functional requirements

What defines a Service?

Three key aspects



CONTRACT

INTERFACE

IMPLEMENTATION

INTERNAL USER

- Purpose
- Policies
- Operations/Exceptions
- Semantics
- Protocol, QOS
- Invocation Style
- Pre/Post Conditions

EXTERNAL USER

- Security++

INTERNAL USER

- Invocation Style

EXTERNAL PROVIDER

- QOS
- Security

- Asynchronous
- SOAP

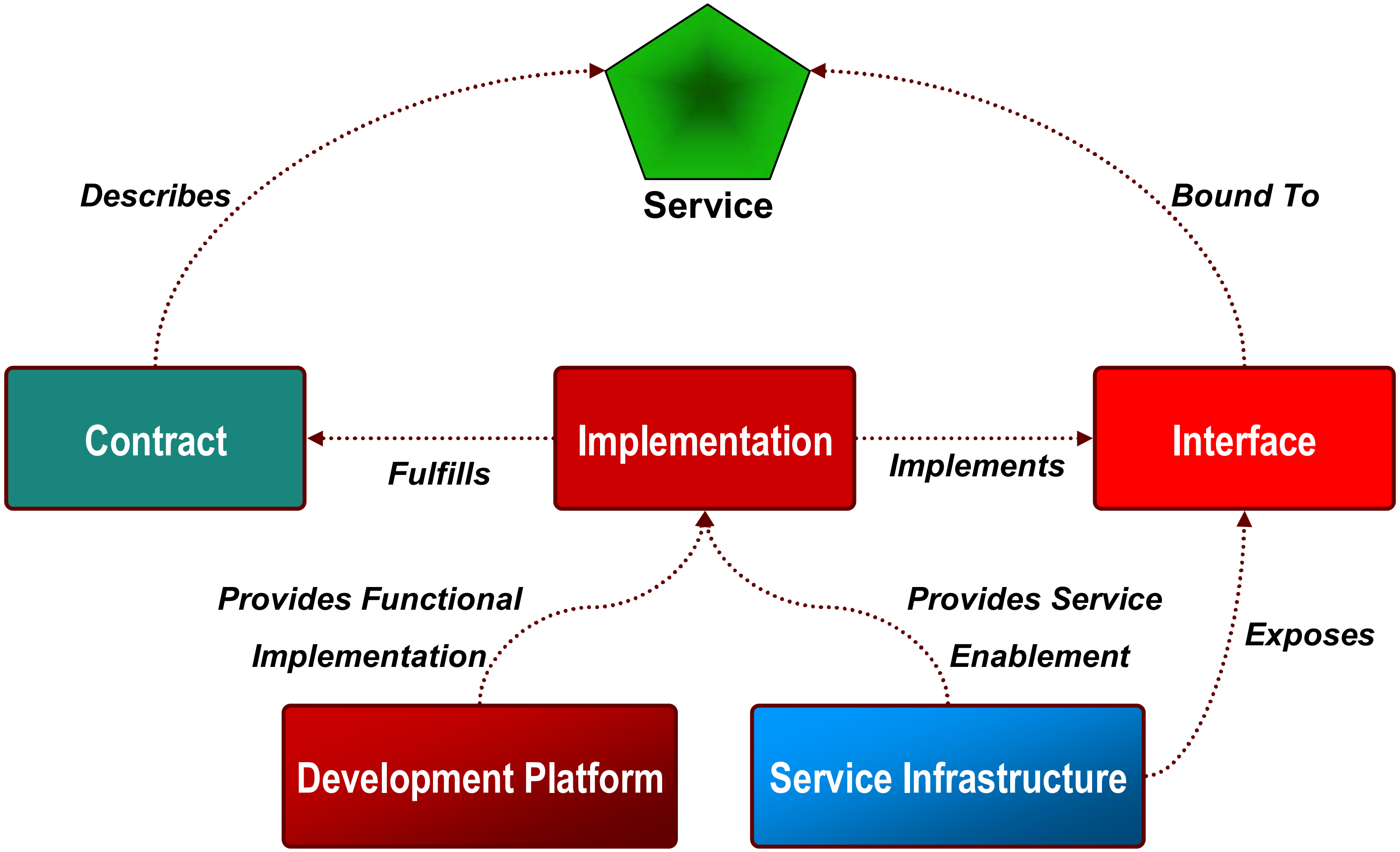
- Synchronous
- SOAP

- Asynchronous
- SOAP

- C++, Java, COBOL, …
- SAP, PeopleSoft, …
- J2EE, .NET
- …

Shared Service Relationships

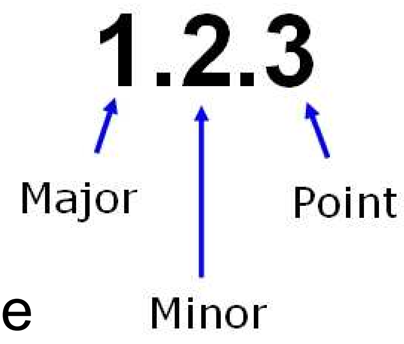
For a single service



Service Versioning

Service lifecycle

- Evolving services requires a service versioning strategy
  - Major Release
    - Breaks backward compatibility
    - If common indicates service identification is not working
  - Minor Release
    - New capability with backward compatibility
    - Adding new methods, not changing existing methods
  - Point Release
