

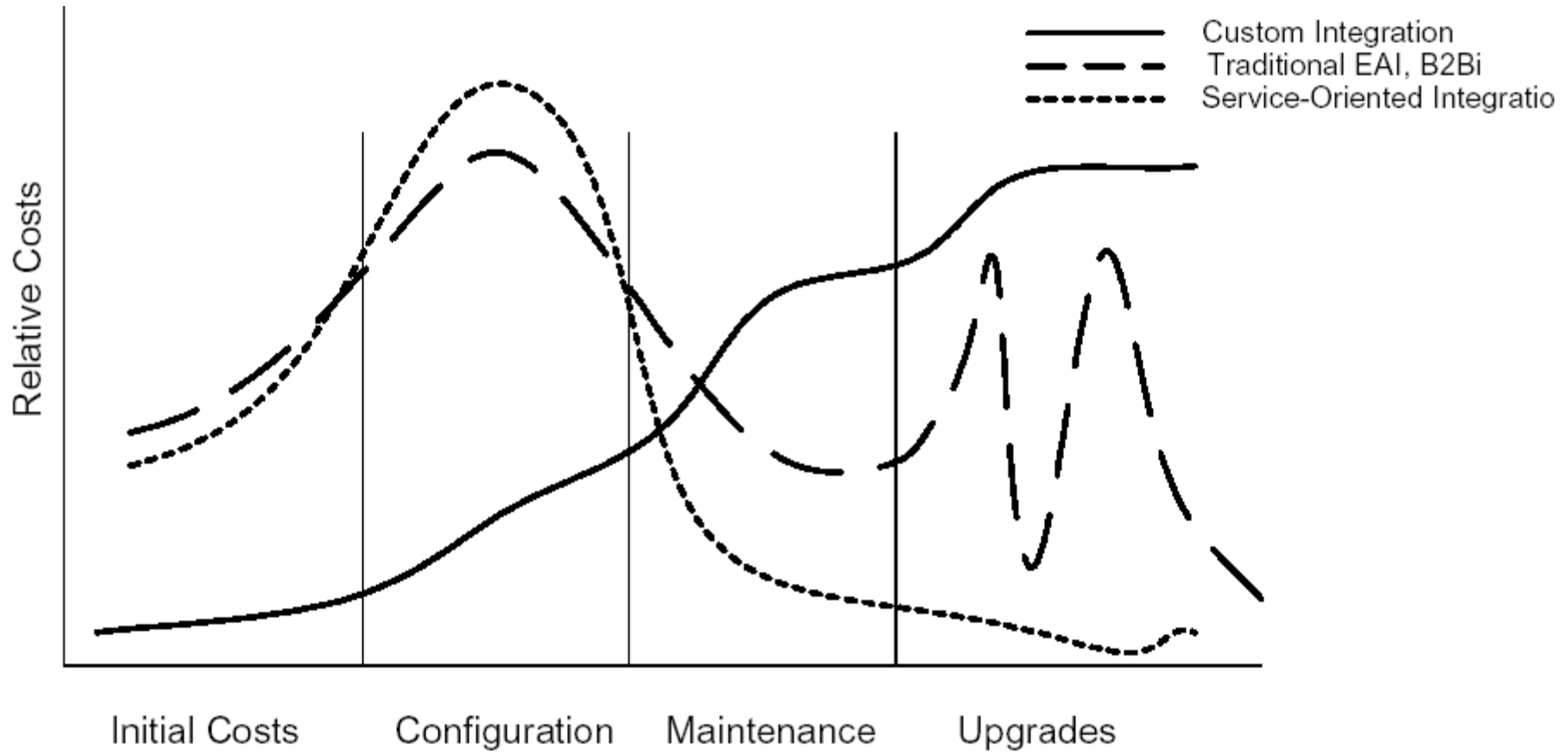


Architectural Requirements for an SOA Based on Web Services

**Jim Bole
VP, Engineering
Infravio, Inc.
April 23, 2003**

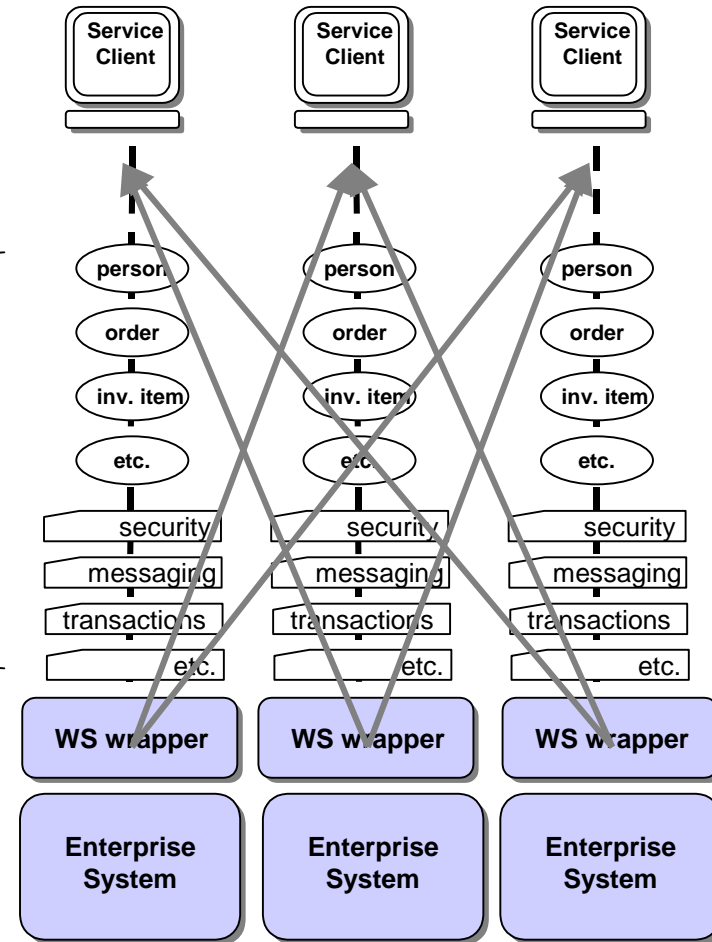
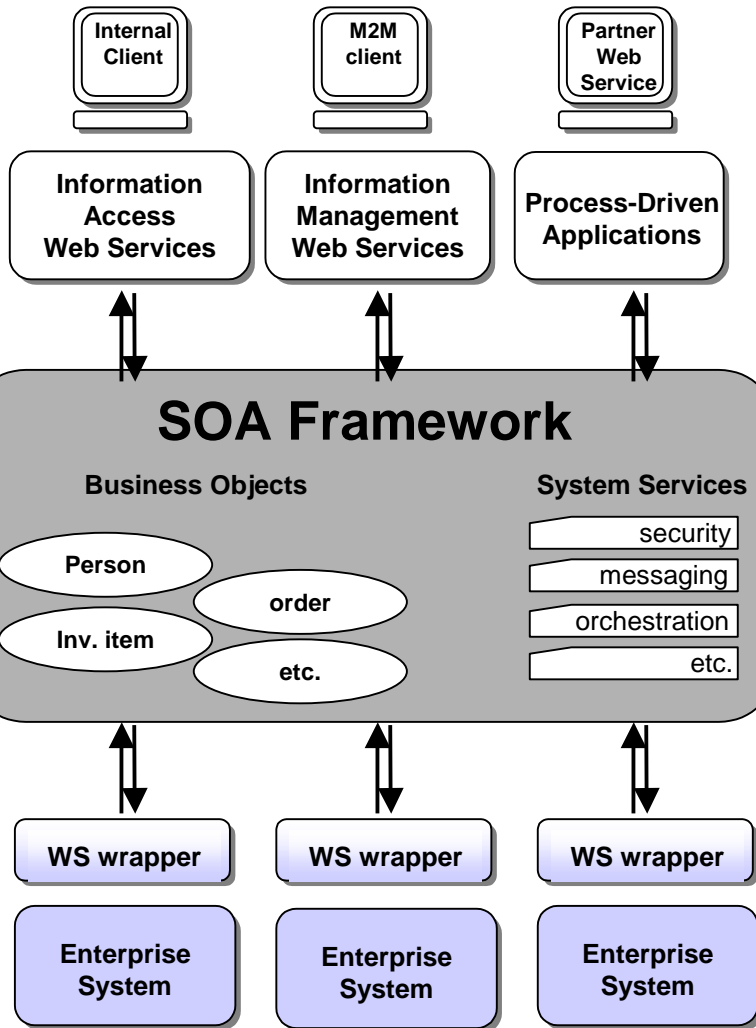
- Web Services is an integration technology
- Web Services' role in an SOA
- Unique Features and Challenges of Web Services
- Architectural Requirements of an SOA
 - ◆ Standards-based solutions
 - ◆ Software-based solutions
- Case Study
 - ◆ A Web Services-Based SOA at Providence Health System

