Delivery no.: D2.6a
Specification of data collection system

Photo: By & Havn / Ole Malling

CEE-DTU
Author, Anders Laage Kragh
Date [14, 01, 2019]

Public deliverable ☒
Confidential deliverable ☐

Preface

EnergyLab Nordhavn – New Urban Energy Infrastructures is an exciting project which will continue until the year of 2019. The project will use Copenhagen's Nordhavn as a full-scale smart city energy lab, which main purpose is to do research and to develop and demonstrate future energy solutions of renewable energy.
The goal is to identify the most cost-effective smart energy system, which can contribute to the major climate challenges the world are facing.

Budget: The project has a total budget of DKK 143 m (€ 19 m), of this DKK84 m (€ 11 m) funded in two rounds by the Danish Energy Technology Development and Demonstration Programme (EUDP).

Forord

EnergyLab Nordhavn er et spændende projekt der løber til og med 2019. Projektet vil foregå i Københavns Nordhavn, og vil fungere som et fuldskala storbylaboratorium, der skal undersøge, udvikle og demonstrerer løsninger for fremtidens energisystem.
Målet er at finde fremtidens mest omkostningseffektive energisystem, der desuden kan bidrage til en løsning på de store klimaudfordringer verden står overfor nu og i fremtiden.

Budget: Projektets totale budget er DKK 143 mio. (EUR 19 mio.), hvoraf DKK 84 mio. (EUR 11 mio.) er blevet finansieret af Energiteknologisk Udviklings- og Demonstrationsprogram, EUDP.

Project Information

Deliverable no.: D2.6a
Deliverable title: Specification of data collection system
WP title: Data and Measurements
Task Leader: Anders Laage Kragh
WP Leader: Benny S. Hansen
Comment Period: 01-16-2019 to 01-30-2019

For further information on this specific deliverable, please contact:

Anders Laage Kragh
Technical University of Denmark
Frederiksborgvej 399
Building 776
4000 Roskilde
Denmark
Direct +45 93510782

For other information regarding EnergyLab Nordhavn, please contact:

EnergyLab Nordhavn Secretariat
Center for Electric Power and Energy, DTU Electrical Engineering
Elektrovej
Building 325
DK-2800 Kgs. Lyngby
Denmark

E-mail eln@dtu.dk
Tlf. +45 45 25 35 54

www.energylabnordhavn.dk

Table of Contents

List of Abbreviations – terminology

Data (measurements)	A measurement e.g. a temperature, CO2 level, switching on a switch in combination with a time stamp for the observation, see Figure 2
Data set	A grouping of one or more measurement data, a data set could be all data from an apartment or all data from all apartments in a building.
Metadata	Data describing measurement data
Data provider	The person or legal entity that make data available to the data responsible. The data provider may either collect the data or be the subject that is observed, e.g. a person's actions, see Figure 1
Data responsable	The person or legal entity (e.g. DTU) that do analysis, research or in other ways investigating the data. The data responsible is obliged to secure that the legal requirements are fulfilled (check definition here)
Data handler	A person or legal entity that on behalf of the data responsible operate the data (retrieve, store, make available for data responsible) but in no way study the data. (check definition here)
Data user (user)	A person who has access to the system and from the system can extract the data he can access. A user cannot define new measurements or define new data sets consisting of data from existing data.
Data administrator	In case the data responsible is a legal entity, the data administrator is the person that act on behalf of the data responsible.
Project	A grouping data sets. A project could be (ELN) EnergyLab Nordhavn
Export data	The result of a query, request for data
Open data	Data available to all, i.e. data that can be published on the Internet (under a given license)
Business critical data	Data that the data provider only allow to be shared with users that she approves. Data is not necessarily person-identifiable or personally sensitive data.

