

Data Acquisition as Industrial Cloud service

Ivan Stankevichus

Bachelor's thesis

March 2016

Technology, communication and transport

Degree Programme in Automation Engineering

Author(s) Stankevichus, Ivan	Type of publication Bachelor's thesis	Date 01.03.2016
	Number of pages 38	Language of publication: EN Permission for web publication: x
Title of publication Data acquisition as Industrial Cloud Service		
Degree programme Automation Engineering		
Supervisor(s) Rantapuska, Seppo , Flyktman Teppo		
Assigned by Siemens Osakeyhtiö		
Abstract <p>The aim of this thesis was to develop a practical solution for industrial remote monitoring of automation equipment including real-time data acquisition from Siemens automation devices and storage of the obtained data to the SAP HANA Cloud database. Because HANA will be implemented by Siemens as a Cloud service, it was also reasonable to introduce important features and benefits of HANA technology for the industrial sector.</p> <p>The implementation of data acquisition solution was divided into three sections. The first part consists of a small research about HANA features and benefits and the search for technologies for data acquisition application. The second part includes the development of remote monitoring interface and data acquisition application. Finally, s120 frequency converter was used for testing and demonstration purposes of the developed remote monitoring solution.</p> <p>As a result, UA Connector application was developed as a service for data acquisition using web technologies. The data acquisition application uses the OPC UA technology to get values of Tags from s7-1500 programmable logic controller or WinCC RT Advanced application in real-time. The application was developed using Nodejs, Angular.js and WebSocket technology. The application complies with assigned requirements.</p> <p>Additionally, the remote monitoring interface was developed using HANAXS and SAPUI5 tools. Via the user interface power unit's cooling fan the operating hours of the s120 frequency converter can be monitored as one example.</p> <p>Because the data acquisition application can be provided as a service and SAP HANA is also a service, the presented solution is very close to the Internet of Services concept.</p>		
Keywords/tags (subjects) Cloud service, Industrial Internet of Things, Industry 4.0, In-memory database, SAP HANA, data acquisition.		
Miscellaneous		

Tekijä(t) Stankevichus, Ivan	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä 01.03.2016
	Sivumäärä 38	Julkaisun kieli Englanti
		Verkkojulkaisulupa myönnetty: x
Työn nimi Tiedonkeruu pilvipalveluna		
Tutkinto-ohjelma Automaatiotekniikan koulutusohjelma		
Työn ohjaaja(t) Seppo Rantapuska, Teppo Flyktman		
Toimeksiantaja(t) Siemens Osakeyhtiö		
Tiivistelmä <p>Opinnäytetyön tavoitteena oli kehittää käytännöllinen teollisuusautomaatiolaitteiston etävalvontaratkaisu, joka sisältää reaaliaikaisen tiedonkeruun Siemensin automaatiolaitteistosta ja saadun datan varastoinnin SAP HANA -pilvitietokantaan. Koska HANA tekniikka tulee käyttöön Siemensin pilvipalveluna, niin oli myös järkevä esittää tietokannan tärkeimmät ominaisuudet ja edut, jotka tämä tekniikka tuo teollisuussektorille yleisemminkin.</p> <p>Työn toteutus sisältää kolme pääosaa. Ensimmäinen osa on tutkimus HANA -pilvitietokannan ominaisuuksista ja hyödyistä sekä toteutustekniikan kartoitus tiedonkeruusovellusta varten. Toiseen vaiheeseen sisältyi tiedonkeruusovelluksen ja etävalvonnan käyttöliittymän kehittäminen. Lopuksi suunniteltiin ja otettiin käyttöön s120 -taajuusmuuttajalle kehitetty etävalvontaratkaisu testausta ja esittelyä varten.</p> <p>Tuloksena on saatu UA CONNECTOR -tiedonkeruusovellus, joka on toteutettu palveluna käyttäen web-tekniikkaa. Tiedonkeruusovellus käyttää OPC UA -standardiliityntää, jolla saadaan Tagien arvot s7-1500- ohjelmoitavasta logiikasta tai WinCC RT Advanced sovelluksesta reaali-ajassa. Tiedonkeruusovellus on toteutettu käyttäen mm. Nodejs, Angular.js ja WebSocket -tekniikkoja. Sovellus on toimeksiantajan vaatimusten mukainen.</p> <p>Lisäksi oli kehitetty etävalvontakäyttöliittymä käyttäen HANAXS- ja SAPUI5- työkaluja. Käyttöliittymässä valvotaan s120 -taajuusmuuttajan virtalähdetyksikössä olevan jäähdystysolettimen käyttötunteja. Koska kehitetty tiedonkeruusovellus voidaan tarjota palveluna ja SAP HANA itsestään toimii palveluna, niin lopuksi on esitetty ratkaisu, joka on hyvin lähellä Internet of Services -konseptia.</p>		
Avainsanat (asiasanat) Pilvipalvelu, Industrial Internet of Things, Industry 4.0, In-memory database, SAP HANA		
Muut tiedot		

Contents

1	Introduction	8
1.1	Preface	8
1.2	Goals.....	10
1.3	About Siemens Osakeyhtiö	11
2	Background of the HANA platform	11
2.1	Overview of the HANA platform as a service	11
2.2	In-memory processing engine	13
2.2.1	Data storage formats	13
2.2.2	Compression.....	14
2.2.3	Data partitioning	15
2.2.4	Delta buffer	15
2.3	Service security overview.....	16
2.3.1	Data layer security.....	16
2.3.2	Communication layer security	16
3	Background of the data acquisition service.....	17
3.1	HANA Connector	17
3.2	OPC UA.....	18
3.3	Other tools and technologies.....	19
3.3.1	Node.js.....	19
3.3.2	MongoDB.....	19
3.3.3	WebSocket	19
3.3.4	Angular.js.....	19
4	Practical implementation.....	20
4.1	Assignments	20

4.2	Requirements.....	20
4.3	Data acquisition service development.....	20
4.4	Results.....	21
4.4.1	Overview	21
4.4.2	Architecture.....	23
4.4.3	Connector’s interface	24
4.4.4	Source code (Confidential).....	25
4.4.5	Cloud system structure	25
5	Demo Case	26
5.1	Purpose	26
5.2	Tools.....	26
5.2.1	TIA portal.....	26
5.2.2	Simatic WinCC	26
5.2.3	STARTER.....	26
5.2.4	HANA XS and SAPUI5.....	27
5.3	Results.....	27
5.3.1	Overview	27
5.3.2	Preparation.....	28
5.3.3	Interfaces.....	29
6	Conclusions	31
6.1	Cloud service benefits.....	31
6.2	Further Development.....	33
6.3	Results.....	33
7	Discussion.....	33
	References	35

Appendices.....	37
Appendix 1. UA CONNECTOR service dependencies.....	37
Appendix 2. UA CONNECTOR web user interface dependencies.	38

Figures

Figure 1: Industrial revolution process (SAP 2015).....	8
Figure 2: Cyber-Physical System concept (Karnouskos 2013).	9
Figure 3: SAP HANA cloud platform overview (Schneider 2013).	12
Figure 4: The architecture of HANA DBMS (Fäber 2012).	12
Figure 5: Columnar and row storage layouts (Plattner 2015).	13
Figure 6: Database compression example	14
Figure 7 Write/Read operations on HANA database.....	15
Figure 8: HANA connector communication via HTTP(s) protocol.....	17
Figure 9: Cloud service connectors and adapters.....	18
Figure 10: Opened DB tunnel to HANA trial instance in neo-java-sdk console.....	22
Figure 11: Example: Data transfer schema	23
Figure 12: UA CONNECTOR 0.1.0 Architecture.....	24
Figure 13: UA CONNECTOR User Interface	24
Figure 14: Cloud system schema - interaction between services	25
Figure 15: HANA XS and SAPUI5 (SAP SCN 2014)	27
Figure 16: Trend line for \$\$S120_PU_FAN tag on embedded view in UA CONNECTOR (local)	29
Figure 17: Trend line for \$\$S120_PU_FAN tag on HANA Cloud platform.	30
Figure 18: Factory Remote Monitoring Home Page	30
Figure 19 : Service usage example	32
Figure 20: Gartner's 2015 Hype Cycle for Emerging Technologies (Gartner. 2015)	34

Tables

Table 1: List of available data adapters.	18
Table 2: Demo components.....	28

Abbreviations

DB – Database

DBMS – Database Management System

FTP – File Transfer protocol

HANA – High-Performance Analytic Appliance

HCP – HANA Cloud platform

HDB –HANA database

HDD – Hard Disk Drive

HMI – Human Machine Interface

IoT – Internet of Things

IP – Internet Protocol

JDBC – Java Database Connectivity

JSON – JavaScript Object Notation

LTS – Long Time Support

PLC – Programmable Logic Controller

SDK – Software Development Kit

SSD – Solid State Drive

SSL – Secure Sockets Layer

SQL – Structured Query Language

TCO – Total Cost of Ownership

TSL – Transport Layer Security

UI – User Interface

Terminology

API

API, short for "Application programming interface" is a set of functionalities. API is used as common boundary between two applications allowing them to communicate with one another.

HTML

HyperText Markup Language is the standard markup language to create structure and content of web pages. HTML5 is the latest version of the language.

HTTP

Hypertext Transfer Protocol is a protocol that is used to transfer data between browser clients and web services.

