
ORACLE®

ORACLE OPEN WORLD

experience INNOVATION

November 11–15, 2007



ORACLE

Data Services for Enterprise SOA:

A Review of Key Patterns and Deployment Strategies

Jeff T. Pollock, Senior Director Fusion Middleware

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.

Program Agenda

- *Key Points of This Talk:*
  - Data Service Concepts are Still Forming
  - Data Services Value is at Enterprise Scale
    - (not really a per-project value proposition)
  - Data Services are Key for SOA Success
  - Data Services Functional Patterns in Detail

Functional Patterns	Deployment Patterns
• Master Data Services • Query Data Services • Bulk Data Services • Data Access Services • Registry Data Services • Metadata Services	• Standalone Java Data Services • Standalone SOAP Data Services • Grid-Based Data Services

  - Data Service Futures – ~~Semantic~~ Web 3.0? (yes)


Data Services Backgrounder
ORACLE

A World Without Data Services:

…data services are a part of your everyday world

Q: Why can't the bank reconcile my London account with my NY account?



A: Master Data

Often, banking product codes are different by geography and must be reconciled with controlled hierarchy derivation rules, using appropriate regulatory guidance.

Q: Why do my credit card reward points take 15 days to appear on my airline website?



A: Batch Data

Usually there are specific cross-organization policies dealing with the recognition and conversion of rewards points, these are generally taken care of in regularly scheduled batch routines.

Q: Why does this store keep sending me three of the same catalogs every month?



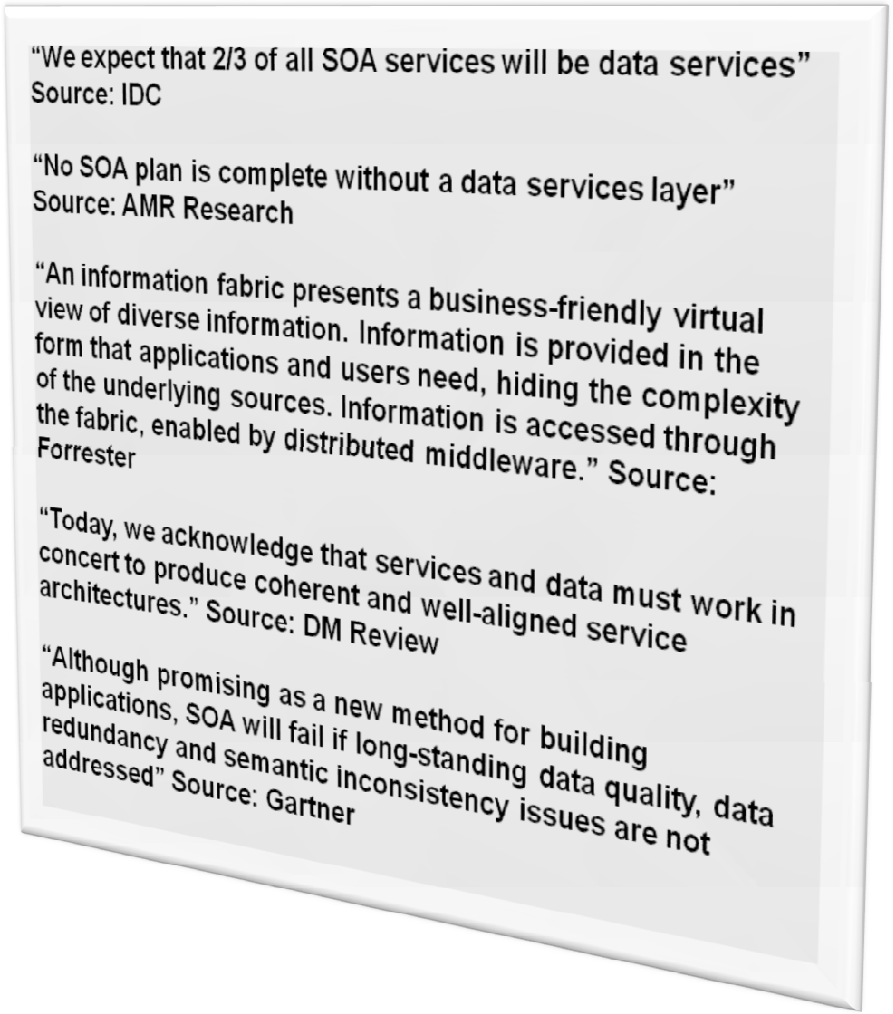
A: Data Quality

Many times large retailers maintain separate mail marketing lists which contain similar but slightly different data, without the data quality intelligence to match the duplicates.

Data Services for SOA:

Analyst Coverage Highlights Importance – and Confusion

- **Analysts Writing About "Data Services" or "IaaS" or "Information Fabric"**
  - Gartner
  - Forrester
  - IDC
  - AMR
  - Burton Group
  - Bloor Research
  - Zap Think
  - …etc etc.
- **One Thing They All Seem to Agree On:**
  - They're Crucial to SOA Success !
- **One Thing They Can't:**
  What are they ?



A Bit of Data Services History

A Rose by any Other Name…

- **Reality Check**
  - This new hype is about an old idea – **reusable data control is a good thing**
  - 1968 – formation of EDI document translation standards for banks
  - 1986 – Design by Contract: Precondition and Postcondition driven services
  - Early 1990's – tech roots of Informatica, IBM DataStage and Ab Initio
  - 1994 – Design Patterns: Factory, Adapter, Façade, Proxy, Mediator…
  - Late 1990's – the application server data services (Kiva, Net Dynamics…)

- **Before Java (and Before EAI or SOA)**
  - Data interchange among: banks, railways, airlines, militaries, etc.
  - Massive data warehouses for analytic purposes
  - Enterprise-scale software for mass data transform and DB loads
  - Basic object-oriented design patterns for decoupled data access

- **So What's Different About Data Services Today?**
  - Modern (ie:different) standards & protocols
  - Business scale, complexity, and expectations

So What Are Data Services?

Casting a Wide Net…

- **A General Software Pattern?**
  - Using contract-based development patterns for data distribution…
- **A Java Software Pattern?**
  - Pick your favorite TLA: DTO, DAS, SDO, JPA, ORM, JAXB, OXM…
- **A Web Services Stereotype?**
  - Dedicated SOAP services for reusable data manipulation and access…
- **An Enterprise Class of Conventional IT Services?**
  - The same tools we used to work with data before Java & SOA…

ALL OF THE ABOVE !

Java/J2EE Patterns

Data Services from a Java/Object/Component Viewpoint

BUSINESS SERVICE PATTERNS

- **SCA**
  - Service Component Architecture – Key framework spec for Java Business Services

DATA SERVICE PATTERNS

- **DAS, DTO & JDO**
  - Data Access Service – Java service for accessing RDB's
  - Data Transfer Object – Java object for moving data (w/in container)
  - Java Data Object – Java object for data containment (POJO-oriented)
- **JPA & SDO**
  - Java Persistence API – EJB 3.0 spec for accessing external data (typically relational)
  - Service Data Object – Entity objects targeted for SOA data use
- **ORM**
  - Object Relational Mapping – A generic term typically applied to some solution approach to handling "impedance mismatch"
    - Oracle Toplink – A proven industrial strength data services layer for managing Java object persistence
- **JAXB / OXM**
  - Java Architecture for XML Binding – Standard set of APIs for XML
  - Object to XML – Toplink extended implementation of JAXB

Conventional Data Management

Usually Integrated to SOA with Java/WSDL Wrappers

Business Intelligence & Analytic Platforms

- Function: crucial for business visibility and manipulation of data
- Key Value: most sophisticated data analysis and reporting
- SOA Intersection: integrated into SOA for closed-loop analytics

Data Integration

- Function: moving and transforming data, geared towards enterprise-scale data challenges, includes: **ETL**, **Quality**, **CDC…**
- Key Value: proven, fastest and feature rich data integration
- SOA Intersection: SOA adds new APIs for easier service coupling

The Rest…

- EII – Federated Query
- Data Quality Engines
- Master Data Management
- Metadata Management
- Enterprise Search
- Content Management


Data Services @ Oracle
ORACLE

A Taxonomy of Data Services

JAVA DATA SERVICE [IMPLEMENTATION] PATTERNS

- **DAS & DTO –** Access and in-container data transfer
- **JPA, JDO & SDO –** Enterprise intra-container data entity transfer
- **JAXB & OXM (Toplink) –** Access and binding to XML data

