Mahindra Satyam

SAP Business Analytics and Technology

Speed of Business Change

Years It Took to Reach a Market Audience of 50 Million

Radio

38 years

TV

13 years

Internet

4 years

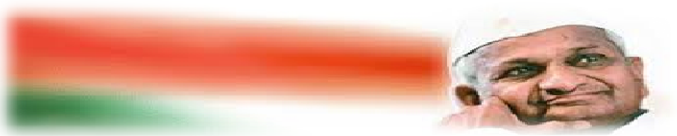
iPod

3 years

Facebook

2 years

"The greatest danger in times of turbulence is not the turbulence; it is to act with yesterday's logic." – Peter Drucker, 1980



In-Memory Computing

The elements of In-Memory computing are not new. However, dramatically improved hardware economics and technology innovations in software has now made it possible for SAP to deliver on its vision of the Real-Time Enterprise with In-Memory business applications

Conventional Databases
Disk Read
5 milliseconds

In-Memory Databases
Disk Read
5 nanoseconds

1 million times faster!

HW Technology Innovations

Multi-Core Architecture
(8 x 8core CPU in one server)

Massive parallel scaling with many blades

One blade ~$50.000 = 1 Enterprise Class Server

64bit address space
max at 2TB in current servers

100GB/s data throughput

Dramatic decline in price/performance

SAP SW Technology Innovations

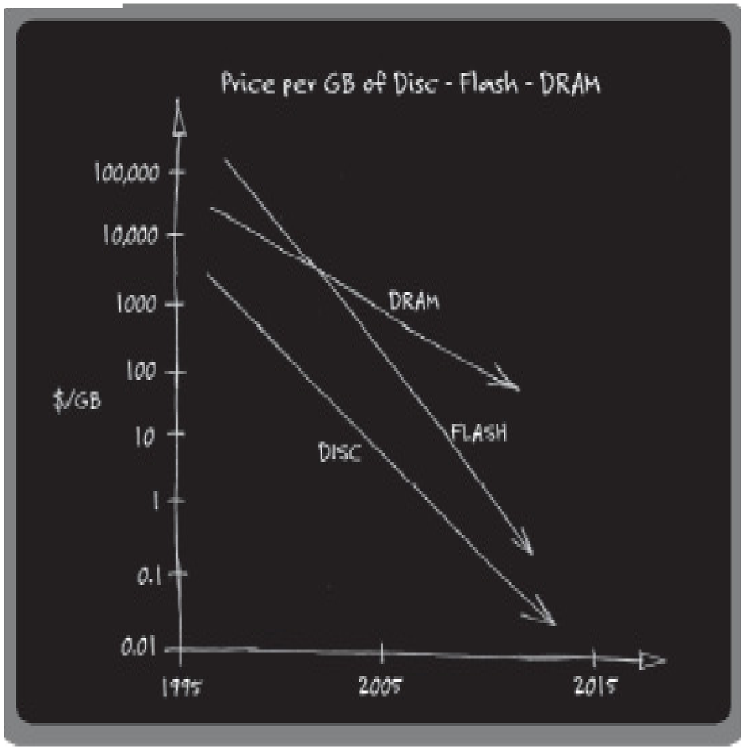
Row and Column Store

Compression

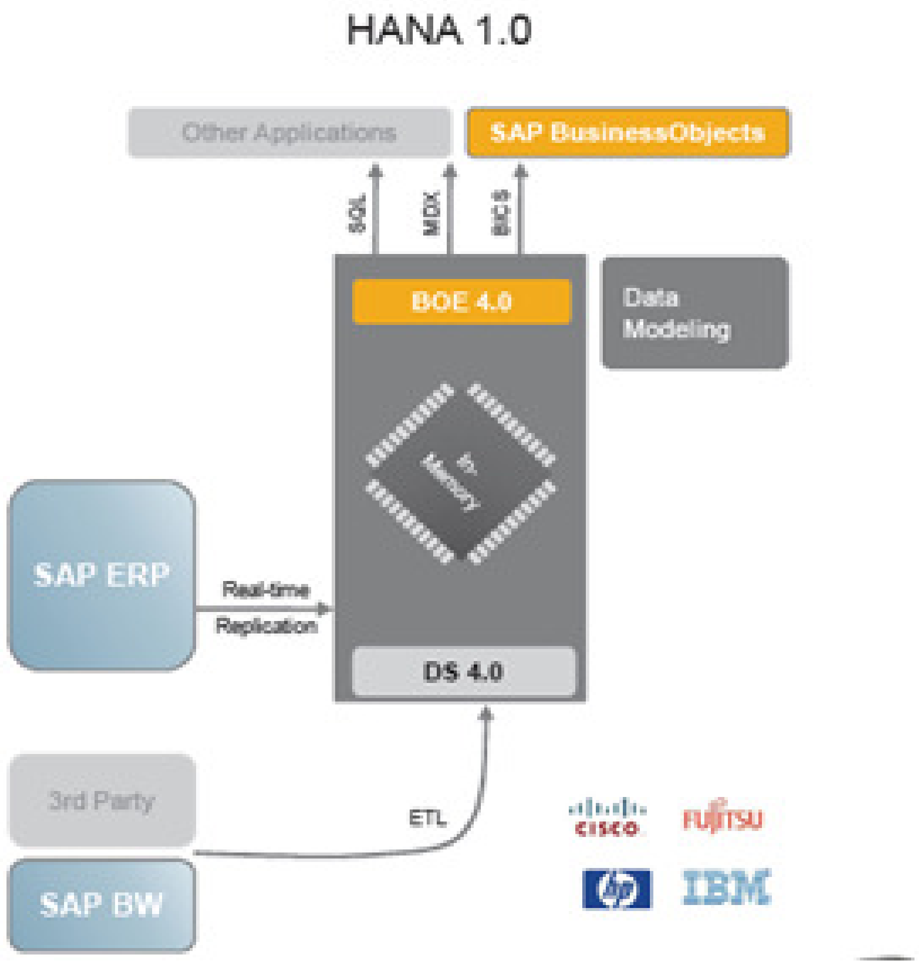
Partitioning

No Aggregate Tables

Insert Only on Delta



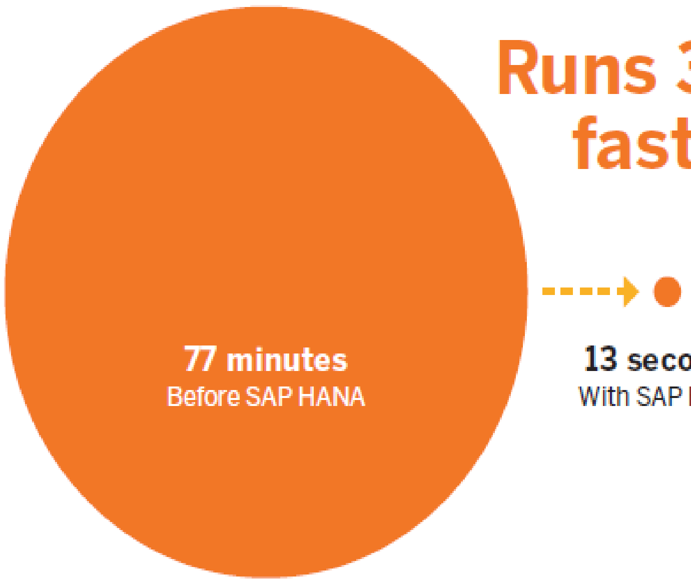
SAP -HANA – High Performance ANalytic Appliance
- In Memory Computing



HANA Proof Points

SAP High Performance Analytic Appliance (SAP HANA)

SAP HANA is the engine of the real-time enterprise. It provides a foundation on which to build a new generation of applications, enabling customers to analyze large quantities of data from virtually any source, in real time. The example below showcases actual customer performance of a core reporting process.



Runs 350x faster

77 minutes
Before SAP HANA

13 seconds
With SAP HANA

Experience the real-time Enterprise in action

A live analysis by a consumer products company reveals how SAP HANA analyzes current point-of-sale data in real time—empowering this organization to review segmentation, merchandising, inventory management, and forecasting information at the speed of thought.

70 retailers

460 billion records

50 terabytes of data

0.04 seconds
analysis response time

on any device,
anywhere, anytime

Agenda

- Components
- HANA Architecture

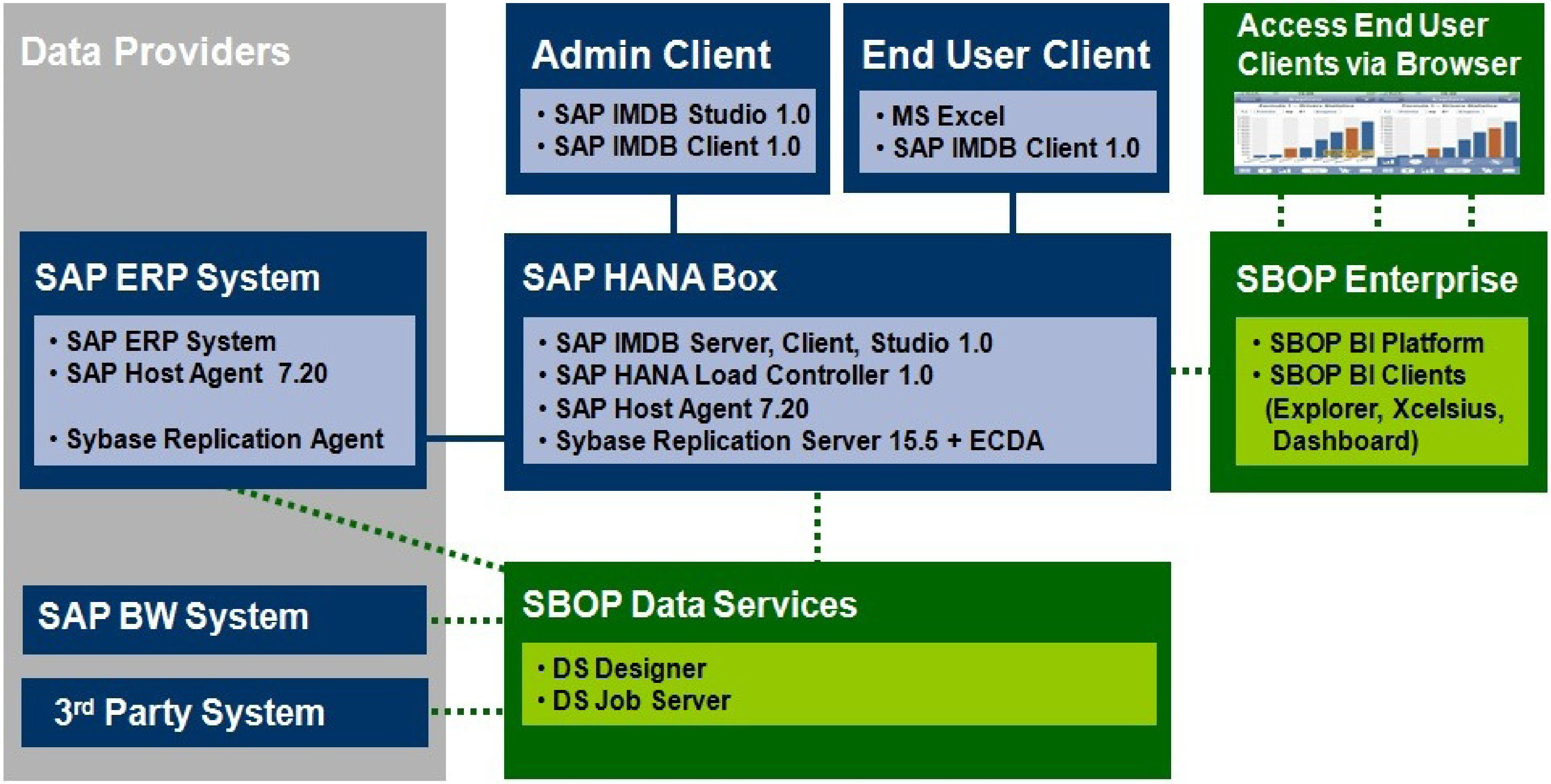
- Loading Data in HANA
- Modeling in HANA
- Reporting on HANA

- Request Processing and Execution
- Row Store
- Column Store
- Administration – Persistence Layer

- HANA Road Map
- Licensing Overview
- T-Shirt Sizing & Costs

Components

SAP HANA 1.0 Landscape Including External SAP BusinessObjects Servers

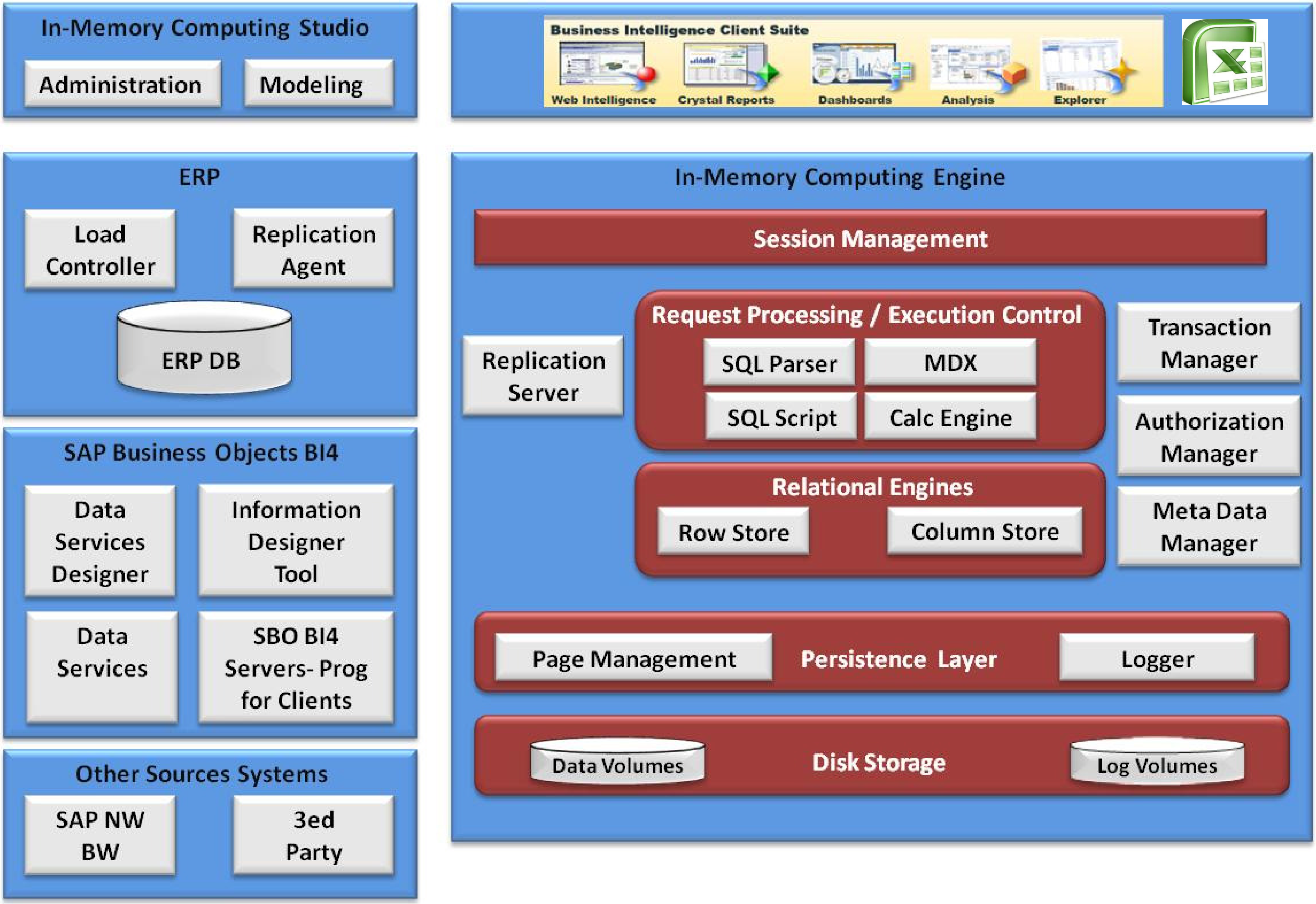


IMDB = SAP In-Memory Computing Engine
SBOP = SAP BusinessObjects Portfolio

HANA Architecture

HANA Architecture & Surrounding

Mahindra Satyam



In-Memory Computing Studio
- Administration
- Modeling

Business Intelligence Client Suite
- Web Intelligence
- Crystal Reports
- Dashboards
- Analysis
- Explorer

