NOVA IMS Information Management School

MGI

Master Program in

Information Management

Designing a Scalable Real-Time Smart City Data Platform

Tiago Vieira Delgado

Thesis

presented as partial requirement for obtaining a master's degree in Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

DESIGNING A SCALABLE REAL-TIME SMART CITY DATA PLATFORM

by

Tiago Vieira Delgado

Thesis presented as a partial requirement for obtaining a master's degree in Information Management, specialization in Knowledge Management and Business Intelligence.

Advisor: Professor Vitor Duarte dos Santos

July 2023

INTEGRITY STATEMENT

I hereby declare that I have conducted this academic work with integrity. I confirm that I have not resorted to plagiarism or any other form of misuse of information or falsification of results during the preparation of this work. I also declare that I am aware of the Rules of Conduct and the Honor Code of NOVA Information Management School.

Tiago Delgado

Lisbon, July 9 2023

ABSTRACT

Smart cities face various challenges considering the current volume, velocity, and variety of data produced by the different systems that affect a city's operations. These challenges include the number of devices available from different manufacturers, the amount of detailed data produced at very high rates, and the number of different systems that need to integrate with a centralized hub for further processing. A technical challenge emerges from the difficulty in deriving value, from data produced within a city, in an effective and efficient way. In this research, an architecture for implementing a data platform capable of addressing these challenges is proposed, including data ingestion, storage, transformation, and serving. Some optimizations for increasing the platform's cost-efficiency and GDPR compliance are also suggested. Although this research acknowledges the people-related aspects of smart cities, the presented work is focused solely on the technical aspects to support such requirements. The architecture takes advantages of services provided by AWS but can be easily adapted to other popular cloud providers or hybrid-solutions. The individual components integrating the final architecture are evaluated considering their scalability and latency, both key factors in a scalable real-time smart city data platform.

KEYWORDS

Smart Cities; Big Data; Data Architecture; Data Platform; Data Engineering; Cloud Computing

SUSTAINABLE DEVELOPMENT GOALS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACL – Access Control List

API – Application Programming Interface

AWS – Amazon Web Services

B2B – Business-to-Business

CDC – Change Data Capture

DMS – Database Migration Service

DSR – Design Science Research

ETL – Extract, Transform, Load

GB/s – Gigabytes per second

GDPR – General Data Protection Regulation

ICT – Information and Communications Technology

IoT – Internet of Things

IT – Information Technology

LoRaWAN – Long Range Wide Area Network

MB/s – Megabytes per second

RTSP – Real Time Streaming Protocol

TPS – Transactions per second

UI – User Interface

VPN – Virtual Private Network

ZSTD – Zstandard

1. INTRODUCTION

1.1. BACKGROUND AND PROBLEM DEFINITION

A smart city is a concept that aims to use technology and data to enhance the quality of life for its citizens, improve efficiency and sustainability, and create new economic opportunities (Bibri, 2019). One key characteristic of a smart city is the ability to collect and analyze data from a variety of sources in real-time, such as sensors, video feeds, and social media feeds (Perwej et al., 2019). This data can then be used to optimize the city's services, such as transportation and energy management, and to provide citizens with real-time information about traffic, weather, and other conditions (Kankanhalli et al., 2017). However, there are significant technical and organizational challenges associated with building and operating data platforms for smart cities at scale. These challenges include the diverse range of data producers, the volume and complexity of data that needs to be processed and stored, and the governance and compliance of stored data.

The diverse range of data producers, including different types of sensors, systems, and applications, can make data integration and ingestion challenging. Smart cities generate large amounts of data in real-time, which must be collected, stored, and processed in a timely manner to enable real-time decision-making. This requires data platforms that are scalable and robust enough to handle the volume and complexity of the data (Adiba et al., 2015). Additionally, the data must be stored in a way that allows for easy discovery and exploration, which may require the use of tools such as data catalogs and domain-specific dashboards (Lea & Blackstock, 2014).

Data-driven decision making relies on smart cities, which means that the data must be accurate, timely, and trustworthy (Shahat Osman & Elragal, 2021). However, the data collected by smart city platforms may come from multiple sources, may be collected and processed by different organizations and may be used by different stakeholders. Therefore, it's necessary to address data governance and ownership issues. This may include developing policies and procedures for data security, data access, and data privacy, as well as determining who is responsible for maintaining and updating the data platform (Paskaleva et al., 2017). Additionally, it may require addressing ethical concerns around data collection and use, specifically privacy and bias, and building-in safeguards to mitigate them (Kitchin, 2016).

To support these data-driven initiatives, cities need scalable and robust data platforms that can collect, store, and process large amounts of data in real-time. These platforms must also be able to integrate data from different sources, such as different types of sensors and systems, and to provide data in a format that can be easily consumed by different applications and stakeholders (Puiu et al., 2016). Despite the increasing demand for real-time data platforms in smart cities, there are several technical and organizational challenges that need to be addressed to build and operate these systems at scale.

A study on scalable real-time smart city data platforms should explore various aspects of the technical and organizational challenges associated with building and operating these systems. It should investigate new methods and technologies for collecting, storing, and processing data, such as edge computing, distributed systems, machine learning, and it should explore the use of data platforms in different city domains and use cases, such as transportation and environmental monitoring.

Additionally, the study should examine the policy and governance issues around data platforms, including data security, data ownership, and data privacy.

1.2. OBJECTIVES

The aim of this research is to investigate methods and technologies to address the technical challenges of the diverse range of data producers, and the volume and complexity of data that needs to be processed and stored to enable scalable real-time data platforms for smart cities, which could be implemented in different domains such as transportation and environmental monitoring. Additionally, this research will investigate use cases for this data, allowing users to make the most out of it.

The main goal of this research is to design an architecture of a data platform capable of scaling and supporting real-time analytics.

In order to achieve this goal, the following intermediate objectives were defined:

- Conduct a comprehensive study on smart cities and the technologies required.
- Design reusable components to integrate and ingest different types of data sources.
- Design a new architecture capable of addressing the storage and processing demands.
- Validate the architecture by guaranteeing infrastructure scalability while maintaining low latency.

1.3. STUDY RELEVANCE AND IMPORTANCE

Smart cities are becoming increasingly data-driven, and the amount of data being generated is growing at a fast pace (Bibri & Krogstie, 2020). This data is coming from a variety of sources such as sensors, cameras, and different platforms, and it is being used to make important decisions about how to manage the city's resources and infrastructure (Sarker, 2022).

A scalable data platform is essential for managing and processing these potentially huge volume of data. Without scalability, the platform would be overwhelmed by the data and would not be able to provide the insights and analysis necessary for making effective decisions (Santana et al., 2017).

Moreover, real-time data processing is vital in today's urban environment, since it allows city managers to detect, prevent and react to events in a timely manner, such as monitoring traffic congestion, identifying areas prone to flooding, and optimizing the use of resources such as energy and water (Khansari et al., 2014).

A scalable real-time smart city data platform is also important for the future of a city. As the city grows and technology evolves, so too will the amount of data generated, and the platform must be able to keep up with this growth to continue providing valuable insights and analysis. This will ensure that the smart city can continue to improve the quality of life of its citizens and move towards a more sustainable development. Therefore, a scalable real-time data platform is an essential component of a smart city, allowing it to grow and adapt to changing needs and provides decision makers with the information they need to make timely and informed decisions (De Guimarães et al., 2020).

2. LITERATURE REVIEW

2.1. SMART CITIES

2.1.1. CONCEPT AND CHALLANGES

The concept of smart cities has gained significant attention in recent years, with various definitions and interpretations emerging from different sources, as well as different challenges and implications.

A study conducted with the objective of making a comprehensive proposal on the definition of what a smart city is, found it to be a geographical area where advanced technologies such as information and communications technologies (ICT), logistics, and energy production are used to enhance citizen well-being, inclusion, participation, and environmental quality. The implementation of smart cities, however, faces challenges such as the need for a shift from purely technological aspects to economic and financial issues, the weakness of the market, and the need for consolidated strategies due to the scattered nature of current projects. Technical implementation emphasizes environmental sustainability, citizen participation, and the importance of intellectual capital (Dameri, 2013).

Another study focused on the concept of a smart city defined it as a creative, sustainable area that enhances the quality of life, fosters a friendlier environment, and bolsters economic development prospects. According to this study, the intelligence of a smart city is perceived as the aggregate of various improvements in urban infrastructure, resources, and public services, with the goal of optimizing public resources, improving the quality of services offered to citizens, and reducing operational costs of public administration. The authors identify key elements of a smart city: smart economy, smart mobility, smart environment, smart people, smart living, and smart governance. These elements, while each being a wide field of research, form an integral and inseparable whole in the practical context of a smart city. The authors emphasize that the real needs of people should always be placed first, with smart technology serving as a tool for achieving goals, not an end. This perspective emphasizes the human-centric approach in the concept of smart cities, where technology serves as a facilitator rather than the focal point (Winkowska et al., 2019).

Other authors also highlight that smart cities should not be merely about technology but represent a mode of living and an ideal city to develop and rediscover. According to these authors, smart cities should be seen as evolving through the creation of application-specific tools, and without the active role of citizens, real smart cities cannot exist. Important aspects that smart cities try to improve include communication, energy, resources, information systems, monitoring, and control. The authors identify several challenges in the implementation of smart cities, including the need for a holistic approach that involves the meaning of the city, the global picture of it, and the shift of focus from technology to people. They argue that the smart city is the challenge that aims to approximate the real world and the virtual world, all in accordance with a sustainable approach (Zubizarreta et al., 2015).