SOA DATA SERVICE [FUNCTIONAL] PATTERNS

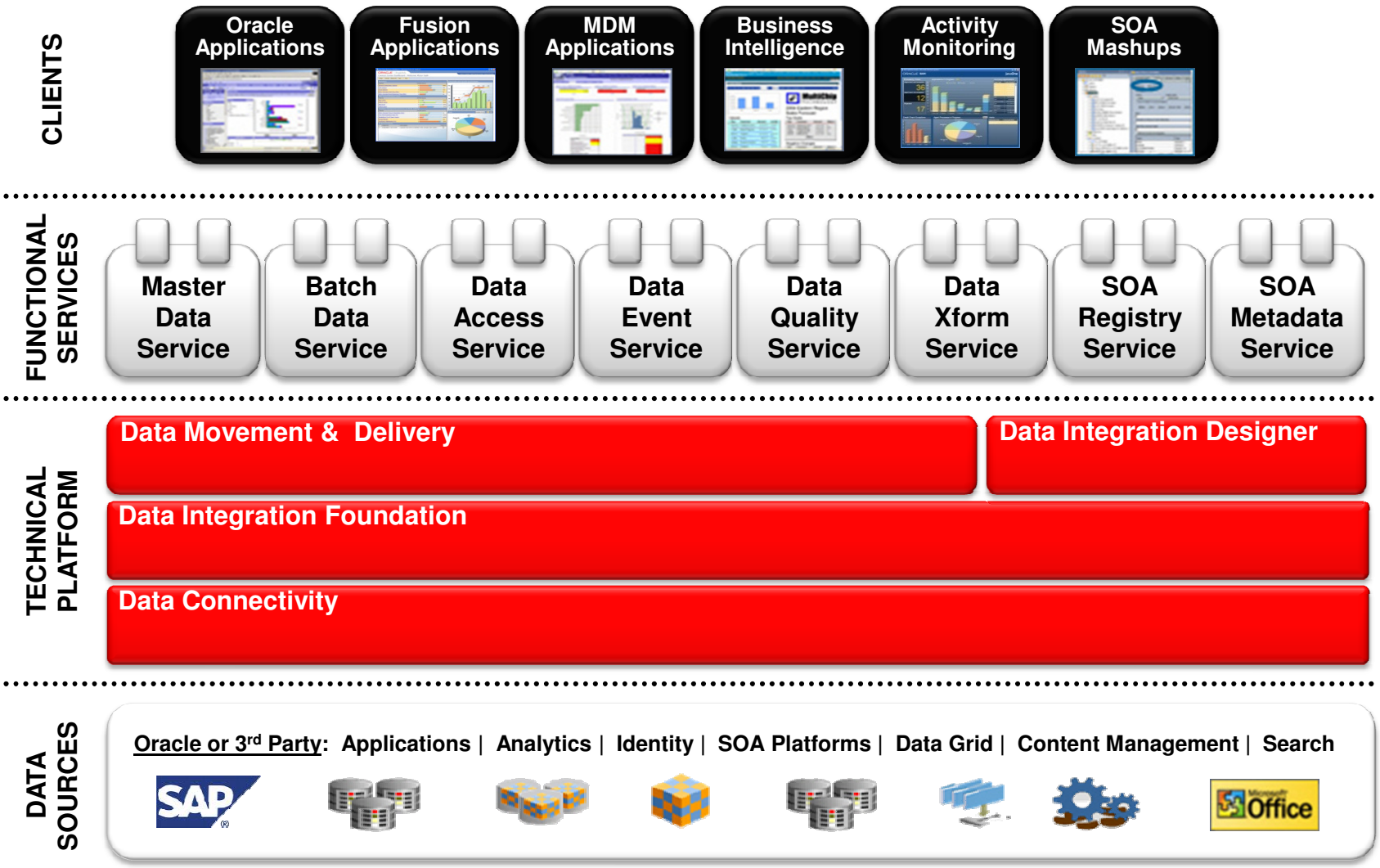
- **Master Data Services –** Golden Records, Hierarchies, & Policies
- **Batch Data Services –** Bulk Refresh, CDC, Trickle Feeds, SCD…
- **Data Access Services –** Virtualization, Federated Query, ORM…
- **Data Event Services –** Low Latency CDC, Trigger-Based CDC…
- **Data Quality Services –** Cleansing, Matching, De-Duplication…
- **Transformation Services –** XSLT, Large Doc/XML Xform, EDI/B2B…
- **Registry Services –** Data, Service, and System Registries/Directories
- **Metadata Services –** Model and Attribute Persistence, Versioning etc.

DATA SERVICE [DEPLOYMENT] PATTERNS

- **Data Hub –** Centralized cache (variation of Data Virtualization) or repository
- **Data Grid –** Distributed RAM-based data partitions with decentralized access

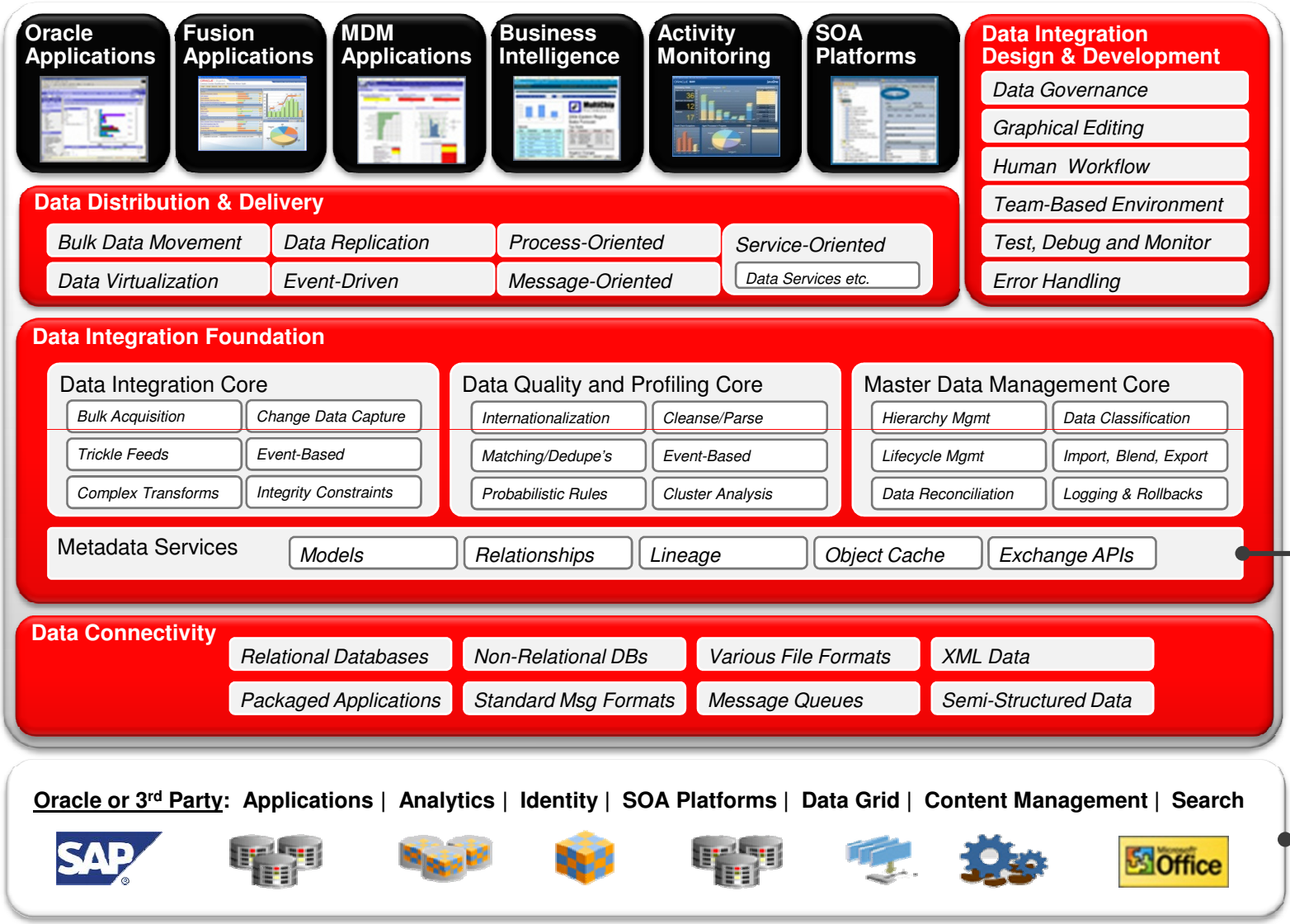
Data Services Logical View

In an Enterprise Context, how do Data Services Fit?



