Stream data from Arduino to streaming platform to perform data analytics on dashboard

**L. ANTHONY**

 **orcid.org/0000-0000-0000-000X**

ITRI671: Literature Review for the degree *Honours:* *Bachelor of Sciences* in *Computer Science and Information Systems* at the North-West University, Potchefstroom Campus 2022

Supervisor: Prof. R. Goede

Co-supervisor: N/A

Graduation: 31 December 2022

Student number: 32969694

Table of contents

[ABSTRACT vi](#_Toc107934814)

[EXTRACT vii](#_Toc107934815)

[Chapter 1: Introduction 1](#_Toc107934816)

[1 Introduction to Research 1](#_Toc107934817)

[1.1 Project Description 1](#_Toc107934818)

[1.2 Project Background 2](#_Toc107934819)

[1.3 Goals and Objectives: Aims 3](#_Toc107934820)

[1.4 Objectives of Study 3](#_Toc107934821)

[1.5 Research Paradigm: Critical Realism (CR) 3](#_Toc107934822)

[1.6 Research Methodology: Design Science Research (DSR) 4](#_Toc107934823)

[1.7 Project Management: Scope 5](#_Toc107934824)

[1.8 Project Management: Limitations 6](#_Toc107934825)

[1.9 Project Management: Risk Analysis 7](#_Toc107934826)

[1.10 Project Plan 9](#_Toc107934827)

[1.11 Description of development platform, tools, and environments to be used 10](#_Toc107934828)

[1.12 Preliminary Chapter Division 10](#_Toc107934829)

[1.13 Executive Summary 11](#_Toc107934830)

[Chapter 2: LITERATURE REVIEW 12](#_Toc107934831)

[2 Conducting the Literature Study 12](#_Toc107934832)

[2.1 Introduction 12](#_Toc107934833)

[2.2 Research Paradigm 13](#_Toc107934834)

[2.2.1 Critical Research (CR) 13](#_Toc107934835)

[2.3 Research Methodology 17](#_Toc107934836)

[2.3.1 Design Science Research (DSR) 19](#_Toc107934837)

[2.3.2 DSR implementation of artefact 22](#_Toc107934838)

[2.4 Streaming Databases 28](#_Toc107934839)

[2.4.1 Introduction 28](#_Toc107934840)

[2.4.2 Management and processing of real-time data 29](#_Toc107934841)

[2.4.3 Applications 31](#_Toc107934842)

[2.4.4 Benefits and Drawbacks 32](#_Toc107934843)

[2.5 Arduino Sensors 34](#_Toc107934844)

[2.5.1 Introduction 34](#_Toc107934845)

[2.5.2 Applications 35](#_Toc107934846)

[2.6 Kafka Streaming Platform (Apache Kafka) 37](#_Toc107934847)

[2.6.1 Introduction 37](#_Toc107934848)

[2.6.2 Applications 38](#_Toc107934849)

[2.6.3 Streaming data from sensors to Kafka 39](#_Toc107934850)

[2.6.4 Processing real-time data and queries 41](#_Toc107934851)

[2.7 Dashboards: Tableau Desktop 42](#_Toc107934852)

[2.7.1 Introduction 42](#_Toc107934853)

[2.7.2 Applications 43](#_Toc107934854)

[2.7.3 Transferring data from Kafka to dashboard environment 45](#_Toc107934855)

[2.7.4 Data visualization techniques/Data analysis 47](#_Toc107934856)

[2.8 Chapter Summary 50](#_Toc107934857)

[Chapter 3: Name of chapter (Chapter) 52](#_Toc107934858)

[3 Heading 1 won’t print. Don’t delete – doing so will lead to incorrect numbering. 52](#_Toc107934859)

[3.1 Heading 2 52](#_Toc107934860)

[3.1.1 Heading 3 52](#_Toc107934861)

[3.2 Heading 5 52](#_Toc107934862)

[3.2.1.1 Heading 6 52](#_Toc107934863)

[3.2.1.1.1 Heading 5 52](#_Toc107934864)

