Enterprise Java Tools & Techniques

Data Aggregation & Grids in SOA

SiliconIndia – Java Conference – October 15th, 2011
Hyderabad

SUNILA GOLLAPUDI
Technology Specialist

Broadridge®

Accurate | Dependable | Efficient

AGENDA

- Broadridge
  - *A Quick note on Our Business*
- Broader Picture – The Financial Service Industry
  - *Key Trends, Requirements and Challenges*
- Outlining the Problem Context and Solution Areas
- A Solution that "**Just Works**" …
  - Service Oriented Architecture & Data Services
    - *Scope, Architecture, Solutions & Options*
  - Data Aggregation/Composition & Data Federation/Virtualization
    - *Scope, Architecture, Solutions & Options*
  - Data Availability & Reliability/Quality
    - *Scope, Architecture, Solutions & Options*
  - Scalability in large volume context
    - *Scope, Architecture, Solutions & Options*
- Q&A

Broadridge

We are an Industry Leader

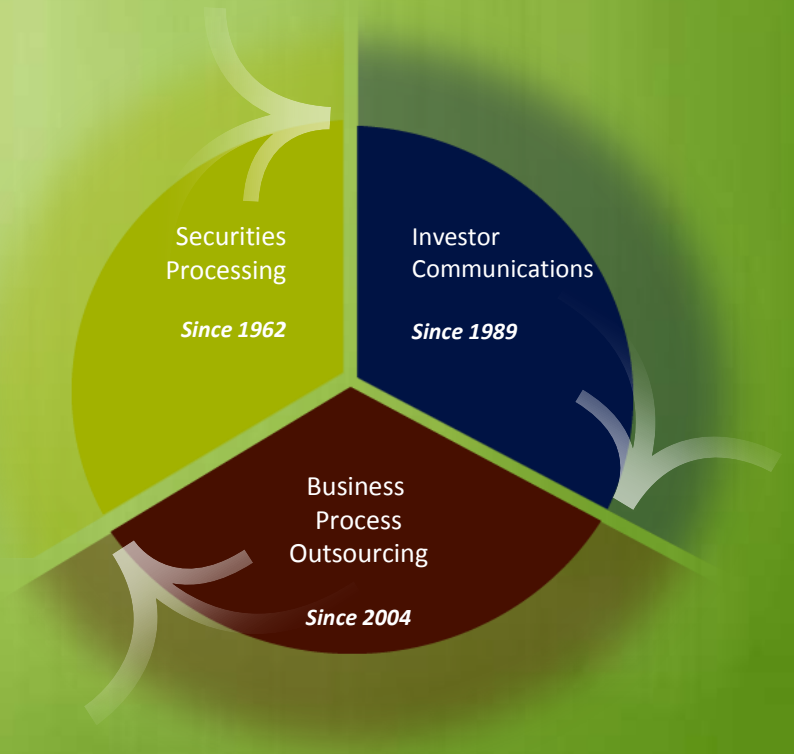
- **$2.2 billion** in annual revenues as of **FY 2010**
- **$342 million** in annual pre-tax earnings
- **40 years** of experience in securities processing business
- Senior management team averages **15 years** of tenure
- Ranked #1 in Black Book of Outsourcing: Brokerage Processing Providers Survey
  - #1 in 14 of 18 categories surveyed vs. 52 other providers

We provide Mission-Critical Solutions

- Pure **Technology** and Outsourcing service provider
- Components of our securities processing solutions used by 8-of-the-top-10 U.S. broker-dealers
- **Process on average $4 trillion securities settlements daily**
- **Process on average 3 million trades per day**
- **Process more than 1 billion investor communications** annually for every broker/dealer in the U.S.
- **Over 30 million customer accounts are custodied on our brokerage platforms**

Our offerings are broad and flexible

- Securities processing capabilities for more than 50 countries
- Solutions ranging from hosted service bureau, to customized BPO supporting full outsourcing



Broadridge

Financial Service Industry

(Key Trends, Requirements & Challenges)

- **Mergers & Acquisitions**

  - Financial Industry is most dynamic and M&A that enhance market position or add offerings is frequent
  - ***Challenge is to federate the Duplicate Data***

- **Compliance Reporting**

  - Compliance to changing Regulatory norms is one of the biggest needs in Financial Service Industry.
  - As internal systems are optimized for operations & not compliance, integrating the data from these systems often proves to be expensive
  - ***Challenge is to virtualize data across these operational systems***

- **Risk Management**

  - Financial institutions must continuously monitor exposure to a range of risks
  - Aggregating a single view of institution-wide risk in real-time is the requirement
  - ***Challenge is DATA: Data Quality, Data Availability and Data Access***

- **Reference Data Sharing**

  - Reference data facilitates rapid, error-free execution of financial transactions and analysis across multiple geographical markets
  - **Challenge is single virtual source for reference data and enables federation of additional master and operational data to complete a financial transaction or analysis**

Broadridge

Outlining the Problem Context

- Technology Agnostic way of Accessing Data

  Data Services and Service Oriented Architecture

- Centralized Data Access

  Data Aggregation/Composition & Date Federation/ Virtualization

- Data Reliability & Availability – **Seamless access to Real-time Data**

  Data Cleansing, Clustered Caching, Transparent Data Partitioning with Failover and Fail backing

- Enormous Data Loads and Transaction Volumes

  Parallel Programming, MapReduce Techniques, NoSQL Options, & the Data GRID

Let's look at various Technology Options, Tools and Framework Alternatives ?