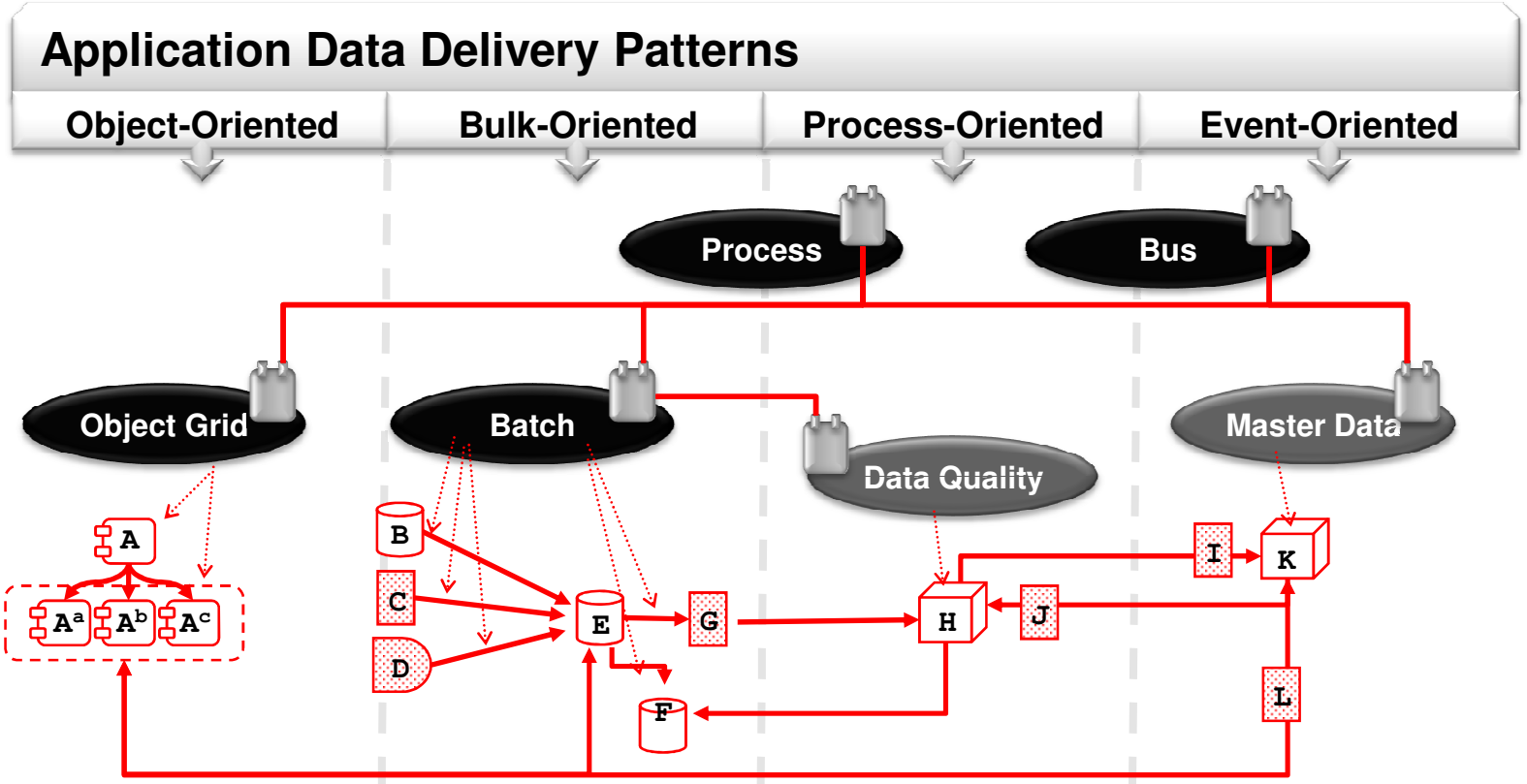
Oracle Data Integration Platform

SOA & BPEL Driven Data Integration Offering from Oracle



Data Delivery Styles

A Key Enabler for Data Services is Flexible Service Styles



Object-Oriented	Bulk-Oriented	Process-Oriented	Event-Oriented
Use Case: *Hard Realtime Composite Applications*	**Use Case:** *Realtime and Batch BI/BAM Applications*	**Use Case:** *Long-lived or Workflow-oriented Transactional Systems*	**Use Case:** *Event-Action Business Rule-driven Transactional Systems*
eg: Trade Data Cache	**eg : BI/DW Refresh**	**eg : Supply Chain Data**	**eg : Sales/Cust Data Hub**
• **Fast Data Cache**	• **Fast Bulk Transforms**	• **Multi-step XA Flow**	• **Publish and Subscribe**
• **Object-based API**	• **Set-Based Mgmt.**	• **Approvals Workflow**	• **Changed Data Capture**

Key Oracle Products for Data Services

- **Data Integration Core Components**
  - Oracle Data Integrator
  - Oracle Data Quality & Data Profiling
  - Oracle Data Relationship Manager (Hyperion Master Data Management)
  - Oracle Coherence Data Grid
  - Oracle BPEL Process Manager
  - Oracle Enterprise Service Bus
- **Other Key Oracle Products**
  - BI-EE – Closed Loop Analytic Services
  - Oracle DB – RDF / Spatial DB


SOA Data Service Patterns
• Master Data Services
• Batch Data Services
• Data Access Services
• Data Quality Services
• Data Transformation Services
• General Registry Services
• General Metadata Services
ORACLE

Master Data Services

SOA Data Services Pattern

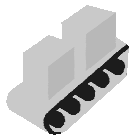
Name: Master Data Services
Motivation: The complexity of enterprise data environments makes it hard to find or assemble trusted, high quality business data, hierarchies, and policies.
Usage: May be used as a reference service during SOA transactions, during bulk data movement, typically applied with transformations.
Variations: Master Data Hub, Master Data Cache, Master Data Applications (CDI, PIM, FIN…etc)
References: See Oracle MDM, IBM MDM, SAP MDM, Kalido, Siperian, etc.

Business Motivation

…keeping track of trusted business metrics



Ledger #	Acct	Legal Entity
1-MR7	6-123	AMER-SW01
6-2X8	J-F3	AMER-SW01

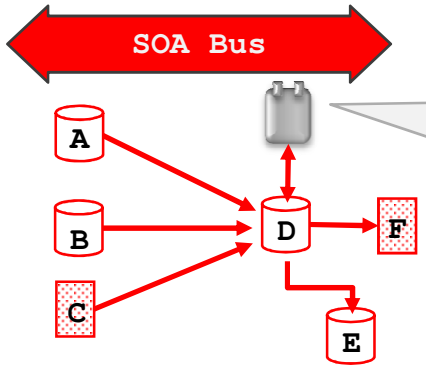


Vendor #	Name	Address
6-123	ABC	3 Main Street
2-234	HGL	1 Market Ave.



Part #	Desc	Unit of Measure
ABC	Bolt	12 ct box
HGL	Widget	Piece

Technology Picture



Trusted single point of reference

- **Golden Records**
- **Master Hierarchy**
- **Canonical Objects**

```
A|B|C→D = Load
D→E = Synchronize
D→F = Audit & Govern
```

Oracle Solutions

- Oracle Master Data Management Applications
  - Oracle | Siebel Universal Customer Master
  - Oracle Customer Hub
  - Oracle Product Information Hub
  - Oracle Financial Consolidation Hub
- Oracle Fusion Middleware MDM Solutions
  - Oracle | Hyperion Data Relationship Mgmt.
  - Oracle Data Integrator, Active Hubs
  - Oracle Coherence Data Grid (Object Cache)

© Copyright 2007, Oracle Corporation. All Rights Reserved.

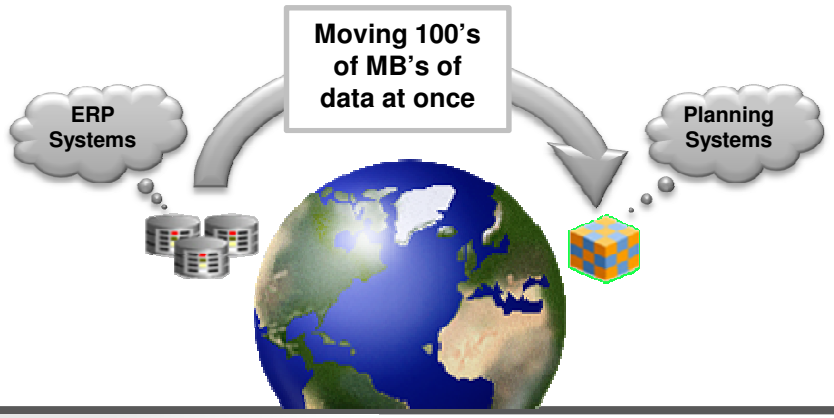
Batch Data Services

SOA Data Services Pattern

Name: Batch Data Services
Motivation: ERP, Data Warehouses, Business Intelligence and Performance Management Applications require bulk data movement.
Usage: May be used for Replication, Bulk Refresh, Application Migration, Large File Transformations, Changed Data Capture
Variations: ETL (requires new hardware), E-LT (low cost, high performance), Low-Latency Logminer CDC, Declarative or Procedural Design
References: See Oracle Data Integrator, IBM DataStage, Informatica, Ab Initio, etc.

