Supercharging SOA Registries With XML Persistence and Management

Ash Parikh
Director of Technology

Ajay Ramachandran
CTO

Premal Parikh
Lead Architect

XAP, Raining Data Corporation

TS-8098

Supercharge SOA Registry-Repository

Leverage the Power of XML DB and XQuery

Learn How to Supercharge Your SOA Registry-Repository for Improved Governance and Performance and Integration

Benefits: Supercharging SOA Registry-Repository With XML and XQuery

- Improved Governance
  - Flexibility = agility
  - XQuery = most effective metadata governance tool
  - Need for sophisticated metadata and content search
- Improved Performance
  - Data repurposing service = improved performance
  - Scalability = XML-based caching
  - Federated services
- Improved Integration
  - Composite data = composite apps
  - Data and query services = re-usable data integration
  - Right Information—data, events, processes, semantics, models, analytics—in the Right Format, at the Right Time, in the Right Context, with the Right Semantics, and with the Right Security

Agenda

Definitions

Problems

Solution

Platforms

Demo

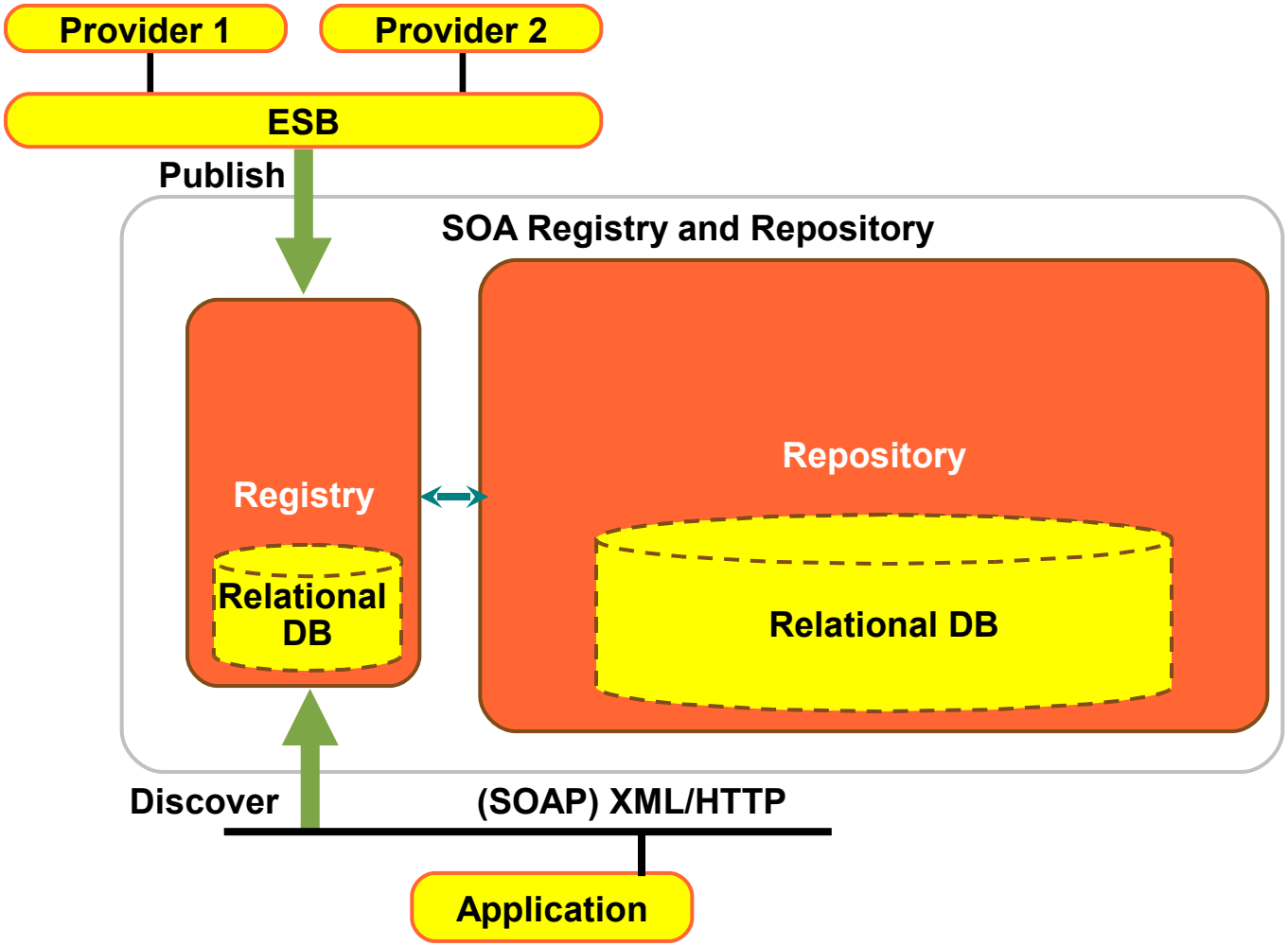
Q&A

Definitions

SOA Registry-Repository Definition

- SOA Registry-Repository is like a local library
  - Repository is like book shelves—stores content
  - Registry is like card catalog—describes content
  - Registry and repository are administered jointly
  - Registry and repository provide a unified service [SD Times]
- Registry stores only links to artifacts
- Repository stores actual artifacts