Another study focused on assessing smart city projects determined smart cities to be an integrated system where human and social capital interact using technology-based solutions. The aim should be to efficiently achieve sustainable and resilient development and a high quality of life based on a multi-stakeholder, municipally based partnership. The challenges of smart cities are numerous and multifaceted, spaning across various dimensions including governance, economy, mobility,

environment, people, and living. The author emphasizes that the challenges cities face today and, in the future, must be addressed in an integrated manner. In terms of technical implementation, the author highlights the use of ICT as a key aspect of smart city projects. The author also emphasizes the importance of a comprehensive approach to city management and development, with technology acting as an enabler for the interconnection of all urban aspects. The author further notes that the development of smart strategies must be based on a multi-stakeholder municipally based partnership, and that smart city solutions must apply inclusive approaches (Monzon, 2015).

2.1.2. INTERNET OF THINGS (IOT)

The Internet of Things (IoT) has been identified as a transformative force across various sectors, with the potential to significantly impact economies worldwide. A comprehensive analysis by the McKinsey Global Institute estimates the potential economic impact of IoT applications to be as much as $11.1 trillion per year by 2025. This emphasizes the transformative potential of IoT across different sectors of the economy (Manyika et al., 2015).

The McKinsey report (Manyika et al., 2015) highlights several opportunities presented by IoT. One of the key findings is the importance of interoperability between IoT systems, which is required for 40 to 60 percent of potential value across IoT applications. The report also points out that most IoT data is not currently used, suggesting a vast untapped potential in the data generated by IoT devices. Furthermore, it is suggested that business-to-business (B2B) applications can create more value than pure consumer applications, potentially generating nearly 70 percent of the potential value enabled by IoT. The report also identifies a significant potential for IoT in developing economies, further emphasizing the global reach and impact of IoT technologies. More specifically, IoT devices can be used to improve various urban services such as waste management, water management, and public safety (Razmjoo et al., 2022). These allow the monitoring of waste levels in bins and the optimization of collection routes, while reducing waste collection costs (Misra et al., 2018). Similarly, IoT devices can also be used to monitor water levels in rivers and reservoirs, enabling faster responses to water shortages and floodings (Perumal et al., 2016).

However, the McKinsey report (Manyika et al., 2015) also identifies several challenges that need to be addressed to fully realize the potential of IoT. These include issues related to data security and privacy, the need for data-driven decision making, and the complexity of achieving interoperability among IoT systems, suggesting issues related with data integration. The report also highlights the need for continuous innovation and investment in IoT technologies to realize the full potential from IoT applications. These challenges highlight the need for further research and development in the field of Data Engineering to address these issues and unlock the full potential of this transformative technology.

2.2. DATA PLATFORMS

2.2.1. CONCEPT

In the contemporary data-driven business environment, the concept of a data platform has emerged as a critical component of organizational data management strategies. A data platform serves as the central repository and processing hub for an organization's entire data ecosystem and plays an important role in deriving valuable business insights (Moses & Gavish, 2023).

According to a review on Big Data, a data platform is a system for collecting, storing, processing, and analyzing large and complex data sets (Sagiroglu & Sinanc, 2013). This definition highlights the importance of scalability and the ability to handle complex datasets in a data platform. In a similar manner, the ASTERIX project defines a data platform as an integrated and scalable infrastructure for data processing, storage, and analysis (Behm et al., 2011). These definitions emphasize the technical aspects of a data platform and its role in enabling organizations to manage, process, and analyze large amounts of data.

Focusing on urban data platforms, at least two models have been identified: data services and score cards. The former prioritizes access to government data as an asset or input into wider services innovation, providing access to raw data sources in open publishing formats for deeper external engagement. The latter, on the other hand, is geared towards monitoring progress or performance against agreed indicators, focusing on improving the granularity and responsiveness of government reporting rather than the accessibility of underlying data itself (Barns, 2018).

These models encourage external users, such as citizens, software developers, or other businesses, to co-design government digital services. Governments, in turn, should facilitate access to data in open, machine-readable formats, fostering wider digital innovations (Barns, 2018). Thus, a data platform, in this context, is a tool or system that facilitates the collection, integration, storage, and use of data, with its specific design and function depending on the goals and needs of the city or organization using it. This perspective provides a comprehensive understanding of the role and potential of data platforms in urban governance and smart city initiatives.

2.2.2. DATA STREAMING

Data streaming is a method to process high-speed and real-time data that is continuously generated from various sources. This data is often referred to as big data and can come from a variety of sources such as search engines, social networks, computer logs, and sensor networks (Soumaya et al., 2017). The two most popular architectures for integrating real-time and batch data analytics are the lambda and kappa architectures (Lin, 2017).

The lambda architecture consists of a batch processing layer, a transient real-time processing layer, and a merging layer on top. The batch processing layer handles large volumes of data and performs computations over a long period of time. The real-time layer handles streaming data and performs computations that provide immediate insights. The merging layer combines the results from the batch and real-time layers to provide a comprehensive view of the data. On the other hand, the kappa architecture simplifies the lambda architecture by treating all data as a stream. This means that is only

necessary a stream processing engine, eliminating the need to maintain separate batch and real-time implementations and, ultimately, reducing complexity (Lin, 2017).

2.2.3. DISTRIBUTED SYSTEMS

Distributed systems are described as a collection of autonomous computing elements that appear to be a single coherent system. These elements are generally referred to as a node, which can be either a software process or a hardware device (van Steen & Tanenbaum, 2016).

However, traditional on-premises systems often require significant resources and time to add more nodes, which creates additional complexity. With that, cloud computing, a transformative paradigm shift in data storage, access, and processing, addresses several key challenges inherent to traditional on-premises systems. Its scalability and flexibility allow businesses to adjust IT resources rapidly on demand, offering a cost-efficient alternative to maintaining expensive data centers. The cloud's ubiquitous accessibility fosters global collaboration and remote work, while robust disaster recovery systems ensure business continuity (Wang et al., 2010). By outsourcing IT infrastructure, businesses can focus on their core operations, fostering innovation and accelerating product deployment. Although concerns around data privacy and provider dependency persist, many cloud providers offer advanced security measures, including encryption, access controls, and security intelligence. This paradigm shift has significant implications for businesses, particularly in terms of cost efficiency, operational flexibility, and strategic focus (Wang et al., 2010).

Data architectures that leverage distributed systems also benefit from improved reliability and fault tolerance. They can be used to provide high availability by replicating data across multiple nodes, reducing the risk of data loss and downtime in case of failure (Ahmed & Wu, 2013).

2.2.4. OBJECT STORAGE

Object storage, a concept introduced in the early 1990s, soon became increasingly adopted by the industry as a future building block for large storage systems. It represented a paradigm shift in storage technology, moving from block storage to a more abstracted system where an object store enables the creation of self-managed, shared, and secure storage for networks. This system moves lower-level functionalities, such as space management, into the storage device itself, accessed through a standard object interface. An object store presents itself as a collection of objects, each a container of storage (object data and object metadata) exposing an interface similar to a file (Factor et al., 2005).

Nowadays most object stores live in the cloud, simplifying IT operations by allowing applications to operate without the need for on-premises infrastructure. Cloud storage offers scalable storage for both structured and unstructured data, utilizing infrastructure spread across various geographical locations easily accessible via web user interfaces (UIs) and application programming interfaces (APIs). This evolution of cloud and object storage is reshaping the landscape of data storage and management in cloud computing (Odun-Ayo et al., 2018).

2.3. NETWORKING

2.3.1. LORAWAN

Long Range Wide Area Network (LoRaWAN) is a low-power wide area networking technology that offers long-range communication, enabling new types of services. It's the most adopted technology in this field and is particularly promising for outdoor IoT applications due to its simplicity in network structures and management (Adelantado et al., 2017).

It supports a raw maximum data rate of 27 KB/s and the ability to collect data from thousands of nodes deployed kilometers away, making it a promising solution for applications such as smart lighting, smart parking, and smart waste collection (Adelantado et al., 2017). However, the technology also presents challenges. For instance, its performance is dependent on various factors such as the number of end devices, selected spreading factors (SFs), and number of channels. Furthermore, the growth in popularity of technologies such as LoRaWAN introduces coexistence challenges, requiring the development of coordination mechanisms to limit interference and collisions in scenarios with coexisting gateways. Lastly, due to its data rate limitations, LoRaWAN is not suitable for applications requiring higher data rates, such as video surveillance (Adelantado et al., 2017).

2.3.2. REAL TIME STREAMING PROTOCOL (RTSP)

The Real-Time Streaming Protocol (RTSP) is a designed to send and receive audio and video data, like a live video feed or a stored video clip. It's flexible and can work with different types of data delivery mechanisms, making it a useful tool for streaming multimedia over the internet (Schulzrinne et al., 1998).

RTSP 2.0, an updated version, has made several improvements. It has new settings for better control, clearer instructions for sending and receiving messages, and new rules for maintaining reliable connections. It also provides better definitions for various commands used in the protocol. Importantly, RTSP 2.0 has improved security measures, including better authentication methods and secure connections. These updates make RTSP 2.0 more reliable and safer for streaming real-time video (Schulzrinne et al., 2016).

2.3.3. VIRTUAL PRIVATE NETWORK (VPN)

Virtual Private Networks (VPNs) are a key tool for secure network communication. They use either public or distributed network infrastructure to create a safe network environment. VPNs can be set up using various communication channels and they create a secure line of communication over a public network between two computers. There are two main types of VPNs: Site-to-Site, which connects different locations, and Client-to Site VPN, which connects individual users to a company server from anywhere (Jiang et al., 2017).

OpenVPN and Wireguard are currently two of the most popular VPN systems. Comparing the two, WireGuard outperforms OpenVPN in terms of performance, particularly on multi-core machines where it can fully leverage multi-threading. The deployment and configuration process is also

simplified with WireGuard, which installs quickly and requires only one configuration file shared among peers. In contrast, OpenVPN has a larger footprint, making the installation slower and more complex with separate configuration files needed for servers and clients. WireGuard's lean and light codebase, implemented in just over 4000 lines of code, is another advantage as it simplifies auditing and vulnerability checks, compared to OpenVPN's larger implementation of around 60000 lines of code (Mackey et al., 2020).

