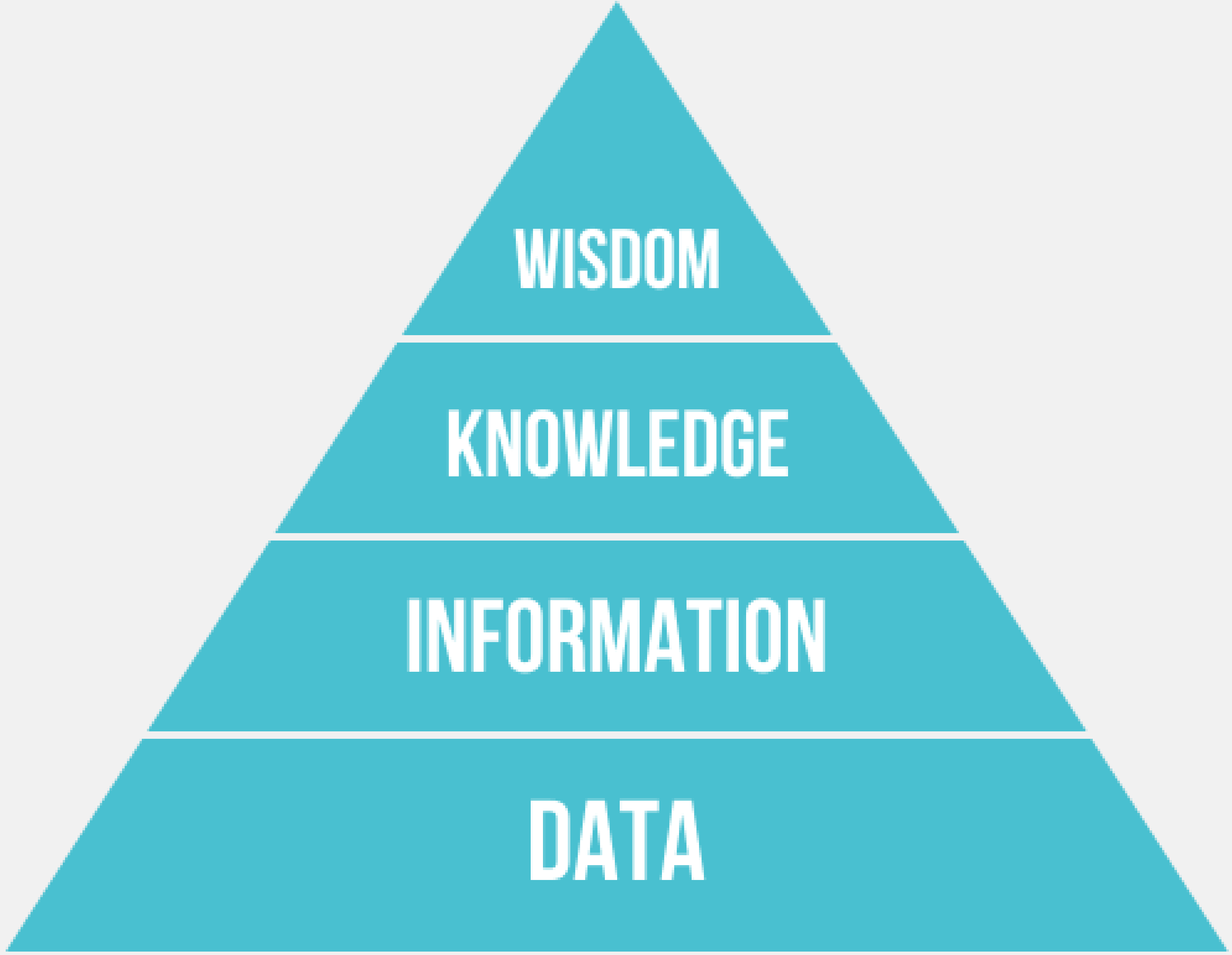
Enlighten your Data

Satish Kale
Senior Solutions Architect
June 2015

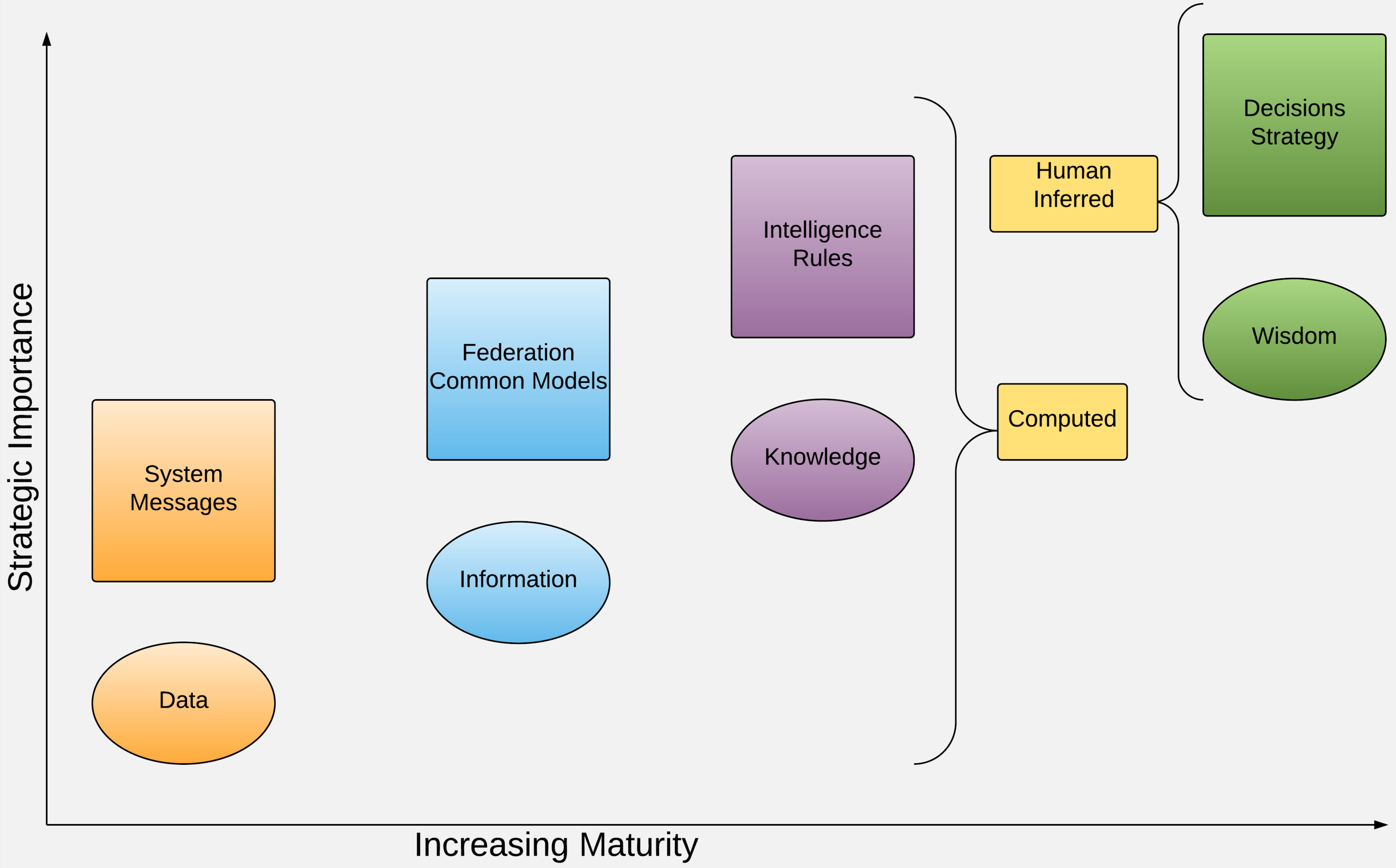
Enlighten

- To have greater knowledge or understanding
- Provide better insight
- Have a great revelation
- Remove confusion, elucidate