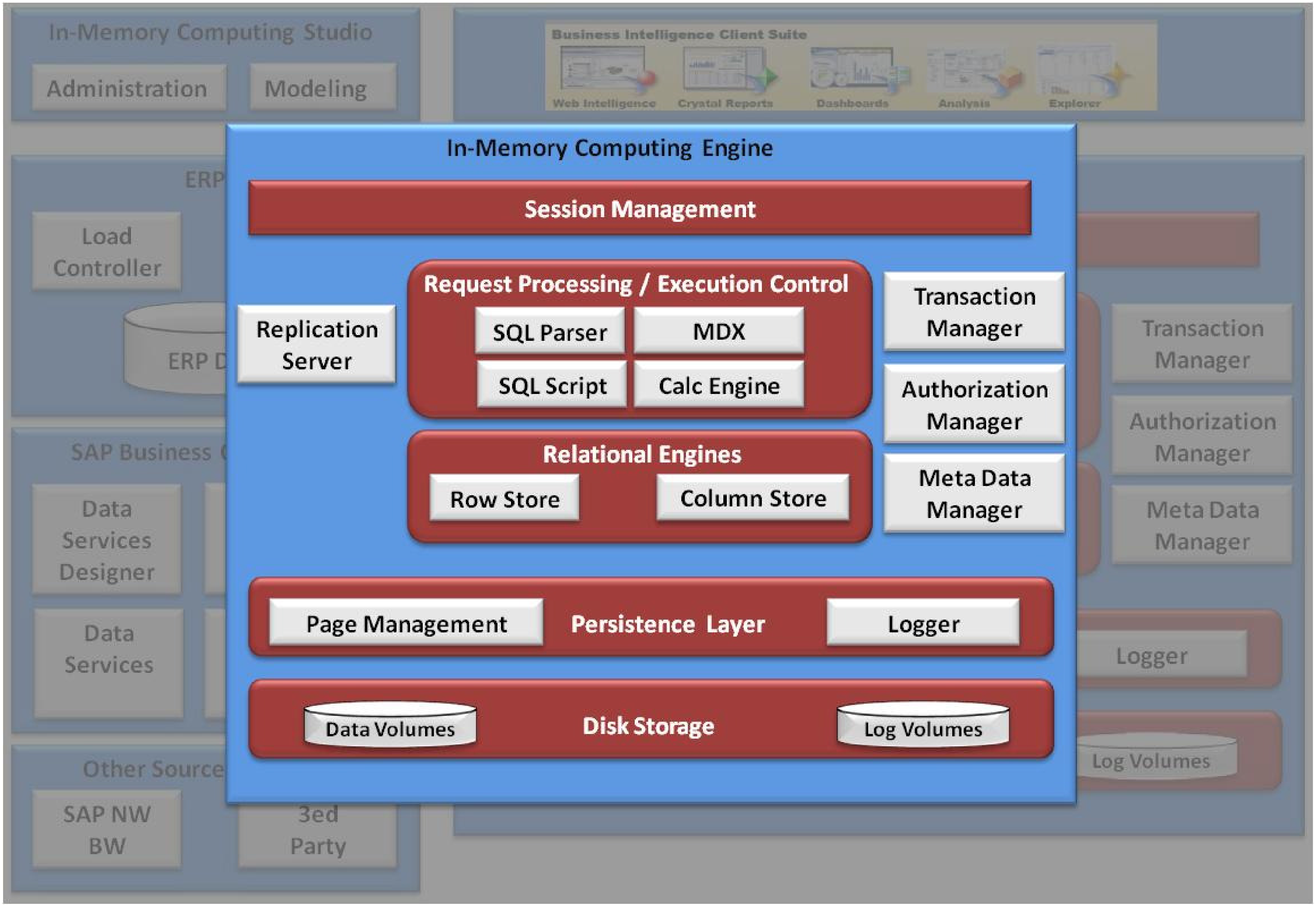
ERP
- Load Controller
- Replication Agent
- ERP DB

SAP Business Objects BI4
- Data Services Designer
- Information Designer Tool
- Data Services
- SBO BI4 Servers- Prog for Clients

Other Sources Systems
- SAP NW BW
- 3ed Party

In-Memory Computing Engine
- Session Management
- Replication Server
- Request Processing / Execution Control
  - SQL Parser
  - MDX
  - SQL Script
  - Calc Engine
- Relational Engines
  - Row Store
  - Column Store
- Transaction Manager
- Authorization Manager
- Meta Data Manager
- Persistence Layer
  - Page Management
  - Logger
- Disk Storage
  - Data Volumes
  - Log Volumes

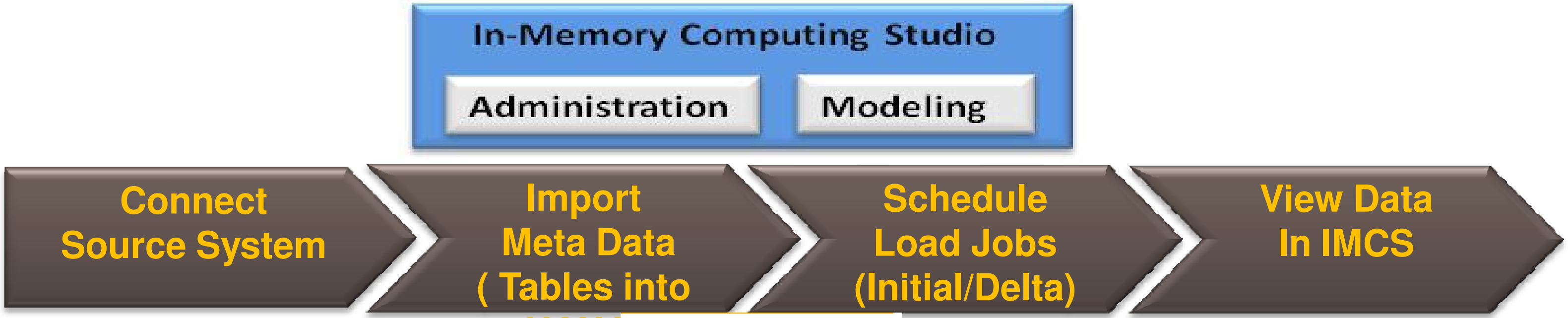
In-Memory Computing Engine



Loading Data
in
HANA

Loading Data into HANA
Options

Type	Tool
Trigger-Based Replication	SAP Landscape Transformation (SLT)
ETL-Based Replication	Business Objects - Data Services
Log-Based Replication	Replication Server


In-Memory Computing Studio

Administration

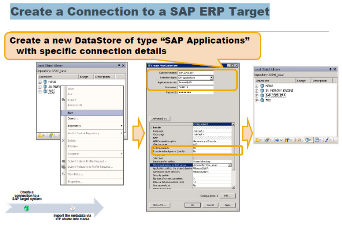
Modeling

Connect Source System

Import Meta Data (Tables into

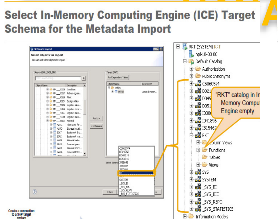
Schedule Load Jobs (Initial/Delta)

View Data In IMCS

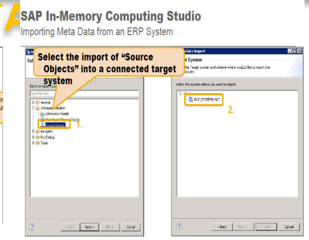

Create a Connection to a SAP ERP Target

Create a new DataStore of type "SAP Applications" with specific connection details

Create a connection to a SAP target system
Import the metadata via ICE studio into HANA

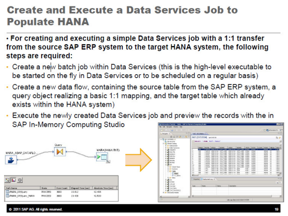

Select In-Memory Computing Engine (ICE) Target Schema for the Metadata Import

"RKT" catalog in In-Memory Computing Engine empty


SAP In-Memory Computing Studio

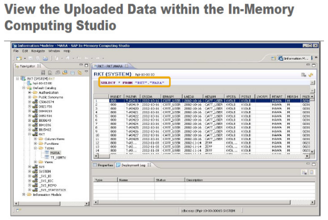
Importing Meta Data from an ERP System

Select the import of "Source Objects" into a connected target system


Create and Execute a Data Services Job to Populate HANA

- For creating and executing a simple Data Services job with a 1:1 transfer from the source SAP ERP system to the target HANA system, the following steps are required:
  - Create a new batch job within Data Services (this is the high-level executable to be started on the fly in Data Services or to be scheduled on a regular basis)
  - Create a new data flow, containing the source table from the SAP ERP system, a query object realizing a basic 1:1 mapping, and the target table which already exists within the HANA system)
  - Execute the newly created Data Services job and preview the records with the in SAP In-Memory Computing Studio

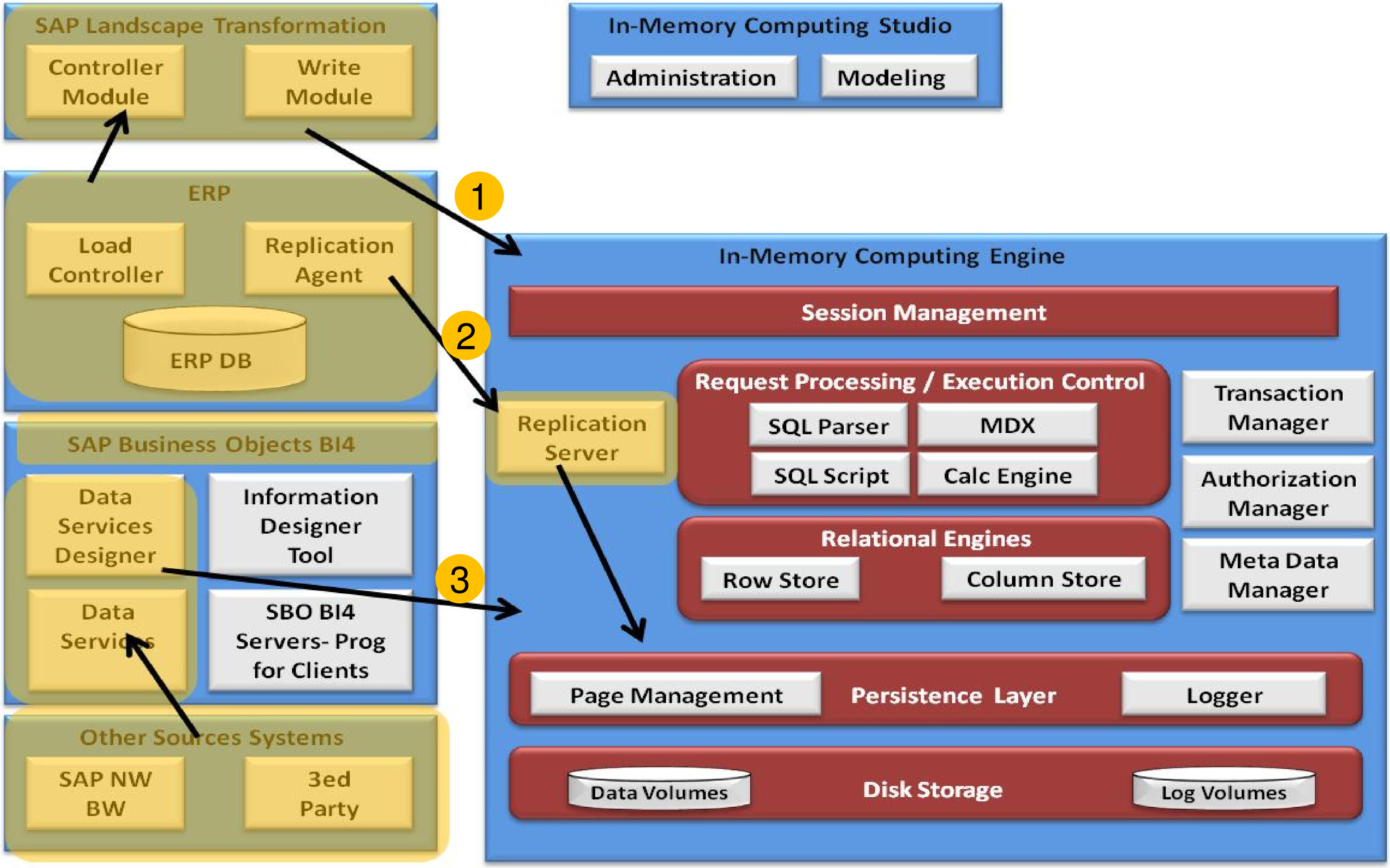
© 2011 SAP AG. All rights reserved. 19


View the Uploaded Data within the In-Memory Computing Studio

```
SELECT * FROM "RKT"."MARA"
```

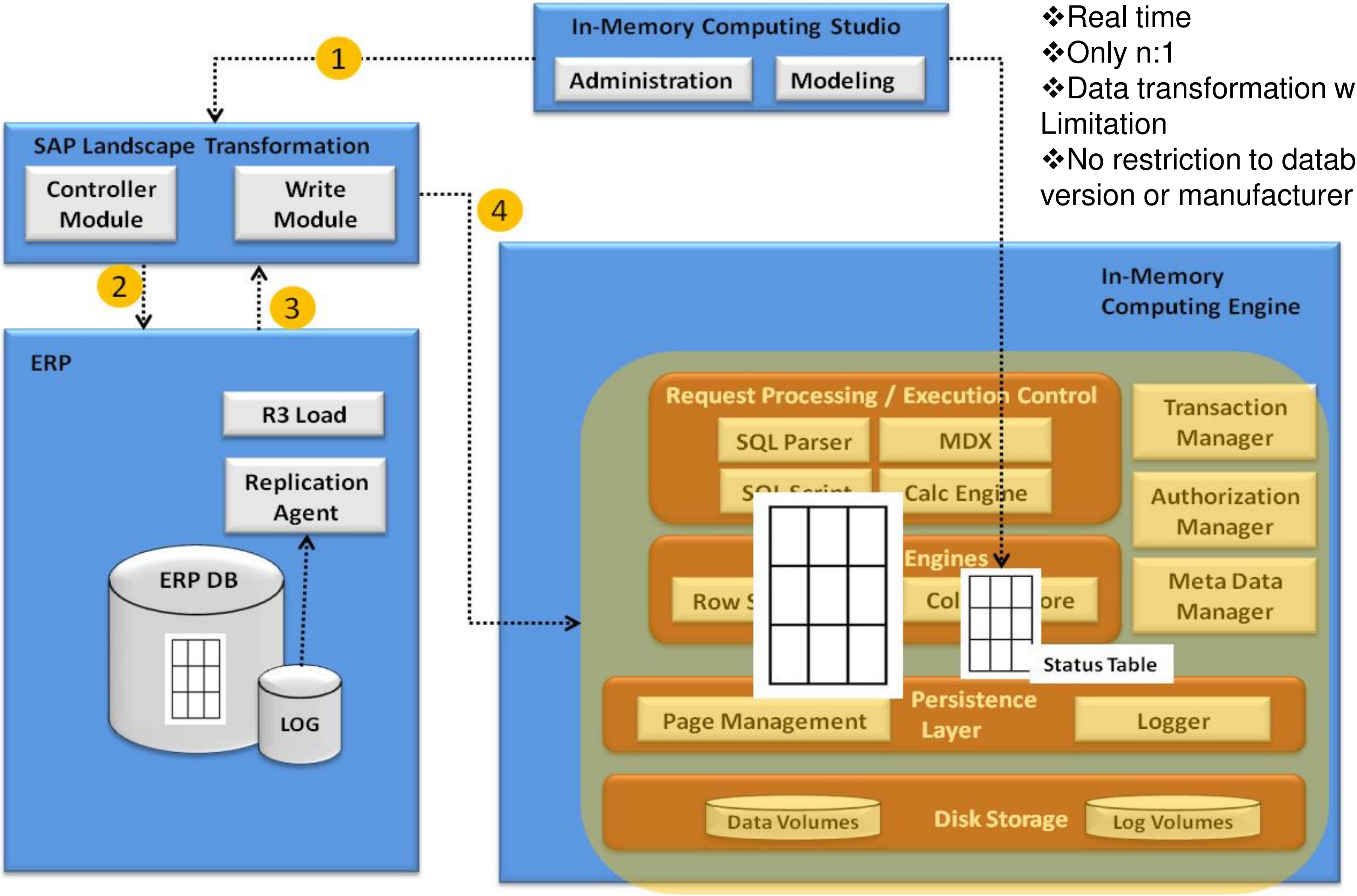
© Mahindra Satyam 2011

Loading Data into HANA



Trigger-Based Replication

SAP Landscape Transformation (SLT)