3. METHODOLOGY

3.1. DESIGN SCIENCE RESEARCH

The Design Science Research (DSR) methodology was chosen as the research approach for this study as it is well-suited to addressing complex design problems in the field of Information Systems. DSR emphasizes the development and evaluation of innovative artifacts, such as systems or models, as a means of advancing knowledge in the field.

In the context of Information Systems, an architecture can be seen as an artifact. It is a tangible and explicit representation of the design of a system, which can be evaluated and improved upon. The DSR methodology focuses on the development and evaluation of artifacts, which makes it an ideal choice for addressing the design problem of creating a scalable real-time smart city data platform.

The goal of using the DSR methodology is to create new knowledge by constructing a novel and relevant artifact that addresses an important research problem (Winter, 2017). This can be in the form of a new system, technology, or model, and it is expected that the artifact will be generalizable, it will have some level of maturity, and be useful for others in the field. DSR is a valuable approach for conducting research in fields that focus on design and development, as it allows for a rigorous and thorough examination of the design problem and the creation of an artifact that can be evaluated and improved upon, contributing to the advancement of the field. The following diagram illustrates the methodology's process model (Peffers et al., 2007).

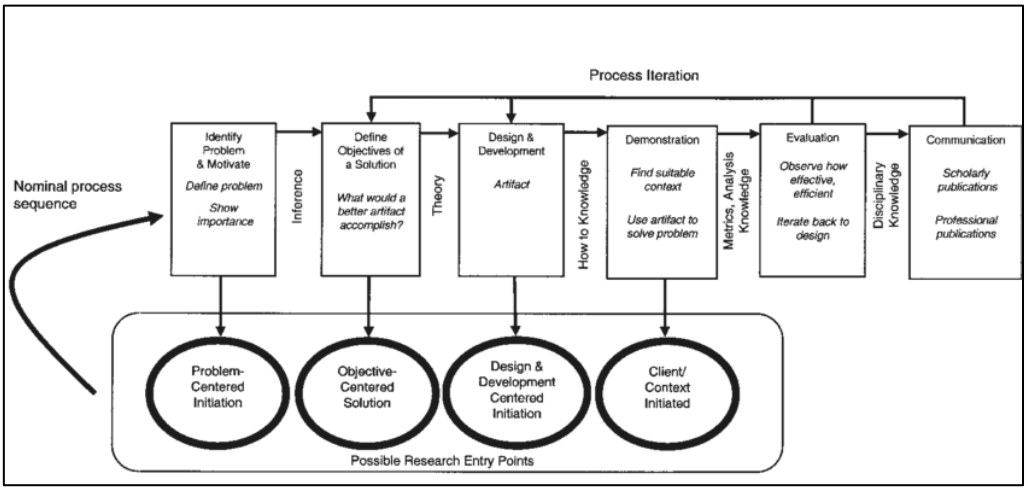

Figure 1. DSR Methodology Process Model (Peffers et al., 2007)

The DSR methodology consists of six main steps: problem identification, requirements definition, design, development and implementation, evaluation, and communication.

1. **Problem Identification:** The first step in the DSR methodology is to identify a research problem that is important and relevant to the field. This step involves understanding the context of the problem, identifying the gap in current knowledge, and defining the objectives

of the research. The problem should be well-defined and specific, and it should be clear how the research will contribute to the advancement of the field (Peffers et al., 2007).

2. **Objectives Definition:** After the problem is identified, the next step is to define the objectives for the artifact. The objectives should be based on the first step of the process (Peffers et al., 2007). This step is crucial because it ensures that the artifact will be relevant and useful for the intended purpose.

3. **Design and Development:** Once the objectives are defined, the next step is to design and develop the artifact. This step should result from the two previous steps and all research conducted in the process (Hevner et al., 2004).

4. **Demonstration:** After the design and development are completed, the next step is to demonstrate how the new artifact contributes to solving the problem that was previously identified. This step might require a simulated environment or a business case (Peffers et al., 2007).

5. **Evaluation:** After the demonstration of the new artifact, the next step is to evaluate its effectiveness and usefulness. This step usually involves testing the artifact with real scenarios and feedback to assess its performance and usability (Peffers et al., 2007).

6. **Communication:** The final step in the DSR methodology is to disseminate the results and findings to the research community. This step usually involves publishing the results in peer-reviewed journals and conference proceedings and making the artifact available for others to use and build upon. This step is crucial as it allows others to build on the research, and it allows the researcher to contribute to the knowledge of the field.

Overall, DSR is a systematic approach to addressing complex problems in the field of information systems and technology, with the goal of advancing our understanding and creating new knowledge that can be applied to improve the world around us.

3.2. RESEARCH IMPLEMENTATION

The first step in designing a scalable real-time smart city data platform is to identify the research problem. This involves understanding the concept of a smart city, identifying the gaps in current knowledge and the challenges faced by smart cities in terms of data management, and defining the objectives of the research. In this research, the objective is to design a data platform that can effectively handle and analyze the growing volume of data generated by smart cities.

After the problem is identified, the next step is to define the requirements for the platform. The requirements are based on the research problem and must be specific yet generalizable. In this research, the requirements include the ability to handle large volumes of data in real-time, the ability to support different types of data, the ability to provide insights and analysis in real-time, and the ability to scale as needed. The specifics of these requirements might change from implementation to implementation, highlighting the importance of a reusable and modular architecture.

Once the requirements are defined, the next step is to design and develop the architecture. This involves creating a detailed plan for the platform, including specifications, diagrams, and models. The

design should be based on the research problem and the objectives, and it should be consistent with the state-of-the-art in the field. In this case, the design should include a distributed architecture that can handle the data processing and storage needs, and a variety of big data technologies and tools that can support the required analysis and visualization capabilities. The development and implementation of the resulting architecture should be done on a case-by-case basis, depending on the city's characteristics and needs, such as size, budget, and data sources.

The next step is to demonstrate the effectiveness of the proposed architecture by evaluating the design choices against the initial objectives, allowing to determine if the chosen components and technologies contribute to solving the research problem. In this research, the components selected for the architecture should be analyzed based on their specifications.

After the platform's effectiveness is successfully demonstrated, the next step would be to evaluate its usefulness. This would involve testing the platform with real data and real users and getting feedback on its performance and usability. The evaluation should assess the platform's ability to solve the research question in a real scenario and specific implementation.

The final step is to communicate the results and findings to the research community. This involves the submission of this research by making the architecture available for others to use and build upon. It is important to share the learnings and knowledge acquired during the project by making it available to other researchers and practitioners to build on and leverage to create more efficient systems, as well as city managers to implement the proposed architecture.

4. PROPOSED ARCHITECTURE

In this chapter, the proposed architecture for a scalable real-time smart city data platform is presented. The architecture has been developed based on the Design Science Research (DSR) methodology and the insights gathered from the literature review presented in Chapter 2.

The DSR methodology has been fundamental in guiding the development of this architectural blueprint. It has provided a structured framework to address the problem identification, evaluation, and iterative refinement of the artifact. By building upon the knowledge obtained from the literature review, an architecture was created that is able to ingest data from a variety of sources, integrates both real-time and batch processing capabilities, and is capable of storing very large amounts of data, ultimately ensuring that the platform is adaptable to the ever-changing demands of smart cities.

The purpose of this chapter is to provide a detailed overview of the proposed architecture, highlighting its components and their interconnections. This architecture serves as a blueprint for implementing a smart city data platform that can scale with the growing needs of urban environments while delivering valuable insights for city administrators, planners, analysts, and citizens.

4.1. ASSUMPTIONS

In the process of designing the architecture for a scalable, real-time smart city data platform, several assumptions have been made. These assumptions provide a foundation for the designing of the proposed architecture and influence various design decisions. The following is a list of key assumptions:

1. **Scalability** – The architecture should be designed to handle an increasing number of IoT devices, data sources, and data volumes. The system should remain performant and responsive as the city grows, ensuring that the infrastructure can easily adapt to the evolving needs of the city and its stakeholders.
2. **Real-time data processing** – The architecture must support real-time data processing to enable timely insights, decision-making, and responses to events happening within the smart city. This is crucial for use cases that require immediate action, such as emergency response or traffic management.
3. **Batch data processing** – The architecture must also support batch data processing for more complex data analysis, machine learning, and reporting tasks that may not require real-time processing. By incorporating batch processing capabilities, the architecture is expected to serve a broader range of analytical requirements and support diverse workloads.
4. **Data storage** – The architecture should assume a mix of time-series data and other data types (structured, semi-structured, and unstructured). It should provide appropriate storage solutions to accommodate these different data formats efficiently and cost-effectively, ensuring that the platform can handle diverse data sets generated by various systems.
5. **Data security** – Security should be a critical aspect of the architecture, and the design should incorporate mechanisms to protect data at rest and in transit. The chosen cloud provider should offer robust encryption, authentication, and authorization mechanisms to ensure the privacy and integrity of the data collected and processed by the platform.