[Chapter 4: Name of chapter (Chapter) 53](#_Toc107934865)

[4 Heading 1 won’t print. Don’t delete – doing so will lead to incorrect numbering. 53](#_Toc107934866)

[4.1 Heading 2 53](#_Toc107934867)

[4.1.1 Heading 3 53](#_Toc107934868)

[4.1.1.1 Heading 4 53](#_Toc107934869)

[4.1.1.1.1 Heading 5 53](#_Toc107934870)

[Bibliography 54](#_Toc107934871)

[Annexures (TOC\_Heading) 57](#_Toc107934872)

List of Tables

[Table 1: Summary of Methodological Principles for Conducting CR-based Case Study Research 4](#_Toc107934873)

[Table 2: Summary of Methodological Principles for Conducting CR-based Case Study Research 15](#_Toc107934874)

[Table 3: DBMS and DSMS comparisons 33](#_Toc107934875)

[Table 4: Sensors based on their detection properties 35](#_Toc107934876)

[Table 5: Summary of Arduino Application areas 37](#_Toc107934877)

[Table 6: Dashboard purposes and features 45](#_Toc107934878)

List of Figures

[Figure 1: Information Systems Research Framework (Hevner et al., 2004:80) 5](#_Toc107934879)

[Figure 2: Project Plan 9](#_Toc107934880)

[Figure 3: Key Relationships between Ontological, Epistemological and Methodological Principles (Wynn & Williams, 2008:9) 15](#_Toc107934881)

[Figure 4: A taxonomy of information systems research approaches (Dobson, 2001:203) 16](#_Toc107934882)

[Figure 5: Information Systems Research Framework (Hevner et al., 2004:80) 18](#_Toc107934883)

[Figure 6: Design Evaluation Methods (Hevner et al., 2004:86) 18](#_Toc107934884)

[Figure 7: Design Science Research Methodology Process Model (Peffers et al., 2007:54) 20](#_Toc107934885)

[Figure 8: Architecture of Continuous Query Processing System (Jiang & Chakravarthy, 2006) 30](#_Toc107934886)

[Figure 9: Abstract reference architecture for a data stream management system (Golab & Ozsu, 2003:3) 30](#_Toc107934887)

[Figure 10: Big Data Challenges (Gürcan & Berigel, 2018:4) 34](#_Toc107934888)

[Figure 11: Kafka in a streaming system 40](#_Toc107934889)

[Figure 12: A hybrid architecture supporting Kafka and Tableau integration (Sousa *et al.*, 2021:15) 47](#_Toc107934890)

[Figure 13: A typical Tableau Dashboard (Akhtar *et al.*, 2020:37) 50](#_Toc107934891)

ABSTRACT

Streaming databases are data repositories that works with real-time data, with the goal in mind to enrich, process, and store data streams. Data streams are streams or flows of data generated uninterruptedly from a variety of data sources. Without the need to first requiring access to all of the data, data streams are processed consecutively using stream processing methods. The analysis of real-time data is made possible through the use of streaming databases. Data streams are instantly processed, and the results of related registered queries are updated real-time. This is one of the most important capabilities of streaming databases: the analysis of real-time data and measuring how the data evolved over time, unlike a traditional RDBMS. A variety of database categories exists that can manage data streams in real-time, such as NoSQL databases, NewSQL databases, time-series databases, in-memory databases, or in-memory data grids.

The main objective of this study is to design and implement a functional IoT system via the use of a streaming database, using Design Science Research principles. The design and implementation of the IoT system should be carefully documented to contribute to a knowledge base on how to efficiently design and implement a practical IoT system via using a streaming database. The generation and analysis of real-time data is a key objective of the study. The study will discuss and analyse the use of efficient tools and platforms to develop a streaming database as well as the transfer/processing of real-time data.

