NTNU
Norwegian University of
Science and Technology

Building a Data Management Architecture for Zero-Emission Neighbourhoods in Smart Cities

Andreas Midjås Ystmark

Master of Science in Computer Science
Submission date: June 2018
Supervisor: John Krogstie, IDI

Norwegian University of Science and Technology
Department of Computer Science

Problem Description

In smart cities and their subdomains, data needs to be collected from a large number of relevant sources in a variety of formats. This data needs to be aggregated and visualised for different stakeholders in a cohesive manner. The task relates to creating such an architecture in a way that makes it easier to transfer solutions for smart cities and communities to other cities. A zero-emission neighbourhood data management ICT system is to be built, based on the architecture, to assess whether or not the created architecture is applicable for a smart city domain.

Abstract

Smart cities generate large amounts of data from a range of sources, like sensor networks, social media and general citizen-city interaction. In order for the data to be valuable for citizens and decision-makers, the data needs to be managed cohesively. The data should be collected, stored, processed and disseminated in such a way that it can benefit the city, either in terms of providing data for other smart city applications, or through providing information that helps citizens and decision-makers gain knowledge about the city.

This thesis proposes an architecture for smart city ICT systems that consists of components for data collection, storage, processing, and dissemination, and detail how these can be built and interact with each other. The architecture focuses on how the collected dat is stored, transformed, and how to effectively disseminate the data to different users. In order to research how well the architecture can be applied to a real smart city context, the architecture is implemented as a prototype for a data management and Key Performance Indicator (KPI) monitoring system for zero-emission neighbourhoods in smart cities. This system is intended for use in the FME ZEN research centre pilot projects designing, creating and testing zero-emission neighbourhoods. The prototype translates the components of the architecture into technological modules that manage the associated aspects. The prototype implements the collection, storage, and transformation of raw data to KPI data for the ZEB Living Laboratory. It then disseminates the data through an open data platform and a web based interface. This thesis shows that a data management system for zero-emission neighbourhoods in smart cities can be built using the suggest architecture. This can hopefully contribute to further the work of creating a full data management ICT system for the ZEN research centre pilot projects and improve knowledge related to smart city ICT architecture.

Sammendrag

Smarte byer genererer store mengder data fra data kilder som sensor nettverk, sosiale medier og interaksjoner mellom innbyggere og byen. For at denne dataen skal skape mest verdi for innbyggere og beslutningstagere må dataen bli håndtert med perspektiv på hele prosessen. Dataen må bli samlet inn, lagret, prosessert og publisert på en slik måte at den kan bli dratt nytte av for å utvikle byen videre, enten gjennom å samle data for andre smart city applikasjoner, eller gjennom å gi informasjon som hjelper innbyggere og beslutningstagere med å få ny kunnskap om byen.

Denne oppgaven foreslår en arkitektur for smart city IKT systemer som består av komponenter for data innsamling, lagring, prosessering, og publisering. Arkitekturen beskriver også hvordan disse komponentene er bygget opp, og hvordan de samhandler. Arkitekturen fokuserer på hvordan den innsamlede dataen blir lagret, hvordan den transformeres, og hvordan den best kan vises for for forskjellige brukere. For å undersøke hvordan arkitekturen kan brukes i sammenheng med smarte byer, har den blitt implementert som en prototype av et IKT system for data håndtering og overvåking av ytelsesindikatorer (KPI) i nullutslipp-nabolag. IKT systemet er tenkt til bruk i pilotprosjektene til forskningssenteret FME ZEN der nullutslipp-nabolag i smarte byer utvikles. Prototypen oversetter de forskjellige komponentene fra arkitekturen til teknologiske moduler som håndterer de forskjellige aspektene ved systemet. Prototypen tar for seg innsamling, lagring, og transformering av data rådata til KPI data for ZEB Living Laboratory prosjektet. Deretter publiserer prototypen den transformerte KPI via både en åpen data platform og et web-basert grensesnitt. Denne oppgaven viser at et data-håndtering system for ZEN kan utvikles ved hjelp av den foreslåtte arkitekturen. Dette kan forhåpentligvis bidra til arbeidet med å lage et fullverdig data-håndtering system for pilotprosjektene til forskningssenteret ZEN.

Preface

This thesis is the final part of my Masters of Science degree in computer science at the Norwegian University of Science and Technology (NTNU). The project was conducted under the Department of Computer Science (IDI).

I would like to thank Professor John Krogstie at the Department of Computer Science, my supervisor for this project, for his support and guidance during this thesis. I would also like to thank Jonas Wäfler at Powel for suggesting and encouraging this project for my thesis.

Andreas Midjås Ystmark
June 2018

Table of Contents

List of Tables

List of Figures

Abbreviations

ICT	=	Information and Communication Technology
FME	=	Centres for Environment-friendly Energy Research
ZEN	=	Zero Emission Neighbourhood
ZEB	=	Zero Emission Building
GHG	=	Greenhouse Gas
LCA	=	Life Cycle Assessment
IoT	=	Internet of Things
RDI	=	Restricted Data Interface
ODI	=	Open Data Interface
KPI	=	Key Performance Indicator
USD	=	United States Dollar
NTNU	=	Norwegian University of Science and Technology

Chapter 1

Introduction

1.1 Introduction

Over half of the worlds population lives in cities, and with current globalisation and centralisation that number is bound to increase [56]. Housing and infrastructure in urban areas is being stretched to accommodate the rise in population, and cities need to continue to provide new and sustainable ways of planning and improving our current and future cities. Smart cities are one way of developing and achieving modern and more sustainable urban areas.

There is no single definition of what a smart city is, but a general description is a city where ICT plays a significant role in urban planning, municipal services, and the day-to-day routines of its citizens. Some of the objectives of smarter cities is for them to be environmentally sustainable, efficient and more liveable for their citizens. Smart cities gather data via Internet of Things devices, sensors, GPS, data from mobile devices and a variety of different data repositories in order to make data driven decisions about innovation and improvements of the city. A robust data management ICT system is required to make the optimal decisions with the data collected.

1.2 Motivation

The motivation for this research is based on the Zero-Emissions neighbourhoods [59] research centre and its work. The purpose of ZEN is to research and design solutions for future zero-emission neighbourhoods in smart cities. In order to test these solutions, the research centre is developing several pilot projects. These projects will generate large amounts of data that needs to be managed. The data generated by the pilot projects can provide valuable information for future development of projects and further research. Finding value and information in a large amount of data can be difficult. An ICT system for collection, storage, analysis, and dissemination of these data is needed to organise and discover information and gain new insight and knowledge from the data. This system requires require a detailed architecture to ensure the quality of the data management. In this project, such an architecture is designed and implemented as a ZEN data management system prototype.

1.3 Project Description

This thesis is a continuation of a specialisation project from 2017. The specialisation project conducted a state-of-the-art literature review of smart city architectures and proposed a conceptual architecture for use in urban mobility applications [51].

Building on that previous project, the architecture is improved and modified to suit general smart city systems as well as the zero-emission neighbourhood domain. It is then adapted to and implemented as a data management and KPI monitoring system for ZEN research centre pilot projects.

The purpose of this thesis is to design and implement an ICT architecture and system for data management in smart cities. The implementation is a prototype of an ICT system for monitoring key performance indicators in pilot projects for the ZEN research centre. Finally the thesis evaluates the architecture in the context of the implementation domain to assess if the architecture is suitable for building a full data management ICT system in the zero-emission neighbourhood domain of smart cities.

1.4 Thesis Outline

Past this first, introductory chapter, the rest of the thesis will be structured as follows: **Chapter 2** is the theoretical background on smart cities, zero-emission buildings and neighbourhoods, and the technologies that facilitate the development of these areas. The chapter also reviews some previously designed and implemented smart city architectures from literature. **Chapter 3** presents the chosen research method and research questions for the thesis. **Chapter 4** defines stakeholders and requirements for the proposed architecture, and describes the architecture and its components. **Chapter 5** presents the implementation process and the functionality of the prototype. **Chapter 6** shows the results of the implementation. **Chapter 7** discusses and evaluates the prototype and architecture. The last chapter is **Chapter 8** which concludes the thesis and summarises its contribution.

Chapter 2

Background

This thesis proposes a smart city architecture and a prototype a architecture zero-emission neighbourhood data management system utilising that architecture. The zero-emission neighbourhood is a domain of the smart city that relies on many of the same technologies and solutions. The background chapter of the thesis will start by giving a general overview of the smart city, components, and challenges. The chapter then goes on to describe zero-emission buildings and neighbourhoods and presents the FME ZEB and FME ZEN research centres and their projects, which are the basis for the prototype that is developed later in the thesis. Lastly, the chapter describes the main technological paradigms in smart cities, as well as architectures that have been used in other smart city contexts and applications, which will used as a basis for the architecture developed later in the thesis.

2.1 Smart Cities

The concept of a smart city has been around for some time. Using Information and Communication Technology (ICT) to modernise cities and facilitate more efficient city planning and management is an established idea [43][16]. The most prominent technologies that are used in modern smart cities are evolving and maturing rapidly, and more and more cities are taking advantages of these technologies and developing smart city programs.

Smart cities have been defined many times in many different ways [2]. There is no concrete consensus on what the official definition of a smart city is, but definitions can often be divided into two separate categories [25]. Category one is considered more technically oriented. Definitions from this category determine the smartness of a city by the level of “connectedness” of the city. It argues that IoT and ubiquitous computing devices (everyware) provide the city with data to mange, regulate and plan urban areas, thus creating a smarter city. The main idea behind the category of definitions is that it is the technology and data collection that make the city inherently smart. The other category sees ICT as facilitating the government and citizens to make the city smarter via innovation in both professional and public services. Definitions in this category focus on the innovation, creativity and entrepreneurship of smart people as a determining factor of the smartness of the city, rather than the degree of which the city is digital and connected. So although the smart city of the second category of definition relies on ICT and the connectedness in the city, it is different from the first category in that while ICT is crucial in facilitating the smartness, it is not simply the amount of technology or data collection in the city that makes it smart, it is how citizens and decision makers use that data and technology. The common ground for both categories is that ICT is a crucial part of solving management and planning issues in urban spaces and creating smart cities.

The smart city impacts many different aspects of urban management and planning [23][7]. Shown in figure 2.1, it concerns it self with mobility, the environment, the economy, governance and the lives of its citizens. In 2016 34% of European cities had developed some form of smart city initiative or application applicable to one of the areas [23]. The article also lists some more concrete domains within the smart city areas where ICT can be utilised to meet the needs of citizens and create a smarter, sustainable city:

- Safety and security
- Environment and transportation
  - Controlling pollution levels
  - Smart street lights
  - Congestion solutions
- Home energy management
- Educational facilities
- Tourism
- Citizens health

Some enabling technologies for smart cities are identified in [23][47]. These technologies facilitates smartness in the city and help transform and modernise the different domains:

- **Big data:** Addresses the challenges of capturing, storing, searching, processing, analysing, and visualising large amounts of data. In the context of smart cities, data from sensors, social media, documents, and other sources needs to be collected, stored, and processed to give context to, and information about what is happening in the city. To achieve this, a big data framework is needed in the city.

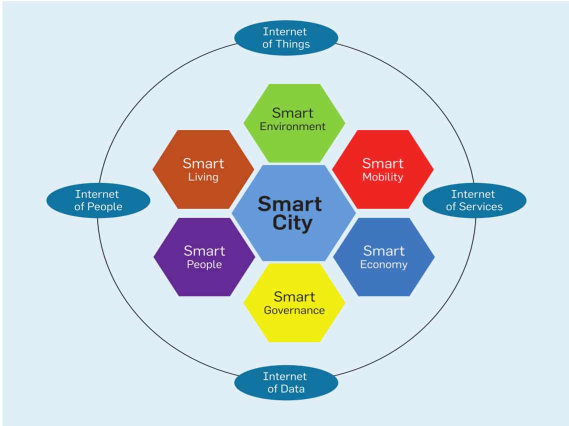


Figure 2.1: Overview of different areas of a smart city [23].

- **Networking:** Both wired, wireless and mobile networks must be robust to ensure reliable communications across the city, consistency in data collection and security. Unreliable networking and network security can cause data to be lost or stolen, and can cause critical failures in areas of the smart urban environment relying on consistent networking.
- **IoT:** Sensors and other connected devices in the Internet of Things monitor many aspects of the city, such as noise, traffic congestion, energy consumption, smart parking, and smart lighting. The devices can either control the aspects they are monitoring (like automatically closing entry to a parking garage when sensors pick up that the garage is at capacity) or report the gathered data to the city in order to help people manage the city or make decisions about future solutions.
- **Cloud Computing:** Allowing cities to access shared, configurable and reliable computing resources in the cloud can help the manage and lower its ICT cost for the smart city [34]. In addition to lowering costs, it allows for cities to access computing and storage infrastructure and applications they may not have the resources to create and manage themselves.

There are many technical challenges that need to be tackled when developing a smart city. The size and variety of the data being processed may be too complex for traditional data mining, leading to the need for big data techniques. Data sharing between collaborating private and government organisations in the city may offer challenges in regards to privacy. Different organisations have different standards for storage, management and distribution of data. In order to get a more complete view of the data ecosystem and make more informed decisions, the city needs to facilitate data sharing between different entities. Increased energy consumption due to the addition of ICT solutions is also an issue. One of the goals of many smart cities is environmental sustainability [39] and increased energy consumption may impede that goal unless cities focus on creating renewable, low-energy solutions at the same time.

2.2 Zero Emission Buildings and Neighbourhoods

Reduction of greenhouse gas emissions is an important aspect, and a driving factor for developing smart cities [23]. Managing the energy consumption of the building sector is key to creating a smart, sustainable city. In 2014, the building sector accounted for nearly 40% of total energy consumption and 36% of greenhouse gas emissions in the EU [40].

The energy consumption from buildings, especially in cities, accounts for a large portion of the worlds greenhouse gas emissions [26] making the reduction of energy consumption a priority in urban planning and development. This has lead to new low-energy and passive house [38] standards in construction and operation for zero-emission buildings, contributing to minimising energy consumption and greenhouse gas emissions of both new and existing structures. In reducing energy consumption, the term net zero-energy [33] is used to highlight the relationship between energy consumed by the building and energy generated by the building that is supplied back into the electrical grid. On a larger scale, zero-emission neighbourhoods are defined in terms of a collection of buildings connected by infrastructure [33]. Neighbourhoods take into account emissions and energy consumption of a range of other aspects of urban environments such as open spaces (roads, green spaces), networks (water, telecommunications, sewage, heating distribution, electricity distribution) and mobility [31].

To further contribute to reduction in GHG emissions, Zero-emission buildings and neighbourhoods consider their emissions over the entire lifespan of the building or neighbourhood, not just when the building is in use. The lifecycle of a building or neighbourhood includes planning, construction, operation and end-of-life phases of the lifecycle [31]. While making the building reach net-zero energy consumption is still an important aspect of the building and neighbourhoods in relation to reducing total GHG emissions, other factors also need to be considered in order to reach carbon-neutrality. Aspects such as extraction of raw materials, transport of construction materials, construction and waste disposal need to be managed in order to reach more carbon neutral buildings and neighbourhoods.

In Norway, there are a few research centres working on solutions for zero-emission buildings and zero-emission neighbourhoods through The Centres for Environment-Friendly Energy Research Program (FME). The first one was the research centre on Zero-Emission Buildings (ZEB), and later, the research centre on Zero-Emission Neighbourhoods (ZEN).

2.2.1 ZEB - Zero Emission Buildings

The Research Centre on Zero Emission Buildings (ZEB Centre) is a research centre started in 2009 and ended in 2017 dedicated to research on near zero emission planning, construction and operation of buildings. It was an early proponent of measuring greenhouse gas emissions in a lifetime perspective.

A lifetime perspective on emissions refers to all emissions across the entire lifecycle of a building. This lifecycle is divided into stages, which for ZEB are defined as[12]:

1. **Product Stage:** Extraction of raw materials and manufacturing the products used in the buildings
2. **Construction Process Stage:** Transportation of the products to the building site, construction of the building, and energy and waste associated with the construction.
3. **Use Stage:** Operation of building, including maintenance, repair, and replacement of products in the building.
4. **End of Life Stage:** Demolition of the building and disposal of waste.

The precise objective of ZEB defined in [12] is: *to develop materials and solutions of new and existing buildings resulting in zero greenhouse gas emissions over the lifetime of the assets, limited to the boundary of a building and the building site.*

To offset the GHG emissions in these stages, the ZEB should become a net-producer of energy in the use phase. If a building can produce more energy than it consumes, so the energy can be delivered to the surrounding electrical grid, it will reduce the need for external energy produced by energy sources that produce GHG emissions. The ZEB research centre created test facilities to try different solutions for developing zero-emission buildings.

2.2.2 NTNU ZEB Living Laboratory

The Zero Emission Building Living Laboratory is a test facility built by ZEB in order to investigate a number of factors related to zero-emission buildings.

The initial goal of the Living Lab building was to create a energy positive cabin. A cabin in this sense is a second home, primarily used by Norwegians during weekends and holidays. It is characterised by being close to nature, rustic and lacking in modernity. It was designed to be independent of the energy grid and with a low environmental impact. Despite its original idea, it was later developed into being a multi-purpose testing facility for testing building equipment components, ventilation strategies, and lifestyle and technology challenges, where the ZEB centre studies interactions between users and the zero-emissions buildings. Moving away from the cabin use case, the Living Laboratory has been altered to represent a typical Norwegian single family detached house. The house has a floor area (heated surface) of approximately 100 m^2 and a volume of approximately 500 m^3. Designed as a single family home, the floor plan of the building is designed to be as flexible as possible, accommodating couples, families, and students [15].

The house is outfitted with technologies for energy conservation and renewable energy generation in the form of solar energy exploitation. The goal of the Living Laboratory is to demonstrate how CO_2 neutral construction and building operation can be performed in a Nordic climate and environment. The building is ultimately designed to minimise energy demand in the operational phase and self-generate more energy through solar energy harvesting than it demands on a yearly basis through active and passive measures. Heating, ventilation and hot water demand is primarily covered with a ground source heat pump. The on-site renewable energy harvesting is done with two solar panels.

Monitoring and Sensors

In order for the Living Laboratory to be useful as a experiment and testing facility, a Data Acquisition (DAQ) system is implemented in order to:

- Monitor both indoor and outdoor environmental qualities such as air temperature, humidity, pressure, CO_2 concentration and solar radiation.
- Monitor user behaviour in the Living Laboratory such as room occupancy, window opening/closing, use of appliances and lighting.
- Measure energy used in heating, hot water, lighting and other uses.
- Measure self-generation of energy through solar panels.

In order to measure and record these factors and qualities, the ZEB Living Laboratory is outfitted with a range of sensors both outside and inside the building.

Outdoor Sensors

For outdoor measurements, a weather station equipped with sensors for air temperature, relative humidity, barometric pressure, wind velocity and global solar irradiance is situated above the roof of the building. Global solar irradiance is measured in two other locations as well (roof slope plan and south façade). A luxmeter is also placed on the roof in order to measure global illuminance. There are two additional sensors measuring outdoor temperature situated on the north and south façade of the building respectively. Both sensors are protected against influence from direct solar radiation.

Indoor Sensors

Many of the indoor qualities are measured separately for each room. Air temperature, relative humidity, and CO_2 concentration are all measured individually in each room.

Occupancy patterns of each room of the building is measured with motion detectors as well as sensors monitoring position (open/closed) of windows as well as use of artificial lighting.

Energy Use and Generation

Energy use and supply is monitored for each system requiring or producing energy:

- Energy use
  - Heating
  - Ventilation
  - Domestic hot water
  - Lighting
  - Appliances
  - Other uses
  - Control of automated windows and shading systems
  - Monitoring and control system for sensor network.
- Energy generation
  - Power converted by means of PV roofs
  - Power supply from grid
  - Thermal energy output from solar thermal panels
  - Thermal energy extracted from surface collector field

A comprehensive description of all sensors and data collection in the ZEB Living Laboratory can be found in [15]).

2.2.3 ZEB Dashboard Tool

For the FME ZEB project, a Microsoft Excel based ZEB tool was developed in order to calculate the life cycle assessment for buildings, and give an understanding of the impact of building materials on greenhouse gas emissions within the system boundary. The tool calculates the GHG emissions over the entire lifecycle of the building. A visualisation dashboard for this excel tool was created in [50]. The dashboard makes it easier for non-expert decision makers to interact with the tool to see the GHG calculations for various building material choices, and use that in their LCA. The dashboard uses the database of GHG emissions for different construction materials, and based on the choices of the user, the total estimated GHG emissions are calculated. Due to its focus on the GHG emissions of the building materials used, it is a tool primarily used in the early planning phase of the LCA and zero-emission building planning.

2.2.4 ZEN - Zero-Emission Neighbourhoods

The ZEN research centre is an NTNU affiliated research centre dedicated to sustainable and zero-emission neighbourhoods. It takes the issues researched in ZEB, and contextualises them in the context of zero-emission neighbourhoods and smart cities. ZEN has a broader focus, researching how to provide sustainable transport, economic sustainability and innovation in the neighbourhood, along with strategies for zero-emission construction of buildings and infrastructure. ZEN is conducting several pilot projects that are currently in different stages of building zero emission neighbourhoods, and the common research question for these projects is *How should sustainable neighbourhoods of the future be designed, built, transformed, and managed to reduce greenhouse gas emissions towards zero?*

Terms and Definitions in ZEN

In order to understand how to properly assess the ZEN research question, it is important to have a common understanding of the terms used in relation to ZEN [55].