Business Motivation

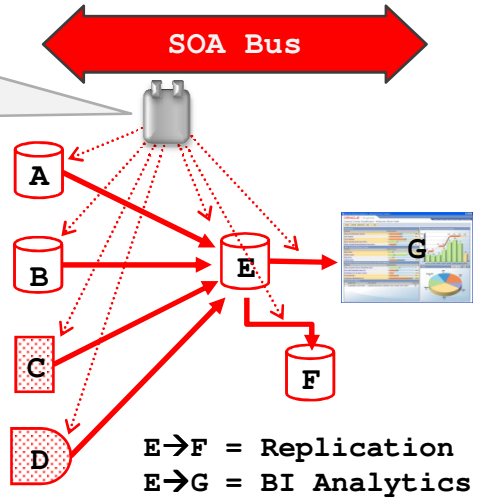
…data must be fresh, regardless of quantity



Technology Picture

Loose-coupled bulk data service

- **ETL / E-LT**
- **Replication**
- **Changed Data**
- **Bulk X-Form**



Oracle Solutions

- Oracle Data Integrator
  - High-Performance E-LT Architecture
  - 100% Native Java & SOA Services
  - Declarative Design Framework
  - Open, Heterogeneous Platform
  - Event-Driven | Changed Data Capture
  - Built-in Data Quality
- Oracle Warehouse Builder
  - Built-in to Oracle Database 11g
  - Native Support for Oracle OLAP

© Copyright 2007, Oracle Corporation. All Rights Reserved.

Data Access Services

SOA Data Services Pattern

Name: Data Access Services
Motivation: Combine a shared abstraction (Canonical Model) with instance virtualization (Data Mashup) to present a simplified Query Interface to consuming Applications.
Usage: Traditionally part of J2EE/.NET layer, in a SOAP environment the extra step of conversion to XML (usually Canonical) is added to the process
Variations: Query Federation, Data Hub & Spoke (Object|SQL|XQuery), Object-Relational Mapping
References: See Oracle Application Server, Oracle Coherence Data Grid, Oracle Data Integrator, BEA AquaLogic, Ipedo, Composite Software, Meta Matrix, IBM DB2ii

Business Motivation

…data must be easily accessible

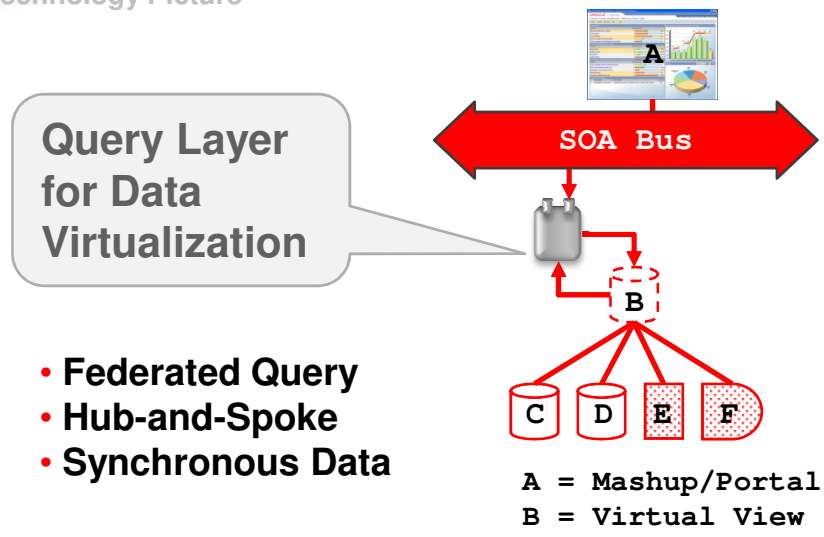
Query Federation (no cache)

- Motivation: business policy will not allow co-mingling of data in same repository

Data Virtualization (with cache)

- Motivation: save development time and money by presenting a simplified view, or other interface, to data whose origin is distributed geographically – *note there is no true requirement for federation*

Technology Picture



- **Federated Query**
- **Hub-and-Spoke**
- **Synchronous Data**

Oracle Solutions

- Oracle Application Server Enterprise Edition
  - SQL ORM Layer for SDO/JDO (TopLink)
- Oracle Data Integrator
  - Active Data Hub (Database Virtualization)
- Oracle Coherence Data Grid
  - Object Grid (Application Data Virtualization)
- Oracle Business Intelligence-Enterprise Edition
  - Included Query Federation Engine
- Oracle Database 11g (Streams & Gateways)
  - Included Query Federation Engine

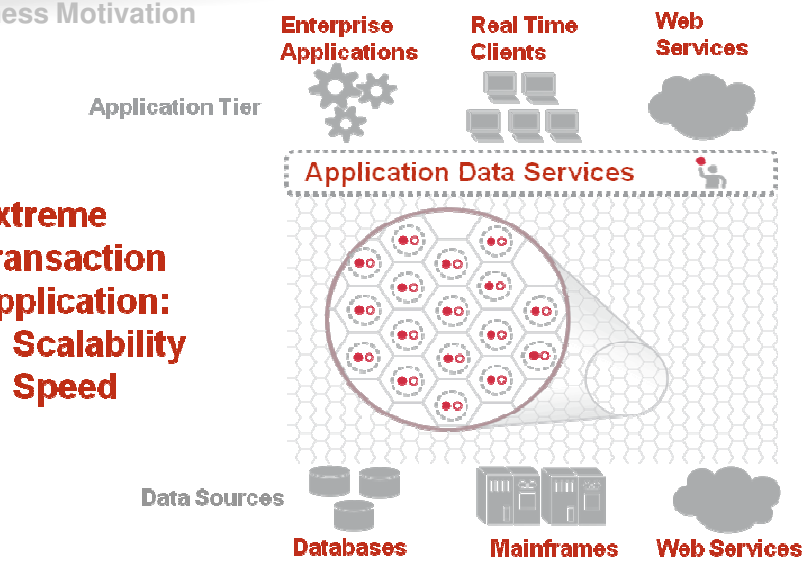
© Copyright 2007, Oracle Corporation. All Rights Reserved.

Virtualization with Data Grid

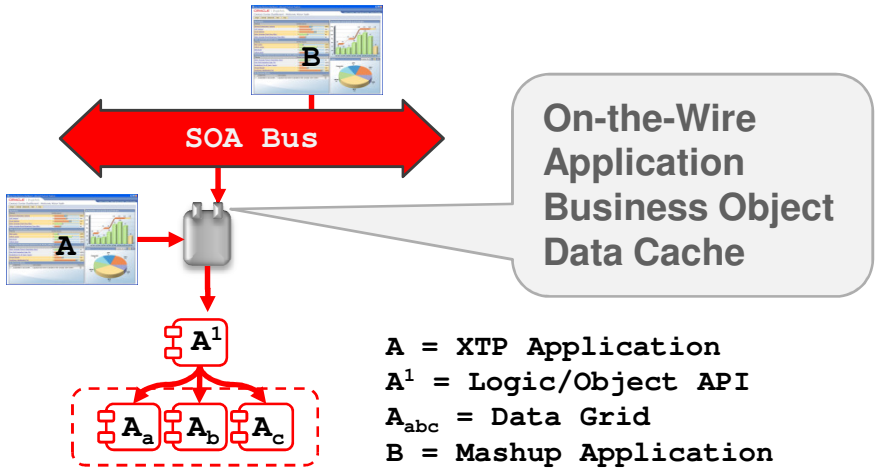
SOA Data Services Pattern

Name: Data Grid Services
Motivation: In-memory data frequently needs to span multiple applications, due to geographical factors, or limitations of RAM capacity.
Usage: Typically deployed for federated state-full persistence at business object tier, in order to predictable "scale-out" applications while maintaining exceptionally fast performance
Variations: Java, .Net, C++ variations. Peer-to-Peer and Hierarchical Clusters, UDP/TCP…
References: See Oracle Coherence, Gigaspaces, Gemstone

Business Motivation



Technology Picture



Oracle Solutions

- Oracle Coherence
  - Clustering
  - Distributed Objects, Maps & Caching
  - Real-Time Events, Listeners
  - Parallel Queries & Indexing
  - Data Processing and Service Agents
  - Continuous Views
  - Aggregation
  - Persistence, Sessions
  - Java, .NET and soon C++ support

© Copyright 2007, Oracle Corporation. All Rights Reserved.