The IoT system contains three important components: an Arduino Uno R3 with sensors, used to receive real-time data streams, the Kafka streaming platform (streaming database) and finally the dashboard environment (Tableau Desktop) used to visualize the real-time data and to perform data analytics. The research question at hand is how to develop an efficient/effective IoT system using the tree abovementioned components. It is also important to conduct research and discuss how the flow of data is managed from the sensors and pipelining it through the streaming database to the dashboard environment. A thorough study will be conducted on Arduino sensors, the Kafka streaming platform and Tableau Desktop (dashboard). Results from the literature study and design/implementation of the IoT system will be visually represented. The study must contribute as a knowledge base for future generations of developers to easily design and implement smart systems via using a streaming database that eases the handling of real-time data.

**Keywords: Streaming database, real-time data, Kafka, Tableau Desktop, dashboards, IoT, data analysis, Design Science Research (DSR), Critical Realism (CR)**

EXTRACT

Chapter 1: Introduction

# Introduction to Research

This study is done to contribute to a knowledge base on how to design and implement an IoT system. As mentioned in the background of the study, few IoT systems gets deployed real-time. The relevant question is how to stream data from sensory input via using an Arduino to the Kafka streaming platform and convert the data into a pivot-table format. How should one then transfer the formatted data to a dashboard (like Tableau Desktop) and perform data analytics? To put the problem another way: What tools/developer environment should be researched and used in order to design and implement an effective/efficient IoT System? The study should contribute to the effort of enabling more IoT systems to get deployed in the future, smart and automated systems should eventually become a daily occurrence in the future generation’s everyday lives, such as breathing or walking. This study aims to find a unique perspective on how to implement and design an IoT system, contributing to gaps currently in this research area. This research should enable junior developers and individuals with little to no experience in the IoT or software development field to quickly setup and run an IoT system. The study aims to provide overly simplistic documentation as well as tools and architecture to design and implement an IoT system. As such anyone should be able to build and deploy their very own IoT system with just a few instructions and guidelines. The study should provide documentation on how to resolve errors within an IoT system or how to find a different approach to resolve an obstacle when trying to design and implement an IoT system.

## Project Description

The purpose of this study is to obtain knowledge on how to read sensory input from an Arduino R3 Uno as a stream and send it to a streaming platform (Kafka is preferable in this study). To undertake data analysis, the data must be translated into a pivot-table format and displayed on a dashboard (preferably Desktop Tableau) after it has been stored on the platform. Based on the results of the tableau, one must be able to make certain business decisions as a result of the analysis. Thus, the study focuses on the design and implementation of an artefact that can receive sensory input, store data received as streams and display the results in a tableau in order to perform analytics. Experiences and observations should be documented and used as recommendations for designing and implementing an IoT system, as well as providing insights into how to approach difficulties from a different perspective. Sections that will follow is a background of the study, which provides an overview of the existing literature that concerns this study, as well as a description and goals of this study. The problem statement provides a more in-depth overview of the study and what problems this study aims to solve. The paradigmatic perspective mentions the research paradigm applicable to this study and its essence. Methods within the paradigmatic perspective will be discussed that helps to conduct research. Aims and objectives of the study will be discussed: the main goals of the study and tasks that flows from the main goals. The research methodology implemented in the study will be discussed, wherein the artefact and lifecycle of the project is a key aspect. Methods on how to sample data and conduct research will be discussed. The approach to project management and the project plan involves the scope, limitations and risks associated with the study. The project plan will be a Gantt Chart that lists activities/tasks that needs to be completed during the entire lifecycle of the project. A provisional chapter division will be provided that provides the framework for the literature study. Finally, an executive summary will be provided that provides an overall “conclusion” of the study.

## Project Background

In recent years, the Internet of Things (IoT) concept has been widely adopted within a variety of disciplines, ranging from the industry to smart cities (Calvillo-Arbizu *et al.*, 2021:1). In the health sector, IoT enables new healthcare delivery scenarios as well as the collecting and analysis of real-time health data via sensors so that better judgments can be made on how to solve health issues (Calvillo-Arbizu *et al.*, 2021:1). This sector, however, is complex and involves a number of technological obstacles (Calvillo-Arbizu *et al.*, 2021:1). Despite the abundant literature on the subject, IoT applications in healthcare just scratch the surface of the sector's needs (Calvillo-Arbizu *et al.*, 2021:1). The majority of IoT technologies are multipurpose and must be customized to meet the specific needs of each industry (Calvillo-Arbizu *et al.*, 2021:1). These requirements are frequently overlooked in health IoT-driven solutions, and there happens to be no systematic evaluation of the burgeoning literature from the standpoint of health IoT-driven solutions (Calvillo-Arbizu *et al.*, 2021:1).