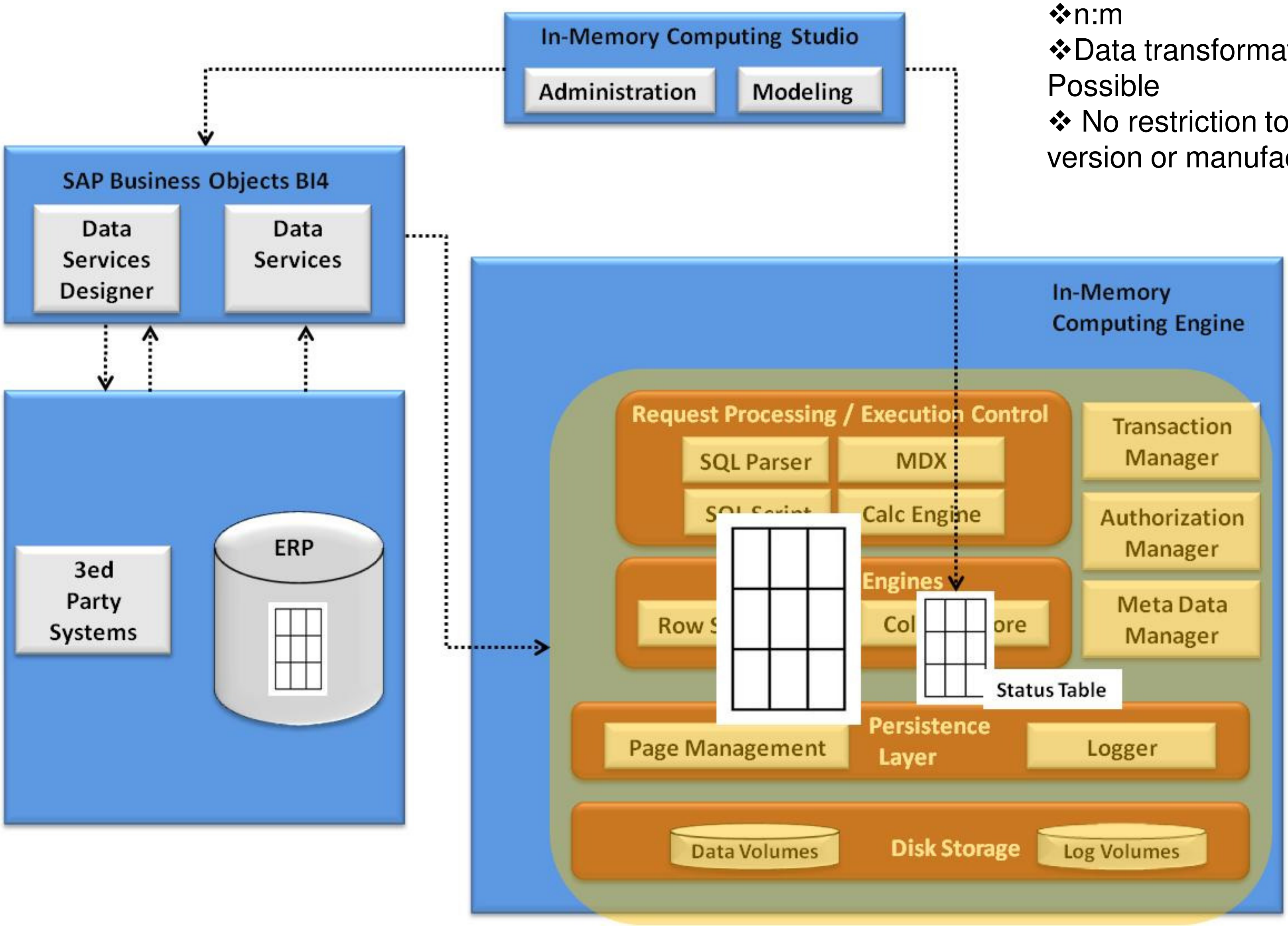


- ❖Real time
- ❖Only n:1
- ❖Data transformation with Limitation
- ❖No restriction to database version or manufacturer

ETL-Based Replication

Data Services

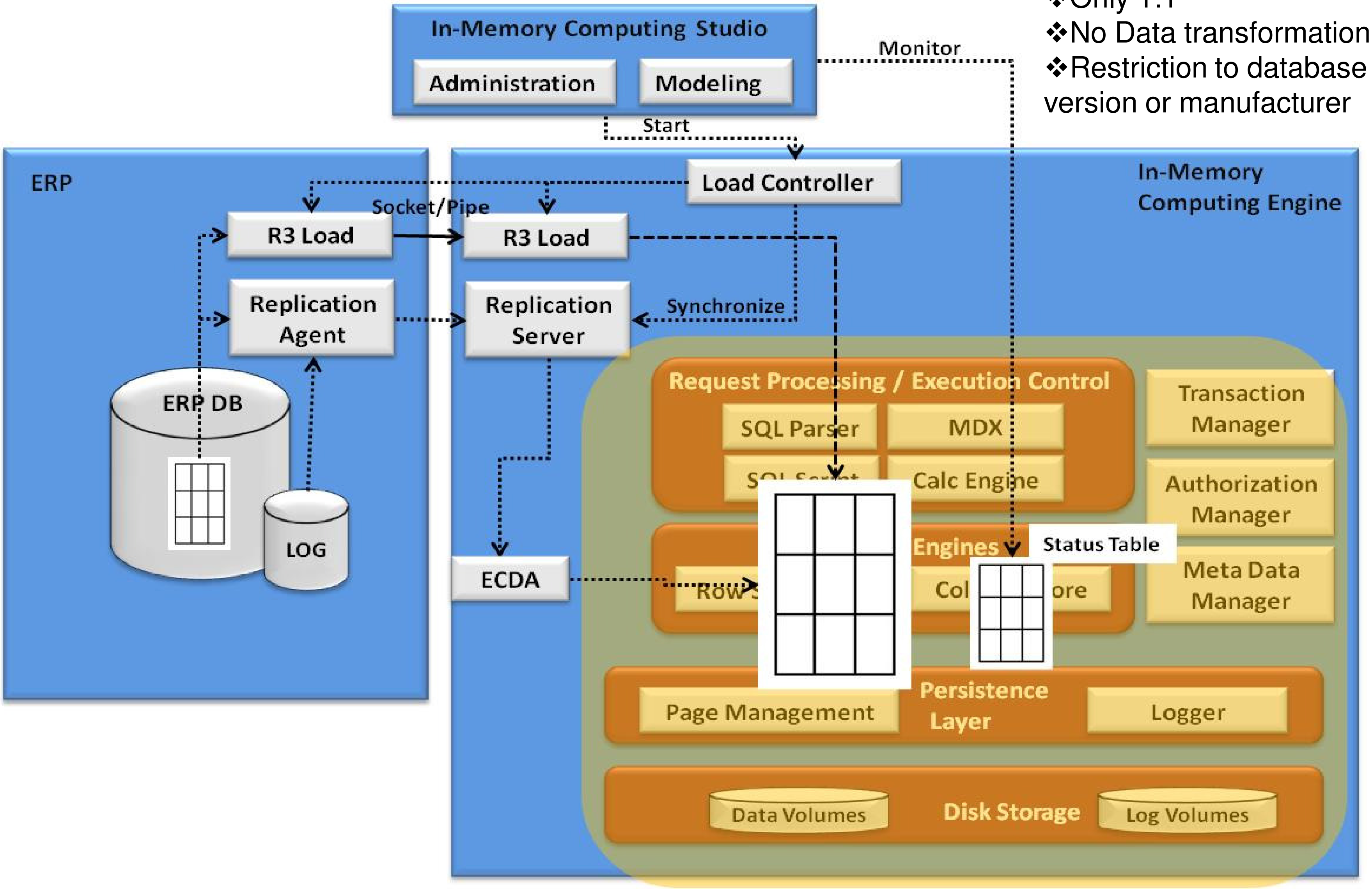
❖Delay more than 30 minutes
❖n:m
❖Data transformation Possible
❖ No restriction to database version or manufacturer



Log-Based Replication

Replication Server

- ❖Real time
- ❖Only 1:1
- ❖No Data transformation
- ❖Restriction to database version or manufacturer

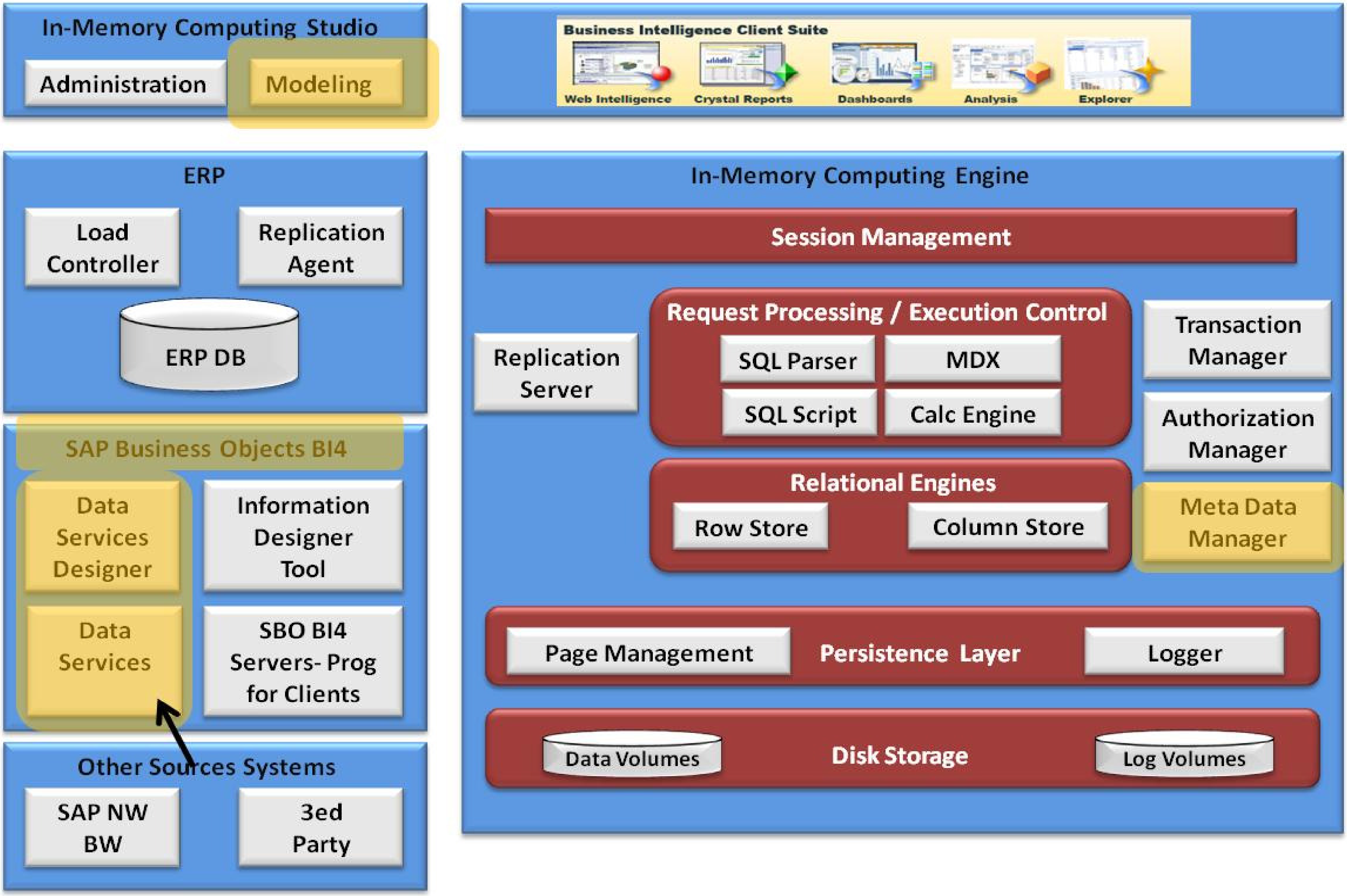


Replication Methods Comparison

Features	Trigger-Based Replication	ETL-Based Replication	Log-Based Replication
Real time or latency	Real time	Delay more than 30 minutes	Real time
n:m replication (many-to-many relationship support)	NO Only n back ends to 1 SAP HANA	YES	No Only 1:1
Table data transformation during replication (add or remove fields, conversions)	Yes, but is to be defined manually during project setup	YES	No
Free of restriction to database version or manufacturer	Yes	Yes	No
Free of restriction to operating system version	Yes (separate SLT system required if source system kernel < 720)	No NW720 required (Delta queue patch)	No
Pool /Cluster tables supported	Yes	Yes	No