Neighbourhood: A group of interconnected buildings with distributed energy resources such as solar energy system, electric vehicles, charging stations and heating systems, located within a confined geographical area with a well-defined physical boundary for the electric and thermal grids. The neighbourhood is not seen as a self-contained entity, but is connected to the surrounding mobility and energy infrastructure and will be optimised in relation to larger city and community structures. The initial description of a neighbourhood is focused on the energy aspects, but as a whole, the focus of all the pilot projects is on reducing a neighbourhoods greenhouse gas emissions towards zero within its lifecycle.

Assessment Criteria: Requirements to be fulfilled in order for a neighbourhood to be socio-economically sustainable.

Key Performance Indicator (KPI): A predetermined, quantifiable performance measurement that tracks the performance of an aspect of a zero-emission neighbourhood over time. Used to compare different projects and solutions to each other.

Sustainability: State of environmental, social and economic aspects of the global system, where current needs are met without compromising the ability of future generations to meet their needs.

Project phases: ZEN projects are assessed in several different phases:

- Planning phase, consisting of various regional and municipal planning phases: Outputs are plans and strategies for further work.
- Preparation phase: Outputs are project objectives, outcomes and the project budget as well as feasibility studies and information about the site of the project.
- Early design phase: Outputs are project outline and results of preliminary studies related to selection of technical, functional and physical structures.
- Detailed design phase: Outputs are main drawings, building details and tender.
- As built phase: Output is the finished building after construction.
- Operational phase: Consists of of use, maintenance, repair and replacement as well as energy, water and transport use by the neighbourhood.

Greenhouse Gas Emissions: Divided into direct and indirect emissions, it is expressed as kg of CO_2 equivalence calculated according to IPCC AR5 report [46]. Direct emission are emissions produced in operation, while indirect emissions are emissions produced through construction, transportation and disposal. For example, a car will have direct emissions from being used, but it will also have indirect emissions from the production, transportation, and refining of oil into fuel, and emissions associated with the production of the car itself.

Energy: Refers to potential work or work performed over a period of time. Defined as the integral of power over time. In terms of an electrical or heat system, it refers to the load on the energy grid over time. Energy is measured in kWh.

Power: Instantaneous rate at which work is performed. Power is the derivative of energy, and in terms of an electrical or heat system it refers to the load on the grid at a specific point in time. Power is measured in kW.

Economy: In the context of ZEN, it refers to economic sustainability. This is expressed as life cycle costs for buildings and infrastructure in the neighbourhood.

Mobility: Refers to transport patterns of the citizens to, from, and inside the neighbourhood.

Spatial qualities: Design of the public space in terms of amenities and services provided to citizens.

ZEN Goals

ZEN has outlined goals related to the creation and management of smart, sustainable neighbourhoods in order to reach its main objective [55]:

1. Become highly energy efficient and smartly powered by local renewable energy sources.
2. Manage energy flows in the built environment and with the surrounding energy system in a flexible way to facilitate the transition towards a decarbonised energy system.
3. Design and construct buildings, infrastructure, components and materials in the neighbourhood to consider zero or low GHG emissions in the whole life cycle; from extraction of raw materials, production, transport, installation, use, maintenance, repair, replacement, deconstruction, waste treatment, reuse, recovery, and final disposal.
4. Plan, design, and operate with respect to economic sustainability, and with a focus on minimising life cycle costs.
5. Achieve sustainable transport patterns through the overall design of the neighbourhood and implementation of sustainable smart mobility systems both locally and regionally.
6. Plan and localise amenities in the neighbourhood to produce good spatial qualities that consider reduction of emissions also through their relationship with the larger region and promote sustainable behaviour. Sustainable behaviour includes behaviour that results in reduced waste and resource use with a view to stimulate a circular economy.
7. Development of the area is characterised by innovative processes based on new forms of cooperation between the involved partners leading to innovative solutions.

The research centre determined KPIs to track how well projects address the different goals. The KPIs measure specific values related to energy, power, emissions, economy, mobility, spatial qualities, and innovation. The KPIs, along with their measurement units can be found in Table 2.1.

Key Performance Indicators

Planning Instruments for Smart Energy Communities (PI-SEC) is a Norwegian research project that aims at developing a toolkit for effectively planning both building project development (Bottom-up) and municipal development (Top-down) with regards to energy issues [53].

The research project collected and examined KPIs from existing literature and determined 21 indicators that are important for smart sustainable cities. The goals that the KPIs contribute to are categorised in five areas:

- CO_2 reduction
- Increased use of renewable energy
- Increased energy efficiency
- Increased use of local energy sources
- Green mobility

When examining KPIs there are a number of attributes that need to be considered when assessing how much value they provide for the smart city project.

- **Relevance:** A KPI should have importance to the evaluation of the smart city. This means that it should have a strong link to a concrete goal of the project. It should be defined so that the implementation of the smart city measures pertaining to the indicator show clear changes in the KPI
- **Availability:** Data for the KPIs must be obtainable with limited time and effort. Data that is expensive or time-consuming to gather are not suitable for building KPIs.
- **Measurability:** It should be easy to measure the KPI objectively.
- **Reliability:** The KPI should have a clear and concise definition that is not open to interpretation.
- **Familiarity:** KPI should be understandable by users.
- **Scalability:** A KPI should be relevant at different scales of the smart city: buildings, neighbourhoods, cities, districts and countries.
- **Phase applicability:** The KPI should be relevant across more than one phase of the smart city project (planning, design, construction, operation, end-of-life). If not measurable in more than one phase, it might be simulated or predicted in other phases.

The KPI attributes identified are qualitative in nature. To more easily compare different KPIs in regards to the attributes, PI-SEC quantifies the attributes based on a Likert scale [30]. A KPI is then rated on a scale from 1-5 with respect to each attribute. By using the attributes and the quantification of KPIs, the article creates a tool that determines how different factors and measurements in a project influence a KPI on the neighbourhood level.

The work on smart city KPIs and the quantifying these is essential to measuring the progress of a ZEN pilot area.

Data Management in ZEN

In order to measure the progress of the different ZEN projects, implementing a data management ICT system is suggested in [1]. The system should support reasoning and monitoring of ZEN KPI data and other relevant data sources from the pilot projects. These projects will produce a wide range of data than needs to be managed with appropriate transparency and quality in order for ZEN to reach its goals.

Data Types in ZEN

ZEN pilot projects deal with large amounts of data from a variety of sources. The data has in [1] been divided into three broad categories by their function and relevance to the projects:

1. **KPIs:**

   ZEN has a large variety of projects running, and common KPIs are important data that needs to be measured and monitored. A list of KPIs is available in Table 2.1.

   The KPIs are the most important data type to consider in the ZEN system, as they are the deciding indicators as to whether a project reached its goal or not. KPIs are usually not just data measured from one source, but research data collected from different sources within the project, combined and transformed, in order to measure the performance of a concrete aspect of the zero-emission neighbourhood. They are considered in multiple of the project phases, and will consist of different data types. They can often be some combination of data from:

   - Reports
   - Quantitative assessment
   - Streaming real-time measurement data
   - Aggregated and processed data based on the streaming data

   This variation in data types included in the KPIs means that the measurements will have different dimensions we need to consider. On the data side you have the spatial dimension and the temporal dimension. In terms of methodology for collecting data you either have quantitive or qualitative and the distinction of survey-based versus automatically calculated or tool-generated data.

   The KPIs used to measure the progress of ZEN are largely based on the KPIs identified for smart neighbourhoods in the PI-SEC project [53].

2. **Research data:** Data collected from the ZEN projects. Linked, transformed, and combined in order to determine the evolution of KPIs of the project. Typically measures some concrete aspect of a neighbourhood. Some examples are:

   - Smart meter data and other energy data from buildings.

- General building data such as room utilisation and heating data.
- Building occupant behaviour.

3. **Context data:** Data that helps interpret and gives context to research data and KPIs, such as:
   - Weather data
   - Mobility data
   - Social media data
   - Smart city datasets from outside the neighbourhood

Requirements in a ZEN Data Management System

There are many requirements that must be considered when creating a data management system for ZEN, a general set of requirements for a platform is identified in [1]:

- Hosting of KPI, research and context data and metadata
- Big data and visualisation services
- End user applications for municipalities, citizens and businesses
- Logs and data on the use of services provided
- Open API for third party developers
- Support for exploratory analysis of datasets
- Support for management of KPI data
- Support for research based on the data on the platform
- Future-proof for the duration of the ZEN pilot projects at a minimum 8 years.
- Flexibility in order to adapt to changes in KPIs and collected data

The listed requirements were considered by the report as key requirements for the technological platform for it to be able to help the ZEN project reach its zero emission goals.

Table 2.1: Key performance indicators for the ZEN pilot projects identified in [55], categories and unit of measurement for the indicators

Category	KPI	Unit
Energy	Energy efficiency in buildings: - Energy need	kWh/m^2/year
	Energy carriers: - Energy use - Energy generation - Delivered energy - Exported energy - Self consumption - Self generation	kWh/year kWh/year kWh/year kWh/year % %
Power	Power performance: - Yearly net load profile - Net load duration curve - Peak load - Peak export - Utilisation factor	kW kW kW kW %
	Power flexibility: - Daily net load profile	kW
Emissions	Total greenhouse gas emissions	tCO_{2eq} tCO_{2eq}/m^2/year
	Greenhouse gas emissions reduction	% reduction compared to base case
Economy	Life cycle cost	NOK NOK/m^2 heated floor area/year
Mobility	Mode of transport	% share
	Access to public transport	Meters
Spatial Qualities	Demographic needs	Qualitative
	Delivery of services, facilities and amenties	No. of amenities Meters (distance from buildings)
	Public space	Qualitative

2.3 Smart City Technologies

Smart cities consist of complex ICT systems utilising a variety of different enabling technologies to collect and manage data, monitor the city, and make decisions regarding future urban development. These technologies are evolving at a fast pace in order to keep up with their importance in different ICT contexts. Big data, IoT and cloud computing are identified as important technologies when realising a smart city [23][47]. These technologies need to interact in a comprehensive way in order for the city to be able make use of the data being generated to benefit citizen oriented applications, urban planning and decision making.

2.3.1 Big Data in Smart Cities

Big data is a category of data characterised by "The three V's:" Volume, Velocity, And Variety [28]. What these V's mean is that big data deals with a large amount (Volume) of data, being generated at a high frequency (Velocity), originates from different sources, is of varying size and arrives in different formats, often everything from text to sensor data to multimedia (Variety). The data produced by smart cities often fall under these characterisations, producing massive amounts of data from a variety of sources each day. Management of big data in the city is vital to gaining insight and knowledge into how citizens and the city interact and how the city can be improved and modernised.

Big data has a major role in facilitating smartness in many areas of the smart city [17]. **The smart grid** aims to make energy consumption and production smarter by using big data analysis. Real-time energy data is collected from automatic meter readings and other energy sensors in the city and processed to show data consumption and generation trends over time. This can help citizens become more aware of their energy habits, and cities manage and predict future energy use to accommodate energy need [44]. The healthcare sector can also benefit from the data they produce. Healthcare generates a lot of data both related to individual patients and to general trends in the population. **Smart healthcare** can use big data analysis techniques to predict epidemics, diseases and cures. On a city-level, healthcare professional can also analyse citizen health data to determine future healthcare needs and provide services to the population [42]. **Smart transport** and urban mobility is an important aspect of creating a modern smart city. Large amounts of sensors in the city can measure mobility-related factors like traffic congestion, public transport usage, parking availability, and air pollution. The city can provide services like alternate traffic routes [3], dynamic scaling of public transport services based on predicted need [18] and providing citizens with the information associated with road traffic and parking [52] using the data and analyses. **Smart government** and e-governance can also benefit from big data. Analysis of government-to-citizen, government-to-business, and government-to-government interactions can improve processes and cooperation between the government, private organisations and citizens [36].

In [17] a few examples where big data has been used in smart city projects are presented. The city of Stockholm gathered waste-collection vehicle data and and analysed it in order to identify inefficiencies in waste-collection routes. The analysis was used to propose a new plan for the routes that would reduce traffic congestion occurring from the inefficiencies. In Helsinki, over 1000 databases have been made publicly available as open data. The data that has been opened spans areas such as transport, employment, quality of life, and economics. The databases were made public in order to and make citizens more involved in public decision making. A smart city initiative in Copenhagen is using traffic and public transport big data to expand their network of cycling lanes based on where the data collection and analysis shows it will benefit the most.

2.3.2 Internet of Things

The Internet of Things (IoT) is a technological paradigm based around the idea of certain “things” or objects (vehicles, home appliances, sensors, smart energy meters, etc.) are embedded with technology allowing them to communicate, exchange data and interact with each other to reach common goals [14]. Smart solutions in a variety of contexts, including cities are technologically driven by the IoT network, the data collection, and their ability to impact the physical entities around the devices [5].

The different elements of IoT, and how they are applied in a smart city setting are discussed in [19]. One of the most fundamental parts of the IoT in enabling smart cities is the sensor technology. There are three main sensing paradigms IoT devices belong to.

The first is RFID, in which passive RFID tags in objects can be sensed using an external RFID reader in the same environment. One advantage of RIFD is that the devices themselves do not need a battery or another energy source to work, as they rely on the power from the RFID reader to be registered. RFID application in smart cities can be in an access control capacity.

Wireless Sensor Network (WSN) are often prevalent in smart cities. WSNs often consist of more advanced sensors that require energy sources. WSNs allow for collection, processing, analysis, and dissemination of data sensed in different environments [57] and it is crucial in collecting data about different aspects of smart cities.

The crowd sensing paradigm has emerged due to the prevalence of social networking. It encourages citizens to contribute in urban management by using their smart devices (watches, phones, tablets) as sensors when they interact with the city.

IoT networks in smart cities can be designed in different ways depending on the use case. Two common IoT network architectures described in [19] are:

1. Autonomous networks: Devices are connected to each other and connect to the public internet through one central entity. The ZEB Living Laboratory [15] is an example of an autonomous network. Sensors are used to measure a variety of factors such as energy usage, air quality, temperature, etc, and sent to a central monitoring

module. The control and monitoring module receives data and manages the sensors. The collected data from those sensors can then be sent to other entities from that monitoring module.

2. Ubiquitous networks: Devices are all accessible through the public internet. Authorised entities can access data and functionality of all devices directly without going through a gateway.

2.3.3 Cloud Computing

Cloud computing is a prominent paradigm in computing. It is continually expanding its reach in the IT world as big cloud service companies provide more and more services for its users; boasting advantages such as lower initial investments and on demand scalability [34].

Due to their positions in the market, monetary capital, and large accumulated computing resources, established large IT companies such as Google, Amazon and Microsoft have been able to commodify their infrastructure, computing power, and storage [6]. Organisations are increasingly discovering that their investments in IT services like computing power are under-utilised and maintenance is becoming the main spending area of organisations IT budget [34], and opt to "rent" these resources from external providers.

The National Institute of Standards and Technology (NIST) defines cloud computing as: *A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.* [37]

NIST also defines characteristics and service and deployment models for cloud computing:

Characteristics

1. **On-demand self-service:** Users can access, scale, and purchase resources (Storage, computing power, software services) on-demand without human interaction with the service provider.

2. **Broad network access:** Resources are available over network to heterogenous client devices like mobile phones, tablets, laptops and workstations.

3. **Resource pooling:** The service providers resources are pooled. They are available to multiple customers at the same time, where the resources are distributed to the customers dynamically according to demand.

4. **Rapid elasticity:** Resources have the ability to quickly scale outwards or inwards based on the demand of the consumer.

5. **Measured Service:** The services provided are metered, so resource usage can be monitored, controlled, and reported, leading to transparency both for provider and user.

Service Models

1. **Software as a Service (SaaS):** The consumer of the SaaS uses a software application provided on a cloud infrastructure. The application is often available on multiple kinds of client devices through a web browser or a program-interface. In SaaS, the user does not manage or control the infrastructure of the application, like network, servers, operating system or even storage.

2. **Platform as a Service (PaaS):** The consumer is provided with the service of deploying their applications on cloud infrastructure. The consumer does not have the ability to manage the underlying infrastructure in PaaS, but has control over the hosting environment for the application.

3. **Infrastructure as a Service (IaaS):** The consumer is provided with the control over some components of the infrastructure. This includes control over provided processing power, storage, networks, operating systems and applications. This is provided in order for the consumer to deploy and run any arbitrary software on the cloud environment. The underlying infrastructure that provides the IaaS capabilities is still not managed or controlled by the consumer, but they may exercise some control over some networking components like host firewalls.

Deployment Models

1. **Private cloud:** The cloud infrastructure is provided for a single organisation. It may be owned and managed by the organisation itself or by a third party.

2. **Community cloud:** The cloud infrastructure is provided for a community of organisations that have the same concerns or goals.

3. **Public cloud:** The cloud infrastructure is provided for use by the public.

4. **Hybrid cloud:** Some combination of clouds using the other deployment models. The clouds remain separate entities, but are connected via some standardisation or proprietary technology enabling cooperation and data flow between them.

Cloud Computing in Smart Cities

Cities, governments and municipalities looking to modernise are using ICT technologies to improve their infrastructure and cater to the needs of future cities their citizens in terms of urbanisation, economic growth and environmental sustainability [39]. The smart city is one response to this evolution [9], and in many cases, cloud computing is a component of these solutions. Looking at different city governments [20][29][45][9] and smart city applications, architectures and frameworks [24][21][22][48], many local and national governments use cloud services in their modernisation in order to create more efficient and cost effective solutions [34].

2.3.4 Open Data

Open data is a part of many smart city projects [48][41]. Publicly available datasets collected, managed and distributed by the city can help citizens, organisations, and businesses reason about the city and make smarter choices that benefit the city. This can be citizens choosing different modes of transport based on predicted congestion or service and retail businesses choosing to change locations based on closeness to public transport hubs or volume of pedestrians in an area. Cities with open datasets can also give access to services and provide APIs so that external developers can use the data provided to create new smart city applications [22].

Many cities provide open data portals where data collected by the city about aspects such as energy, traffic, demographics, tourism, etc. are available. Some examples are: Amsterdam[1], Trentino[2], and Helsinki[3]. The open data portals are used to disseminate the information the cities collect to the public.

In [18], the authors point out some challenges that a city must be aware of when disseminating open datasets. It is important to note that data is not fact. Providing open data about the city to the public will be interpreted as the city acknowledging that it believes the data is correct. Data varies in quality, and raw data is vulnerable to mistakes like faulty sensors and missing measurements. In order for there to be value in opening data, the quality of the data being disseminated must be sufficiently high, so that citizens, businesses and other city stakeholders can be confidant that the conclusions they draw from that data are correct.

Anonymisation of data is a factor cities need to take into account when providing open data. Robust privacy practices when collecting and publicly disseminating data is vital to maintaining citizen trust in the smart city. Anonymisation of open data can be done by removing identifiers and aggregating the data before publishing to a data portal. When anonymising, the value of that data needs to be kept intact during the process. Too much aggregation may remove valuable information. This potential loss of information needs to be balanced with privacy concerns. Keeping data non-identifiable while still maintaining the value of the data is a crucial part of creating an open data environment in a smart city.

[1] Amsterdam - `https://data.amsterdam.nl/`
[2] Trentino - `https://dati.trentino.it/`
[3] Helsinki - `https://hri.fi/`

2.3.5 Commercial Smart City Solutions

Governments are investing heavily in the smart cities [32]. With these investments, a market for commercial smart city development has formed. Technology companies are taking advantage of cities and governments trying to adapt to this paradigm by providing their own smart city technologies and platforms. There are different scales of the platforms being provided and there are many actors in the market. Solutions range from dedicated IoT management services, to full scale infrastructure, storage ,and analytics platforms complete with consultancy and developer teams.

Many big software and cloud solution companies provide the full spectrum of services for smart cities. Some of the big cloud providers like Amazon and Microsoft have services dedicated to developing smart cities[4],[5]. In some cases, like with Amazon, this is just marketing their existing cloud solutions to a smart city market, providing use cases and examples for how their products and services have been implemented in smart city applications. Other companies like Microsoft and IBM[6] provide their own smart city applications and solutions.

There are both advantages and disadvantages to going with large commercial actors when creating smart city applications. Using large cloud providers for smart city solutions has benefits in both economy and security [34]. Using a commercial cloud provider can drastically lower costs of initial investment, maintenance, and security [4], which is a quite attractive given how large the monetary investments in smart cities are [32]. On top of the cost of security, actually maintaining robust security in a smart city is difficult [13]. Smart cities also become more susceptible to cyber-attacks [23], and are popular targets [25]. Having a major company like Microsoft or Amazon providing security with their services can be a benefit, providing expertise and resources that city and municipal governments do not necessarily have. There are not only advantages to using commercial providers. The main issue with sticking to one smart city provider for all technological aspects of a smart city is vendor lock-in. Cities can become too dependent on one company [25]. This is little risk of this becoming an issue when different technologies and providers are used. However, using companies that create full smart city solutions both for technology and development of platforms and applications can make switching companies and solutions costly. This may make cities hesitant to change providers even if a different solution might be better. To reduce the risk for vendor lock-in in cities, they should manage their use of proprietary and open software as well as internal and external development to get the most benefit out of these external service providers and technological components while still maintaining the ability to provide the best solutions for the city and its citizens.