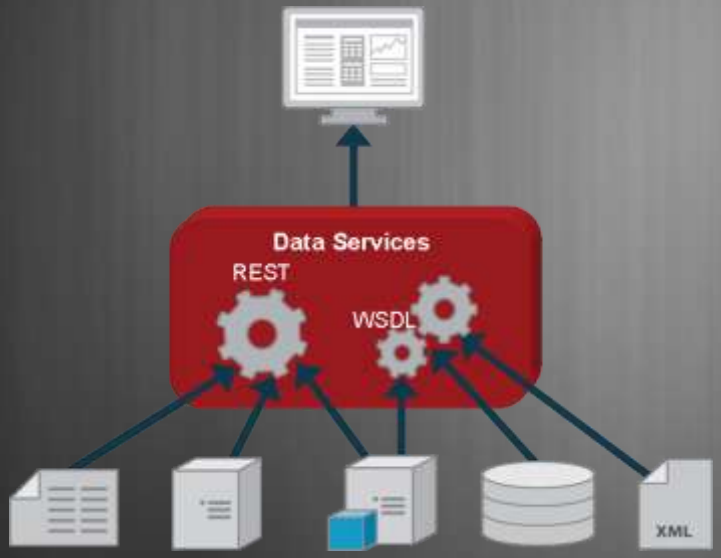
An Integration Platform

Data Services & Service Oriented Architecture

- **Service Oriented Architecture**
  - An Architecture Style that provides a Technology agnostic way of Integrating Disparate Applications within an Enterprise by improving Reuse and eliminating duplication of Application Logic
  - Componentization & Servicization is what typically happens here.
  - Types of Components, in the order of increasing consolidation:
    - **Data Services that provide access to data without a need for worrying the data representation or storage**
    - **Business Services that contain business logic for specific, well-defined tasks and perform business transactions via data services**
    - **Business Processes that coordinate multiple business services within the context of a workflow.**

An Integration Platform

Data Services & Service Oriented Architecture



Data services provide agility and reuse

Data Services are essentially the SOA-equivalent of the **Data Access Object pattern**

- **Implementation Options for Service Oriented Data Access:**

  The heart of any enterprise application is data. Applications provide the ability to view, sort, filter, edit, create, and delete data

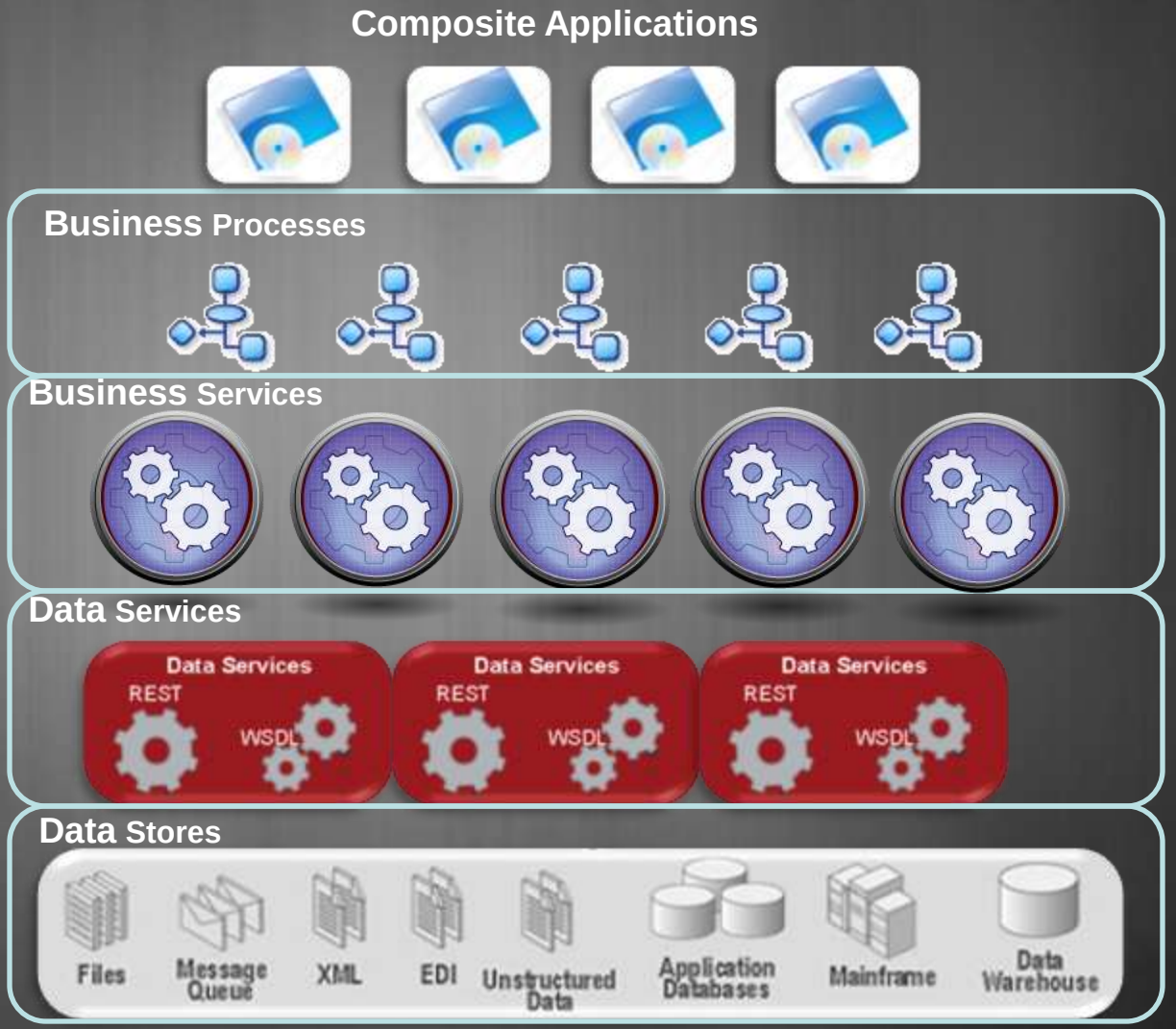
  - Expose a Database as a Service (above an ORM Layer)
  - Expose a Stored Procedure as a Service
  - Expose a DAO as a Service
  - Wrap an existing business object (EJB or POJO) with a web service