NoSQL

NoSQL is used to describe all non-relational document databases. NoSQL databases mainly are not using SQL and have other than tabular relations.

REST

REST, short for "Representation State Transfer", is an architectural model based on HTTP-protocol for creating application programming interfaces.

Web Service

A Web Service is a type of API that operates over HTTP.

1 Introduction

1.1 Preface

New technologies and innovative solutions are essential for industrial and economic growth, thus development is a continuous process in an industry. The history of the industrial development presents a few important revolutionary steps. The first significant phase of the development process began at the end of the 18th century with the speed up production by the adaptation of steam and hydropower. In the 20th century the invention of electricity enabled industrial mass production. The third improvement was unlocked by the invention of computers in the early 1970^s, which enabled an industrial automation driven by programmable controllers as seen up to now.

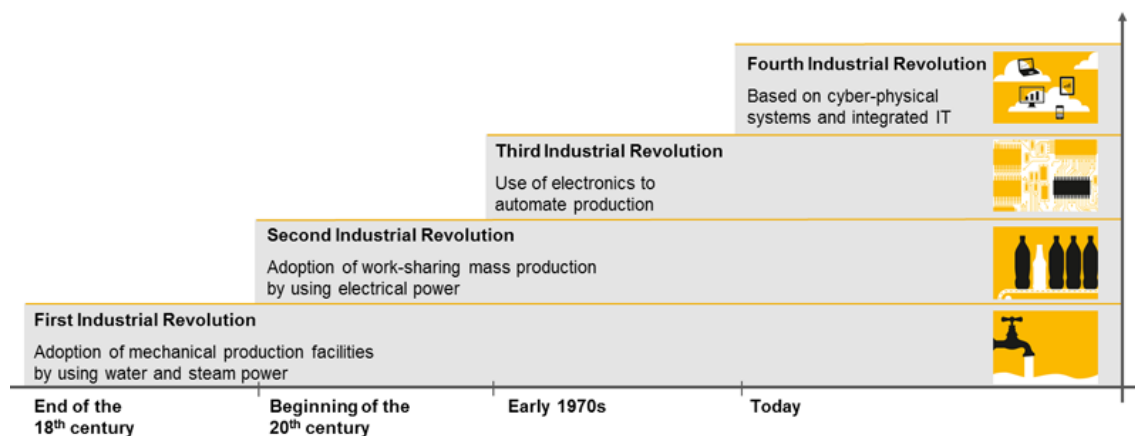


Figure 1: Industrial revolution process (SAP 2015)

The fourth industrial revolution began with many significant improvements in information and communication technologies. To describe this significant phase in industry, the German government has introduced the term Industry 4.0 that represents a transformation of industrial production in the fourth industrial revolution. The term Industry 4.0 refers to the transition from distinct embedded systems to cyber-physical systems. The complexity of the involved technologies in each stage of industrial development is increasing. The Internet of Things, Data and Services are components of the Cyber-physical system (Germany Trade & Invest 2014).

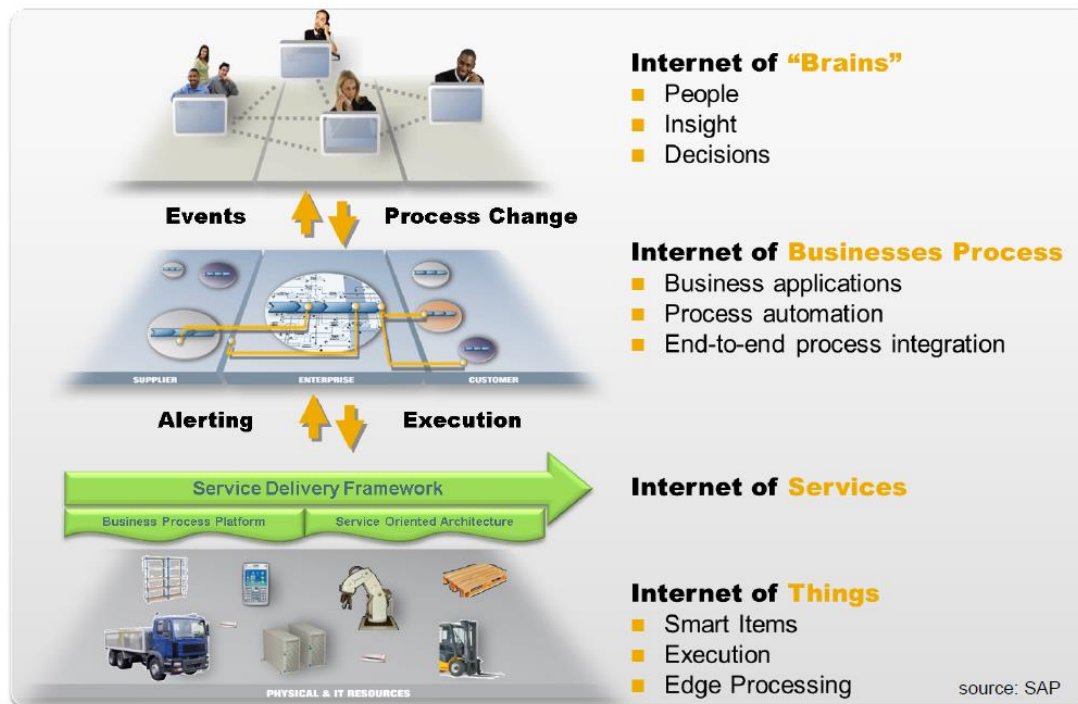


Figure 2: Cyber-Physical System concept (Karnouskos 2013).

Internet of Things is a concept of globally networked, accessible, observable and manageable intelligent physical objects. The intelligent object consists of an embedded system, sensors and actuators (Delicato 2013).

In the Industrial Internet of Things, industrial machines and devices, such as frequency converters represent IoT objects if they have embedded system with implemented Internet Protocol and communication abilities. Usually, the implementation of the Industrial IoT requires a renovation of the device's hardware and software perspectives. Today most industrial devices do not yet have abilities for direct communication over the Internet as independent objects. Therefore, to adapt incompatible devices to the new IoT concepts there are many technologies for this purpose, such as OPC.

Internet of Services is another important concept in the industry. It is a value-added service with defined specific enterprise tasks for the Internet of Things offered on the global network. It may be IoT related data acquisition, analytics or management services, and production processes like customized production or logistics services. From IT perspective, it looks like service-oriented applications for IoT (Fischer 2014).

Implementation of the enterprise grade cyber-physical system requires a large and scalable IT infrastructure because Cloud computing can be used instead of building specialized industrial IT system locally or in a so-called on-premise way.

Cloud Computing is a hosted web service offered over the Internet. Enterprise customers can extend and scale up their existing IT system by remote computing service use on-demand (Mahmood 2011).

One of the Cloud computing service delivery models is a Platform as a Service. In this model, a provider of the Cloud service offers IT infrastructure with libraries, services, and tools for customer-created applications. The service provider maintains an entire IT system, applies all related updates and guarantees the permanent availability of the service. Clients of the Cloud service only manage their platform based applications (Kavis 2014).

Sensors and embedded systems in industrial devices are becoming more intelligent and are able to send structured data to a control system, e.g. comprehensive diagnostics, operation states and real-time values etc., therefore relatively to industry 4.0, connecting the intelligent devices into a complex cyber-physical system increases the amount of available data in the system. Consequently, that brings about a data management challenge for approaches to make the data available for business analytics on-demand. The challenge is called “Big Data” (Stackowiak 2015).

The SAP Company has introduced a technical solution named SAP HANA that was created specially to solve “Big Data” challenge. HANA is a cloud computing platform for real-time analytics. A core component of the platform is an in-memory database that is designed for storing data to achieve high data mining performance, for example, applications for business operation analytics. The platform consists of many modules e.g. databases, connectors, services, and application development environments (Mankala 2013).

1.2 Goals

The goal of this thesis was to develop an industrial Cloud service. The main feature of the service is real-time data acquisition from Siemens automation devices for remote monitoring and data analytics.

The general output is software as a service for data acquisition from OPC UA to Cloud database. The practical example case was to store frequency converter's parameters to database and graphical presentation of the data on remote user interface. The goal was also to create material for end user presentations containing a case example and benefits of the Cloud service in condition monitoring and preventative maintenance.

Before starting the practical part of this work, the most important features of the HANA platform such as pre-built communication abilities and HANA database management system needed to be described.

This thesis was assigned by Siemens Osakeyhtiö for their industrial service development purpose as part of an industry digitalization strategy. The solution presented in this work is not a part of future official Siemens Cloud for Industry solution. The content of this work is a customized demonstrative solution useful for understanding benefits and possibilities of the industrial data acquisition Cloud services based on SAP HANA platform.

1.3 About Siemens Osakeyhtiö

Siemens AG is a global company focusing on the areas of electrification, automation, and digitalization and it is one of the world's largest producers of energy-efficient, resource-saving technologies. In infrastructure and industry solutions, the company plays a pioneering role. As of September 30, 2015, Siemens has around 348,000 employees in more than 200 countries. Siemens had €75.6 billion revenue in 2015. Siemens Osakeyhtiö is the Finnish subsidiary of Siemens located in the Espoo region, and it has about 500 employees.

2 Background of the HANA platform

2.1 Overview of the HANA platform as a service

The HANA platform is built for enterprise-grade applications with pre-built tools and technologies for data real-time analytics. Real-time means operational speed of milliseconds to seconds. To make a custom application on the platform, HANA includes software development components and programming libraries.