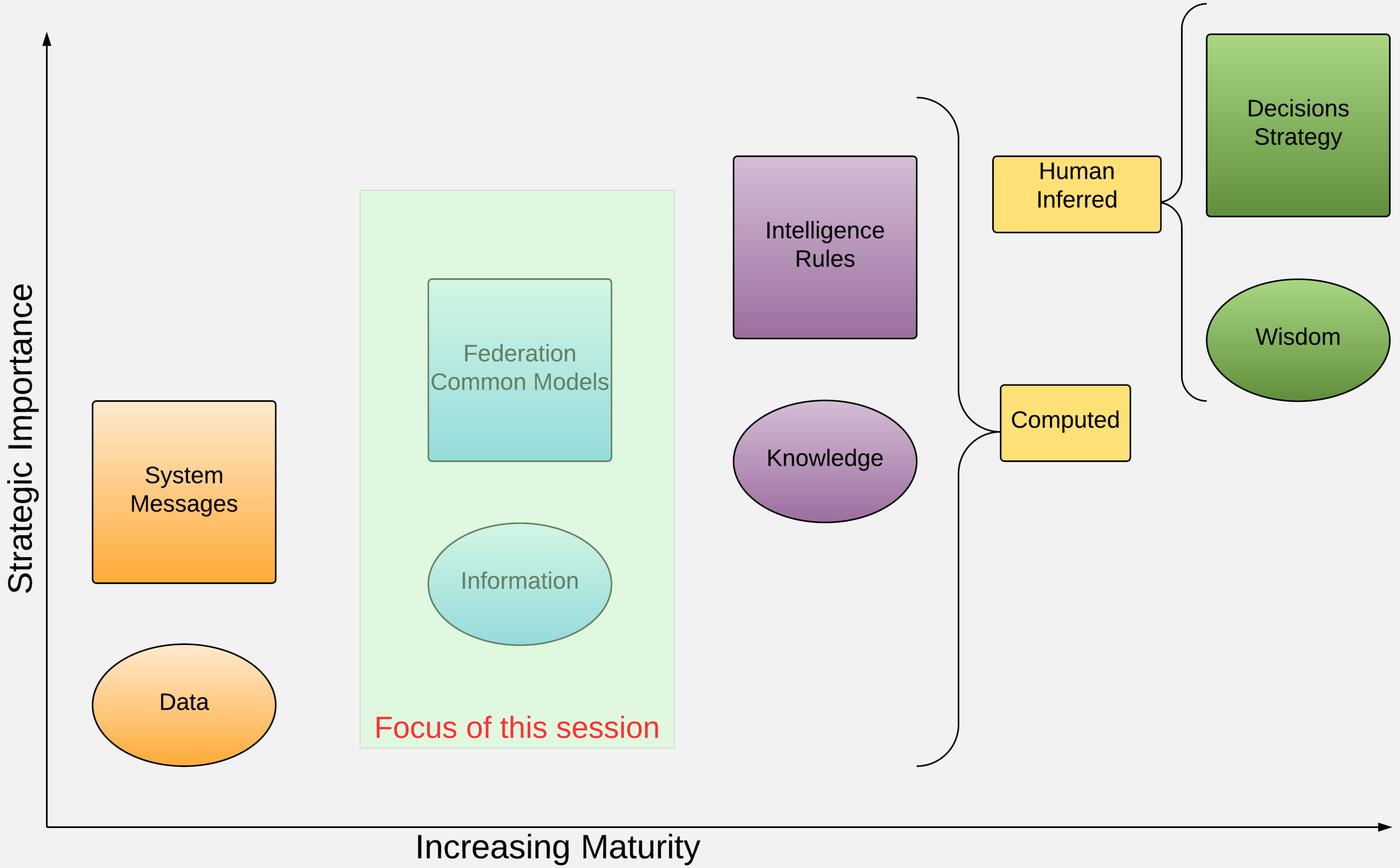
The DIKW Pyramid



The Path to Enlightenment



The Path to Enlightenment



Strategic Importance

Data

System Messages

Information

Federation Common Models

Focus of this session

Knowledge

Intelligence Rules

Computed

Human Inferred

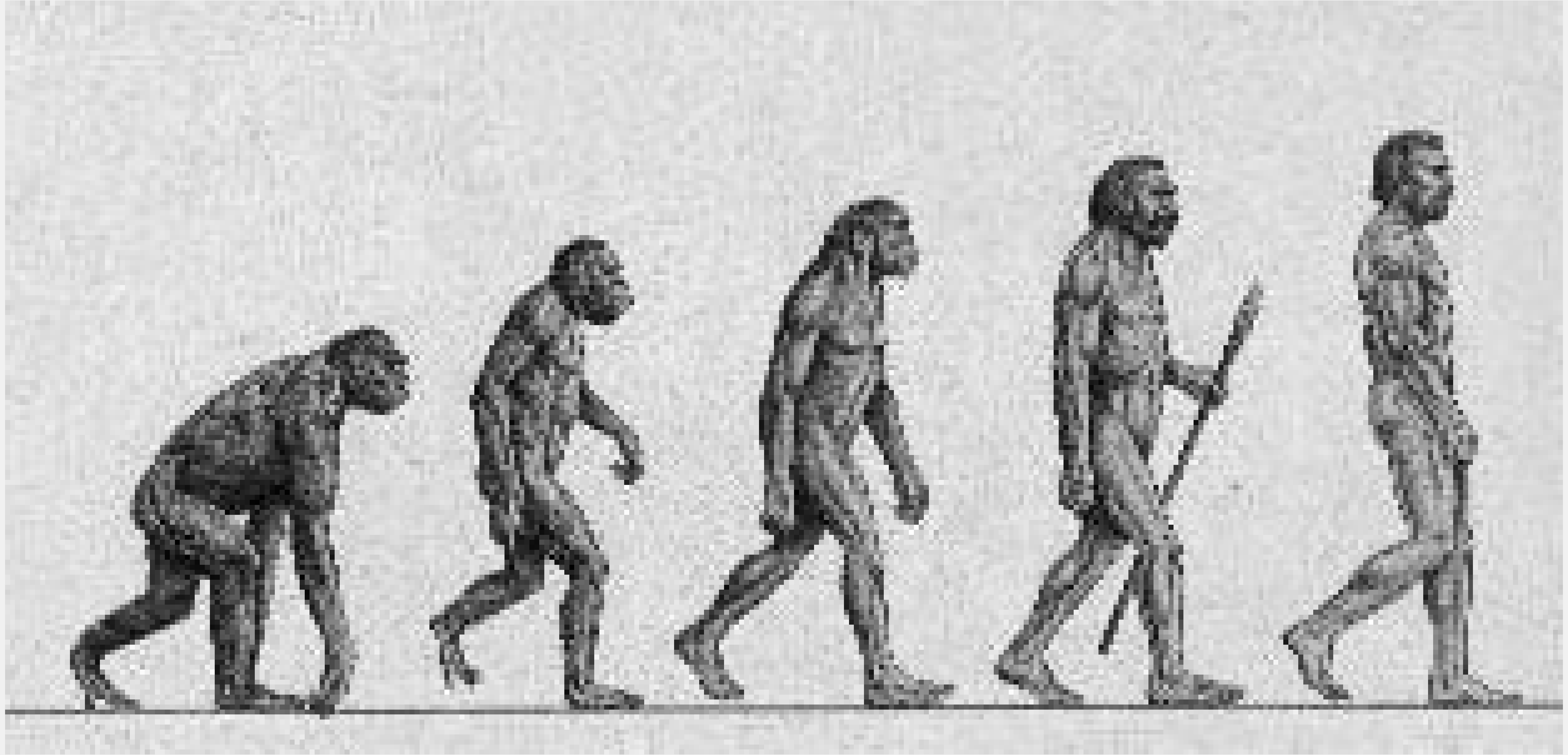
Wisdom

Decisions Strategy

Increasing Maturity

Evolve

Evolve - From DATA to INFORMATION



From DATA to INFORMATION

- Understand the differences
- Understand the challenges
- Devise solution

Differences between Data and Information

Data

Information

Differences between Data and Information

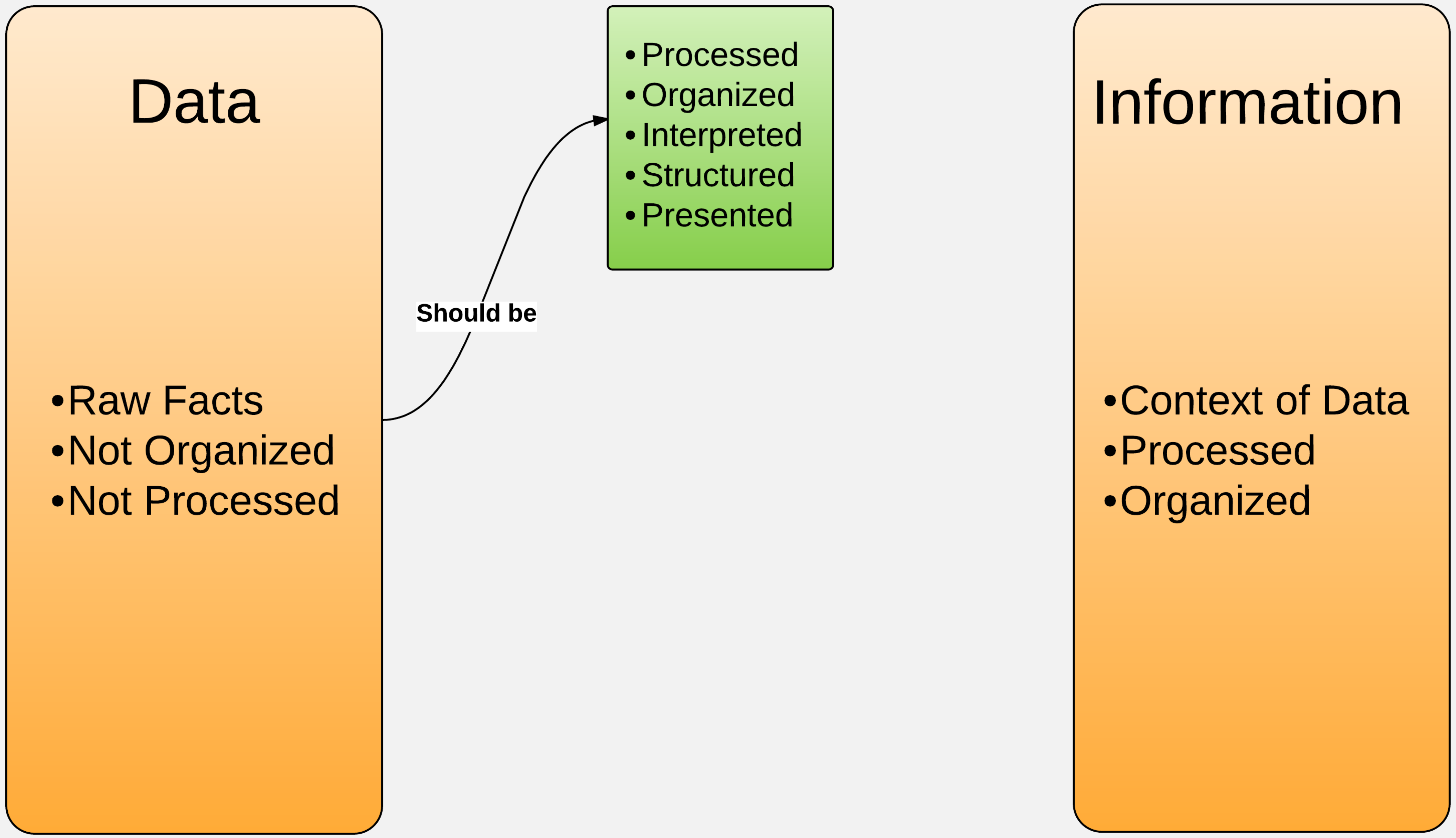
Data

- Raw Facts
- Not Organized
- Not Processed

Information

- Context of Data
- Processed
- Organized

Differences between Data and Information



Data

- Raw Facts
- Not Organized
- Not Processed

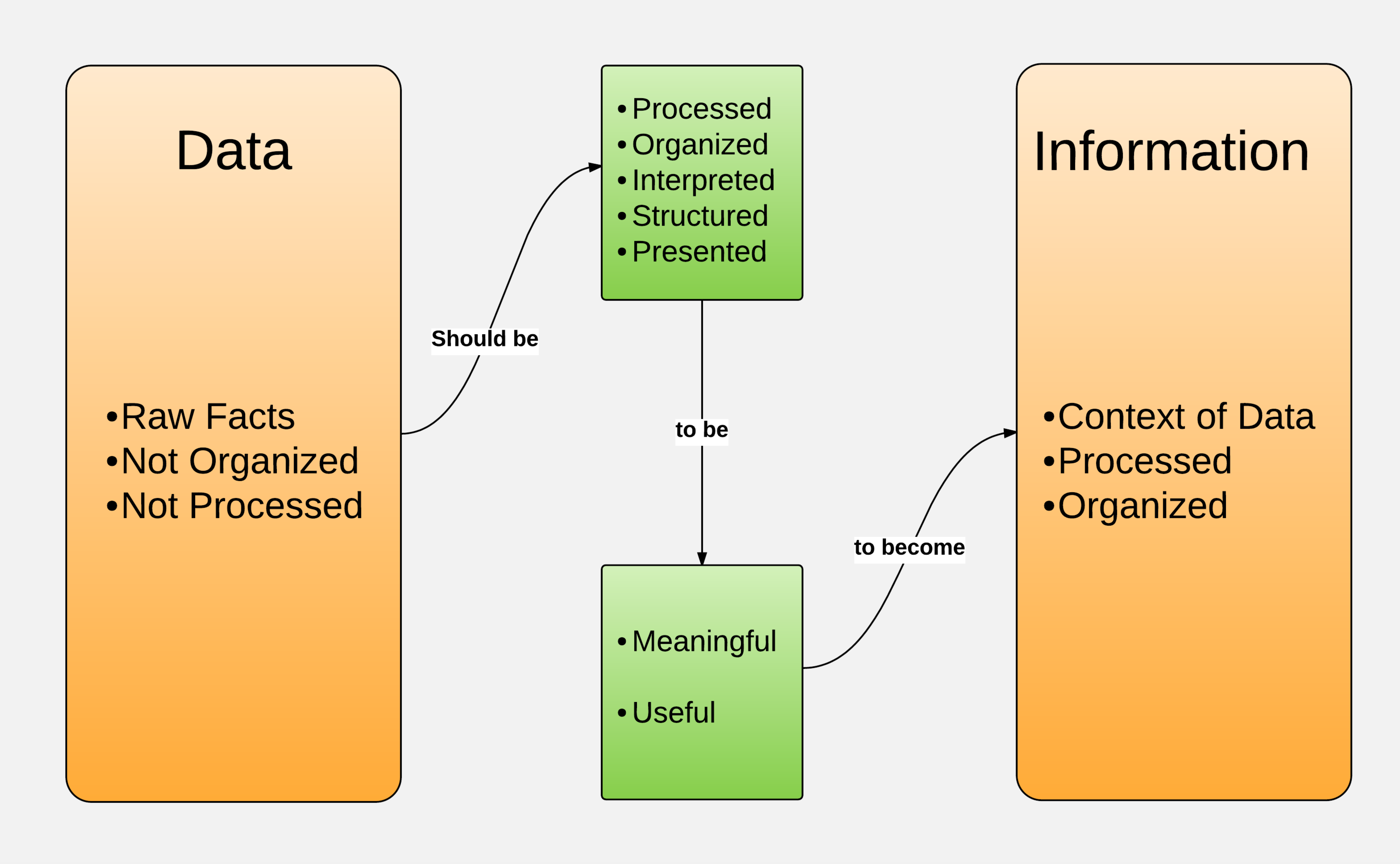
Should be

- Processed
- Organized
- Interpreted
- Structured
- Presented

Information

- Context of Data
- Processed
- Organized

Differences between Data and Information



Data

- Raw Facts
- Not Organized
- Not Processed

Should be

- Processed
- Organized
- Interpreted
- Structured
- Presented

to be

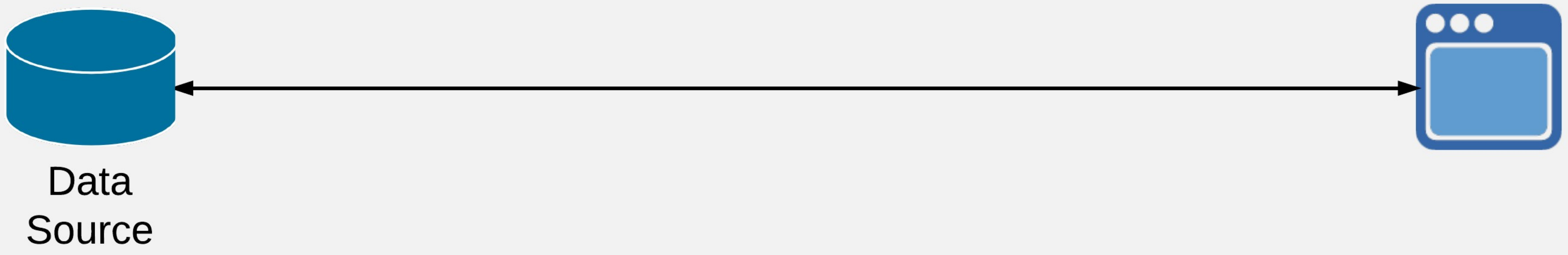
- Meaningful
- Useful

to become

Information

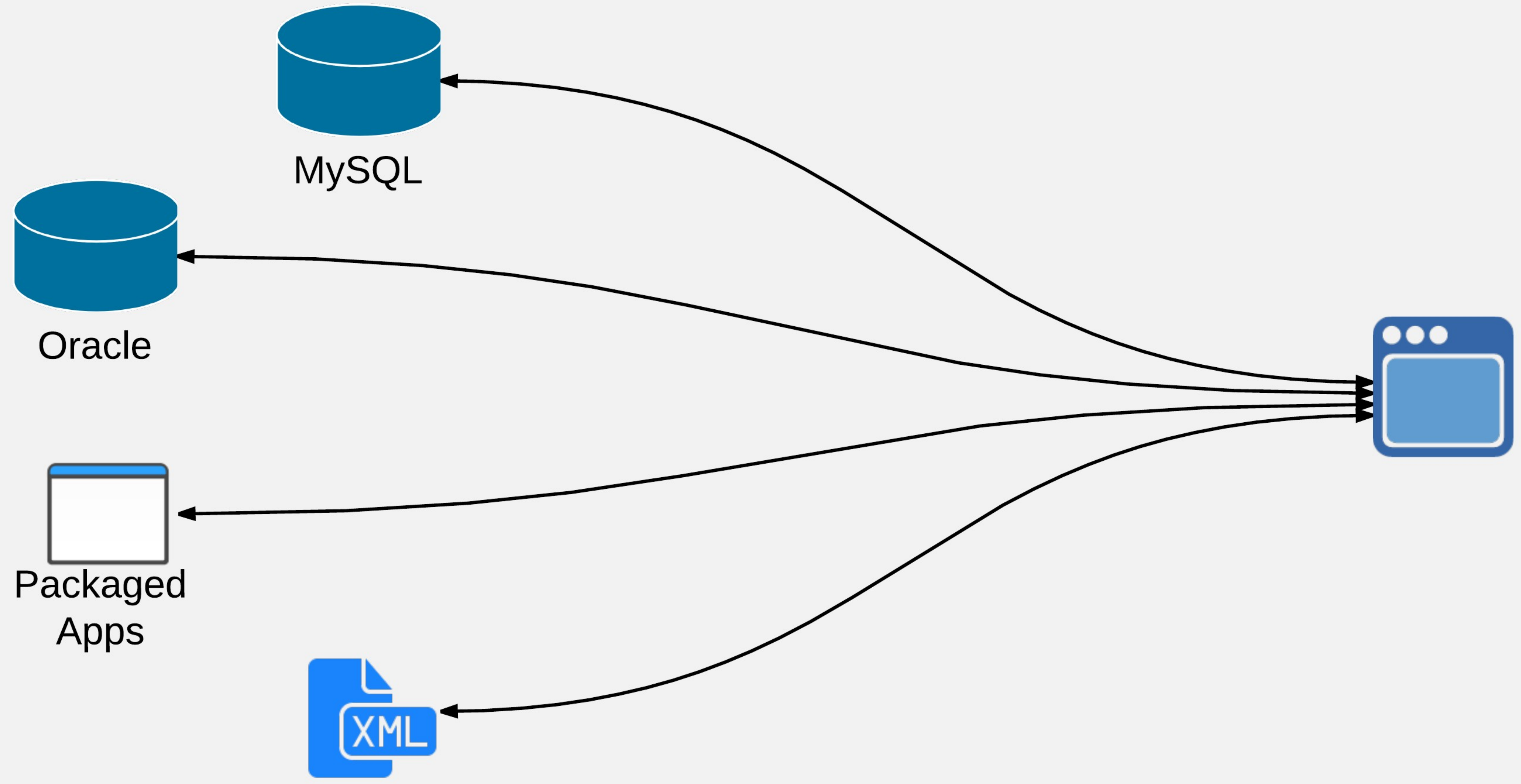
- Context of Data
- Processed
- Organized

Production – Consumption



This is easy

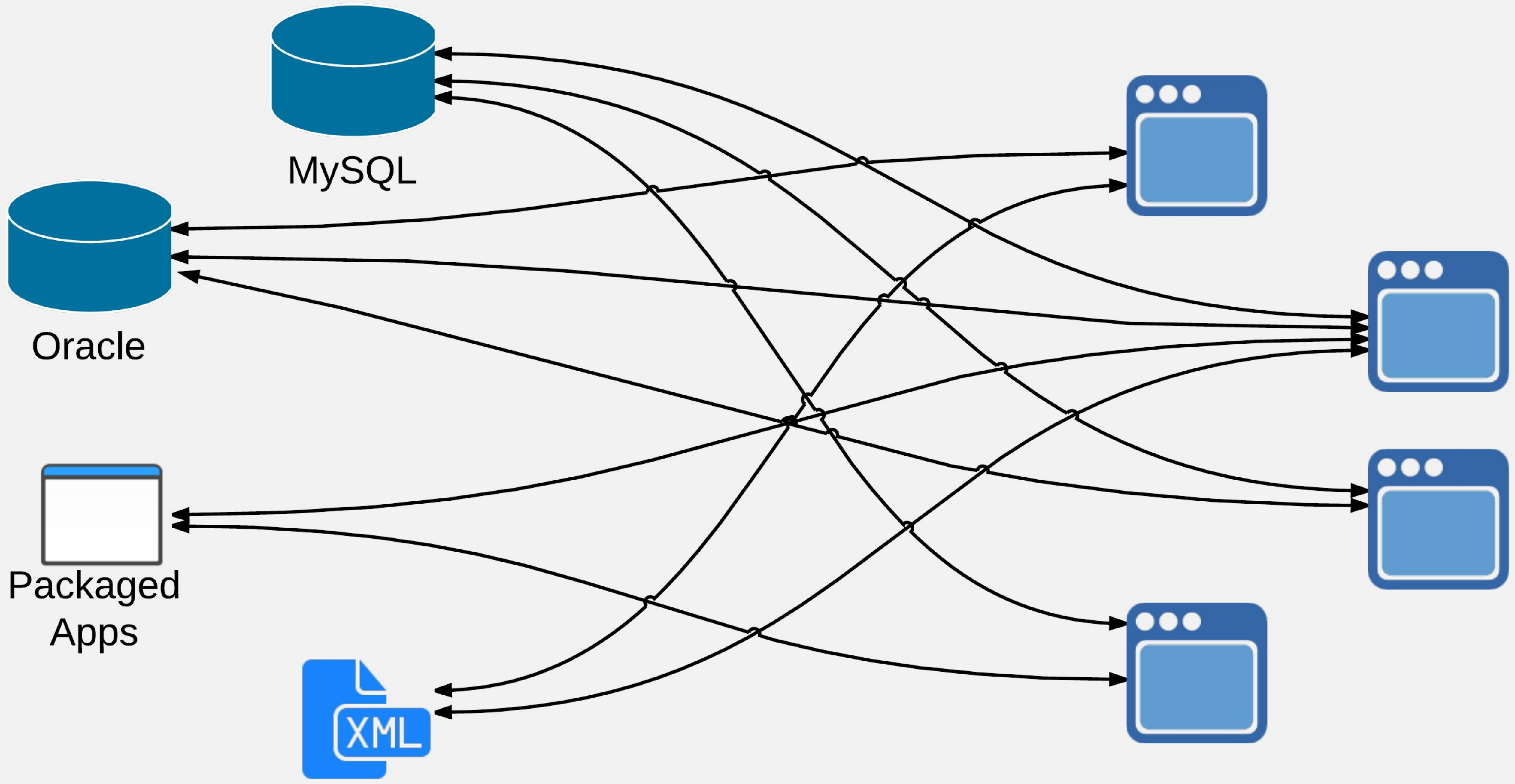
Production – Consumption



Reality is more complex.