Existing SOA Registry-Repository Architecture



SOA Governance—Definitions

- Governance is the process of defining and enforcing organizational policies and standards

"In 2006, lack of working governance mechanisms in midsize-to-large (greater than 50 services) post-pilot SOA projects will be the most common reason for project failure"

Gartner

"Many organizations don't start to think about governance until things are completely out of control"

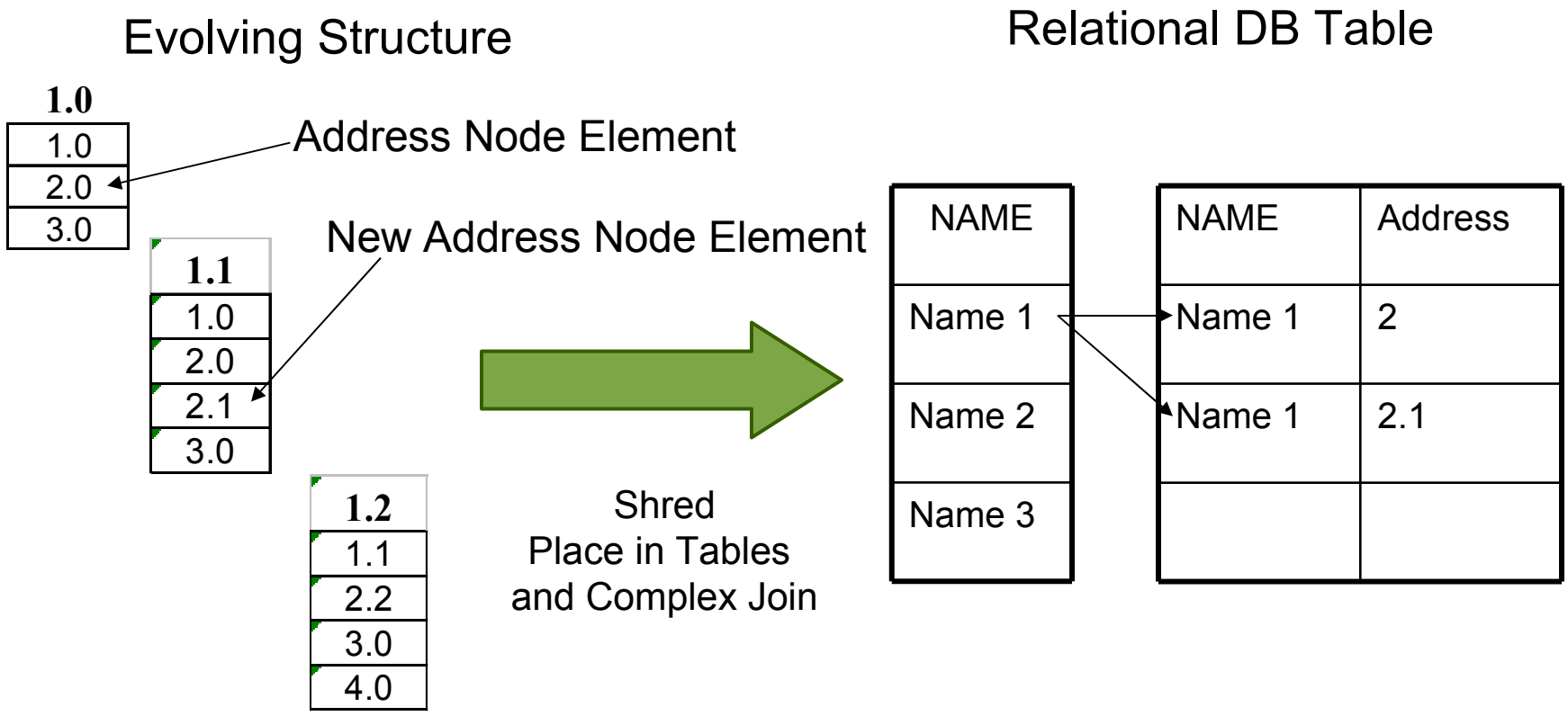
Anne Thomas Manes, Burton Group

Problems

Current SOA Registry-Repository Problems

- Relational databases are inflexible
  - Rigid, brittle data models
  - No support for evolving schemas for dynamic SOA
- Large quantities of disparate metadata require sophisticated search capabilities
- SOA metadata must adhere to XML-based standards
- Performance bottlenecks for information access exist

Relational Database Structures are Rigid and Brittle and Have Limited Support for Evolving Schemas