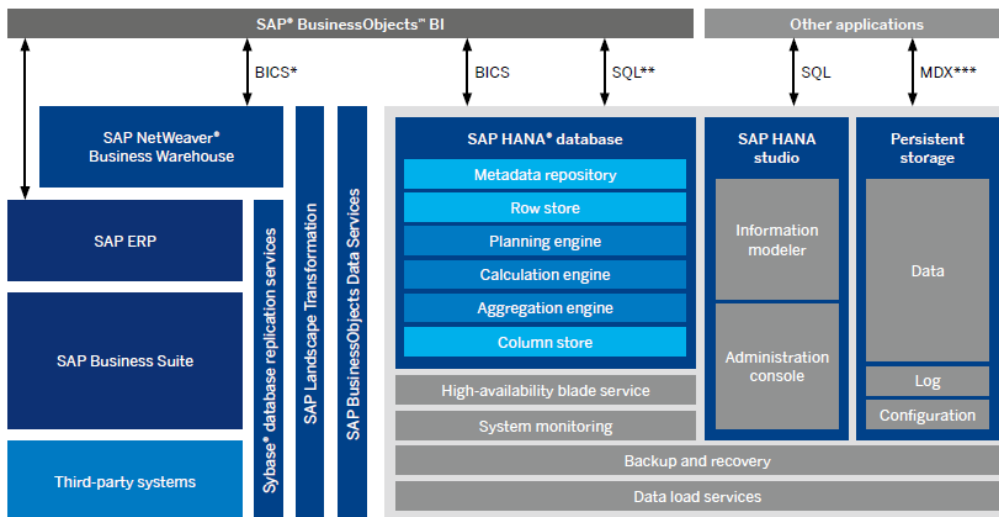


Figure 3: SAP HANA cloud platform overview (Schneider 2013).

In-memory database management system is a core component of the platform (Figure 4).

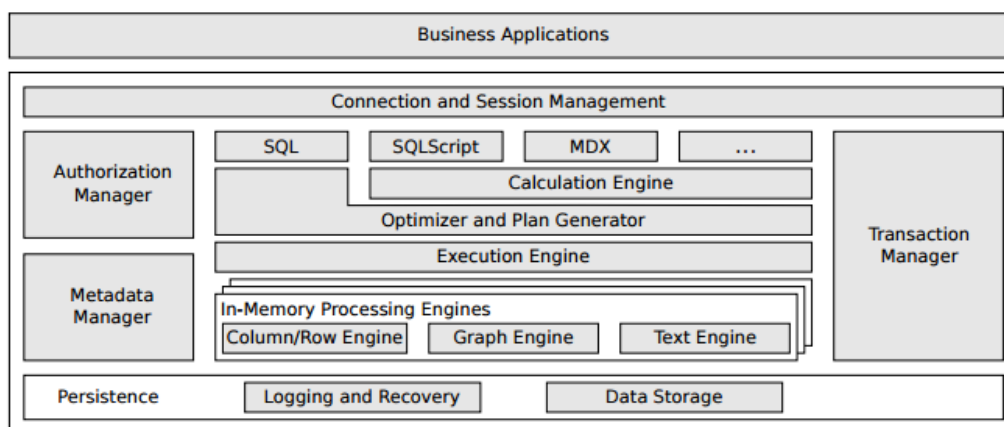


Figure 4: The architecture of HANA DBMS (Fäber 2012).

The Database Management System has all the necessary features for independent data processing. The difference between HANA in-memory DBMS and traditional database management systems is that the calculation engine is located in the database layer instead of the application layer (Fäber 2012).

The most important part of DBMS is an in-memory processing engine that is described in the next section.

2.2 In-memory processing engine

In contrast to traditional database management systems, the primary physical storage type of in-memory database is the computer's main memory. The main memory is the medium between persistence and operation systems such as SSD and CPU. The main memory is made of DRAM volatile memory type providing a constant access time to any memory location. Nowadays, the main memory capacity in enterprise-grade Cloud systems can reach a capacity of many terabytes. An in-memory database is combined with a traditional database system for data logging and recovery as well as a storage for outdated data (Plattner 2011).

2.2.1 Data storage formats

The main memory is one dimension vector of addressed blocks. In column oriented storage, each table column is stored in memory blocks one by one, then column operation accesses data in memory linearly without iterative data searching. Therefore, SQL functions for data aggregation such as SUM, MAX, MIN, AVG and other relevant functions are much faster in comparison to a traditional row-oriented storage where entire rows are stored in memory blocks. Column-oriented storage, in contrast to traditional row-oriented storage, is well suited for complete column reading. This can be useful for aggregation and column scans.

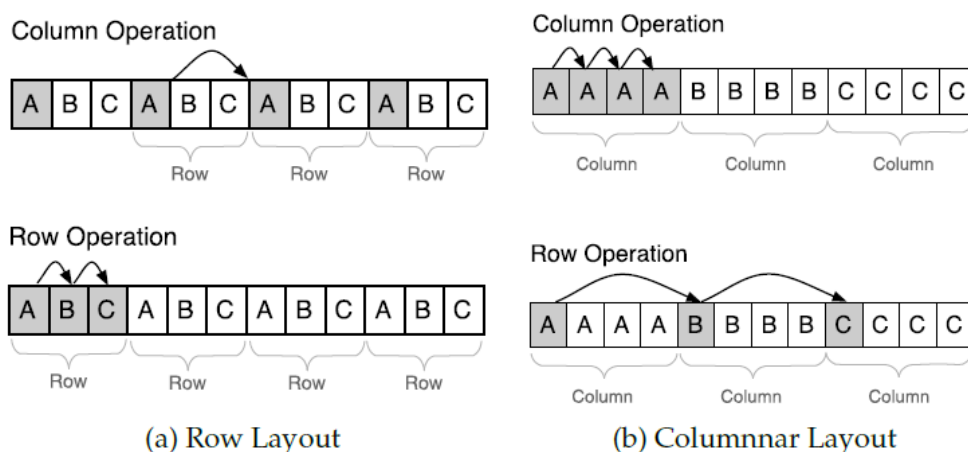


Figure 5: Columnar and row storage layouts (Plattner 2015).

To reduce the amount of data that needs to be transferred between storage and processor, DBMS supports data compression technique as a Dictionary encoding by default (Plattner 2015).

2.2.2 Compression

The storage capacity of the main memory is not as large as in a traditional HDD system, so it is important to decrease the space required for data with compression techniques. The memory consumption and transfer time can be reduced by decreasing the amount of bits used for storing data. The in-memory database uses dictionary encoding by default. The main function of dictionary encoding is that long values are represented in the table as short integer values.

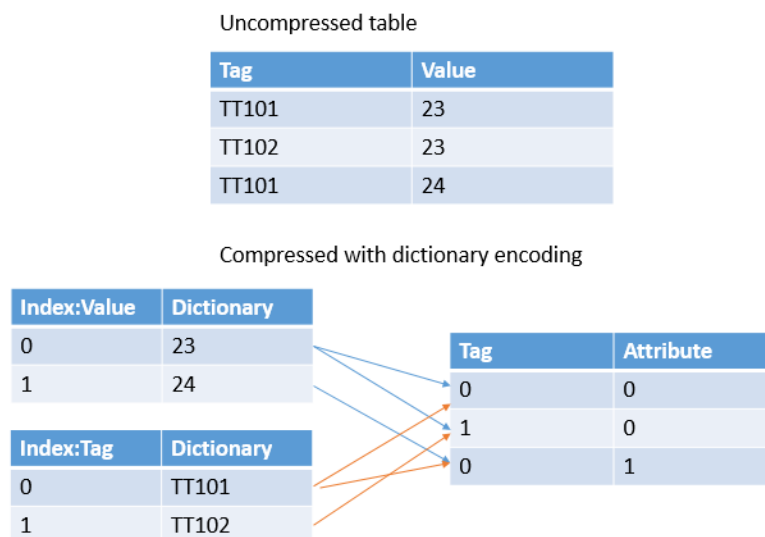


Figure 6: Database compression example

As shown in Figure 6, each column in dictionary encoding compression consists of dictionary and attribute vectors. Each element in the attribute vector refers to the index of the original value in the dictionary vector. The dictionary vector has only distinct values of each table column (Plattner 2014).

2.2.3 Data partitioning

Current multi-core CPUs are capable of handling several parts of data at same time. Parallel data processing in multicore CPU hardware systems requires a logical database distribution into independent datasets. Parallel computing enables performance improvement in enterprise data processing. Data partitioning is the process of data tables divided into small datasets. The main reason is to meet data handling parallelism on multi-core CPU hardware. Therefore, partitioning is a technical step to increase the query speed. Data-level parallelism impacts performance gains, however, at the same time it improves availability and manageability of datasets (Plattner 2014).

2.2.4 Delta buffer

Data storage in a columnar way in the main memory has an impact on data input performance. In write operations, data that goes to the database is consistent row data, therefore, to insert each value to its place in memory takes longer time than in the row storage type. To solve that performance problem, the concept of a differential buffer was introduced, where all new entries are written to a differential buffer storage first and then, at a specific point in time, data in buffer will be merged into the main data storage (Plattner 2014).

Write workflow always goes through the differential buffer (Figure 7).