SOA Reduces Integration Costs Over Time



Source: ZapThink

Web Services ≠ SOA



	Tightly Coupled	Loosely Coupled
Interaction	Synchronous	Asynchronous
Messaging Style	RPC	Document
Message Paths	Hard Coded	Routed
Technology Mix	Homogeneous	Heterogeneous
Data Types	Dependent	Independent
Syntactic Definition	By Convention	Published Schema
Bindings	Fixed and Early	Delayed
Semantic Adaptation	By Re-Coding	Via Transformation
Software Objective	Re-Use, Efficiency	Broad Applicability
Consequences	Anticipated	Unexpected

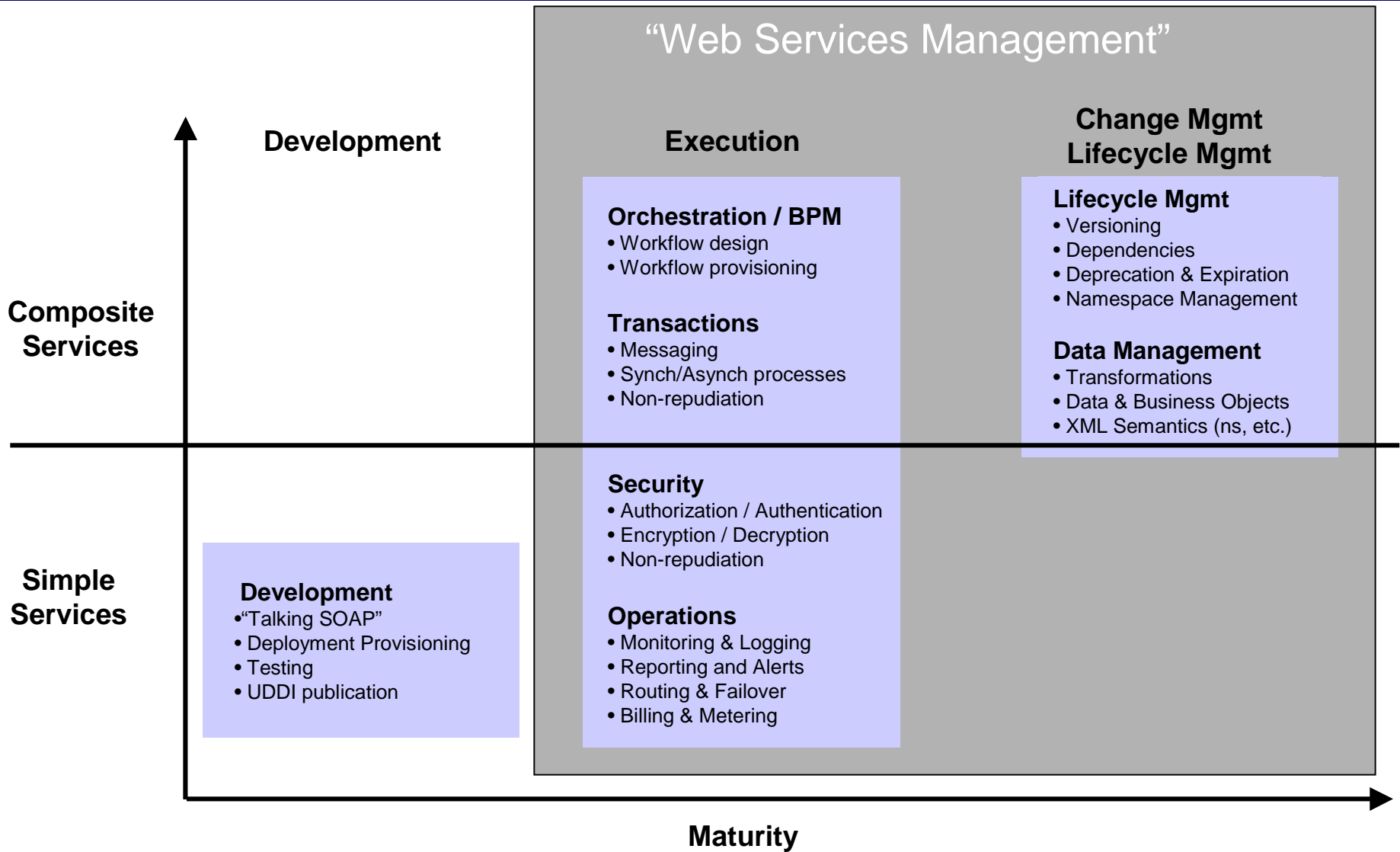
Source: *Loosely Coupled*, Doug Kaye

- Change Management
 - ◆ No method of versioning services
 - ◆ No way to track dependencies between components
 - ◆ No way to ensure backward compatibility when services upgraded