Data is scattered all over.

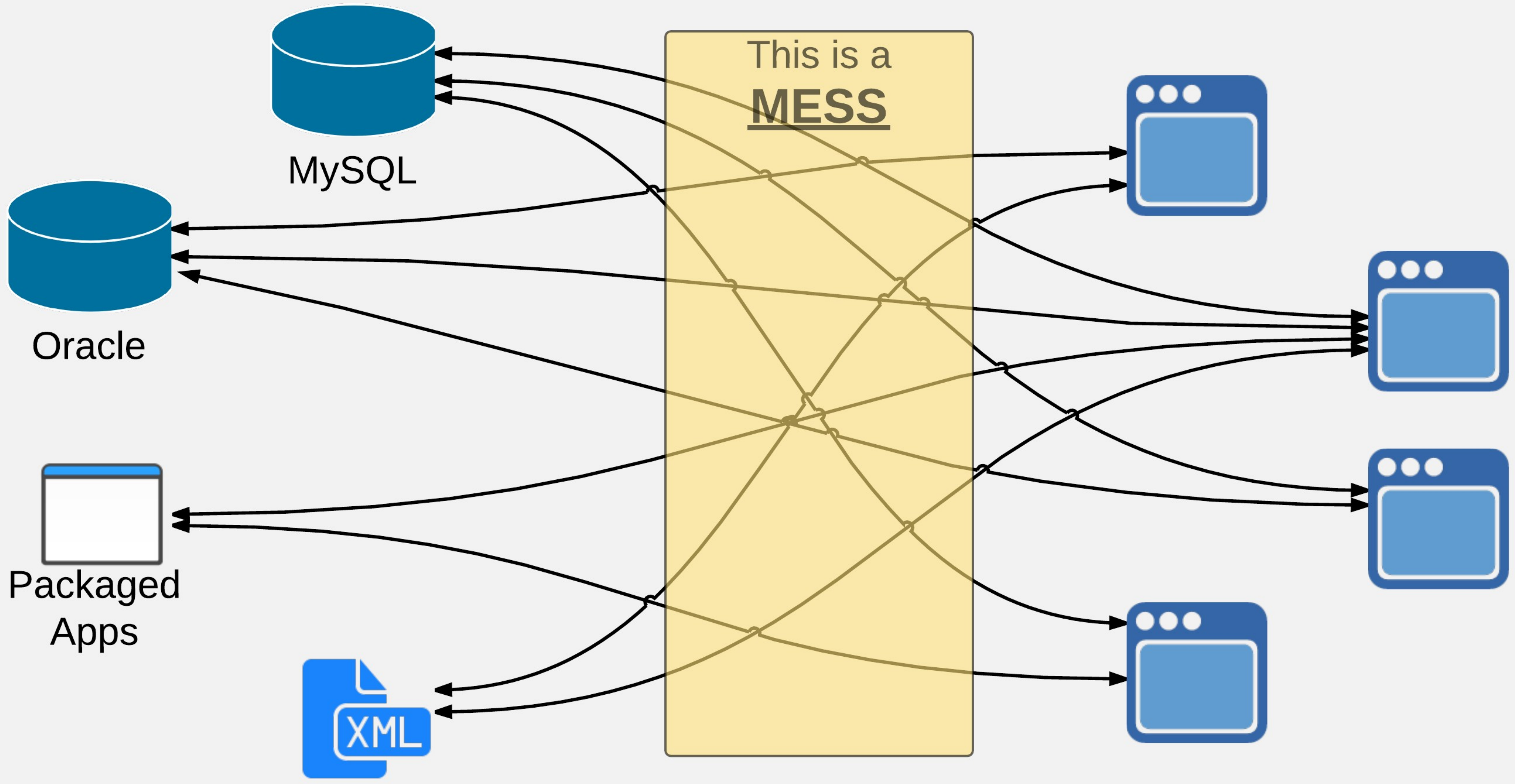
Production – Consumption



Even more complex.

There are multiple clients.

Problem : Production – Consumption



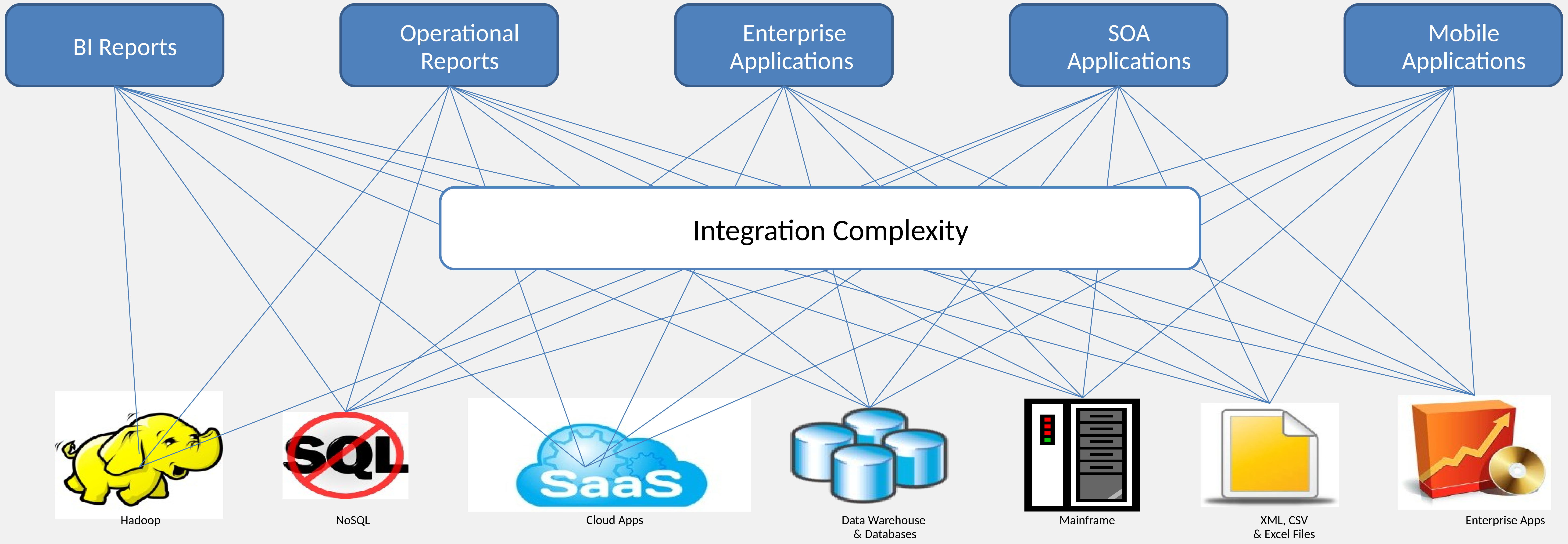
Data needs to be located, fetched, combined from various disparate sources

And made available to various clients, interfaces and formats.

Data Challenges Getting Bigger - Big Data, Cloud, and Mobile

Existing Data Integration approaches are not sufficient

- Extracting and moving data adds latency and cost
- Every project solves data access and integration in a different way
- Solutions are tightly coupled to data sources
- Poor flexibility and agility



BI Reports

Operational Reports

Enterprise Applications

SOA Applications

Mobile Applications

Integration Complexity

Hadoop

NoSQL

Cloud Apps

Data Warehouse & Databases

Mainframe

XML, CSV & Excel Files

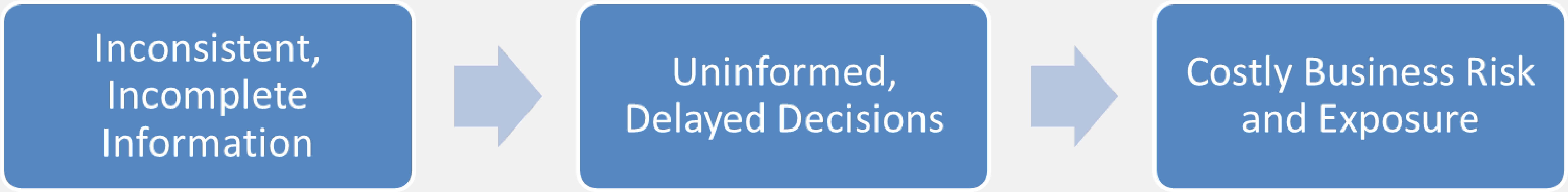
Enterprise Apps

Constant Change

How to align?

Siloed & Complex

Consider...



Inconsistent, Incomplete Information

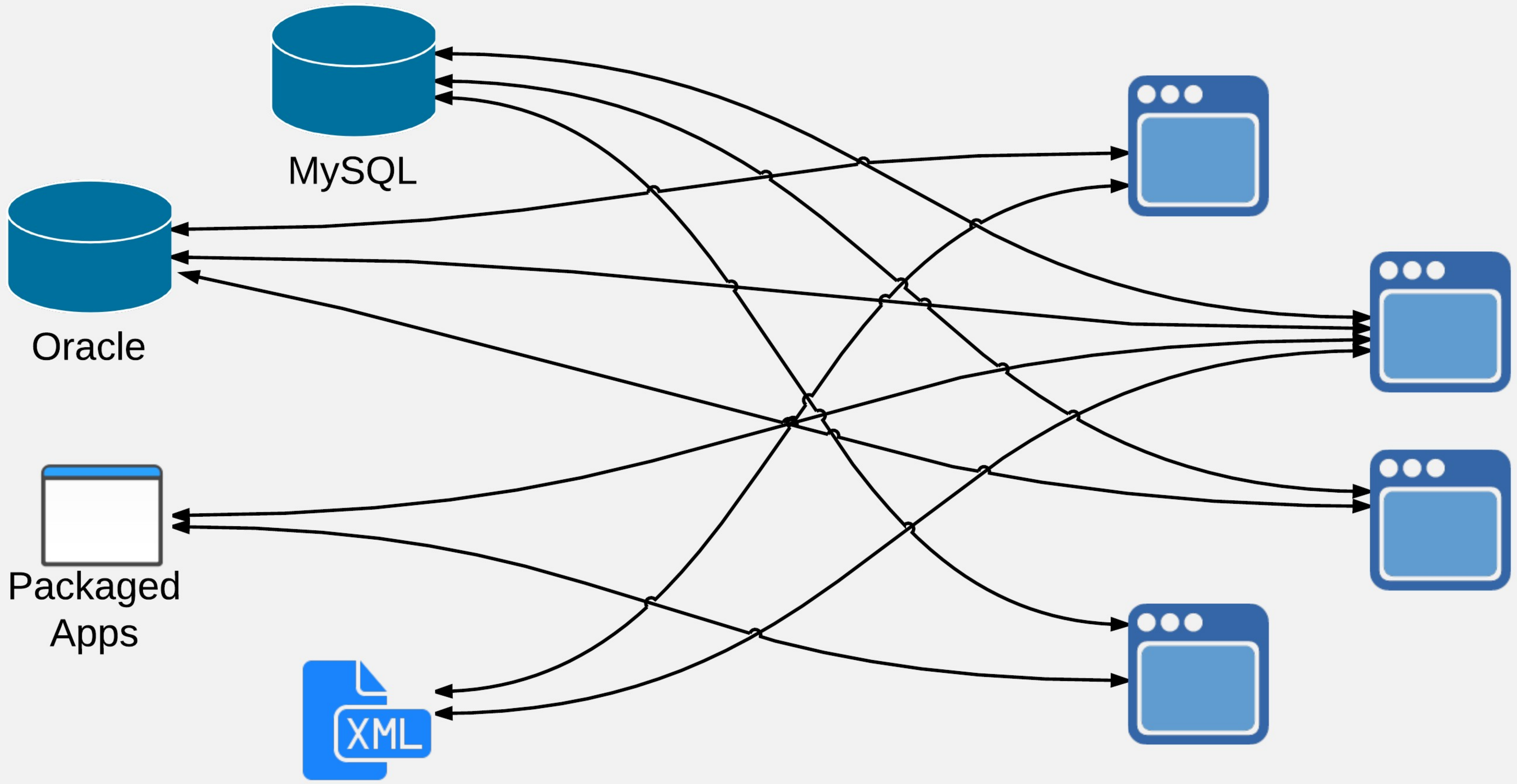
Uninformed, Delayed Decisions

Costly Business Risk and Exposure

How would your organization change...

- If data were **readily reusable in place** rather than requiring significant effort to build new intermediary data tiers?
- If data could be repurposed **quickly** into new applications and business processes?
- If all applications and business processes could get **all of the information needed** in the form needed, where needed and when needed?

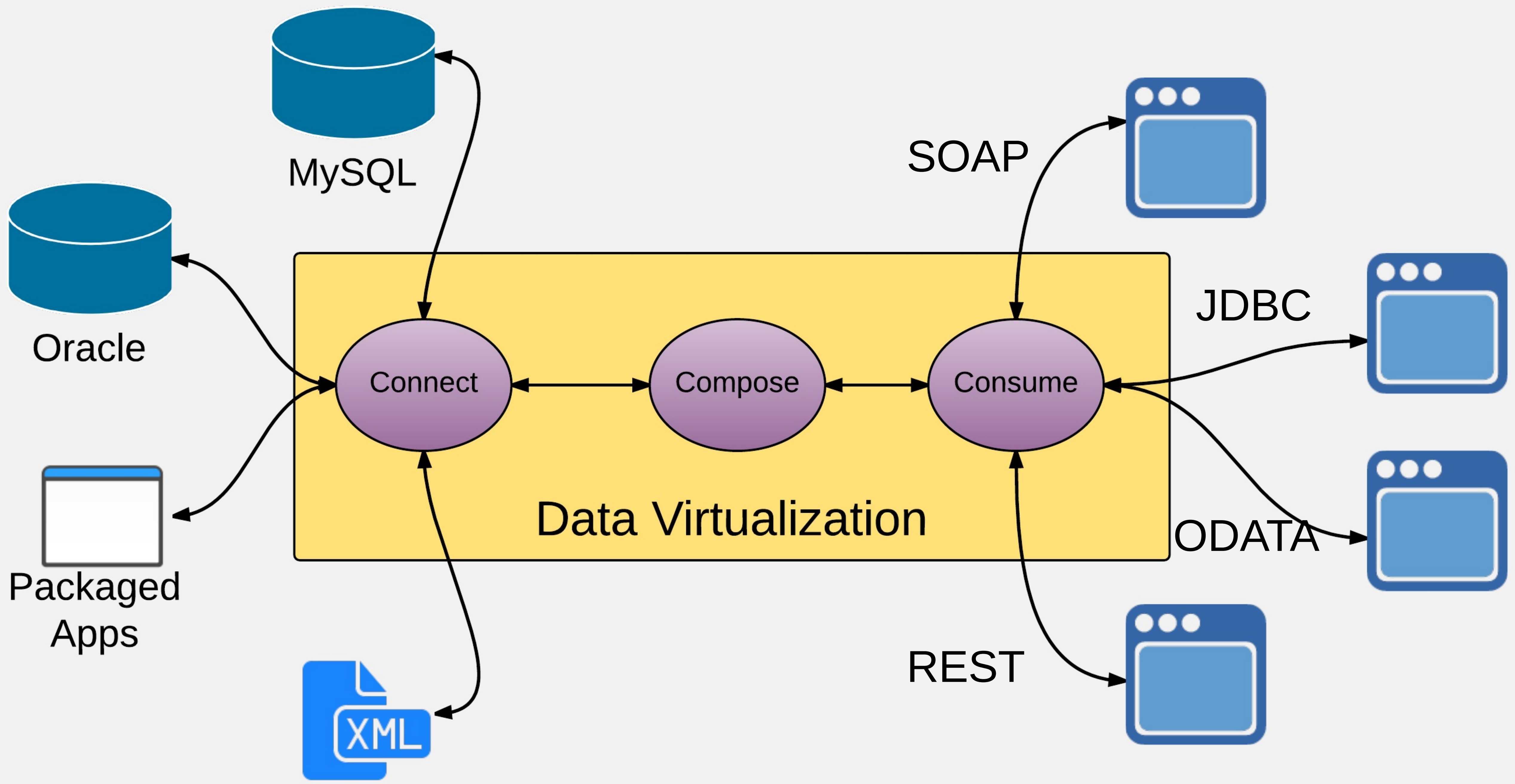
Back to the problem



Data needs to be located, fetched, combined from various disparate sources

And made available to various clients, interfaces and formats.