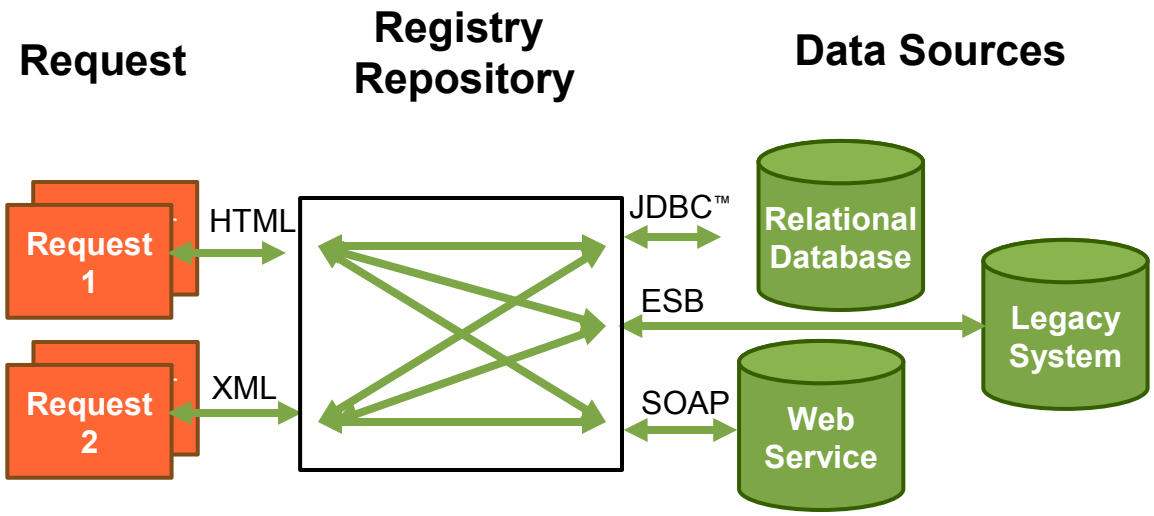
* Always Rebuilding Table to Handle New Schema. For example, schema 1.0 has one address element whereas schema 1.1 has permanent and secondary address elements.

Large Quantities of Disparate Metadata Require Sophisticated Search Capabilities

- SOA metadata is diverse
  - Information about business processes
  - Rules used by business users
  - Information about system, data stores, services
  - Relationships and dependencies
- SOA metadata exists in large quantities
- SOA metadata is spread across enterprises
- Simple search of large quantities not enough
- Fast, meaningful metadata search is complex