[4] AWS Smart City - `https://aws.amazon.com/smart-cities/`

[5] Microsoft CityNext - `https://enterprise.microsoft.com/en-us/industries/citynext/`

[6] IBM Smarter Cities - `https://www.ibm.com/smarterplanet/us/en/smarter_cities/overview/`

2.4 Smart City ICT Architectures

Creating a comprehensive ICT architecture for smart city application requires methods and technologies for data collection, data storage, and data processing. Given the variety of types of sensors collecting data, there is need for technologies and solutions for managing sensor and IoT communication with the rest of the architecture, including capturing data streams from the sensors. Storage typically involves both relational and non-relational databases as well as object storage for files not suitable for database table structures. Due to the size of the data collected in smart cities there may be issues when it comes to capacity for storage and processing of the data. Cloud providers like Amazon[7], Google[8] and Microsoft[9] can provide cities with storage and processing capacity without the city needing to invest in its own data warehousing facilities. Computing power for complex data analysis through big data processing tools like Spark can also be achieved through cloud service providers, reducing the barrier to entry for cities developing smarter ICT solutions. Finally, the data is used to achieve some goal. This can be disseminating the acquired data in an open data portal for citizens, providing data exploration tools for city decision makers or serve a specific smart city application like automatically managing traffic lights or a traffic application with dynamically changing routes information and parking availability.

The fundamental concepts of data collection, storage and processing/analytics are present in most smart city ICT architectural solutions. However, different architectures use different technologies, and have different approaches to organising the components and their interactions.

The Lambda architecture [35] is designed as a data-managing backend architecture for general big data applications. High throughput and minimisation of network maintenance costs are the primary focus. Not originally designed for smart cities, the architecture is applied to that context in [24]. Figure 2.2 shows the architecture implemented on Amazon Web Services. The architecture is divided into three main components. One that manages batch processing, one that manages real time stream processing and a component that responds to user queries. The focus of the architecture is on big data processing and [24] effectively implements the solution in the cloud, arguing that a cloud implementation of the architecture is suited for smart city solutions due to its access to large computing and storage resources and its price efficiency. Cloud solutions are present in many smart city ICT architectures. In [21], a layered cloud-based architecture is proposed with the purpose of serving multiple context-aware applications. The data management in the architecture starts wide and generic and becomes more context-specific as it passes through the layers. Compared to Lambda, this architecture proposes solutions for a bigger part of the ICT system. The Lambda implementation in [24] has its focus on the details of the data processing part of smart city ICT systems while the architecture in [21] takes integration of IoT sensors, accessibility of data, storage, security, and data management into

[7] Amazon Web Services - `https://aws.amazon.com/`
[8] Google Cloud - `https://cloud.google.com/`
[9] Microsoft Azure - `https://azure.microsoft.com/`

account as well. The different layers in figure 2.3 provide different levels of services to the smart city ICT platform, with the data progressing from generic layers to specific layers. Layer 1 integrates sensors and ensures data accessibility, layer 2 allows the ICT system to access remote data sources in the cloud environment. Layer 3 applies context to the data, categorising and harmonising the data that is intended for a specific application or category of applications. Layer 4 provides tools to allow for specifying workflows and linking different data sources and processing components, performing necessary processing of data. The idea behind this architecture is for it to be a PaaS where developers can link and process their data sources as needed and deploy their application on top of it.

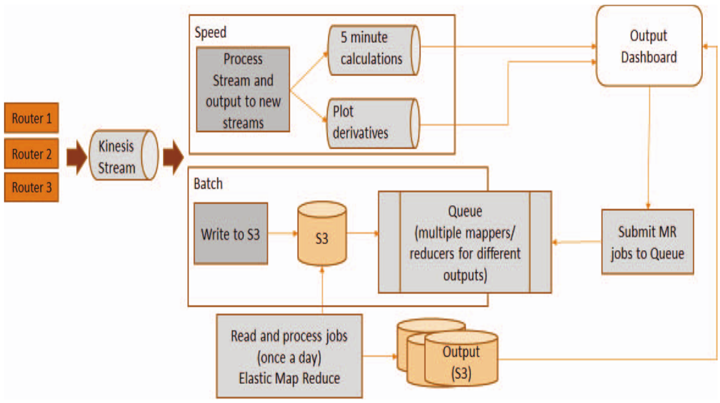


Figure 2.2: AWS implementation of Lambda Architecture [24]

In addition to the prominence of cloud computing, open data platforms are common in smart city architectures, especially as a basis for the data used in the systems. While the architecture in [21] is general and aims at providing a platform for other smart city applications. Architectures that are more application specific[48] [22] can use open data provided by smart city platforms to solve concrete smart city challenges. A smart tourism application for Smart Connected Communities in Trentino, Italy and its associated ICT architecture is presented in [48]. The application uses data from Open Trentino and a crowd sensing "tourist bracelet" developed by the researchers that interacts with users smartphone and tourist attractions in the city. CKAN[10] is used for the open data platform in Trentino. The architecture and application demonstrate how to take data both from a dedicated open data portal and dynamic crowd sensing data and link it, in order to provide a data management system and a smart city service where information about the city and tourist recommendations are disseminated through a mobile application. A similar project [22] uses open data from UKOpen to provide the basis for its smart city application. The main goal is to process data from the open data portal and show correspondence between health, mortality, air quality, quality of life, house prices, household income and crime and

[10]CKAN - `https://ckan.org/`

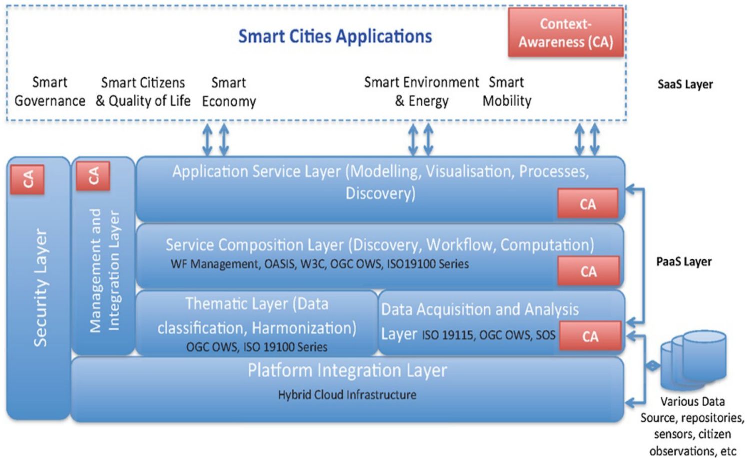


Figure 2.3: Conceptual framework for a cloud-based big data architecture [21]

rank different areas of the UK city of Bristol according to a number of factors related to liveability. The architecture is divided into three layers which are shown in figure 2.4. The data acquisition, analysis and filtering layer for storage and data cleansing, the resource data mapping and linking layer, in charge of finding new workflows and relationships when linking datasets, and the interactive explorer, the high level and application specific analytics layer.

Managing all the data going through a smart city environment is challenging. Different actors and organisations in the ecosystem will have different data structures, and different policies for data sharing. Data may have different granularities and different privacy needs, both for sensor data and data-repositories. The data sources may initially not be compatible enough with each other to produce any valuable information, knowledge or insight for the city. In [22], UK Open Data[11] from the city of Bristol is used. UK open data collects data from a number of sources, and there is a chance that the datasets published contain incomplete missing data.

Data is often aggregated on different time and geographical scales based on the original intention for the data, so although statistical relationships can be found, due to the different granularity it may be difficult to make definitive conclusions. For example, the Bristol geographical open data [22] is aggregated in a ward scale (administrative divisions). So although the data is aggregated on a geographically confined scale, for the purpose of the use case: "Checking quality-of-life factors in an area when considering buying a house." the granularity of the data available in the open data portal is too rough to determine quality-of-life factors for a specific street or neighbourhood.

The architectures discussed in this section are relatively similar in structure in spite

[11] DATA.GOV.UK (UK Open Data) - `https://data.gov.uk/`

of them fulfilling different purposes in the smart city ecosystem. They collect data from a variety of sources, consolidate, link, process, and analyse them. Architectures like [21] are concerned providing a platform other developers can utilise to create applications, while [22][48][24] build architectures for domain-specific applications. The fundamental components are the same, but the way the components are organised is different in an application performing a specific purpose versus a platform that manages data for many different application with different needs. The agreement on the fundamentals can be used as a starting point for other smart city initiatives building architectures for their ICT systems.

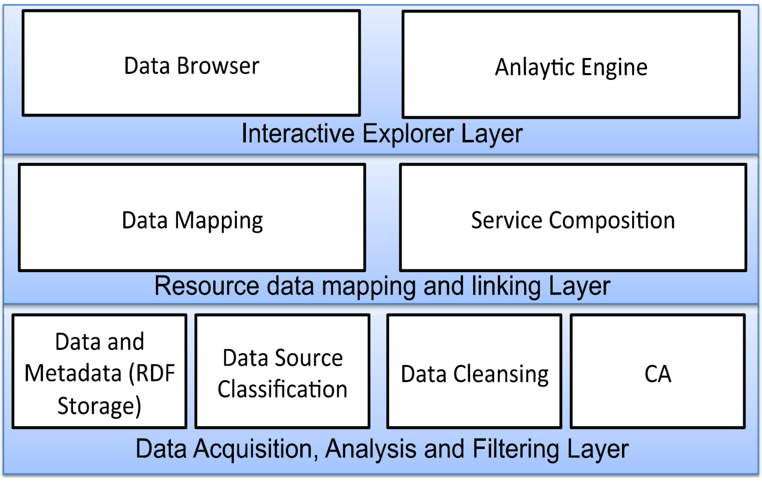


Figure 2.4: Architecture design for an application using UK Open Data [22]

Table 2.2: Different smart city ICT architectures identified in background research, along with goals and technologies used in the systems created with them.

Architecture	Description	Technologies	Area of Use
Lambda [24]	Architecture for batch and streaming data. Focus on high throughput and low pressure on network.	Infrastructure: Amazon EC2 Storage: Amazon S3 Databases: Amazon Relational Database, Amazon Dynamo DB for NoSQL Analytics: Batch jobs - Hadoop with MapReduce. Streaming - Apache Spark, Apache Storm	General architecture. Multiple kinds of analysis for data collected by sensors.
TreSight [48]	Architecture for combination of open data and real-time GPS CrowdSensing data.	Storage: MongoDB on HFS cluster Analytics (Event Processing): Fiware - PROTON Analytics (Social media): COSMOS	Smart Tourism. System runs under mobile application. Provides recommendations to tourists based on insights gathered from data in system.
Cloud Based Analytics [22]	Management of Big Open Data with focus on implementation of architecture as a potential cloud analytics platform.	Storage: Cassandra (NoSQL) PostgreSQL (relational) Virtuoso (RDF) Analytics: Rapidminer + R	General architecture with use case: Open Government data in Bristol to determine factors leading to quality of life improvement.
Cloud-based context-aware framework [21]	7 layer PaaS architecture proposed to handle platform integrations, classification, workflow specification, visualisation and security in smart city applications.	Not specified in article.	Build a comprehensive architecture to facilitate smart city application development

Chapter 3

Research Methodology

This chapter describes how the research in this project was conducted, which research questions were chosen, and how the results of the research is evaluated.

3.1 Research Method

This project is part of a large body of research into creating data management architectures and ICT systems for smart cities. In preparation for the thesis, an ad hoc (state-of-the-art) literature review was conducted as a specialisation project in autumn 2017. The literature that was reviewed consisted of architectures and implementations of smart city platforms and applications used in urban mobility. The end product of the specialisation project was a design sketch for an architecture for an urban mobility project [51].

The original domain in the specialisation project was switched out for a zero-emission neighbourhood use case for the thesis. The change in use case did not invalidate the state-of-the-art literature review and the architecture sketch, as most of it was intended to be applicable for a variety of smart city domains. For this thesis, background research was done to order to understand the smart city, zero-emission building, and neighbourhood domains and the technologies and architectures that facilitate them. The architecture sketch was redesigned and improved in order to fit the new domain. The architecture is then implemented on the use case, before it is evaluated in relation to requirements identified for general smart city ICT systems and requirements specifically identified for a zero-emission neighbourhood data management system.

The research method used is similar the *design science research* method. Design science research is meant to complement the traditional research paradigms (positivist, interpretative, and critical perspective). It aims to improve and understand the behaviour of information systems through design, creation, and analysis of artefacts (Algorithms, human computer interfaces, system architectures, ICT systems, design languages, etc.) within the research field of information systems [27]. Design science research uses a 5-step process proposed in [49] which consists of:

1. Awareness of problem - Awareness of an interesting or significant problem or opportunity in an industry or research discipline.
2. Suggestion - A solution to the problem from step 1 in the form of an artefact is designed and suggested.
3. Development - Implementation of the artefact suggested in step 2.
4. Evaluation - Artefact that has been designed and implemented is evaluated in terms of pre-defined criteria.
5. Conclusion - End or research cycle. Results of research are deemed "good enough" and the knowledge gained from the research is documented.

The research done in this project follows this process. *Awareness of problem* being the specialisation project and the background research in chapter 2. The *suggestion* step is done in chapter 4, where the conceptual ICT architecture for smart cities is designed and proposed. *Development* is done in chapter 5, where the proposed architecture is implemented as a zero-emission neighbourhood data management ICT system. *Evaluation* of the prototype and architecture is done according the the requirements presented in section 5.1.1. If the research is found not to be significant enough, the design research method dictates that the research would not be complete, and the entire process is repeated until satisfactory results are achieved, iterating on the previous solution. The work done in this thesis can be seen as already iterating upon the work done in the specialisation project. Eventually, the *conclusion* step is reached, where the knowledge gained from the research is presented, and potential future work is suggested.

3.2 Research Questions

RQ1: What are the key components of a data management architecture for smart cities and zero-emission neighbourhoods?

RQ2: How well does the architecture translate into implementation in the domain?

RQ2.1: Which technologies and tools are suited for implementing the architecture?

RQ3: Does the implementation contribute to further development of a data management system for ZEN?

3.3 Evaluation Criteria

The criteria which the success of the artefact will be measured against is whether it meets the requirements identified based on the previous research. The artefact will be evaluated against requirements identified for general smart city ICT systems identified through the background work of this project, and specific requirements for the zero-emission neighbourhood application identified by the ZEN research centre.

Chapter 4

Proposed Smart City Architecture

This chapter proposes an architecture for smart city ICT systems. The architecture is designed based on previous approaches to smart city platform and application architectures discussed in section 2.4. The proposed architecture is designed with the goal of fulfilling general requirements for ICT systems and applications in all domains of the smart city, with a focus on zero-emission smart neighbourhoods. The architecture is created to be as general as possible, so it can be applied to more scenarios while still keeping in mind the zero-emission neighbourhood domain. Due to the general nature of the model, most technological choices are not defined in this chapter, as this varies based on what the architecture is being used for. A short description of horizontal components along with possible technologies is presented in table 4.1.

4.1 Architecture

4.1.1 Stakeholders

1. **Citizens:** Users of services provided by the systems built from the architecture.
2. **City Government:** Whether the architecture is implemented in a single city or across multiple cities, city or municipal governments will either directly or indirectly through a inter-municipal department manage the smart city system. ZEN has multiple pilot projects in different cities and municipalities, and with a ICT system spanning all these projects, multiple municipal and city governments will be stakeholders for the same system.
3. **Planners and decision makers:** Use the data and functionality provided by the

architecture to reason about the domain and the overall city, plan and make decisions based on the information.

4. **National government:** Oversee regulation in regard to the smart city. Privacy and security regulation as well as data standards and policies across all smart city projects.

5. **Application developers:** Use data and information gathered from the smart city ICT system to develop applications.

6. **Data providers:** Provide some of their data for the system, and may benefit from the processing, linking, and analysing. They want a system that benefits them as much as possible, so they get the most value in return for the data they put in.

4.1.2 Requirements

- **Flexible processing and analytics:** With the wide variety of use cases within smart cities, the architecture to be able to process and analyse incoming data effectively using a variety of different tools and technologies to gather and present the information to the users of the system. Different stakeholders will benefit from different data and different analyses.

- **Interoperable storage for heterogenous data:** The architecture should be able to handle different types of structured and unstructured data originating from both local and external data repositories and streams. The data should be stored in such a way that it does not limit its use, and so it can be linked and used with other data.

- **Cohesive data management:** Quality and granularity of data is important to get reliable results. This is discussed in [22] where lack of specific data granularity leads loss in value in the results. The architecture should collect, store, process and disseminate data in such a way that the quality of the data is kept intact in every component. This means that no component should alter the data in such a way that it makes it more difficult for other components to use, or that the data loses value for somewhere else in the system.

- **General approach:** The architecture should be suitable for smart city applications spanning different areas of the smart city ecosystem.

- **Facilitation of open data dissemination:** Data and information collected and generated by the system should be publicly available. The data should also be available in such a manner that it is easy to access, read, and explore for anyone.

- **Vendor independent:** In order to avoid vendor lock-in, discussed in [25], the architecture should not rely heavily on one provider or vendor. In an architectural sense, this refers to creating a architecture that can be implemented anywhere, without there being a specific requirement for a single company, organisation, or technology to be included in the implementation.

These are requirements identified as the most important for a general smart city ICT system architecture The identification of requirements is based on the smart city architecture background research from chapter 2. Technology and domain specific requirements are discussed in relation to implementation of the architecture, as the nature of each implementation may vary.

4.1.3 Architecture Overview and Description

The architecture overview in figure 4.1 shows how all the components are in the architecture are connected and how data flows between layers and components. The two layers are the data layer and the application layer, each with associated components. There are also two vertical components that span both layers: The data management and policy component and the privacy and security component.

The data layer consists of the collection component, the access component and the processing and analytics component. The data collection component is specifically for data collected in the domain of the smart city ICT system, typically sensor data from IoT devices. The data access component is divided into two separate logical subcomponent. The data that is stored, collected and generated by the developer of the smart city, and the data that belongs to private businesses, other government agencies and other collaborating entities. The analytics component transforms, processes, and visualises the data for further use in the application layer as well as going back and storing the result in the access component.

The application layer consists of both the open and restricted data interfaces as well as the smart city applications that can be built on top of the system. The data interfaces are where developers, citizens, and decision makers access the data collected, stored, and processed in the system. There are two data interfaces; the restricted interface, intended for people who need access to specific tools and features such as city planners and decision makers, and the open data interface which is open to anyone, providing data for citizens and other external parties interested in viewing and using the data produced by the city.

In addition to these two main layers, there are two components that are present both in the data and application layer, and due to their relevance to all of the other components, they are represented as vertical components rather than horizontal like the rest. The privacy and security component is responsible for access rights and other threats to data security within each component as well as privacy and anonymisation of data. The data management and policy component defines and makes sure all data is stored and collected according to policy, and to ensure interoperability. The data management and the security component are not necessarily technical components with different modules like the horizontal components. They represent the policy and planning aspects of maintaining data interoperability, data quality, security, and privacy throughout the system.

The access component and the analytics components can be managed and run locally by the city or optionally in the cloud. Chapter 2 details the advantages of using a cloud provider for storage and processing, and while there are some instances where the cloud is not an optimal solution, for example in case of sensitive data that should not be stored outside a country's border, using a cloud provider for hosting storage and processing solutions can reduce costs and energy consumption of storage and processing data, so they should be considered when implementing smart city ICT systems.

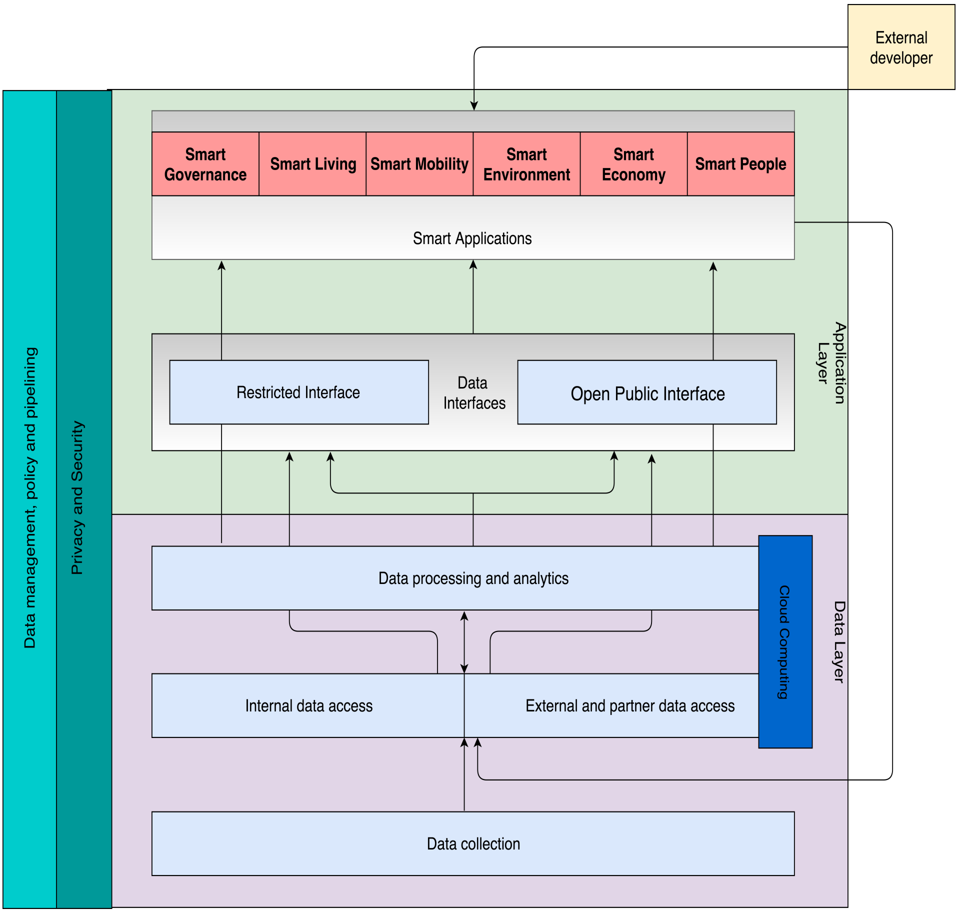


Figure 4.1: Overview of the proposed architecture, its components, and the data flow in the architecture.