Solution Approach – Data Virtualization



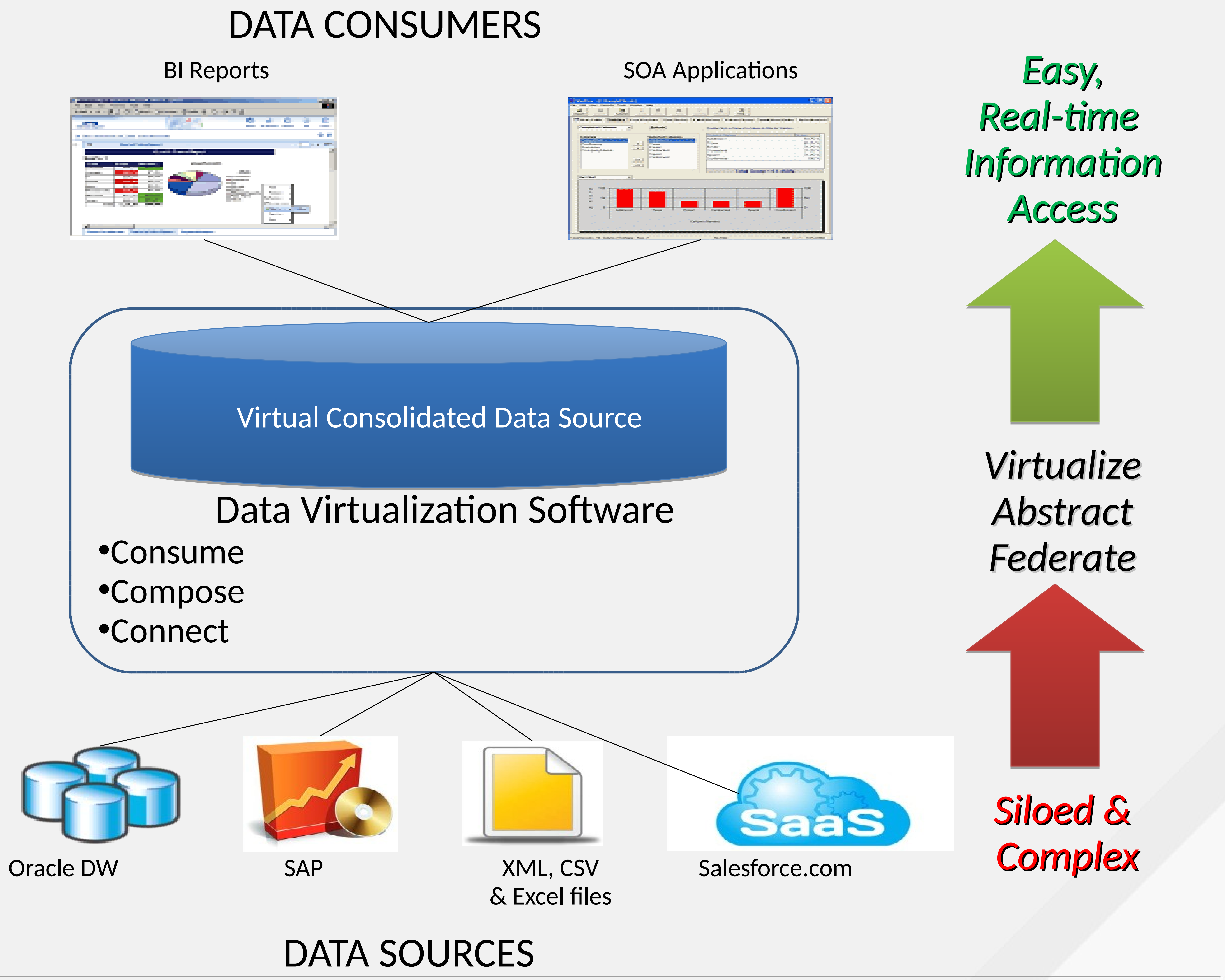
Decoupled
Choice of connectivity, accessibility
Flexible

What does Data Virtualization software do?
Turn Fragmented Data into Actionable Information

Data Virtualization software virtually unifies data spread across various disparate sources; and makes it available to applications as a single consolidated data source.

The data virtualization software implements 3 steps process to bridge data sources and data consumers:

- **Connect**: Fast access to data from diverse data sources
- **Compose**: Easily create unified virtual data models and views by combining and transforming data from multiple sources.
- **Consume**: Expose consistent information to data consumers in the right form through standard data access methods.



DATA CONSUMERS

BI Reports

SOA Applications

Virtual Consolidated Data Source

Data Virtualization Software

- Consume
- Compose
- Connect

Oracle DW

SAP

XML, CSV
& Excel files

Salesforce.com

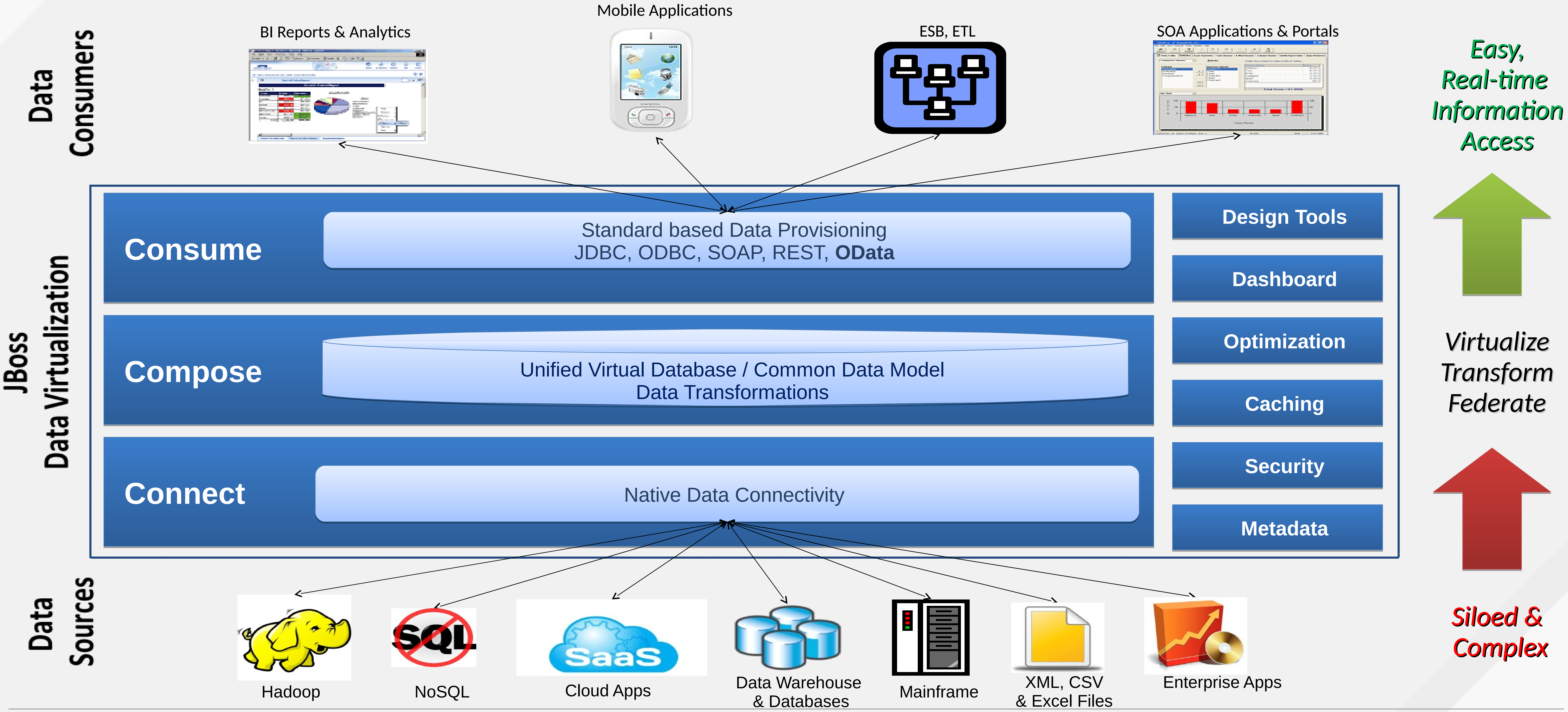
DATA SOURCES

Easy, Real-time Information Access

Virtualize Abstract Federate

Siloed & Complex

Turn Siloed Data into Actionable Information



Data Consumers

BI Reports & Analytics

Mobile Applications

ESB, ETL

SOA Applications & Portals

Easy, Real-time Information Access

JBoss Data Virtualization

Consume

Standard based Data Provisioning
JDBC, ODBC, SOAP, REST, **OData**

Compose

Unified Virtual Database / Common Data Model
Data Transformations

Connect

Native Data Connectivity

Design Tools

Dashboard

Optimization

Caching

Security

Metadata

Virtualize
Transform
Federate

Data Sources

Hadoop

NoSQL

Cloud Apps

Data Warehouse
& Databases

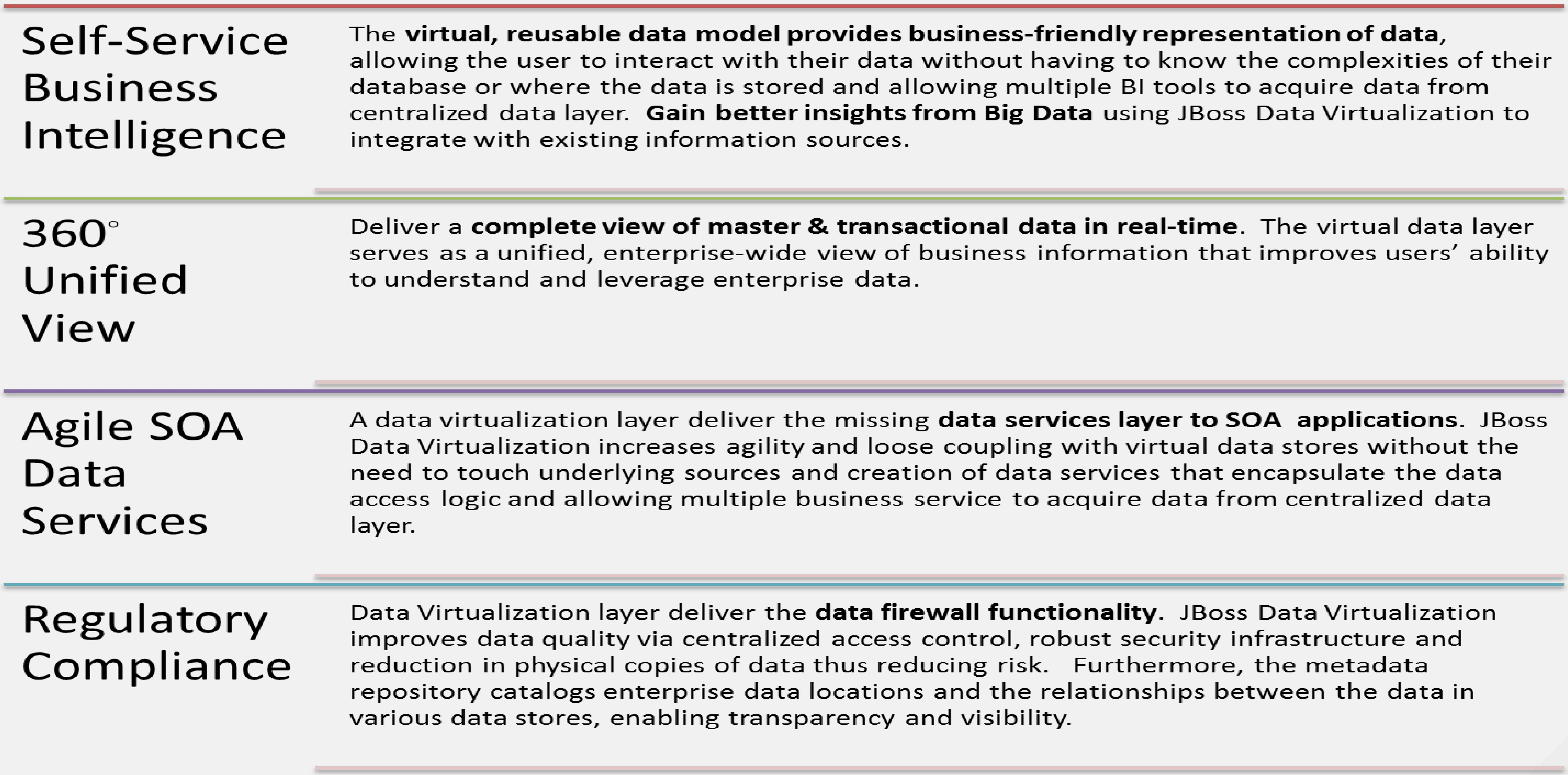
Mainframe

XML, CSV
& Excel Files

Enterprise Apps

Siloed & Complex

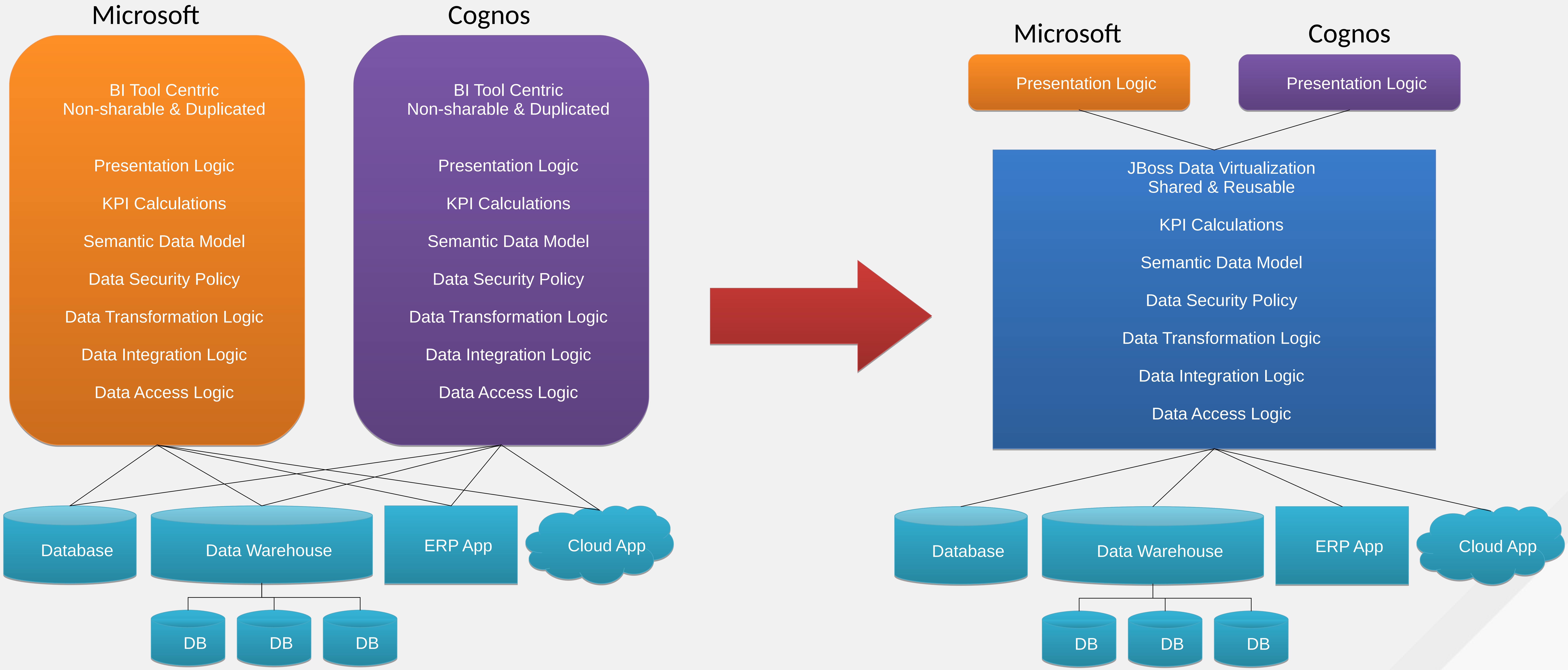
JBoss Data Virtualization – Use Cases



Self-Service Business Intelligence	The **virtual, reusable data model provides business-friendly representation of data**, allowing the user to interact with their data without having to know the complexities of their database or where the data is stored and allowing multiple BI tools to acquire data from centralized data layer. **Gain better insights from Big Data** using JBoss Data Virtualization to integrate with existing information sources.
360° Unified View	Deliver a **complete view of master & transactional data in real-time**. The virtual data layer serves as a unified, enterprise-wide view of business information that improves users' ability to understand and leverage enterprise data.
Agile SOA Data Services	A data virtualization layer deliver the missing **data services layer to SOA applications**. JBoss Data Virtualization increases agility and loose coupling with virtual data stores without the need to touch underlying sources and creation of data services that encapsulate the data access logic and allowing multiple business service to acquire data from centralized data layer.
Regulatory Compliance	Data Virtualization layer deliver the **data firewall functionality**. JBoss Data Virtualization improves data quality via centralized access control, robust security infrastructure and reduction in physical copies of data thus reducing risk. Furthermore, the metadata repository catalogs enterprise data locations and the relationships between the data in various data stores, enabling transparency and visibility.