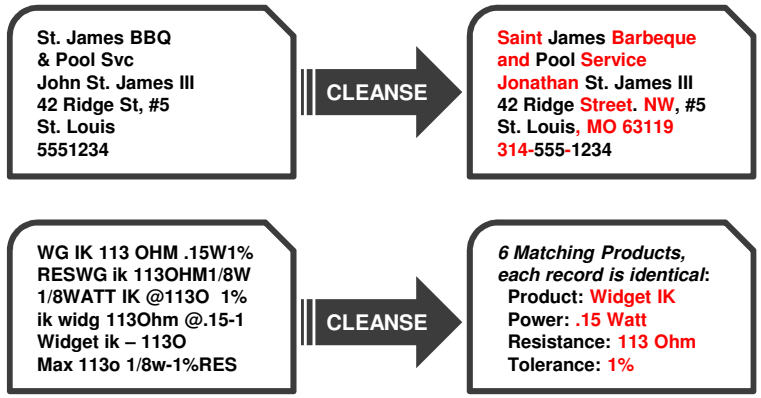
Data Quality Services

SOA Data Services Pattern

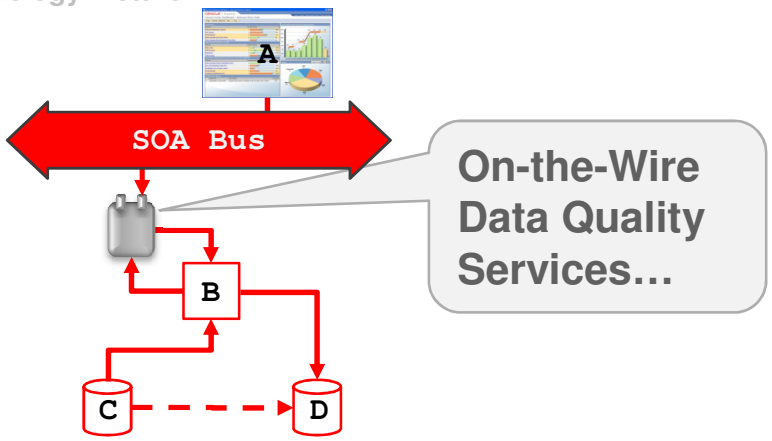
Name: Data Quality Services
Motivation: Automatically improve the quality of bad data so that legacy data resources become more valuable and usable.
Usage: Traditionally applied in batches to clean up Data Warehouses and BI repositories, the usage is now shifting to realtime and preventative use case, to cleanse the data before it's a problem
Variations: Declarative/Rule-Driven, Probabilistic or Statistical Learning-based, Domain-Specific and Content-Oriented Data Quality
References: See Oracle Data Integrator, Oracle Warehouse Builder, Informatica etc.

Business Motivation

…data must be clean and standardized



Technology Picture



A→B→A = Message Cleansing | Matching
C→D = Bulk Cleansing | Matching

Oracle Solutions

- Oracle Data Integrator
  - Built-in Data Quality & Data Profiling
  - High-Performance E-LT Architecture
  - 100% Native Java & SOA Services
  - Declarative Design Framework
  - Open, Heterogeneous Platform
  - Event-Driven | Changed Data Capture
- Oracle Warehouse Builder
  - Built-in to Oracle Database 11g
  - Native Support for Oracle OLAP

© Copyright 2007, Oracle Corporation. All Rights Reserved.

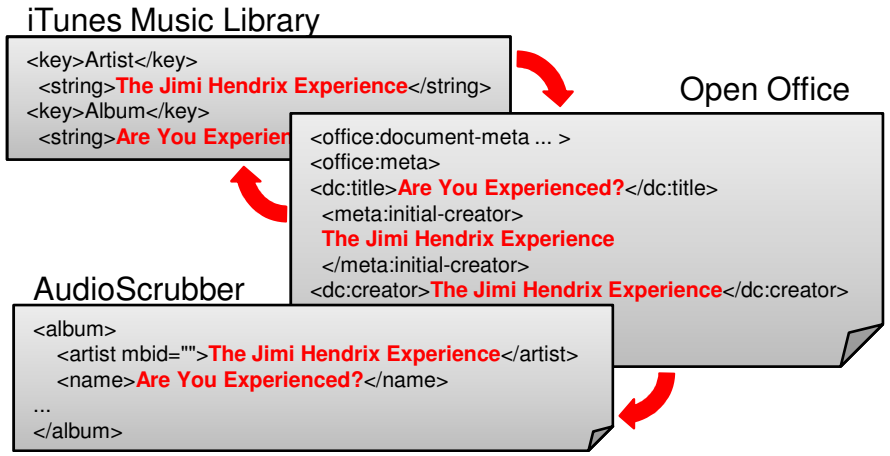
Transformation Services

SOA Data Services Pattern

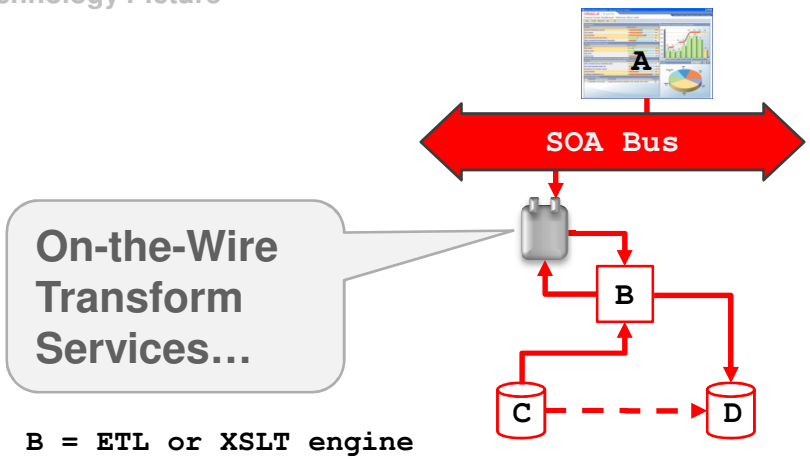
Name: Data Transformation Services
Motivation: Present a reusable service for WSDL-driven data transformation – generally supporting multiple types of transformation (such as: RDB→RDB, XML→RDB, XML→XML, etc)
Usage: Best practice for enterprise systems with centrally maintained service families.
Variations: XSLT Factory, ETL Engine, Canonical Mediator Service (either XSLT or ETL driven)
References: See Oracle Application Server, Oracle Data Integrator, Contivo Builder, Tibco Data Exchange, BEA Aqua Logic, etc.

Business Motivation

…data must be in correct format



Technology Picture



B = ETL or XSLT engine
A→B→A = Message/Doc Transformation
C→D = Bulk Data Transformation

Oracle Solutions

- Oracle Application Server Enterprise Edition
  - JAXB/OXM Layer etc (TopLink)
  - SQL ORM Layer for SDO/JDO etc (TopLink)
- Oracle Data Integrator
  - High-Performance E-LT Architecture
  - 100% Native Java & SOA Services
  - Declarative Design Framework
  - Open, Heterogeneous Platform
  - Event-Driven | Changed Data Capture
  - Built-in Data Quality

© Copyright 2007, Oracle Corporation. All Rights Reserved.

Data Event Services

SOA Data Services Pattern

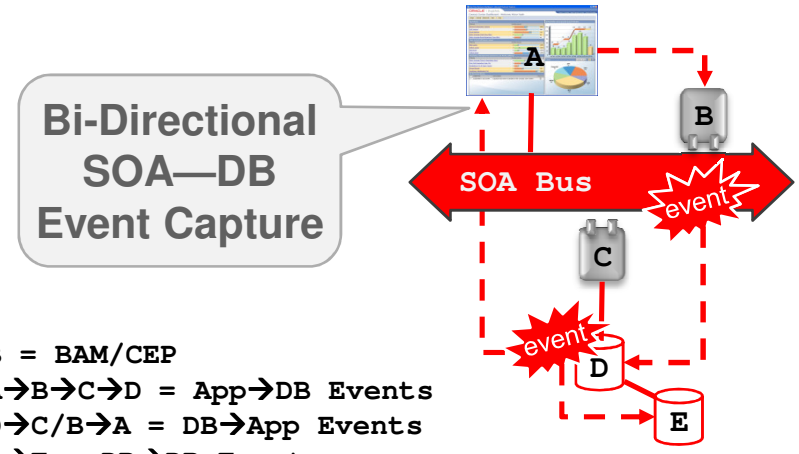
Name: Data Event Services
Motivation: Every part of the data environment must be capable of trapping actions, checking policies and taking action based on those policies
Usage: Typically deployed on a given technology tier (eg: within Java, or on a Bus, or in a DB), but should be capable of calling to other event systems (eg: Java event triggers SOA triggers DB)
Variations: EDA (Event Driven Architecture), CDC (Change Data Capture), CEP (Complex Event Processor), Java Event Listeners…etc.
References: See Oracle Application Server, Oracle Data Integrator, Oracle Event Driven Architecture Suite, Tibco CEP. BEA Event Server

Business Motivation

…actions must happen in realtime



Technology Picture




```
B = BAM/CEP
A→B→C→D = App→DB Events
D→C/B→A = DB→App Events
D→E = DB→DB Events
```

Oracle Solutions

- Oracle Data Integrator
  - Trigger-based CDC
  - Log-based CDC
  - Notifications
  - Built-in Integrity
  - Declarative Rules
- Oracle Event Driven Architecture
  - Business-Rule Driven
  - Associative Transaction Correlation
  - Tightly Integrated w/Activity Monitor

© Copyright 2007, Oracle Corporation. All Rights Reserved.