4.2 Components

This section describes each component of the proposed architecture. The architecture is partitioned into the data layer, consisting of the data collection, data access, and data processing and analytics components, and the application layer, consisting of the data interfaces and the smart city application components. In addition to the two horizontal layers, the architecture includes the data management and policy as well as the security and privacy component, that need to be considered for the whole system. Each component is described in a general architectural sense, with some focus on the zero-emission neighbourhood domain. While the architecture is designed as a general approach to smart smart city ICT systems, it may be useful to keep in mind the domain in order to look at what each component would contribute with in a specific situation in a data management solution. The components in the data layer also contain a graphical view in order to visualise how the component works on a conceptual level. The graphical views are as general as possible, omitting any technological choices.

4.2.1 Data Layer

Data Collection

The data collection component is shown in figure 4.2. As described in section 2.3.2, IoT networks can be organised in different ways. Two common ways are ubiquitous networks and autonomous networks. Ubiquitous networks connect all individual devices to the public internet and make them accessible, and autonomous networks use a central management module to connect the devices to the wider public network. In this architecture, an autonomous approach is chosen, where sensors are connected to a central management module before being sent to a data stream manager that can take multiple data streams from different sensor networks and send them to the storage component.

In a zero emission neighbourhood, you will typically have sensors collecting a wide variety of data from many different sources. Smart electricity meters, thermometers, parking sensors, and motion sensors are a few examples. This means that edge nodes in a distributed IoT network need to sort what type of sensor data is coming into the network at any given time, and relate it to the geographical nature of that data. The chosen alternative is to keep each network of IoT devices that perform one specific task (e.g monitor energy use in a specific area) separated from each other, with their own management node connecting them to the public network. This makes the data from each sensor network more easily identifiable amongst the other data streams. As long as the data formats and standards for the sensors are interoperable with little need for individual preprocessing, this method will create usable and identifiable subsets of task-specific, geographical data that can be linked together further down the line in the access component and during processing and analytics.

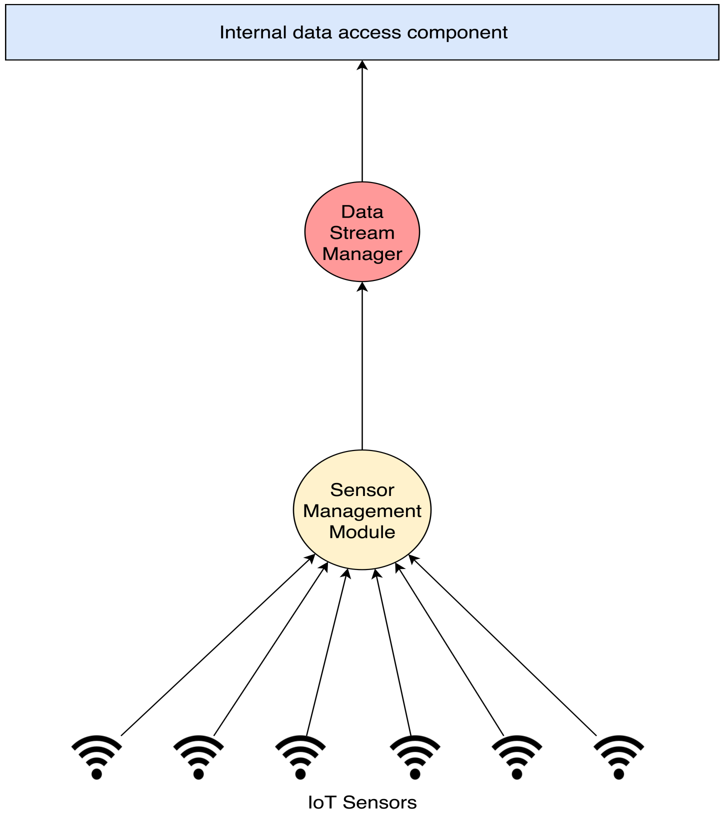


Figure 4.2: Data collection component of the proposed architecture. Sensor network is managed by a central sensor management module that also collects the data. The collected data from the sensor network is sent to a data stream manager before being sent to the internal access component.

Data Access

Data access is separated into two logical subcomponents shown in figure 4.3. The internal access component is responsible for storage and access to data owned by the city, government or whichever entity is responsible for developing the smart city ICT system. The external access component is responsible for the rest of the data relevant to the smart city, such as data from collaborating organisations and businesses. The data from external organisations and stakeholders is represented as one entity in figure 4.3, but that is just for logical separation of the concepts of internal and external data. Data from collaborating entities will most likely be spread across multiple organisations with different physical locations and access policies in an implementation scenario. This means that the city and the partners are required to work to effectively standardise data storage formats and policies in order to effectively work with, integrate, and link their data. Having clear data policies will help the city keep data linked.

Data in the internal access component in a ZEN context may consist of sensor data from the data collection component, historic repositories of this data and reports and estimations from the planning and construction phases of the neighbourhood. Sensor data enters into the access component without context from other sources. The data that is collected can either be stored in one repository, or be separated out to repositories for each different sensor network, allowing the data to be linked at a later stage for contextualisation.

The internal access part of the component can also contain the results from the processing and analytics component. The information gathered, patterns recognised and visualisations can be stored for further use. If storage is limited in the system, this is not necessary, but having access to historic analytics can be helpful in future decision making and planning, by looking at what has worked, and not worked in the past. Access to historic data is important in a zero-emission neighbourhood system where solutions and experiments often are evaluated by the evolution of the KPIs over time. Metrics for successful solutions may be reducing load on the electrical grid over time, or an increase of citizens using public transport. Historic data may be stored in repositories as raw data, but having the already processed and analysed data ensures easy and quicker access in the future.

The external access subcomponent contains data not owned by the organisation itself. In a zero-emission neighbourhood project, the organisation will typically own data from the planning process and sensor data from the monitoring of that project. In order to gain insight into the entire neighbourhood data ecosystem, the system may need additional data to contextualise what it is collecting internally. This can be cellphone data, social media data, parking data, public transport data, and other data about citizen interaction with the neighbourhood, city, and infrastructure around them. Data collected and stored by many different entities and from different sources is bound to have different standards and a varying level of quality. On top of this, data is collected, stored, and disseminated at different levels of detail and granularity based on the needs of the entity collecting it. When implementing the architecture, it is be important to keep in mind these differences in order for the internal data and external data to be interoperable, which is further considered by the data management and policy component.

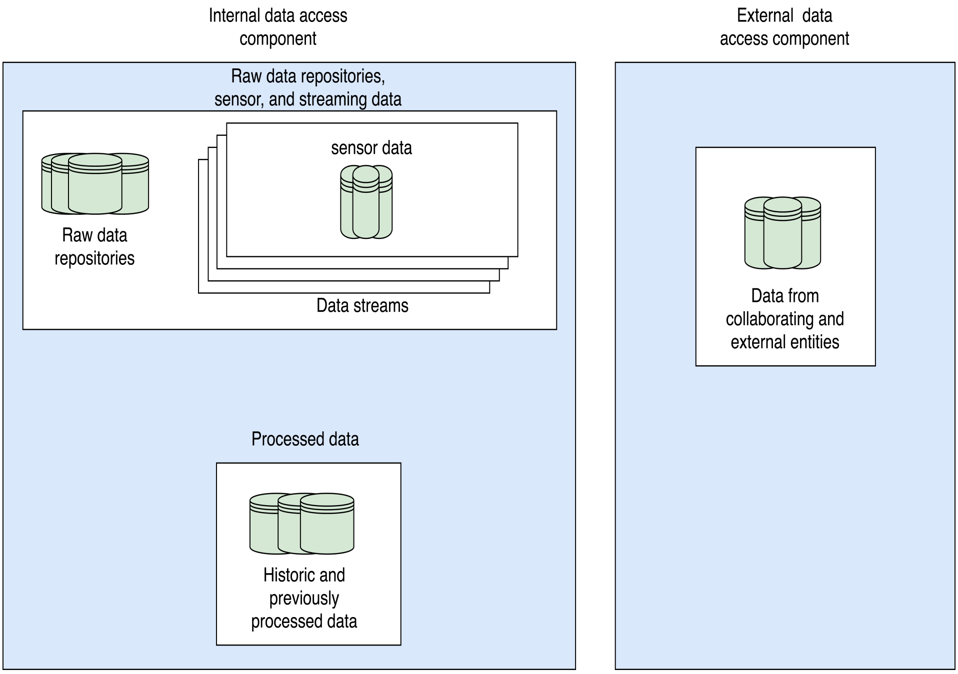


Figure 4.3: Data access component for proposed architecture. Separated into the internal and external data access subcomponents. Internal data is the data collected by the organisation or entity developing the ICT system. External data is the data collected by external and collaborative organisations and business that are used by the system.

Data Analytics

The design of the analytics component is heavily dependent on the use case for the system. There are different factors that have to be taken into account when implementing and designing this part of the architecture. The component can either process only batch or streaming data coming from the data access component, or more likely, a combination of both. It can also get data from external sources not in the data access component, such as data streams from social media analysis frameworks like COSMOS [8]. The analytics component may not only be one large entity, but several modules in the system operating in different parts of the system. Data cleaning and aggregation may be done as preprocessing before storage of the data. More contextual and domain-specific processing and analysis may be done later, with different technologies, where the data looks substantially different than in the preprocessing stage.

Different usage of data requires different analytics tools and changes in the component. It is therefore important that the architecture is flexible enough to accommodate changes rapidly. The model of the analytics component in figure 4.4 is illustrated very generally to indicate the flexibility of the component in the architecture. In addition to processing and aggregation, the system may perform deeper analysis on the processed data. Depending on what the data is used for this can be visualisation, machine learning, data mining or event detection. The result of processing and analysis is either accessed from one of the data interfaces, or used directly in a smart city application. It makes sense to have an API that can access the component, especially for internal development, so developers can automate and run their own analysis and processing jobs on the system. Having direct access to the analytics component is essential when developing data heavy applications for the city.

The type of analysis and the results they provide to the data interfaces should be pre-defined and kept updated regularly. In a zero-emission neighbourhood context, visualisations of energy demand, energy generation, public transport demand, and parking occupancy maps are examples of data that can be generated by the analytics component and presented to citizens and decision makers in the data interfaces. The results of the data sent to the interfaces should also be transferred back to the data access component and stored for potential future use.

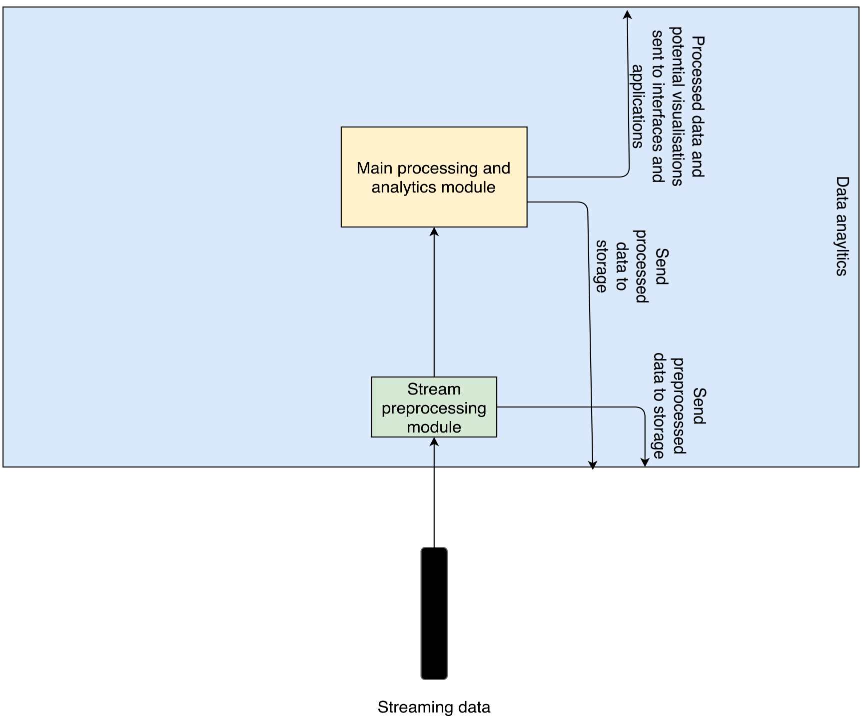


Figure 4.4: Analytics component for proposed architecture. Data arrives into the component, and processing is done in various steps, based on how much processing is required. After processing, data is sent to interfaces and applications for use and dissemination, and back to data access component for storage.

4.2.2 Application Layer

The application layer in the architecture consists of the *data interfaces* component and the application component. The application component is out of scope for this architecture, as it will be significantly different depending on the domain and use case of the application. The smart city system will typically also have a number of different applications that utilise the data collected, stored and processed in the other components. As a result, only the data interfaces are described as a part of the architecture in this section.

Data Interfaces

The data interface component is what connects the system to many of its users. Through the different interfaces, citizens and decision makers in the city will have access to much of the data and information that is collected and processed. The interfaces manage the data that is stored and processed in the data layer and makes it available to the relevant stakeholders. Different stakeholders require different tools and data, and different interfaces need to be created and maintained in order to meet these needs.

The idea for the design of this component is to split it into two different interfaces: The Restricted Data Interface (RDI) and the Open Data Interface (ODI). The RDI is meant for planners and decision makers and will require authorisation to be accessed. The distinction between the two interfaces is twofold. The tools required by decision makers in the city are not the same as tools required by the average citizen. Decision makers need tools for drilling and querying the data. The data shown to decision makers can not always be shown to everyone else. The RDI will consist of data of a certain granularity or with certain attributes that are not suitable or not legal to have open on a public interface, or data that there is low perceived value to publishing to an open interface.

One option for allowing for management of different data on the interfaces while allowing for separation of data is to have one service providing access to all the data, where an access control mechanisms are enforced, but everything is contained on one service (for example an open website with a log in service for the specific RDI tools and data). The disadvantage to this approach is that a universal service or platform may not be optimal for providing different stakeholders and users with the different tools and features they require. With one service, it may not be possible to provide the right tools to both decision makers and citizens with the technologies at disposal for that service. Another approach is to provide the interfaces through different services, requiring authorisation only on the service providing the RDI, and leaving the ODI service completely open without restricting any access. The RDI can implement more levels of access control as desired based on privileges of the user. Separating the interfaces into different services will make it easier to manage privacy and security of data. No data that is supposed to be open will not be in the ODI, removing the need to control who has access to what anywhere on the service. The RDI will need access control in the form of authentication to access the interface at all, as well as authorisation for accessing data with different levels of restriction. Another advantage of this approach is that maintenance of data and further development of tools in the interfaces are clearly separated from each other, avoiding confusion between what is meant for citizens and what is meant for decision makers.

The ODI could be implemented as a open data portal similar to UK Open Data where data sets are available for download and can be retrieved using a RESTful API. The ODI is intended for academia, businesses and other entities not affiliated with the smart city program to be able use the data collected by the city in research and development. The data also helps contextualise information, which citizens can use to make more informed and sustainable decision about their interactions with the city.

The RDI differs from the ODI in intention and use. The users of the RDI in a smart city context are mainly city planners and municipal decision makers. These types of users need to be able to drill down, aggregate, contextualise and query the smart city data in much more detail than the users of the ODI. A simple open web portal or an API is not necessarily suitable for non-expert decision makers to perform the needed operations and analysis on the data. An alternative can be using some type of dashboard application similar to modern Business Intelligence dashboards. Visualisations performed in the analytics components can be published to this dashboard with tools to explore and drill in the the visualisations and query underlying analysis.

4.2.3 Privacy and Security

Privacy and security in the smart city consists of implementing sensible security measures for all aspects of the system. The data management system should be implemented such that it complies with the relevant privacy legislation while still providing the city with timely and useful information. Smart cities can be targets for cyber attacks, and must implement security solutions and services to protect the smart city and make sure data does not get stolen. Measures for protecting information in smart cities can include: Encrypting communication and data transferred between components, activity logging, and access control mechanisms in all aspects of the system.

4.2.4 Data Policy and Data Management

The data management component is responsible for how data is handled in the rest of the components in the architecture. In the overview in figure 4.1 it is shown as a vertical component, as data management needs to be taken into account in all parts of the system. The component manages the interoperability between different data providers and different data types. Different organisations and businesses store their data on different formats and different granularities which needs to be kept in mind when creating the system, so that data from each provider can be linked and worked with together. Interoperability also needs to be managed for the different datasets. Data stored in a variety of formats may need to be linked, and the data management component should make sure that the datasets work together. In addition to ensuring interoperability between dataset, the data policy and management component should have clear policies on maintaining high data quality throughout the collection, storage, and processing components. This include making sure that potential gaps in data are explained to users and that raw data used in processing and analysis is kept stored in the system, so the basis for analyses are available.

Table 4.1: Table showing components of the proposed architectures with description, goals, and potential technologies.

Component	Description	Goal/Objective	Technologies
Data Collection	Collect data from sensor networks.	Send data from city to storage and to further processing.	WSN, RFID, ad hoc networks
Data Access	Storage of data from data collection component and other internal repositories and data streams. Manage access to external partner data.	Store and manage access to different data throughout the system. Both raw and previously processed/analysed data.	NoSQL: Column, Document, Key value, Graph Object datastore SQL Cloud data storage: Amazon, Google cloud, Microsoft Azure
Data Analytics	Modules for processing and gaining insight into data from data collection and storage through data transformation, prediction, and visualisation.	Discover information from data to help with smart city planning, decision making and provide analysis for other smart city applications.	Processing and analytics: Apache Spark, Apache Flink, Apache Storm Social Media Processing: COSMOS Visualisation: Tableau, Amazon Quicksight, Microsoft PowerBI, Google Cloud Datalab
Data Interface	Interface where decision makers and citizens can access the raw and processed data, analyses, and visualisations.	Effective dissemination of data, information and knowledge to stakeholders.	Open Data Distribution: CKAN, Dataverse Restricted Interface: Web application
Smart Application	Applications developed by internal and external developers using data gathered from the components in the system.	Solve further smart city problems using the services provided by the other components.	Technology choices dependant on application area.

Chapter 5

ZEN Architecture Prototype

In this chapter, a prototype is created based on the architecture presented in chapter 4. The purpose of creating the prototype is to assess the feasibility and suitability of the architecture in the context of a smart city ICT system. The prototype is an ICT system for data management and monitoring of Key Performance Indicators in zero-emission neighbourhood pilot projects for the ZEN research centre. The prototype is limited in functionality and scope. It considers a few different KPIs, and uses data from the ZEB Living Laboratory described in section 2.2.2. The prototype is intended to give a functional example of how a ZEN data management ICT system based on the proposed architecture may work.

The prototype implements all the components the architecture with the expected functionality from chapter 4, except the *smart application* component, which is out of scope for the project. The data collection, data access, data processing, and data interfaces modules are all considered. The data being used in the prototype is energy data from a series of experiments in the NTNU ZEB Living Laboratory [15] between October 2015 and April 2016 where data was collected by the sensor network in the building.

The basic flow of the prototype begins with the data being sent into the ecosystem from an emulated sensor network. As it enters the ecosystem it is processed and stored at different levels and granularities in order for the original data to be kept intact while it is being transformed. Once sufficiently processed and transformed to show the the KPIs in the project timeframe. The KPIs are displayed for users in the open data interface and the restricted data interface.

5.1 Requirements, Limitations, and Scope

5.1.1 Requirements

The prototype should take into account the principles set by the requirements for smart city ICT architectures identified in section 4.1.2:

- Flexible processing and analytics
- Interoperable storage for heterogenous data
- Cohesive data management
- General approach
- Open data dissemination
- Vendor independent

When building the prototype, these general requirements should be considered before moving on to more domain specific requirements.

The code for the processing and analytics should be easily readable and well commented. This makes it easier to change and modify as the needs for processing and analytics in the system change and evolve over time.

The prototype only considers one data source, but storage solutions should be chosen in consideration to storing and working with a number of different data sources and data types, everything from relational databases and NoSQL tables to other unstructured and semi-structured data varying from text documents and JSON files to audio and video files.

Cohesive data management in the context of the prototype, and for any smart city ICT system with similar functionality is preservation of the data and the metadata. While the data is being transformed and altered throughout the system, it should be available in its original form, and subsequent partial transformations. Data management also refers to keeping this data readable and understandable throughout all the transformations by clearly labelling what data and what transformation has been done at every point in the process through text and by keeping the relevant code well-commented.

For the *general approach* requirement, it is related to the architectural approach for the prototype, as the implementation is built for a specific purpose (Monitoring KPIs). However, the way the prototype is built should show that the core components and infrastructure can be transferred to systems with other purposes and goals, reflecting the generality of the architecture, technologies and tools.