Person-related data	Personal-related data is data that is relatable to a single person (or household).
Particularly sensitive person data	This is data that “of other strict private nature” than e.g. political, health or sexual information regarding individuals
Group Administrator	A user who has rights to create groups and add and remove users from a group
Group	One or more data users who have the same rights to the same measurement data and data sets
System administrator	A person who has access to maintain DMS as an IT system and can create data administrators, groups and group administrators and define metadata
ELN	EnergyLab Nordhavn
Energydata.dk (DMS)	The name of the data warehouse. DMS = data management system

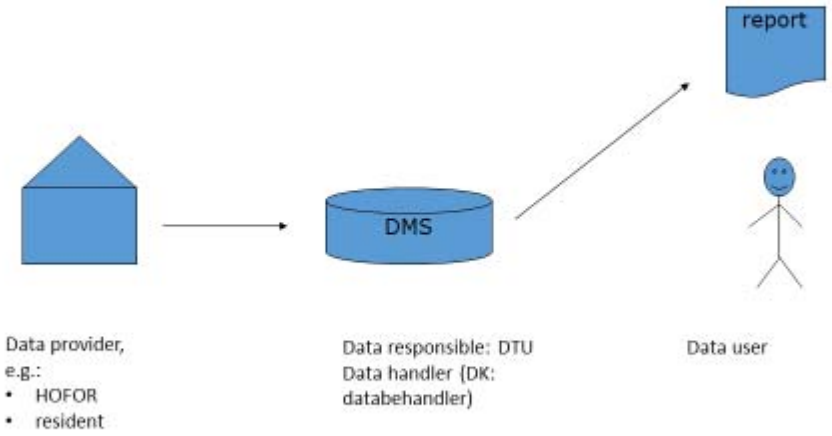

Figure 1 The different terms related to data

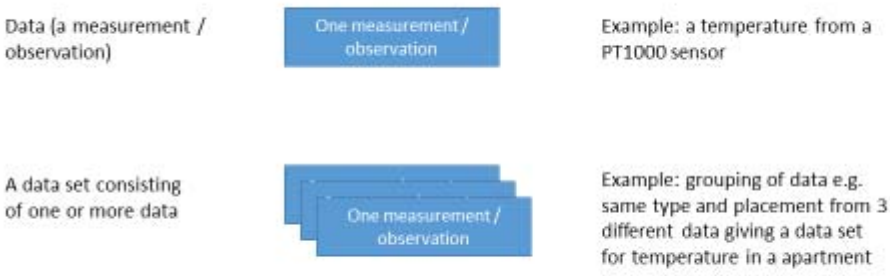

Figure 2 Definition of data and data set

Executive Summary

In this requirement specification a number of requirements are formulated for the Energydata.dk data management system (DMS). The requirement specification is not a complete and firm requirement specification, requirements may be removed or added as it is foreseen that the development of the DMS follows an iterative process and not a “waterfall” process.

The requirement specification is divided into a number of subsections each dealing with different parts of the DMS. Each subsection contains three parts:

1. System objective
2. Requirement formulation
3. Requirement List

The system objective serve as the goal for the requirement formulation and requirement list and shall be kept in mind in discussions about the detailed requirement formulation and list. The objective is what we want to achieve, the list is how we will achieve this.

Resumé

This document is the specification of data collection system for the EnergyLab Nordhavn project. The data collection system is named Energydata.dk or for short DMS (data management system). The specification consists of three major parts:

- Functional requirements for the data collection system, chapter 2
- The software selected for the system, chapter 3
- The hardware needed for the implementation, chapter 4

Version Control

Version	Date	Author	Description of Changes
0.1	[2018-12-05]	Anders Laage Kragh	First release for review
0.2	2019-01-07	Benny Stougaard Hansen	1'st review
0.3	2019-01-10	Benny Stougaard Hansen	2'nd review
0.4	2019-01-14	Benny Stougaard Hansen	Final review

Quality Assurance

Author	Reviewer	Approver
Anders Laage Kragh	Benny Stougaard Hansen	WPL group

Status of deliverable		
Action	**By**	**Date/Initials**
Sent for review	Anders Laage Kragh	**2019-01-03**
Reviewed	Benny Stougaard Hansen	**2019-01-14**
Verified		
Approved	WPL group	**2019-02-01**

1. Introduction

This document is the specification for the data collection system for the EnergyLab Nordhavn project. The data collection system is named Energydata.dk.
Originally is was planned, that the data collection system should be an extention to the SCADA system installed at CEE at DTU. However, for different reasons, it was decided to build a dedicated system. This document is a specification for the system.

2. Functional requirement for the data collection system

This chapter lists requirements for the data collection system. The list of requirements will most likely not be complete as new requirements may occur during development and operation of the system while others may be excluded. The requirements can be grouped into these groups:

- Requirements related to defining data
- Requirements related to receiving data
- Requirements related to data resilience
- Requirements related to administration of the system
- Requirements related to query and export of data; logging
- Requirements related to control signaling

2.1 Requirement attributes

Table 1 list four attributes that should allocated to each requirement in order to prioritize these during the implementation.