As indicated by Calvillo-Arbizu *et al.* (2021:4), extensive research and material does exist on IoT topics, specifically the healthcare sector. A total of 12108 items were found after searching databases (Calvillo-Arbizu *et al.*, 2021:4). After the process of removing duplicates (3 602 papers), the remaining literature were sorted by title (5 934 papers) and abstract (2 376 papers were excluded) (Calvillo-Arbizu *et al.*, 2021:4). The contents of the resulting papers were examined in order to identify literature studies that were beyond the scope of the sector (Calvillo-Arbizu *et al.*, 2021:4). Out of all the resulting articles, 86 were found to be eligible for review (Calvillo-Arbizu *et al.*, 2021:4). Although extensive research on IoT topics exists, literature studies on the design and implementation of IoT and streamed data applications are limited and the complexity of IoT systems restricts potential applications, and rarely current IoT solutions gets deployed real-time (Calvillo-Arbizu *et al.*, 2021:8).

Based on the abovementioned literature study, this study examines how to receive sensory input from an Arduino R3 Uno and read the input as a stream to the Kafka streaming platform and identify and demonstrate an area of application. Gaps in the healthcare industry might have to be researched, and the artefact should contribute to fill the gap of that sector. After the data streams are contained on the platform, the data needs to be converted into a pivot-table format and displayed in a tableau (dashboard) in order to perform data analysis. From the analysis one has to be able to make some business decisions based on the results from the tableau. Design Science will be applied to this study, which means errors will be documented as well as the progress made in order to provide a meaningful perspective of the knowledge gained by designing and building the artefact. The overall objective of this study is to design and develop an artefact that implements a streaming database/platform via sensory input, to provide meaningful information from the data received.

It is rather apparent that a different approach will have to be followed and different literature studies might have to be examined to design and implement an IoT data streaming and tableau analysis-based application in order to build a knowledge base on how to successfully receive sensory input from an Arduino R3 Uno and read the input as a stream to the Kafka streaming platform.

## Goals and Objectives: Aims

To build a knowledge base on how to design and implement a solution to receive sensory input from an Arduino R3 Uno and read the input as a stream to the Kafka streaming platform. The knowledge base must serve as learning material for future generations on how to build a functional IoT system

## Objectives of Study

* To build an Arduino Uno R3 that are able to receive sensory input (via temperature/moisture or motion scanners for example)
* To conduct a literature study and write a thesis that substantiates the research (design and implementation of IoT system)
* To identify and demonstrate an area of application for the IoT System
* To document findings and progress on designing and implementing the IoT System
* To document errors and obstacles while trying to design and implement the IoT System

## Research Paradigm: Critical Realism (CR)

Ontology describes the nature of objects being studied, nature and qualities of a variety of entities that exist, as well as whether reality exists objectively or subjectively with regards to people observing the reality (Wynn & Williams, 2008:4). Critical realism is based on three essential beliefs: objective reality, stratified ontology, and an open systems perspective (Wynn & Williams, 2008:4). Epistemology focuses on acceptable truths/beliefs via specifying criteria and defining the process of how to perform assessment on assertions compiled by researchers (Wynn & Williams, 2008:7). The way knowledge claims are presented, how they are evaluated for truth or validity, and how they are weighed against existing knowledge are all determined by one's epistemological beliefs (Wynn & Williams, 2008:7). CR aims to explain reality as objective through a researcher’s observations/interpretations/examinations (Wynn & Williams, 2008:7). Knowledge assertions created by researchers are based on identifying components of reality that must exist in order for certain phenomena within a system to occur (Wynn & Williams, 2008:7). CR aims to explain rather than to predict or comprehend complex phenomena (Wynn & Williams, 2008:7). Both observability and unobservability within systems plays an important role in CR research (Wynn & Williams, 2008:7).