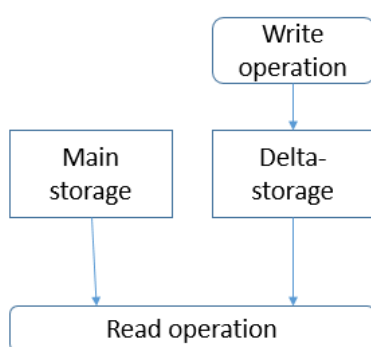


Figure 7 Write/Read operations on HANA database

2.3 Service security overview

Data protection is important in manufacturing, as well as in any other business. In this section data processing security of the industrial cloud service will be described. Data acquisition to a remote database includes data transfer over internet environment and storage on remote service. Especially, data handling in a secure way is an important issue in the connection of industrial systems and devices to internet environment.

2.3.1 Data layer security

HANA database system has multi-tenancy feature, which enables to store data from many clients of service without physical separation of databases. Each database user has their own identifier and assigned roles. Client's databases are isolated through operating system mechanisms and cross-database communication is not enabled by default. Additionally, data volume encryption can be enabled in the persistence layer for higher data protection. The service also enables logging for all performed actions.

In the cloud service data is stored in a volatile memory, meaning that data will be erased permanently on power failure events. Database management system has the ability to protect data on failures by systematic backups and recovery. On an event that affects service availability such as server system downtime, critical data can be automatically replicated by data replication services (SAP help 2015).

2.3.2 Communication layer security

Cloud service supports few authentication mechanisms as Kerberos-based or SAML-based authentication. All network connections to databases or client systems can be encrypted with Secure Sockets Layer (SSL) and Transport Layer Security (TLS) techniques. HANA Cloud connector has features that can be used to ensure end-to-end delivery of every message (SAP help 2015).

Additionally, in the case of remote data acquisition there is a need to make sure that no data is lost because of communication failures.

3 Background of the data acquisition service

3.1 HANA Connector

The HANA connector is a client-side software which works as a communication interface for acquisition service on a client system. The connector maintains security and consistency of data between the service and Remote HANA Database (HANA help 2016).

Figure 8 describes HANA connector communication via HTTP protocol.

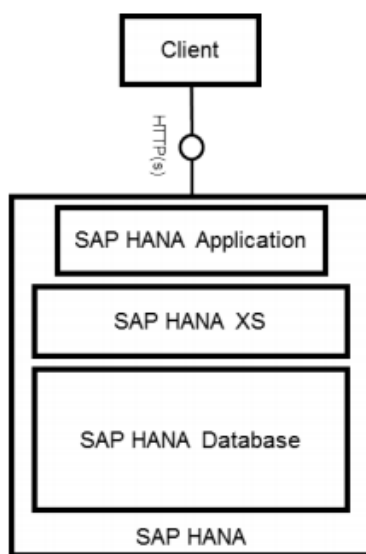


Figure 8: HANA connector communication via HTTP(s) protocol

Systems and data sources can be adapted via SDK software and open source tools. These tools can be used for the adaptation of standard industrial communication protocols, e.g. HART, PROFIBus, and OPC. Figure 9 shows data acquisition adapters.

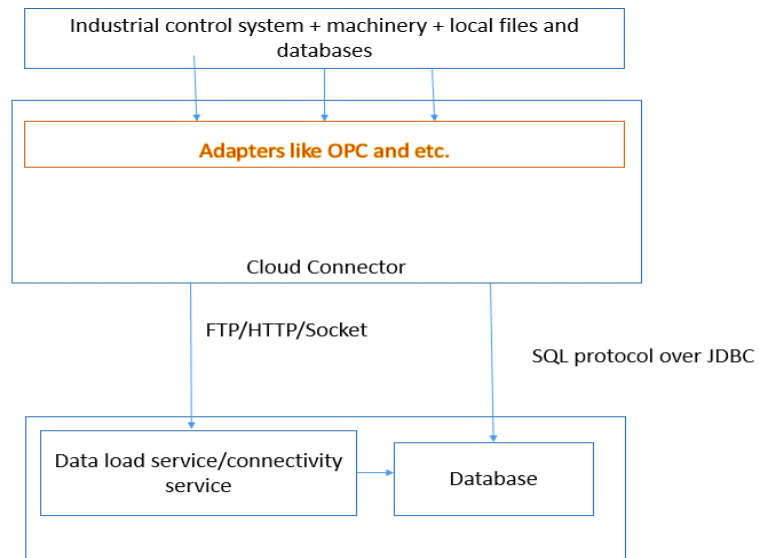


Figure 9: Cloud service connectors and adapters

Next table (Table 1) gives an overview of the supported adapters.

Adapters	Communication protocol	Data carriage	Comments
Web Services	REST	XML, HTTP, JSON	Web based comm. technologies. non-blocked by firewall
Database tunnel	JDBC, ODBC	SQL	Direct SQL insert
Files	FTP	CSV	Periodical data upload from CSV file

Table 1: List of available data adapters.

3.2 OPC UA

“The OPC Unified Architecture (UA) is a platform independent service-oriented architecture that integrates all the functionality of the individual OPC Classic specifications into one extensible framework” (OPC Foundation 2016).

OPC UA is standardized by OPC Foundation and thus is supported by many industrial automation equipment providers and manufacturers. OPC UA (Unified Architecture) has a client-server communication concept. Siemens Simatic.NET industrial communication software has integrated OPC servers. Siemens s7-1500 PLC tags can be observed on OPC UA client system via Simatic.NET OPC UA server.

3.3 Other tools and technologies

3.3.1 Node.js

“Node.js is asynchronous event-driven JavaScript framework with cross-platform runtime to build network applications”. It has a package ecosystem that offers many community supported open source libraries (Nodejs 2016).

3.3.2 MongoDB

MongoDB is NoSQL cross-platform document database that stores data as complex structured documents in JSON format (MongoDB Inc. 2016).

3.3.3 WebSocket

For real-time communication WebSocket protocol can be used that provides bi-directional communication over a single TCP connection between web clients and servers. The protocol was standardized by the IETF as RFC 6455. For the WebSocket implementation in Node.js environment JavaScript socket.io library was used (IETF RFC6455 2011).

3.3.4 Angular.js

Angular is a JavaScript framework for building web application user interfaces with Model-View-Controller or Model-View-ViewModel architectural pattern (Angularjs 2016).

4 Practical implementation

4.1 Assignments

The assignment was to develop a software that can transfer data from local industrial systems and devices to remote HANA database wherefrom data can be used for remote system monitoring and data analytics. Additionally, to test the developed solution the requirement was to make a lightweight demonstrative remote monitoring user interface. As a useful demo case, a frequency converter was chosen with its power unit's cooling fan working hours as an example of remote monitoring.

4.2 Requirements

Acquisition service functionality should include:

1. Transfer tag value from PLC S7 – 1500.
2. Acquisition interval at least 1s.
3. Data should not be lost on online connection breaks.
4. Data should be automatically replicated when connection is established again.
5. At least INT and REAL data types should be supported.
6. Tag's value needs to be stored with its source information as name and source timestamp.

The methods and tools to meet the software requirements can be chosen freely.

Demo application functionality requirements were the following:

1. Show some simple time axis trend line for stored "fan operating hours" parameter's value.
2. HANA web service user interface.

For demonstrative case SIMATIC S120 frequency converter with asynchronous vector motor was selected.

4.3 Data acquisition service development

As a solution for real-time data acquisition modern web development tools were selected. The application is programmed in JavaScript and runs on Nodejs runtime as a service and utilizes open-source programming libraries and tools. It was decided to implement software as a service with independent modules. The service is accessible via single API. As a common

communication interface to s7-1500 PLC Simatic.NET OPC UA server was chosen. As a communication protocol between Connector and Cloud service SQL protocol was chosen to provide direct data insertions onto database tables. The service software was developed with asynchronous functionality and real-time communication features. Interface for this service is a REST web application. In Appendix 1 all used libraries for the UA CONNECTOR service are listed. Appendix 2 lists all used libraries for UA CONNECTOR user interface.

4.4 Results

4.4.1 Overview

Name: UA CONNECTOR (Generic name). **Version:** 0.1.0. **Versioning model:** Major.Minor.Patch

Service runtime prerequisites:

1. Nodejs v. 4.x.x LTS.
2. MongoDB v. 3.x.x or greater¹.
3. DB tunnel: SAP Neo-java-web SDK 1.92 or higher. Alternative: SAP Cloud Connector.

Features:

1. REST JSON communication with real-time web socket events.
2. Asynchronous code design.
3. Windows, Linux and Mac environments supported: See Nodejs official documentation at <https://docs.appdynamics.com/display/PRO40/Node.js+Supported+Environments>.
4. Modular and functional JavaScript code style.

Functions:

1. OPC UA Web-based client.
2. Simatic.NET adapted s7-1500 tag reading.
3. WinCC RT Advanced adapted tag reading.
4. 3 tag read modes: 1 second, 10 seconds and subscription.
5. 30 s DB write interval.
6. Two database types local: MongoDB (NoSQL), Remote: HANADB.
7. Real-time connection monitoring: online/offline and automatic storage type switch.
8. Data provisioning to Analytics view from HANADB **when** HDB client is connected.
9. Three tag types REAL, INT, DINT.
10. PLC tag types automatically mapped to HANA Table data types.
11. Multi-PLC discovery inside one OPC UA server session.
12. Single system client.