6. **API access** – The design process must assume that external applications will require access to the data stored in the platform, and it should include components to expose the data through APIs. This will enable developers to build applications and services that leverage the data collected by the smart city platform, fostering innovation and collaboration.
7. **Data visualization and reporting** – The architecture should support real-time dashboards, visualizations, and batch reports to provide actionable insights for decision-makers and citizens. It should include tools for exploring, analyzing, and sharing data, empowering stakeholders to make data-driven decisions that improve the city's operations and quality of life.
8. **Machine learning** – The architecture should incorporate machine learning capabilities, enabling predictive analytics, anomaly detection, and other data-driven insights. This will help stakeholders proactively address issues, optimize resources, and better understand patterns and trends within the city's environment.
9. **IoT communication protocols** – The architecture must support various IoT devices and data sources that use different communication protocols, such as MQTT, HTTP/HTTPS, and LoRaWAN. The chosen cloud provider should support these protocols, allowing for seamless integration and communication among diverse IoT devices within the smart city data platform.
10. **Integration and interoperability** – The architecture should be designed with integration and interoperability in mind, ensuring that various components can work together seamlessly. The chosen cloud provider should offer a range of tools, services, and APIs that facilitate the integration of different systems and enable the exchange of data between them. This will help create a cohesive and interconnected smart city platform that can evolve over time.
11. **Ease of management and deployment** – The architecture should be designed in a way that simplifies the management and deployment of the platform. The chosen cloud provider should offer tools and services that automate tasks such as scaling, monitoring, and maintaining the system, allowing the city's IT staff to focus on higher-value tasks and ensuring that the platform remains stable and performant.
12. **Flexibility and customization** – The architecture should be flexible and customizable, allowing cities to adapt the platform to their specific needs and requirements. The chosen cloud provider should offer a wide range of services and configurations that cater to diverse use cases and enable cities to tailor the platform according to their unique challenges and objectives.
13. **Choice of cloud provider** – The cloud provider should be chosen considering the following:
    - **Wide range of services** – The provider should offer a comprehensive suite of services that cater to the diverse needs of a smart city data platform, including IoT, data processing, storage, analytics, and machine learning.
    - **Scalability and performance** – The provider must be reliable and scalable, enabling the platform to grow seamlessly as the smart city expands.
    - **Security and compliance** – The provider should have a strong focus on security and compliance, offering various tools and best practices to help ensure the protection of data and applications running on its infrastructure.
    - **Cost-effectiveness** – A pay-as-you-go pricing model must be preferred, allowing for cost optimization, and reducing the need for upfront investments in infrastructure.

This enables the smart city platform to remain cost-effective and accessible, particularly for smaller cities with limited budgets.

By considering these assumptions and choosing a cloud provider that aligns with these requirements, the architecture will provide a solid foundation for implementing a scalable real-time smart city data platform that can adapt to the evolving needs of cities and their stakeholders.

4.2. ARCHITECTURAL COMPONENTS

This section presents the components for the proposed architecture for a scalable and real-time smart city data platform that can be deployed on Amazon Web Services (AWS). The design considers the needs and challenges that smart cities face, and it includes a variety of components and technologies to tackle these issues effectively. The architecture is flexible, supporting both real-time and batch processing tasks, and it's also designed to work seamlessly with different IoT devices, data sources, and data types.

The architecture consists of several connected components, each playing an important role in how the platform works. These components have been chosen and arranged to provide the best performance, scalability, and reliability, while also making it easy to manage and deploy. By focusing on how well these components work together, the proposed architecture makes sure data flows smoothly through the platform, giving city officials and other stakeholders the information they need to make smart decisions and improve the city's operations.

In this section, we will go through each component of the proposed architecture in detail, explaining what it does and why it's important for the overall system. We'll also look at how these components are connected, showing how data moves and communicates within the platform to ensure it works efficiently and can grow as needed.

By providing a clear understanding of the architecture and its components, this section aims to be a guide for creating a scalable real-time smart city data platform. By carefully considering the platform's needs and the available technology, the proposed architecture offers a strong and adaptable solution that can be customized to fit the specific needs and challenges of different smart cities.

4.2.1. DATA INGESTION

Data ingestion refers to the process of importing, transferring, loading, and processing data for immediate use or storage in a database. It involves acquiring data from multiple sources and moving it into a system where it can be accessed and analyzed. Within the context of the proposed smart city data platform, we consider three types of data sources: video cameras, external databases, and Internet of Things (IoT) devices.

4.2.1.1. VIDEO CAMERAS

The diagram below (figure 2) illustrates the process of video data ingestion from cameras into the smart city data platform. These cameras stream real-time video data, providing information that can be used for various smart city use cases, such as traffic monitoring, public safety, and event detection.

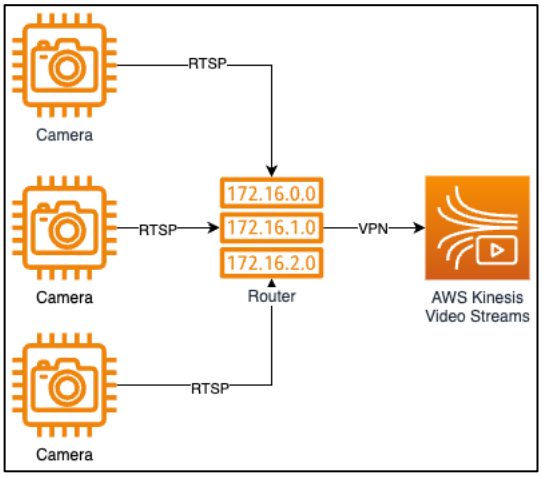

Figure 2. Video Cameras Ingestion Architecture

The ingestion of video data is facilitated by AWS Kinesis Video Streams, a fully managed service that securely ingests, processes, and stores video data, providing a secure, durable, and scalable platform that allows the extraction of meaningful insights from video data.

The video cameras connect to Kinesis Video Streams via the Real-Time Streaming Protocol (RTSP) over a secure VPN connection established between the local network, where the cameras are located, and the AWS cloud network. This ensures that the video data being transmitted is secure and only accessible to authorized users and systems.

AWS Kinesis Video Streams is capable of handling real-time processing and storage, making it a suitable solution for a wide range of smart city applications. Its capability to ingest and process video streams in real time ensures that any event captured by these cameras can be analyzed promptly and stored in the data lake.

4.2.1.2. EXTERNAL DATABASES

As seen in the diagram below (figure 3), external databases form a critical component of our data sources, providing structured and semi-structured data which is vital for the operation of the smart city data platform. The process of extracting this data and bringing it into our data platform is achieved using AWS Database Migration Service (DMS).

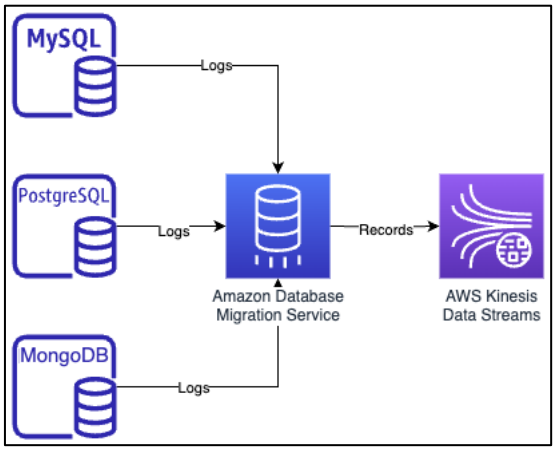

Figure 3. External Databases Ingestion Architecture

AWS DMS uses a method known as Change Data Capture (CDC) to facilitate the real-time ingestion of data. CDC is a design pattern that captures data events that modify the information in a database and saves this change information. The use of CDC techniques means that changes in the source database, such as inserts or updates to existing entries, are automatically reflected in our data platform.

This service supports various database engines, thus enhancing the architecture's flexibility in terms of data source compatibility. The DMS handles the heavy lifting involved in continuous data replication with high availability and robustness, thus making it a suitable choice for the smart city data platform's ingestion needs.

4.2.1.3. IOT DEVICES

As shown in the diagram above (figure 4), IoT devices represent another key data source in the smart city data platform, providing real-time sensor data that supports many smart city operations. The ingestion of this data is facilitated by AWS IoT Core, a managed cloud service that lets connected devices easily interact with cloud applications and other devices.

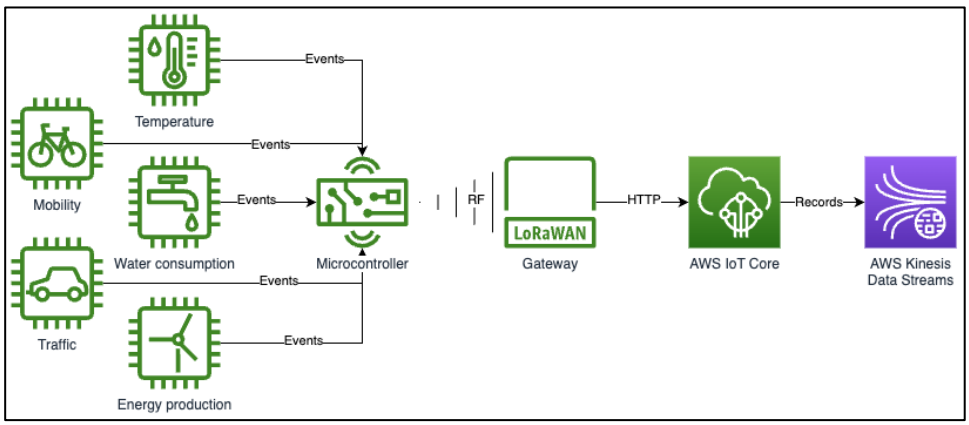

Figure 4. IoT Devices Ingestion Architecture

The IoT devices communicate securely with AWS IoT Core through LoRaWAN and HTTP/HTTPS protocols. This service is scalable and ensures reliable and secure bi-directional communication between an IoT device and the AWS cloud. Each microcontroller can support multiple IoT devices, depending on the number of GPIO pins, given that they are in the same location. The microcontrollers with the capability of communicating via LoRaWAN, benefiting from its long-range capabilities, then send the data to a LoRaWAN gateway. This gateway serves as an interface between the microcontroller and the internet, allowing a more flexible and cost-effective solution when deploying new IoT devices.

The ability of AWS IoT Core to handle millions of messages per second ensures the real-time ingestion and processing of vast amounts of data generated by the smart city's IoT devices.

4.2.2. DATA STORAGE

As shown in the diagram below (figure 5), the architecture of the smart city data platform exploits distinct storage systems based on the specific characteristics and requirements of each researched data type: real-time video streams, data from external databases, and IoT device data.