  The term "Service" here is corresponds to a Web Service

Broadridge®

Tools & Framework Options:

1. Any Java Application / Web Server
2. Web Service Engine like Axis / Metro
3. Specific Data Service Frameworks like
   1. WSO2 Data Services with any Application Server
   2. Composite Software Data Services etc …
   3. JBoss Enterprise Data Services Platform



Are we missing something ?

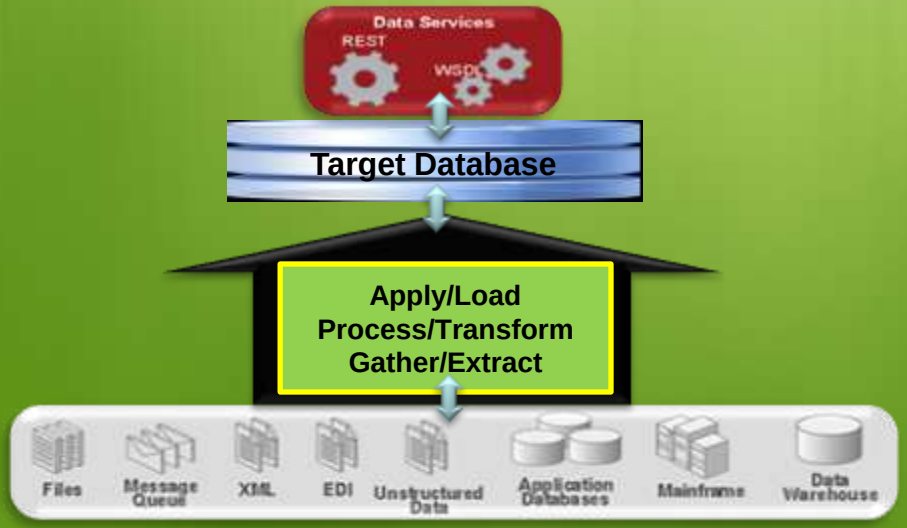
Yes, What about the issue of Data Duplication?

Broadridge®

Data Integration Techniques

- Going beyond a Simple Data Access …
- Data Integration Patterns:
  - Data Federation Pattern (also referred as Virtualization)

  - Data Aggregation Pattern (also referred as Consolidation)