¹ x = any number.

13. Tag value is stored with its source information as ID, NAME, and TIMESTAMP.
14. TIMESTAMP is date and time of the source stored in ISO 8601 format.

Security considerations:

The connection to HANA database is secured by using DB tunnel with JDBC connection via neo-java-sdk. The tunnel opens to HANA account with the following console command:

“Neo open-db-tunnel -a ACCOUNT -h hanatrial.ondemand.com -u USER -i DB_SCHEMA”

This command provides a password, which is required by service so that it can get access to user’s HANA database. Additionally, SAML and encrypted with X.509 certificate communication are available. For production use native HANA Cloud Connector or similar instead of the neo-java-sdk for HANA database connectivity is recommended. Neo-java-sdk is used for developing and testing purposes only.

UA CONNECTOR service is tested with HANA Trial instance of HANA Cloud platform. Figure 10 shows database tunnel opening in Neo-java-sdk tool.

```
SAP HANA Cloud Platform Console Client
Password for your user:
Opening tunnel...
Tunnel opened.
Use these properties to connect to your schema:
Host name      : localhost
Database type  : HANAXS
JDBC Url       : jdbc:sap://localhost:30015/?currentschema=XXXXXXXXXXXXXXXXXXXX
Instance number : 00
User           : DEMO_USER
Password       : XXXXXXXXXXXXXXXXXXXX
Schema name    : XXXXXXXXXXXXXXXXXXXX
This tunnel will close automatically in 24 hours on Feb 23, 2016 at 9:21 PM or when you close the shell.
Press ENTER to close the tunnel now.
```

Figure 10: Opened DB tunnel to HANA trial instance in neo-java-sdk console.

The service is tested with OPC UA unencrypted communication. Optional Basic128Rsa15 encryption is supported by the service, however, not yet enabled in UI. The service uses local port 8000 by default. Access from a local network to service is not possible while service’s default port is not be opened in operation system’s firewall. In this software version, the service’s user interface is not protected with any access credentials.

Data protection functionality

On any unexpected Internet communication breaks, UA CONNECTOR goes to offline state and stores all incoming information to local MongoDB database. Connection state is monitoring continuously inside data storage cycle. If at some point in time the remote database connection will be restored, all locally stored data replicates to HANA database. The data in local storage will be deleted only on succeeded replication (Figure 11).

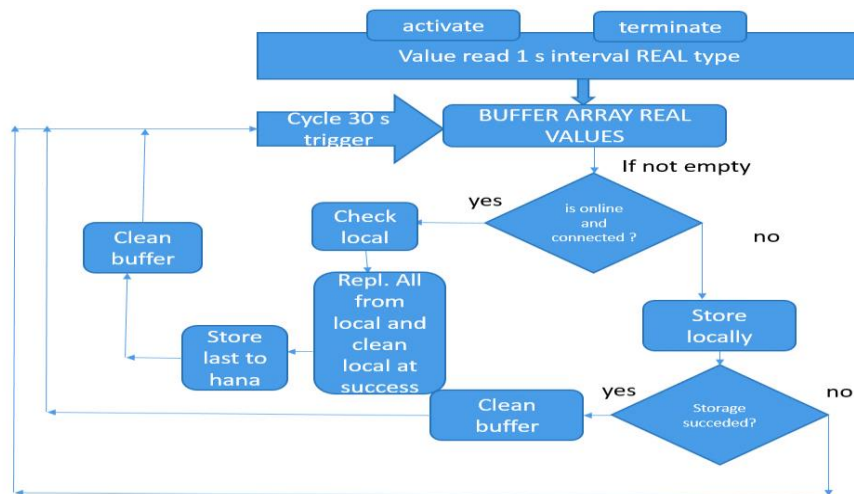


Figure 11: Example: Data transfer schema

4.4.2 Architecture

The service is built with centralized API that processes requests from user Interfaces and sends back responses asynchronously. Asynchronous functionality means that if one of request processes is waiting for some function executions it does not lock other functions and requests. Figure 12 shows an architectural approach for UA CONNECTOR.

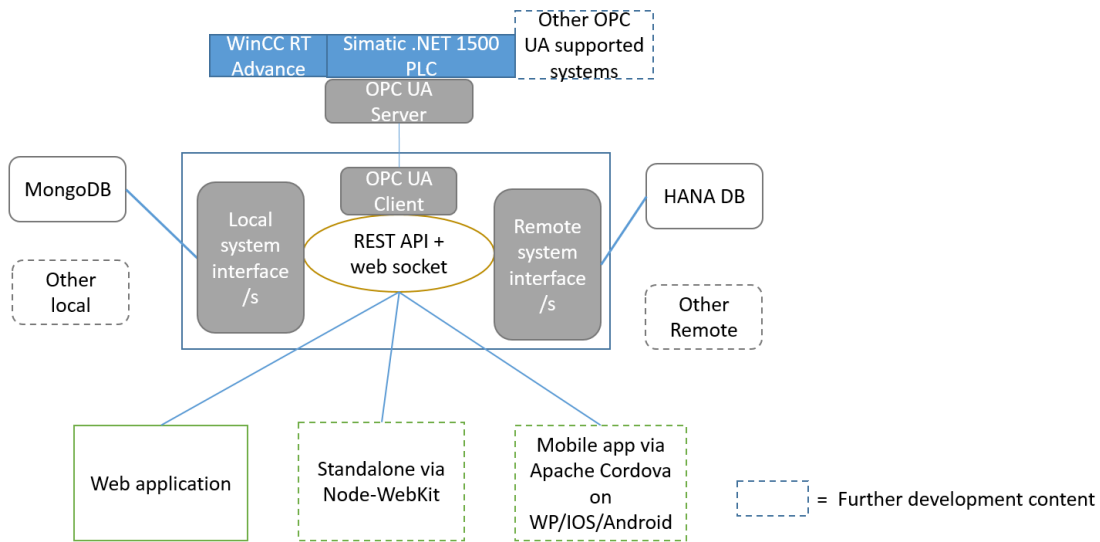


Figure 12: UA CONNECTOR 0.1.0 Architecture

4.4.3 Connector’s interface

The web interface is developed as a web application that uses REST JSON communication type and partly WebSocket communication protocols to communicate with UA CONNECTOR service. The interface is adaptive and developed for desktop and mobile device. The functionality is tested with Internet Explorer 11 and latest Google Chrome browsers. Figure 13 shows the user interface of the UA CONNECTOR service.

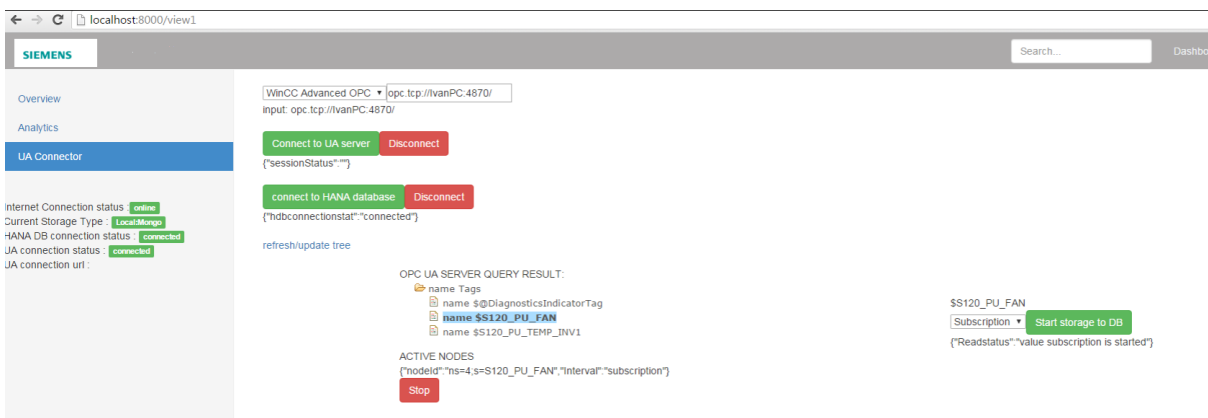


Figure 13: UA CONNECTOR User Interface

Via web interface, the user can:

1. Connect and disconnect OPC UA server and HANA database system.
2. See important connection states : Internet connection status, current storage type, HANA DB connection status, UA connection status, UA connection/session URL
3. See expandable tree view with all WinCC RT Advanced/s7-1500 PLC tags
4. See active storage processes/nodes
5. Start and stop storage process of the Tag value.

4.4.4 Source code (Confidential)

The source code of this software solution is not part of this publication because of confidentiality agreements with Siemens Osakeyhtiö.

4.4.5 Cloud system structure

HANA system has an admin user, who has privileges to add new service's clients and assign to each client a database schema with a specific generic ID. In each schema, client's admin can create objects e.g. tables, calculation views, applications, however, a use of these objects by endpoint application user is not possible without application specific privileges. Even the admin users of the schema should add themselves as users of this data, for example, grant SELECT rights of the specific database table or view (Figure 14).