Open data dissemination gives citizens tools to make informed choices about the city and the environment, which can lead to smarter and more sustainable interactions between citizens and their surroundings. The prototype manages energy data for a single family home, so the data may not be particularly interesting for an open data portal. Despite this, in the context of creating a prototype, it is important to explore as many aspects of the overall architecture as much as possible. In the context of a zero-emission neighbourhood, open data dissemination is important, and to make sure the prototype is as applicable as possible to the zero-emission neighbourhood domain, the implementation of an open data dissemination solution is important in gaining more confidence in the validity and suitability of the architecture.

Vendor independence in the prototype shows that the architecture is not implementable with the technologies used and can be created and altered without the use of the same service providers.

In addition to these general requirements, the prototype should fulfil the requirements identified in [1] and listed in 2.2.4. These requirements are specifically identified for the ZEN data management and KPI monitoring ICT system and the desired functionality.

- Hosting of KPI, research and context data and metadata.
- Big data and visualisation services.
- End user applications for municipalities, citizens and businesses.
- Logs and data on the use of services provided.
- Open API for third party developers.
- Support for exploratory analysis of datasets.
- Support for management of KPI data.
- Support for research based on the data on the platform.
- Future-proof for the duration of the ZEN pilot projects at a minimum 8 years.
- Flexibility in order to adapt to changes in KPIs and collected data.

The ZEN data management and monitoring system requirements are more specific than general smart city ICT system requirements. Both sets of requirements are needed to help guide the development of the implementation and to show that the architecture is capable and suitable of creating a cohesive data management system for zero emission neighbourhoods in smart cities.

5.1.2 Limitations

The main limitation of the prototype is the data that is accessible. As ZEN is relatively new, there is little data accessible for the prototype from operational ZEN pilot projects. The best data accessible at the time of the development of the prototype was data from the ZEB project, the predecessor to the ZEN project, where detailed sensor data was collected from the Living Laboratory described in 2.2.2. Due to the lack of more neighbourhood-oriented data from a ZEN pilot area, the prototype is limited to using this single building-specific data. This is a limitation that affects the requirements for the architecture in terms of **cohesive data management**. The requirement refers to many aspects of data management, and one of these aspects is the management of data from an entire neighbourhood, where data from several building like the Living Laboratory would send similar but not necessarily identical data into the system. Although this is a limitation of the prototype in regards to a full data management system, it does not make a prototype with a single data source invalid, especially if the environment the prototype is built in supports the addition of multiple sources without major alterations to the core of the architecture for the system.

Another limitation is the number of KPIs the prototype can monitor. Based on the data that is accessible, only some of the KPIs can be calculated. This is however not a large problem for the prototype, as it still shows the ability to monitor KPIs, and functionality for adding more.

5.1.3 Scope

The scope of the prototype includes satisfying the requirements for the ZEN data management system and requirements for general ICT systems. It is smaller than a full data management system in a neighbourhood or a city, as it uses fewer data sources and less complex data visualisation. The implementation includes all components of the proposed architecture except for the smart city application component, which is considered an extension of the ICT system, rather than one of the core components. The scope of the prototype is justifiable if it successfully shows the suitability of architecture in the ZEN context.

5.2 Data Description

The data being used for the prototype is collected from the ZEB Living Laboratory described in section 2.2.2. The data was collected between 09.10.2015 and 24.04.2016. During the collection period, six different groups of people inhabited the laboratory at different times. It was inhabited in the following intervals by different compositions of inhabitants: individuals, couples, and families:

- Group 1: 12.10.2015 - 06.11.2015
- Group 2: 09.11.2015 - 04.12.2015
- Group 3: 04.01.2016 - 29.01.2016
- Group 4: 01.02.2016 - 25.02.2016
- Group 5: 27.02.2016 - 23.03.2016
- Group 6: 28.03.2016 - 23.04.2016

Data was collected from all of the sensors in the building [15]. For this prototype, the focus is on the energy KPIs, which means the data collected on energy consumption and generation in the building is most relevant, and data collected about other aspects is not included in the data for the prototype.

The first group of inhabitants did not move in until 12.10.2015. However, the data collection started earlier than that, on 09.10.2015. The sensors did not stop collecting data during the time between the different groups, which means that there is data collected when nobody was living in the building. This leads to noticeably large variation in energy consumption between the time periods where people lived in the building, and when it was empty.

For energy consumption, the data collected comes from sensors monitoring the following aspects of the building:

- Monitor and control system
- Fridge
- Hob
- Oven
- Extraction hood
- Dishwasher
- Washing machine
- Tumble dryer
- Socket living room south and entrance
- Socket living room north
- Socket kitchen
- Socket bedrooms
- Socket bathroom
- Socket technical room
- Lighting
- Shading systems
- Window motors
- Hydronic circuit pumps and actuators
- Water tank lower electric coil
- Water tank upper electric coil
- Ventilation AHU
- Ventilation AHU electric coil
- Heat pump

For the energy generation, the data was collected from the following aspects of the building:

- Surface collector field
- Solar thermal panel
- Inverter PV roof south
- Inverter PV roof north

The sensors collected and sent data every 30 seconds, providing a timestamp as well as the measured value for the sensor. For sensors collecting energy data, the measured value for the sensor is cumulative watt hour (Wh) for the sensor since last reset. Table 5.1 shows en example row from the dataset being used in the prototype. Actual energy consumed or produced in the interval between two measurements of a sensor value is calculated by taking the difference between the most recent value and previously registered value for that sensor. For example, if the sensor for the energy consumed by the dishwasher measures 300 Wh at one reading and 302 Wh at the next reading 30 seconds later, the energy consumed by the dishwasher in that timespan is 2Wh.

Table 5.1: Example of a row from the ZEB dataset. Every column is a different consumer or producer of energy. Rows contain values of each energy related sensor, either consumption or generation in the format of total watt hour registered by the sensor. Energy consumed or produced in the interval between two rows is the difference between the two values.

Timestamp	*Dishwasher*	*Kitchen socket*	*Solar PV 1*	...
09.10.2015;17:38:41	80.0	57.0	35.0	...

5.2.1 Dataset Quality

The dataset has some shortages in terms of quality which require attention and management when being used.

Although most of the data is timestamped with 30 second intervals, there are gaps in the data, where the monitor system has not been able to collect the values from the sensors, so the raw dataset is not entirely continuous, with gaps occurring with varying frequency. Gaps are dealt with by populating them with values where no energy generation or energy consumption occurs. The difference between the cumulative watt hours are calculated, giving the actual energy generated between sensor messages, and the gaps in the dataset are populated with new data showing no energy consumption or generation in the time when the original sensor data was not captured. This makes aggregation and visualisation smoother, as the dataset has values every 30 seconds for the entire period of data collection. There is also no actual change to any aggregated or processed values, as the new rows of data are timestamps and rows where no energy consumption and production has happened. There is no guarantee that no consumption and generation happened during the gaps, but since no data was collected during that period, the prototype assumes no energy was produced or generated.

In terms of the reliability of sensor values, the data is measured in accumulated watt hour (Wh) for the sensor since last reset. A reset could happen at any time, and were mostly caused by windows updates and power outages. When the sensor resets, its cumulative value is set to 0 again. This means that the calculated value for consumption or generation between the sensors current value of 0 and its last value will be negative. Negative changes is production and consumption only happens when a sensor resets, as changes in sensor value are positive if energy is generated or consumed, and if not, the change in value is 0. By finding every instance of negative generation or consumption in the dataset, we can account for sensor resets by changing the negative values to 0. This may impact intervals where generation or consumption happens at the same time as a reset, but it happens infrequently, and since the value is recorded every 30 seconds, this has little impact on even hourly aggregations.

Issues like these are expected when dealing with such fine grained sensor data over a relatively long period of time. Considering the issues and the way they are solved in the prototype have little impact on even the smallest aggregation level (hourly) due to the frequency at which data is collected, it is determined that the data is of a sufficient quality to be used for the prototype.

5.3 Technologies

In the interest of being able to quickly create a functional prototype that illustrates the applicability of the architecture, many parts of the prototype are pre-made Azure services. Remaining technologies are chosen with respect to how familiar the developer was with the tools and their interoperability with the rest of the tools and technologies. This section gives a general description of the tools and technologies used in the prototype.

5.3.1 Microsoft Azure

Microsoft Azure is one of the major cloud services. It has a number of different services that enable users and developers to host, develop, deploy, and manage their applications in the cloud, eliminating the need for local hosting and management.

In addition to general application development services, Azure provides components and services intended for the development big data applications that are relevant for the prototype: Storage (Relational through SQL and semi/non-structured through NoSQL document databases and object storage), data stream management (IoT Hub, Event Hub), processing, analysis and insight in data (Stream analytics, HDInsight clusters), as well as tools for visualisation through integration with PowerBI.

Azure also provides data management services that, while not part of the prototype, can potentially be integrated into later iterations of the prototype in order to further extend the capabilities in terms of analysis, storage, hosting and management of a smart city ICT system. This includes data warehousing, web application deployment, content delivery network services and machine learning. The component based nature of cloud services provides a framework where it is possible to extend the prototype with new modules if system grows, changes, and evolves.

Azure Storage

Azure Storage provides an object store, a file system, messaging and a NoSQL store as a scalable cloud service. It can provide both local and geographically distributed redundancy to uphold availability as well access control for security. Azure provides SDKs for accessing the data through many popular programming languages (Java, .NET, Node.js, Python, PHP, Ruby) as well as a REST API.

The *Blob Storage* is the Azure object store, and is optimised for storage of large amounts of unstructured data like text, binary files or multimedia. The data can quickly be accessed for analysis and further processing in both Azure and outside the Microsoft ecosystem.

The NoSQL *Table Storage* in the Azure storage use Cosmos DB to provide schema-less key-value storage that can effectively store non-relational data tables.

Event Hubs

Event Hubs is the Azure data streaming platform that takes care of storing events and data generated and sent by IoT devices and sensors. It can be used as an entry point into the rest of the system, taking data from sensors and forwarding it to different services for storage and processing. This can either be done directly through setting specific outputs of the module, or components can subscribe to data from certain streams, if the event hub manages multiple streaming data sources.

Stream Analytics

Azure Stream Analytics is an event-processing engine for real-time streaming data from IoT devices and sensors. The streaming data can arrive from a module like Event hubs. The input for the analytics module is defined through a stream analytics job, where processing and transformation is done to a input data stream. The transformed data stream can then be sent to another component, application, or to storage. The stream analytics module is highly scalable, allowing one instance to perform work on 1 GB of incoming data per second. Using the publish-subscribe model with Event Hub, data can be divided amongst jobs, increasing the scalability further to fit contexts where extremely large amounts of data arrive at the system at the same time.

HDInsight

HDInsight is Microsofts open-source analytics service. It lets users host their preferred open-source big data framework (Hadoop, Spark, Hive, Kafka, R, etc.) and use the Azure cloud to create and scale clusters on demand to be able to dynamically shift the processing power of the application based on the amount of incoming data. In this prototype, Spark has been chosen as the big data framework in HDInsight.

PowerBI

PowerBI is a Microsoft SaaS (Software-as-a-Service) solution for data exploration and visualisation. It can be used in order to visualise the data and the information gained from the processing and analysis that happens earlier in the application. It is a tool designed to help decision makers by providing a set of visualisations that can be exploratory manner by drilling up and down and changing the scale of which the data is aggregated on. The visualisations can then be added to a dashboard that can then be published on a web page where it is accessible to users.

5.3.2 Apache Spark

Apache Spark is an open source data processing engine often used for big data processing and analysis. It is a cluster computing framework allowing for large amounts of data to be processed at the same time using resilient distributed datasets [58]. Spark focuses on data parallelism and fault tolerance during the data processing operations, and is based on the MapReduce [11] framework. It differs from MapReduce in that it works with the data in memory rather than on disk, meaning data can be processed a lot quicker than in traditional MapReduce applications. Spark is able handle both streaming and batch data processing, making it a very versatile tool in big data applications. It can be deployed on Azure HDInsight clusters. Spark applications can be written in a number of languages, with Scala, Java, Python and R being the most common programming interfaces.

5.3.3 CKAN

CKAN (Comprehensive Knowledge Archive Network) is a open source, web-based data management system. The purpose of CKAN is to store and disseminate open data. It is used to create, manage and publish websites where open data is accessible in an open data portal format, where users can view and download the datasets.

The way CKAN works is that it lets cities, municipalities and other data holders create their own web portal for publishing open datasets based on the CKAN framework. Much of the data portal is pre-made by CKAN, but it is configurable and adaptable for different purposes.

CKAN is used by both national, local and municipal governments in order to publish and disseminate their data to the public:

- Open data Berlin[1]

[1] Open Data Berlin- `http://daten.berlin.de/`

- DATA.GOV.UK - Open data for UK government [2]
- DATA.GOV - US Open Data [3]

5.4 KPI Selection and Description

Based on the size of the prototype and the data available, a selection of KPIs from the energy category was made.

Based on the data, relevant KPIs are chosen from the energy category of Table 2.1. The following are the KPIs monitored by the prototype:

- Energy use - kWh/day, kWh/month, kWh/year
- Energy generation - kWh/day, kWh/month, kWh/year
- Self consumption - daily and monthly percentage (%)
- Self generation - daily and monthly percentage (%)

The prototype deviates from Table 2.1 in terms of the timeframe for the KPIs. In Table 2.1 and the ZEN guideline report [54] these KPIs are managed and displayed on a yearly basis. For the dataset used in the prototype, this timeframe of data aggregation is not as meaningful or interesting as the only aggregation due to the short timespan of the data collection. The dataset does not contain a years worth of data, having collected data for less than three months in 2015 and just over three months in 2016. For the purpose of creating and testing a prototype, it is more interesting to have a larger number of data points in transformations, intermittent datasets and visualisations. Adding per-day and per-month aggregations as well as the original per-year aggregation for energy use and energy generation and per-day and per-month calculations for self-generation/consumption gives a more interesting prototype in terms of showing the potential of the storage, data processing, and visualisation.

Energy use and energy generation is measured by summarising the values of every column in a row of the dataset into either *total energy consumption* or *total energy generation* for that sensor timestamp based on whether the column is a sensor for measuring energy consumption or energy generation. The rows are then aggregated into the desired time intervals and the values corrected so they display kilowatts hours instead of the original watt hours.

Self consumption and self generation are more complex calculations. In the ZEN guidelines [54] they are defined as two complementary aspects of the interaction between

[2]DATA.GOV.UK - `https://data.gov.uk`
[3]DATA.GOV - `https://www.data.gov`

energy use and generation, and describe the mismatch between energy generated at the building and energy used by the building. Precise description of the terms are:

Self-consumption: What percentage of the energy generated locally is consumed locally. Looking at Figure 5.1, the ratio of self-consumption would be area C, which is the self-consumed part of the locally generated power relative to all generation that day: Area B + C. Calculating the self-consumption ratio can be done using the following equation from [54]:

$$SelfConsumption = \frac{C}{B + C} \tag{5.1}$$

Self-generation: What percentage of the energy consumed locally is generated locally. From Figure 5.1, the ratio of self-generation is area C relative to total energy use: A + C Calculating the self-generation ratio can be done using the following equation from [54]:

$$SelfGeneration = \frac{C}{A + C} \tag{5.2}$$

The values for self-generation and self-consumption will have the same value if annual energy generation is equal to total annual energy use, leading to annual net zero energy demand, which is one of the goals for the ZEN pilot projects. For other situations than net zero energy demand, the values of self-generation and self-consumption will reflect each other. If a building or a neighbourhood generates only small amounts of energy, self- consumption will be high (close to 100%) while self-generation will be low (close to 0%). If the neighbourhood becomes a net exporter of energy, meaning it generates more energy than it consumes, the behaviour of the values for the two KPIs are switched, with self-generation being higher than self-consumption. The equations calculate the ratio on a scale from 0 to 1 before it is converted to a percentage.

5.5 Prototype Architecture

When developing an architecture for the prototype, it was important to keep in mind what the goals of the prototype are. In addition to creating a KPI monitoring system for ZEN, the goal of the prototype is to show the applicability of the architecture proposed in chapter 4. It is therefore important to keep the ideas, solutions, and components proposed by the original architecture in mind when designing the architecture for the prototype. The value of the prototype in terms of validating the architectural choices is significantly less if it does not adhere to those choices when adapted to a functional ICT system. When designing and developing the prototype, the relationship between the original architecture and the actual implementation was kept in mind throughout the process.

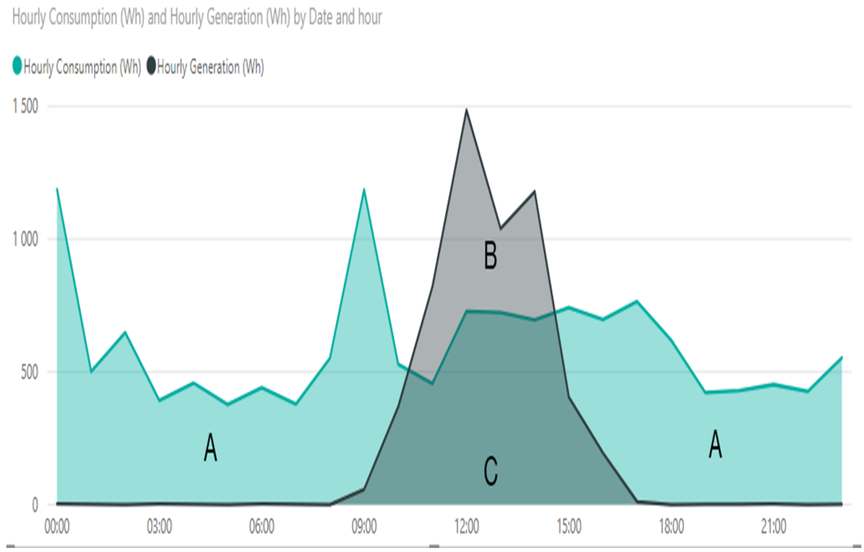


Figure 5.1: Example of daily energy consumption and generation profile. Area A is the energy consumed that day. Area C is the part of the daily generated energy that is consumed in the same timeframe of the generation. Area B is the excess of the generated energy.

The architecture of the prototype, the data flow and individual modules is shown in Figure 5.2. The data being used for the prototype has been collected from previous experiments, and is available as a complete dataset. This means that to get a prototype that takes every aspect of the environment of data management in smart cities into account (from collection to dissemination), the data collection aspect needs to be emulated. This was accomplished in the prototype by creating a script that sent the dataset to the Azure Event Hub row by row in specific time intervals, as if it was collected directly from sensors. The dataset then arrives as perceived streaming sensor data to the Event Hub. It is then preprocessed in the Stream Analytics module, before being stored in an Azure storage blob. The next step is more detailed data processing in the HDInsights module. For the processing module, Apache Spark[4] was chosen. The Spark cluster processes the data in batch, transforming the preprocessed sensor data into values for the KPIs. After the data has been processed in Spark, it is stored in a new storage blob. The storage blob containing the processed data connects to both a CKAN website for open data dissemination, and to PowerBI for creation of visualisations for the restricted decision-maker data interface.

The processed dataset is published to CKAN where it can be downloaded and viewed. CKAN is connected to the dataset via url, meaning if the dataset is added to, altered, or transformed in any way in the Azure blob, this will be reflected in the CKAN portal. The PowerBI dashboard is also intermittently updated with the newest data from the dataset in the storage blob. In PowerBI, the data is further transformed as needed to fit the visual-

[4]Apache Spark - `https://spark.apache.org/`

isations that are desired for the restricted data interface. The visualisations are added to a dashboard and published to a webpage where the they can be viewed and explored by users.

5.6 Prototype Development

The prototype is developed with the tools and components listed and described in the previous section. Many of the components in the prototype are Azure components, where development mainly consists of configuring the components to work with the data. Programming was necessary mostly for the data processing, where T-SQL and some Javascript was used for preprocessing and Python with Pandas[5] and pySpark[6] was used for the detailed processing and transformation of data into KPI values. Some Python programming was also necessary in order to emulate sensors transferring data to the prototype.

5.6.1 Data Collection Component

The data used in the prototype had been collected during experiments in the ZEB Living Laboratory. The data was originally collected in TDMS[7] files that recorded all sensor data in the building. The energy data had for a previous project been extracted from these files and collected in a tab-separated file for easier access.

In order for the data to appear as if it enters the rest of the prototype as real-time sensor data, a python script was written that parsed the dataset line by line and sent each row individually to an active instance of an Azure Event Hub. The Event Hub is a component for routing streaming data sources to other components in the system, and in the prototype the data was sent to the Stream Analytics component, where light preprocessing was done before storage. Due to the minor level of preprocessing, it was determined that storing both the raw data and the preprocessed data was not necessary, and only the preprocessed data is stored.

5.6.2 Data Access Component

Data access in the prototype uses the blob storage in Azure. Blobs exist as a part of the Azure storage account, and is typically meant for storing binary files and unstructured data. From the stream analytics preprocessing, the dataset is stored as a JSON file in an Azure blob. From there, the file can be accessed remotely through the Azure Storage API,

[5]Pandas - `https://pandas.pydata.org/`
[6]pySpark - `https://pypi.org/project/pyspark/`
[7]NI TDMS File Format - `http://www.ni.com/white-paper/3727/en/`

and can easily be accessed from other Azure services in the file-system structure of the Storage account.

The access component also consists of the final processed data that arrives after being processed by Spark in the HDInsight module. This is stored in a different blob than the preprocessed data, and for the ease of CKAN end users, the processed datasets are stored as tab-separated CSV files.

5.6.3 Data Processing and Analytics Component

The data processing and analytics component consists of three modules in the prototype, performing various transformations of the data.