- Interoperability
 - ◆ Not just SOAP, WSDL, UDDI, etc.
 - ◆ Data formats / semantics

- Lack of Reuse
 - ◆ No central point of control
 - ◆ No way to track multiple versions

Requirements for an SOA





Standards for Web Services Management

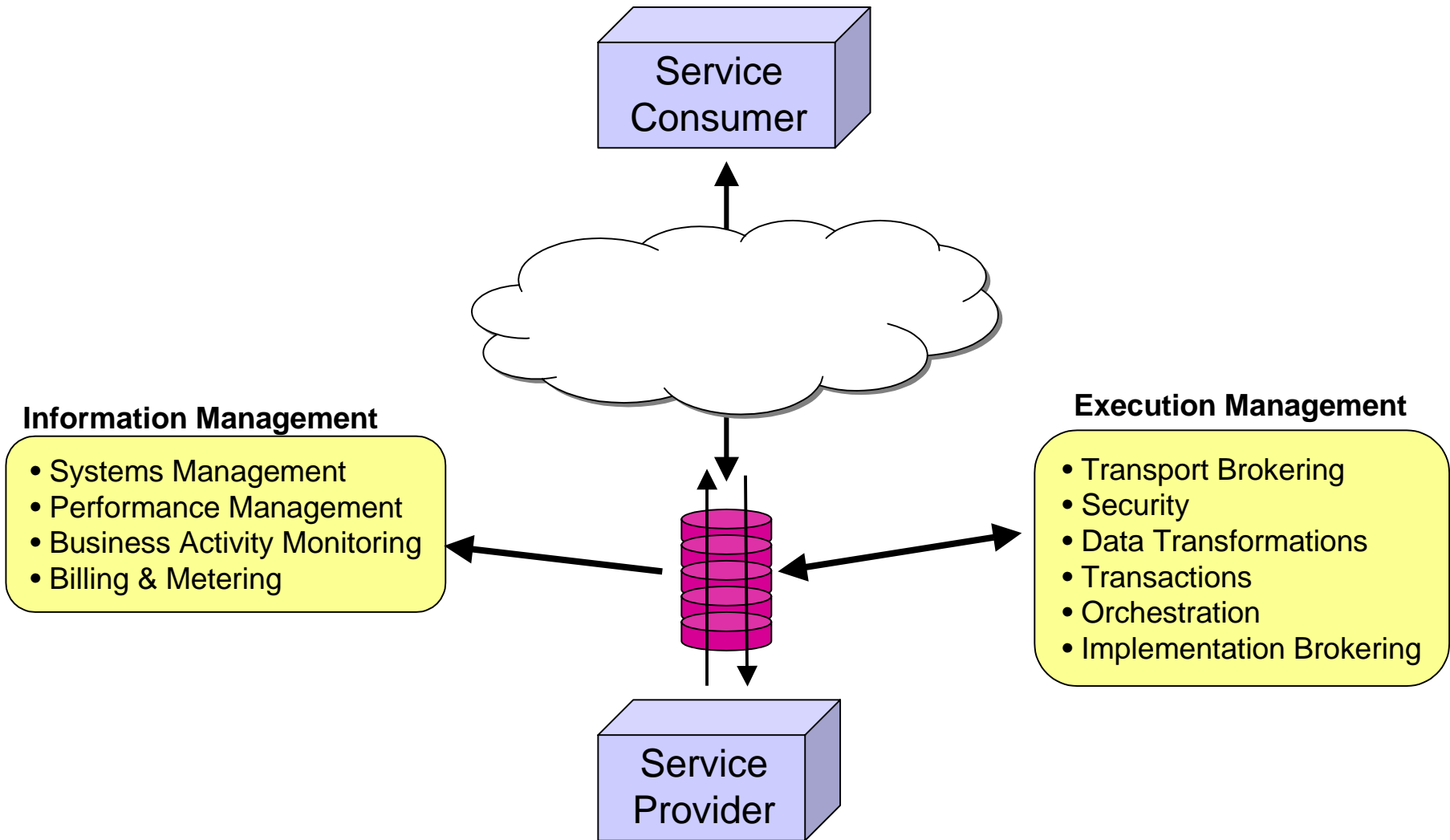
- Security
 - ◆ Authentication & Authorization
 - ◆ Encryption
- Performance management
 - ◆ Logging and monitoring
 - ◆ Reporting and alerts
 - ◆ Routing and failover
 - ◆ Deployment and provisioning
- Transactionality
 - ◆ Reliable messaging
 - ◆ Long running processes
- Change management
 - ◆ Versioning, lifecycle, expiration
 - ◆ Records and semantics