    - No new features
    - Bug fix or performance enhancement
- Service testing strategy is key to successful service evolution
  - Full service regression test for each release
  - Consumers retest for major and minor releases

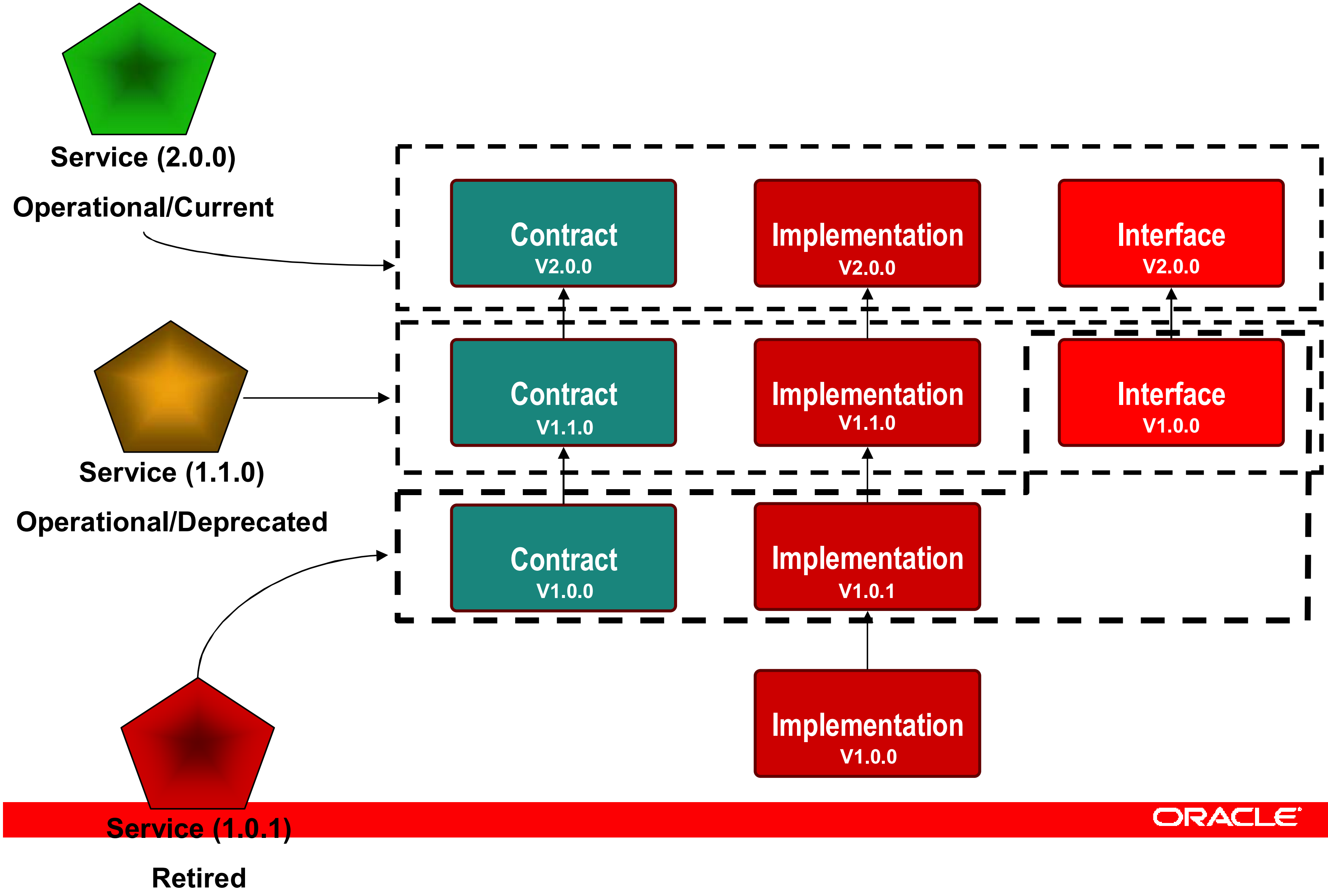


Service Versioning Principles

- Architecture should support several versions of a service being in production concurrently
  - Governance should limit the number of concurrent versions
  - Advantage of using a Service Bus to route requests
  - Consider adding a consistent message header for version information
- Consumers should not be forced to upgrade immediately
  - Consumers must migrate eventually
  - Services retired gracefully
- Consumers must be able to determine a suitable version
  - Advantage of using registry/repository for service lookup
- Services should strive for backward compatibility whenever possible
  - Point releases should never break backward compatibility