Priority	Semantics
Must have	Mandatory requirements that are fundamental to the system
Should have	Important requirements that may be omitted
Could have	Requirements that are truly optional (realize if there are time)
Want to have	Requirements that can wait for later releases

Table 1 Requirments attribues

From UML 2 and the Unified Process [Arlow & Neustadt]

This prioritization has not been performed.

2.2 Requirements related to defining data

Data is the key part of the system. The term data can be divided into three parts:

1) The measurement or observation itself e.g. the temperature in a room measured by a sensor and the time for the measurement. Several measurements from this sensor form a time series of observations
2) Description of the serie of measurements or observations all belonging to the same measurement device – meta data for the time series, e.g. the type of sensor, the placement of the sensor
3) Description of a single measurement – meta data for the observation, e.g. quality of the measurement, time source, validity

This means that data is a number of measurements with a time stamp associated to each observation. The time serie is described by metadata. It must be possible to describe the data accurately by the metadata. By doing so it should be possible to understand what is observed. Further as the object under observation may change it must be possible to change the metadata description. Further as it may be known for each observation that some external factors impact the observation or quality of the observation, it must be possible to associate each observation by metadata describing the observation. This could be the time stamp can come from two different clocks, that the sensor is known to be un-calibrated etc.

Further data can classified according to their sensitivity. This can be from a personal perspective be classified as:

- Anonymous data – can be distributed freely
- Personal data – cannot be distributed freely
- Sensitive personal data, cannot be distributed freely, subject to special data handling requirements and agreements

Please see Delivery 2.1C-1: Report on data handling requirements (Privacy)

In addition to the above data can be business critical data which means it is sensitive to a business partner and cannot be distributed freely.

Therefore access to data shall be protected, regulated and the access shall be logged.

From the above a number of requirement statement can be formulated. The list is not complete but shall serve as basis for development.

ID	Requirement	**MSCW**
	Define data	
	Data must be defined using metadata	
	Metadata is defined by the system administrator	
	It must be possible to load metadata via a file (batch definition of metadata)	
	It must be possible for a system administrator to edit metadata. Updated metadata must be approved by the system administrator before they are applied. Updated metadata only apply to data received after approval.	
	It must be possible to set a time when new metadata should apply.	
	It must be possible to define new data and data sets based on existing data and data sets	
	New data, data sets inherit access rights to data from its parents data	
	It must be possible for the data responsible to classify data as open, business critical, person-related and particularly sensitive person data	
	The data responsible must be able to change the classification of data. The changes must be logged.	
	A user's access to data shall be restricted in accordance to the rights she has to view data	
	A data set should be able to have more providers	
	The data responsible decides who has access to data	
	It should be possible in the metadata to include reference to further documentation for the data, e.g. a reference to a drawing showing the placement of sensors.	
	It must be possible to define data as belonging to a project	
	It must be possible to share data between several projects	
	It must be possible to define a test period. During the test period, data must be marked as "under test", all other requirements of this requirement specification must also apply to data "under test" unless otherwise stated. It shall be possible to mark several different test labels (categories)	
	It must be possible to archive data from a completed project	

2.3 Requirements related to receiving data

The main objective for the data reception sub system is:

- It must support a number of standard protocols for receiving data.
- It shall be possible for DTU to configure the above protocols, i.e. the implementation must be flexible so data elements can be added or removed without the need for programming
- It must be possible to add new protocols; in the future new protocols may be needed and the system must be designed so new protocols can be included in a smooth way.
- It shall be possible to import and receive data from other databases. The import can be used when replicating the data from the other database and the “receive” part when subscribing to data in another database so whenever new data is inserted in source database, this data is replicated in the DMS
- It must be possible for DTU to add new protocols without external assistance, i.e. the system must support a kind of generic data interface which DTU can utilize for creating its own protocol implementation.