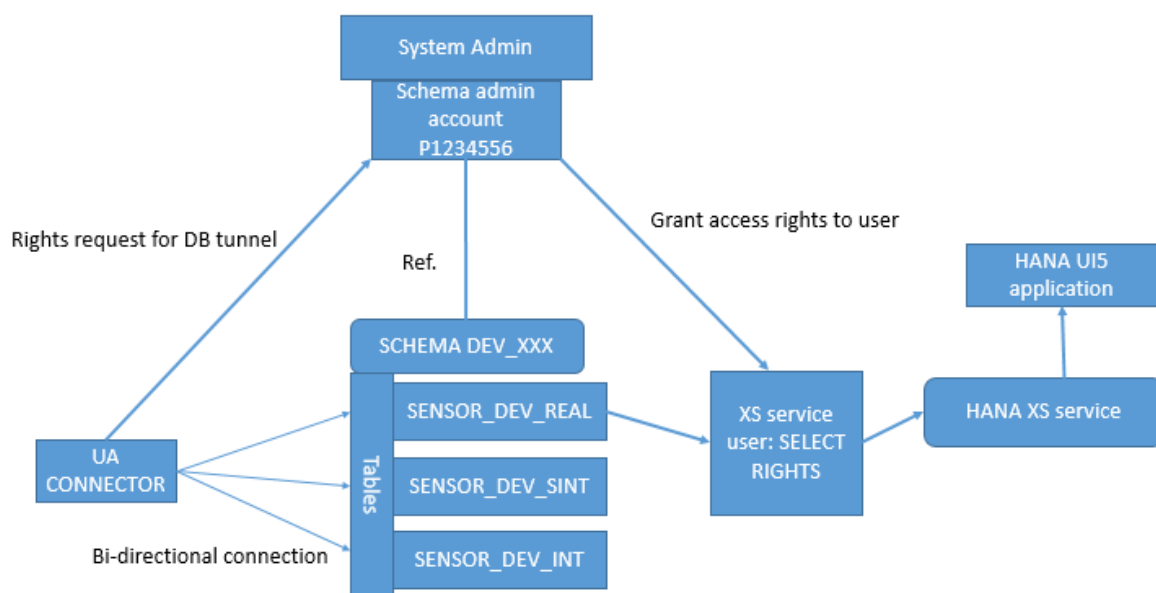


Figure 14: Cloud system schema - interaction between services

5 Demo Case

5.1 Purpose

After discussion with support specialists about useful maintenance cases of SINAMICS s120 frequency converter, it was decided to check power unit module's fan operating hours via developed UA CONNECTOR acquisition service and HANA SAPUI5 application. A reason for remote monitoring is that if the Power Unit fan's operating hours reach value 4000, the converter's operation should be stopped until the fan is replaced with a new one. Because the fan operation depends on the converter's operation periods, it is difficult to estimate when a fan should be replaced. A cooling fan for s120 converter is quite an expensive part, the price of which depends on its type, therefore, it should be changed in due time. Remote health monitoring can be offered as an after-sales service for the frequency converter. For example, when fan operation hour parameter's value reaches 3900 hours the service would inform the customer that in the next planned maintenance session it is recommended to change the cooling fan in this converter.

5.2 Tools

5.2.1 TIA portal

Siemens TIA portal (Totally Integrated Automation) is an automation engineering framework for Siemens automation products: programmable logic controllers, distributed I/O, HMI, drives, and motion control (Siemens. TIA 2016).

5.2.2 Simatic WinCC

Simatic WinCC is a development software and runtime for process visualization system. WinCC is used to monitor and control physical processes in industry (Siemens 2016a).

5.2.3 STARTER

STARTER is a commissioning tool for Siemens's frequency converters. The tool is used for parameterizing, commissioning, and troubleshooting converters (Siemens 2016b).

5.2.4 HANA XS and SAPUI5

The HANA platform has an integrated software as a service development framework to make enterprise web applications. HANA Extended Application Services (XS) is an application server in HANA platform for developing JavaScript services with integrated data processing technologies (SAP SCN 2014). SAPUI5 is a front-end framework similar to Angularjs (Figure 15).

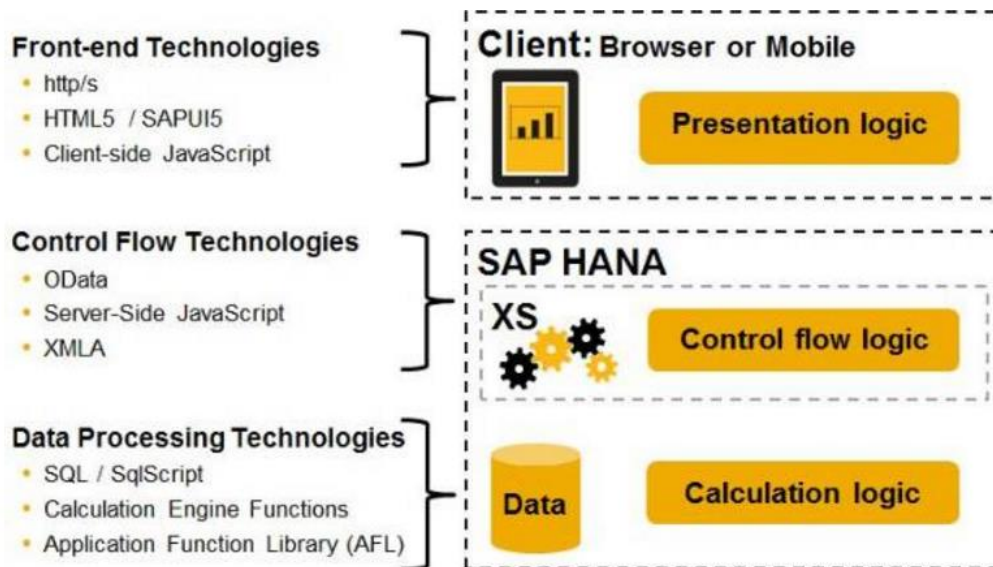


Figure 15: HANA XS and SAPUI5 (SAP SCN 2014)

5.3 Results

5.3.1 Overview

For this demo case, s120 demo installation with complete electrical interconnections was used including motor and encoder. The first task was to configure parameters for the motor and the frequency converter. For this purpose, Siemens STARTER software was used. Network communication to WinCC RT Advanced was configured with TIA portal v13. WinCC RT Advanced is a PC-based HMI system. In this project, WinCC Advanced v 13 was used that has OPC UA built-in server for WinCC tags. UA CONNECTOR uses this OPC UA server to get access to tags defined in WinCC tags table.

The demo components are listed in the table (Table 2) below.

Name	Type
Control Unit	SINAMICS S120 CU310-2 PN
Power Module	v 340
Runtime (Middleware)	WinCC RT Advanced v13 with OPC UA
Motor	230V 0.73A 0,12KW 50HZ 1350 1/min
Encoder	HTL 1024
Communication type	PROFINET S7ONLINE s7 300/400 Acyclic
Data acquisition service	UA CONNECTOR
Database	SAP HANA DB
Interface	HANA XSAPP and UI5

Table 2: Demo components

5.3.2 Preparation

For this task it was needed to read values from parameter “p251[0] Operating hours power unit fan”. Any s120/g120 parameter’s address for WinCC tag can be calculated with the following formula:

DB{parameter’s number}.DBD|W|B[1024*Drive Object number+ array position number].

Data block type depends on parameter’s data type. Drive Object number for s120 converters with one motor is usually 2.

The parameter p251[0] tag’s address is DB251.DBD2024. Because the change frequency of the parameter’s value is so low, it is reasonable to store the value with subscription interval. The subscription interval means that the value will be transferred to database on a change in value.

The frequency converter on the demo environment was driven via STARTER control panel available in its commissioning tools.

5.3.3 Interfaces

For the purpose of testing of the bi-directional connection with HANA database, an embedded graphical data representation was made (Figure 16). For remote monitoring service, another interface using HANAXS services and SAPUI5 was created (Figure 17).

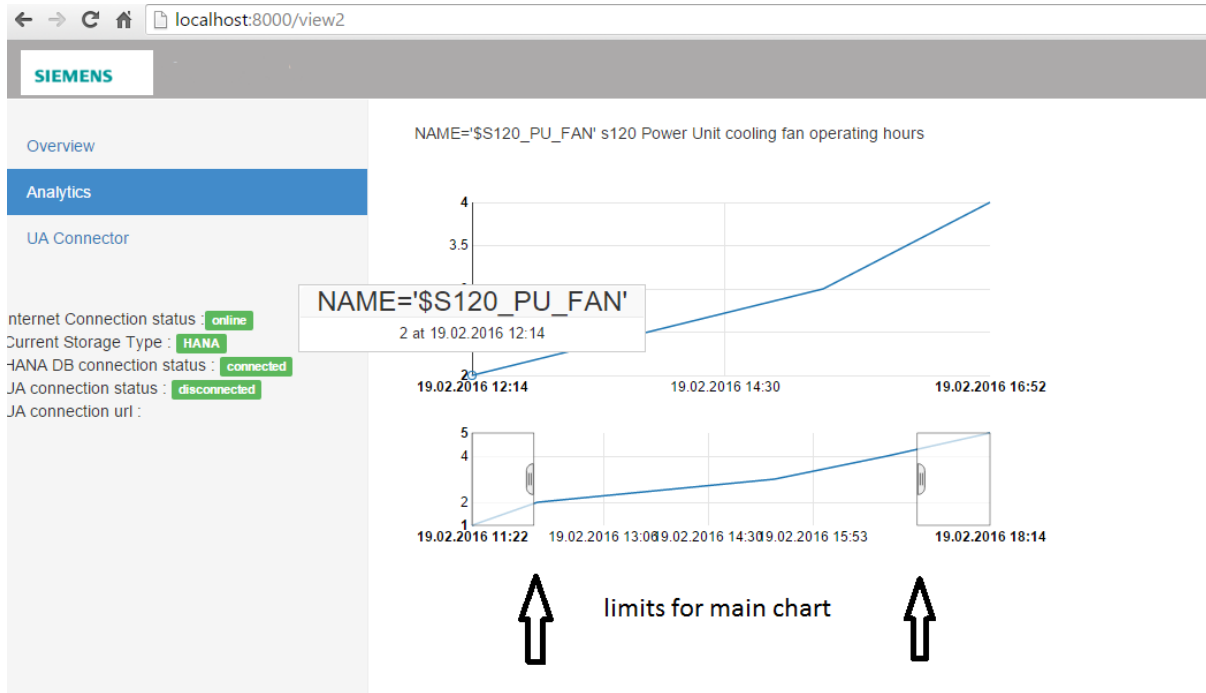


Figure 16: Trend line for \$\$S120_PU_FAN tag on embedded view in UA CONNECTOR (local)