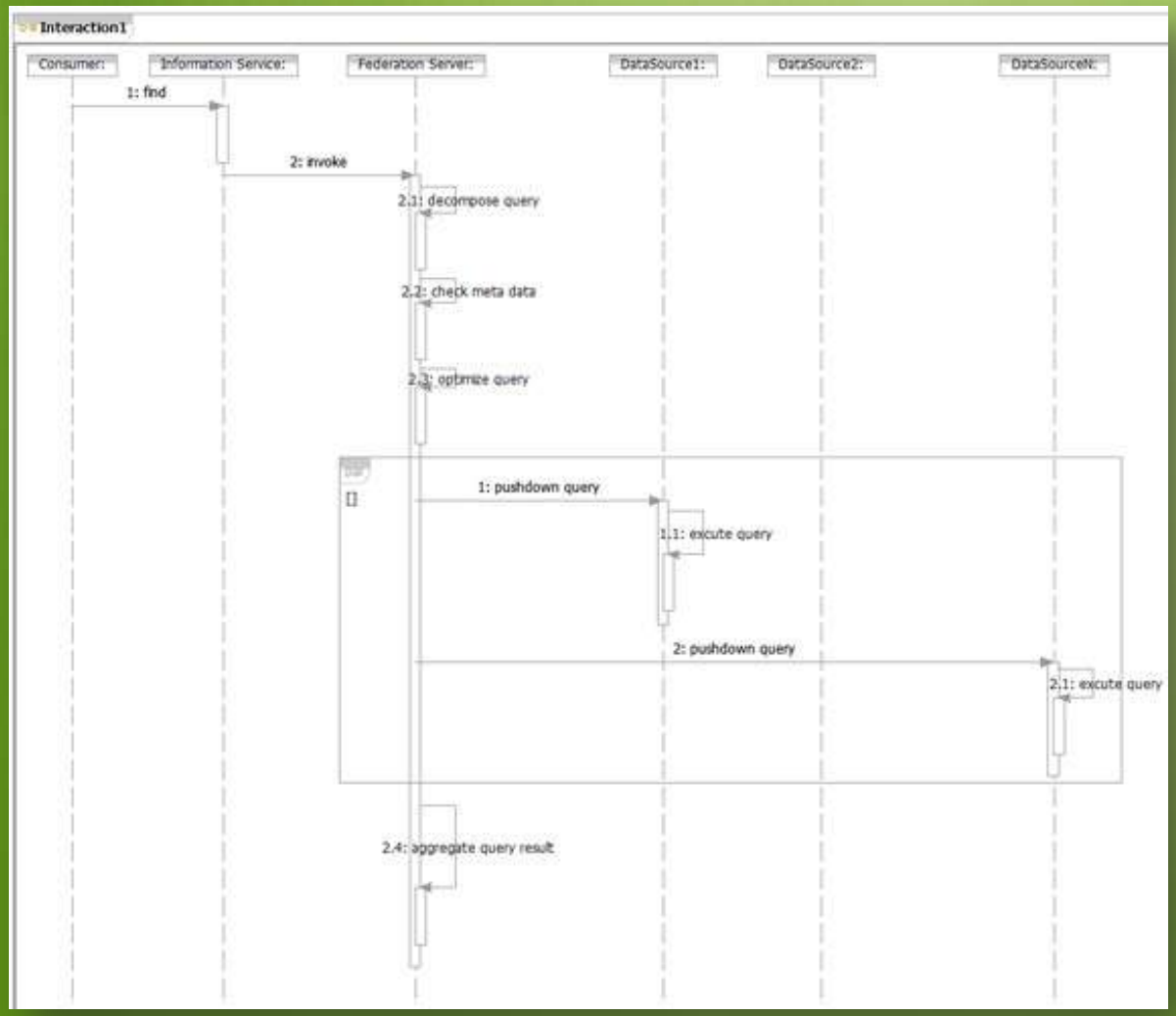


Broadridge

Data Services using Federation Technique

- Context
  - Need for unified view of data that often involves the integration of a bewildering array of disparate backend sources, and services
- Value
  - Transparency of underlying heterogeneity
  - Time-to-market advantage
  - Reduced Development & maintenance costs
  - Performance Advantage
  - Reusability Advantage
  - Improved Governance
- Scope
  - Effectively join and process information from heterogeneous sources
  - Receive a query, transform it using complex query optimizing algorithms, creating a series of sub-operations that are invoked on the base system
  - Finally assembling the results and returning to the requesting system.

The Data Federation Pattern



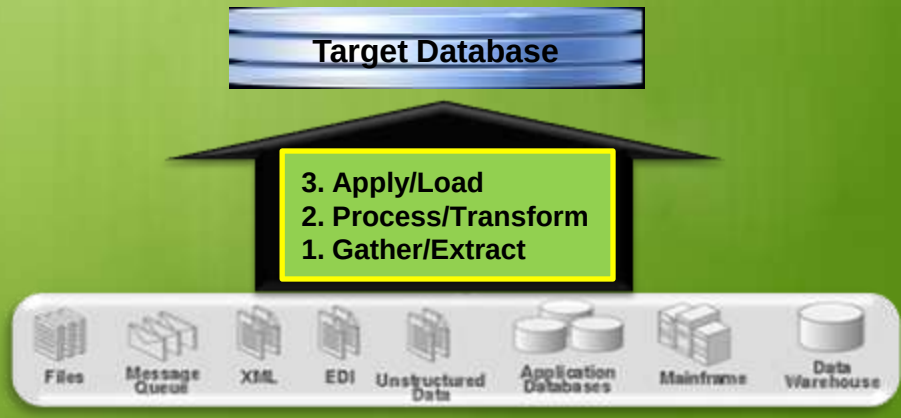
The functionality of the data federation pattern can be implemented using either database-related technologies such as optimizer or compensation, or by home-grown applications. Due to the complexity of query optimization over heterogeneous sources, it is an industry best practice to use a data federation implementation that leverages query optimization technology as provided by most database management systems