So-Uhhhh, If Everything's XML, Choose the Right Wheels

Performance Bottlenecks for Information Access Exist

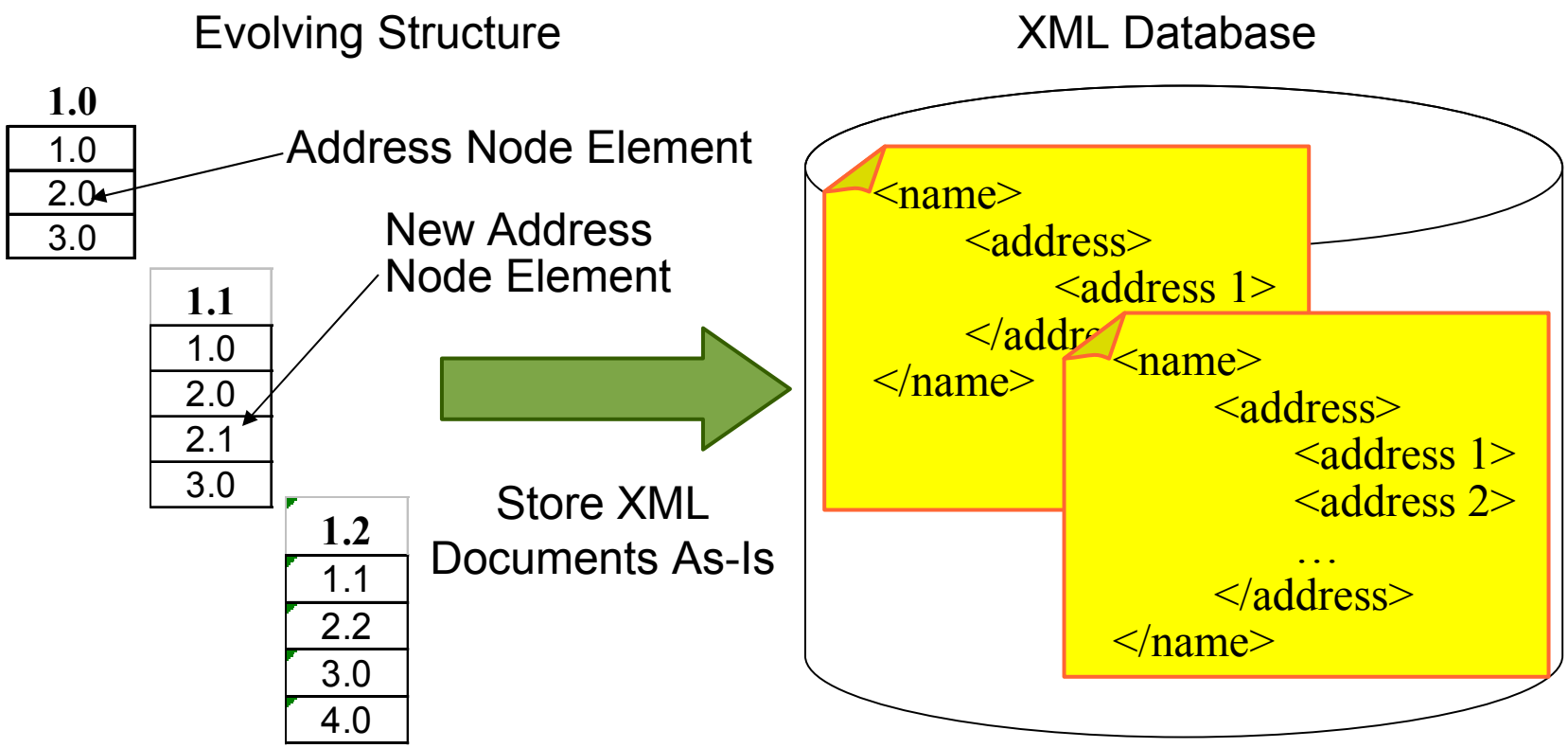


- Need frequent access of information store
- Long running business process orchestrations need JIT information
- Performance decreases as requests increase
- Simply storing metadata is not enough
- Traditional queries not enough for diverse XML

The Current SOA Landscape

Solutions

Simple XML Schema Mapping and Evolution



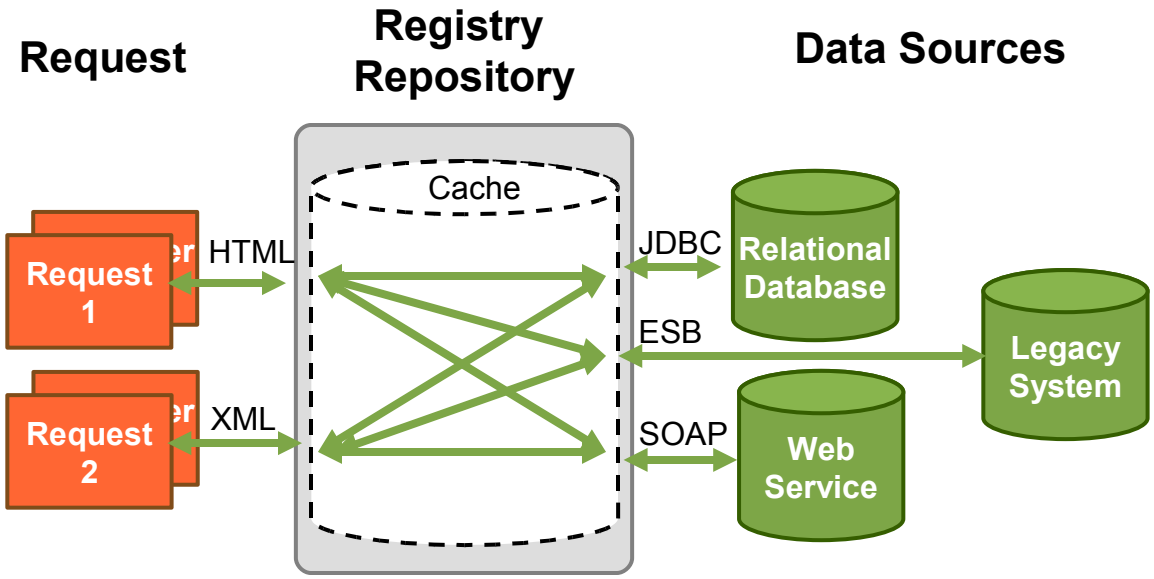
* XML Elegantly Handles Changes to the Customer Information Data Structure. For example, Permanent and Secondary Addresses are Both Under the Element Address

Large Quantities of Disparate Metadata Demands XML and XQuery

- SOA governance focuses on SOA metadata
- Majority of this metadata are XML documents
- XQuery—Natural querying technology for XML
- XQuery—Fast, effective SOA governance
- XQuery must be used for more than querying
  - Enabling flexibility by writing less code and eliminating hard-coding
  - Creating meaningful, re-usable metadata caches
  - Effecting custom policy-based caching
  - Federation of SOA information sources
  - Linking of SOA metadata
  - Auditing of SOA metadata
  - Supporting dynamic/just-in-time XML transformations
  - Re-purposing results for powerful business intelligence