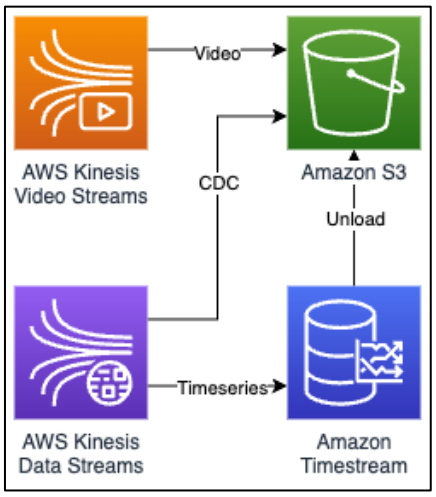

Figure 5. Data Storage Architecture

Amazon Simple Storage Service (S3), is a highly scalable and reliable distributed storage system. S3 allows the storage of virtually any type of data, from documents and images to videos and database backups. One of its main benefits is its scalability, meaning that it can store any amount of data, only paying for the used storage. Furthermore, S3's data durability and availability make it a good choice for disaster recovery, data archiving, and backups.

Amazon Timestream is a fully managed, serverless time-series database service that facilitates the storage and analysis of extremely high volumes of events per day in a more cost-effective way when compared to relational databases. It is purpose-built to manage the scale and complexity of time-

series data, like IoT sensor data. Timestream automatically scales to accommodate the volume of data ingested, making it a good scalable option.

4.2.2.1. DATA LAKE

Amazon S3 serves as the core data lake for the smart city data platform. Its role is to store different data types, such as raw video streams, external databases' data, and the processed outcomes from batch tasks.

Video feeds captured from IP cameras are transferred directly to S3 via Kinesis Video Streams. The primary objective here is to conserve the raw video streams for potential future needs. Given the significant storage demands often associated with video data, the use of S3's high durability, scalability, and cost-efficiency becomes crucial.

In addition, data generated from external databases is pulled using Change Data Capture (CDC) processes, streamed through Kinesis Data Streams, and ultimately stored in S3. The purpose here is two-fold: to keep a historical record and to have data readily available for further processing. S3's scalable and flexible nature efficiently handles the volume and variety of this data.

4.2.2.2. TIME-SERIES DATA STORAGE

IoT devices, generating time-series data, relay this data to AWS IoT Core, which is then channeled into Kinesis Data Streams. Subsequently, this data stream is written into Amazon Timestream. As a purpose-built time-series database service, Timestream is designed to handle the frequent, high-volume data production of IoT devices, offering downstream low latency analytical capabilities.

Crucially, Timestream ensures real-time consumption of the IoT device data, which is essential for real-time monitoring and immediate decision-making, factors that are often vital within a smart city's context.

For long-term storage and cost-efficiency, Timestream also regularly unloads batches of data into S3. This allows for long-term cost-efficient analysis and storage of a complete history of time-series data, ensuring no loss of crucial information over time.

4.2.3. DATA TRANSFORMATION

Data processing, a crucial component in the data management lifecycle, is where raw data is transformed into valuable insights. Within the architecture of our smart city data platform, AWS Glue plays a significant role in managing these tasks.

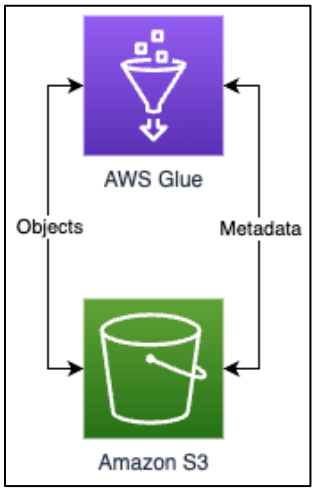

Figure 6. Data Transformation Architecture

AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy to prepare and load data for analytics. In the context of the smart city platform, it performs two fundamental roles: ETL jobs and data cataloging.

4.2.3.1. GLUE ETL JOBS

AWS Glue is leveraged to run ETL jobs that transform, clean, and enrich raw data stored in the S3 data lake, generating meaningful information from it. Data processing could include tasks such as aggregation and normalization of diverse data sources.

In the case of external databases, data from multiple sources could be aggregated to provide a holistic view of specific sectors. For instance, databases from different transportation agencies can be integrated to give an overview of the city's transport system.

When it comes to time-series data from IoT devices that has been unloaded from Timestream, it may require aggregation to facilitate more efficient storage and querying. For instance, sensor data may be aggregated hourly to facilitate long-term trend analysis, or data from multiple sensors might be combined to provide a more holistic view of a specific environment.

AWS Glue is powered by Apache Spark, an open-source, distributed computing system that facilitates data processing on large datasets. The advantage of utilizing Spark within AWS Glue is the scalability it offers. No matter the volume of data, Glue can scale its resources to process the data efficiently.

4.2.3.2. GLUE DATA CATALOG

One of the key challenges in managing a large data lake is enabling users to easily discover and access the data they need. In addition to ETL operations, AWS Glue provides a Glue Data Catalog, a persistent metadata store for all data assets. This centralized catalog assists in maintaining a consistent, organized, and searchable system of all the S3 data lake objects.

By cataloging this metadata, the Data Catalog makes data search and discovery easier, thus saving time and resources in the analytics phase. Furthermore, this catalog integrates with query engines and dashboards, enabling these services to directly and efficiently access the processed and raw data stored in S3. The implementation of an Access Control List (ACL) could also benefit from the Data Catalog by limiting access to specific groups of data assets.

4.2.4. DATA SERVING

This section of the data platform architecture relates to how the data stored in the platform is served to the end users. Three services have been employed for this purpose: Amazon SageMaker, Amazon Athena, and Amazon QuickSight.

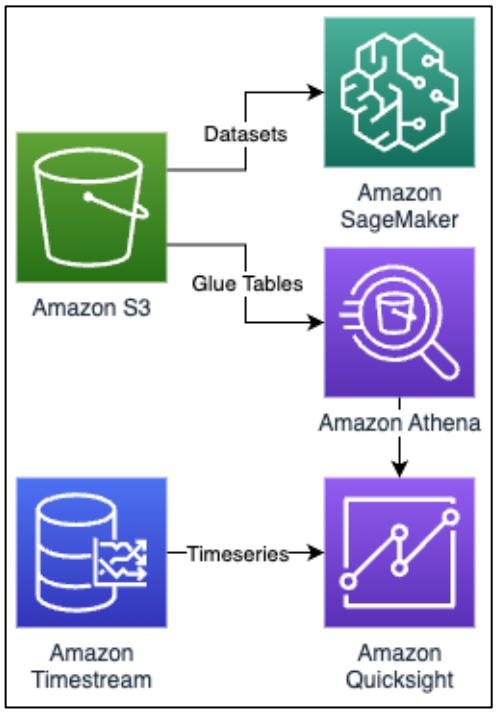

Figure 7. Data Serving Architecture

These services ensures that data is not only stored and processed effectively but also utilized to its full potential. It offers the data platform users a comprehensive toolkit for advanced analytics, machine learning, and real-time monitoring, all critical aspects of managing a data-driven smart city platform.

4.2.4.1. MACHINE LEARNING

Amazon SageMaker is a managed machine learning service that is able consume data stored in the data lake. It facilitates the development, training, and deployment of machine learning models, providing advanced predictive capabilities and analytics. This ability to create machine learning

models opens numerous possibilities, like predicting traffic patterns, detecting anomalies in public utilities, or forecasting energy demand.

4.2.4.2. AD-HOC QUERIES

Amazon Athena is an interactive query service that allows for direct querying of data stored in S3. It integrates seamlessly with the AWS Glue Data Catalog, enabling users to easily access the stored data in a table format. Consequently, users can leverage standard SQL to execute ad-hoc queries and analyze data in a table format running on top of the multiple data lake objects.

4.2.4.3. DASHBOARDS

Amazon QuickSight is a managed business intelligence service that serves as the primary tool for data visualization. It fetches data from both Amazon Athena and Amazon Timestream, offering two distinct forms of data interaction.

With data from Amazon Athena, which is structured and stored in Glue tables within S3, QuickSight facilitates the creation of detailed reports, comprehensive dashboards, and complex data visualizations. It enables users to dive deep into the historical data and gain insights, supporting informed decision-making.

Additionally, Amazon QuickSight also integrates with Amazon Timestream. Given that Timestream is a time-series database, it enables low-latency real-time consumption of data. This aspect is particularly beneficial when dealing with data generated by IoT devices. With real-time data from Timestream, QuickSight can be used for live monitoring, generating immediate insights and real-time alerts. For instance, it can monitor live data from IoT devices, such as traffic lights, weather sensors, or security devices, and provide instant alerts on critical situations.

4.3. OPTIMIZATIONS

In the process of designing a scalable real-time smart city data platform, several optimization strategies have been identified to enhance the efficiency, cost-effectiveness, compliance with regulations such as the General Data Protection Regulation (GDPR), and leverage of the stored data. These strategies are crucial in ensuring the platform's scalability and real-time capabilities while managing vast amounts of data generated in a smart city environment. This chapter will discuss four key optimization strategies: file conversion and compression, data storage partitioning, retention policy application, and the use of a data discovery tool.

4.3.1. FILE CONVERSION AND COMPRESSION

One of the primary optimization strategies involves periodically converting files to Parquet format and compressing them using Zstandard (ZSTD) or Snappy. Parquet is a columnar storage file format

optimized for big data processing frameworks like Apache Spark. The columnar nature of Parquet allows for more efficient querying and handling of big data compared to row-based files like CSV or JSON.

Compression is a critical aspect of this optimization. ZSTD and Snappy are compression algorithms that offer a balance between compression ratio and speed. ZSTD provides a higher compression ratio, which results in smaller file sizes, while Snappy offers faster compression and decompression speeds. The choice between these two would depend on the specific requirements of the data platform, such as whether storage space or processing speed is a higher priority.

4.3.2. DATA STORAGE PARTITIONING

The second optimization strategy is to store data in Amazon S3, partitioned by year, month, and day (s3://bucket_name/data_source/table_name/2021/05/13/file.json). This partitioning strategy is beneficial for several reasons. Firstly, it allows for efficient querying of data. When a query is made, only the relevant partitions need to be scanned, reducing the amount of data read and improving query performance and cost. Secondly, it provides a structured organization of data, making it easier to manage and navigate.