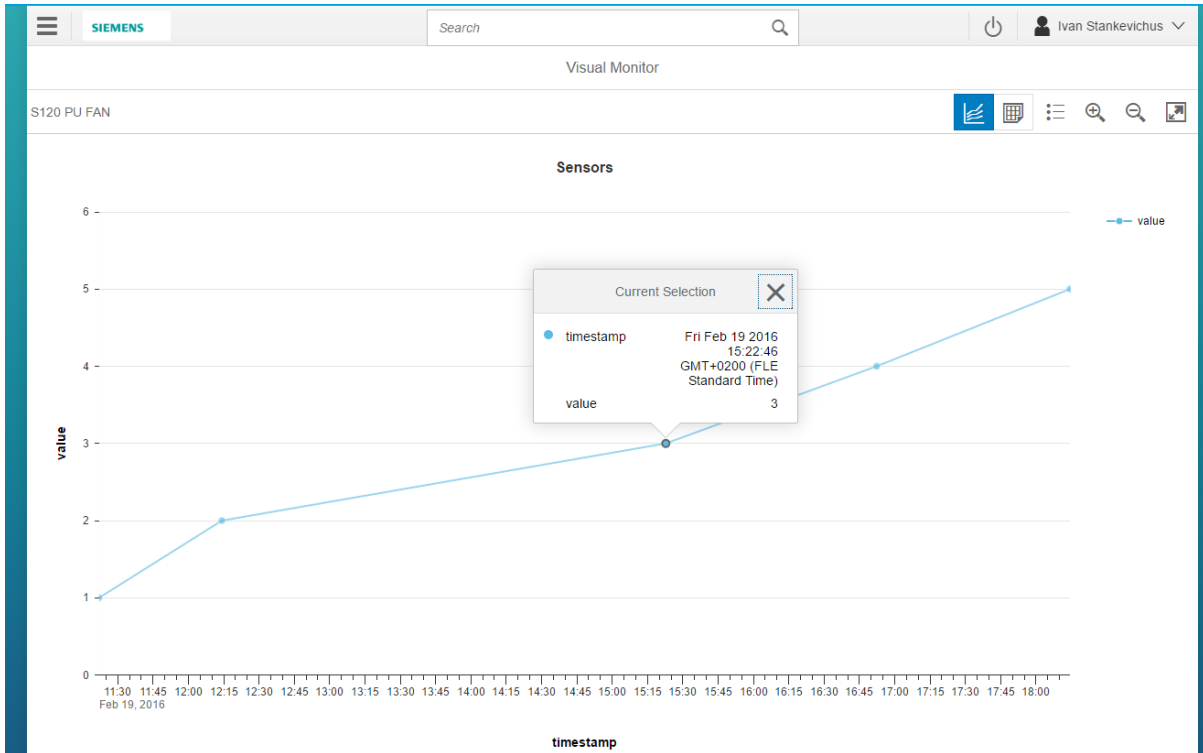


Figure 17: Trend line for S120_PU_FAN tag on HANA Cloud platform.

Figure 18 shows the user Interface for remote monitoring service built with SAPUI5 technology on HANA Cloud platform.

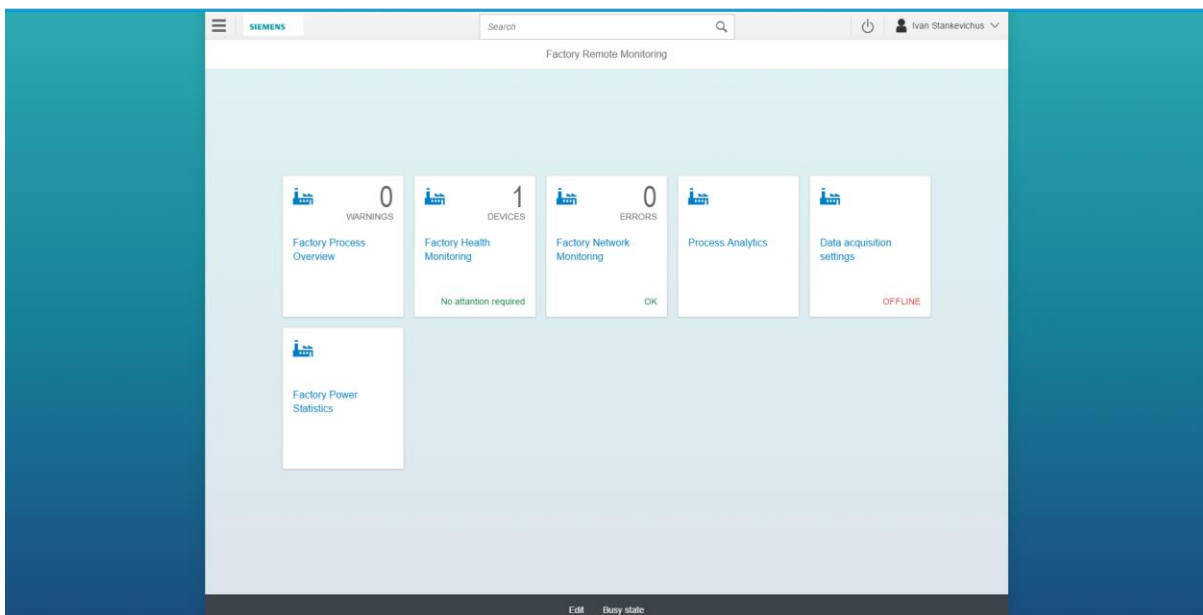


Figure 18: Factory Remote Monitoring Home Page

For demonstrative and test purposes, only a few values from the frequency converter were stored. Since the value changes are slow, there was no reason to load the motor in the test environment longer.

6 Conclusions

6.1 Cloud service benefits

As a conclusion of this work overall possible benefits of using data acquisition as a Cloud service can be examined. First of all, it can be mentioned that cloud services are extending the customer's local IT infrastructure without extra hardware, software and maintenance costs. This reduces Total Cost of Ownership for implementing real-time data analytics and massive data storage. Cloud service based on HANA would be useful platform for manufacturing execution system and integrated with the ERP system. Vertical system integration provides possibilities for tracking production process through the entire business, and thus production can be optimized and its cost reduced.

Industrial machinery such as complete generators, pumps or motors are valuable sources of information. Information about machine environment and operation values can be monitored in real-time, which enables a new service business model. A concept of the service model is that over cloud service, machine operators or support engineers are able to remotely monitor a device's health and predict future maintenance milestones and carry out preventive maintenance. All that enables new service agreements such as process optimization analytics, maintenance service with real-time tracking and data aggregations, or the industrial machine itself can be offered by its manufacturer as a service for manufacturing business.

For example, the pump manufacturer ABS offers pump working power as 1h/1€ for factory FZR, the data about the pump is available over the internet via third party SMN cloud service. In this service, a pump maintenance service specialist from ABS monitors the pump's health over an application that they have built on top of a Cloud service platform. The service predicts the next maintenance milestones and required pump parts etc. The pump is still owned by ABS and the data is stored on SMN database. As a result, FZR has additional

quality service and reduced TCO, and ABS does not need costly IT infrastructure for data storage and analytics, and SMN as a Cloud service provider get more supplementary service contracts, which increases the revenue of the company. Figure 19 shows a potential use case for the service.

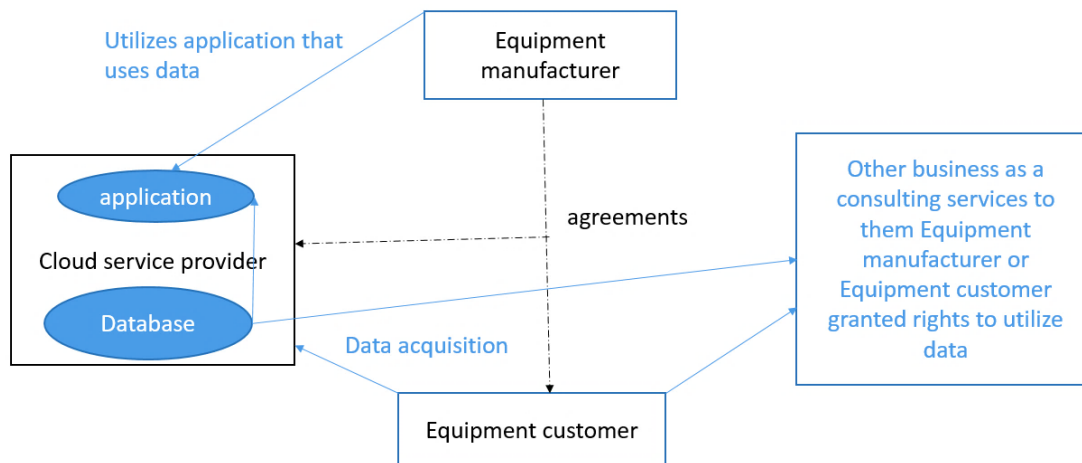


Figure 19 : Service usage example

Furthermore, HANA Cloud service's real-time analytics feature with build-in predictive analytics libraries enables for the industry predictive maintenance that is the most demanded feature for Cloud Service applications. It mostly applies to machinery related maintenance. Regular maintenance prevents failures passively but active tracking can prevent even unexpected failures. Also, gathered data can be used for increasing production yield through clarifying relationships between important parameters and production output.