Enable Self-Service Business Intelligence

Shared, Reusable Logic = Lighter, Faster Client Development



Microsoft

BI Tool Centric
Non-sharable & Duplicated

Presentation Logic

KPI Calculations

Semantic Data Model

Data Security Policy

Data Transformation Logic

Data Integration Logic

Data Access Logic

Cognos

BI Tool Centric
Non-sharable & Duplicated

Presentation Logic

KPI Calculations

Semantic Data Model

Data Security Policy

Data Transformation Logic

Data Integration Logic

Data Access Logic

Database

Data Warehouse

ERP App

Cloud App

DB

DB

DB

Microsoft

Presentation Logic

Cognos

Presentation Logic

JBoss Data Virtualization
Shared & Reusable

KPI Calculations

Semantic Data Model

Data Security Policy

Data Transformation Logic

Data Integration Logic

Data Access Logic

Database

Data Warehouse

ERP App

Cloud App

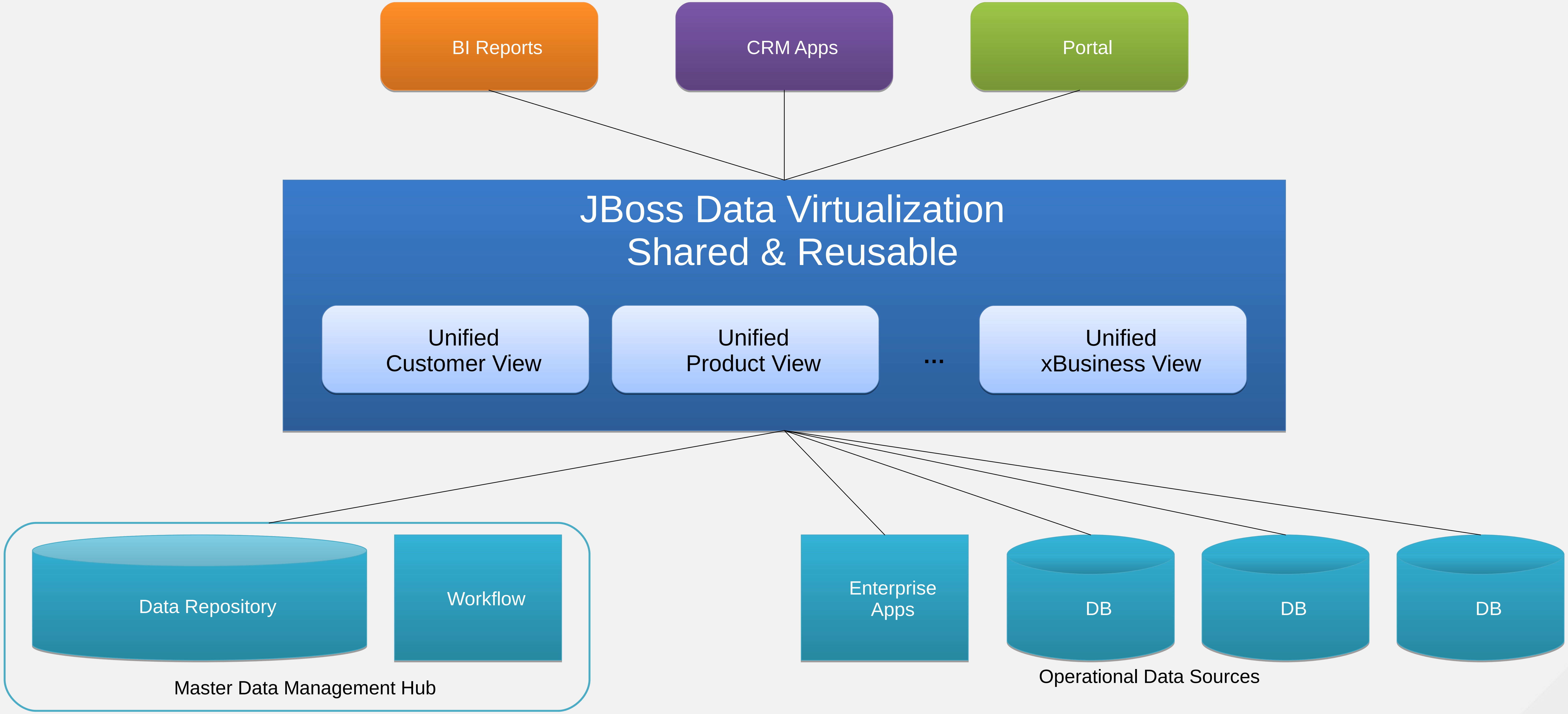
DB

DB

DB

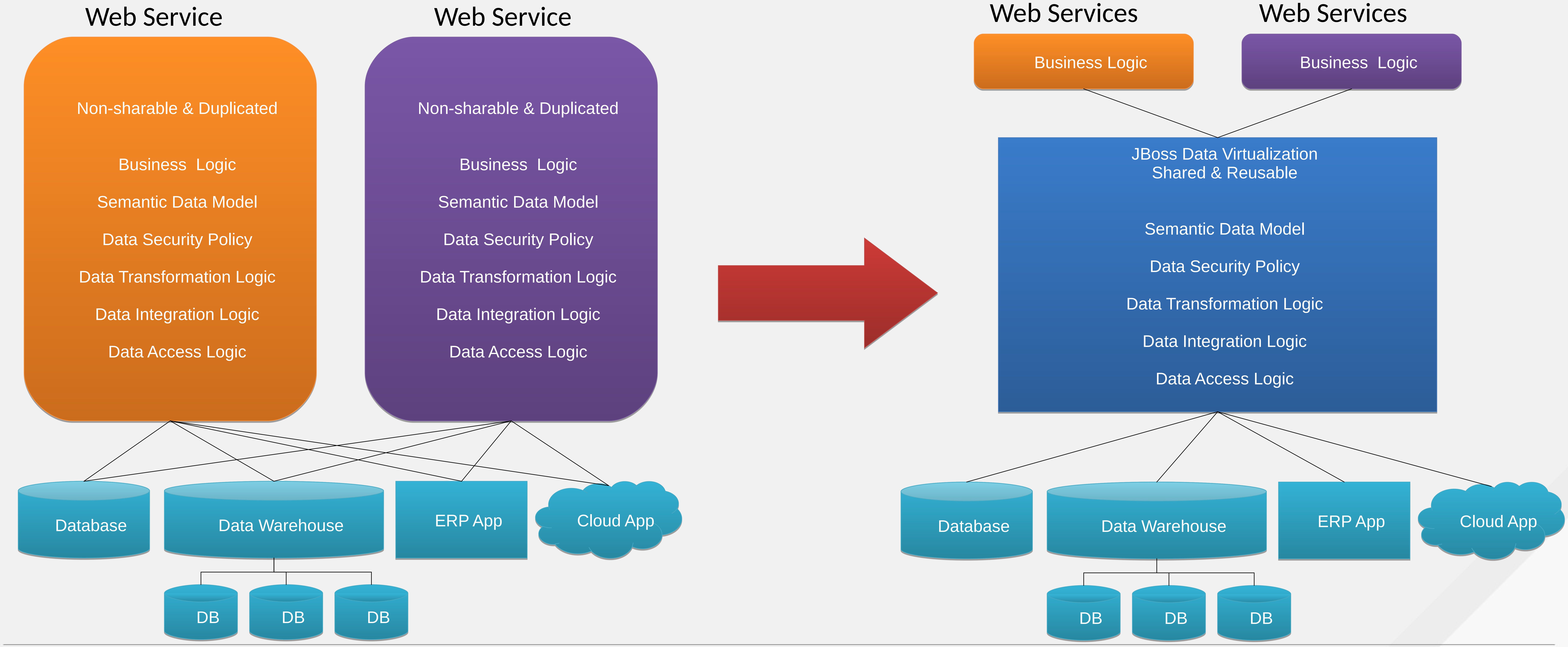
360° Unified View

Complete View of Master and Transactional Data in Real-time



Agile SOA Data Services

Shared, Reusable Logic = Lighter, Faster Service Development



Web Service

Non-sharable & Duplicated

Business Logic

Semantic Data Model

Data Security Policy

Data Transformation Logic

Data Integration Logic

Data Access Logic

Web Service

Non-sharable & Duplicated

Business Logic

Semantic Data Model

Data Security Policy

Data Transformation Logic

Data Integration Logic

Data Access Logic

Web Services

Business Logic

Web Services

Business Logic

JBoss Data Virtualization
Shared & Reusable

Semantic Data Model

Data Security Policy

Data Transformation Logic

Data Integration Logic

Data Access Logic

JBoss Data Virtualization

Key Business Values

Increase ROA

- Improved utilization of data assets
- Derive more value from existing investments
- Complements existing systems

Boost Agility

- Better/faster than hand coding
- Faster, less costly than batch data movement
- Data virtualization provides loose coupling

Improve Productivity

- Right data at the right time to the right people
- Decision support, BI with a complete view of information

Better Information Control

- Powerful security, Auditing, Data Firewall
- Avoid data silo proliferation
- Central data access and policy, Compliance

JBoss Data Virtualization

Key Differentiators

Lowest TCO

- Cost leadership lower adoption barrier
- Core based subscription provide flexibility across small to large deployment

Openness

- Open, community based innovation
- No vendor lock-in

Cloud Ready

- Private, public and hybrid cloud deployments

Comprehensive

- Integrated with JBoss Middleware portfolio for end-to-end business solution
- Single vendor support simplify IT operations

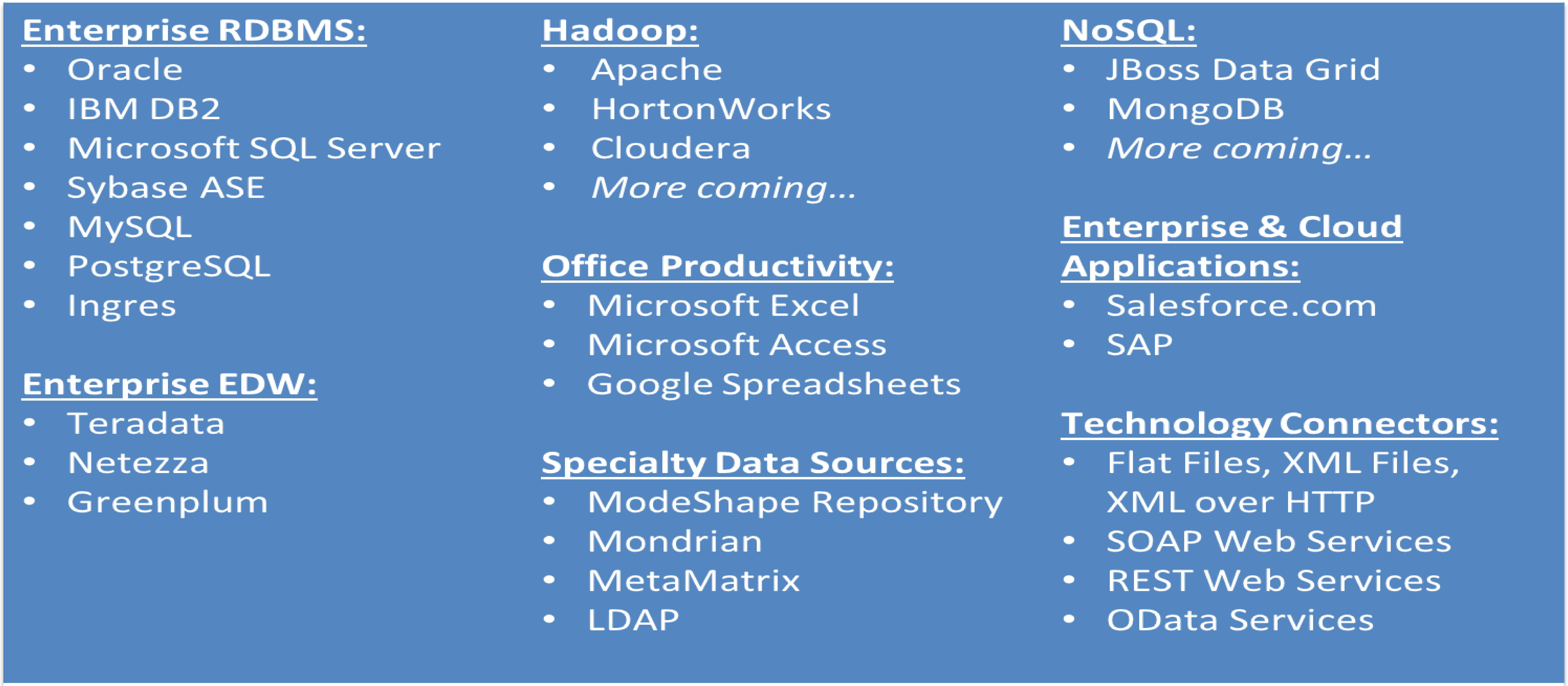
Performance

- Fast query processing optimizations, low footprint
- Comprehensive data provisioning options
- Quick data visualization through business dashboard

Product Details

JBoss Data Virtualization:

Supported Data Sources



Enterprise RDBMS:

- Oracle
- IBM DB2
- Microsoft SQL Server
- Sybase ASE
- MySQL
- PostgreSQL
- Ingres

Enterprise EDW:

- Teradata
- Netezza
- Greenplum

Hadoop:

- Apache
- HortonWorks
- Cloudera
- *More coming…*

Office Productivity:

- Microsoft Excel
- Microsoft Access
- Google Spreadsheets

Specialty Data Sources:

- ModeShape Repository
- Mondrian
- MetaMatrix
- LDAP

NoSQL:

- JBoss Data Grid
- MongoDB
- *More coming…*

Enterprise & Cloud Applications:

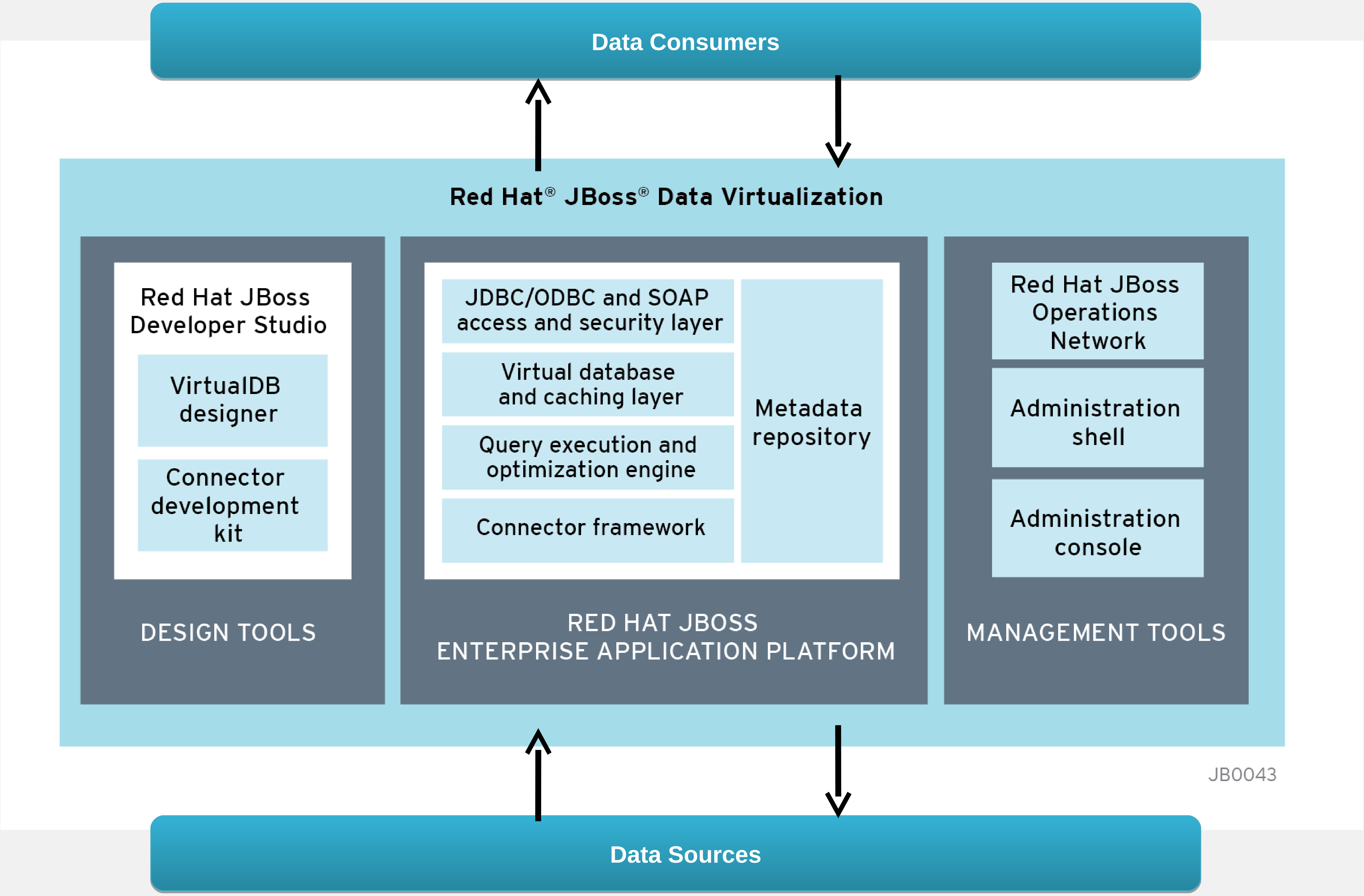
- Salesforce.com
- SAP

Technology Connectors:

- Flat Files, XML Files, XML over HTTP
- SOAP Web Services
- REST Web Services
- OData Services

JBoss Data Virtualization

Logical Architecture



Data Consumers

Red Hat® JBoss® Data Virtualization

Red Hat JBoss Developer Studio

VirtualDB designer

Connector development kit

DESIGN TOOLS

JDBC/ODBC and SOAP access and security layer

Virtual database and caching layer

Query execution and optimization engine

Connector framework

Metadata repository

RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM

Red Hat JBoss Operations Network

Administration shell

Administration console

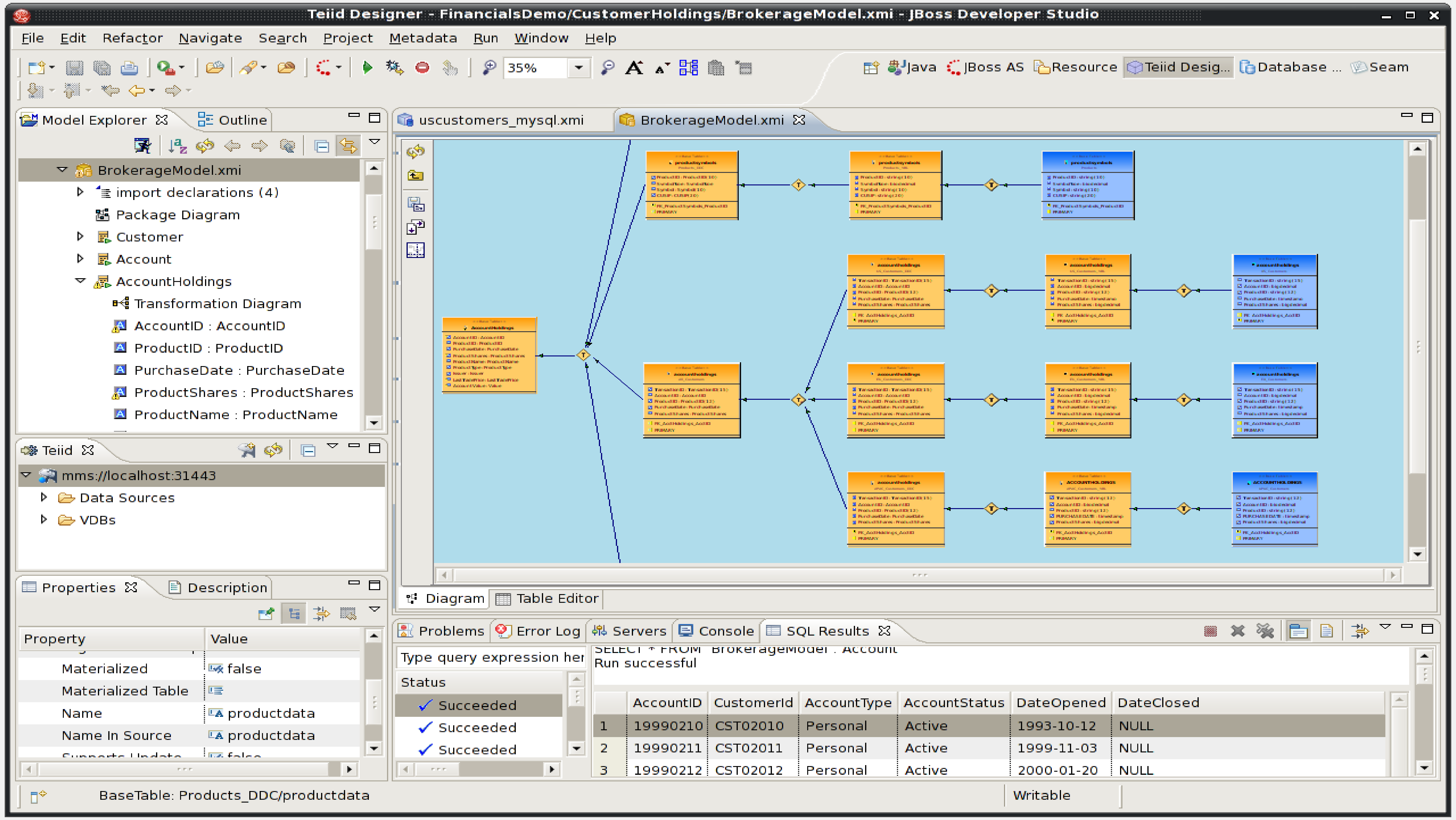
MANAGEMENT TOOLS

JB0043

Data Sources

Data Virtualization Designer

Model Driven Development



- BrokerageModel.xmi
  - import declarations (4)
  - Package Diagram
  - Customer
  - Account
  - AccountHoldings
    - Transformation Diagram
    - AccountID : AccountID
    - ProductID : ProductID
    - PurchaseDate : PurchaseDate
    - ProductShares : ProductShares
    - ProductName : ProductName

- mms://localhost:31443
  - Data Sources
  - VDBs

Property	Value
Materialized	false
Materialized Table	
Name	productdata
Name In Source	productdata
Supports Update	false

Type query expression her

Status
- ✓ Succeeded
- ✓ Succeeded
- ✓ Succeeded

SELECT * FROM BrokerageModel . Account
Run successful

	AccountID	CustomerId	AccountType	AccountStatus	DateOpened	DateClosed
1	19990210	CST02010	Personal	Active	1993-10-12	NULL
2	19990211	CST02011	Personal	Active	1999-11-03	NULL
3	19990212	CST02012	Personal	Active	2000-01-20	NULL

BaseTable: Products_DDC/productdata   Writable

Eclipse-based graphical tool for

- modeling,
- analyzing,
- Integrating,
- resolving semantic differences and
- testing

multiple data sources to produce

- Relational,
- XML and
- Web Service Views

that expose your business data without any programming.

- Shows structural transformations and dependencies
- Defines transformations

Rich Security Capabilities

Multiple forms of Authentication:

- Client Authentication: LoginModules (File, LDAP); Kerberos (JDBC/ODBC); HTTP Basic, WS UsernameToken Profile (Web Services)
  - PassThrough Authentication
- Source Authentication: Source credentials, Caller Identity (same credentials as client), RoleBasedCredentialMap (credentials per role), Execution payload/Custom

Authorization:

- Create, Read, Update, Delete, Execute permissions
- Row-based security
- Column masking

Additional Security:

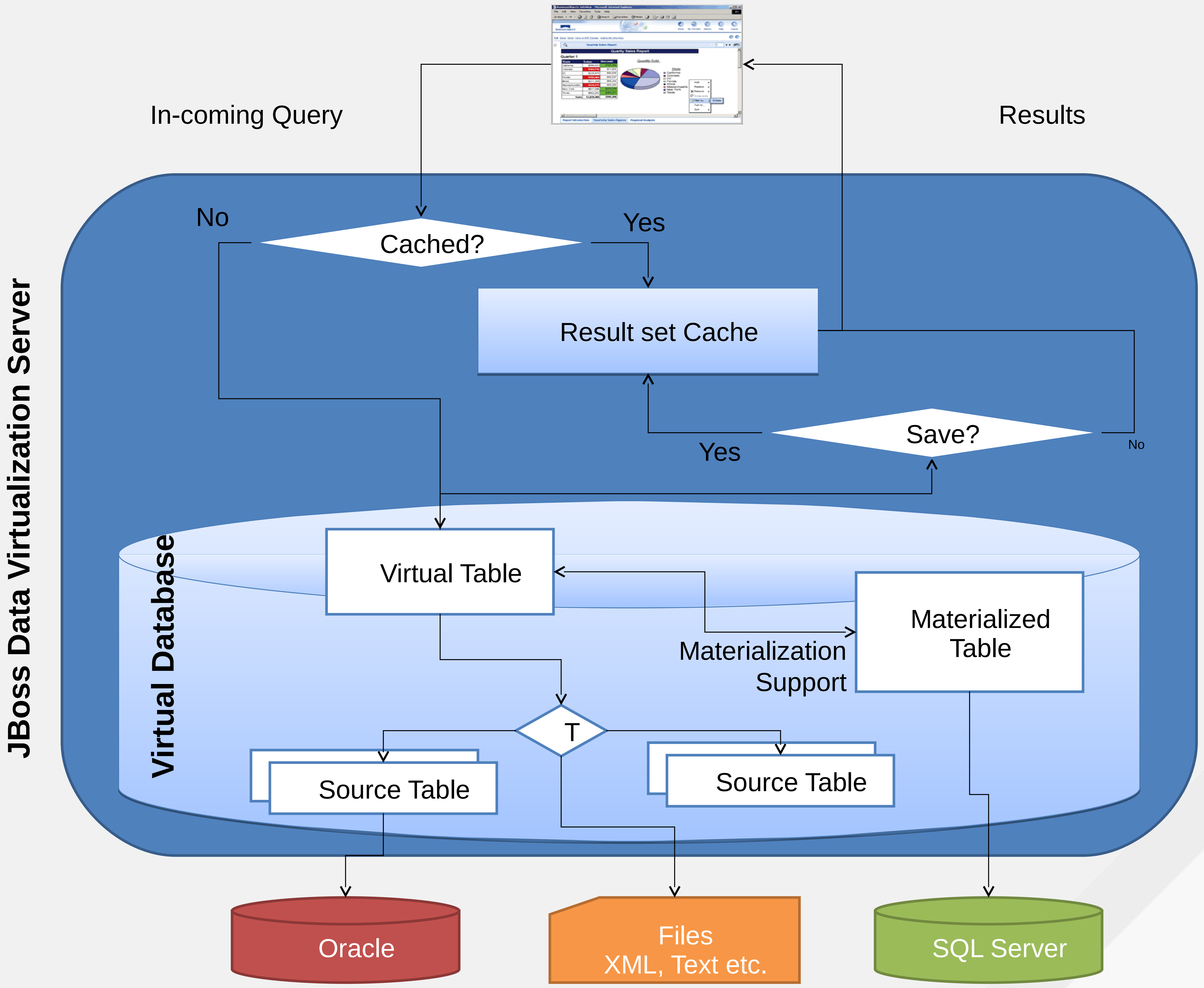
- Transport encryption (SSL: Anon, 1-way, 2-way)
- Password encryption

Performance Optimization

Caching & Materialized View

Multiple levels of caching to meet performance requirements and manage load on source systems

- Materialized Views
  - External or Internal materialized views
  - Ability to override use of materialized views
- Result set Caching
  - Applied to results return from user queries and virtual procedure calls
  - Configurable time to live and max. number of entries
- Code Table Caching
  - Suited for integrating reference data with transaction/operational data e.g. Country code, State Code etc.
- Caching hints to set time-to-live, memory preference, and updatability



Performance

A good segway into other aspect of “Enlighten”

En-”lighten”

- Make data light weight
  - Reduce latency
  - Improve performance
- Bring data closer to processing

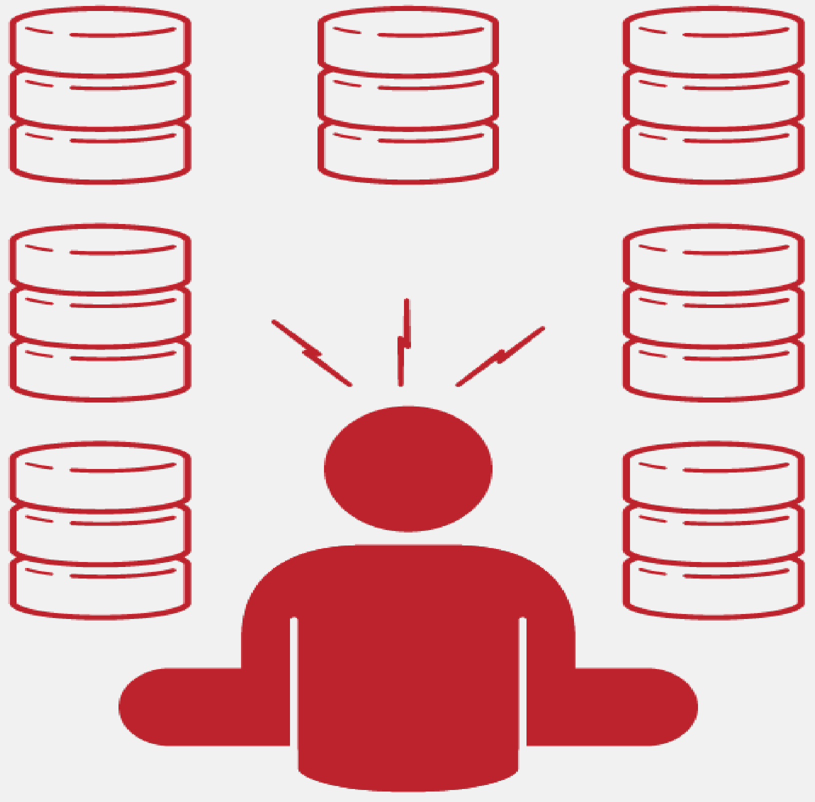
The challenge

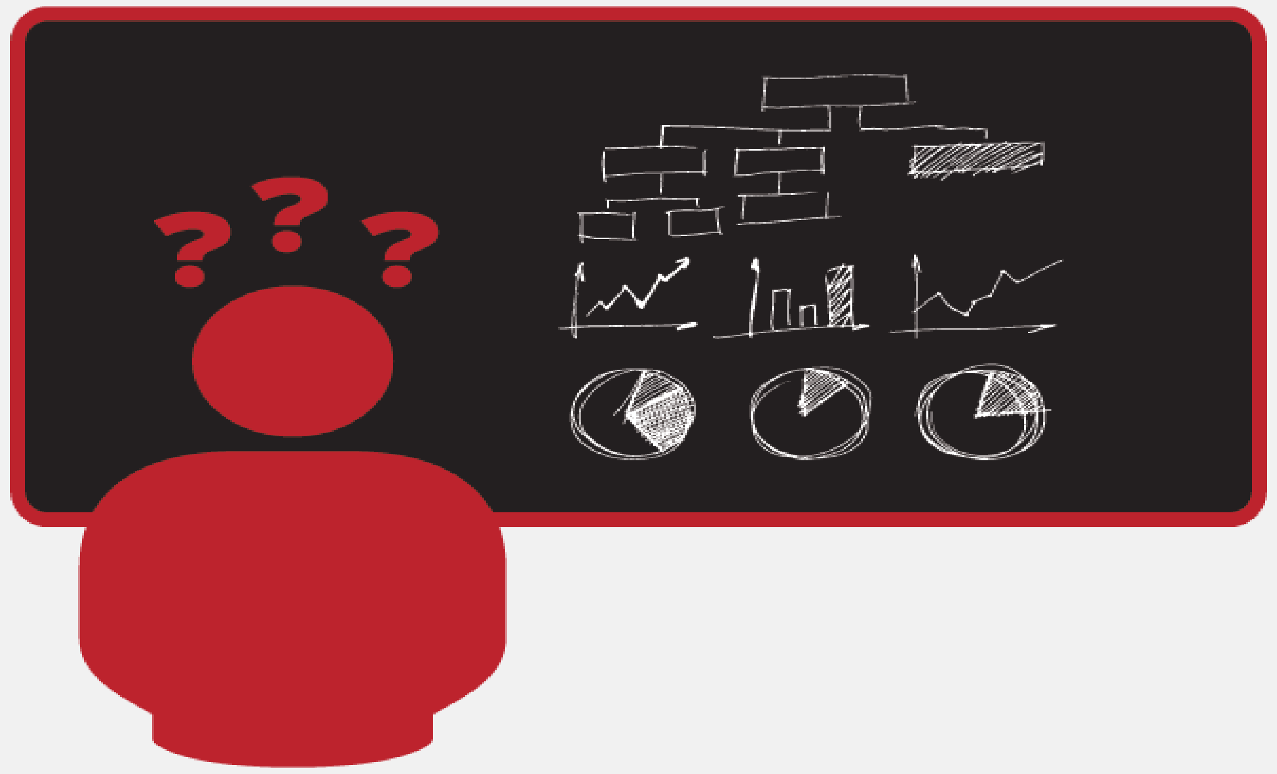
How do you design your application for:

- Performance during unprecedented transaction volumes?
- Availability to meet high uptime requirements?
- Flexibility in open hybrid cloud environments?
- Reliability to provide accurate, real-time information?
- Independence from the complex, rigid data-tier?