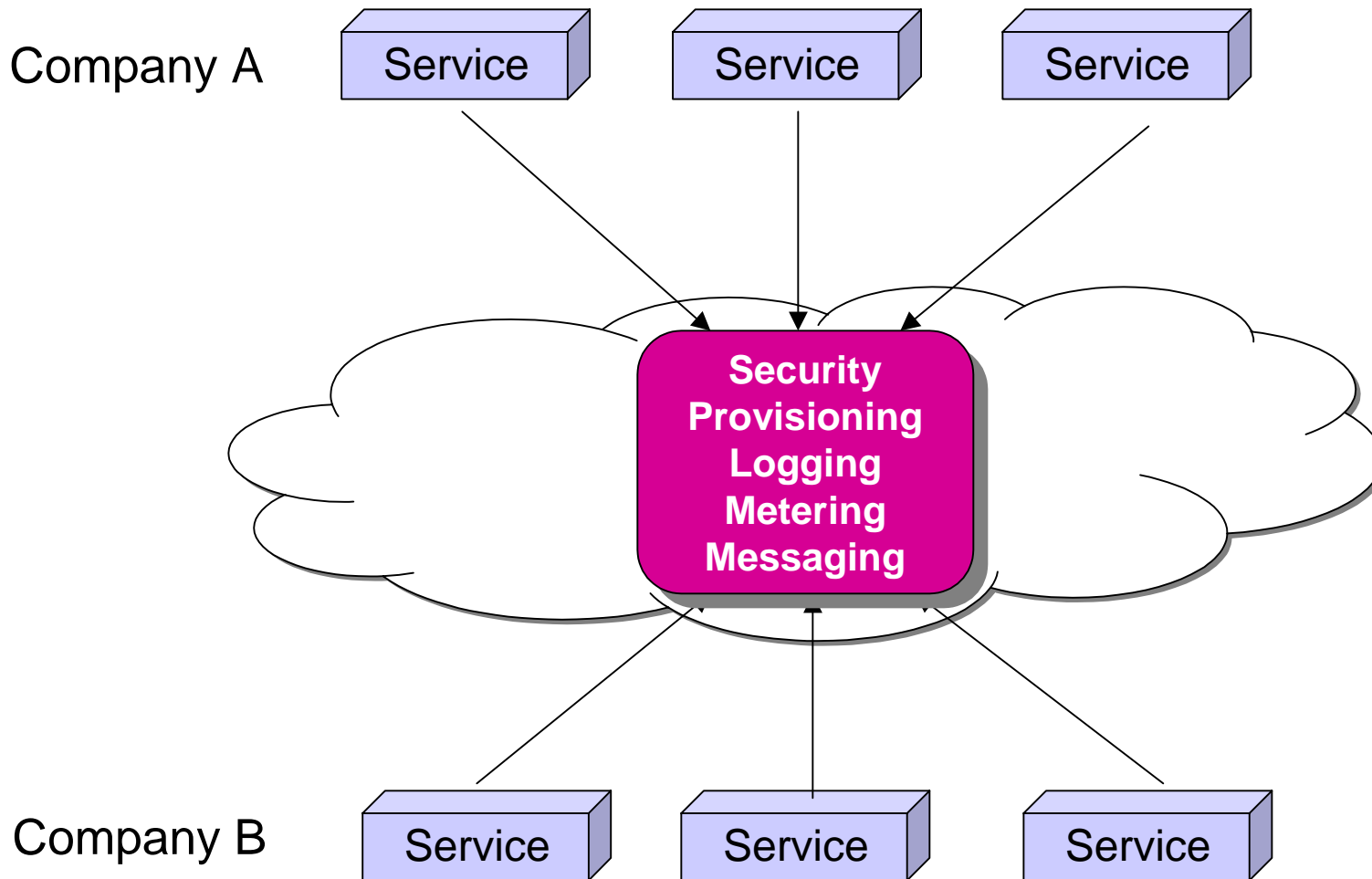
LDAP, SAML, XACML, WS-Security
XML Enc, SOAP/XML-DSig
NONE
OMI, SNMP
WS-Routing, WS-Referral
NONE
JMS, WS-R
WS-Transaction, BPEL, WSCI, ebBPSS
WS-Change, UDDI, WS-Inspection
XSLT, Namespaces