The preprocessing module uses Azure Stream Analytics. It receives the raw data from the Event Hub, and does some minor preprocessing as the streaming data arrives, before sending it to the storage module. The stream analytics module in Azure uses T-SQL and Javascript transform data and perform queries. The preprocessing done consists of string formatting to make timestamps more easily compatible with the python libraries, and of casting different fields of the incoming data to the correct data types before sending it to storage.

The main part of the data processing happens in the HDInsight module. This is where the preprocessed data is taken from the storage blob, and transformed into the KPI values. The HDInsight module runs a Spark cluster with a pySpark script doing the necessary transformations and aggregations on the preprocessed data. The transformed datasets containing KPI values are then sent to storage for further use. The script first summarises each row of data into either the consumption column or the generation column. It then does the data cleaning described in 5.2.1 by first removing the values with negative growth (sensor resets) and resampling the dataset every 30 seconds. Any gaps are filled with data that does not impact the aggregations, but gives the dataset the same time difference between each row of data. Energy consumption and generation is then aggregated with the *resample().sum()* method for hourly, daily, weekly, and monthly aggregation of the values and converted to kilowatt hours. Daily and monthly self-consumption and self-generation is then calculated with equation *5.1* and *5.2* from section 5.4. hourly, daily, weekly, and monthly energy consumption and generation, and daily and monthly values for self consumption and self-generation are then written to an Azure storage blob as individual tab-separated CSV files. The full pySpark script with comments to the code can be found in Appendix A.

The final part of the data processing happens in Microsoft PowerBI. The processed data from the HDInsight module is imported to PowerBI and prepared for visualisation. This consists of adjusting data types. Most of the dataset columns are automatically detected as text values, and must be changed to date and time values and numerical values so the datasets are compatible with the PowerBI visualisations. The visualisations for the prototype is chosen according to the guidelines for the pilot studies [54]. For energy consumption and generation, a line graph shows the relationship between the daily KPI values and makes it easy to visually compare the two values. The self-generation and self-consumption values for each month are shown in a table. A daily energy profile area graph is also created in order to see hourly energy consumption and generation with the associated self-generation and self-consumption values for that day shown in a table. The visualisations are placed in a PowerBI dashboard and the data is completely done being transformed.

5.6.4 Data Interfaces Component

The prototype has two interfaces to access the data. The open data interface and the restricted data interface.

The open data interface is a website where the datasets are publicly published and can be downloaded as CSV files. For the purpose of the prototype, the website was hosted locally on a Ubuntu server. The website is created with CKAN which lets users create a fully functional open data portal. CKAN includes functionality for user creation and organisation creation, and organisations can freely publish datasets on the portal. The data is published to the open data platform via a URL source to the Azure Storage. CKAN uses Apache Solr[8] as a search platform, so datasets are searchable after they made available on the portal. After the dataset is published, it can be freely downloaded by anyone using the portal.

The restricted interface is where stakeholders and decision-makers in the ZEN projects can monitor the KPI values for the project. The dashboard created in PowerBI is embedded in a webpage with HTML and accessible through web browser on any device (mobile phone, tablet, laptop, desktop computer).

5.6.5 Privacy and Security Component

Implementing privacy measures beyond what is built in to the software and hardware that was used was not a priority in the prototype. There are built-in security measures that Azure provides, which is one of the advantages of using a cloud service provider when building a smart city application.

The Azure SDK that is used when sending data to the Event Hub provides symmetric key encryption so clear-text data is not sent. Event Hub provides tools for key management as well, so different keys can be used for different applications or data collectors. Azure storage encrypts all data stored using 256-bit AES encryption [10]. Azure also requires user authentication to access and manage components in the ecosystem. With these features, Azure provides some basic cloud security and privacy measures.

[8] Solr - `https://lucene.apache.org/solr/`

5.6.6 Data Policy and Data Management Component

The homogenous nature of the data used in the prototype makes it difficult to assert some specific data management strategies that were used. The data is preserved at every level of processing. The first data being stored in the prototype has been through the preprocessing component. The preprocessing is quite light, only doing minor string manipulation and data type casting. It is considered quite close to the raw data. The preprocessed data is stored in JSON format as it is a common format for sending data over the internet. All of the transformed KPI datasets are stored on the same file format (CSV). Both these data formats are quite common and compatible with a lot of other software and data processing tools, so there is no technological barrier to using the data produced by the prototype. The need for complex data management strategies are minimal. Special consideration is given to making all the raw and processed data available in storage so no data is lost. The data is also aggregated into hourly, daily, weekly, and monthly values, such that if some information is lost at a level of aggregation, other levels of aggregation are available.

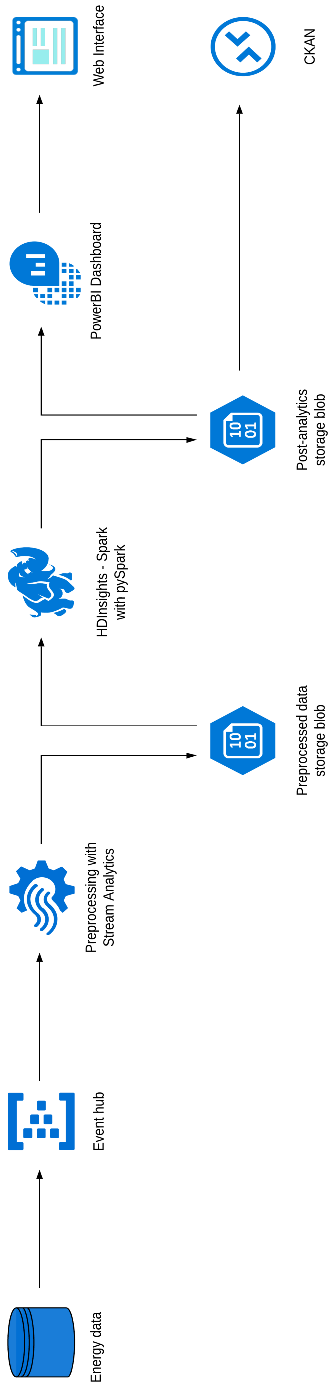


Figure 5.2: Components and data flow for the Azure prototype.

Chapter 6

Results

The result of the thesis is the ZEN data management and KPI monitoring ICT prototype, and the architecture that facilitated its development.

6.1 ZEN Data Management System

The results of the ZEN monitoring prototype are best shown through the data interfaces where the data is available and the KPIs can be monitored. The prototype produced the two interfaces discussed in the proposed architecture, the open data interface and the restricted interface. The two interfaces provide functionality for two different groups of stakeholders. The open data portal in primarily for public use, and for use by potential third parties who require access to the datasets produced by the data management system for further application. The KPI Monitoring Dashboard is for stakeholder involved in planning and managing the monitored ZEN pilot project.

6.1.1 ZEN Open Data Portal

The open data portal/interface is a CKAN-based webpage where organisations can publish their datasets and anyone can access and download them. All of the datasets produced by the data processing in the prototype are published to the portal: Hourly, daily, weekly and monthly aggregations of energy generation and consumption, and daily and monthly self-generation and self-consumption percentages. The datasets are published to show off CKANs suitability as a open data dissemination interface for smart city data management ICT system.

The front page of the portal is shown in figure 6.1. The page consists of a search field where all the datasets on the portal that are tagged with a category can be searched for. It also gives a brief view with descriptions of some of the datasets on the page. The *datasets* button directs users to the overview of available datasets. In the dataset overview, shown in figure 6.2, users can view and search for different datasets. The list of datasets can be filtered by publishing organisation, tags, dataset format and license. Dataset tags are defined and added by the organisation or user that publishes the data. From the dataset overview the datasets can be accessed, and figure 6.3 shows the detailed view that a user is directed to when they want to download a dataset. The view gives users a description of the dataset, an option to follow changes in the dataset, view tags and detailed information about when it was created and last changed. From here the dataset can be viewed online and downloaded.

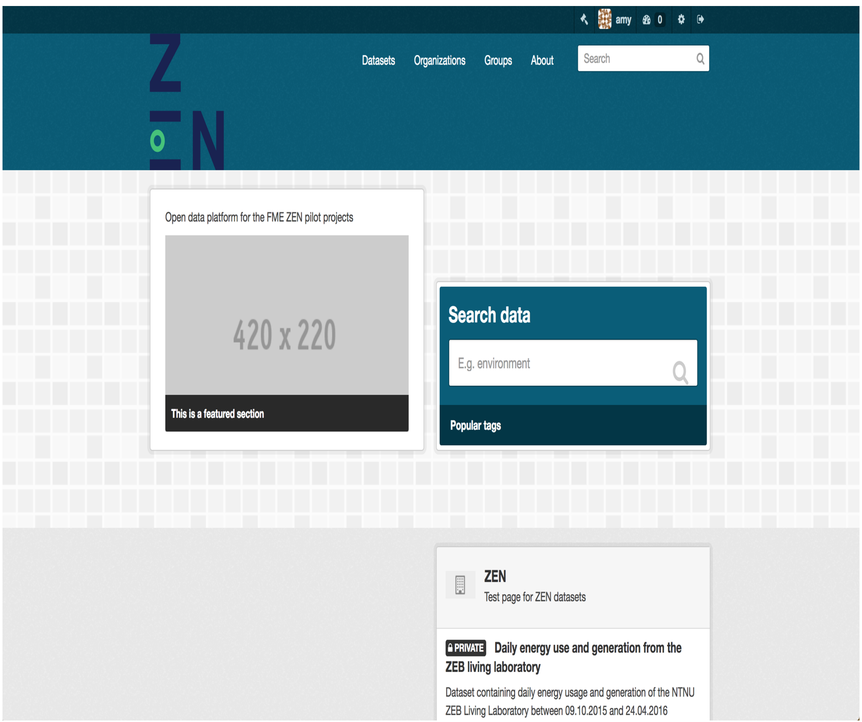


Figure 6.1: Front page of the ZEN Open Data Portal

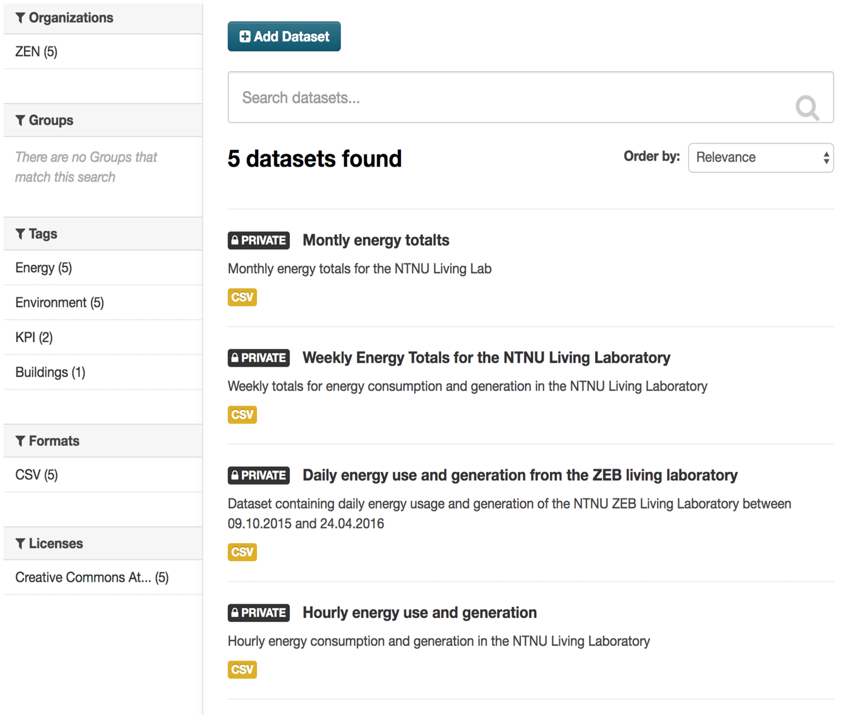


Figure 6.2: Overview of available datasets on the ZEN Open Data Portal. The view of datasets can be filtered according to user made tags, listed on the left side.

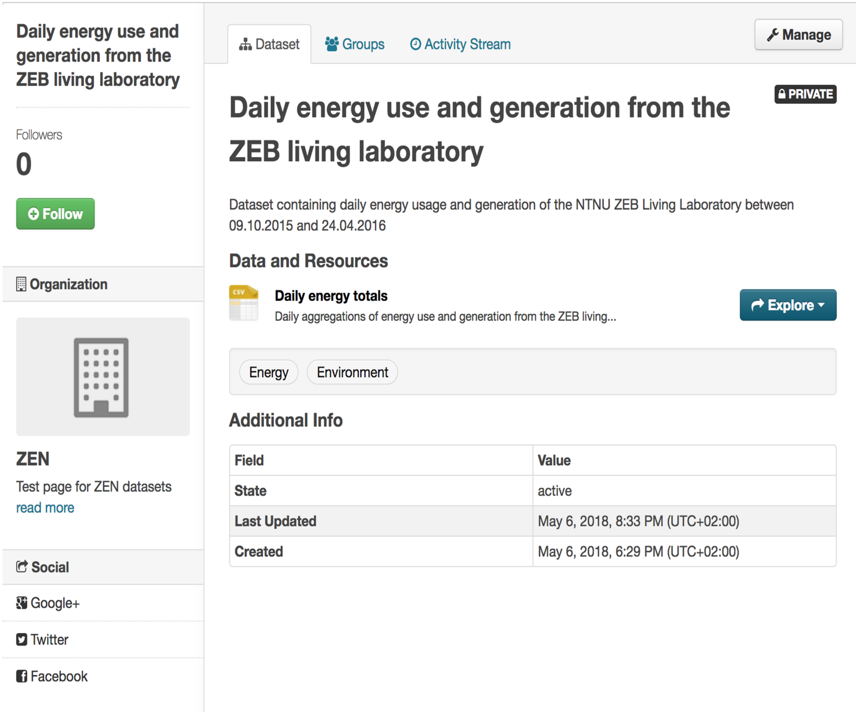


Figure 6.3: View for detailed information about each dataset in the ZEN Open Data Portal. Includes information about when the dataset was added to the portal, when it was last updated, user made tags, and a description of the dataset.

6.1.2 ZEN KPI Monitoring Dashboard

As opposed to the open data interface, which is intended for public access to the datasets produced by the data processing components of the prototype, the KPI Monitoring Dashboard (the restricted data interface) is intended for stakeholders involved in the planning, research and decision-making in the monitored project.

The dashboard (shown in Figure 6.8) consists of two different parts that function independently of each other, divided horizontally. The top part of the dashboard is a line graph showing the evolution of energy consumption and energy generation of the ZEB NTNU Living Laboratory throughout the data collection period and the monthly self-consumption and self-generation percentage values. The graph chart allows users to view the data at different granularities. Data from the entire period can be viewed as aggregated on a per year basis, per month basis or per day basis (Figure 6.4) by using the *drill up* and *expand all down one level in the hierarchy* buttons in the top left corner of the visualisation.

Users can also drill down on a specific year and month (shown in figure 6.5) by enabling “drill down mode” in the top right corner of the chart, and clicking on the desired data point on the graph. Navigating the levels can also be done by right clicking and selecting “drill up” and “drill down”. The drill down to month level also highlights the self-consumption and self-generation value for that month (Figure 6.5)

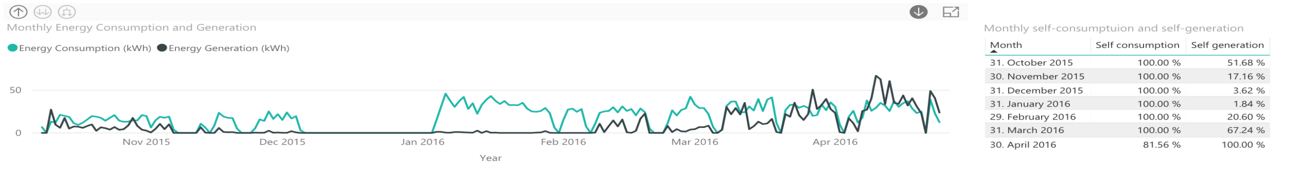
The bottom of the dashboard (Figure 6.6) consists of three parts. A date picker, an area chart (Figure 6.7) showing the daily energy consumption/generation profile of the NTNU Living Laboratory and a table containing the value for self-consumption and self-generation for the selected date. The users pick the day they want to see the energy profile for, and the area chart and self-consumption/generation update based on that selection.

The dashboard is available online.[1]

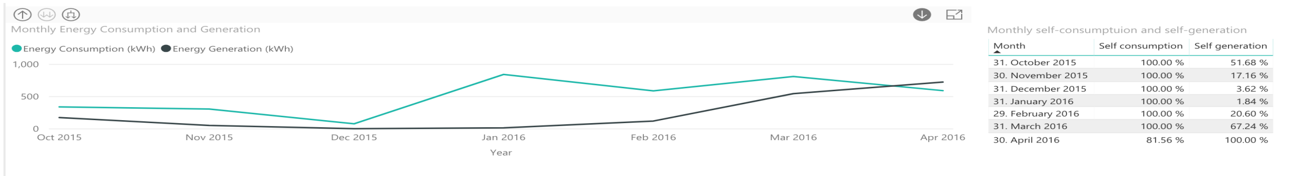
[1]ZEN KPI Monitoring Dashboard - http://folk.ntnu.no/andreay/webpage

6.2 Components and Architecture

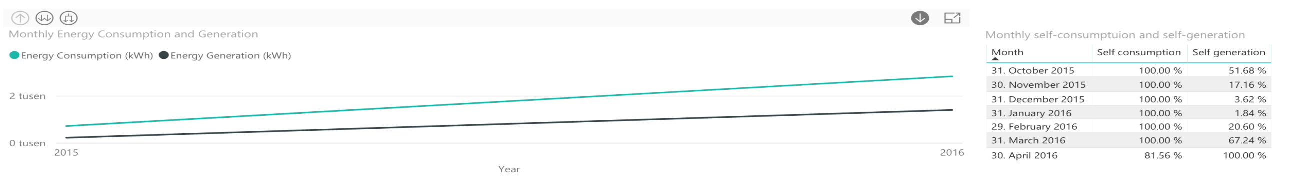
A result of the research conducted is the blueprint and component design for planning, designing, and developing smart city data management ICT systems from the basis of the proposed architecture. The successful use of the architecture suggests that the architecture is be suitable for facilitating the development of these kinds of systems. Figure 5.2 displays a concrete composition of both Azure specific and external components that make up a KPI monitoring ICT system for zero emission neighbourhoods. The selection, arrangement, configuration, and coding of these components in the way described in this thesis result in a functional prototype of a smart city data management ICT system.



(a) Daily Aggregation



(b) Monthly Aggregation



(c) Yearly Aggregation

Figure 6.4: Dashboard can drill up and down in the data to see variations in consumption/generation over entire period on different aggregation levels. Monthly self-consumption/self-generation percentages are presented in a table beside the line graph.

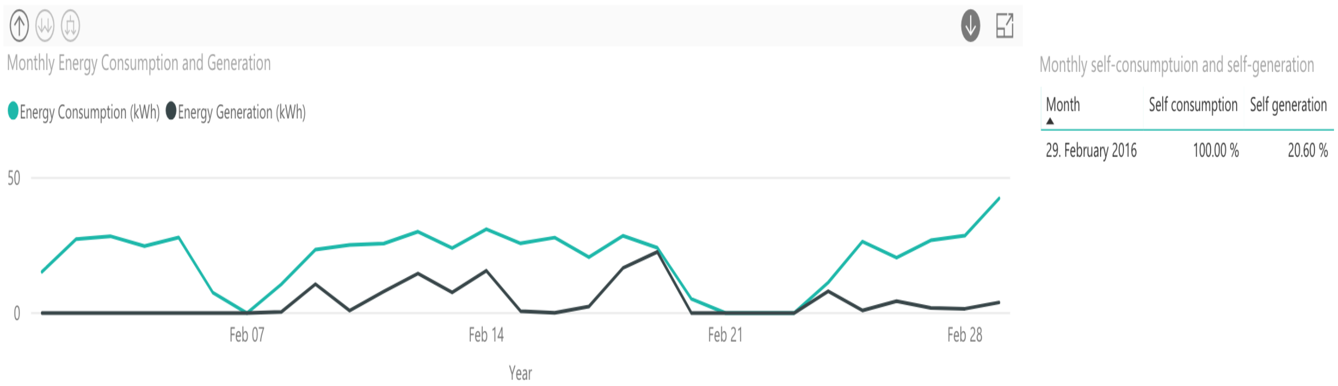


Figure 6.5: Dashboard can drill down to view daily values for a particular month of the production/consumption graph. Associated self-consumption/self-generation percentages for that month is shown.