Modeling in HANA

Modeling in HANA



In-Memory Computing Studio

Administration

Modeling

Business Intelligence Client Suite

Web Intelligence

Crystal Reports

Dashboards

Analysis

Explorer

ERP

Load Controller

Replication Agent

ERP DB

SAP Business Objects BI4

Data Services Designer

Information Designer Tool

Data Services

SBO BI4 Servers- Prog for Clients

Other Sources Systems

SAP NW BW

3ed Party

In-Memory Computing Engine

Session Management

Replication Server

Request Processing / Execution Control

SQL Parser

MDX

SQL Script

Calc Engine

Transaction Manager

Authorization Manager

Relational Engines

Row Store

Column Store

Meta Data Manager

Page Management

Persistence Layer

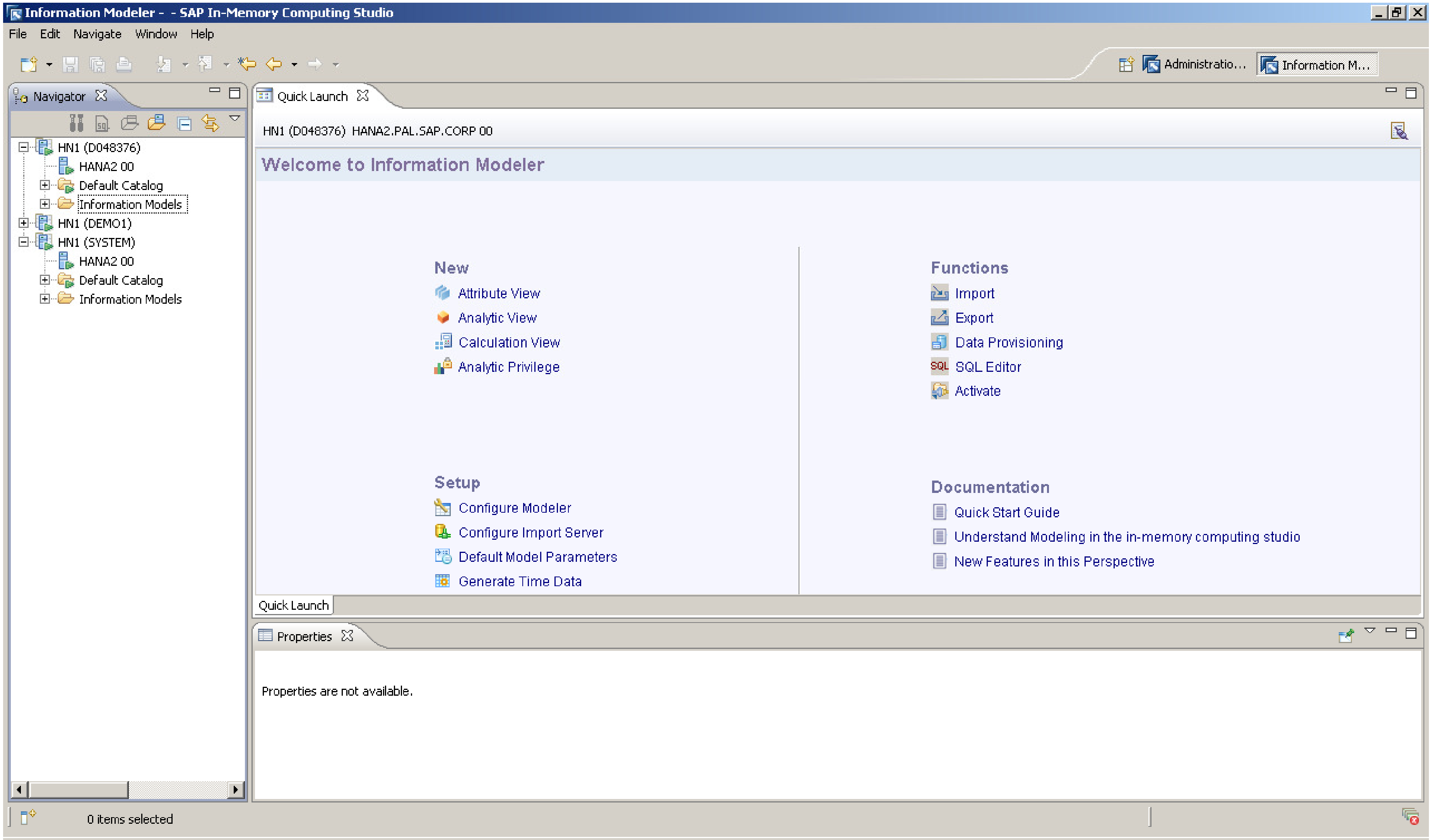
Logger

Data Volumes

Disk Storage

Log Volumes

SAP In-Memory Computing Studio Look and Feel



- HN1 (D048376)
  - HANA2 00
  - Default Catalog
  - Information Models
- HN1 (DEMO1)
- HN1 (SYSTEM)
  - HANA2 00
  - Default Catalog
  - Information Models

HN1 (D048376) HANA2.PAL.SAP.CORP 00

Welcome to Information Modeler

New
- Attribute View
- Analytic View
- Calculation View
- Analytic Privilege

Functions
- Import
- Export
- Data Provisioning
- SQL Editor
- Activate

Setup
- Configure Modeler
- Configure Import Server
- Default Model Parameters
- Generate Time Data

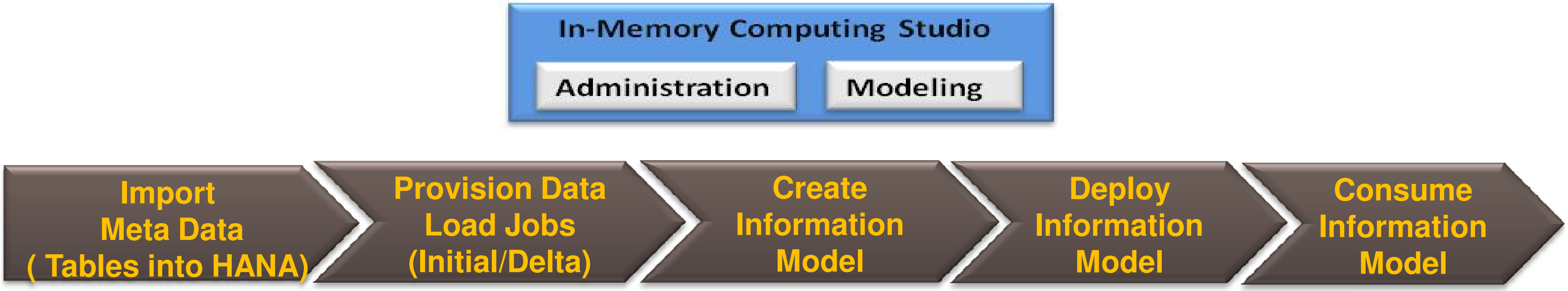
Documentation
- Quick Start Guide
- Understand Modeling in the in-memory computing studio
- New Features in this Perspective

Quick Launch

Properties are not available.

0 items selected

Modeling in HANA



In-Memory Computing Studio

Administration | Modeling

Import Meta Data (Tables into HANA) → Provision Data Load Jobs (Initial/Delta) → Create Information Model → Deploy Information Model → Consume Information Model

Data

Attributes : Descriptive Data (Also known as Characteristics in BW terminology)

Measures : Data that can be Quantified & Calculated . (Also known as Key Figures in BW terminology)

Views

Attribute View: i.e. Dimensions

Analytic View : i.e. Cube

Calculation View: Similar to virtual providers with service concept in BW

Hierarchies

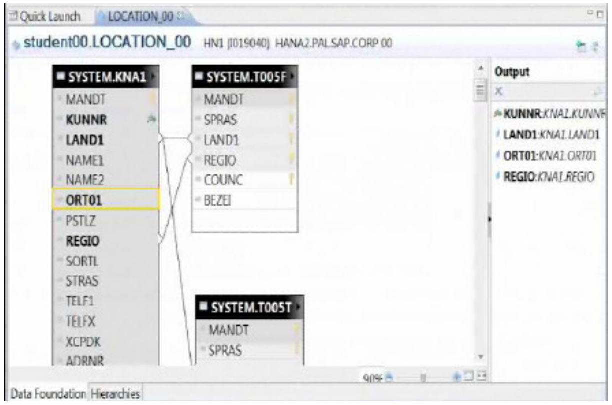
Leveled – based on Multiple Attributes

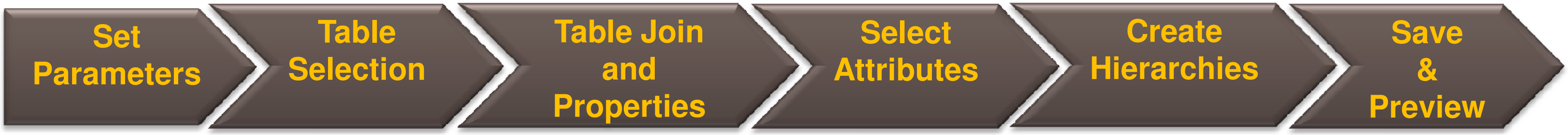
Parent Child Hierarchy

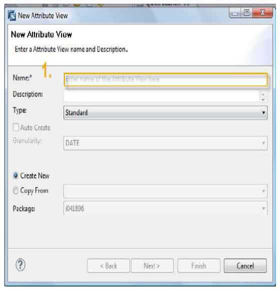
Analytical Privileges – **Security Objects**

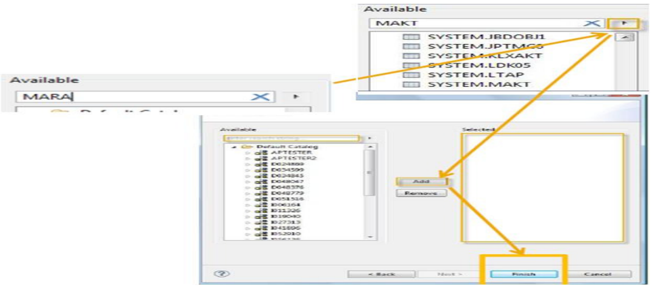
Modeling in HANA
Attribute View

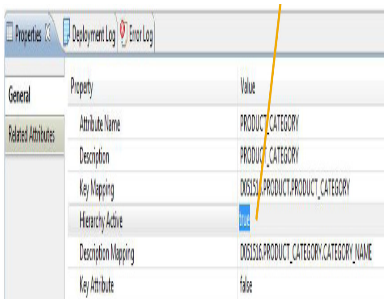
- Attributes add context to Data
- Attributes are modeled using Attribute Views
- Can be regarded as Master data Tables
- Can be linked to fact tables in Analytical views



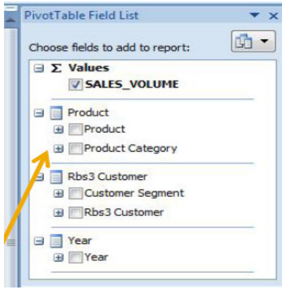

Set Parameters → Table Selection → Table Join and Properties → Select Attributes → Create Hierarchies → Save & Preview

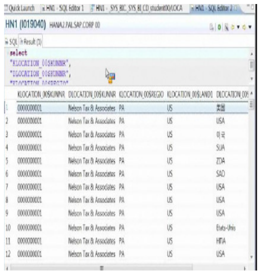






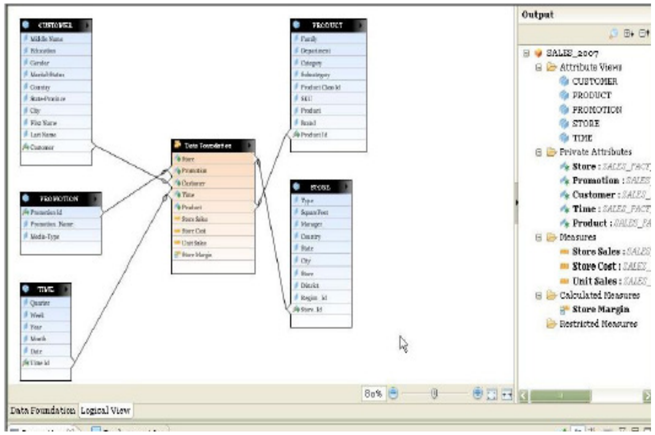
Property	Value
Attribute Name	PRODUCT_CATEGORY
Description	PRODUCT_CATEGORY
Key Mapping	D0515:PRODUCT.PRODUCT_CATEGORY
Hierarchy Active	true
Description Mapping	D0515:PRODUCT_CATEGORY.CATEGORY_NAME
Key Attribute	false

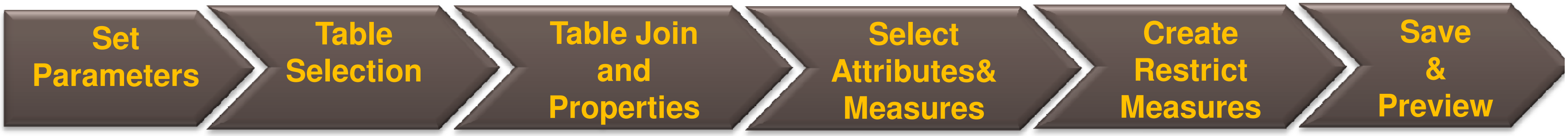


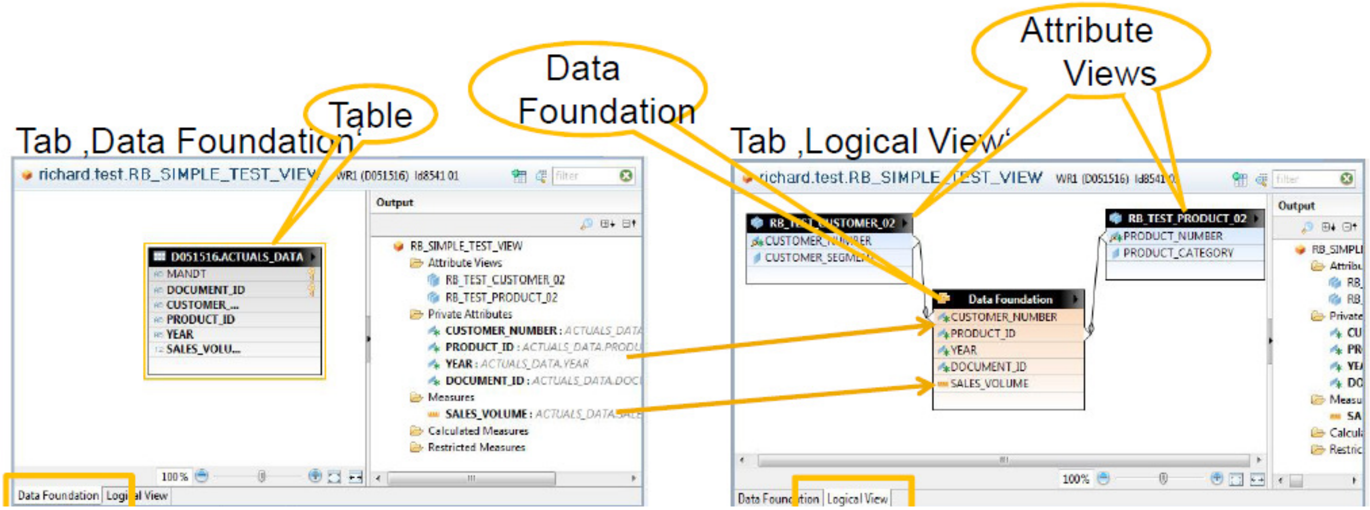


Modeling in HANA
Analytical View

- Analytical View can be regarded as Cube
  - Multidimensional modeling
  - Fact table joined against modeled dimensions
- Analytical Views do not store data
  - Data is read from the joined tables
  - Joins and calculated measures are evaluated in runtime
  - Master data for MDX/BICS are stored in system tables



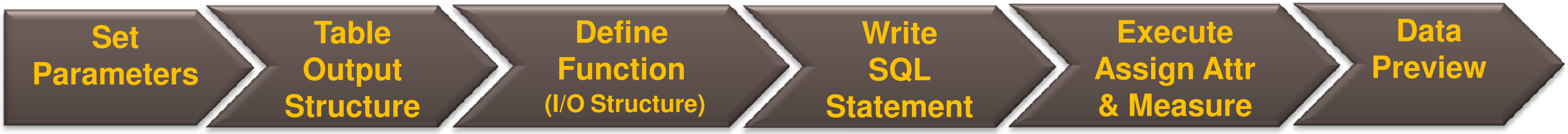

Set Parameters → Table Selection → Table Join and Properties → Select Attributes& Measures → Create Restrict Measures → Save & Preview



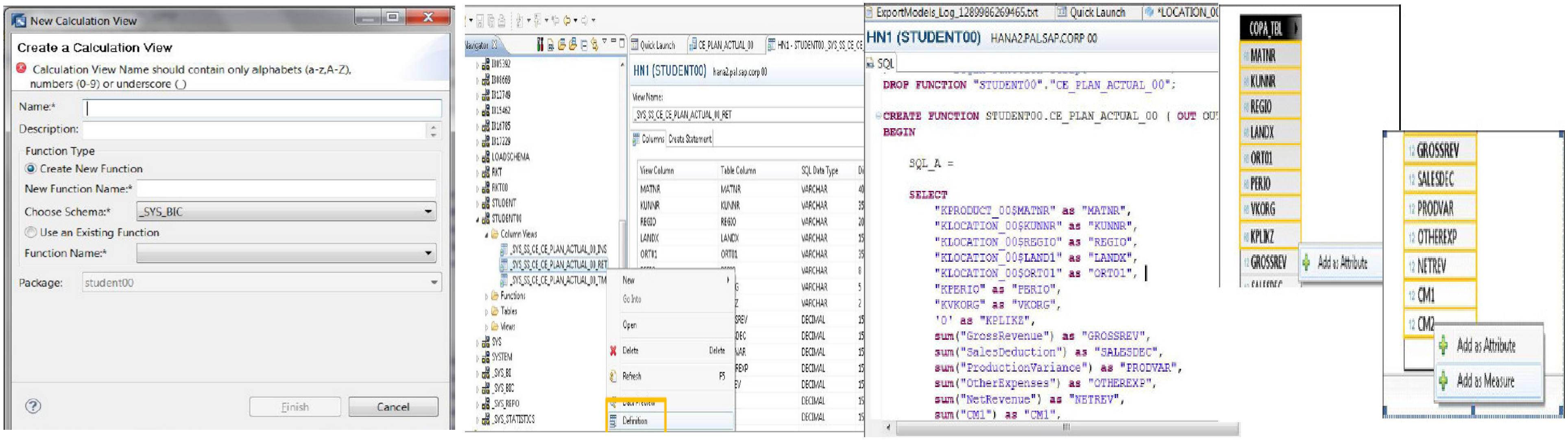
Modeling in HANA
Calculation View

Calculation view are used to create your own data foundation using database tables, attribute views, analytic views and calculation views to address a complex business requirement.