Quality issues in production can be identified in detail that makes it possible to identify a time when defects began and may help with product mass rollback to clarify how much and what products must be returned.

Additionally, data can be stored with source geographical information and used for production related logistics optimization (Plattner 2015).

6.2 Further Development

Following features should be added before using UA CONNECTOR data acquisition service in a customer's system:

1. Software automated unit and behavior tests with test coverage at least >90%.
2. SAML-based HANA DB user authentication.
3. Tag properties read on-demand.
4. User login system and settings protection.
5. Multi-session OPC connection.
6. Event logging features.
7. Online help documentation.
8. Make OPC UA client component of the service more unified with support all OPC UA compatible products.
9. Remote UA CONNECTOR service health monitoring.
10. Make Alarm & Events system.

6.3 Results

Connector software fulfills all assigned requirements and can be used as a base platform for further development and extensions. The Demo is a simple showcase of the usage of Cloud services for remote monitoring and can be used in marketing for end customers.

7 Discussion

The total time that I spent with this project is about 480 hours including learning, development work and report writing. In total about 3,000 lines of code were written including the demo case and the remote monitoring interface. In this project, I have used helpful JavaScript libraries and tools which helped me to save developing time. For example, using Jade templating language instead of pure HTML reduced the code length to half for the user interface development. Technologies and tools used in UA CONNECTOR service were quite new for me and I spent plenty of time to learn them.

Because of using free of charge HANA Trial account, it was not possible to make interesting additional functionalities. HANA Trial account has many development restrictions and limitations. For example, in the trial version, it was not possible to make OEE real-time calculation. I am waiting for Siemens's internal HANA platform with all enabled features. Then it is possible to make a product from the data acquisition service. Anyway, in this work

it was not required to make the service ready for the customer, however, I believe it is a suitable platform for further development. I expect that it takes a few months to prepare this solution for production use.

This work took about three weeks longer than was expected since the demonstrative part of the project took longer than planned. I noticed that parameter p251[0] cannot be attached in cyclic data exchange with s7-1500 plc, therefore read operation should be performed only over a special functional block or directly via WinCC. Direct communication to WinCC was not implemented in UA CONNECTOR, thus I should spend more time to add this functionality to my software. A reason to use WinCC for direct parameter reading is that PLC in a real factory environment may be protected and any changes should be supplied with acceptances, tests etc. If we look at the trend line in Gartner's 2015 Hype Cycle for Emerging Technologies (Figure 20), advanced analytics with self-services and Internet of things they are highly growing technologies in next 2 to 5 years and the demand for solutions presented in this thesis work is rising.



Figure 20: Gartner's 2015 Hype Cycle for Emerging Technologies (Gartner. 2015)

References

- Angularjs. Conceptual Overview. Accessed on 12 February 2016. Retrieved from <https://docs.angularjs.org/guide/concepts>
- Delicato F. 2013. Middleware Solutions for the Internet of Things. London: Springer
- Fischer S. 2014. Towards the Internet of Services: The THESEUS Research Program: Challenges of the Internet of Services. Switzerland: Springer
- Fäber F. 2012. The SAP HANA Database – An Architecture Overview. Accessed on 3 February 2016. Retrieved from <http://sites.computer.org/debull/A12mar/hana.pdf>
- Gartner. 2015. Gartner's 2015 Hype Cycle for Emerging Technologies. Accessed on 23 February 2016. Retrieved from <http://www.gartner.com/newsroom/id/3114217>
- Germany Trade & Invest. 2014. Industrie 4.0 smart manufacturing for the future. Germany Trade & Invest brochure. Accessed on 19 January 2016. Retrieved from http://www.gtai.de/GTAI/Content/EN/Invest/_SharedDocs/Downloads/GTAI/Brochures/Industries/industrie4.0-smart-manufacturing-for-the-future-en.pdf
- HANA help. 2016. SAP HANA Cloud Connector. Accessed on 19 February 2016. Retrieved from <https://help.hana.ondemand.com/help/frameset.htm?e6c7616abb5710148cfcf3e75d96d596.html>
- Karnouskos S. 2013. Cloud-based Cyber-Physical Systems in Industrial Automation presentation. Accessed on 15 February 2016. Retrieved from http://www.iecon2013.org/files/IECON2013_IF1_02_Karnouskos.pdf
- Kavis M. 2014. Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS). WILEY e-book.
- Mahmood Z. 2011. Cloud Computing for Enterprise Architectures. London: Springer
- Mankala C. 2013. SAP HANA Cookbook. Birmingham: Packt Publishing Ltd.
- MongoDB Inc. 2016. Introduction to MongoDB. Accessed on 23 February 2016. Retrieved from <https://docs.mongodb.org/manual/core/introduction>
- Node.js. 2016. About Node.js. Accessed on 20 February 2016. Retrieved from <https://nodejs.org/en/about/>
- OPC Foundation. 2016. Unified Architecture. Accessed on 23 February 2016. Retrieved from <https://opcfoundation.org/about/opc-technologies/opc-ua/>
- Plattner H. 2011. In-Memory Data Management: An Inflection Point for Enterprise Applications. Potsdam: Springer.
- Plattner H. 2015. The In-memory revolution: How SAP HANA enables business of the future. Switzerland: Springer
- Plattner H. 2014. A Course in In-Memory Data Management. The Inner Mechanics of In-Memory Databases. Potsdam: Hasso Plattner Institute course material.

- RFC6455. 2011. WebSocket. Accessed on 21 February 2016. Internet Engineering Task Force IETF. Retrieved from <https://tools.ietf.org/html/rfc6455>
- SAP. 2015. Production of the Future: How to Prepare for the Fourth Industrial Revolution. Accessed on 15 January 2016. Retrieved from http://global.sap.com/community/ebook/2013_11_28302/enUS/index.html
- SAP help. 2015. SAP HANA Security Guide. Accessed on 15 January 2016 Retrieved from http://help.sap.com/hana/SAP_HANA_Security_Guide_en.pdf
- SAP SCN. 2014. Introducing...SAP HANA Extended Application Services (XS). Accessed on 20 February 2016. Retrieved from <http://scn.sap.com/docs/DOC-60322>
- SAP SCN. 2015. What is SAP HANA? Accessed on 21 February 2016. Retrieved from <http://scn.sap.com/docs/DOC-60338>
- Siemens. 2016a. Simatic WinCC. Accessed on 22 February 2016. Retrieved from <http://w3.siemens.com/mcms/human-machine-interface/en/visualization-software/scada/simatic-wincc/pages/default.aspx>
- Siemens. 2016b. STARTER Commissioning Software. Accessed on 22 February 2016. Retrieved from <http://w3.siemens.com/mcms/mc-solutions/en/engineering-software/starter-commissioning-tool/pages/starter-commissioning-tool.aspx>
- Schneider E. 2013. SAP HANA Platform – Technical Overview. Accessed on 4 February 2016. Retrieved from http://www.ntua.gr/announcements/general/uploads/2013-07-01_171178_SAP_HANA_Platform_Technical_Overview.pdf
- Stackowiak R. 2015. Big Data and the Internet of Things: Enterprise Information Architecture for a New Age. California: Apress.

Appendices

Appendix 1. UA CONNECTOR service dependencies.

```
"dependencies": {
  "async": "^1.5.0",
  "body-parser": "^1.14.2",
  "error-handler": "^1.0.0",
  "express": "^4.13.3",
  "hdb": "^0.5.3",
  "is-reachable": "^1.2.0",
  "jade": "^1.11.0",
  "method-override": "^2.3.5",
  "morgan": "^1.6.1",
  "node-opcua": "0.0.51",
  "serve-favicon": "^2.3.0",
  "socket.io": "^1.4.0"
},
"devDependencies": {
  "grunt": "~0.4.5",
  "grunt-contrib-jshint": "^0.11.3",
  "grunt-contrib-uglify": "^0.11.0",
  "grunt-contrib-watch": "^0.6.1",
  "grunt-express-server": "^0.5.1"
}
```


Appendix 2. UA CONNECTOR web user interface dependencies.

```
"dependencies": {  
  "angular": "~1.4.8",  
  "angular-route": "~1.4.8",  
  "angular-tree-control": "~0.2.23",  
  "angular-socket-io": "~0.7.0",  
  "bootstrap": "~3.3.6",  
  "angularjs-nvd3-directives": "~0.0.7",  
  "angular-local-storage": "~0.2.3"  
}
```