- Evolve when
 - ◆ There is a need for interoperability
 - ◆ Requirements are fully understood
 - ◆ Dominant players support the standard
- Fail when
 - ◆ The need is not yet well understood
 - ◆ Dominant players are threatened by the standard
- For Infrastructure
 - ◆ Are easier to develop and implement by vendor agreement
 - ◆ Are more technology focused – point solutions
 - ◆ Have (s)lower impact on the business
- For Verticals
 - ◆ Need large business driver to justify customer involvement
 - ◆ Develop slowly over many iterations
 - ◆ Generate massive ROI's if widespread adoption is achieved

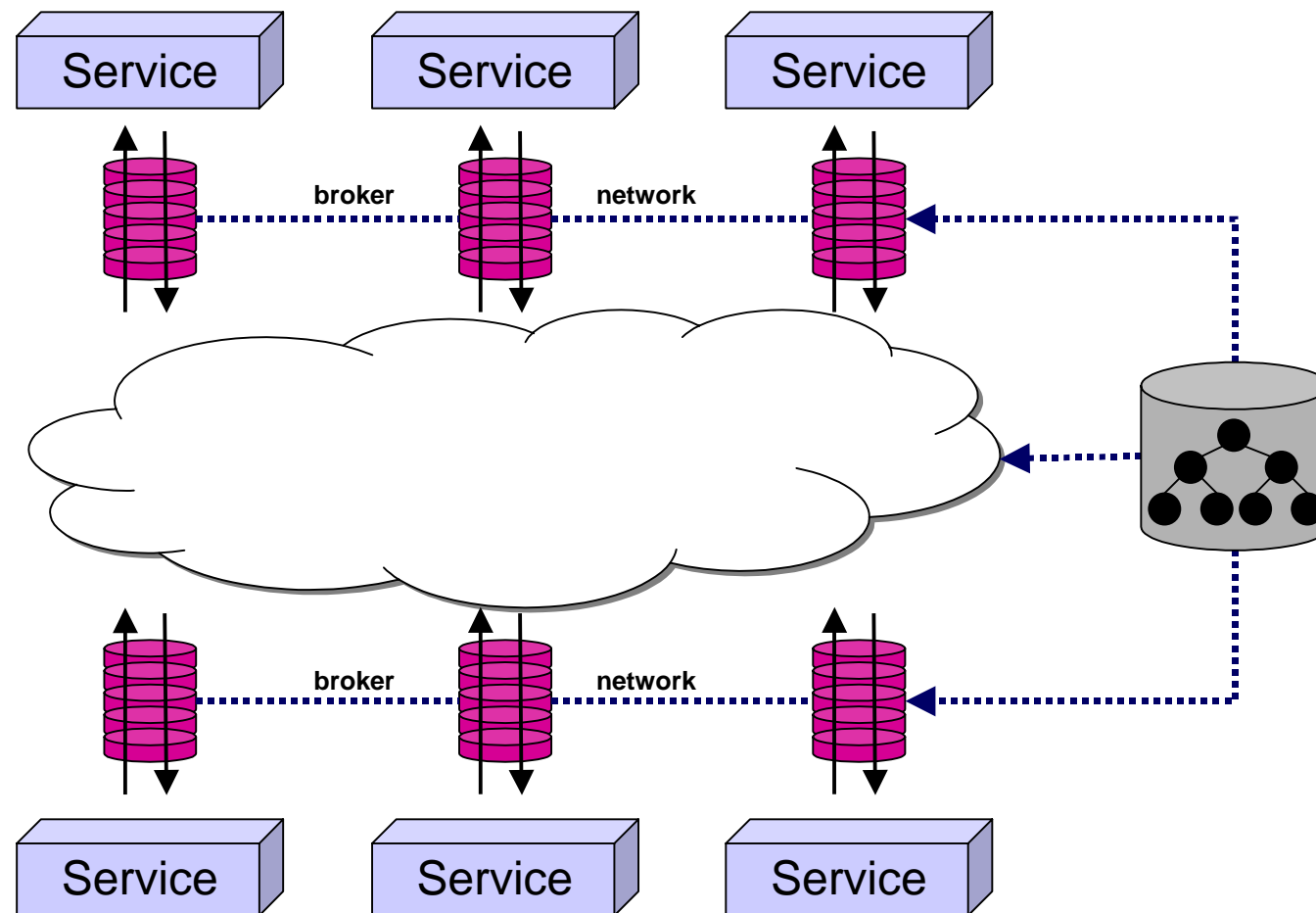


Software Solutions





Broker Network with Central Management



- Systems management
- Semantics
 - Data Mappings
 - Business objects
 - Namespaces
- System Services
 - Defined providers
- Change management
 - Assigns versions
 - Maps dependencies



Case Study

Objective: Managing a Web Services-based SOA



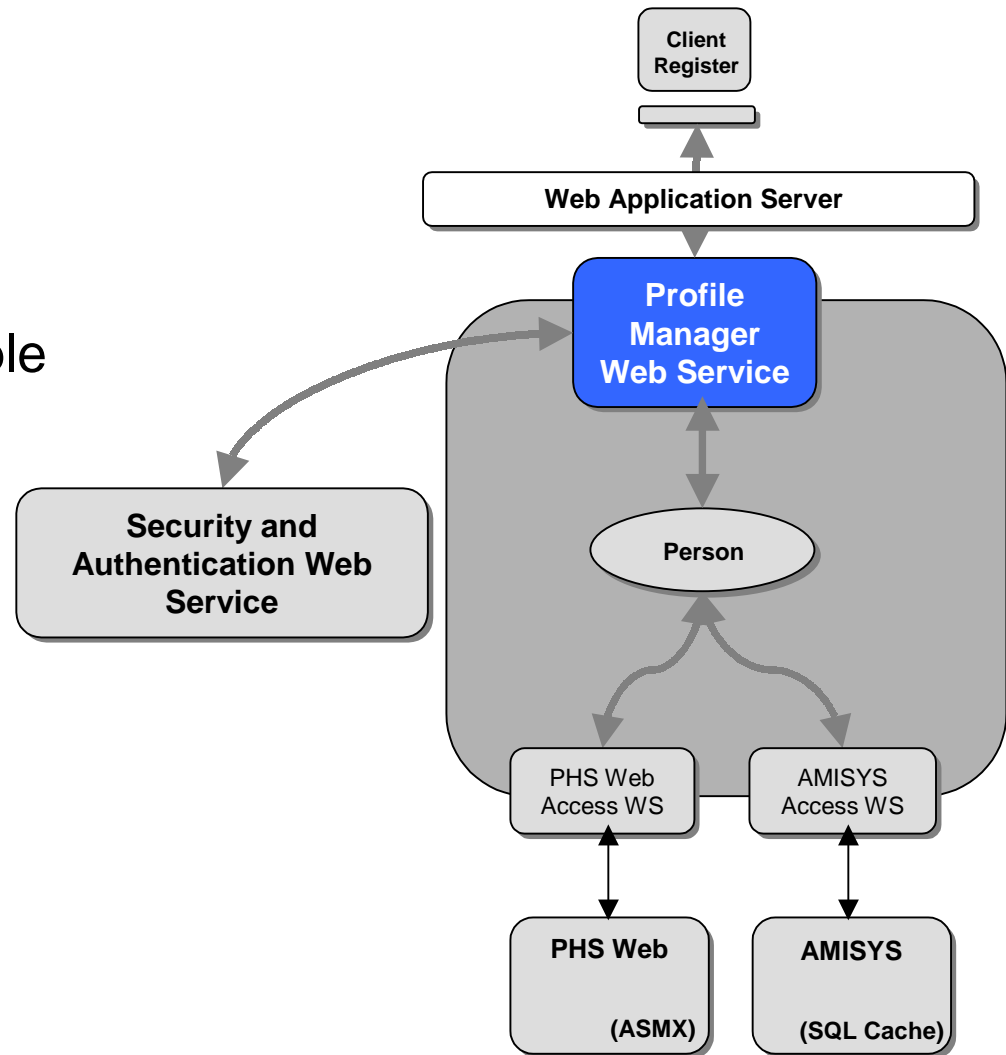
Providence | Health System

- Problem
 - ◆ Integrate Customer Data from Multiple Systems
 - ◆ Move to a Service Oriented Architecture