For example, Compare the sales of product in a particular region for the last two years.



Set Parameters → Table Output Structure → Define Function (I/O Structure) → Write SQL Statement → Execute Assign Attr & Measure → Data Preview



New Calculation View

Create a Calculation View

Calculation View Name should contain only alphabets (a-z,A-Z), numbers (0-9) or underscore (_)

Name:*

Description:

Function Type

● Create New Function

New Function Name:*

Choose Schema:* _SYS_BIC

○ Use an Existing Function

Function Name:*

Package: student00

HN1 (STUDENT00) hana2.pal.sap.corp 00

View Name: _SYS_SS_CE_CE_PLAN_ACTUAL_00_RET

View Column	Table Column	SQL Data Type	Di
MATNR	MATNR	VARCHAR	40
KUNNR	KUNNR	VARCHAR	35
REGIO	REGIO	VARCHAR	20
LANDX	LANDX	VARCHAR	15
ORT01	ORT01	VARCHAR	35
		VARCHAR	8
	G	VARCHAR	5
	Z	VARCHAR	2
	SREV	DECIMAL	15
	DEC	DECIMAL	15
	VAR	DECIMAL	15
	REXP	DECIMAL	15
	EV	DECIMAL	15
		DECIMAL	15
		DECIMAL	15

HN1 (STUDENT00) HANA2.PALSAP.CORP 00

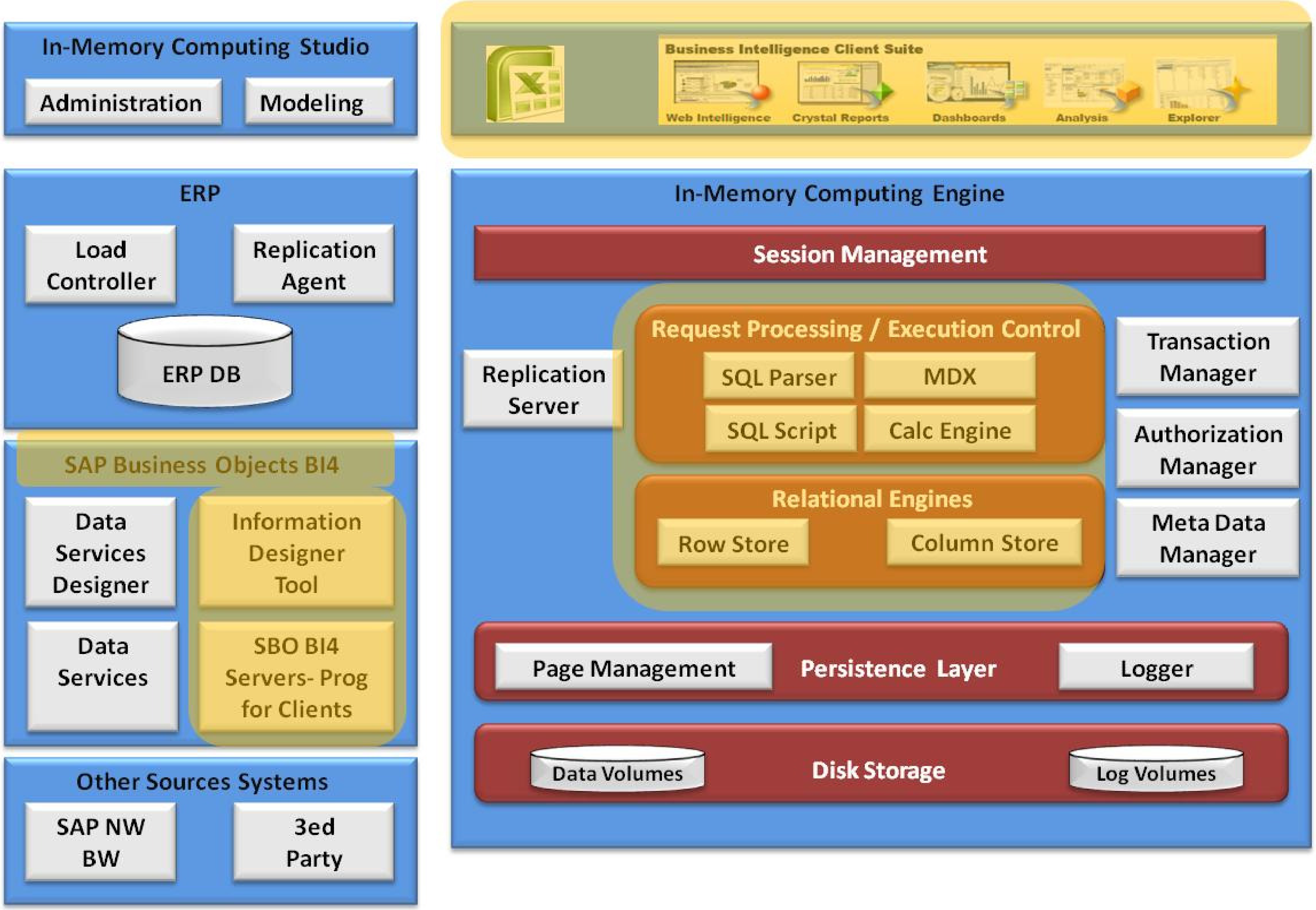
```
DROP FUNCTION "STUDENT00"."CE_PLAN_ACTUAL_00";

CREATE FUNCTION STUDENT00.CE_PLAN_ACTUAL_00 ( OUT OU
BEGIN

    SQL_A =

    SELECT
        "KPRODUCT_00$MATNR" as "MATNR",
        "KLOCATION_00$KUNNR" as "KUNNR",
        "KLOCATION_00$REGIO" as "REGIO",
        "KLOCATION_00$LAND1" as "LANDX",
        "KLOCATION_00$ORT01" as "ORT01",
        "KPERIO" as "PERIO",
        "KVKORG" as "VKORG",
        '0' as "KPLIKZ",
        sum("GrossRevenue") as "GROSSREV",
        sum("SalesDeduction") as "SALESDEC",
        sum("ProductionVariance") as "PRODVAR",
        sum("OtherExpenses") as "OTHEREXP",
        sum("NetRevenue") as "NETREV",
        sum("CM1") as "CM1",
```

© Mahindra Satyam 2011

Reporting
On
HANA

Reporting

Mahindra Satyam



In-Memory Computing Studio

Administration

Modeling

Business Intelligence Client Suite

Web Intelligence

Crystal Reports

Dashboards

Analysis

Explorer

ERP

Load Controller

Replication Agent

ERP DB

SAP Business Objects BI4

Data Services Designer

Information Designer Tool

Data Services

SBO BI4 Servers- Prog for Clients

Other Sources Systems

SAP NW BW

3ed Party

In-Memory Computing Engine

Session Management

Replication Server

Request Processing / Execution Control

SQL Parser

MDX

SQL Script

Calc Engine

Relational Engines

Row Store

Column Store

Transaction Manager

Authorization Manager

Meta Data Manager

Page Management

Persistence Layer

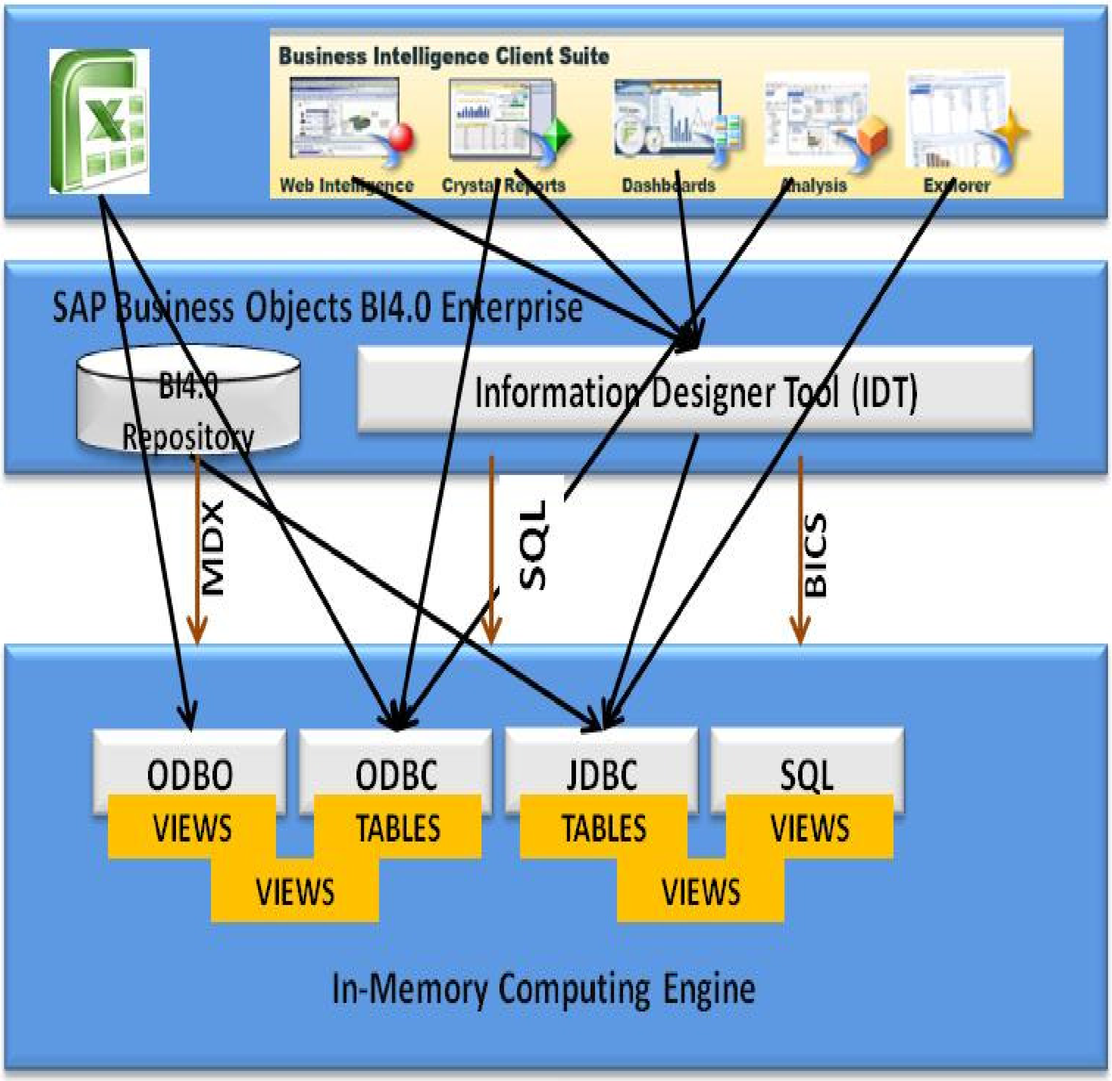
Logger

Data Volumes

Disk Storage

Log Volumes

Various Interface Reporting Options



Reporting Interface Options

ODBO-(OLEDB for OLAP)

- Microsoft-driven specification for multidimensional(cross-tab style)reporting
- Requests are sent to the database via **MDX**(Multi Dimensional eXpression language)

ODBC-(Open Database Connectivity)-

Microsoft-driven specification for relational reporting .Database requests are made via SQL(Structure Query Language)

JDBC –(Java Database Connectivity)-

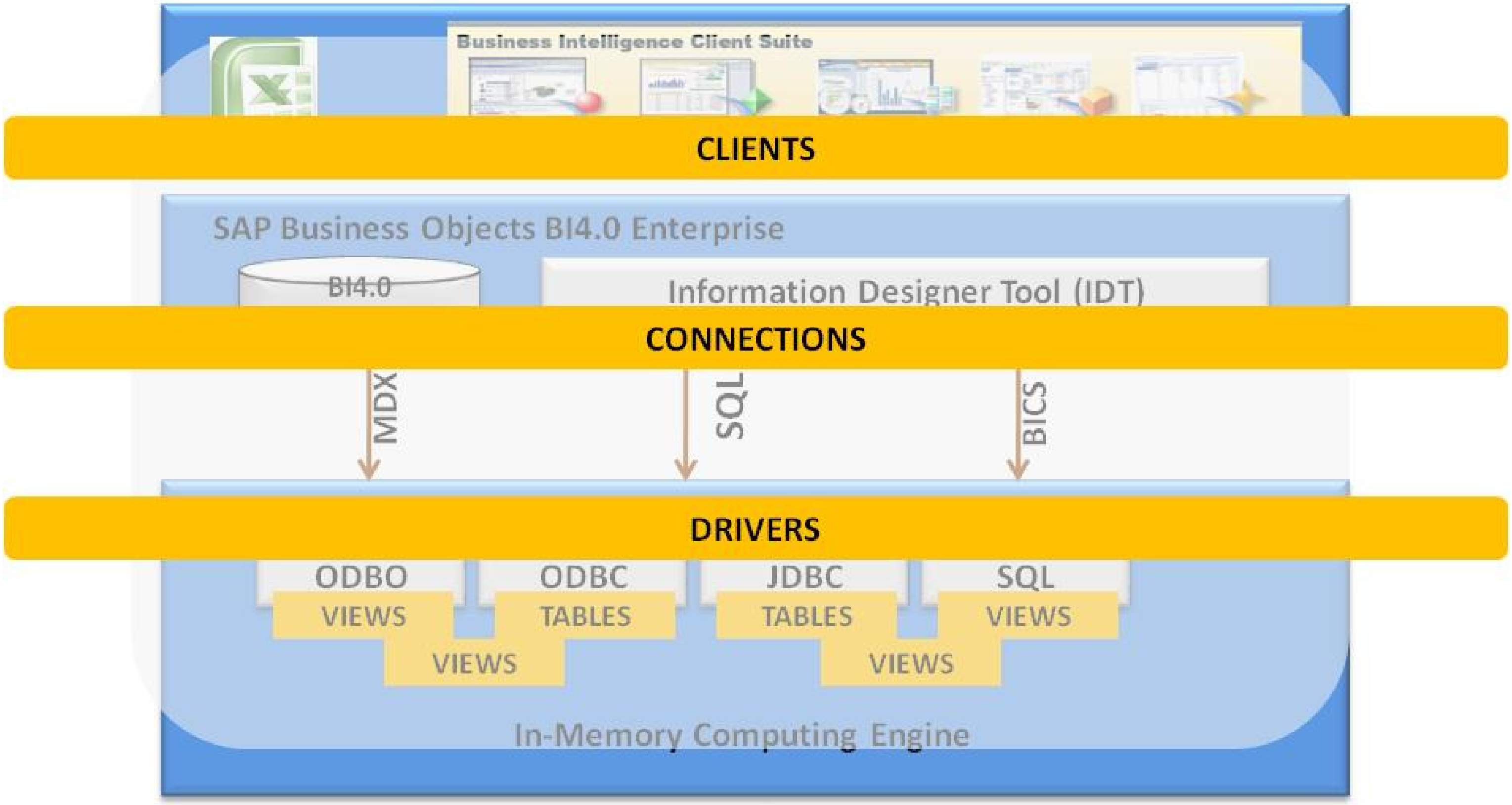
Relational reporting drivers specified by the Java community.