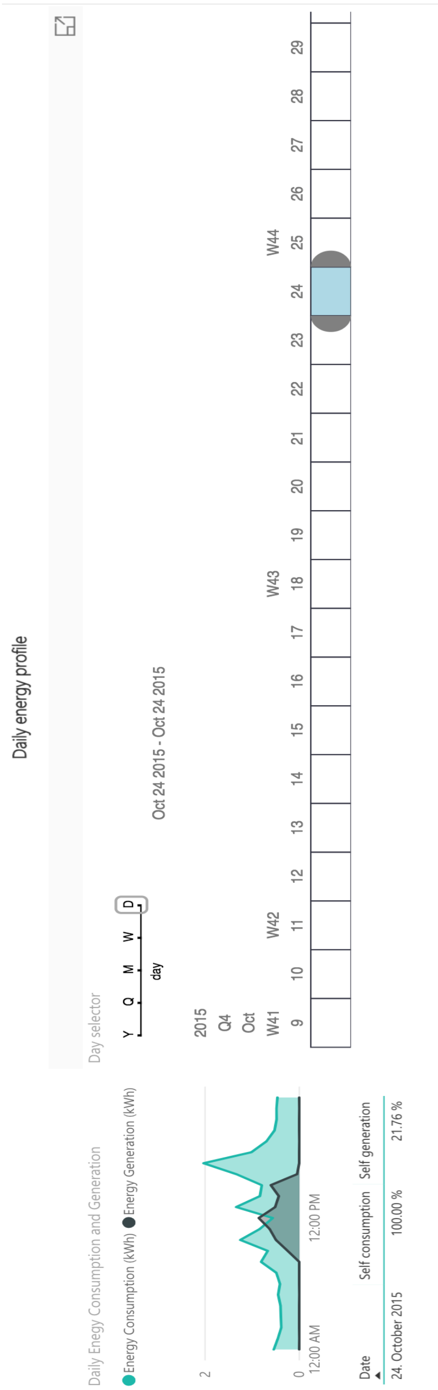


Figure 6.6: Daily energy profile with date picker.

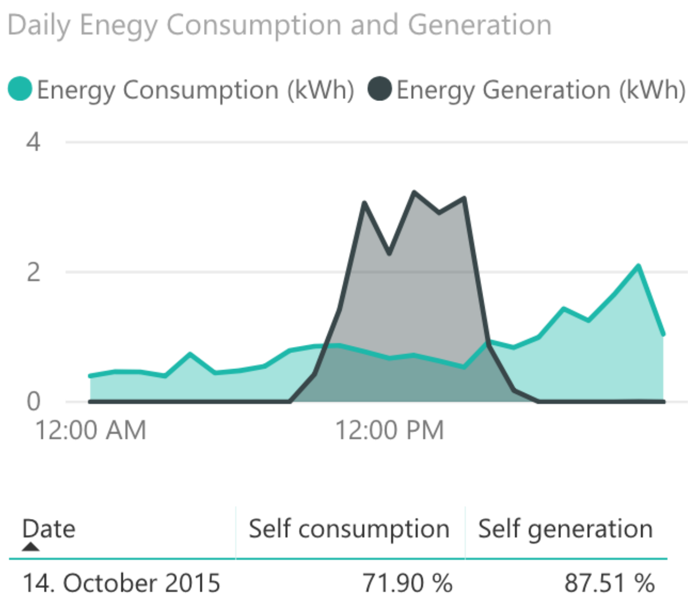


Figure 6.7: Dashboard graph of daily energy profile for energy consumption and generation. Also shows daily self-consumption and self-generation for the chosen day.

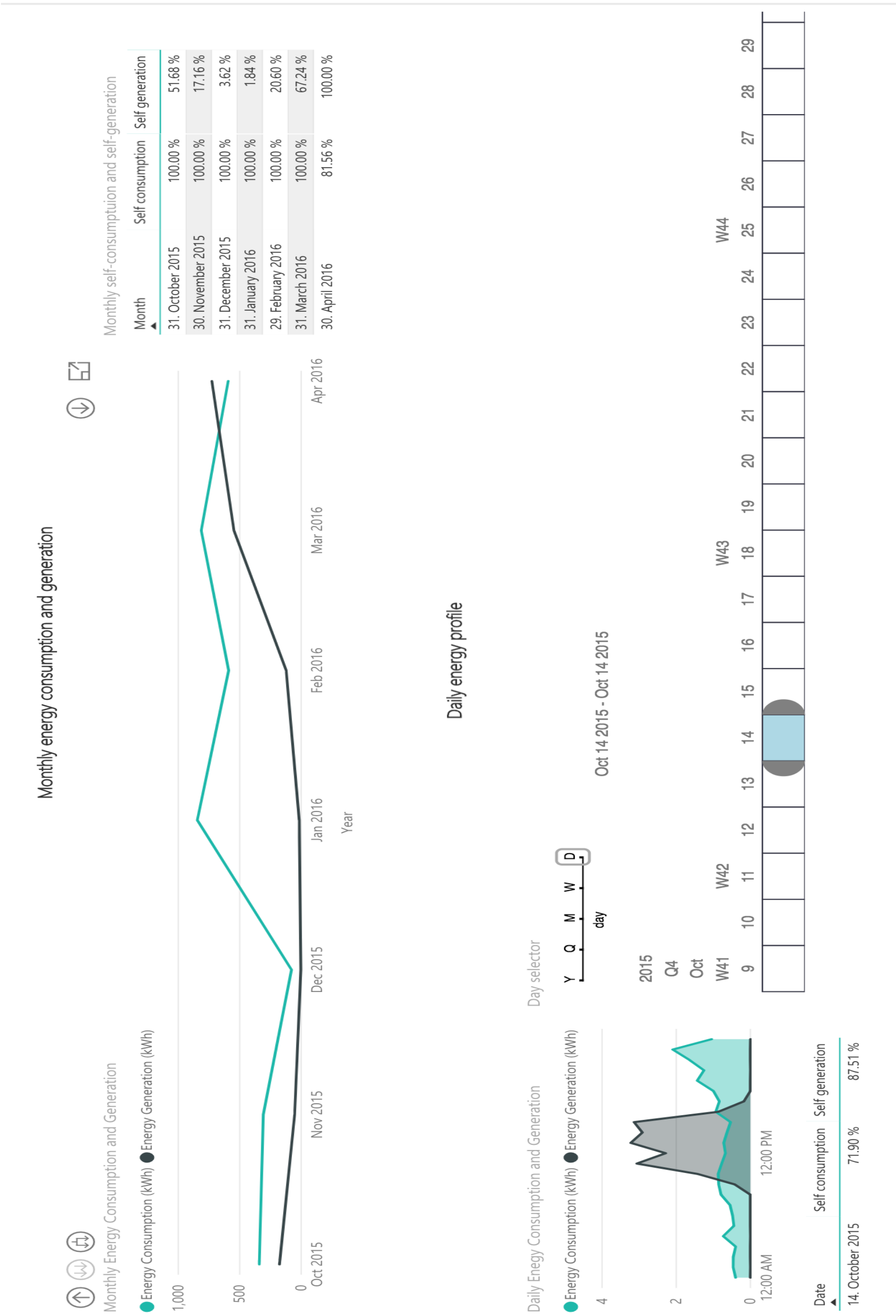


Figure 6.8: Full ZEN KPI monitoring dashboard

Chapter 7

Discussion and Evaluation

This chapter discusses the process of working on the project, and evaluates the artefact against the general smart city ICT system requirements identified in 4.1.2 from the background research, and the ZEN data management system requirements from [1] (listed in 2.2.4). The chapter also discusses how faithfully the implementation follows the structure of the original architecture.

7.1 Discussion

This thesis is a continuation of the work done in the specialisation project. The domain switch from that project (urban mobility) to zero-emission neighbourhoods required changes and a redesign of the architecture in order to fit better into the new context. Zero-emission neighbourhoods and buildings had to be researched in order to understand what the architecture and prototype had to be able to provide, and how it would be provided. The original architecture was redesigned for use in the ZEN data management system. The work on the implementation started by figuring out what the different parts of the architecture translated to in regards to the ZEN data management and KPI monitoring system. Once each component had been related to a service or technology, the actual implementation could begin.

Developing an application in the Azure environment and using Azure services requires an Azure subscription, and costs money. They provide students with 100 USD of credit, which for the most part was more than sufficient for creating the prototype. Most of the cloud services were priced by volume of use, which with a raw dataset of 145MB is manageable with the student credit. The HDInsight cluster however, was priced on a per hour of deployment basis, which, even for the smallest cluster available became expensive.

Because of this, much of the coding done for the HDInsight processing cluster was tested locally before being deployed in the Azure environment. Most of the local transformation was done using the Pandas Python library, before being translated to pySpark when run on the cluster. The script was run on the actual Azure HDInsight spark cluster only once it could perform the entire transformation, before the cluster was shut down again due to the pricing. The rest of the prototype is running all the time but due to the HDInsight service, the entire prototype can not be running all the time. In a scenario where the system is run by an organisation with funding, this might be less of a problem, since a budget for this type of system is hopefully more than 100 USD. It would be optimal to be able to run both the interfaces and make them accessible for anyone during and after this thesis. The restricted data interface (the web interface in figure 7.1) is available on the webpage provided in section 6.1.2. The open data interface was unfortunately only locally hosted, and can only be viewed in figure 6.1, 6.2, 6.3. Despite this, the prototype gives a good impression of what a data management and KPI monitoring system for ZEN may look like using the proposed architecture.

7.2 Evaluation

The general smart city ICT system requirements are listed in table 7.1, and the requirements for the ZEN system in table 7.2. The tables list the requirements and wether they were fulfilled or not. This section discusses some select requirements in detail. The discussed requirements are either not fulfilled, only partially fulfilled or worth noting due to some other reason.

7.2.1 Evaluation of General Smart City ICT System Requirements

Table 7.1: Fulfilment of general requirements for smart city ICT systems, identified in section 4.1.2.

Requirement	Fulfilled
GR1: Flexible processing and analytics	Yes
GR2: Interoperable storage for heterogenous data	Yes
GR3: Cohesive data management	Partially
GR4: General approach	Yes
GR5: Facilitation of open data dissemination	Yes
GR6: Vendor independence	Yes

Most of the general requirements have been satisfied by the prototype.

GR3 is marked as partially fulfilled due to the relatively homogenous data used in the prototype. The prototype manages its data cohesively in terms of file formats, aggregation levels and maintaining the information present in the raw data, but that seems somewhat trivial in a system with only one data source. In a potential extension of the ZEN data management system where more data sources are present, this requirement will require more policy and procedure for standardisation than what has been done in this prototype. So although the requirement is fulfilled, it is not considered fully satisfied before it is tested on a larger data management system with more heterogenous data sources.

GR6 is marked as fulfilled, but it seems necessary to highlight some factors with the assessment of this requirement. The prototype relies significantly on one technology provider: Microsoft. This does not inherently mean that it is dependent on Microsoft. The services provided by Microsoft Azure are available through the other major cloud service providers as well, and a majority of the analytics is done in Spark, which is open source and is not dependent on the Azure ecosystem. Essentially, the ZEN KPI monitoring system can be implemented in other cloud environments than Microsoft's. This means that the prototype is vendor independent, but still somewhat dependent on the existence of a vendor for the cloud services. The prototype is implemented in such a way that one vendor provides a significant part of the infrastructure and software. Cloud services provided by the same company usually interface and work well together, and along with the benefits of using cloud services (described in section 2.3.3) makes the use of a cloud provider compelling. The use of such a cloud provider is not automatically bad, and typically not a big deal. In many cases public services performed by a single private company in a city (public transport services, waste management, etc.) is the norm. Despite this, the commitment to a cloud service provider warrants consideration when creating an ICT infrastructure for a potentially significant part of the city.

7.2.2 Evaluation of ZEN Specific Requirements

Table 7.2: Fulfilment of requirements for the ZEN data management and KPI monitoring system, identified in [1].

Requirement	Fulfilled
ZR1: Hosting of KPI, research, and context data and metadata.	Yes
ZR2: Big data and visualisation services.	Yes
ZR3: End user applications for municipalities, citizens, and businesses.	Yes
ZR4: Logging and data on use of services.	No
ZR5: Open API for third party developers.	Partially
ZR6: Support for exploratory analysis of datasets.	Yes
ZR7: Support for management of KPI data.	Yes
ZR8: Support for research based on data on the platform.	Yes
ZR9: Future-proof for duration of ZEN pilot projects.	Yes
ZR10: Flexibility to adapt to changes in KPIs and collected data.	Yes

ZR1 refers to the hosting of different data in the system. This requirement is fulfilled for the prototype, as the prototype hosts all the data it takes as input and generates. It does, however, not deal with all the types of data listed in the requirement. The prototype hosts the original sensor data, which is research data. It also hosts the KPI data that results from the processing in the prototype. Some metadata is hosted about the data, such as location in the Azure file system, file type, size and date of last modification. This metadata is generated automatically by Azure, and stored in an SQL metadata database. There is no context data present in the prototype, but Azure storage is flexible enough that it would we able to handle the addition of more data.

ZR4: Logging and usage statistics were not prioritised during the development. The functionality is useful for this type of system, but for such an early prototype, other functionality was deemed more important in assessing the feasibility of the prototype based on the architecture. Logging and usage statistics should be implemented in a later iteration of the monitoring system, when the core functionality of the system is adequately implemented and more refined than in the current prototype.

ZR5 is marked as partially fulfilled. Many of the individual services in the prototype have their own APIs that allow third party developers to use the services provided by the prototype. CKAN open data portals automatically have a REST API with documentation where third party developers can import data to their applications from the portal. Azure provides SDKs and REST APIs for most of their services that authorised users can access. Access to these services are on an individual service basis, meaning third party developers would need to be authorised for each Azure resource separately. In order for the requirement to be fully satisfied, a solution where third parties can make calls to a single, unified API for all the systems services should be considered.

7.3 Comparison to ZEB Tool Dashboard

The previously created ZEB Tool dashboard [50] can in a ZEN context be useful to combine with the created data management system. The ZEB Tool dashboard is used in the early planning phase of a the LCA of a zero-emission building. It is used to calculate the total GHG emissions of different construction materials used in the building over their entire lifecycle. The ZEB tool dashboard is a simple calculation tool that uses pre-defined values for materials and calculates GHG values based on the users choice of materials and parameters such as layers and thickness. The ZEN data management system built as a larger system, facilitating data collection, storage, along with the processing and visualisation aspects that the ZEB current ZEB dashboard look at. The prototype for the ZEN management system is also focused on monitoring operational energy consumption and generation. The two systems are quite different but can complement each other in a larger ZEN system if they are integrated and scaled up fully to the neighbourhood level.

7.4 Relationship Between Conceptual Architecture and Implementation

One of the goals of the implementation of the ZEN data management and KPI monitoring system is to show the applicability and suitability of the created architecture in the context of the zero-emission neighbourhood domain of smart cities. In order for the prototype to illustrate this applicability, it is useful to show how the implementation correlates with the original architecture. In figure 7.1 the components of the prototype are shown in relation to the original architecture, with each Azure service being linked to the components and layers of the architecture.

The collection component consists of the ZEB energy dataset and the Event Hub. The reason the original data set is in the collection component is due to the way that the prototype operates. In chapter 5, it is explained that the data is sent to the Event Hub from the dataset row by row to emulate sensors streaming data to the Event Hub. In the context of the prototype, the dataset is seen as sensor observations arriving as a stream, rather than a complete dataset from the start. Without this solution, the prototype would not have a data collection component at all. The rest of the prototype fits well within the different layers and components of the original architecture. The analytics component consists of the preprocessing, Spark, and PowerBI services and the data access component contains two separate storage blobs. The original architecture distinguishes between internal data and external data in the system. The prototype only uses one data source which is provided by the ZEN centre themselves, which means all the data is in the internal data component, and management of access to external data sources is not assessed. In future work on the prototype, where more ZEN related data sources are available, assessing the access to external data sources by the prototype should be a priority. The data interface component from the architecture consists of both the CKAN open data portal and the “web interface” in figure 7.1. These two interfaces are the open and restricted data interfaces in the architecture. The restricted web interface is in the prototype available to anyone without any access restrictions, but it is designed as an interface for decision-makers, rather than an open data portal, which is the main distinction between the two interfaces.

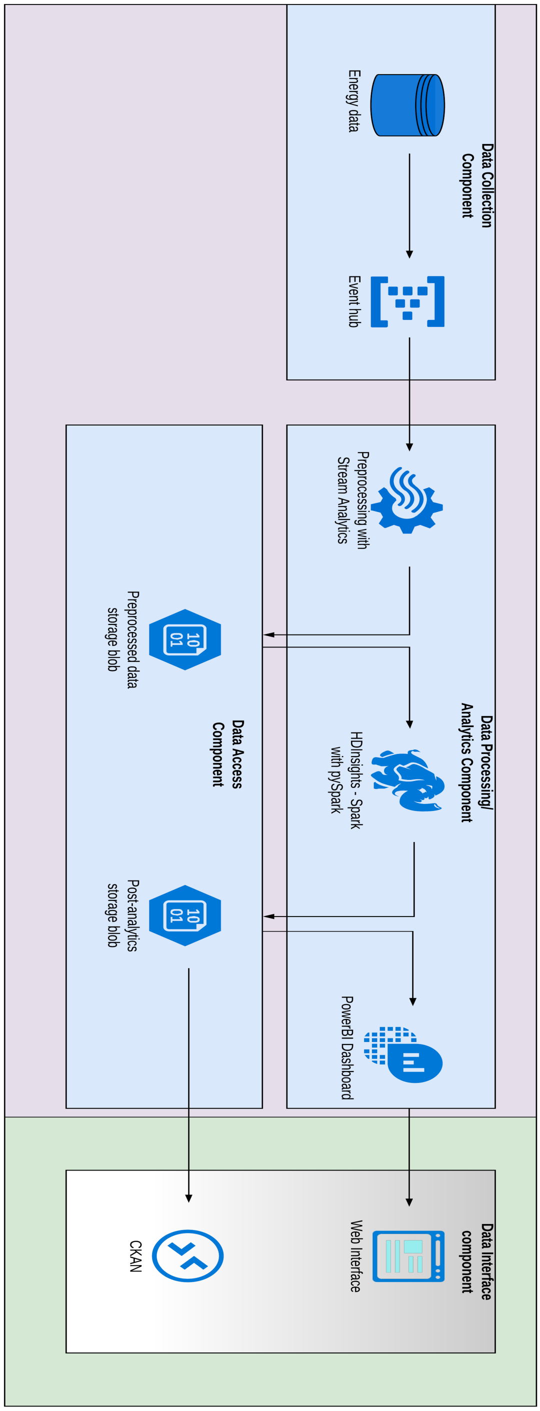


Figure 7.1: Overview of the components and services used in the prototype, and how they relate to the components of the original architecture.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

This thesis had the objective of continuing and adapting the work from the specialisation project, designing an architecture for smart city ICT systems, focusing on zero-emission neighbourhoods. It goes on to implement that architecture in a zero-emission neighbourhood context by means of a data management and Key Performance Indicator monitoring system for the FME ZEN research centre, using data from a precursor project (ZEB) to illustrate how a system based on the proposed architecture may look.

The data management and KPI monitoring system prototype is built using popular tools and technologies, and is suitable for further development and use in ZEN pilot projects. The description of the implementation in chapter 5 makes it reproducible, and it can be used with multiple other data sources, visualisations, and transformations to support the monitoring of more KPIs.

The projects research method is categorised as a *design science research* project, and as such has a knowledge contribution. The knowledge contribution of this project is the design of the architecture, the applicability and suitability of the architecture in the zero-emission neighbourhood domain with the technologies and tools used in the prototype.

8.2 Research Questions

RQ1: **What are the key components of a data management architecture for smart cities and zero-emission neighbourhoods?**

The background research and design of the architecture concludes that in general, there are a set of components that are typically present in some form or another in data management ICT architectures for smart cities: Data collection, data access, data processing/analytics, and data dissemination. While there are many different ways of designing and organising architectures, these components fit in the architecture in one way or another.

Data collection is the gathering of any data in the city (sensor data, reports, social media feeds, etc.). Data access consists of the access and storage of internal data, and management of access to data sources external to the system. Data processing refers to the transformation of the collected data to gain more information. Data dissemination is any way the data is made available to stakeholders, either by providing datasets for further use, or through data viewing interfaces like in the created prototype.

RQ2: **How well does the architecture translate into implementation in the domain?**

With the Azure services and tools used, the architectural components translate directly to either one or a set of technologies and tools.

- **Data collection** translates to the Event Hub in Azure, connecting sensors or sensor networks to the rest of the Azure environment

- **Data access** is the Azure storage blobs in the prototype, storing, hosting and giving developers access to the datasets.

- **Data processing/analytics** translates to the three different processing services used in the prototype. The first is the preprocessing of incoming streaming data. The second is the Spark processing, transforming preprocessed data to KPI values. The final tool relating to the processing component is PowerBI where the visualisations are created.

- **Data dissemination** in the prototype consists of the two interfaces that deliver the information to potential users. The CKAN open data portal (The prototypes open data interface) allows for viewing and downloading of the processed datasets. The web interface is the restricted data interface meant for decision-makers, where various visualisations are displayed, facilitating the monitoring of the KPIs for the project.

No specific tools or platform had any impact on how the original architecture was designed, and when keeping that in mind, the translation between the architecture and the actual modules and services in the implementation is effective and understandable.

RQ2.1: **Which technologies and tools are suited for implementing the architecture?**

The technological choices made for the prototype were developed through the background and state-of-the-art research. CKAN is a popular choice for creating an open data portal, and as mentioned in chapter 5, used by entities with large amounts of open data sets, like the US and UK governments. Based on these observations, CKAN was deemed a good choice for an open data portal. For use in the prototype, CKAN provides everything that is expected by an open data dissemination interface as described and envisioned in the architecture by accommodating quick access to the data produced by the prototype.

Cloud services are a major part of the prototype. As development continued, the cloud service provider became more important to the prototype than what it was supposed to in the original architecture, with the data collection through the Event Hub relying on the cloud computing infrastructure, as opposed to just the data access and data processing which was intended in the architecture. The major cloud providers (Microsoft, Google, and Amazon) provide many of the same services, and Azure was chosen after a meeting with Jonas Wäfler, a developer at Powel who are using Azure for their smart grid project. The services provided by Azure were well suited for the implementation.