This can be translated into this summary:

It’s mandatory to receive data from a number of different data sources this could be measurement equipment installed in apartments, in the energy infrastructure (pipes and pumps) or from other databases. Several protocols for receiving data need to be supported and it is designed to easy add new protocols , configure the format for data reception following a given protocol without the need for re-programing the interface. E.g. data may be received using FTP then it shall be possible to configure the data received – add or remove data fields, the separator (semi colon, comma tabulator), the frequency for data exchange and have several data streams in parallel, i.e. receiving price information from one data provider and information for district heating from another data provider.

It must be possible for DTU to program and implement a new protocol for receiving data (the “red box” in Figure 3). This mean that the DMS must have a generic API for inserting data into the DMS (the “green box” in Figure 3).

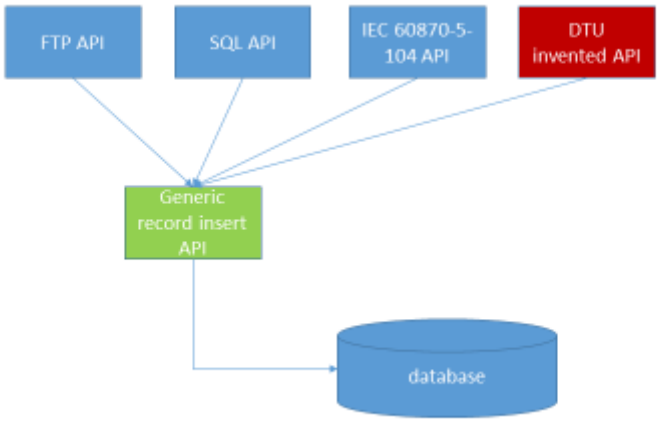

Figure 3 API for receiving data

From the above a number of requirement statement can be formulated. The list is not complete but shall serve as basis for development.

ID	Requirement	**MSCW**
	Receive data	
	It must be possible to receive data according to the following standard protocols, • KNX • FTP (csv files) • IEC 60870-5-104 • Modbus	
	It must be possible to add new data protocols to receive data	
	It must be possible to receive data according to a non-standardized protocol if there is a specification of the protocol.	
	It must be possible to receive data from other database systems e.g. SCADA-systems and building management systems (BMS)	
	It must be possible to specify a frequency for receiving data for a given interface; If no data is received, an alarm shall be generated	
	It shall be possible to monitor the communication lines between the DMS and external systems generating the data	
	The system must be able to poll for data	
	The system must be able to receive data as push data	
	It must be possible for existing measurement data to replace the sensor, calibrate the sensor or similar work on the sensor without the	

	need to define new measurement data but just to update metadata to reflect the change.	
	It must be possible to send data via encrypted connections from the sensors to the DMS system	
	The system must be able to mark data as faulty if validation rules for the data exist	
	It must be possible to import existing measurement data from other systems, databases or files	

2.4 Requirements related to data resilience

Data shall be protect against data loss, see Delivery 2.1C-3: Report on Cyber Security. The loss can happen because of several incidents:

- Fire or other disaster at the building housing the computer equipment
- HW failure in the computer equipment, e.g. a disc crash
- SW failure impacting the access to data, e.g. in relation a SW release upgrade / update
- Failure by a human, e.g. deleting data by accident

To protect against the above the system must be deployed to protect against single site disaster. This can be done as 1+1 deployment (completely redundant) on two sites or as clustered deployment on two or more sites, where data is replicated among the clusters and failure on one site will imply lower performance but no data loss. Please note, that the current deployment does not fulfil this.

To protect against SW failure and faulty operation of the system several generation of backups may be considered so data can be restored from generation n-1, n-2, n-3 ect, where n is current generation, n-1 is the version before etc.

The term data shall be understood broad, it applies to both the data (measurements), the metadata and all the SW application, configuration files, scripts, log files etc.

A list of requirements from above considerations are shown below.

ID	Requirement	**MSCW**
	Data resilience	
	There must be a secondary system for receiving data	
	It must be possible to direct data flows to a secondary system in case of system maintenance on the primary system	
	Data must be automatically received on the secondary system if the primary system is out of order (error situation)	
	It must be possible to synchronize (received) data between the primary and secondary systems	
	It must be possible to do a regular backup of data without affecting the system's ability to receive data.	
	It must be possible to restore data from a backup	

2.5 Requirements related to administration of the system

When in operation the system shall be managed and maintained as any other IT system and to do this a number of system administrators shall be appointed. The system administrators shall maintain the system, which include but is not limited to:

- Monitor and act on alarms from the system.
- Create new users in the system with appropriate access to data.
- Remove existing users if they no longer need access to the system.
- Add new data sources
- Configure and maintain system configuration files and script
- Supervise the utilization of the, act on scare resources, e.g. disk

As a guiding rule DTU must be able to do daily operation of the system without the need to involve the supplier in solving these tasks. To obtain this, the system must be in a kind of “released version”, i.e. no programing skills are required and “no hacks” are needed to operate the system. Further, the documentation and work flow must support this.