SQLDBC is SAP native database SDK

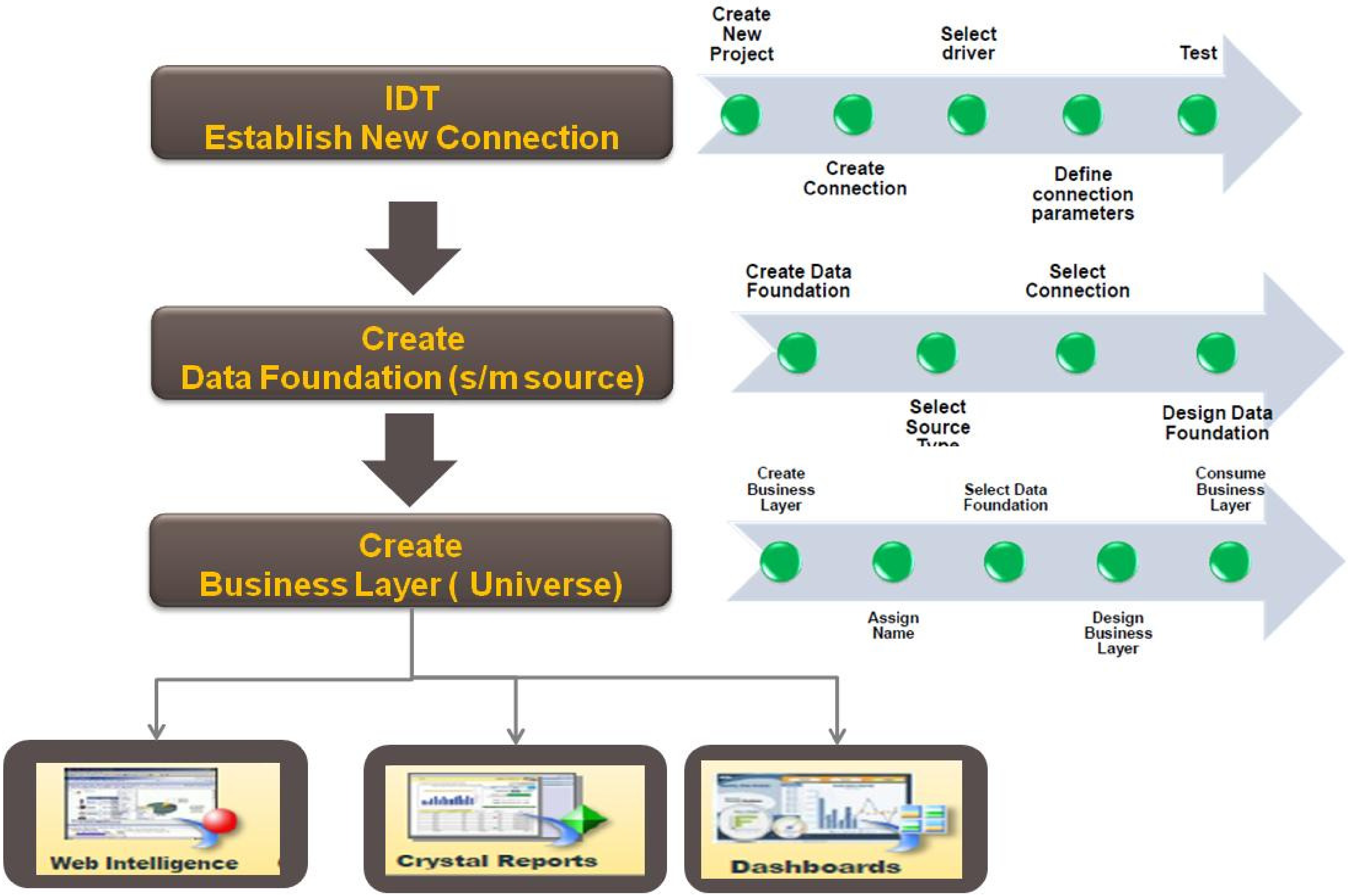
BICS-BI Consumer Services

- This is the common driver technology used by SAP Business Objects Analysis, Office Edition for connectivity to SAP NetWeaver BW

Reporting Interfaces


Business Intelligence Client Suite
CLIENTS
SAP Business Objects BI4.0 Enterprise
BI4.0
Information Designer Tool (IDT)
CONNECTIONS
MDX
SQL
BICS
DRIVERS
ODBO
ODBC
JDBC
SQL
VIEWS
TABLES
TABLES
VIEWS
VIEWS
VIEWS
In-Memory Computing Engine

Reporting on HANA with Business Objects 4.0 Information Designer Tool



IDT
Establish New Connection

Create New Project
Create Connection
Select driver
Define connection parameters
Test

Create
Data Foundation (s/m source)

Create Data Foundation
Select Source Type
Select Connection
Design Data Foundation

Create
Business Layer (Universe)

Create Business Layer
Assign Name
Select Data Foundation
Design Business Layer
Consume Business Layer