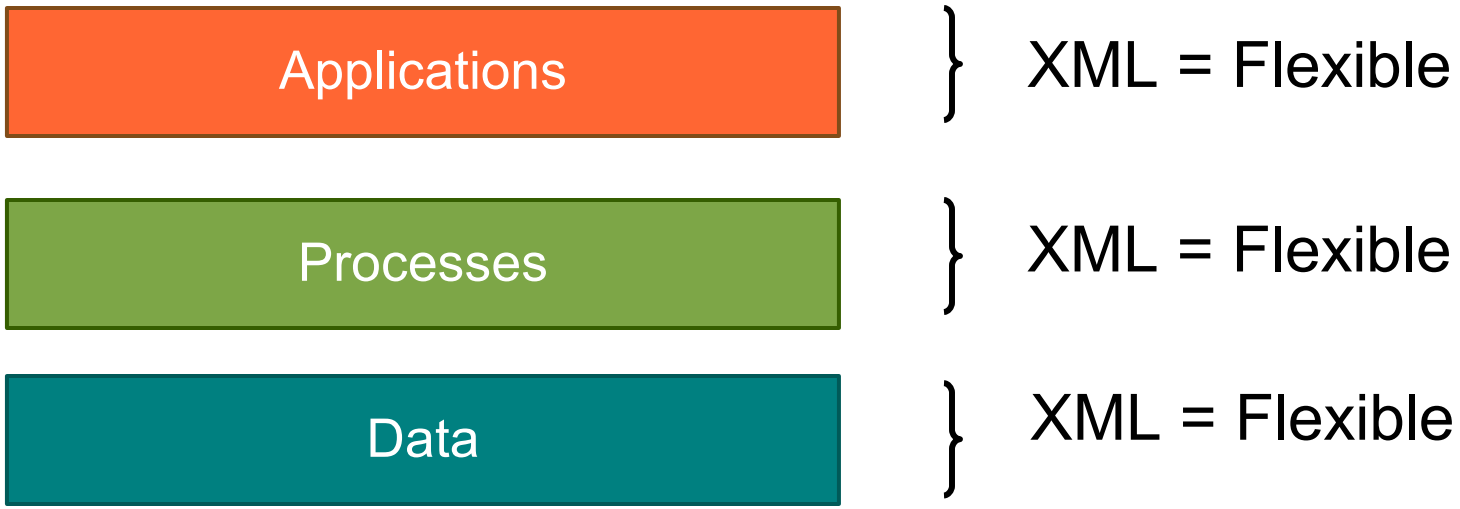
The Right Wheels: XML DB and XQuery

Alleviate Performance Bottlenecks for Information Access With Caching



- Frequent access of information store via policy-based caches
- Time-to-live caches for long running business processes
- Performance increases as requests increase
- XQuery is well suited for querying diverse XML

The Improved SOA Landscape



The Solution—XML DB and XQuery Power SOA Registry-Repository

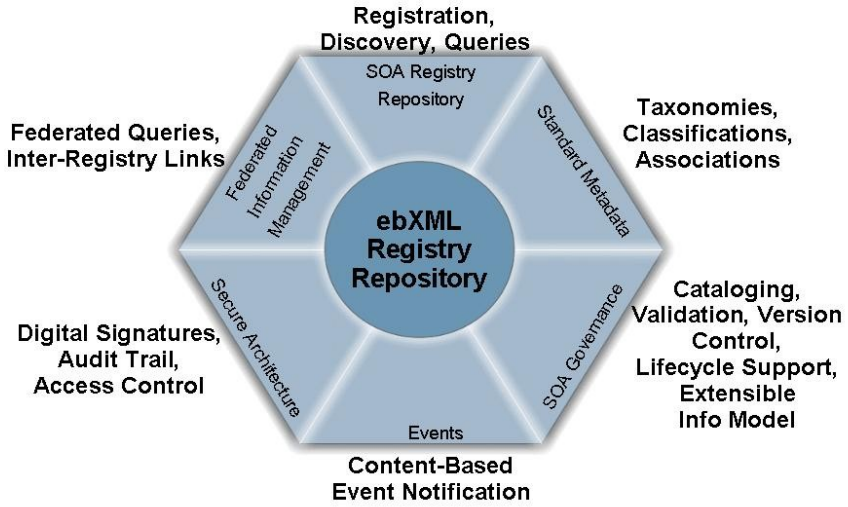
- XQuery enables **Sophisticated Search** for metadata discovery
- XQuery-enabled metadata and transactional payload persistence Enables a **Policy-Based Caching Service**
- XQuery-based persistence enables a **Data Repurposing Service** to improve performance
- XQuery optimizes metadata XQuery-based metadata persistence enables a SOA Registry-Repository to scale by supporting the **Federation of Services** within a SOA implementation
- XQuery optimizes metadata management by **Improving Agility and Efficiency** of SOA administration and maintenance across the enterprise
- XQuery improves SOA governance and policy management by **Enforcing Policies** on SOA transactions and payloads
- XQuery enables high-performance composite applications with **Real-time, Flexible Composite Data** by exposing heterogeneous legacy data as consumable data services

Platforms

Standards/Implementations—ebXML Registry Repository, Sun Service Registry, UDDI

ebXML Registry Repository:

OASIS



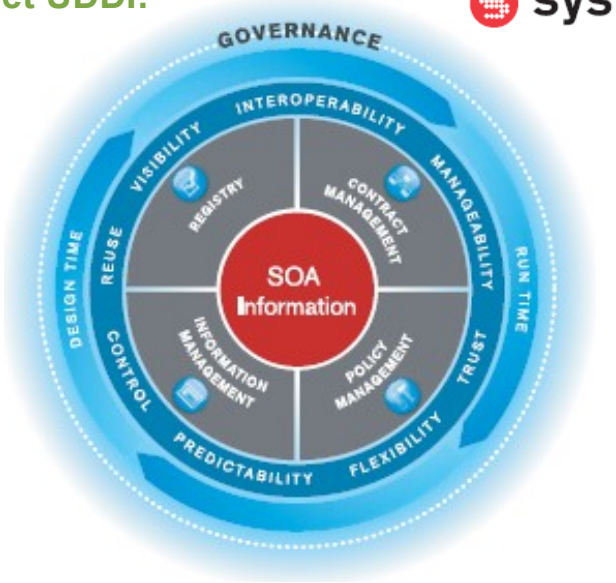
Sun Service Registry:

Sun microsystems

- Integrated Registry Repository Service
- Secure federated information management for artifacts:
  - XML, schema, WSDL, BPEL, policy…
- Enables SOA governance for artifacts
- Built on XML, Java technology standards—ebXML registry, UDDI, JAXR…
- Integrated with and component of Sun Java ES
- Built on open source freebXML Registry

Systinet UDDI:

systinet

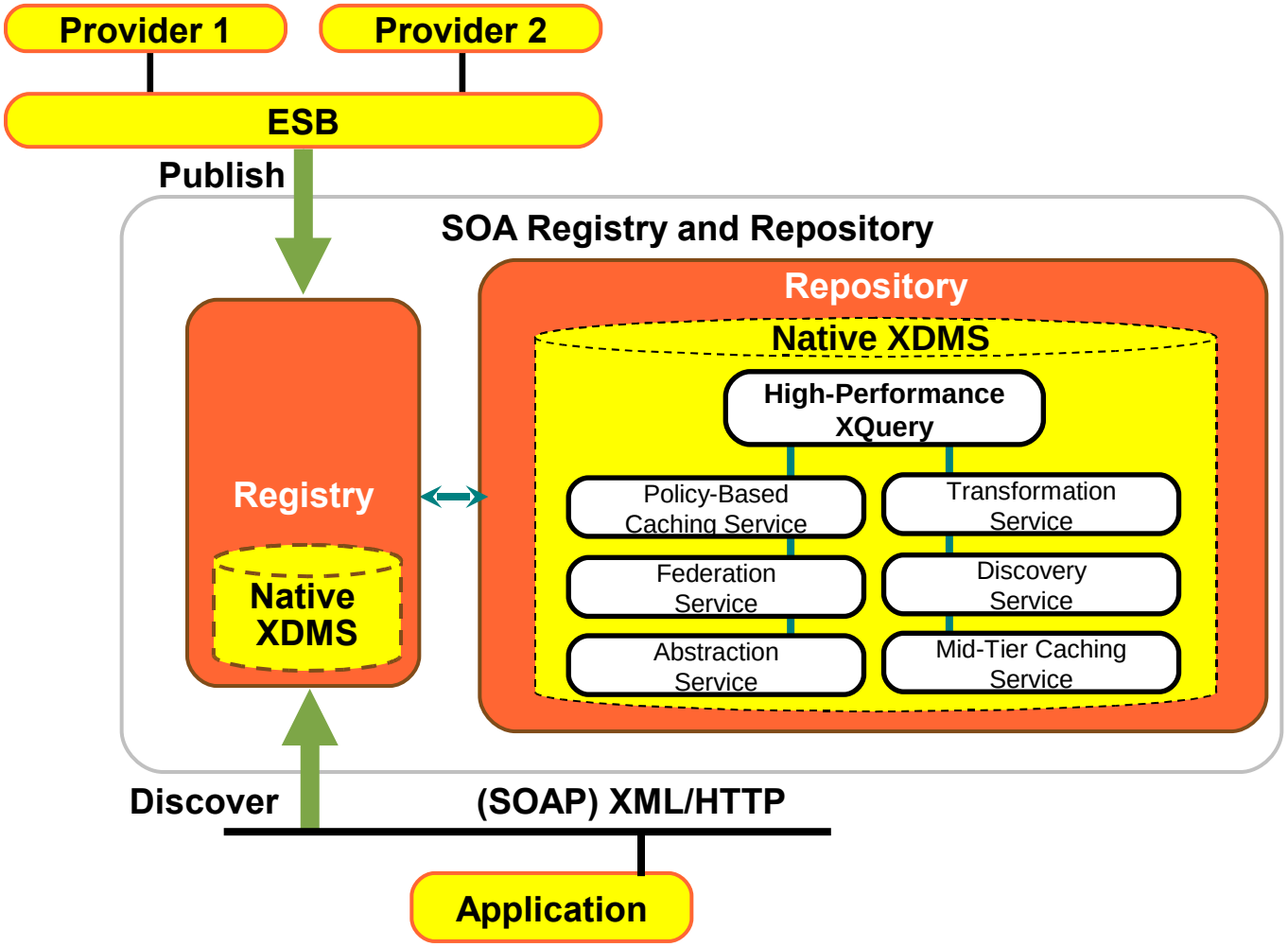


- Publish and discover services
- Design-time validation of services
- Create and enforce policies
- Facilitate contracts and consumption agreements
- Seamlessly implement changes to business services

Supercharge ebXML Registry-Repository, Sun Service Registry and UDDI With XML DB and XQuery

Critical Registry Repository Functionality	Benefit of XML DB and XQuery
Cataloging	**Improved Performance and Governance** XQuery Cataloging Policies Increase Discovery Efficiency and Self-Evolving XML Data Model Increases Manageability
Validation	**Improved Governance** XQuery Rules Ensure Conformance to XML Organizational Policies by Semantic Publish-Time Validation
Federated Services	**Improved Performance** XQuery Data Repurposing Services Enable XML-Based Single Source of Truth by Sharing, Linking of Large and Diverse Data Sources
Queries	**Improved Performance and Governance** Sophisticated Search by Flexible, Expressive, Powerful Ad-Hoc XQueries with Complex Predicates, Parameteric Syntax with Query Syntax Governance
*** Data Services	**Improved Integration** Reusable XQuery Data Services Enable Creation of Composite Data that is Exposed by the SOA Registry-Repository for the Building of Composite Applications
Caching	**Improved Performance** Native XML Caching Avoids Object/Relational Transformations; Policy-Based Caching Services Enable the Creation, Execution and Life-Cycle Management with Time-to-Live Metrics and Reliability of XML Caches