Solutions?

Modern challenges, traditional solutions?





Design for more...

- Pile on complex code, servers, databases, DBAs
- Cost-prohibitive
- Quick fix until you need to scale again

Start from scratch...

- Completely re-architect
- Sharding? Denormalization?
- Complicated
  - Time- and resource-intensive
  - Risky

How about a modern, agile approach?

Develop a new application strategy with data grids

The data grid solution:

- Handle high transactional throughput
- Meet strict performance requirements
- Meet high up-time requirements
- Streamline interactions with the traditional data tier

Benefits:

- Cost-effective
- Linear scalability
- Eliminates single point of failure
- Low-latency, fault-tolerant
- Responsive, available, flexible, elastic
- Cloud- and virtualization-ready

What is a data grid?

- An in-memory distributed data store designed for fast access to large volumes of data and scalability
- Commonly a complementary layer to the relational database and the application.

Key data grid characteristics:

- In-memory, distributed caching
- Elastic scalability
- Advanced querying
- Data replication
- Processing for streaming data
- Transaction capabilities

Our Solution:
Red Hat JBoss Data Grid

RED HAT® JBOSS® DATA GRID

Fast access to data

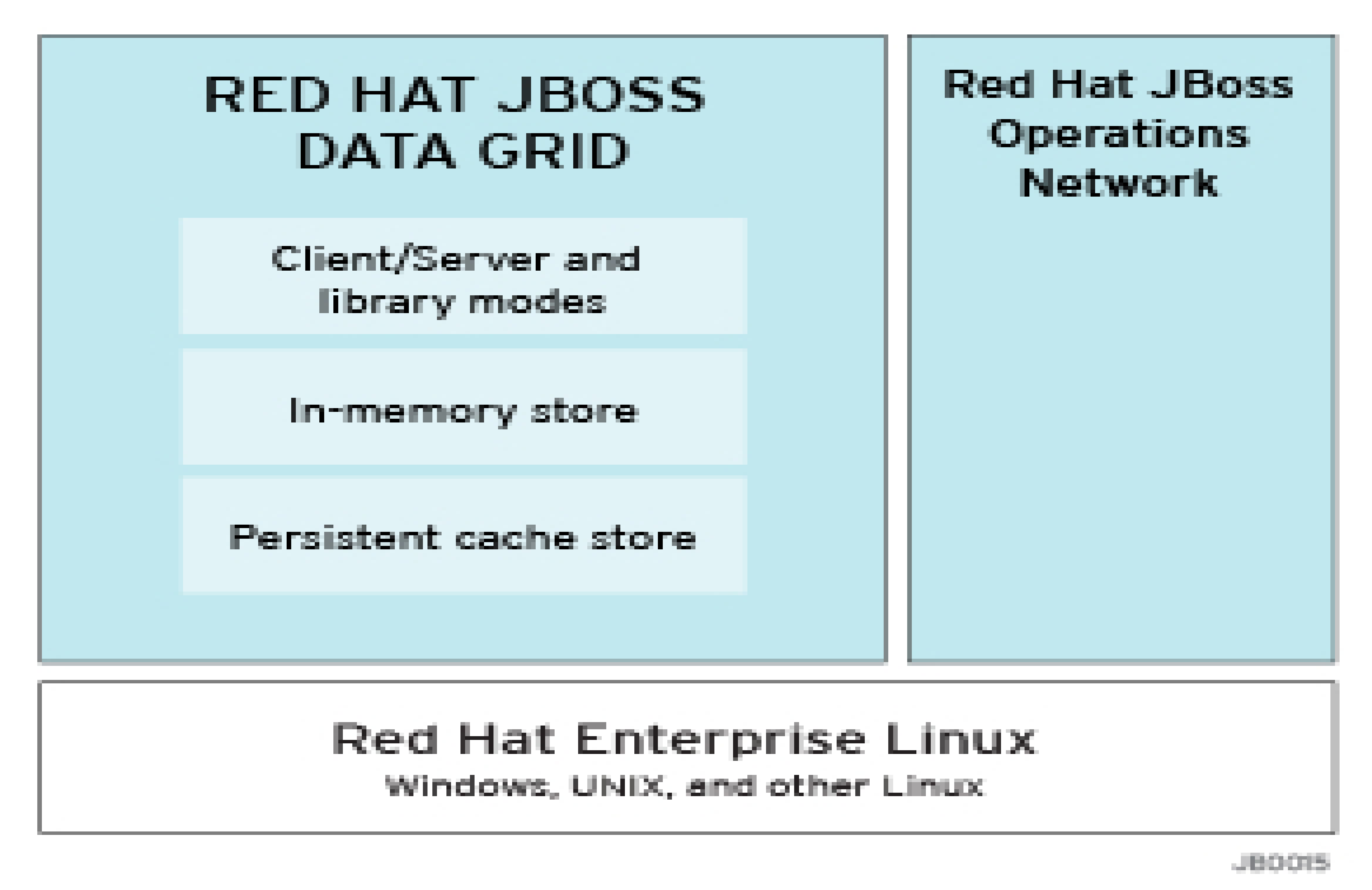
- In-memory speeds, high availability, reliability, elasticity
- Built on proven, popular open source Infinispan technology

Flexibility beyond Java

- Compatible with Java and non-Java platforms

Premium and advanced features in a cost-effective subscription

- Includes Red Hat JBoss Operations Network Management for management tooling
- Includes remote clients


RED HAT JBOSS DATA GRID
Client/Server and library modes
In-memory store
Persistent cache store
Red Hat JBoss Operations Network
Red Hat Enterprise Linux
Windows, UNIX, and other Linux
JB0015

Red Hat JBoss Data Grid

Accomplish more...

High availability to access data within and across datacenters

- Provide a complementary layer to the application and its relational store
- Meet data-retention requirements and up-time SLAs

Maintain fast response times with elastic scale

- Add or remove nodes using a straightforward process
- Data is distributed and replicated in the background

Designed for open hybrid cloud environments

- Independent control over the lifecycle, maintenance and costs of the application, its database, and the data grid

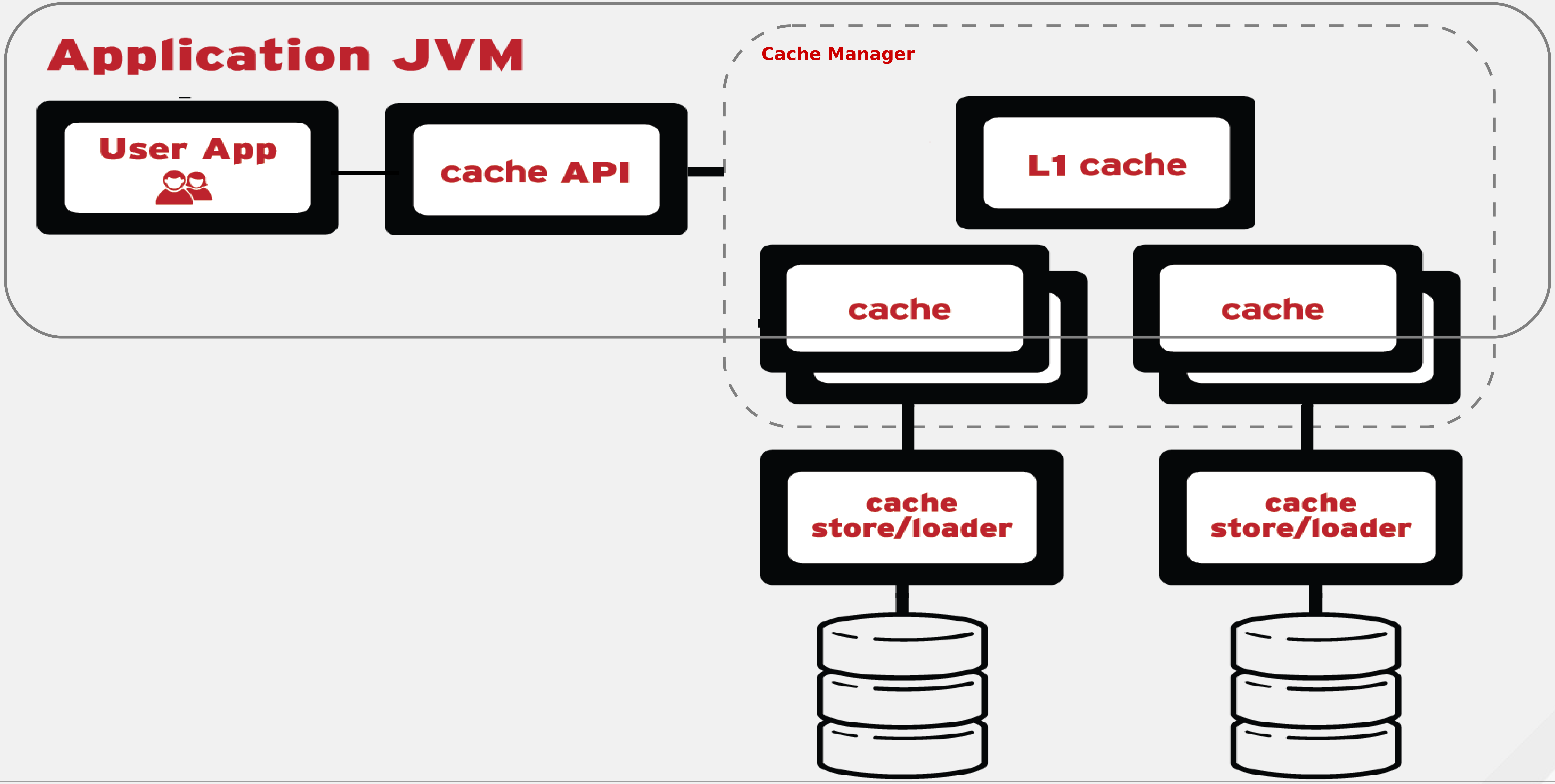
Flexibility to deploy your data, your way

- Free up IT budget by avoiding vendor lock-in and licensing costs
- Deploy your data, your way with multiple protocols and a Java API
- Developer-friendly, compatible, adaptable technology