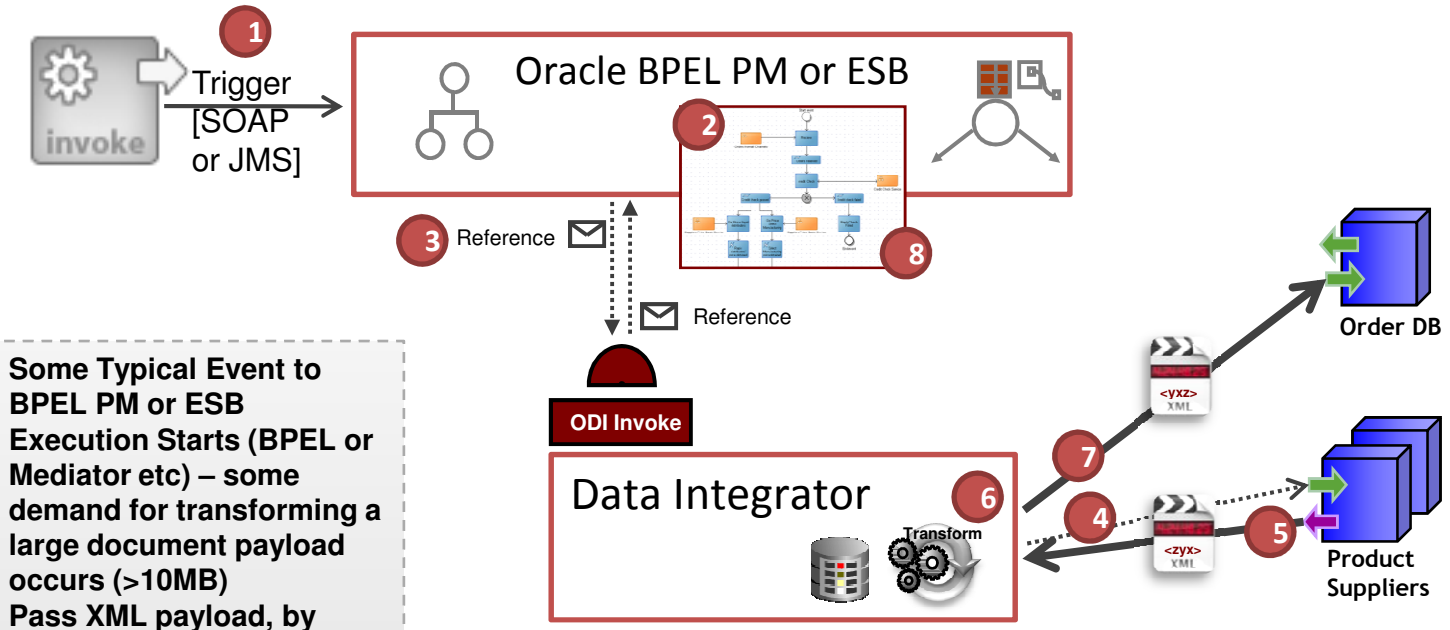

Simple Data Service Use Cases for SOA
ORACLE

Simple Data Services for SOA Platform

- **Transformation Services for Large Documents**
  - XML Document Transformation
  - EDI (B2B) Document Transformation
  - File-to-DB Loading Services
- **Batch Data Services & DB Orchestration**
  - DB Orchestration, for DW Refresh
  - Heterogeneous DB Replication
- **Data Access Services & Data Virtualization**
  - DB Web Service Wrappers, with CDC
  - Canonical Data Hub Lifecycle Management

Data Services for Large XML/EDI Docs

(Data Transformation Services)



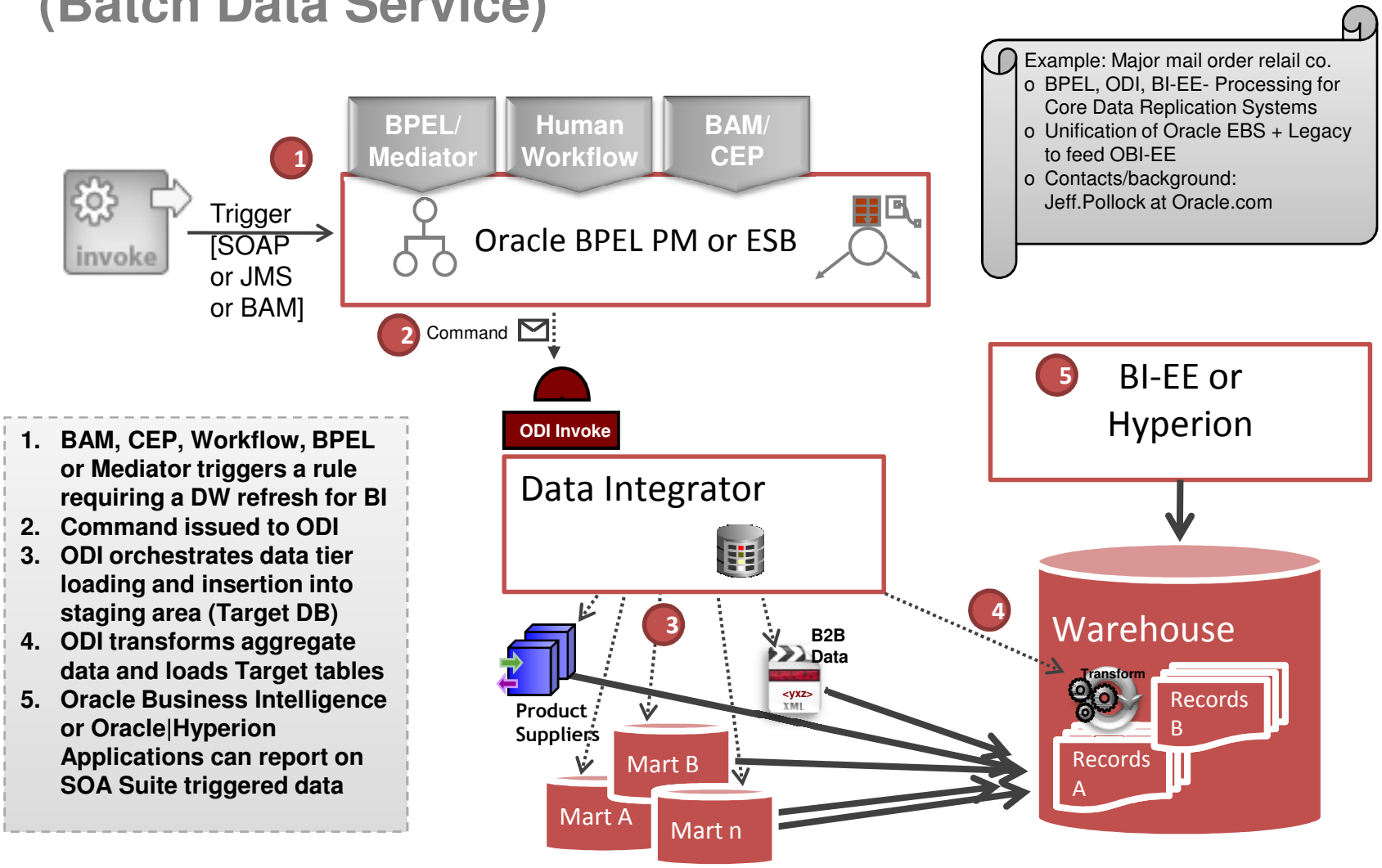
1. **Some Typical Event to BPEL PM or ESB**
2. **Execution Starts (BPEL or Mediator etc) – some demand for transforming a large document payload occurs (>10MB)**
3. **Pass XML payload, by reference, to ODI**
4. **ODI requests payload**
5. **ODI loads payload**
6. **ODI transforms payload**
7. **ODI sends payload wherever instructed**
8. **Core BPEL/ESB processing completes**

Example: Major domestic retailer

- BPEL, ESB, ODI - Processing of large book catalog files and retail supplier data
- Conversion of 30-70MB Flat files, expands to 200-400MB XML docs
- Contacts/background: Jeff.Pollock at Oracle.com

Data Service for BI/DW Control

(Batch Data Service)



Example: Major mail order relail co.
- BPEL, ODI, BI-EE- Processing for Core Data Replication Systems
- Unification of Oracle EBS + Legacy to feed OBI-EE
- Contacts/background: Jeff.Pollock at Oracle.com

1. **BAM, CEP, Workflow, BPEL or Mediator triggers a rule requiring a DW refresh for BI**
2. **Command issued to ODI**
3. **ODI orchestrates data tier loading and insertion into staging area (Target DB)**
4. **ODI transforms aggregate data and loads Target tables**
5. **Oracle Business Intelligence or Oracle|Hyperion Applications can report on SOA Suite triggered data**

Understanding Performance Choices

When you need to transform data at large size

Depends on whether an intermediary XML format is useful for other processing (use ESB), or if joining File data to tabular RDB data is required (use ODI)

Less than 10MB

(source) \ (target)	XML	File	DB
XML	ESB	ESB	ESB
File	ESB	ESB	depends
DB	ESB	depends	ODI

Depends on ho much cross-referencing among the data values and rows is required during transformation – the more there is, the faster ODI will perform relative to ESB

Between 10-50MB

(source) \ (target)	XML	File	DB
XML	depends	depends	ODI
File	depends	ODI	ODI
DB	ODI	ODI	ODI

If the source and target are both XML, and there is no cross-referencing of data among rows, then a streaming-type or parallel-engine-type approach might scale

Greater than 50MB

(source) \ (target)	XML	File	DB
XML	depends	ODI	ODI
File	ODI	ODI	ODI
DB	ODI	ODI	ODI