The table below gives a summary of the methodologies typically implemented in a CR study:

Table 1: Summary of Methodological Principles for Conducting CR-based Case Study Research

Table

Description automatically generated

Source: (Wynn & Williams, 2008:10)

## Research Methodology: Design Science Research (DSR)

According to Peffers *et al.* (2007), Design Science is a methodology or paradigm which can be used to develop and assess IT artefacts, the artefacts themselves aimed at resolving identified organizational issues. DSR entails a thorough process of designing artefacts to solve observable problems, as well as contributing to research, evaluating the planned designs, and communicating the results from observance and experiences to relevant audiences (Peffers *et al.*, 2007:49). Typical artefacts produced using the DSR methodology includes instantiations, methods, models and constructs, to name a few (Peffers *et al.*, 2007:49). To sum up: any product created with an incorporated solution in mind to an extensively researched topic is included in this description (Peffers *et al.*, 2007:49). The methodology that will be used to conduct this study is Design Science Research that falls under the Critical Realism paradigm. The goal of this research is to learn how to design and implement a real-time IoT system. Another important aspect of this study is to contribute to a knowledge base on how to design and implement an effective IoT system that can be deployed real-time. Experiences and observations should be documented, and they should serve as guidelines on how to successfully design and setup an IoT system, while also providing insights on how to tackle problems by looking at a different angle. The following figure demonstrates the DSR framework:

Diagram

Description automatically generated

Figure 1: Information Systems Research Framework (Hevner et al., 2004:80)

To conclude on this section: the study will be more empirical by nature and will contribute to a knowledge base on how to design and implement an IoT system, while documenting the experiences in designing and implementing the solution as well as to document obstacles or errors that occurred during the design/implementation of the artefact. The documentation on the obstacles/errors should be able to guide the user to quickly setup their own IoT system, and when the user does have a problem to implement the system according to guidelines, they should be able to get an alternative perspective on solving a problem based on the documentation.

## Project Management: Scope

* To identify and demonstrate a use case for an Internet of Things real-time system.
* Conduct a literature review and produce a thesis that backs up findings (design and implementation of IoT system)
* To develop a knowledge foundation on how to design and implement a solution that receives sensory input from an Arduino R3 Uno and reads it as a stream into the Kafka streaming platform. Future generations must be able to learn how to design a functional IoT system from the knowledge base.
* To build an Arduino Uno R3 that can accept sensory information (for example, via temperature/moisture sensors or motion scanners)
* To identify the right sensors for the identified use-case
* To create the best circuitry for the Arduino.
* To gain a better understanding of the Kafka streaming platform (features and implementations)
* To send data from sensors to the Kafka streaming platform.
* Using Tableau Desktop, convert data stored on Kafka into a pivot-table format and move the data to a dashboard.
* To explore the Tableau Desktop environment and techniques for data visualisation and analytics
* To perform analytics on the data on the dashboard via data visualisation techniques. Based on the results of the analysis, one should be able to make an informed business decision
* To document findings and progress on designing and implementing the IoT System
* To document errors and obstacles while trying to design and implement the IoT System
* To write a program/script that generates enough data that can be streamed and provides meaning when analytics is performed

## Project Management: Limitations

According to Watt (2014:14), limitations and constraints of projects typically include the following:

* **Cost**

A budget that has been approved, which covers all the necessary completion costs (Watt, 2014:14). Because many initiatives receive money or subsidies with rigorous contract terms,, project managers must strike a balance between not running out of funds and not underspending inside companies (Watt, 2014:14). Budget preparations that aren't well-executed can lead to a number 99 scramble to spend the funds available (Watt, 2014:14). Cost is ultimately a limiting constraint for almost all projects: few projects can go beyond budget without requiring a remedial step (Watt, 2014:14).