For such a relatively small dataset, the use of a Spark cluster was somewhat unnecessary. Spark is intended for much larger datasets than what was used in the prototype. However, Spark did work well with the data and while it may not be necessary at this scale, having Spark as the standard makes the implementation scalable for larger amounts of data.

RQ3: **Does the implementation contribute to further development of a data management system for ZEN?**

The implementation of this prototype shows that a data management and KPI monitoring system can be built using the architecture and tools in this thesis. It may not have all the desired functionality and be ready to work in large scale projects just yet, but it represents a robust, fundamental starting point for further research and development of this system. Researchers and developers working on the ICT system for ZEN can look at a functional architecture and prototype as a basis. Whether or not the researchers and developers decide to use the work, the thesis represents one way of creating that system.

8.3 Future Work

The work done in the thesis illustrates the design and creation of a smart city ICT architecture and its applicability for adaptation to a zero-emission neighbourhood data management system. Further work that expands on the work can be done in several different areas to continue the development of a the ZEN data management and KPI monitoring system.

Future work can focus on satisfying the requirements that were not prioritised in the implementation. The two requirements that were not fulfilled were: creating a robust, unified API for third party developers, and creating a service providing logging and usage statistics for the system . Azure provides services for facilitating both these requirements. Since most of the individual modules have their own APIs, fulfilling this requirement will be creating an unified API tying the individual service APIs together. This can be done either with Microsofts own API management service, or an API can be created outside of the Azure environment that calls the individual APIs in Azure and CKAN. This should allow external developers to only have to deal with a single point of entry into the system. Microsoft provides a activity logging for the use of their cloud services. In order to provide usage statistics for the entire system, a logging service for the web interface should also be developed.

A next step for the prototype can be to put the system in a “live” setting. A live setting would be putting the system in an environment where streaming sensor data from a current pilot project is used, rather than a pre-collected dataset, as was used in the prototype. In the future, when adequate data collection is available in the ZEN pilot projects, implementing the prototype to manage data for these projects would be a good test to see if the architecture and system is sufficient and suitable for the required work, and which improvements can be made. Similarly, using the prototype in a more heterogenous data environment would be useful in determining how to best handle multiple data sources. When more data sources become available, more transformations and visualisations of KPIs can also be created. There are two main expansions of the prototype. One expansion is to scale out from building scale to neighbourhood scale, collecting the data from multiple buildings and infrastructure. The other expansion is to add more variety in the data and adding data sources. Both expansions make it possible to create more and more complex visualisations in order to make the web interface more complete as a KPI monitoring platform, allowing for more of the KPIs from table 2.1 to be monitored.

Integrating with the ZEB tool dashboard [50] is a potential extension of the ZEN data management and KPI monitoring system. The ZEB tool focuses on determining the GHG emissions of different choices in building materials, and can either act as an extension of the web interface, or as a standalone application using the system as a service for its data management. The combination of systems can increase their value to ZEN researchers by providing a broader set of tools to help decision-making in the projects.

Future work can also be done on the web interface. Improving the user interface is important, especially as more visualisations become available and potential decision-makers need to use the service. Access control should also be added to the interface if it is intended for use with data not available in the public portal.

Bibliography

[1] Dirk Ahlers and John Krogstie. *ZEN Data Management and Monitoring: Requirements and Architecture*. 2018.

[2] Eiman Al Nuaimi et al. "Applications of big data to smart cities". In: *Journal of Internet Services and Applications* 6 (2015), p. 25. DOI: 10.1186/s13174-015-0041-5.

[3] Sasan Amini, Ilias Gerostathopoulos, and Christian Prehofer. "Big Data Analytics Architecture for Real-Time Traffic Control". In: *Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. 2017, pp. 710–715. DOI: 10.1109/MTITS.2017.8005605.

[4] Michael Armbrust et al. *Above the Clouds: A Berkeley View of Cloud Computing*. Feb. 2009. URL: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html.

[5] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The Internet of Things: A survey". In: *Computer Networks* 54.15 (2010), pp. 2787–2805. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2010.05.010.

[6] Maricela-Georgiana Avram. "Advantages and Challenges of Adopting Cloud Computing from an Enterprise Perspective". In: *Procedia Technology* 12 (2014), pp. 529–534. DOI: 10.1016/j.protcy.2013.12.525.

[7] Chitra Balakrishna. "Enabling technologies for smart city services and applications". In: *Proceedings - 6th International Conference on Next Generation Mobile Applications, Services, and Technologies, NGMAST 2012*. 2012, pp. 223–227. DOI: 10.1109/NGMAST.2012.51.

[8] Peter Burnap et al. "COSMOS: Towards an integrated and scalable service for analysing social media on demand". In: *International Journal of Parallel, Emergent and Distributed Systems* 30.2 (2015), pp. 80–100. DOI: 10.1080/17445760.2014.902057.

[9] Trevor Clohessy, Thomas Acton, and Lorraine Morgan. “Smart city as a service (SCaaS): A future roadmap for e-government smart city cloud computing initiatives”. In: *Proceedings - 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, UCC 2014*. 2014, pp. 836–841. DOI: 10.1109/UCC.2014.136.

[10] Joan Daemen and Vincent Rijmen. *AES Proposal : Rijndael*. 1999.

[11] Jeffrey Dean and Sanjay Ghemawat. “MapReduce: Simplified Data Processing on Large Clusters”. In: *Communications of the ACM* 51.1 (2008), pp. 107–113. ISSN: 0001-0782. DOI: 10.1145/1327452.1327492.

[12] Selamawit Fufa et al. *A Norwegian ZEB Definition Guideline*. 2016. DOI: 10.13140/RG.2.2.26443.80162.

[13] Ammar Gharaibeh et al. “Smart Cities: A Survey on Data Management, Security and Enabling Technologies”. In: *IEEE Communications Surveys & Tutorials* 19.4 (2017), pp. 2456–2501. DOI: 10.1109/COMST.2017.2736886.

[14] Daniel Giusto et al. *The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications*. Springer, New York, NY, 2010. ISBN: 978-1-4419-1674-7. DOI: 10.1007/978-1-4419-1674-7.

[15] Francesco Goia, Luca Finocchiaro, and Arild Gustavsen. *The ZEB Living Laboratory at the Norwegian University of Science and Technology: a zero emission house for engineering and social science experiments*. 2015.

[16] Robert E. Hall et al. “The vision of a smart city”. In: *2nd International Life Extension Technology Workshop*. 2000, p. 7. URL: https://www.osti.gov/servlets/purl/773961.

[17] Ibrahim Abaker Targio Hashem et al. “The role of big data in smart city”. In: *International Journal of Information Management* 36 (2016). DOI: 10.1016/j.ijinfomgt.2016.05.002.

[18] Marijn Janssen, Ricardo Matheus, and Anneke Zuiderwijk. “Big and Open Linked Data (BOLD) to Create Smart Cities and Citizens: Insights from Smart Energy and Mobility Cases”. In: *Electronic Government. EGOV 2015*. Springer International Publishing, 2015, pp. 79–90. ISBN: 978-3-319-22479-4. DOI: 10.1007/978-3-319-22479-4_6.

[19] Jiong Jin et al. “An Information Framework for Creating a Smart City Through Internet of Things”. In: *IEEE Internet of Things Journal* 1.2 (2014), pp. 112–121. DOI: 10.1109/JIOT.2013.2296516.

[20] Steve Jones et al. “Risks and rewards of cloud computing in the UK public sector: A reflection on three Organisational case studies”. In: *Information Systems Frontiers* (2017), pp. 1–24. ISSN: 1572-9419. DOI: 10.1007/s10796-017-9756-0.

[21] Zaheer Khan, Saad Liaquat Kiani, and Kamran Soomro. “A framework for cloud-based context-aware information services for citizens in smart cities”. In: *Journal of Cloud Computing* 3.1 (2014), p. 14. DOI: 10.1186/s13677-014-0014-4.

[22] Zaheer Khan et al. “Towards cloud based big data analytics for smart future cities”. In: *Journal of Cloud Computing* 4.1 (2015), p. 11. DOI: 10.1186/s13677-015-0026-8.

[23] Rida Khatoun and Sherali Zeadally. “Smart cities, Concepts, architectures, Research opportunities”. In: *Communications of the ACM* 59.8 (2016), pp. 46–57. DOI: 10.1145/2858789.

[24] Mariam Kiran et al. “Lambda architecture for cost-effective batch and speed big data processing”. In: *Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015* (2015), pp. 2785–2792. DOI: 10.1109/BigData.2015.7364082.

[25] Rob Kitchin. “The real-time city? Big data and smart urbanism”. In: *GeoJournal* 79.1 (2014), pp. 1–14. DOI: 10.1007/s10708-013-9516-8.

[26] Andreas Koch, Sébastien Girard, and Kevin McKoen. “Towards a neighbourhood scale for low- or zero-carbon building projects”. In: *Building Research and Information* 40.4 (2012), pp. 527–537. DOI: 10.1080/09613218.2012.683241.

[27] Bill Kuechler and Stacie Petter. “Design Science Research in Information Systems”. In: *Design Science Research in Information Systems:* (2017), pp. 1–66. DOI: 10.1007/978-3-642-29863-9.

[28] Doug Laney. “3-D Data Management: Controlling Data Volume, Velocity, and Variety”. In: (Jan. 2001).

[29] Yikai Liang et al. “Exploring the determinant and influence mechanism of e-Government cloud adoption in government agencies in China”. In: *Government Information Quarterly* 34.3 (2017), pp. 481–495. ISSN: 0740-624X. DOI: 10.1016/j.giq.2017.06.002.

[30] R. Likert. “A technique for the measurement of attitudes”. In: *Archives of Psychology* 22 140 (1932), p. 55. URL: http://psycnet.apa.org/psycinfo/1933-01885-001.

[31] Marc Lotteau et al. “Critical review of life cycle assessment (LCA) for the built environment at the neighborhood scale”. In: *Building and Environment* 93.P2 (2015), pp. 165–178. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2015.06.029.

[32] Catriona Manville et al. *Mapping Smart Cities in the EU*. 2014. URL: https://www.rand.org/pubs/external%7B%5C_%7Dpublications/EP50486.html.

[33] Anne Françoise Marique and Sigrid Reiter. “A simplified framework to assess the feasibility of zero-energy at the neighbourhood/community scale”. In: *Energy and Buildings* 82 (2014), pp. 114–122. ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2014.07.006.

[34] Sean Marston et al. “Cloud computing - The business perspective”. In: *Decision Support Systems* 51.1 (2011), pp. 176–189. ISSN: 0167-9236. DOI: 10.1016/j.dss.2010.12.006.

[35] Nathan Marz and James Warren. *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*. 1st. Greenwich, CT, USA: Manning Publications Co., 2015. ISBN: 1617290343, 9781617290343.

[36] Albert Meijer and Manuel Pedro Rodríguez Bolívar. "Governing the smart city: a review of the literature on smart urban governance". In: *International Review of Administrative Sciences* 82.2 (2016), pp. 392–408. DOI: 10.1177/0020852314564308.

[37] Peter Mell and Timothy Grance. "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology". In: *Communications of the ACM* 53 (2011).

[38] Liana Müller and Thomas Berker. "Passive House at the crossroads: The past and the present of a voluntary standard that managed to bridge the energy efficiency gap". In: *Energy Policy* 60 (2013), pp. 586–593. ISSN: 0301-4215. DOI: 10.1016/j.enpol.2013.05.057.

[39] Milind Naphade et al. "Smarter cities and their innovation challenges". In: *Computer* 44.6 (2011), pp. 32–39. ISSN: 0018-9162. DOI: 10.1109/MC.2011.187.

[40] Julien Paulou et al. *Financing the energy renovation of buildings with Cohesion Policy funding*. Tech. rep. 2014, p. 102. DOI: 10.2833/18766.

[41] Dan Puiu et al. "CityPulse: Large Scale Data Analytics Framework for Smart Cities". In: *IEEE Access* 4 (2016), pp. 1086–1108. DOI: 10.1109/ACCESS.2016.2541999.

[42] Wullianallur Raghupathi and Viju Raghupathi. "Big data analytics in healthcare: promise and potential". In: *Health Information Science and Systems* 2.1 (2014), p. 3. ISSN: 2047-2501. DOI: 10.1186/2047-2501-2-3.

[43] Hans Schaffers et al. *Smart Cities as Innovation Ecosystems sustained by the Future Internet*. 2012. URL: https://hal.inria.fr/hal-00769635.

[44] Yogesh Simmhan et al. "Cloud-Based Software Platform for Big Data Analytics in Smart Grids". In: *Computing in Science & Engineering* 15.4 (2013), pp. 38–47. DOI: 10.1109/MCSE.2013.39.

[45] Suck-hyun Song, Sun Young Shin, and Jung-yup Kim. "A Study on method deploying efficient Cloud Service Framework in the Public Sector". In: *2013 15th International Conference on Advanced Communications Technology*. 2013, pp. 33–38.

[46] T.F. Stocker et al. *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. 2013.

[47] Kehua Su, Jie Li, and Hongbo Fu. "Smart city and the applications". In: *2011 International Conference on Electronics, Communications and Control, ICECC 2011 - Proceedings*. 2011, pp. 1028–1031. DOI: 10.1109/ICECC.2011.6066743.

[48] Yunchuan Sun et al. "Internet of Things and Big Data Analytics for Smart and Connected Communities". In: *IEEE Access* 4 (2016), pp. 766–773. DOI: 10.1109/ACCESS.2016.2529723.

[49] H Takeda et al. "Modeling Design Processes". In: *AI Magazine* 11.4 (1990), pp. 37–48. DOI: 10.1609/aimag.v11i4.855.

[50] Zorbey Tuncer and Aoife Houlihan. *A Dashboard Approach on The ZEB Tool ©*. 2016.

[51] UBIMOB. *UBIMOB*. URL: http://www.vestforsk.no/en/project/ubiquitous-data-driven-urban-mobility (visited on May 19, 2018).

[52] Eleni I. Vlahogianni et al. "A Real-Time Parking Prediction System for Smart Cities". In: *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations* 20.2 (2016), pp. 192–204. DOI: 10.1080/15472450.2015.1037955.

[53] Harald Taxt Walnum et al. *Preliminary toolkit for goals and KPIs*. 2017.

[54] Marianna Kjendseth Wiik et al. *A ZEN Guideline for the ZEN Pilot Areas*. Dec. 2017.

[55] Marianne Kjendseth Wiik et al. *Zero Emission Neighbourhoods in Smart Cities: Definition, Key Performance Indicators and Assessment Criteria*. 2018.

[56] Yuzhe Wu et al. "Smart city with Chinese characteristics against the background of big data: Idea, action and risk". In: *Journal of Cleaner Production* 173 (2017), pp. 60–66. DOI: 10.1016/j.jclepro.2017.01.047.

[57] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. "Wireless sensor network survey". In: *Computer Networks* 52.12 (2008), pp. 2292–2330. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2008.04.002.

[58] Matei Zaharia et al. "Spark : Cluster Computing with Working Sets". In: *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing* 10 (2010), p. 10.

[59] ZEN. *FME ZEN*. URL: https://fmezen.no (visited on May 19, 2018).

Appendix A

Appendix A contains the code for the pySpark script used for transforming the preprocessed data in the prototype to values for the different KPIs, along with description of the code as code comments.

```
import pandas as pd
import numpy as np

pd.options.display.float_format = '{:.2f}'.format

#read the json file on azure storage blob

df = spark.read.json('wasb://zebdataraw@zebstorage.blob.core.windows.net/
                              zeb30sec.json')

#Local development was done in Pandas
#pySpark has toPandas method in order to work
#with pandas dataframes while on a spark env

pandas_df = df.toPandas()

pandas_df = pandas_df.set_index('date')
pandas_df.index  = pd.to_datetime(pandas_df.index)


#separating energy consumption data and energy generation data

consume_cols = list(pandas_df)
consume_cols.remove('surfacecollector')
consume_cols.remove('solarthermal')
consume_cols.remove('inverterpvsouth')
consume_cols.remove('inverterpvnorth')

produce_cols = list(pandas_df)
produce_cols.append('surfacecollector')
produce_cols.append('solarthermal')
produce_cols.append('inverterpvsouth')
produce_cols.append('inverterpvnorth')


#summarising the energy consumption and energy generation for each row

pandas_df['RowElecConsume'] = pandas_df[consume_cols].sum(axis=1)

pandas_df['RowElecGen'] = pandas_df['surfacecollector'] +
                          pandas_df['solarthermal'] +
                          pandas_df['inverterpvsouth'] +
                          pandas_df['inverterpvnorth']


agg_df = pandas_df[['RowElecConsume','RowElecGen']].copy()
```

```
#create new data frame with the Wh difference from last value.
#diffdf is then the energy consumed or generated since last measured value

diffdf = agg_df.diff()

#cleaning the data by removing negative values because of resets
diffdf['RowElecConsume']= diffdf[diffdf.RowElecConsume >= 0]

diffdf['RowElecGen'] = diffdf['RowElecGen'][diffdf.RowElecGen >= 0]




#Defining the date ranges for the experiments in the ZEB Lab

Dayidx = pd.date_range('10-09-2015', '24-04-2016')
houridx = pd.date_range('10-09-2015', '24-04-2016', freq = 'H')

#resampling so that dataset has a row for every 30 seconds.

diffdf = diffdf.resample('30S').sum()

#Creating aggregation dataframes

#Daily results
day_results = diffdf.resample('Day').sum() / 1000


#Weekly results
weekly_results = diffdf.resample('W').sum() / 1000


#Monthly Results

monthly_results = diffdf.resample('M').sum() / 1000



#Hourly results

hourly_results = diffdf.resample('H').sum() / 1000
hourly_results.fillna(0, inplace=True)



#Dataframes for self consumption


#separating the parts of a day where energy is produced and where energy
                                      is not produced
selfconsume = hourly_results[hourly_results.RowElecGen > 1]
self_other = hourly_results[hourly_results.RowElecGen <= 1]

#Differentce between generation and consumption on an
```

```
#hourly basis for hours where energy is produced
self_dif = selfconsume['RowElecGen'] - selfconsume['RowElecConsume']


#Above calculations aggegated to montly values

self_dif_monthly = self_dif.resample('M').sum()
self_month = selfconsume.resample('M').sum()
self_other_month = self_other.resample('M').sum()

#selfconsume['Day'] = selfconsume.index.date
#selfconsume = selfconsume.groupby('Day').sum()


#creation of monthly percentage values for self-consumption and self-
                                      generation
self_consume_percent = self_month['RowElecConsume']/
                      (self_month['RowElecConsume'] + self_dif_monthly)

self_consume_percent.fillna(0, inplace=True)

self_consume_percent = self_consume_percent.where(
                      self_consume_percent < 1, 1)

self_generate_percent = self_month['RowElecGen']/
                        (self_other_month['RowElecConsume']
                        + self_month['RowElecConsume'])


self_generate_percent.fillna(0, inplace=True)
self_generate_percent = self_generate_percent.where(
                        self_generate_percent < 1, 1)

#creates new dataframe from indvidual percentage pandas series
selffile = pd.concat([self_consume_percent, self_generate_percent], axis=1
                                      )
selffile.columns = ['Self consumption', 'Self generation']
selffile.fillna(0,inplace=True)



#Creation of daily percentage values for self-consumption and self-
                                      generation
#Caps values at 1, representing 100\% self generation and self consumption
self_consume_percent_daily = selfconsume['RowElecConsume']/
                            (selfconsume['RowElecConsume'] + self_dif)

self_consume_percent_daily.fillna(0, inplace=True)

self_consume_percent_daily = self_consume_percent_daily.where(
                            self_consume_percent_daily < 1, 1)

self_generate_percent_daily = selfconsume['RowElecGen']/
                              (self_other['RowElecConsume']
                              + selfconsume['RowElecConsume'])

self_generate_percent_daily.fillna(0, inplace=True)
```

```
self_generate_percent_daily = self_generate_percent_daily.where(
                              self_generate_percent_daily < 1, 1)

#creates new dataframe from indvidual percentage pandas series
selffile_daily = pd.concat([self_consume_percent_daily,
                          self_generate_percent_daily], axis=1)
selffile_daily.columns = ['Self consumption', 'Self generation']
selffile_daily.fillna(0,inplace=True)


#Convert pandas dataframes to spark dataframes
hourly_results = sqlContext.createDataFrame(hourly_results)
day_results = sqlContext.createDataFrame(day_results)
weekly_results = sqlContext.createDataFrame(weekly_results)
monthly_results = sqlContext.createDataFrame(monthly_results)

selffile = sqlContext.createDataFrame(selffile)
selffile_daily = sqlContext.createDataFrame(selffile_daily)


#Write CSVs to storage blob in azure.
#coalesce makes spark write to a single file, in stead of multiple,
#writing to multiple is common as spark is run parallell
#on multiple machines in the cluster, outputting their own file
#using coalesce to make a single file is for
#ease of use in powerBI and for publishing the data in CKAN.

hourly_results.coalesce(1).write.csv('wasb:///hourly_results.csv')

day_results.coalesce(1).write.csv('wasb:///day_results.csv')

weekly_results.coalesce(1).write.csv('wasb:///weekly_results.csv')

monthly_results.coalesce(1).write.csv('wasb:///monthly_results.csv')


selffile.coalesce(1).write.csv('wasb:///self_monthly.csv')
selffile_daily.coalesce(1).write.csv('wasb:///self_daily.csv')
```