Broadridge

Data Services Using Consolidation Techniques

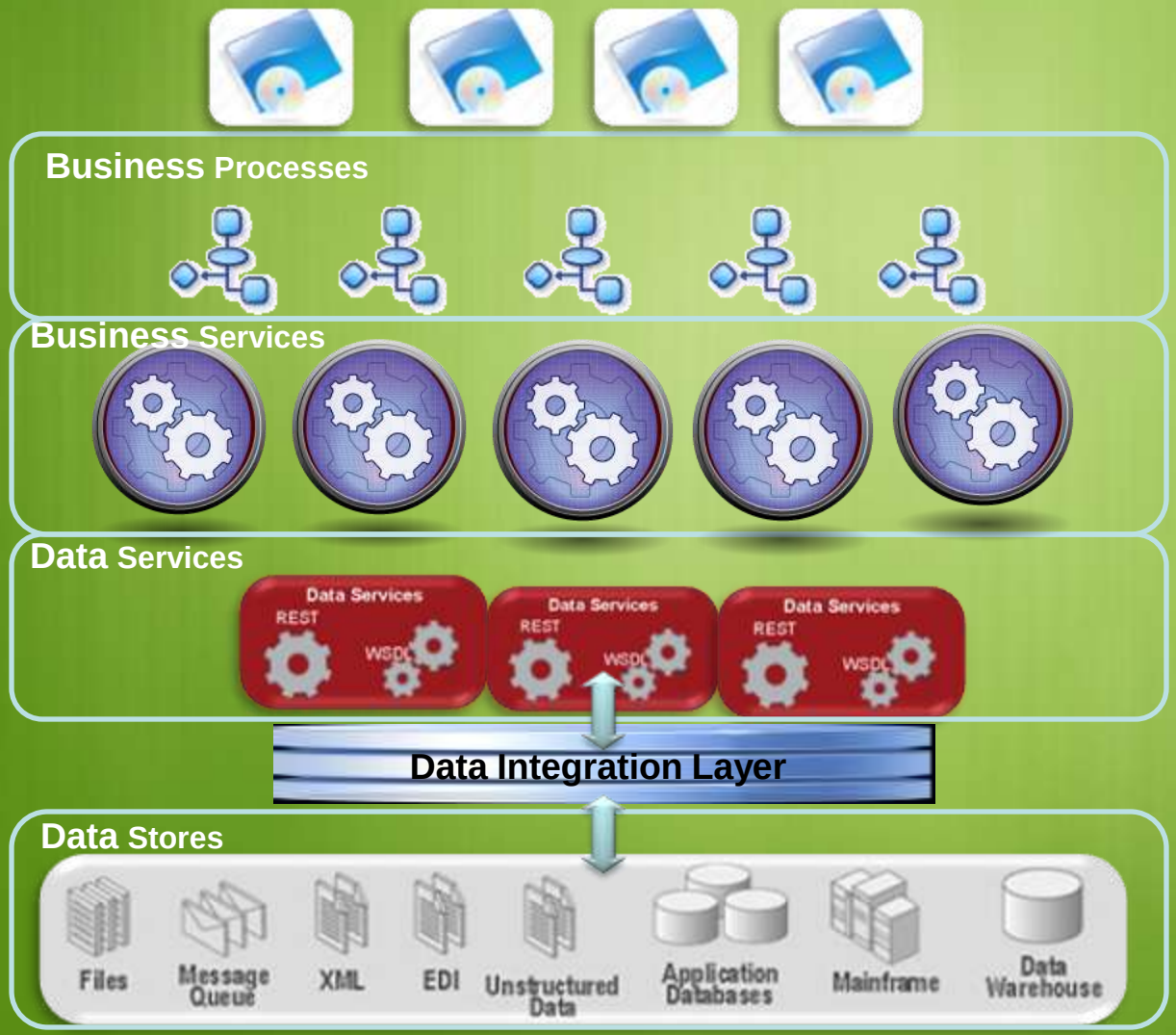
- Context
  - Integrate Information from Sources that are highly Heterogeneous in nature
  - Need for extensive transformation logic to resolve data conflicts
  - Need for Data event publishing
- Value
  - Transparency
  - Reusability
  - Improved Governance
  - Additionally, Single version of truth – high quality that involves resolving of data conflicts

- Scope
  - Phase 1: Data Aggregation Server gathers / extracts data from the data sources
  - Phase 2: Source Data is integrated and transformed to conform to target model
  - Phase 3: Apply the transformed data to the target data store



Broadridge

Data Federation Tools / Frameworks



- IBM
  - Websphere Information Integrator
  - Websphere Information Integrator Classic Federation
  - Websphere Information Services Director
  - Websphere DataStage
- Composite Software Data Federation Service
- Progress Software ObjectStore
- Oracle Data Integrator