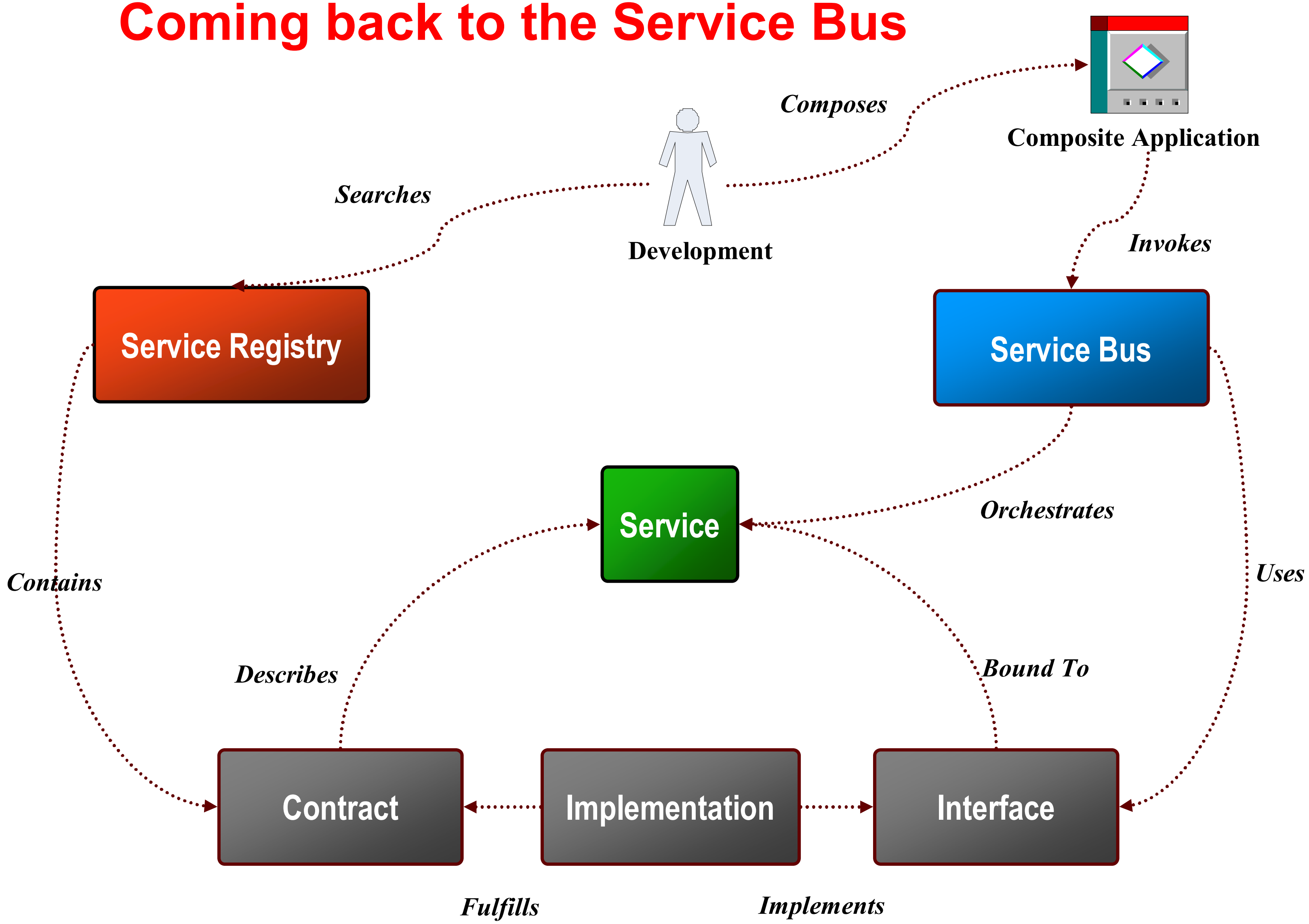
Versioning & Service Deprecation

The service lifecycle



Using a Service

Coming back to the Service Bus

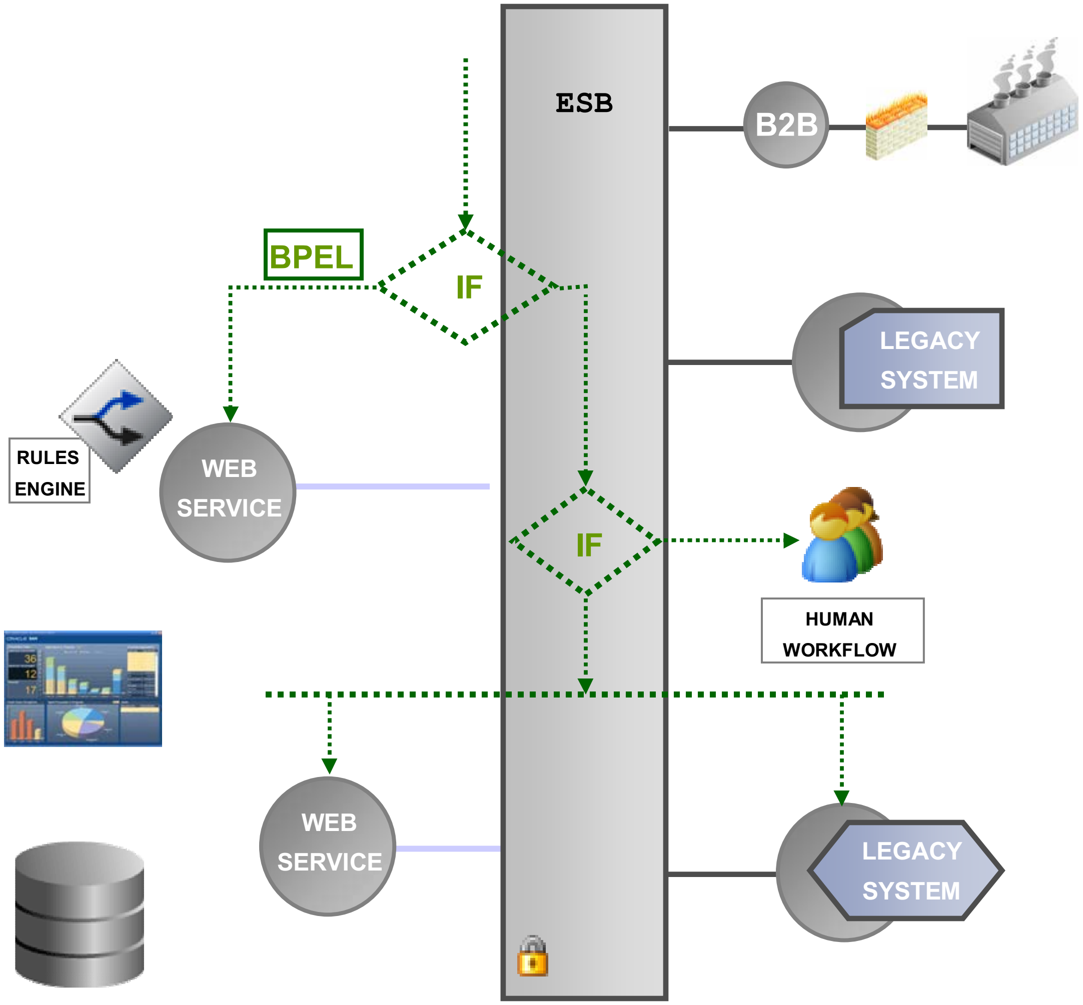


Agenda

- What is SOA?
- Separating concerns
- What is a Service?
- Beyond the Bus Service…
- Summary
- Q&A

Components of a SOA Infrastructure

There's more than a Bus!