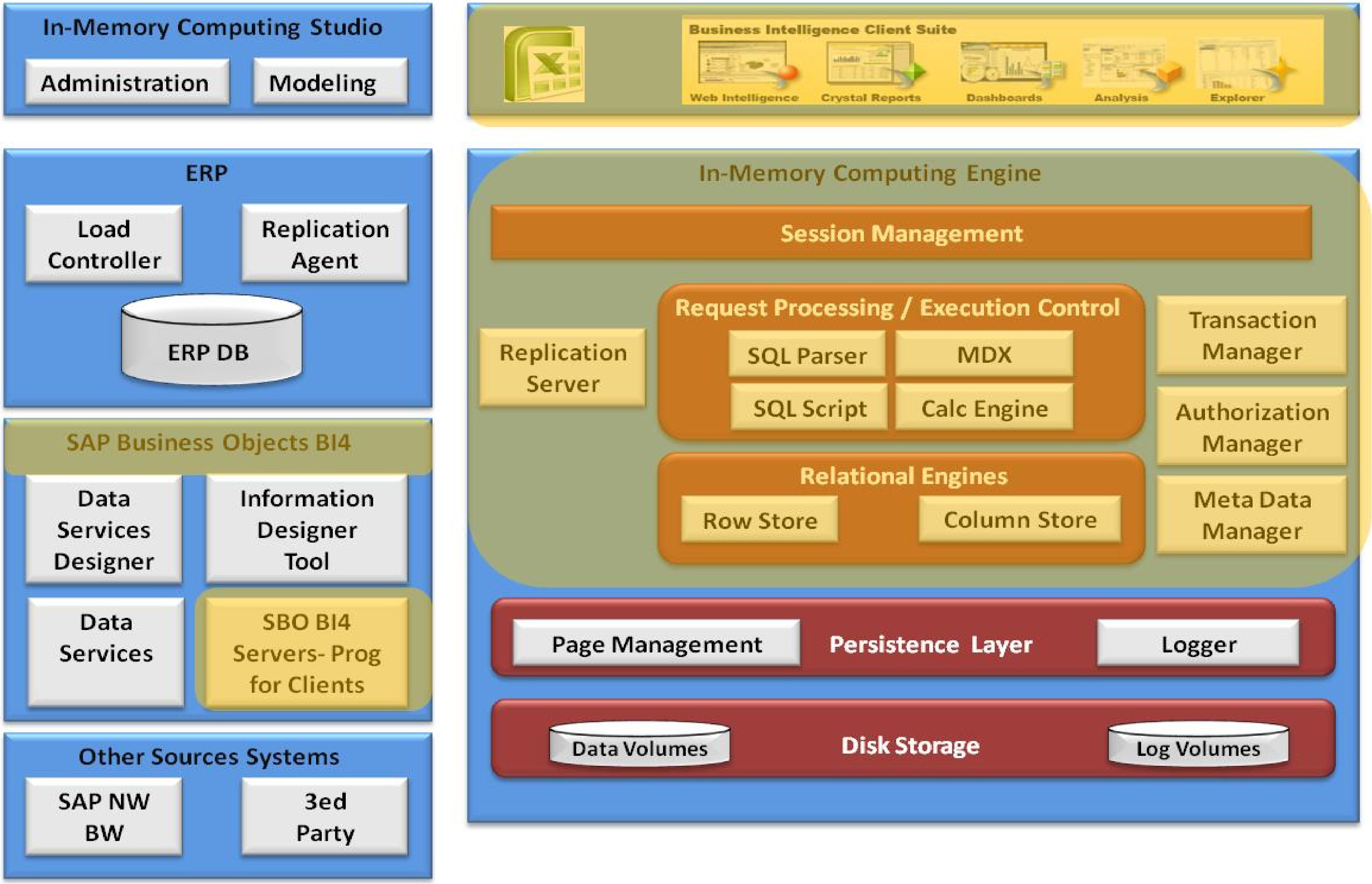
Web Intelligence

Crystal Reports

Dashboards

Request Processing
and
Execution

Request Processing and Execution



In-Memory Computing Studio

Administration

Modeling

Business Intelligence Client Suite

Web Intelligence

Crystal Reports

Dashboards

Analysis

Explorer

ERP

Load Controller

Replication Agent

ERP DB

SAP Business Objects BI4

Data Services Designer

Information Designer Tool

Data Services

SBO BI4 Servers- Prog for Clients

Other Sources Systems

SAP NW BW

3ed Party

In-Memory Computing Engine

Session Management

Replication Server

Request Processing / Execution Control

SQL Parser

MDX

SQL Script

Calc Engine

Relational Engines

Row Store

Column Store

Transaction Manager

Authorization Manager

Meta Data Manager

Page Management

Persistence Layer

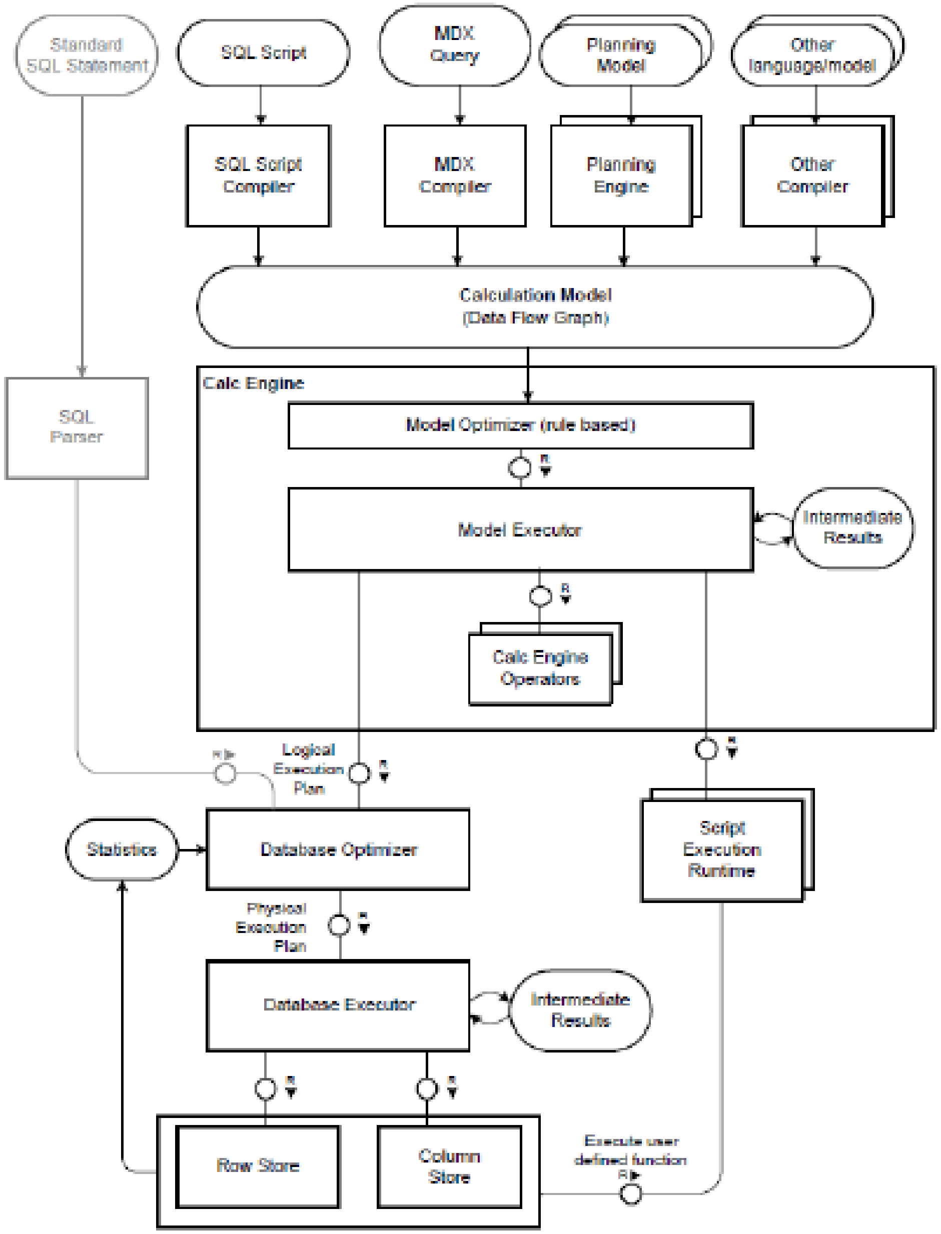
Logger

Data Volumes

Disk Storage

Log Volumes

Request Execution and Processing



Conceptual View

•Standard SQL

Processed directly by DB engine

SQL Script , MDX and planning engine interface

Domain-specific programming languages or models

Converted into calculation models

Calc Engine

Create logical execution plan for calculation models

Execute user defined functions

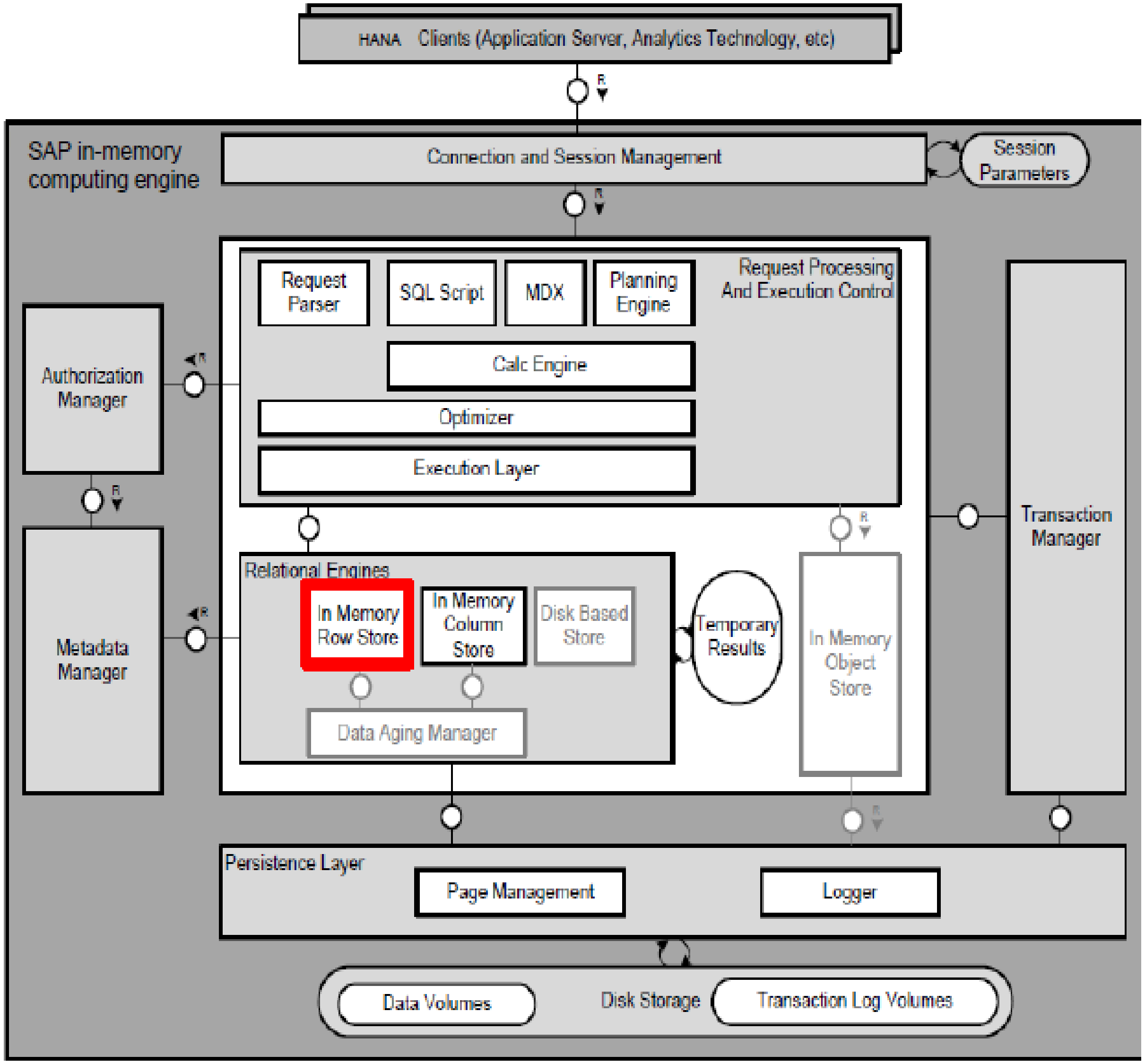
Relational engine

DB optimizer produces physical execution plan

Access to row and column store

Row Store

Row Store - High Level Architecture

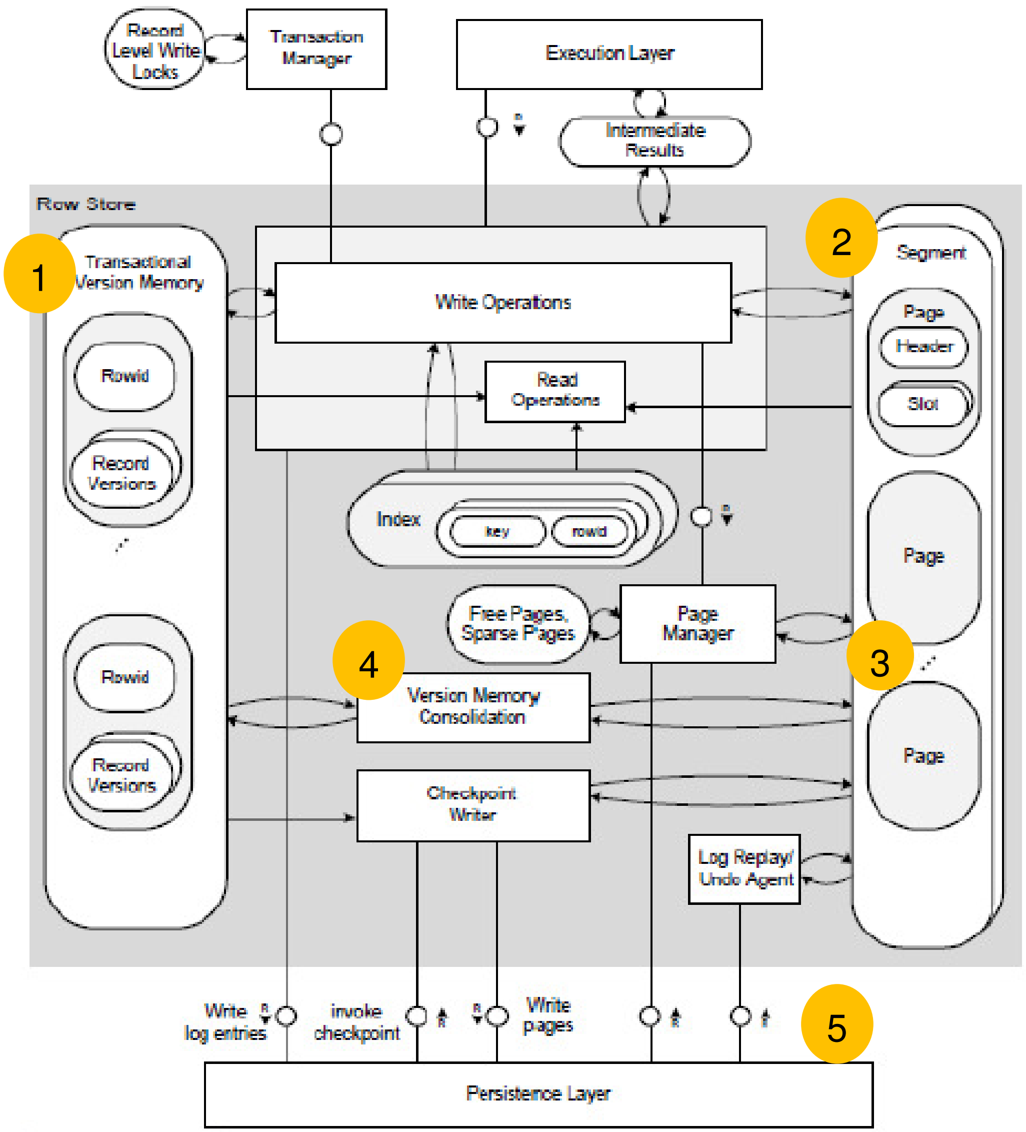


Row Store

- One of the relational engines
- Interfaced from calculation / execution layer
- Pure in-memory store
- Persistence managed in persistence layer

Row Store Architecture

Row Store - Block Diagram



Row Store Block Diagram

1 Transactional Version Memory

Contains temporary versions

Needed for Multi-Version Concurrency Control (MVCC)

2 Segments

Contain the actual data (content of row-store tables) in pages

3 Page Manager

Memory allocation

Keeping track of free/used pages

4 Version Memory Consolidation

Think „garbage collector for MVCC"

5 Persistence Layer

Invoked in write operations (log)

And in performing save points checkpoint writer

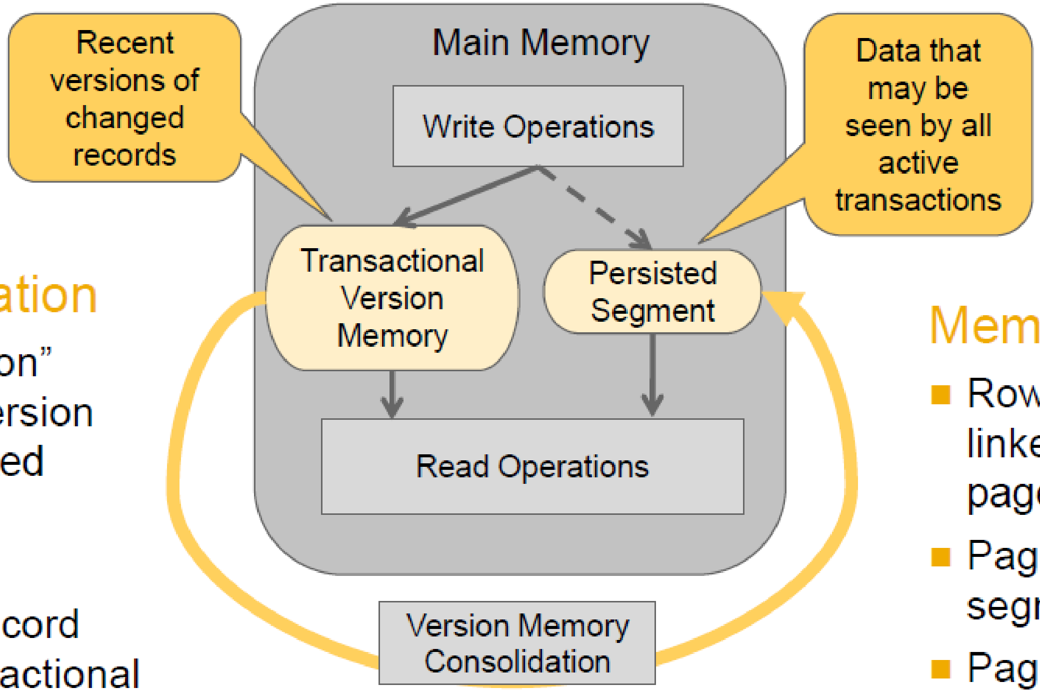
Row Store Architecture – Highlights

Write Operations

- Mainly go into “Transactional Version Memory”
- “INSERT” also writes to Persisted Segment

Persisted Segment

- Contains data that may be seen by any ongoing transaction
- Data that has been committed before any active transaction was started)


Recent versions of changed records
Main Memory
Write Operations
Data that may be seen by all active transactions
Transactional Version Memory
Persisted Segment
Read Operations
Version Memory Consolidation

Version Consolidation

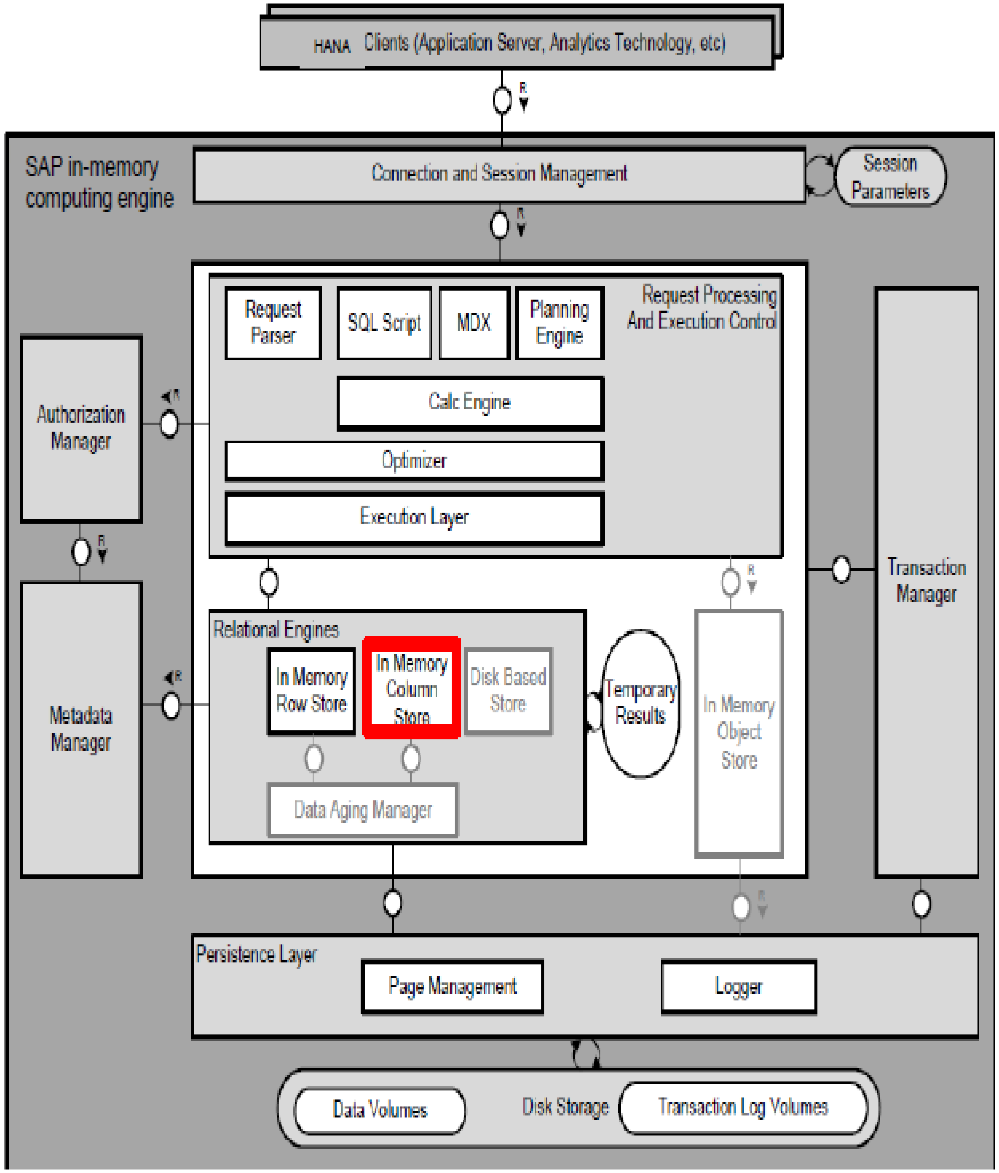
- Moves “visible version” from Transaction Version Memory into Persisted Segment (based on Commit ID)
- Clears “outdated” record versions from Transactional Version Memory

Memory Handling

- Row store tables are linked list of memory pages
- Pages are grouped in segments
- Page size: 16 KB