* **Scope**

What functionality the project should provide and the end goal of the project (Watt, 2014:14). Project outcomes are explicitly defined as well as the processes that go into creating them (Watt, 2014:14). It is the project's raison d'être and goal (Watt, 2014:14). Scope limitations can cause a project to fail, and scope creeping cause extra strain on the project’s time frame.

* **Quality**

Combination of the performance criteria and stipulated standards that must be met by the project's products in order to provide the intended functionality (Watt, 2014:14). The product must deliver on its promises, solve a known problem, and give the expected benefit and value. (Watt, 2014:14). Other performance standards must be met, including maintainability, dependability and availability as well as a good conclusive result (Watt, 2014:14). The practice of evaluating overall project performance on a regular basis to verify that the project satisfies the essential quality requirements is known as quality assurance (QA). (Watt, 2014:14).

* **Risk**

External events, in most cases, will have a negative impact on your project (Watt, 2014:14). Risk is defined as how likely an event will occur as well as the foreseen impact on the project if the risk does occur (Watt, 2014:14). If a risk happens to be a foreseen possibility, and the consequences of the risk are extremely disastrous , one should classify the event as a risk and implement a proactive risk management plan (Watt, 2014:14).

* **Resources**

Any assets that are needed to complete the project's tasks (Watt, 2014:14). Usually resources consist of funds, facilities, equipment, people or any other type of asset (usually labour is not included) that is necessary to view a project activity as being successfully completed (Watt, 2014:14). One can speculate how a project can be negatively affected if inadequate resources are assigned to a project.

* **Time**

The time frame needed to complete a project (Watt, 2014:14). One of the most common project oversights is time (Watt, 2014:14). Missed deadlines and incomplete deliverables are evidence of this (Watt, 2014:14). The proper control of the schedule necessitates the precise identification of tasks to be completed, as well as accurate estimates of their durations, sequence, and allocation of personnel and other resources (Watt, 2014:14). Vacations and holidays should be considered while planning a project to include slack times (Watt, 2014:14).

## Project Management: Risk Analysis

According to (Pries-Heje *et al.*, 2014:63), there are six classes used to identify risks that can occur within project management when using a DSR methodology, and each of the classes contains a number of risks that should be considered:

1. **Business Needs**

This class includes the identification, selection, and development understanding the business needs which includes problems and requirements, This also involves problem analysis and assessing alternatives (Pries-Heje *et al.*, 2014:63). The following potential risks have been identified as a result of this activity's analysis:

* Problems that have no meaning to the stakeholder (Pries-Heje *et al.*, 2014:64).
* Struggling to grasp the problem and the context it occurs within (Pries-Heje *et al.*, 2014:64).
* Incompatible stakeholder interests (some of which may or may not come to light) (Pries-Heje *et al.*, 2014:64).

1. **Grounding**

Searching, recognizing, and understanding relevant information (recovered from the body of human knowledge that has been recorded) (Pries-Heje *et al.*, 2014:63). The following potential risks have been identified as a result of this activity's analysis:

* Incomprehension and/or failing to understand relevant existing research. Failing to comprehend the situation and excessive dependence on personal experience or imagination (Pries-Heje *et al.*, 2014:64).
* Incomprehension and/or failing to understand existing design science research used to study problem-solving technology, i.e., a lack of understanding how technology works (Pries-Heje *et al.*, 2014:64).
* Incomprehension and/or failing to understand important natural and behavioural science research that forms the base theories about understanding the essence of the problem and how to solve the problem (Pries-Heje *et al.*, 2014:65).

1. **Building and designing artefacts**

Construct design theories, including instantiations (hypothetical solutions to solve business needs or problems, theories about the solutions) (Pries-Heje *et al.*, 2014:63). (Pries-Heje *et al.*, 2014:63). The following potential risks have been identified as a result of this activity's analysis:

* Development of an uninstantiated (conjectural) solution that cannot be easily incorporated (being able to actually implement the solution) (Pries-Heje *et al.*, 2014:65).
* Development of a hypothetical solution that cannot solve the problem, i.e., the artefact cannot be implemented in real life and has a lot of socio-technical problems (Pries-Heje *et al.*, 2014:65).
* Developing a hypothetical solution that fails to solve the problem, i.e., requires excessive resource expenditures (Pries-Heje *et al.*, 2014:65).