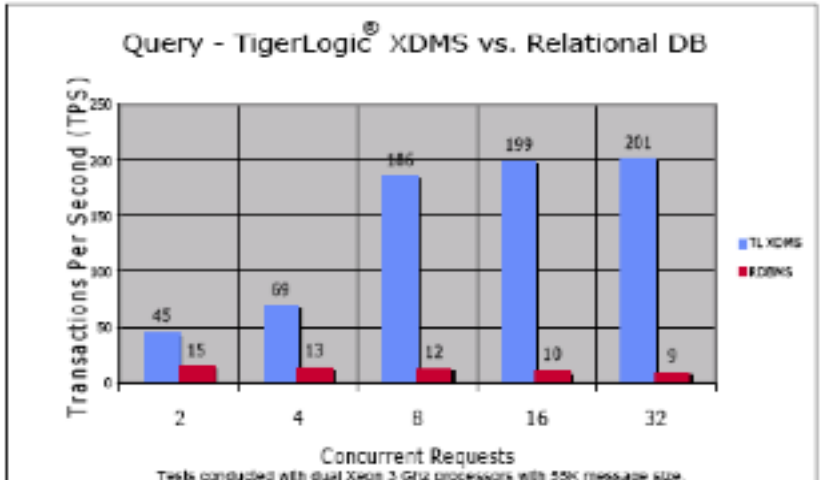
Supercharging SOA Registry-Repository Architecture

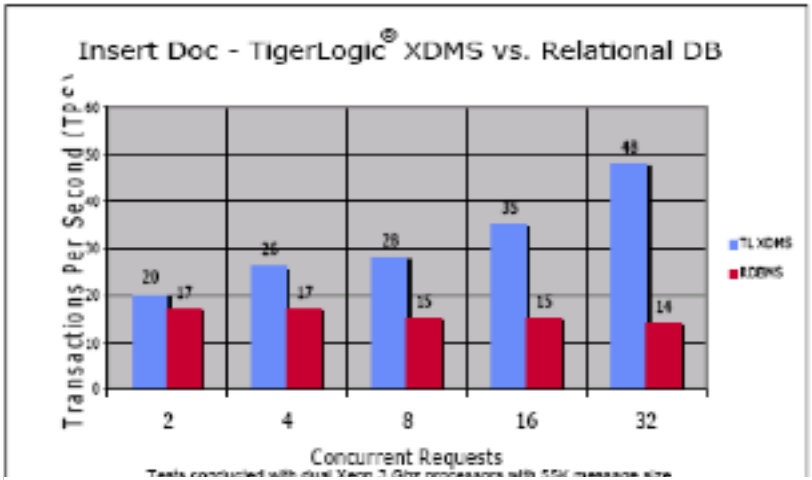


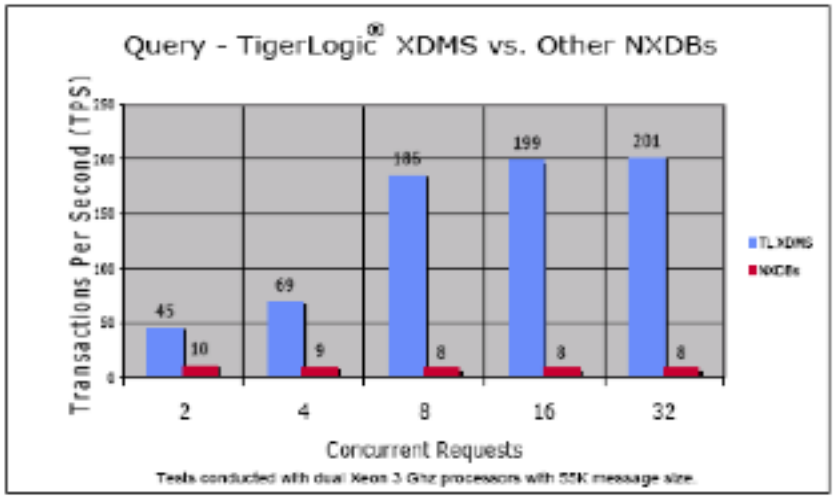
How TigerLogic® XDMS Supercharges Sun Service Registry