Red Hat JBoss Data Grid:
Conceptual architecture

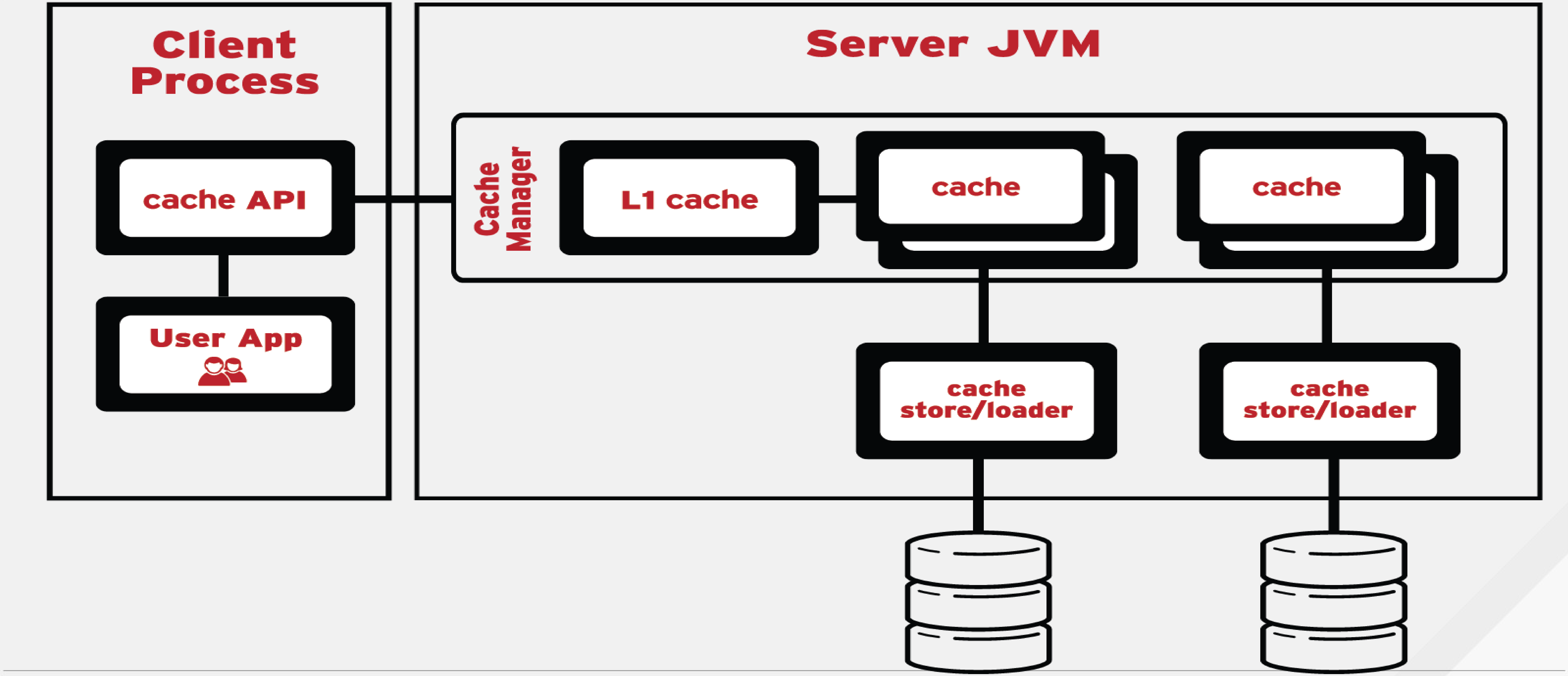
JBoss Data Grid conceptual architecture

Library mode



JBoss Data Grid conceptual architecture

Client / server



Conceptual architecture

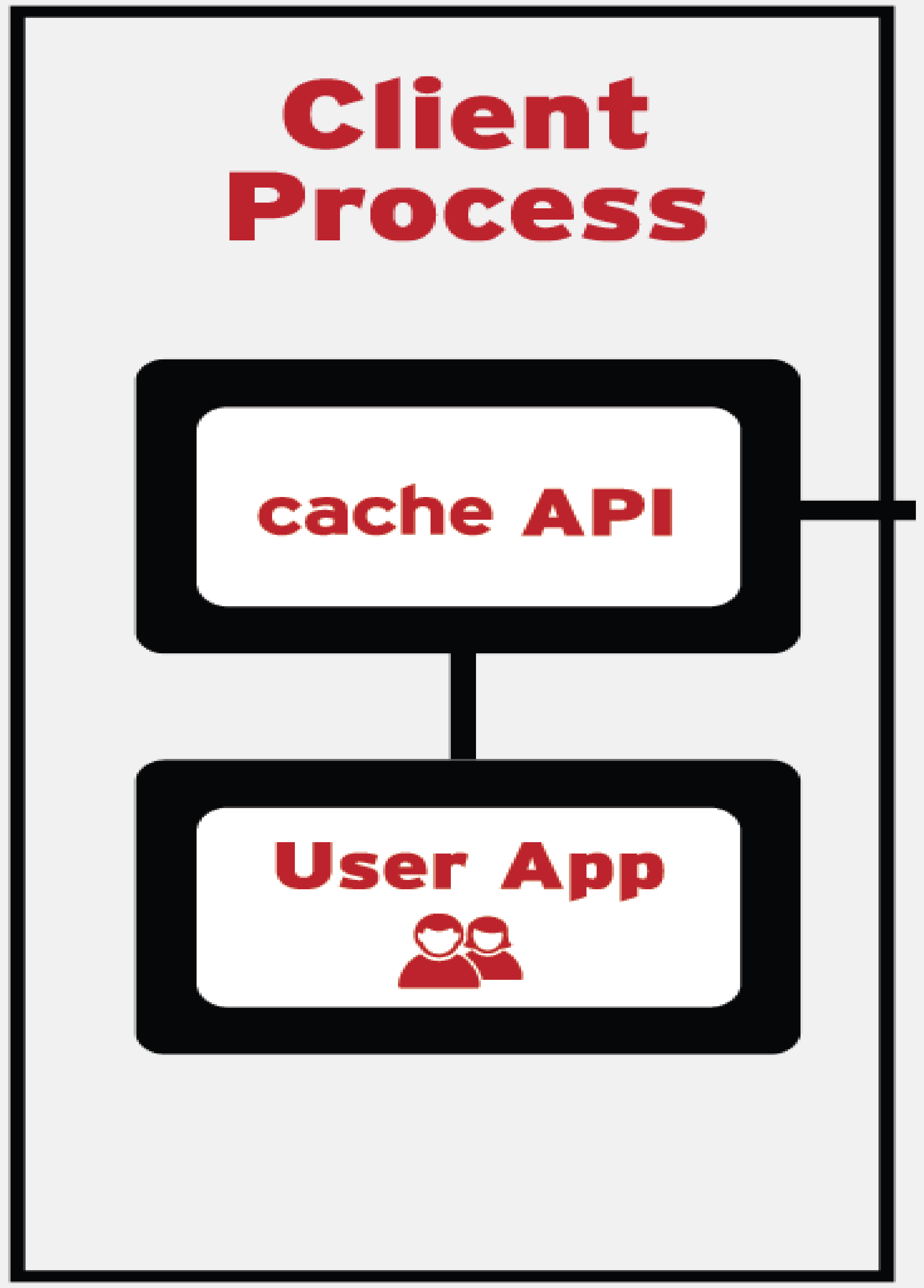
Cache API

User application

- End-user interface (i.e. web application, Java server application)

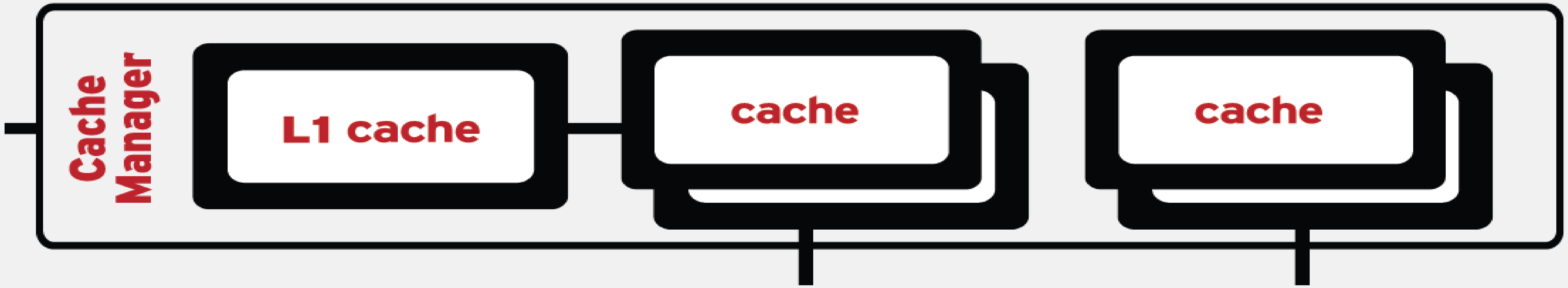
Cache API

- Uses memcached, Hot Rod, or REST APIs



Conceptual architecture

L1 cache, cache and cache manager



L1 cache

- Stores remote cache entries after they are initially accessed
- For fast retrieval and to prevent unnecessary remote fetch operations

Cache

- Houses cache instances

Cache manager

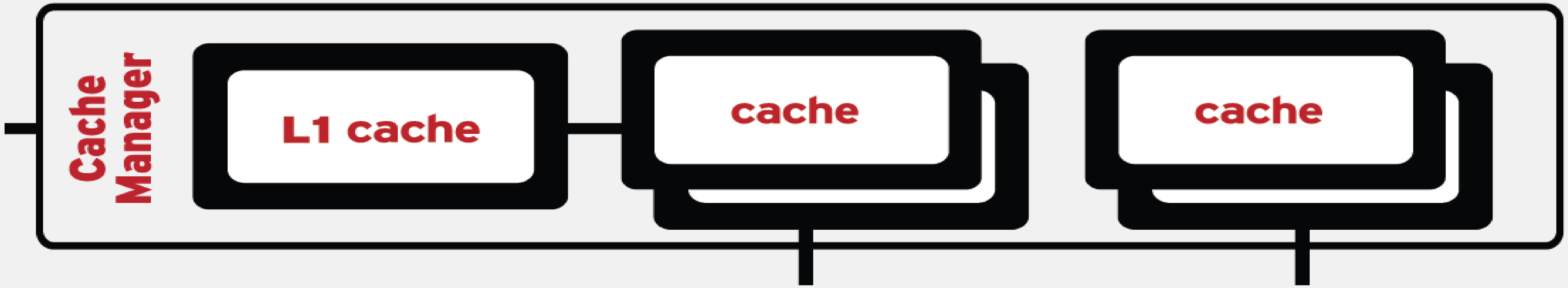
- Primary mechanism to retrieve a cache instance

Flexible setup

- One cache manager per process
- Multiple caches per cache manager
- One interface per cache

Conceptual architecture

L1 cache, cache and cache manager



Cache configuration

- Locking policy
- Transactions
- Eviction policy
- Expiration policy
- Persistence mechanism
- Backups
- L1 cache policy

Cache manager configuration

- Name / Alias / JNDI
- Start-up policy
- Transport policies
- Caches

Conceptual architecture

Cache store, cache loader, and persistent store

Cache loader

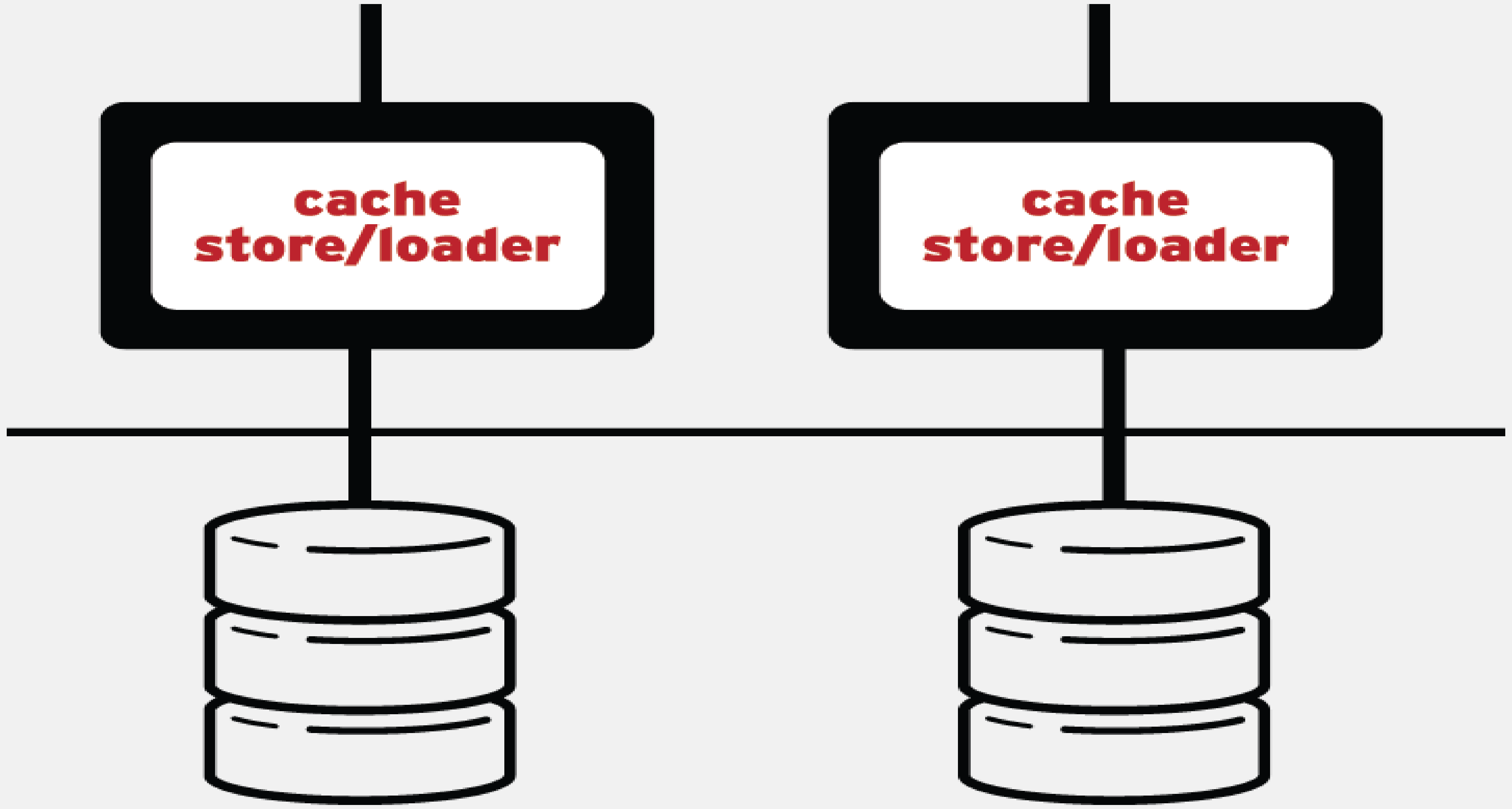
- Ready-only interface – locate and retrieve data

Cache store

- Cache loader with write capabilities

Persistent store

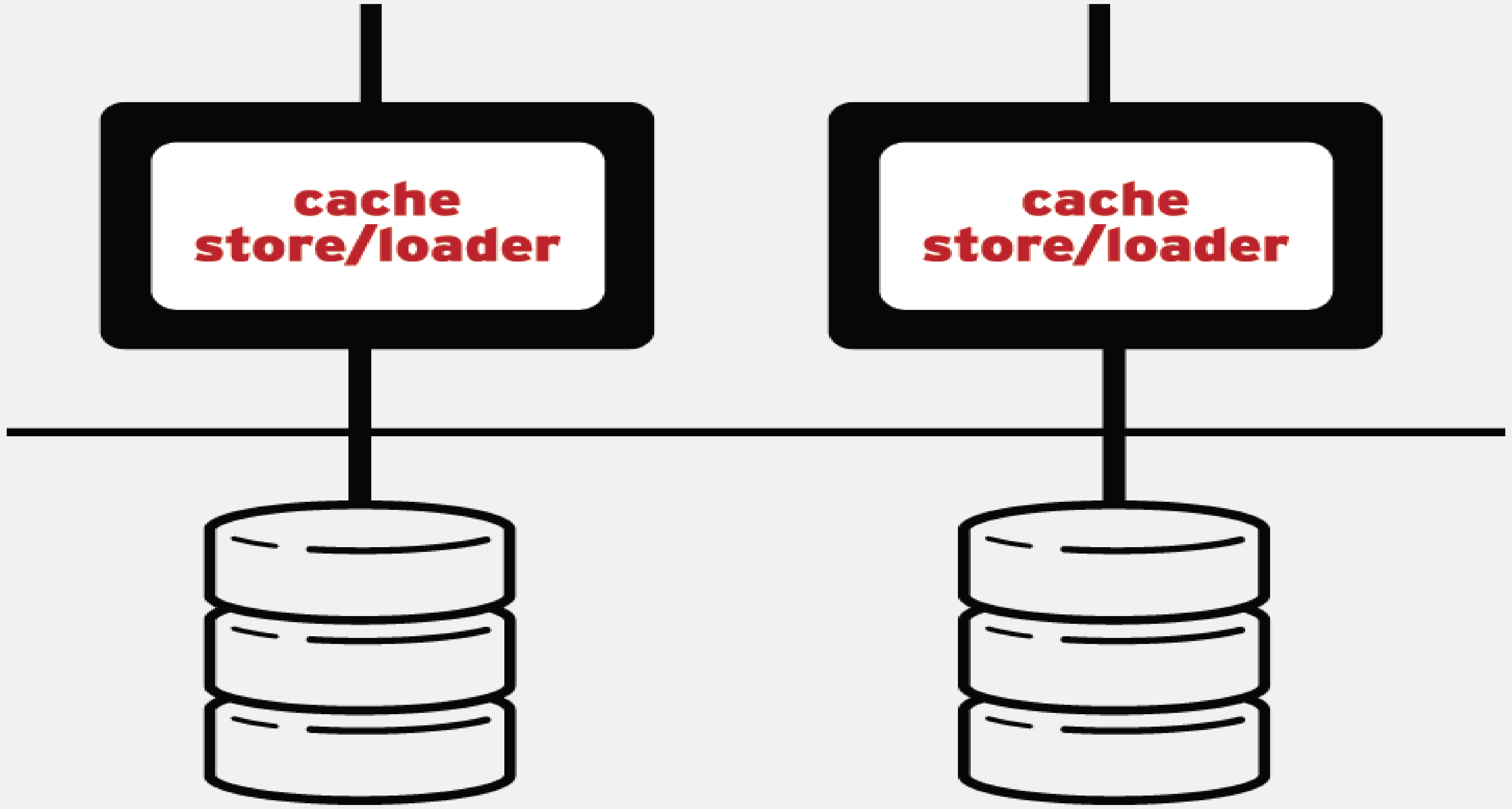
- Permanent store for cache instances and entries (i.e. relational database, file system, etc...)



Conceptual architecture

The cache store

- Write-behind or write-through behavior
- A cache has one or more cache stores
- Cache stores can be chained
- Can be loaded or purged on start
- Open and supported API for custom stores
- File, JDBC, remote



Synergy

Better Together - Big Data and Data Virtualization

Capture, Process and Integrate Data Volume, Velocity, Variety

Integrate & Analyze
Capture & Process
BI Analytics
(historical, operational, predictive)
SOA Composite Applications
>
Data Integration
JBoss Data Virtualization
In-memory Cache
JBoss Data Grid
Messaging and Event Processing
JBoss A-MQ and JBoss BRMS
Hadoop
Structured Data
Streaming Data
Semi-Structured Data
Red Hat Enterprise Linux & Virtualization
Red Hat Storage

Key Takeaways

- Data is Tactical, Information is Strategic
- Convert Fragmented Data to Actionable Information
- Decoupling, Federation, Virtualization are key
- Development and Maintenance should be simplified
- Future ready – Cloud ready
- Security and Performance are critical aspects
- Bring data close to processing
- Scalability, Data Availability

Thank You!

skale@redhat.com

RED HAT SUMMIT

LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.