Build/re-use Services
Web, Legacy or ext. Services

Connect
Adapters & B2B

Route, Transform, Virtualise
ESB

Orchestrate
BPEL/BPM process engine

Externalize business rules
Rules Engine

Involve human beings
Human Workflow

Secure
Global Security Framework

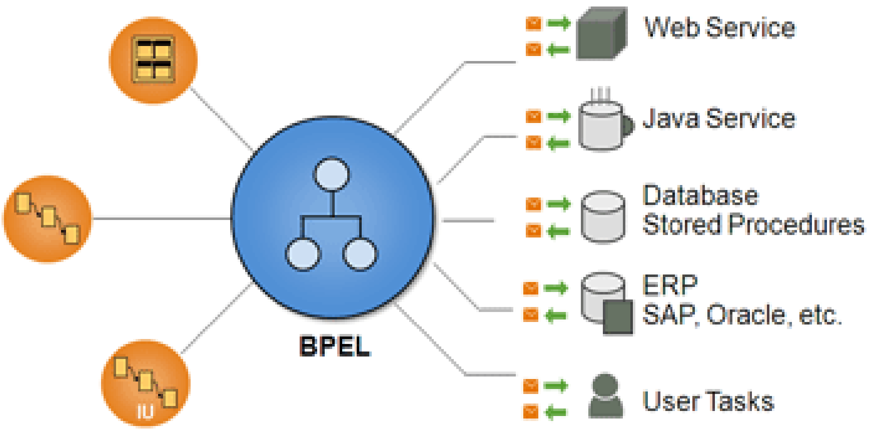
Visualize
Business Activity Monitoring

Manage & Govern
Enterprise Manager,
Repository & Registry

Service Orchestration

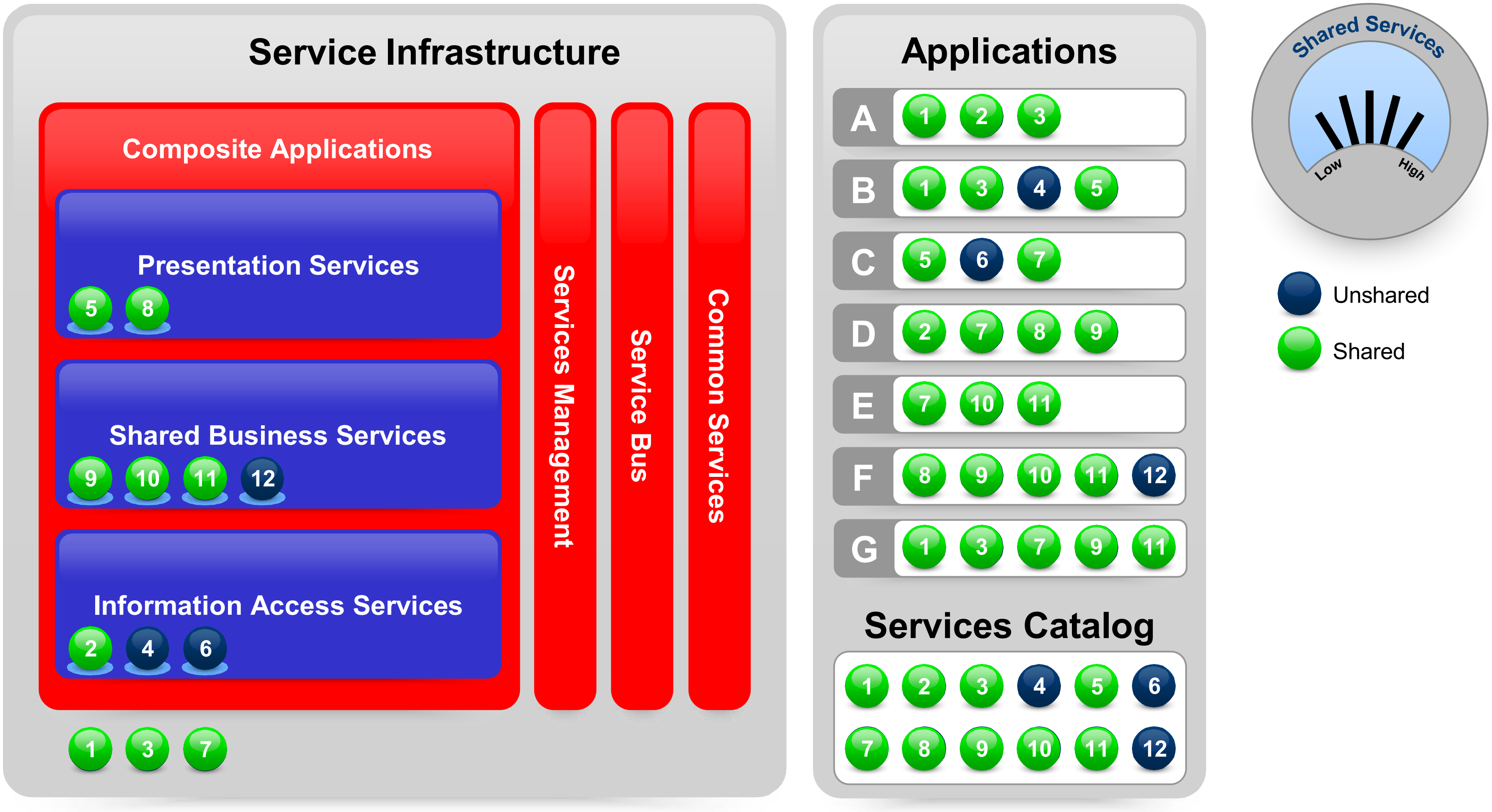
Building a service

- ***Declarative orchestration to rapidly derive new value from existing services & assets:***
  - Systems
  - People
  - Documents
- Separates the business process definition from the implementation
- Built-in asynchronous support
- Long running processes
- Standards based