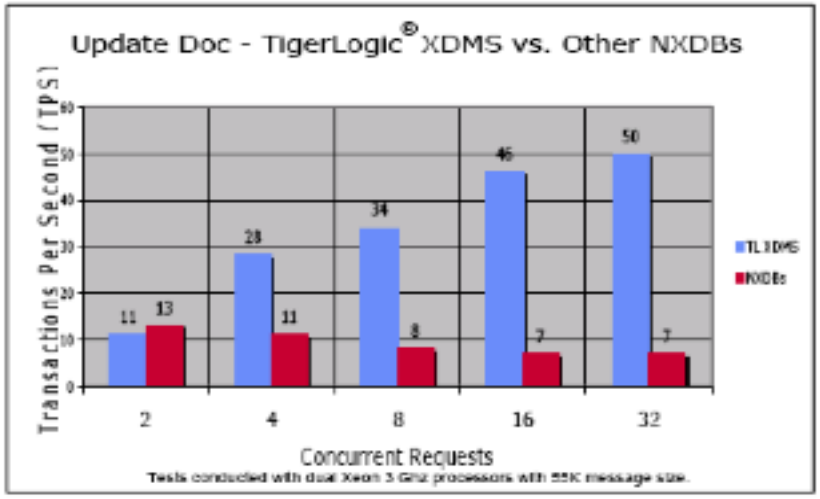
- Plug-In for improved repository features
  - TigerLogic XDMS as repository for Sun Service Registry
- Compelling benefits
  - Advanced SOA governance
  - Simple to flag violations
  - Easy to enforce policies
  - Scalable caching
- Value-added features
  - Query XML content in repository using XQuery
  - Integrate heterogeneous data sources
  - Create XML views of disparate information for BI
- **Wickedly fast handling SOA artifacts!!!**

TigerLogic Performance— Up to 22X Faster









Performance and Scalability Test*

- Three use cases fully implemented
  - Enterprise integration scenario (multiple schemas, large payloads)
  - Streaming information for portal application (simple schemas, high volumes)
  - ebXML enterprise purchasing scenario (complex schemas, large messages)
- Service interfaces tested on TigerLogic 2.6, BEA WebLogic Server 9
- Persistence testing on TigerLogic, Oracle, eXist, other Native XML DBs
- XML parsers: Java™ architecture for XML Binding, XMLBeans, Xerces
- Schemas: ebXML UBL BODS, streaming news, purchase order
- SOAP stacks: Sun Java Web Services Developer Pack 1.5, JAX-RPC, Apache Axis
- Test implemented 100% Java technology-based application, runs anywhere Java technology runs
  - Packaged to run in customer environment
- Equipment: dual 3 Ghz Xeon processors over Gigibit ethernet

* Performance and scalability test kit is based on over 5 years of experience at clients like BEA, Sun and GM. It is a solution that Raining Data offers to its enterprise clients.

CODE & DEMO

Raining Data TigerLogic XDMS

Summary

Supercharging SOA Registry

Improved governance

- Flexibility = agility
- XQuery = most effective metadata governance tool
- Need for sophisticated metadata search

TigerLogic XDMS

Improved governance

- Self-evolving XML data model for cataloging
- TigerLogic XQuery makes it simple to flag violations and easy to enforce policies
- Extended XQuery functionality

Summary (Cont.)

Supercharging SOA Registry

Improved performance

- Data repurposing service = improved performance
- Scalability = XML-based caching
- Federated services

TigerLogic XDMS

Improved performance

- Up to 22x faster than RDBMS, NXD
- Up to 3x faster with XML-centric services interfaces (transformations into Java objects and Java XML is costly)
- XQuery policy-based caching services, clustering and replication for HA
- XQuery federation services for extended connectivity to diverse data sources (JDBC, HTTP, WSDL, FTP, JMS, file system) with common query tool

Summary (Cont.)

Supercharging SOA Registry

Improved integration

- Composite data = composite apps
- Data and query services = re-usable data integration
- Right Information—data, events, processes, semantics, models, analytics—in the Right Format, at the Right Time, in the Right Context, with the Right Semantics, and Right Security

TigerLogic XDMS

Improved integration

- 360˚ XML views of heterogeneous data
- Reusable data via XQuery registry
- Valuable information at the right time, actionable information and intelligence, event driven architecture

"Raining Data's TigerLogic XDMS plug-in for Sun Service Registry adds new value to the Sun Service Registry product by enabling direct and efficient Search of XML content within our repository using the XQuery standard. The new features complement and accentuate the already rich SOA Governance features of Sun Service Registry."

Farrukh Najmi, Federated Information Management Architect,
Sun Microsystems, Inc.

For More Information

- Stirring the Secret Sauce of SOA, Persisting and Querying Metadata— ZapThink, April 2006
  http://www.zapthink.com/report.html?id=WP-0141
- The Power Behind the SOA Repository, JavaWorld, June 2005
  http://www.javaworld.com/javaworld/jw-06-2005/jw-0627-webservices.html
- Sun Service Registry
  http://www.sun.com/products/soa/registry
- Effective SOA Deployment, Using an SOA Registry Repoitory, September 2005
  http://www.sun.com/products/soa/registry/soa_registry_wp.pdf
- On Using XQuery for SOA Governance, Radovan Janecek's Blog, 2005
  http://radovanjanecek.net/blog/archives/On%20Using%20XQuery%20for%20SOA%20Governance.htm
- Raining Data Corporation
  http://www.rainingdata.com

Q&A

Supercharging SOA Registries With XML Persistence and Management

Ash Parikh
Director of Technology

Ajay Ramachandran
CTO

Premal Parikh
Lead Architect

XAP, Raining Data Corporation

TS-8098