ORACLE

*caveat – always benchmark if you are unsure and require best possible results


Caveats and Concluding Thoughts
ORACLE

Data Service Pattern Tradeoffs

Each Pattern Class Carries Assumptions and Limitations

J2EE/Java

J2EE Pattern Assumption

- Business Object tier instantiated in Java
- The master reference point is marshaled to Java at some process point

Limitation #1: Scalability

- RAM memory-model vis JVM
- Query performance across JDO/SDOs
- Non-declarative access methodology (API vs. SQL)
- Transformation overhead (SQL to Java to XML vs SQL to XML)

Limitation #2: Redundancy

- Replicating data instances across compute tiers / inelegant
- Synch consistency issues to persistence becomes a challenge at scale

Limitation #3: Granularity

- Specific patterns solve specific issues (eg: Object to RDB)
- These (DTO, DAS, SDO, JPA, ORM, JAXB, OXM, etc.) are programming patterns – not business or system service-level patterns

SOA/XML-Centric

SOA Service Pattern Assumption

- Business Object tier typically defined with Canonical XML
- The master reference point is "on the Bus" or XML document-oriented

Limitation #1: Bloat

- Data + XML Markup = 3 to 6 times original data size
- This can take a 30MB product catalog to >100MB of XML

Limitation #2: Performance

- XML processing is inherently disadvantaged (DOM/Infoset model)
- Data sets are typically navigated with XPath and/or in-memory
- Streaming XML approaches have limited applicability in complex data sets

Limitation #3: Expressivity

- XML does not guarantee precise modeling semantics (relative to RDB)
- This limits automated capabilities
- It also limits overall transferability of common exchange models (models must be accompanied by explanatory human readable text specification)

Conventional DI

Conventional Pattern Assumption

- Business Object tier may be defined within any technology (typically RDB)
- The master reference point is typically (data+metadata) a proprietary API

Limitation #1: Ownership Cost

- Different core technology components (ETL, MDM, IdM, etc) may have radically different underlying technology, architecture, and integration styles
- Impact – Training Overhead
- Impact – Data Center Overhead
- Impact – Duplicate Hardware
- Impact – Integration Costs

Limitation #2: Flexibility and Reuse

- Once conventional (non-SOA) integrations are tightly coupled
- Data bindings often result in the classical n^2 mapping nightmare

Limitation #3: Lock-in Tendencies

- Monolithic conventional architectures create incentives to buy more from existing vendors, which leads to a repetitive cycle and vendor lock-in (aka: the SAP approach)

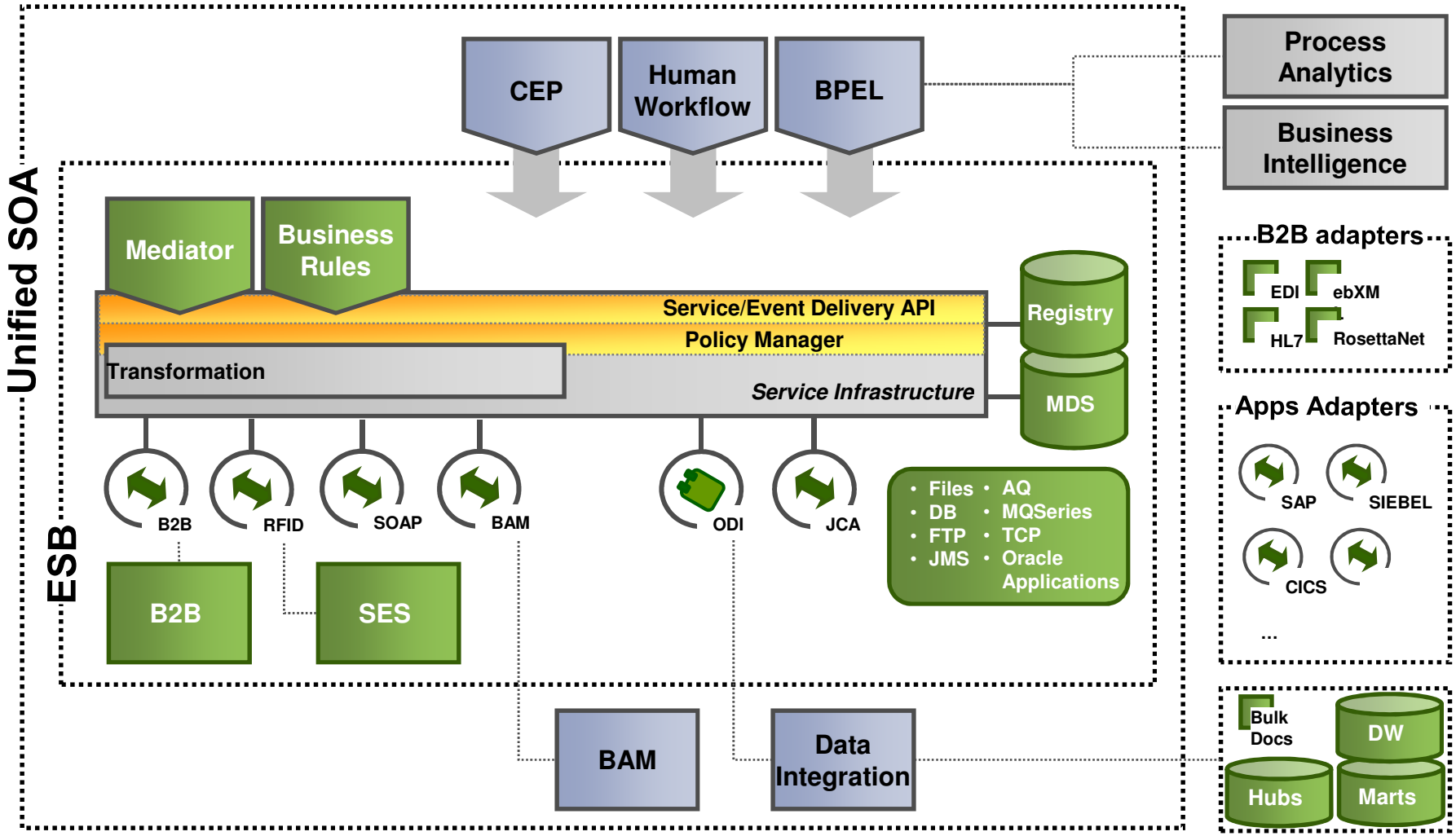
Key Recommendation – Go Hybrid

EG: Select a hybrid platform for Java|SOA|Conventional

- **Core Hybrid Pattern Assumption**
  - Best-of-breed platform enables customer-choice – right tools for the job
  - The master reference points, Business /DataObjects, may be instantiated as Java, XML, or manipulated solely as RDB structures depending on the use case
- **Requirement #1: Leverage Native Architecture Patterns**
  - Avoid “bolt-on Java” – because API-only will not lower TCO, only access layers
  - Avoid “bolt-on SOA” – SOAP-wrapper will not improve performance, or data decoupling
  - Avoid “Java or XML bulk data processing” – limited power when used in isolation
- **Requirement #2: Demand Runtime Integration**
  - Avoid having to integrate the integration platforms!
  - Look for cross-system Human Workflow, common Hardware and Platform runtime arch.
- **Requirement #3: Demand Design-time Integration**
  - Avoid having to train developers on four or five different tools for the same use cases
  - Look for integration tools that use the same Design Platform across runtime engines