4.3.3. RETENTION POLICY APPLICATION

The third optimization strategy involves using the year, month, and day partitions to apply retention policies and automatically delete expired files. This strategy is crucial for two reasons. Firstly, it helps to manage storage costs. By automatically deleting old data that is no longer needed, storage space is freed up, reducing the costs associated with data storage. Secondly, it helps to comply with data protection regulations such as GDPR, which require that personal data should not be kept longer than necessary. By setting a retention policy that aligns with the requirements, the platform can ensure compliance with these regulations.

4.3.4. DATA DISCOVERY TOOL

The last optimization strategy involves connecting the Glue Data Catalog to an open-source data discovery platform, such as DataHub or OpenDataDiscovery. These tools present the metadata stored in the Glue Data Catalog in a user-friendly format, allowing end users, data scientists, and analysts to easily search for and discover data assets within the data lake. Users can search based on various parameters, such as the data source, the type of data, or the time covered by the data. A data discovery platform can also provide additional features, such as data previews, detailed descriptions of the data, and information about who else is using the data and for what purposes.

5. ARCHITECTURE VALIDATION

This chapter focuses on the validation of the proposed data architecture. Following the discussion from the previous chapters on the conceptual design and its projected performance, we will now validate how the proposed architecture succeeds in meeting the predetermined objectives: scalable data storage and low latency data processing.

5.1. SCALABLE DATA STORAGE

5.1.1. AMAZON S3

- **Bucket Limit**: By default, an AWS account can create up to 100 buckets. A bucket is essentially a container for objects stored in Amazon S3. Every object (image, video, document, etc.) is contained within a bucket. It's possible to create more than 100 buckets by requesting a service limit increase to a maximum of 1000 buckets per account. This limit does not affect performance, meaning whether a few buckets or many are used, the performance of the operations will not be affected (*Bucket Restrictions and Limitations - Amazon Simple Storage Service*, n.d.).
- **Storage Limit**: Amazon S3 does not impose a limit on the size of a bucket (the total amount of data that can be stored in a bucket) or the number of objects (files) that you can store in a bucket. This means it's theoretically possible to store an unlimited number of objects. Each individual object can range in size from 0 bytes to 5 terabytes (*Bucket Restrictions and Limitations - Amazon Simple Storage Service*, n.d.).
- **Bucket Operations**: The high availability of Amazon S3 is designed to handle a high volume of GET, PUT, LIST, and DELETE operations. These operations correspond to retrieving, uploading, listing, and deleting objects in a bucket, respectively. Each application interacting with Amazon S3 can achieve at least 3500 PUT/COPY/POST/DELETE or 5500 GET/HEAD requests per second per Amazon S3 prefix. A prefix is the representation of a folder in Amazon's distributed storage system. There are no limits to the number of prefixes in a bucket (*Best Practices Design Patterns: Optimizing Amazon S3 Performance*, n.d.).

5.1.2. AMAZON TIMESTREAM

- **Databases per account**: In Amazon Timestream, a database is a container for tables. Each AWS account can create up to 500 databases. This limit refers to the maximum number of databases that can be created within an AWS account, not the size or capacity of each database. Each database can contain multiple tables, and the data in all tables within a database contribute to the total size of the database (*Quotas - Amazon Timestream*, n.d.).
- **Tables per account**: An AWS account can create up to 50000 tables. This limit refers to the total number of tables that can be created across all databases in an AWS account. Each table can contain multiple records of time-series data (*Quotas - Amazon Timestream*, n.d.).
- **Retention periods**: Amazon Timestream provides two types of data stores: a memory store for recent data, and a magnetic store for historical data. The retention period refers to the

duration for which data is stored in these stores. For the memory store, the minimum retention period is 1 hour, and the maximum is 8766 hours (approximately 1 year). This means that data in the memory store will automatically be deleted after the configured retention period. For the magnetic store, the minimum retention period is 1 day, and the maximum is 73000 days (equivalent to 200 years). Historical data can be unloaded to S3 for better cost efficiency (*Quotas - Amazon Timestream*, n.d.).

5.2. LOW LATENCY DATA PROCESSING

5.2.1. AMAZON KINESIS DATA STREAMS

- **Data Stream Throughput**: The throughput of a data stream in Amazon Kinesis Data Streams refers to the amount of data that can be written to and read from the stream per second. By default, new data streams created with the on-demand capacity mode have 4 MB/s of write and 8 MB/s of read throughput. As the traffic increases, these data streams scale up to 200 MB/s of write and 400 MB/s read throughput. An increase to 1 GB/s write and 2 GB/s read capacity is possible by submitting a support ticket. In the provisioned mode, there is no upper limit. Maximum throughput depends on the number of shards provisioned for the stream. Each shard can support up to 1 MB/s or 1000 records per second write throughput, or up to 2 MB/s or 2000 records per second read throughput. A shard is similar to a partition within the stream (*Quotas and Limits - Amazon Kinesis Data Streams*, n.d.).
- **Immediate Availability**: In Amazon Kinesis Data Streams, as soon as a record is written to the stream, it is available to be read. This means that there is virtually no delay between when data is written to the stream by the producer and when it can be read from the stream by the consumer. This feature allows for real-time data processing (*Quotas and Limits - Amazon Kinesis Data Streams*, n.d.).
- **Throttling**: Throttling in Amazon Kinesis Data Streams refers to the limit on the number of *GetRecords* calls that can be made per second, per shard. This API call corresponds to reading records from a stream and its limit is set at 5 *GetRecords* calls per second, per shard (*Quotas and Limits - Amazon Kinesis Data Streams*, n.d.).

5.2.2. AMAZON KINESIS VIDEO STREAMS

- **Number of Streams**: In Amazon Kinesis Video Streams, a stream is a sequence of data records. Each stream is an independent entity that captures, stores, and transports data records that are continuously added. There is no hard limit on the number of streams you can create. This means that is possible to have a large number of streams operating concurrently, each capturing and transporting data independently of the others (*Kinesis Video Streams Limits*, n.d.).
- **Data Plane API Limits**: The data plane APIs in Amazon Kinesis Video Streams are the APIs used to send and receive data records from a stream. The *PutMedia* API, which is used to send data records to a stream, has a limit of 5 transactions per second (TPS), 1 connection, and a bandwidth limit of 12.5 MB/s, or 100 Mbps. The *GetMedia* API, which is used to retrieve data

records from a stream, has a limit of 5 TPS, 3 connections, and a bandwidth limit of 25 MB/s or 200 Mbps (*Kinesis Video Streams Limits*, n.d.).

- **Latency**: Latency in Amazon Kinesis Video Streams refers to the end-to-end delay from the moment a video frame is captured until it is displayed. This latency is affected by several factors, including video encoding/decoding, fragmentation delay, network latency, Kinesis Video Streams buffering, and fragment buffering by the media player. The latency can be reduced by optimizing the fragment length and the number of buffered fragments in the media player (Kinesis Video Streams Limits - Amazon Kinesis Video Streams, n.d.). In a practical demonstration, it was shown that with the Amazon Kinesis Video Stream Web Viewer, a latency of 1.4 seconds is achievable, which is reasonably low for video streaming applications (Colcott, 2022).

5.2.3. AMAZON DATABASE MIGRATION SERVICE

- **API Request Throttling**: AWS DMS supports a maximum API request quota of 100 API calls per second. This means that API requests are throttled, or temporarily blocked (*Quotas for AWS Database Migration Service*, n.d.).
- **Resource Quotas**: Some of the key resource quotas include 60 replication instances per user account, 1000 endpoints per user account, 600 tasks per user account. A replication instance is a managed infrastructure responsible for reading the changes in the source database and migrating the data. An endpoint represents a source or target database for migration. Tasks are a set of parameters for the migration (*Quotas for AWS Database Migration Service*, n.d.).

6. RESULTS

This chapter presents the outcome of this research, including the data platform architecture diagram, theoretical contributions, and practical implications. In summary, this research is expected to provide significant theoretical contributions and practical implications. The proposed architecture provides a comprehensive solution for managing and deriving value from smart city data, with potential benefits ranging from improved urban management to enhanced public safety.

6.1. DATA PLATFORM ARCHITECTURE DIAGRAM

The main outcome of this research is the proposed data platform architecture, depicted in full in the diagram below (figure 8).