A more detailed list of tasks as inspiration and checklist can be found below, but the above is the guiding principle.

ID	Requirement	**MSCW**
	Administration of the system	
	Access to the system must be checked by a login system. In the login system, the user must be defined so accurately that the user can be identified in person. That is, the user must be defined by the full name, full address, telephone and e-mail.	
	System Administrator must be able to see which users have access to the system.	
	It must be possible for a data responsible to grant a user access to data from individual measurement series and / or data sets.	
	A data responsible can only grant access to the data he or she is responsible for.	
	It must be possible for a data responsible and system administrator to deprive (recall) a user access to data, individual measurement series or / and data groups.	
	System Administrator must be able to see which users have access to data grouped by which data is available for the user	
	System Administrator must be able to see which users **have had** access to view data (log) grouped by what data has been accessed and in which periods (time).	
	System Administrator must be able to see what data a user has access to.	
	System Administrator must be able to see what data a user has accessed.	
	System administrator must be able to deprive (deny) a user access to the system.	
	A system user who is denied access to the system must not be deleted if any existing logs is referring to this user.	
	System administrator must be able to define a group of users.	
	System Administrator must be able to add a user to an existing group of users.	
	System administrator must be able to remove a user from a group of users.	
	A user must be able to be a member of several groups.	
	It must be possible to be both a normal user (by someone else's data) and data administrator for own data.	
	System Administrator must be able to define projects in the system.	
	A user must only be able to see the existence of data from the projects	

	she is affiliated with - the right to see the existence does not imply the right to view data.	
	A user will only be able see the projects he or she has access to (it must be possible to hide projects, keep them secret).	
	It must be possible to create a system administrator for the system. The system administrator shall be able to: • Start and stop the program and its subsystems • Configure the system • Add and configure new sensors • Remove sensors • Add and remove programs, scripts, components • Responsible for monitoring the operation of the system • Configure hardware for the system • Add and remove users to the system • Assign and remove data administrators and group administrator rights to users	
	Projects and data shall be visible unless explicitly asked for to be hidden.	
	An alarm must be raised if a connection to a sensor is out of order or no data has been received according to an expected plan.	
	Alarms shall be cleared if the problems vanish.	
	There must be a log of alarms.	
	It must be possible to see how many requests each user makes.	
	It must be possible to see which user has been logged into the system grouped by day and user for a period defined by the requester.	
	It should be possible to see which users are defined in the system but who have not been logged in for a period defined by the requester.	
	The system must raise an alarm if it lacks disk space and / or memory.	
	There must be performance statistics for the system that shows at least: • CPU load • Disk utilization • Memory utilization	

2.6 Requirements related to query and export of data; logging

Ideally, it shall be possible to browse the stored data either directly from the system or from another application, which access the system and present the data to the user. When

doing such a browsing the user should be able to pan the data in the search for data of interest. It should be possible to filter; e.g. to set some criteria for the selection of data and to combine more data set is the search and browsing.

The browsing should be done on the fly by selecting from a list of options (e.g. data set, setting criteria ect) and then have the result presented directly on the screen.

As basic requirement it shall be possible to select a data set or subset from a data set, e.g. a certain period and then have an export of these data for off-line handling in another application as Matlab, R etc.

A requirement list is presented below.