Single Runtime Infrastructure

Hybrid of Process, Message and Data Integration



Beyond Data Services? –Semantics

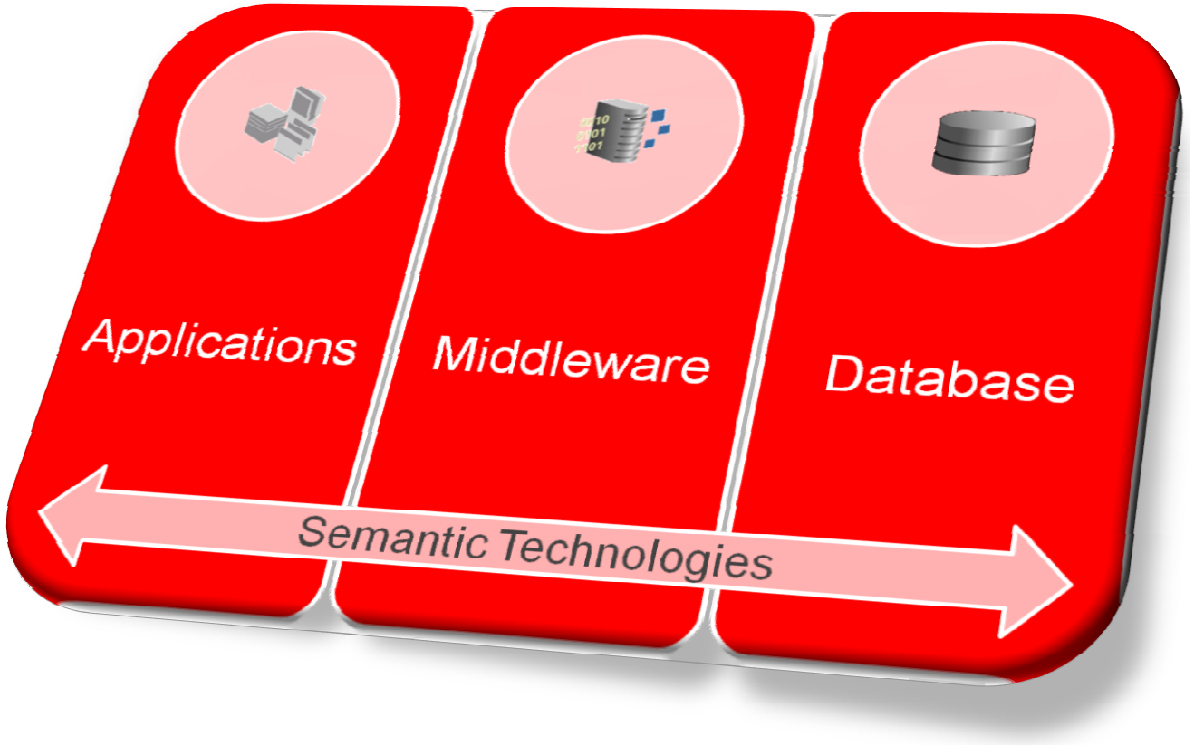
AKA: “Web 3.0”, “Hyperdata,” “Data Web” or “Semantic Web”

- **What is it? - Smarter Metadata**
  - RDB designed for Persistence, XML designed for Documents, UML designed for Code
  - OWL (ontologies) & RDF (graph data) were SPECIFICALLY DESIGNED FOR METADATA
    - Relationships are first-class objects, Data model is most expressive (captures more structural data rules), Schemas are more organic/flexible, Vocabularies are more naturally expressed
- **What can it do? - More Data Automation***
  - Predictable Superset of Reference Data (the point is to automate reference data management, not to replace existing reference data in XML, RDB or flat structures)
  - Transitive Inferences (B partOf A, C partOf B, we may infer that C partOf A) – the value of this is because the inference is on the MODEL, not on the QUERY
  - Queries can be simpler (“Get all A” when model is “A = B union C” versus “Get all A where union B and C” when model is just A, B, C (note the union in MODEL vs. in QUERY)
- **Why for SOA? - More Adaptive Data Services****
  - Automatic and dynamic business process orchestration (non-brittle BPEL)
  - Automatic service discovery (smart UDDI)
  - Transformation-less data mediation (XSLT-free views of local XML documents)
  - Seamless (object → to → XML → to → relational) reference data transfer-ability
  - Bi-directional schema generation regardless of syntax, structure and semantics

* True today
** True tomorrow

Semantic Technology Defined

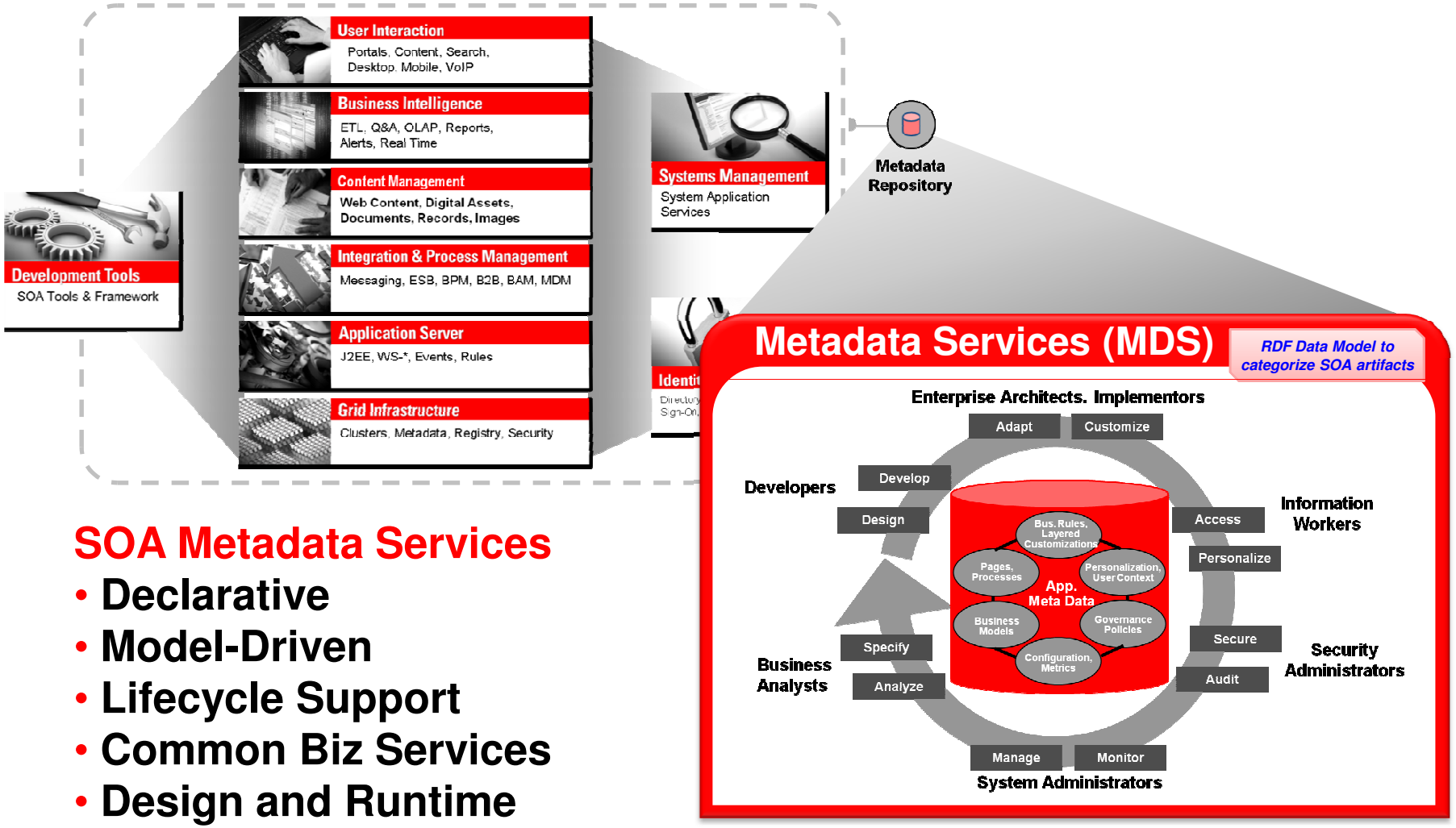
- Support or otherwise natively implement the World Wide Web Consortium's standard metadata formats of RDF (Resource Description Framework) & OWL (Web Ontology Language)



- Employ a predominantly model and metadata driven declarative environment that effectively achieves the effects of an autonomic (self-learning) system.
- Leverage modern forms of Artificial Intelligence (AI) technology to perform expert system type behavior.

Oracle's Flexible Metadata in 11g

Metadata Service (MDS) Uses Native RDF Today



SOA Metadata Services

- Declarative
- Model-Driven
- Lifecycle Support
- Common Biz Services
- Design and Runtime

Recap:

- **Key Points of This Talk:**
  - Data Service Concepts are Still Forming
  - Data Services Value is at Enterprise Scale
    - (not really a per-project value proposition)
  - Data Services are Key for SOA Success
  - Data Services Functional Patterns in Detail

Functional Patterns	Deployment Patterns
• Master Data Services • Query Data Services • Bulk Data Services • Data Access Services • Registry Data Services • Metadata Services	• Standalone Java Data Services • Standalone SOAP Data Services • Grid-Based Data Services

  - Data Service Futures – ~~Semantic~~ Web 3.0? (yes)


Q&A



ORACLE IS THE INFORMATION COMPANY

Thank You

For More Information

search.oracle.com



or

oracle.com

Speaker Contact & Biography



Jeff.Pollock at oracle.com

Mr. Pollock is an integration technology expert and author of the enterprise software book "Adaptive Information“ (John Wiley & Sons). Currently a Senior Director with Oracle’s Fusion Middleware group, he is responsible for management of Oracle’s data integration product portfolio, Mr. Pollock was formerly an independent systems architect for the Defense Department, Vice President of Technology at Cerebra and Chief Technology Officer of Modulant, developing semantic middleware platforms and inference-driven SOA platforms from 2001 to 2006. Throughout his career, he has architected, designed, and built application server/middleware solutions for Fortune 500 and US Government clients. Previously, Mr. Pollock was a Principal Engineer with Modem Media and Senior Architect with Ernst & Young’s Center for Technology Enablement. He is also a frequent speaker at industry conferences, author for industry journals, active member of W3C and OASIS, and formerly an engineering instructor with University of California at Berkeley’s Extension on the subjects of object-oriented systems, software development process and enterprise systems architecture.

The preceding 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.


ORACLE