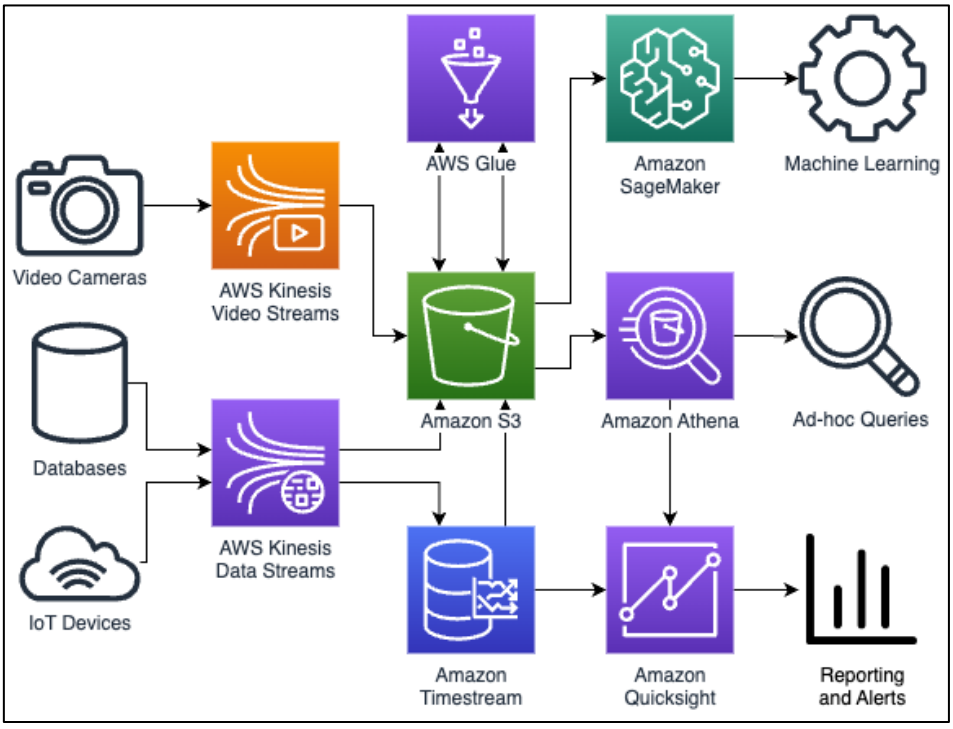

Figure 8. Data Platform Architecture

This architecture leverages a range of AWS services and open-source technologies, illustrating a comprehensive, scalable, real-time solution for managing, processing, and deriving value from a city's data assets. The architecture addresses data ingestion from various sources, distributed storage in a highly available and durable data lake, time-series storage in a low latency and scalable database, efficient data processing workflows, and data consumption by various analytics and visualization services. Additionally, multiple optimizations strategies were proposed with the objective of enhancing the overall platform in terms of efficiency, compliance, and adoption by the end users.

6.2. THEORETICAL CONTRIBUTIONS

This research contributes to the existing body of knowledge in the field of smart city data platforms, specifically in the domain of Data Engineering. By proposing an architecture that handles large volumes of data from various sources and enables both real-time and batch processing, this research addresses a significant gap in the literature. It also offers insights into the application of cloud-based services and open-source technologies in the construction of a scalable, real-time smart city data platform.

The design principles, assumptions, and architectural decisions detailed in this research provide a roadmap for further studies in this domain. As such, this research lays the groundwork for exploring more advanced topics, such as the application of machine learning in smart city contexts and the incorporation of real-time video analytics.

6.3. PRACTICAL IMPLICATIONS

The architecture proposed in this thesis has the potential to significantly impact the way smart cities manage and utilize their data. By providing a scalable and cost-effective solution for real-time data processing, the architecture enables cities to respond more quickly to emerging situations, improving urban management and public safety.

Additionally, the platform's machine learning capabilities can facilitate advanced analytics and predictive modeling, leading to more informed decision-making and improved city planning. The specific usage of the platform must be adapted to each city's requirements by deciding the relevant data sources to ingest, the valuable metrics to calculate, and the most beneficial dashboards and reports.

7. DISCUSSION

7.1. MAIN FINDINGS

The primary goal of this research revolved around the development of a scalable, robust, and real-time data platform architecture applicable to smart city scenarios. The outcome of this research presents several noteworthy findings:

1. **Scalable Data Ingestion** – The study reinforced the inherent potential of AWS services, specifically AWS Kinesis, DMS, and IoT Core, for flexible and scalable data ingestion. These services prove crucial in managing a diverse range of data sources, including video feeds, database data, and IoT devices. By integrating these services within the data platform, the architecture can cater to the expansive scale and velocity of data inherent in a smart city ecosystem.
2. **Time-Series Data Management** – A significant revelation was the effectiveness of AWS Timestream in handling time-series data. It not only allows efficient storage but also enables low latency real-time consumption of time-series data from IoT devices. The ability to manage such data in real-time is critical for smart city applications, which often rely on real-time sensor data for key decision-making processes.
3. **Data Processing and Transformation** – Another significant finding was the power of AWS Glue in enabling distributed data processing. It emerged as a potent solution for managing large-scale data transformation tasks, inherently necessary when dealing with heterogeneous data sources typical in smart city environments.
4. **Real-Time Monitoring and Alerting** – The synergistic combination of AWS Timestream and Amazon QuickSight supports real-time monitoring and alerting. This combination highlights the importance of real-time analytics and the ability to trigger alerts based on time-sensitive data, critical for efficient city management.
5. **Data Discoverability** – Finally, the study underscored the value of metadata in enhancing data discoverability. The metadata stored in the AWS Glue Data Catalog can serve as the foundation for a data discoverability platform, significantly improving the visibility and accessibility of data assets across a smart city's digital ecosystem.

7.2. RELATION TO EXISTING LITERATURE

In the broader academic and practical context, this research contributes significantly to the existing body of literature in several ways:

1. **Data Engineering** – This research aligns well with established data engineering principles and extends their application into the realm of smart city operations. This study presents a practical implementation of the theories sustained in data engineering literature, particularly in the context of scalable data storage and real-time data processing.
2. **Cloud Computing** – By leveraging AWS services, the research reinforces the benefits of cloud computing as presented in the existing literature. It substantiates the practical advantages of cloud computing in managing large-scale, diverse, and real-time smart city data.

3. **Real-Time Data Processing** – This work highlights the growing popularity on real-time data processing and analytics in both academic and real-life applications. By proposing a practical architecture that fulfills this need, this research provides a valuable reference point for future work in this area.
4. **Data Discoverability** – The research addresses a key issue frequently discussed in data management literature – the need for efficient data discoverability. By proposing a practical solution using AWS Glue Data Catalog, the study provides a valuable contribution to the ongoing discourse on effective data management strategies.

7.3. FUTURE RESEARCH

The conclusions of this research provide a foundation for several possible future investigations:

1. **Data Security and Privacy** – As this research proposes the use of cloud-based services for managing smart city data, a natural extension would be to explore ways to enhance data security and privacy within this context. This could involve the evaluation of different security models, encryption methods, or data anonymization techniques.
2. **Multi-Cloud and Hybrid Cloud Architectures** – With the proposed architecture primarily reliant on AWS, future work could explore the potential benefits and challenges of multi-cloud or hybrid cloud architectures. Such research could provide a comprehensive comparison of the performance, cost-effectiveness, and resilience of various cloud service providers.
3. **Machine Learning and Predictive Analytics** – Given the successful integration of Amazon SageMaker for machine learning, more comprehensive research could explore the potential of advanced predictive analytics in smart city management. This could involve developing state-of-the-art ML models for integrating real-time prediction within the existing architecture.
4. **Data Governance** – With data discoverability being addressed in the current research, future investigations could focus on broader data governance strategies. Such research could examine methodologies for managing data quality, data lineage, and metadata management within the context of a large-scale, data-rich environment like a smart city.

8. CONCLUSION

8.1. SUMMARY

The primary objective of this research was to design a scalable real-time data platform architecture suitable for smart city environments. This involved developing an understanding of the challenges associated with handling vast and varied data sources in real-time and leveraging cloud-based technologies to address these challenges.

This research proceeded through a careful review of the existing literature on smart cities' concept, an examination of the theoretical foundations of data engineering, along with a review of existing literature on networking technologies. Based on this acquired understanding combined with industry-specific knowledge, an architecture was designed, primarily utilizing AWS services, to create a comprehensive data platform capable of ingesting, processing, storing, and consuming data from various sources in a smart city context.

8.2. REFLECTION ON THE RESEARCH PROCESS

The research journey was an enlightening and challenging effort, with various barriers and learning opportunities encountered along the way. The study reaffirmed the importance of careful and thorough literature review, conceptual clarity, and iterative design in the process of developing a comprehensive data platform architecture.

While the focus was primarily on the use of AWS services, the study highlighted the importance of understanding the broader landscape of cloud computing and data engineering technologies. It also underscored the significance of considering practical implementation aspects, such as data security, cost-effectiveness, and integration complexities in real-world scenarios.

This research journey concluded in the development of a practical, scalable, and robust data platform architecture for smart city environments. It is hoped that this work will serve as a valuable reference for practitioners, policymakers, and researchers engaged in the domain of smart city data management.

8.3. CLOSING REMARKS

In conclusion, the study demonstrated the potential of leveraging advanced technologies and cloud-based services to address the challenges of smart city data management. However, it also pointed towards the need for further research in areas such as data security, data governance, multi-cloud architectures, and advanced analytics. As smart cities continue to evolve, it is expected that the domain of data engineering will continue to present new challenges and opportunities for innovation and exploration.

REFERENCES

Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J., & Watteyne, T. (2017). Understanding the Limits of LoRaWAN. *IEEE Communications Magazine*, *55*(9), 34–40. https://doi.org/10.1109/MCOM.2017.1600613

Adiba, M., Castrejón, J. C., Oviedo, J. A. E., Solar, G. V., & Martini, J. L. Z. (2015). Big Data Management Challenges, Approaches, Tools, and Their Limitations. *Networking For Big Data*, 43–56. https://doi.org/10.1201/B18772-8

Ahmed, W., & Wu, Y. W. (2013). A survey on reliability in distributed systems. *Journal of Computer and System Sciences*, *79*(8), 1243–1255. https://doi.org/10.1016/J.JCSS.2013.02.006

Barns, S. (2018). Smart cities and urban data platforms: Designing interfaces for smart governance. *City, Culture and Society*, *12*, 5–12. https://doi.org/10.1016/J.CCS.2017.09.006

Behm, A., Borkar, V. R., Carey, M. J., Grover, R., Li, C., Onose, N., Vernica, R., Deutsch, A., Papakonstantinou, Y., & Tsotras, V. J. (2011). ASTERIX: Towards a scalable, semistructured data platform for evolving-world models. *Distributed and Parallel Databases*, *29*(3), 185–216. https://doi.org/10.1007/S10619-011-7082-Y/METRICS

Best practices design patterns: optimizing Amazon S3 performance. (n.d.). Amazon. Retrieved July 3, 2023, from https://docs.aws.amazon.com/AmazonS3/latest/userguide/optimizing-performance.html

Bibri, S. E. (2019). On the sustainability of smart and smarter cities in the era of big data: an interdisciplinary and transdisciplinary literature review. *Journal of Big Data*, *6*(1), 1–64. https://doi.org/10.1186/S40537-019-0182-7/TABLES/3