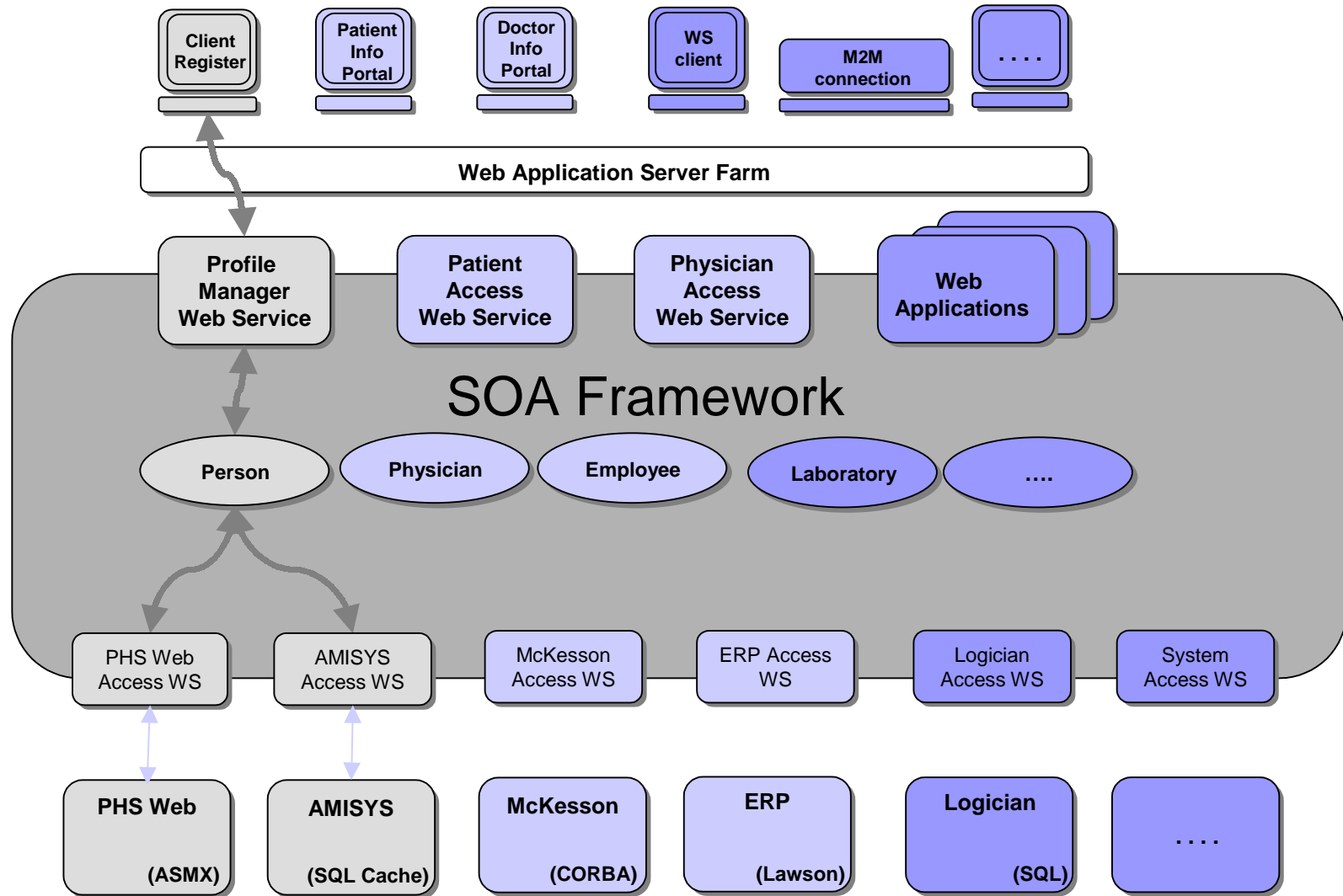
- Established a Point of Comparison
 - ◆ Data Warehouse
 - Never completed

 - ◆ Existing EAI Solution
 - High Cost – Maintenance of multiple point-to-point integrations
 - Rigid – Could not keep pace with change
 - Proprietary – Required significant developer knowledge, services
 - Lack of reuse – Not an open SOA platform

- Integrated Web Application with back-end systems
- Collected and shared customer data across multiple systems
- Created “Profile Manager” Service
- Established Web Services management layer
 - ◆ Security
 - ◆ Logging
 - ◆ Versioning



Expanding the SOA at Providence



Infravio Web Services Resource Center

www.infravio.com