ZERO DEFECTS Manufacturing Platform ZDMP

ZDMP Blog

Storage in Industry 4.0

By Mircea Vasile, Technical Leader, SIMAVI

Some questions for you

- What do you think is the best approach for storage that supports multiple heterogenous applications?
- How do you see a solution for distributed processing of large data sets across clusters of computers?
- How can an API Gateway solution be implemented for accessing multiple storage services?

Why: purpose and motivation

The complexity of applications running under Industry 4.0 paradigm suggests a heterogeneous storage solution, which should meet the space and performance requirements necessary to fulfil the functionalities necessary to reach the objectives and KPIs proposed. That means implementing a hybrid data management solution to meet Big Data challenges and drive new levels of real-time analytics. A highly scalable environment is needed that supports extremely large data volumes, accepting data in its native format from a variety of data sources.

Thus, the most suitable storage solutions are these so-called data-lakes, which include high-performance applications for storing and manipulating files of all kinds (binary, test, structured, semi-structured, images, video, data collections), SQL databases, NoSQL databases, and application components running as microservices.

The main benefit of a data lake is the centralisation of disparate content sources. Once gathered, these sources can be combined and processed using Big Data, search, and analytics techniques which would have otherwise been impossible. The disparate content sources often contain proprietary and sensitive information which requires implementation of the appropriate security measures in the data lake.

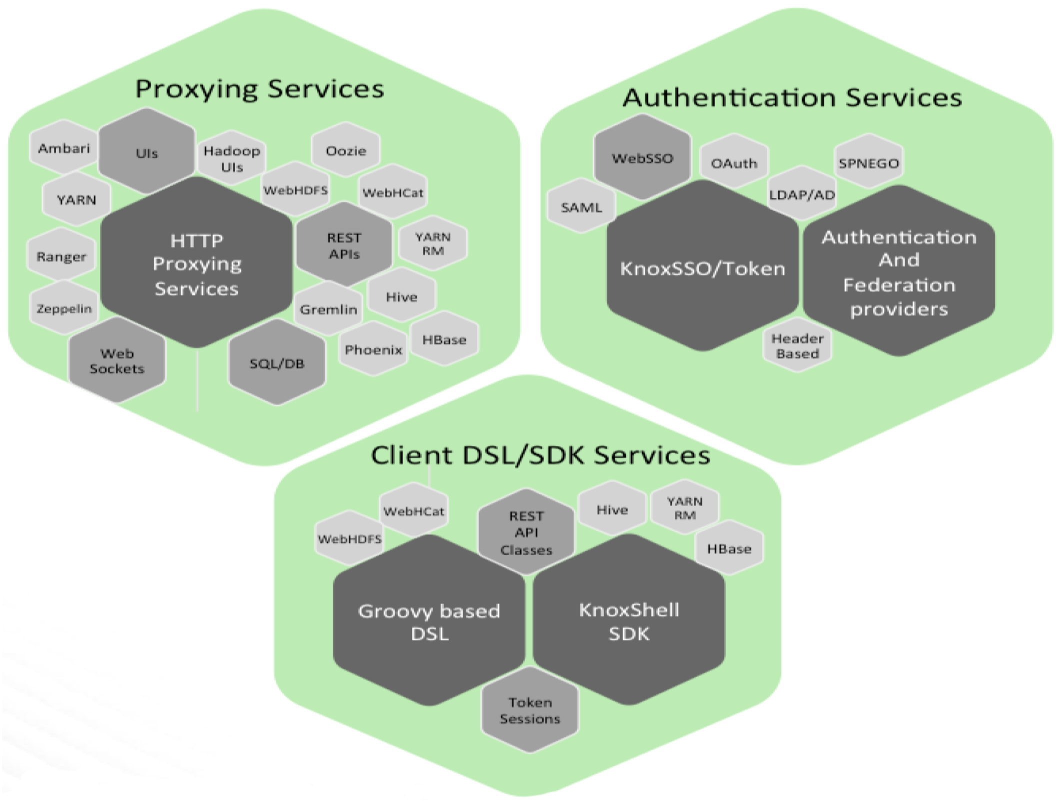
What will ZDMP achieve?

In ZDMP, the functionalities related to the files management are implemented using the Apache™ Hadoop® project[1], an open-source software for reliable, scalable, distributed computing. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale-up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly available service on top of a cluster of computers, each of which may be prone to failures.

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. HDFS relaxes a few POSIX requirements to enable streaming access to file system data. HDFS was originally built as infrastructure for the Apache Nutch web search engine project. HDFS is part of the Apache Hadoop Core project.

For managing the databases in ZDMP it is implemented Apache HBase™ [2], that ensures random, real-time read/write access to " Big Data". This project's goal is the hosting of very large tables -- billions of rows X millions of columns -- atop clusters of commodity hardware. Apache HBase is an open-source, distributed, versioned, non-relational database modelled after Google's Bigtable [3]. Just as Bigtable leverages the distributed data storage provided by the Google File System, Apache HBase provides Bigtable-like capabilities on top of Hadoop and HDFS.

The API that allows access to these Apache Hadoop services is Apache Knox. The Apache Knox™ [4] Gateway is an Application Gateway for interacting with the REST APIs and UIs of Apache Hadoop deployments. The Knox Gateway provides a single access point for all REST and HTTP interactions with Apache Hadoop clusters.



ZDMP Links

• **Architecture Component(s)**	Storage
• **Work Package**	WP6
• **Tasks**	T6.2 Secure Business Cloud

References/Acknowledgements

- [1] http://hadoop.apache.org/
- [2] https://hbase.apache.org/
- [3] Fay Chang Jeffrey Dean Sanjay Ghemawat Wilson C. Hsieh Deborah A. Wallach Mike Burrows Tushar Chandra Andrew Fikes Robert E. Gruber. Bigtable: A Distributed Storage System for Structured Data https://research.google/pubs/pub27898/
- [4] https://knox.apache.org/