Bibri, S. E., & Krogstie, J. (2020). The emerging data–driven Smart City and its innovative applied solutions for sustainability: the cases of London and Barcelona. *Energy Informatics*, *3*(1), 1–42. https://doi.org/10.1186/S42162-020-00108-6/FIGURES/6

Bucket restrictions and limitations - Amazon Simple Storage Service. (n.d.). Amazon; Amazon. Retrieved July 3, 2023, from https://docs.aws.amazon.com/AmazonS3/latest/userguide/BucketRestrictions.html

Colcott, D. (2022). How to reduce latency with Amazon Kinesis Video Streams – Part 1. *The Internet of Things on AWS*. https://aws.amazon.com/blogs/iot/how-to-reduce-video-latency-with-amazon-kinesis-video-streams-part-1/

Dameri, R. P. (2013). Searching for Smart City definition: a comprehensive proposal Searching for Smart City definition: a comprehensive proposal. *International Journal of Computers & Technology*, *11*(5). www.cirworld.com,

De Guimarães, J. C. F., Severo, E. A., Felix Júnior, L. A., Da Costa, W. P. L. B., & Salmoria, F. T. (2020). Governance and quality of life in smart cities: Towards sustainable development goals. *Journal of Cleaner Production*, *253*, 119926. https://doi.org/10.1016/J.JCLEPRO.2019.119926

Factor, M., Meth, K., Naor, D., Rodeh, O., & Satran, J. (2005). Object storage: The future building block for storage systems. *Local to Global Data Interoperability - Challenges and Technologies, 2005*, *2005*, 119–123. https://doi.org/10.1109/LGDI.2005.1612479

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly: Management Information Systems*, *28*(1), 75–105. https://doi.org/10.2307/25148625

Jiang, H., Wang, Q., Zhang, G., -, al, Skendzic, A., & Kovacic, B. (2017). Open source system OpenVPN in a function of Virtual Private Network. *IOP Conference Series: Materials Science and Engineering*, *200*(1), 012065. https://doi.org/10.1088/1757-899X/200/1/012065

Kankanhalli, A., Zuiderwijk, A., & Tayi, G. K. (2017). Open innovation in the public sector: A research agenda. *Government Information Quarterly*, *34*(1), 84–89. https://doi.org/10.1016/J.GIQ.2016.12.002

Khansari, N., Mostashari, A., & Mansouri, M. (2014). Impacting Sustainable Behavior and Planning in Smart City. *International Journal of Sustainable Land Use and Urban Planning*, *1*(2). https://doi.org/10.24102/IJSLUP.V1I2.365

Kinesis Video Streams Limits. (n.d.). Amazon. Retrieved July 4, 2023, from https://docs.aws.amazon.com/kinesisvideostreams/latest/dg/limits.html

Kitchin, R. (2016). The ethics of smart cities and urban science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2083). https://doi.org/10.1098/RSTA.2016.0115

Lea, R., & Blackstock, M. (2014). Smart cities: An IoT-centric approach. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/2637064.2637096

Lin, J. (2017). The Lambda and the Kappa. *IEEE Internet Computing*, *21*(5), 60–66. https://doi.org/10.1109/MIC.2017.3481351

Mackey, S., Mihov, I., Nosenko, A., Vega, F., & Cheng, Y. (2020). A Performance Comparison of WireGuard and OpenVPN. *CODASPY 2020 - Proceedings of the 10th ACM Conference on Data and Application Security and Privacy*, 162–164. https://doi.org/10.1145/3374664.3379532

Manyika, J., Chui, M., Bisson, P., Woetzel, J., Dobbs, R., Bughin, J., & Aharon, D. (2015). *The Internet of Things: Mapping the Value Beyond the Hype*.

Misra, D., Das, G., Chakrabortty, T., & Das, D. (2018). An IoT-based waste management system monitored by cloud. *Journal of Material Cycles and Waste Management*, *20*(3), 1574–1582. https://doi.org/10.1007/S10163-018-0720-Y/METRICS

Moses, B., & Gavish, L. (2023). *What Is A Data Platform? And How To Build An Awesome One*. Monte Carlo Data. https://www.montecarlodata.com/blog-what-is-a-data-platform-and-how-to-build-one/

Odun-Ayo, I., Ajayi, O., Akanle, B., & Ahuja, R. (2018). An overview of data storage in cloud computing. *Proceedings - 2017 International Conference on Next Generation Computing and Information Systems, ICNGCIS 2017*, 38–42. https://doi.org/10.1109/ICNGCIS.2017.9

Paskaleva, K., Evans, J., Martin, C., Linjordet, T., Yang, D., & Karvonen, A. (2017). Data Governance in the Sustainable Smart City. *Informatics 2017, Vol. 4, Page 41*, *4*(4), 41. https://doi.org/10.3390/INFORMATICS4040041

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Perumal, T., Sulaiman, M. N., & Leong, C. Y. (2016). Internet of Things (IoT) enabled water monitoring system. *2015 IEEE 4th Global Conference on Consumer Electronics, GCCE 2015*, 86–87. https://doi.org/10.1109/GCCE.2015.7398710

Perwej, Dr. Y., Haq, K., Parwej, F., & Mumdouh, M. (2019). The Internet of Things (IoT) and its Application Domains. *International Journal of Computer Applications*, *182*(49), 36–49. https://doi.org/10.5120/IJCA2019918763

Puiu, D., Barnaghi, P., Tonjes, R., Kumper, D., Ali, M. I., Mileo, A., Xavier Parreira, J., Fischer, M., Kolozali, S., Farajidavar, N., Gao, F., Iggena, T., Pham, T. Le, Nechifor, C. S., Puschmann, D., & Fernandes, J. (2016). CityPulse: Large Scale Data Analytics Framework for Smart Cities. *IEEE Access*, *4*, 1086–1108. https://doi.org/10.1109/ACCESS.2016.2541999

Quotas - Amazon Timestream. (n.d.). Amazon. Retrieved July 3, 2023, from https://docs.aws.amazon.com/timestream/latest/developerguide/ts-limits.html

Quotas and Limits - Amazon Kinesis Data Streams. (n.d.). Amazon. Retrieved July 4, 2023, from https://docs.aws.amazon.com/streams/latest/dev/service-sizes-and-limits.html

Quotas for AWS Database Migration Service. (n.d.). Amazon. Retrieved July 4, 2023, from https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Limits.html

Razmjoo, A., Gandomi, A., Mahlooji, M., Astiaso Garcia, D., Mirjalili, S., Rezvani, A., Ahmadzadeh, S., & Memon, S. (2022). An Investigation of the Policies and Crucial Sectors of Smart Cities Based on IoT Application. *Applied Sciences 2022, Vol. 12, Page 2672*, *12*(5), 2672. https://doi.org/10.3390/APP12052672

Sagiroglu, S., & Sinanc, D. (2013). Big data: A review. *Proceedings of the 2013 International Conference on Collaboration Technologies and Systems, CTS 2013*, 42–47. https://doi.org/10.1109/CTS.2013.6567202

Santana, E. F. Z., Chaves, A. P., Gerosa, M. A., Kon, F., & Milojicic, D. S. (2017). Software Platforms for Smart Cities. *ACM Computing Surveys (CSUR)*, *50*(6). https://doi.org/10.1145/3124391

Sarker, I. H. (2022). Smart City Data Science: Towards data-driven smart cities with open research issues. *Internet of Things*, *19*, 100528. https://doi.org/10.1016/J.IOT.2022.100528

Schulzrinne, H., Rao, A., & Lanphier, R. (1998). *Real Time Streaming Protocol (RTSP)*. https://doi.org/10.17487/RFC2326

Schulzrinne, H., Rao, A., Lanphier, R., Westerlund, M., & Stiemerling, M. (2016). *Real-Time Streaming Protocol Version 2.0*. https://doi.org/10.17487/RFC7826

Shahat Osman, A. M., & Elragal, A. (2021). Smart Cities and Big Data Analytics: A Data-Driven Decision-Making Use Case. *Smart Cities 2021, Vol. 4, Pages 286-313*, *4*(1), 286–313. https://doi.org/10.3390/SMARTCITIES4010018

Monzon, A. (2015). Smart cities concept and challenges: Bases for the assessment of smart city projects. *International Conference on Smart Cities and Green ICT Systems*. https://ieeexplore.ieee.org/abstract/document/7297938

Soumaya, O., Mohamed Amine, T., Soufiane, A., Abderrahmane, D., & Mohamed, A. (2017). Real-time Data Stream Processing Challenges and Perspectives. *International Journal of Computer Science Issues*, *14*(5). https://doi.org/10.20943/01201705.612

van Steen, M., & Tanenbaum, A. S. (2016). A brief introduction to distributed systems. *Computing*, *98*(10), 967–1009. https://doi.org/10.1007/S00607-016-0508-7/FIGURES/13

Wang, H., Jing, Q., Chen, R., He, B., Qian, Z., & Zhou, L. (2010). Distributed systems meet economics: Pricing in the cloud. *2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 10)*. https://www.researchgate.net/publication/234827796_Distributed_systems_meet_economics_Pricing_in_the_cloud

Winkowska, J., Szpilko, D., & Pejić, S. (2019). Smart city concept in the light of the literature review. *Engineering Management in Production and Services*, *11*(2). https://doi.org/10.2478/emj-2019-0012

Winter, R. (2017). Design science research in Europe. *Https://Doi.Org/10.1057/Ejis.2008.44*, *17*(5), 470–475. https://doi.org/10.1057/EJIS.2008.44

Zubizarreta, I., Seravalli, A., & Arrizabalaga, S. (2015). Smart City Concept: What It Is and What It Should Be. *Journal of Urban Planning and Development*, *142*(1), 04015005. https://doi.org/10.1061/(ASCE)UP.1943-5444.0000282