Tracking assets and services…

Asset visibility



Service Infrastructure

Composite Applications

Presentation Services

5 8

Shared Business Services

9 10 11 12

Information Access Services

2 4 6

Services Management

Service Bus

Common Services

1 3 7

Applications

A	1	2	3		
B	1	3	4	5	
C	5	6	7		
D	2	7	8	9	
E	7	10	11		
F	8	9	10	11	12
G	1	3	7	9	11

Services Catalog

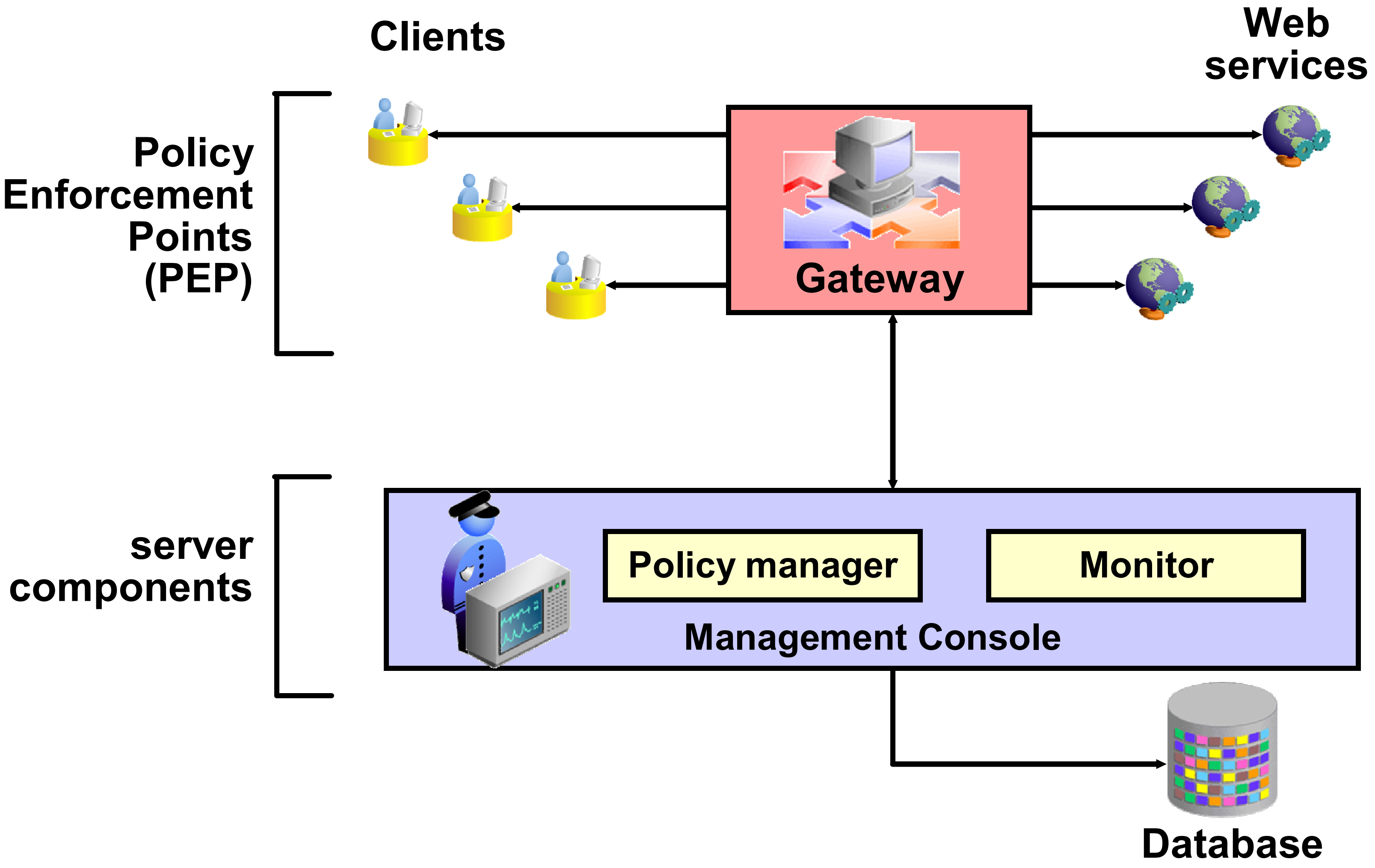
1	2	3	4	5	6
7	8	9	10	11	12

🔵 Unshared

🟢 Shared

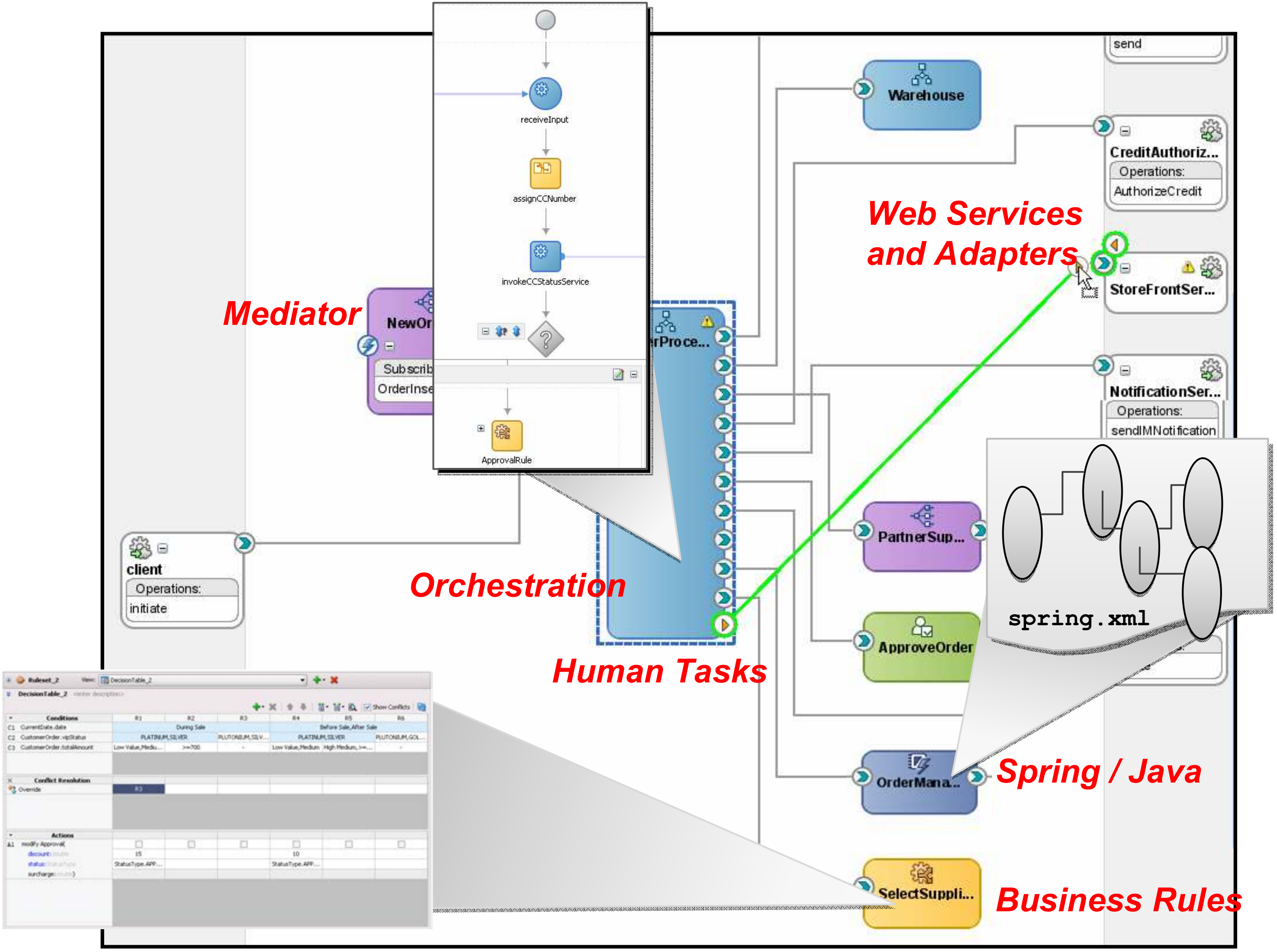
Service Oriented Security

Securing assets in a service network