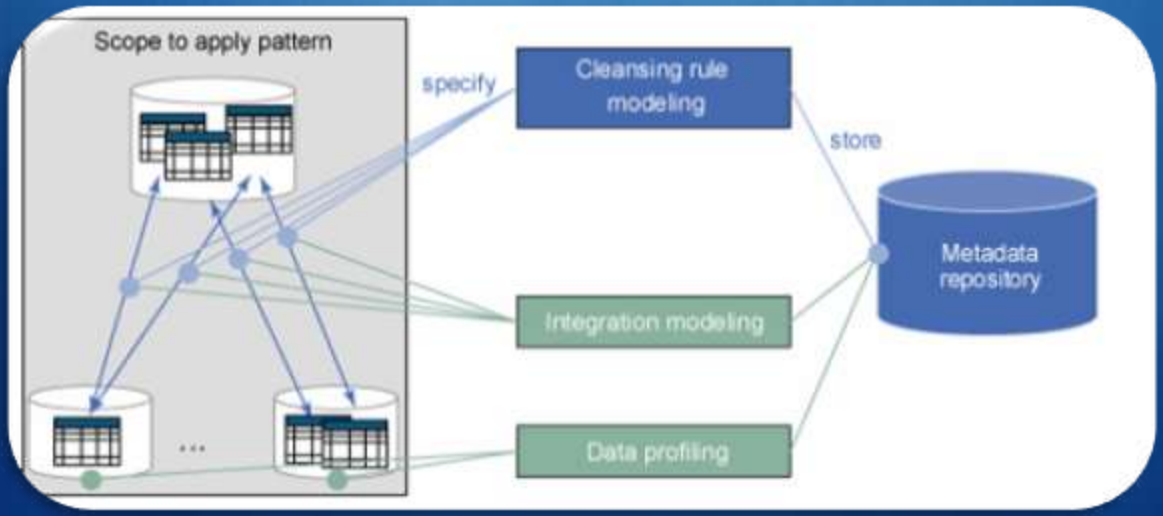
Broadridge

Data Availability & Reliability

- **SOA Demands High Data Reliability and Availability**
- Solutions / Options to achieve reliability & availability
  - **Reliability Techniques**
    - **Data Cleansing Pattern**
    - **Master Data Management**
  - **Availability Techniques**
    - **Clustered Caching**
    - **State Management Through Virtualization**
    - **Transparent Data Partitioning**
    - **Failover and Failback**
    - **Insulation from Failures in Other Services**

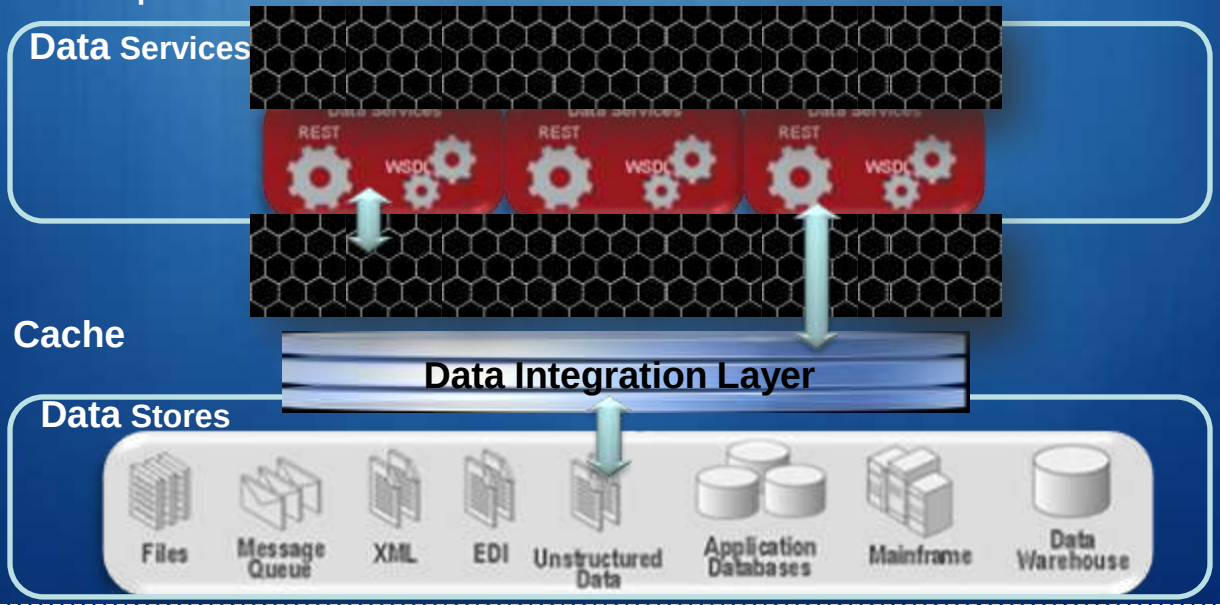
Data Reliability Option: Data Cleansing Pattern

- **Scope:**
  - **Parsing of input data and association to standard and fine-grained elements**
  - **Standardization of data**
  - **Matching and de-duplication of data entries**
  - **Survivorship of the correct information**



Clustered Caching ensures Availability and thus reliability

- In-memory data management solution is what is required
- It makes sharing and managing data in a cluster as simple as on a single server. It accomplishes this by coordinating updates to the data by using clusterwide concurrency control, replicating and distributing data modifications across the cluster by using the highest-performing clustered protocol available, and delivering notifications of data modifications to any servers that request them.



Broadridge®

Data Availability & Reliability : Other Techniques

- State Management Through Virtualization
- Failover and Failback
- Insulation from Failures in Other Service
- Transparent Data Partitioning Achieves Continuous Availability and Reliability

Tool Options

1. IBM Websphere Quality Stage
2. Many MDM Vendors, IBM, Oracle, Informatica etc …
3. Oracle Coherence
4. MemCached
5. EHCache (Open Source)

Performance

(*Scalability, Throughput, and Points of Congestion*)