Column Store

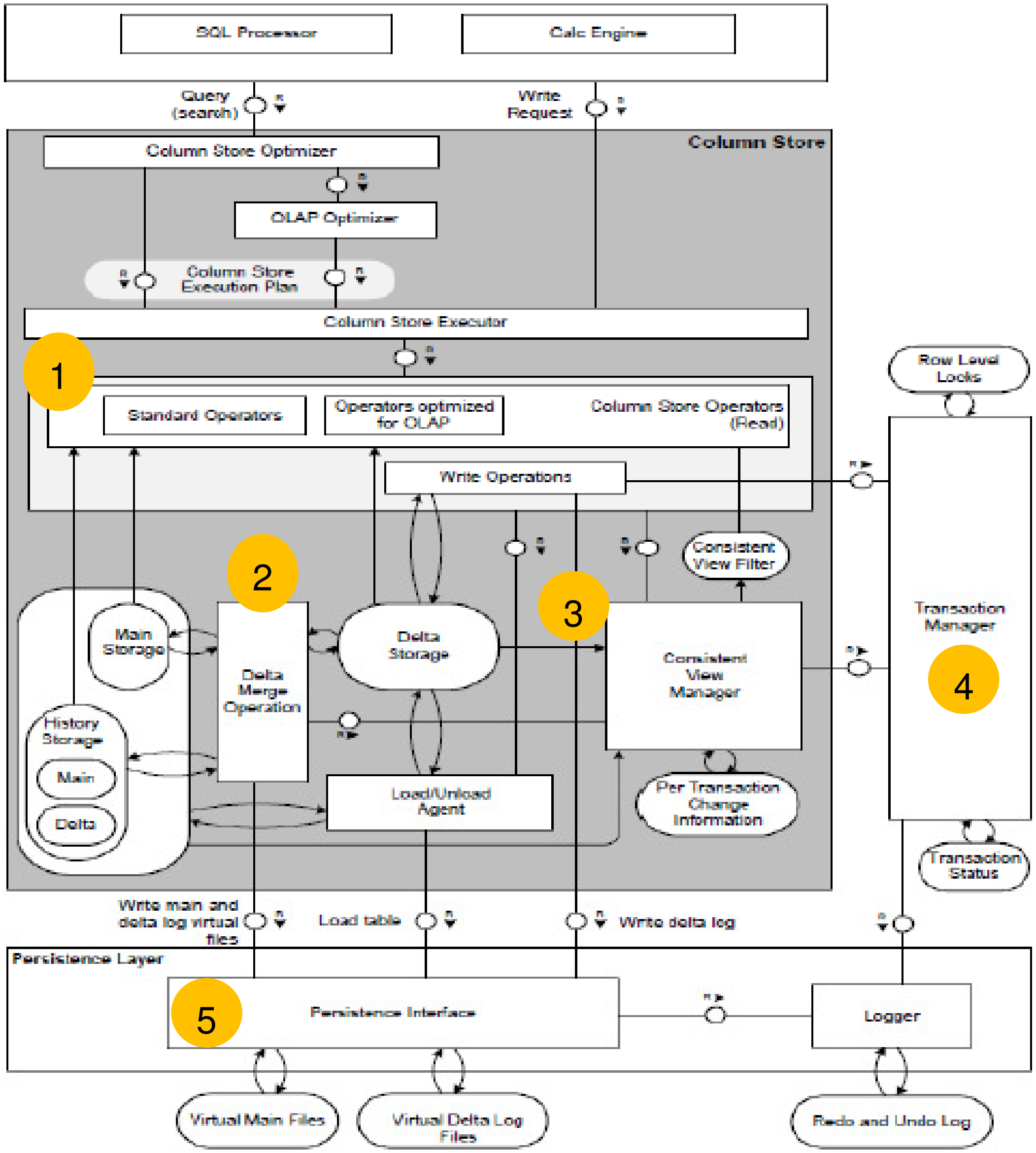
Column Store



Column Store

- One of the relational engines
- Interfaced from calculation / execution layer
- Pure in-memory store
  - Persistence managed in persistence layer
- Optimized for high performance of **read operation**
- Good performance of **write operations**
- Efficient data **compression**

Column Store Architecture



Column Store Block Diagram

Optimizer and Executor

1 Handles queries and execution plan

Main and Delta Storage

2 Compressed data for fast read
Delta data for fast write
Asynchronous delta merge

3 Consistent View Manager

4 Transaction Manager

Persistence Layer

5

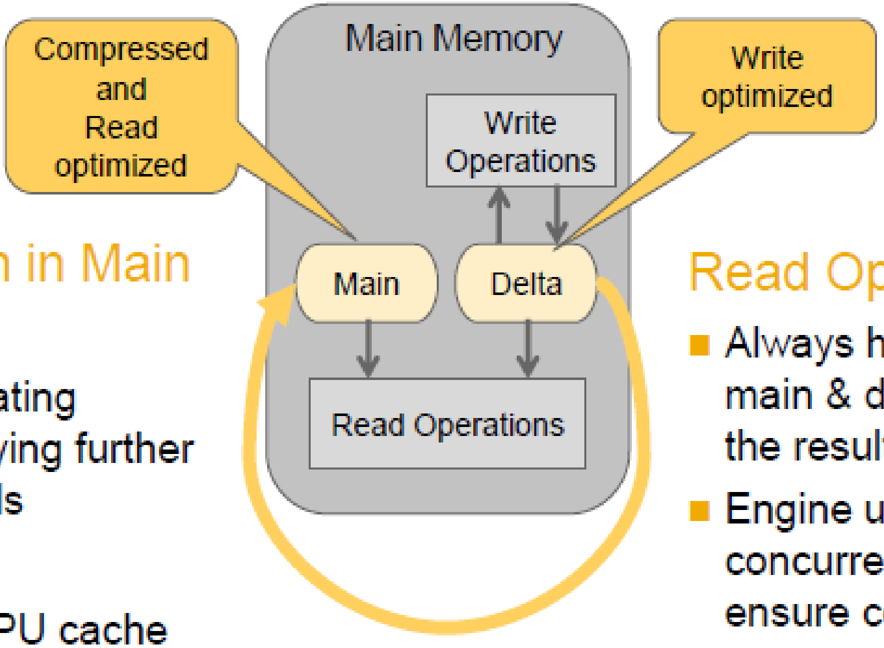
Column Store Architecture
– Highlights

Storage Separation (Main & Delta)

- Enables **high compression** and **high write performance** at the same time

Write Operations

- **Only** in **delta storage** because write optimized.
- The update is performed by **inserting** a new entry into the **delta storage.**



Data Compression in Main Storage

- Compression by creating **dictionary** and applying further compression methods
- Speed up
  - Data load into CPU cache
  - Equality check → Search
- The compression is computed during **delta merge operation.**

Read Operations

- Always have to read from **both** main & delta storages and merge the results.
- Engine uses multi version concurrency control (**MVCC**) to ensure consistent read operations.

Delta Merge Operation

- See next slide

Row Store Vs Column Store

Where to use which Store

Modeling Only Possible For Column Tables

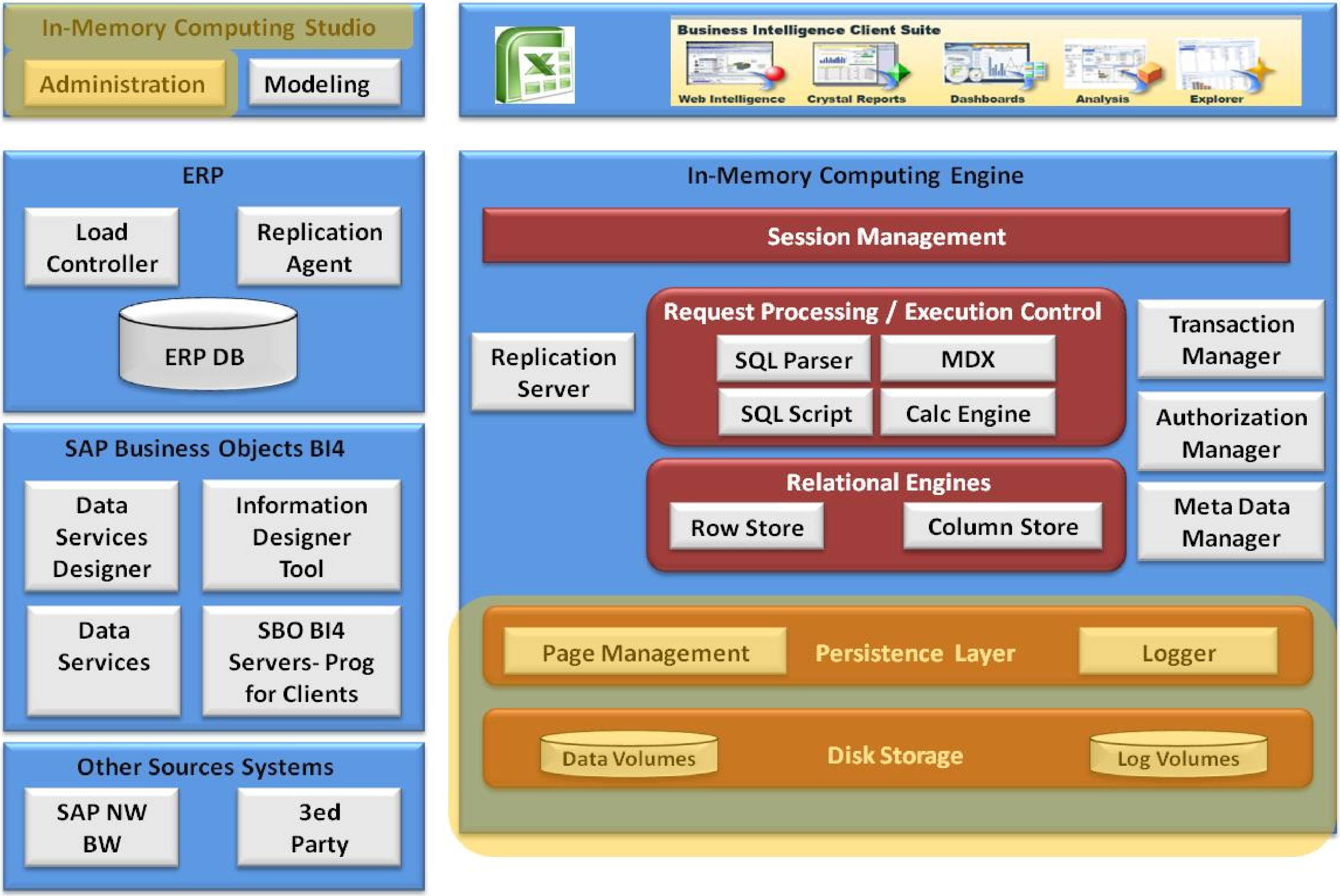
- This answers the frequently asked question:
  "Where should I put a table – row store or column store?"
  - Information Modeler only works with column tables
  - Replication server creates tables in column store per default
  - Data Services creates tables in column store per default
  - SQL to create column table: "CREATE COLUMN TABLE ..."
  - Store can be changed with "ALTER TABLE …"

System Tables Are Created Where They Fit Best

- Administrative tables in row store:
  - Schema SYS → caches, administrative tables of engine
  - Tables from statistics server
- Administrative tables in column store:
  - Schema _SYS_BI → metadata of created views + master data for MDX
  - Schema _SYS_BIC → some generated tables for MDX
  - Schema _SYS_REPO → e.g. lists of active/modified versions of models

Administration
(Persistence Layer)

Persistence layer



In-Memory Computing Studio

Administration

Modeling

Business Intelligence Client Suite

ERP

Load Controller

Replication Agent

ERP DB

SAP Business Objects BI4

Data Services Designer

Information Designer Tool

Data Services

SBO BI4 Servers- Prog for Clients

Other Sources Systems

SAP NW BW

3ed Party

In-Memory Computing Engine

Session Management

Replication Server

Request Processing / Execution Control

SQL Parser

MDX

SQL Script

Calc Engine

Relational Engines

Row Store

Column Store

Transaction Manager

Authorization Manager

Meta Data Manager

Page Management

Persistence Layer

Logger

Data Volumes

Disk Storage

Log Volumes

Persistence Layer

Purpose & Scope

Why Does An In-memory Database Need A Persistence Layer?

- Main Memory is volatile. What happens upon…
  - Database restart?
  - Power outage?
  - ...
  - → Data needs to be stored in a non-volatile way
- Backup and restore

- SAP in-memory computing engine offers one persistence layer which is used by row store and column store
  - Regular “savepoints”
    → full persisted image of DB at time of savepoint
  - Logs capturing all DB transactions since last savepoint (redo logs and undo logs written)
    → restore DB from latest savepoint onwards
  - Ability to create "snapshots"
    → used for backups

HANA Roadmap

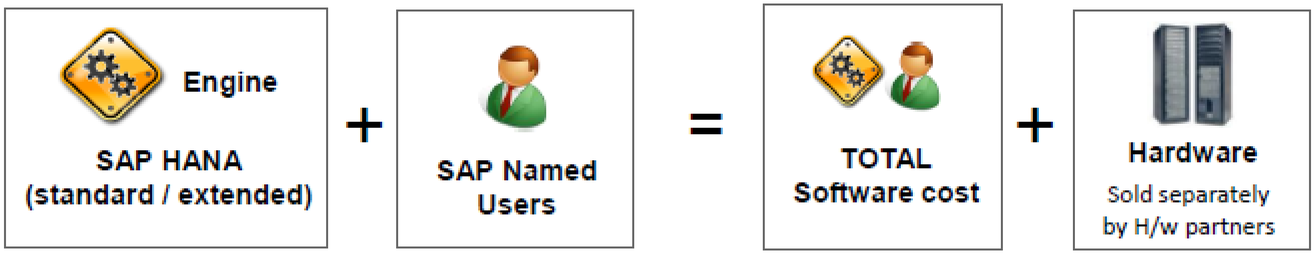
HANA Road Map

	HANA 1.0	“Innovation“ Mid-term	“Transformation” Longer Term
Capabilities	**In-Memory Analytics** • **HANA 1.0 Real-time operational analytics with HANA 1.0** • Complete BI Suite with BI 4.0 (Aurora) runs on HANA	**Next generation applications** • **SAP BW fully running on HANA** • HANA 1.5 platform for IM Apps • **Business Suite runs on HANA** • Further optimization of BI 4.1 Suite for HANA • Industry and LOB Business Analytics Solutions “BAS”	**One Store for Data and Analytics** • HANA only persistence layer for SAP Business Suite • **SAP Business Suite optimized for In-Memory**
Benefits	• Flexible real time analysis of operations at non-aggregated level	• Real-Time operational planning and simulation capabilities: link to execution • Primary persistence and optimized for SAP BW	• Reduced landscape complexity • Value chain transformation

HANA Licensing
&
T-Shirt Sizing

Licensable components

A : Required Licensable components



Engine

SAP HANA (standard / extended)

+

SAP Named Users

=

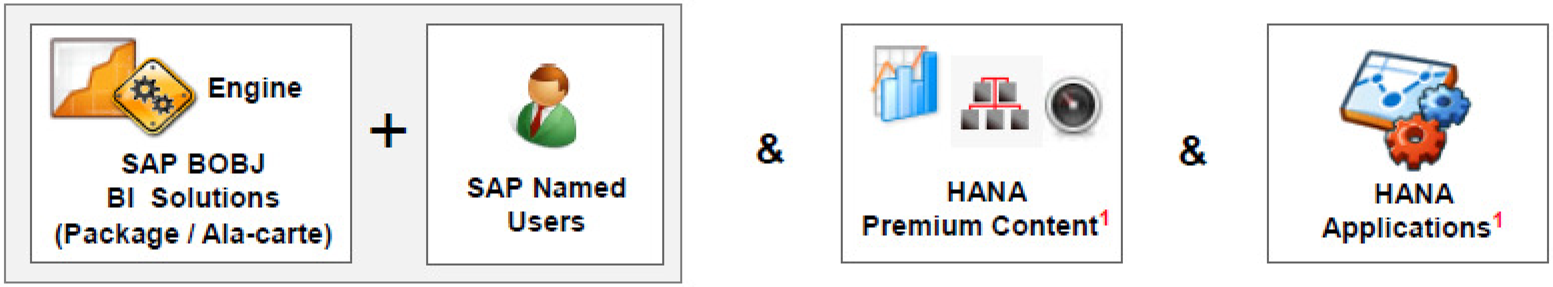
TOTAL Software cost

+

Hardware

Sold separately by H/w partners

B : Additional Licensable Components (when available)



Engine

SAP BOBJ BI Solutions (Package / Ala-carte)

+

SAP Named Users

&

HANA Premium Content[1]

&

HANA Applications[1]

Thank you

mahindrasatyam.com

Safe Harbor

This document contains forward-looking statements within the meaning of section 27A of Securities Act of 1933, as amended, and section 21E of the Securities Exchange Act of 1934, as amended. The forward-looking statements contained herein are subject to certain risks and uncertainties that could cause actual results to differ materially from those reflected in the forward-looking statements. We undertake no duty to update any forward-looking statements. For a discussion of the risks associated with our business, please see the discussions under the heading “Risk Factors” in our report on Form 6-K concerning the quarter ended September 30, 2008, furnished to the Securities and Exchange Commission on 07 November, 2008, and the other reports filed with the Securities and Exchange Commission from time to time. These filings are available at http://www.sec.gov