ID	Requirement	**MSCW**
	Query and export of data; logging	
	It should be possible to export data in the following format: • Csv	
	Export data files must be named in a unique way with any eventually sequence numbers if export data is divided into multiple files	
	A user must be notified when the result of a query is available, i.e. data in an export is available	
	Data in an export file shall be automatically deleted after 72 hours	
	It should be possible to define that data is exported according to a plan, such as on certain time, on certain dates	
	Requests (queries) for data marked as business critical, person-identifiable and particularly sensitive person data must be logged. In the log it shall be noted: • The user • The data that has been queried and presented to the user	
	The result of a query that includes business critical, person-identifiable and particularly sensitive person data must be password protected. Password must be equal to the user's password for the system.	
	It must be possible to query data from a program using an API.	
	It should be possible to forward data in real time, i.e. received data is forwarded as soon as they are received	
	It must be possible to query data that meets a search criteria	
	It must be possible to define a query criteria as greater than, less than or within an interval	
	It should be possible to define query criteria across data, such as a	

	temperature sensor and a wind turbine	
	It should be possible to share anonymous and open data with other data portals, such as Opendata.dk	
	It shall be possible to browse / have data presented in order to view data before data is requested for export	
	It shall be possible to export in different resolutions (lower resolution than received)	
	It shall be possible to save a query and re-use it later. It must be possible to have history of queries.	

2.7 Requirements related to control signaling

The DMS including the interface applications to the data sources is more than a simple datawarehouse. It shall support the ambition of smart usage of energy. In order to support this it must be possible for external control applications to subscribe to data from the DMS. When a subscription to data is made, the DMS will forward the subscribed data to the control application, which then, based on these data and eventually other data and some control algorithm decide for a feed-back signal. This feedback signal may be send to the relevant interface application including a destination (recipient) for the signal and other relevant information for the transmission of the signal, e.g. time to live – expire information for the signal.

Where the header part contains the recipient address and other information related to the transmission of the signal and the data part contains the actual information.

When such a signal is received at the interface application, it may analyze the header part and act on this, e.g. forward the signal, drop it if it has timed out etc. or simple just forward the signal as is.

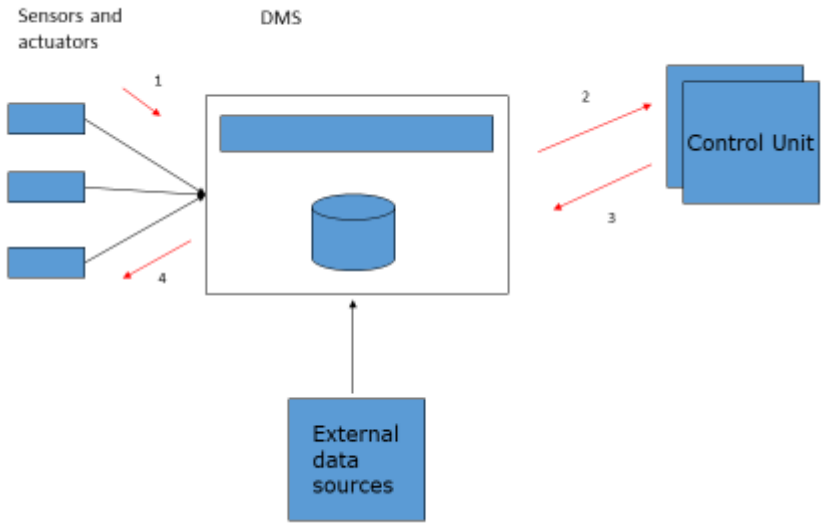

Figure 4 Control logic

Some of the implemented protocols may support two-way communication, e.g. KNX. If so, then these sending parts of these protocols must be supported as-is.

3. Specification for software

There exist several software solutions for Datawarehouse for time series data / event based data. For the energydata.dk solution, it was decided to select software that fulfilled these criteria:

- Open Source, i.e. license free
- Has achieved a certain level of maturity
- Supported by major companies / organizations

Due to these reasons it was decided to base the development of Energydata.dk upon the following solutions:

- Kafka – for handling the data streams
- Cassandra – for data storage
- Spark – for data processing
- Zookeeper – for managing the distributed deployment

These are all supported by the Apache software foundation.

In addition to these (major) component, the following software solutions are used:

- NGINX – web and mail server
- PostgreSQL – SQL database (for handling metadata)
- VerneMQ – MQTT broker

Kafka is a publish - subscribe messaging system built to operate as a cluster. It is designed to allow a single cluster to serve as the central data backbone for a large organization and can elastically and transparently be expanded without downtime. Data streams are partitioned and spread over a cluster of machines to allow data streams larger than the capability of any single machine and to allow clusters of coordinated consumers. It was initially developed at LinkedIn and is in production in a number of large companies.