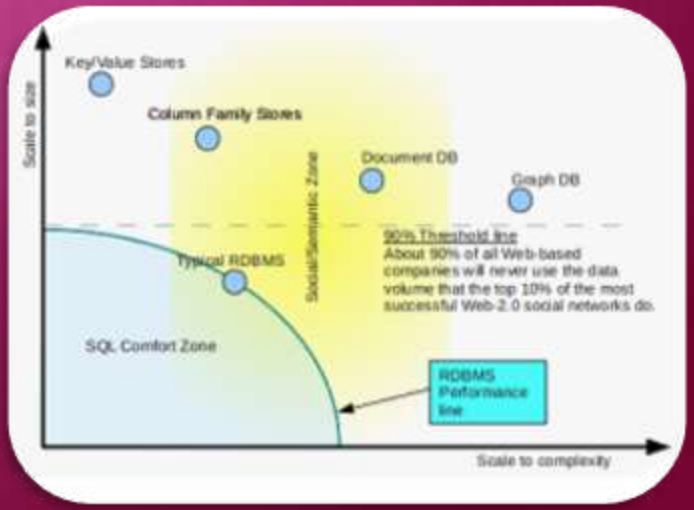
- **Bottlenecks**
  - ***Shared intermediary services*** - Such services perform common integration tasks such as data transformation, content based routing, and filtering.
  - ***The services themselves*** - That is, application code exposed as a service and invoked by other services on the network, whether directly or through an orchestration engine.
  - ***SOA infrastructure operations***

    In most cases, the scalability bottlenecks across these SOA components are caused when disk I/O, memory, or CPU saturation levels are reached.
- **Need**
  - **Large Datasets**
  - **Enormous Loads & Large Transaction Volumes handled Without Compromise**
- **Solution**
  - **Data Store alternatives like NoSQL with Parallel Processing / Programming MapReduce Techniques and**
  - **Data Grids**

NoSQL – Not Only SQL

- **What's wrong this RDBMSs ?**
  - Well, actually nothing… it just has limitations
    - They use table-based normalization approach
    - They allow versioning
    - Performance falls off as RDBMS normalizes data as the data grows.
- **NoSQL Databases**
  - Non Relational
  - Designed for distributed data stores for large data needs
  - Doesn't require fixed table schemes
  - (Usually) avoids join operations
  - Can Scale Horizontally – allows adding more nodes to storage systems
  - Types
    - Key-Value Stores
    - Column Family Stores
    - Document Databases
    - Graph Databases

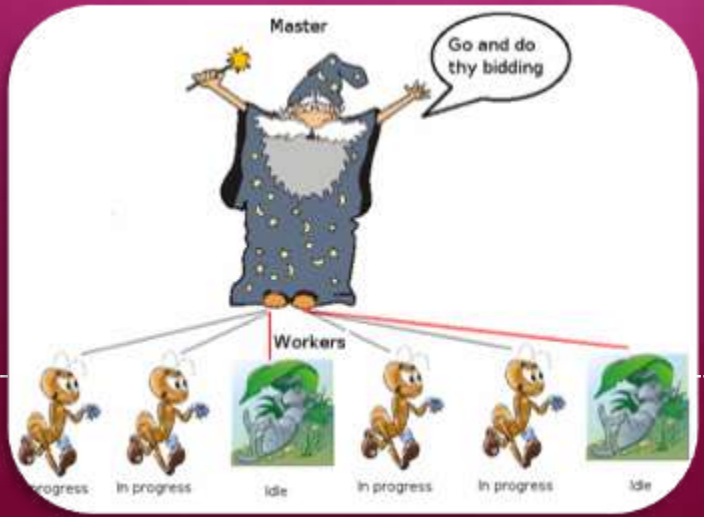
- **NoSQL Usecases**
  - **Logging/Archiving**
  - **Social Computing Insight**
  - **External Data Feed Integration**
  - **Front-ens order processing systems**
  - **Real-time stats & analytics**



Broadridge®

Teamed with MapReduce Technique

- **What's MapReduce ?**
  - Restricted Parallel Programming model for large clusters (User implements Map() and Reduce())
  - Parallel Computing Framework
    - Libraries take care of EVERYTHING else
      - Parallelization
      - Fault Tolerance
      - Data Distribution
      - Load Balancing
  - Map and Reduce (borrowed from Lisp)
    - Map() – Processes a Key-Value Pair that produces immediate Key-Value Pairs
    - Reduce() – Marge all intermediate values associated with the same key



