**SAP HANA Architecture Overview**

The SAP HANA Database was developed primarily in JAVA and C++, designed to run on the SUSE Linux Operating System. in the diagram below provides a high-level view of the SAP HANA Application Software and its associative functions:

Key architecture components include:

* **Client Applications**– SAP HANA can accommodate multiple clients across multiple instances to provide access to useful data across a number of related applications.
	+ SAP HANA can also connect via an optional module (XS Engine) to a Client Application via HTTP/S.  This is helpful for obtaining third-party data or data from disparate locations.
	+ It is currently not recommended to attach more than one ERP per instance of SAP HANA due to potential for table name replication (SAP HANA Database is a non-locking database).
* **Index Server** – The Index Server contains the actual data and engines to process the data. (This is where most of the SAP HANA magic happens. Learn more about the Index Server components below.)
* **Pre-Processor Server** – The Index Server utilizes the Pre-Processor Server to analyze text data and extract data from text when called via the Search function.
* **Statistical Server** – The Statistical Server will analyze and present status, performance and metrics from all SAP HANA Database components listed here, as well as the underlying hardware and OS.
* **Name Server** - The Topology of the SAP HANA Database is recorded here.  In a multi-server SAP HANA distributed landscape, the Name Server will record which data is on which server to decrease time associated with re-indexing. Below please find an example of what a multi-server SAP HANA Distributed Landscape would look like:

* **XS Engine** – Client applications can use HTTP to transmit data via the optional XS Engine which uses SAP ICM for HTTP server. This module provides access to the SAP HANA DB by transforming the persistence model stored in the DB into a consumption model for clients exposed via HTTP.  This allows an organization to host system services that are part of the SAP HANA Database (ex. Search service, built-in web server that provides access to static content in the REPOSITORY)

 **The SAP HANA Power House – More about the Index Server**

The SAP HANA Index Server contains what is arguably the majority of the magic behind SAP HANA. Below, we outline the inner-workings of the Index Server and how it powers robust SAP HANA analytics.

The SAP HANA Index Server performs 7 key functions to accelerate and optimize analytics. Together, these functions provide robust security and data protection and enhanced data access.

* **Connection and Session Management** – This component initializes and manages sessions and connections for the SAP HANA Database using pre-established **Session Parameters**. SAP has long been known for excellence in session management through its integration of SAPRouter into the SAPGUI product used as a front end for accessing the ABAP stack.  SAP HANA retains the ability to configure Connection and Session management parameters to accommodate complex security and data transfer policies instituted.
* **Authentication**– User and role-based privileges are authenticated by the SAP HANA Database. (The Users, Authorizations and Roles within the SAP ERP system are not applicable or transportable to the SAP HANA instance.) The SAP HANA authentication model allows granting of privileges to users or roles, and a privilege grants the right to perform a specified SQL operation on a specific Object. SAP HANA also utilizes a set of **Analytic Privileges** that represent filters or hierarchy drilldown limitations for analytic queries to protect sensitive data from unauthorized users. This model enforces “Segregation of Duty” for clients that have regulatory requirements for the security of data.
* **SQL Processor**– The SQL Processor segments data queries and directs them to specialty query processing engines for optimized performance. It also ensures that SQL statements are accurately authored and provides some error handling to make queries more efficient. The SQL processor contains several engines and processors that optimize query execution:
	+ The **Multidimensional Expressions** (MDX) **Engine**is queries and manipulates the multidimensional data stored in OLAP (OnLine Analytical Processing) data cubes.
	+ The **Planning Engine** enables the basic planning operations within the SAP HANA Database for financial planning operations.
	+ The **Stored Procedure Processor** executes procedure calls for optimized processing without reinterpretation.  (e.g. converting a standard InfoCube into an SAP HANA Optimized Infocube)
	+ The **Calculation Engine** converts data into **Calculation Models** and creates a Logical Execution Plans to support parallel processing.
* **Relational Stores**– SAP has further segmented the storage of In-Memory data into compartments within memory for speedier access.  Data not needed immediately is stored on a Physical Disk as opposed to RAM.  This allows quick access to the most relevant data. The SAP HANA Database houses four relational stores that optimize query performance:
	+ The **Row Store**stores data in a row-type fashion and is optimized for high performance of write operation, and is derived from the P-Time "In Memory System" which was acquired by SAP in 2005.  The Row Store is held fully in RAM.
	+ The **Column Store**stores data in a column-type fashion and is optimized for high performance of write operation, and is derived from **TREX** (Text Retrieval and Extraction)  which was unveiled by SAP in the SAP NetWeaver Search and Classification product.  This technology was further developed into a full relational column based store.  The Column Store is held fully in RAM.
	+ The **Object Store** is an integration of SAP Live Cache Technology into the SAP HANA Database.
	+ The **Disk Based Store** is used for data that does not need to be held in memory and is best used for "tracing data" or old data that is no longer used.  Disk Based Store is located on a hard disk is pulled into RAM as needed.
* **Transaction Manager**– The SAP HANA Database processes individual SQL statements as transactions.  The **Transaction Manager** controls and coordinates transactions and sends relevant data to appropriate engines and to the Persistence Layer. This segmentation simplifies administration and troubleshooting.
* **Persistence Layer**– The **Persistence Layer** provides built-in disaster recovery for the SAP HANA Database. The algorithms and technology is based on concepts pioneered by MAX DB and ensures that the database is restored to the most recent committed state after a planned or unplanned restart.  Backups are stored as **Save Points** in the **Data Volumes** via a **Save Point Coordinator** which is typically set to backup every five to ten minutes.  Any change points that occur after a save point are designated as un-committed transactions and are stored in the **Transaction Log Volume**.  Typically, these volumes are saved to media and shipped offsite for a cold-backup disaster recovery remedy.
* **Repository**- The Repository manages the versioning of Metadata Objects such as Attribute, Analytic Views and Stored Procedure.  It also enables the import and export of Repository content.