SOA Composite Editor

How Oracle helps you pull it all together



Mediator

Orchestration

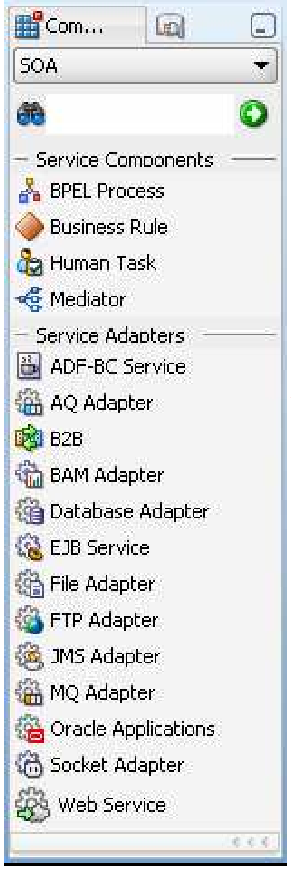
Human Tasks

Web Services and Adapters

Spring / Java

Business Rules

Palette of Service Components & Adaptors



- Service Components
  - BPEL Process
  - Business Rule
  - Human Task
  - Mediator
- Service Adapters
  - ADF-BC Service
  - AQ Adapter
  - B2B
  - BAM Adapter
  - Database Adapter
  - EJB Service
  - File Adapter
  - FTP Adapter
  - JMS Adapter
  - MQ Adapter
  - Oracle Applications
  - Socket Adapter
  - Web Service