Broadridge®

MapReduce Execution Model

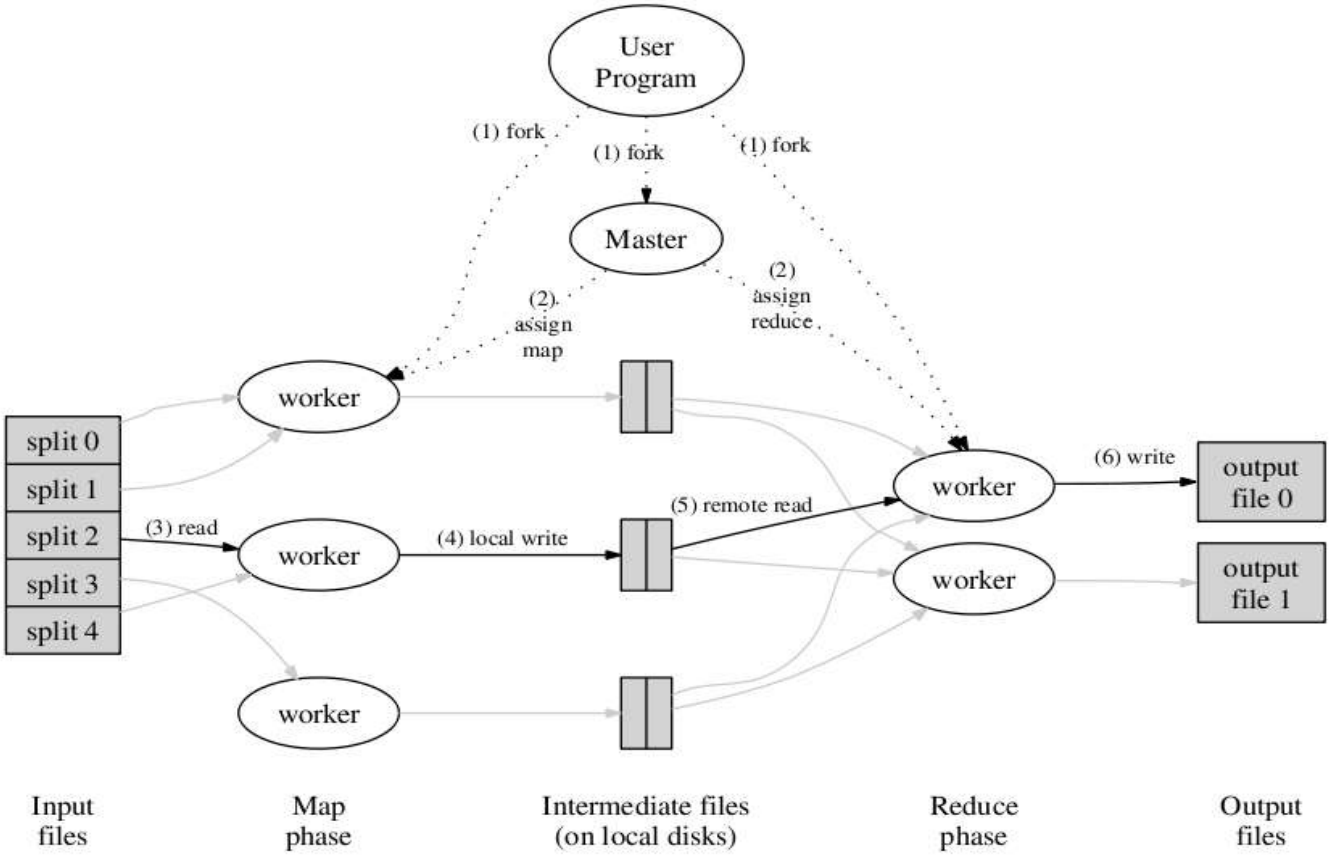
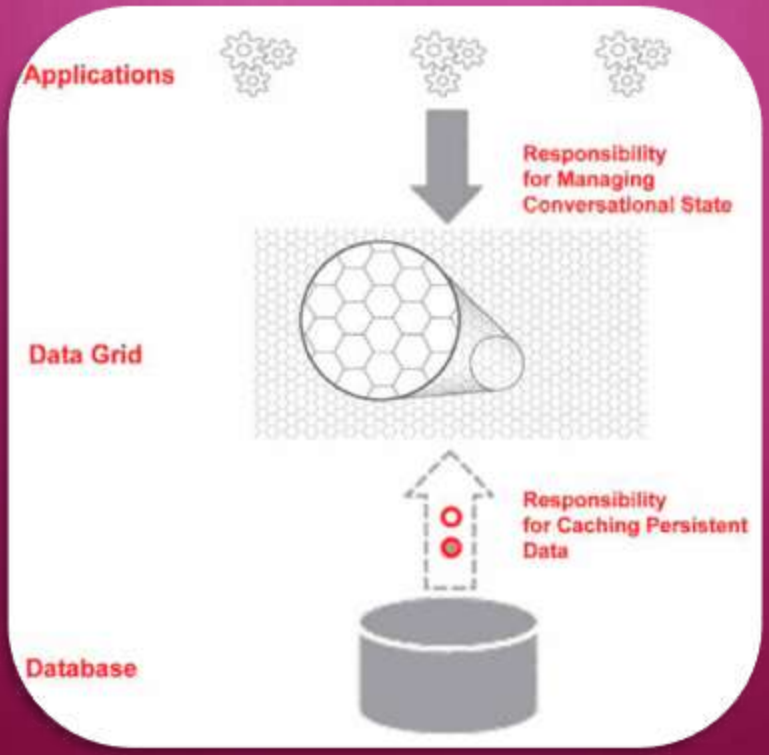


Figure 1: Execution overview

SOA with Data Grid Option
(Service Oriented Computing)

- An SOA grid transparently solves many of the difficult problems related to achieving high availability, reliability, scalability, and performance in a distributed environment



Tools/Products/Frameworks

- NoSQL Options
  - HBase
  - Cassandra
  - CoucheDB
  - MongoDB
- MapReduce Options
  - Hadoop
  - Amazon Elastic MapReduce
  - GreenPlum MapReduce
- Grid Options
  - IBM Globus
  - Sun Grid Engine
  - Oracle Coherence

Q & A

- Sunila.Gollapudi@broadridge.com