Cassandra is a NoSQL database which offers the convenience of column indexes and focuses on scalability and high availability without compromising performance. When more performance is needed then simply add more nodes to the cluster and Cassandras proven linear scalability comes into action. Cassandra supports best-in-class replication of data.

Spark is a powerful open source cluster computing framework which is built around speed, ease of use and sophisticated analytics. Spark is the largest open source project in data processing and since its release, Spark has been adopted by enterprises such Yahoo and Tencent in clusters with over 8.000 nodes. Spark supports SQL queries, streaming data, machine learning and graph processing which cover the use we need for this platform.

ZooKeeper is a high-performance coordination service for distributed applications. It exposes common services - such as naming, configuration management, synchronization, and group services - in a simple interface so it is not required to write them from scratch.

NGINX is a free, open-source, high-performance HTTP server and reverse proxy, as well as an IMAP/POP3 proxy server. NGINX is known for its high performance, stability, rich feature set, simple configuration, and low resource consumption.

PostgreSQL is a powerful, open source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads. The origins of

PostgreSQL date back to 1986 as part of the POSTGRES project at the University of California at Berkeley and has more than 30 years of active development on the core platform.

VerneMQ is a MQTT publish/subscribe message broker which implements the OASIS industry standard MQTT protocol. VerneMQ is built to provide a unique set of features related to scalability, reliability and high-performance as well as operational simplicity.

4. Specification for hardware

Given the chosen software for the development for Energydata.dk the hardware has been chosen to meet the recommendation for each software component.

To run the system fully optimal a specific set of requirements for the hardware needs to be fulfilled and the different servers need to be deployed in a cluster form, i.e. several servers connected and orchestrated by a central (duplicated) manager.

The following requirements will be the minimum cluster form for the system:

- 3 servers for Cassandra and Spark
- 2 servers for Kafka
- 3 servers for Zookeeper

The following requirements is per server.

4.1 The setup for one Spark and Cassandra server

The Cassandra and Spark server is an important part of the system that needs to run in the most optimal way. To meet the requirements Uptime and WP2 recommended these hardware requirements for Cassandra and Spark technologies (follow the links):

- Cassandra
- Spark

These specs are slightly above the minimum recommended requirements. They are expected to provide the best performance for the investment. If more performance is needed, this can be achieved by adding extra nodes to the cluster.

The size of disks for Cassandra has been chosen by estimating the amount of data that is to be stored in the database and taking into account the overhead required by compaction and replication.

The resulting server specifications are as follows:

- 2x Intel Xeon Processor E5-2630 v3 (20M Cache, 2.40 GHz 8 Core)
- 8x 8GB DDR4-2133 1R*4 ECC RDIMM
- 4x Intel DC S3510 800GB MLC
- 5x Intel DC S3510 240GB MLC

4.2 The setup for one Kafka server

Kafka is very efficient at handling many concurrent read and write operations. RAM and CPU are the most critical aspects for performance. Kafka is often used for deployments much larger than Energydata.dk. Thus, the specifications are modest compared to most recommendations.

The recommendations are based on information in the following links:

- Kafka documentation
- Kafka deployment at LinkedIn
- Kafka deployment at Confluent

Server specifications:

- 1x Intel Xeon Processor E5-2630 v3 (20M Cache, 2.40 GHz 8 Core)
- 4x 4GB DDR4-2133 1RX8 1.2V ECC RDIMM
- 2x Intel DC S3510 240GB MLC

4.3 The setup for one Zookeeper server

Zookeeper requires very little in terms of hardware requirements, but the official documentation cautions against using virtualized servers for Zookeeper, for that reason a physical server will be used. In addition, it is important to avoid swapping to disk.
The specification for the server is based on recommendations in the following link:

- ZooKeeper Administrator's Guide

Server specifications:

- 1x Intel Xeon Processor E5-2623 v3 (10M Cache, 3.00 GHz 4 Core)
- 4x 4GB DDR4-2133 1RX8 1.2V ECC RDIMM
- 2x Intel DC S3510 240GB MLC

4.4 Server requirements for VerneMQ, NGINX and PostgreSQL

Due to Kafkas modest server requirements it was decided to implement these application on one of the Kafka servers:

- VerneMQ
- NGINX
- PostgreSQL

4.5 Deployment

The servers are deployed as shown in Figure 5

Figure 5 Energydata.dk server deployment