Agenda

- What is SOA?
- Separating concerns
- What is a Service?
- Beyond Services
- Summary
- Q&A

Summary

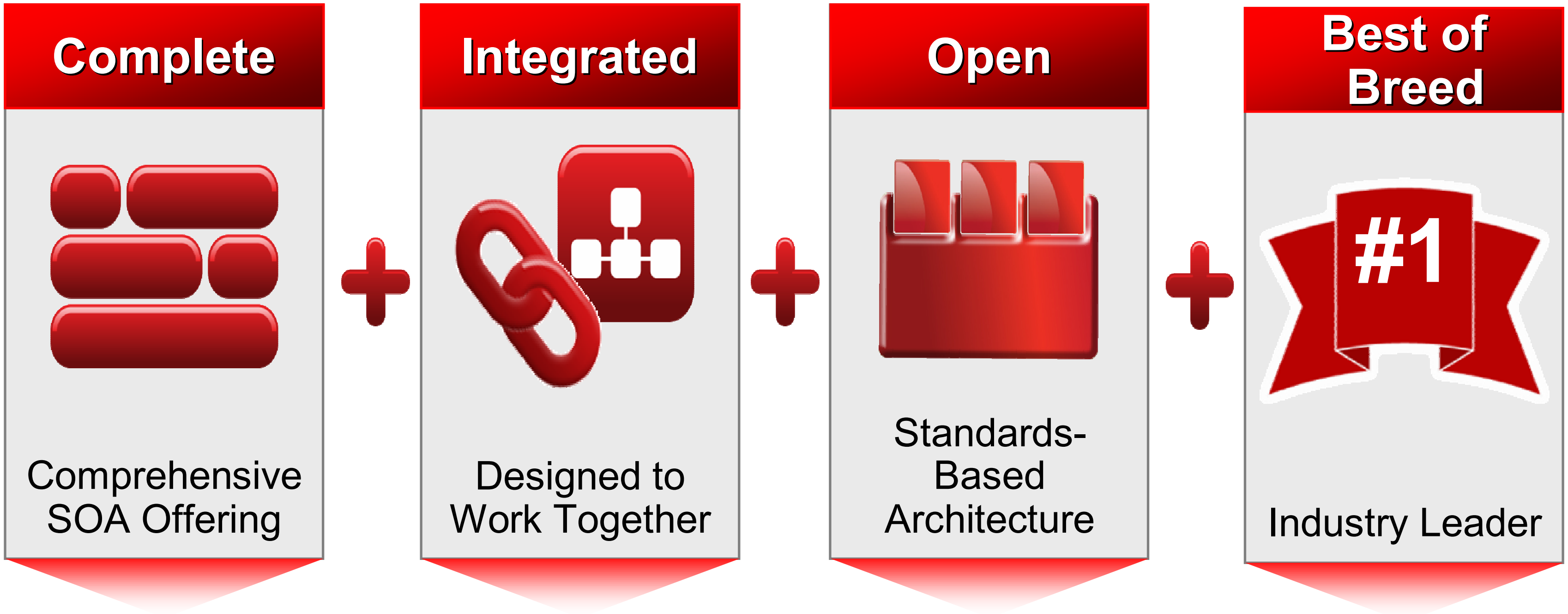
SOA Challenges

- Integration to business systems
- Building effectively for quicker, timely delivery
- Achieving performance levels
- Providing compliance and regulatory data
- Securing services in SOA
- Admin and Management of orchestrated apps

Proving ROI of design and runtime assets

Why Oracle?

SOA Suite 11g



Complete

Comprehensive SOA Offering

+

Integrated

Designed to Work Together

+

Open

Standards-Based Architecture

+

Best of Breed

Industry Leader

Broad & Deep Offering Lower Cost, Lower Risk

Less Effort More Value

More Choice Maximizes Existing Investments

Robust, Proven Industry leading Software

Agenda

- What is SOA?
- Separating concerns
- What is a Service?
- Beyond Services
- Summary
- Q&A

Questions

Useful links

- Mastering SOA Series - http://www.oracle.com/technology/tech/soa/mastering-soa-series/index.html
- Service Component Architecture - http://www.oracle.com/technology/architect/entarch/pdf/oracle_sca_the_power_of_the_composite.pdf
- Building Services with BPEL - http://www.oracle.com/technology/pub/articles/bpel_cookbook/index.html
- New SOA Standards - http://www.oracle.com/technology/tech/standards/soa-standards.html

Thank you