1. **Assess design artefacts and provide evidence for design theories or expertise.**

The following potential risks have been identified as a result of this activity's analysis:

* Tacit requirements (which cannot be disclosed by definition) are not addressed while evaluating solution technology, resulting in the solution technology failing to meet those requirements (Pries-Heje *et al.*, 2014:66).
* While any or all of the relevant needs are not surfaced, they are not addressed when evaluating the solution technology, resulting in the technology failing to meet those specifications (Pries-Heje *et al.*, 2014:66).
* Misalignment of specifications to Information Systems Design Theory (ISDT) meta-requirements results in the IDST being tested and an embodiment of the meta-design being evaluated in a circumstance where none of them should be used (Pries-Heje *et al.*, 2014:66).

1. **Artefact dissemination and use**

Disseminate new artefacts, design theories, and knowledge to individuals and organizations for practical applications in order to meet business needs (Pries-Heje *et al.*, 2014:64). The following potential risks have been identified as a result of this activity's analysis:

* A solution's implementation in practice does not work at all (Pries-Heje *et al.*, 2014:67).
* Misunderstanding the solution's suitable context and constraints (Pries-Heje *et al.*, 2014:67).
* Misunderstanding how to implement the solution (Pries-Heje *et al.*, 2014:67).

1. **Knowledge additions**

Publish new design artefacts and theories, and design knowledge that has practicality in order to tackle or improve business needs and problems (Pries-Heje *et al.*, 2014:64). The following potential risks have been identified as a result of this activity's analysis:

* Impossibility of publishing or presenting research findings (Pries-Heje *et al.*, 2014:67).
* Low-importance research is published (Pries-Heje *et al.*, 2014:67).
* Publication of erroneous study findings (Pries-Heje *et al.*, 2014:67).

## Project Plan

Chart, bar chart

Description automatically generated

Figure 2: Project Plan

## Description of development platform, tools, and environments to be used

In order to develop an IoT system that can stream real-time data, an Arduino Uno R3 microcontroller will be needed as well as various type of Arduino sensors (such as temperature, PIR motion or humidity sensors) to test its data streaming capabilities. A type of streaming database is necessary in order to process and manage the real-time streaming data, in this study the Apache Kafka streaming platform will be utilized. Other tools might be needed to pipeline the streaming data from the Arduino microcontroller to Kafka. Tableau Desktop will be used as the dashboard environment to analyse and visualize data. Other tools and frameworks might be necessary to pipeline the data from Kafka to Tableau Desktop and will be researched at a later stage in this study.

## Preliminary Chapter Division

**Chapter 1: Introduction**

Overall introduction to the literature study, which also serves as the research proposal. This chapter introduces research methodologies, research paradigms as well as a description of the artefact and project plan, background and relevant works.

**Chapter 2: Literature Study**

The background of the project will be examined more attentively here, a more accurate explanation of the artefact and how it will be developed.

**Chapter 3: Development of Project and Documentation surrounding the Artefact**

The chapter contains the following sub-sections:

* 1. Description of Project
  2. The Life Cycle and the different Phases of the Project
  3. Description of the Development of the Project

**Chapter 4: Recommendations**

Based on the development of the project and documentation surrounding the artefact, recommendations will be made on how to finalize the project and answer the research question.

**Chapter 5: Reflection**

Keep the study's findings in a separate document. The research question will be addressed through the goals and objectives established in this chapter. Based on the findings, recommendations will be made.

**Chapter 6: Conclusion and final article**

The final document will be addressed and finalised here. A comprehensive document can be created by combining all of the aforementioned chapters.

## Executive Summary

The purpose of this study is to figure out how to read sensory information from an Arduino R3 Uno as a stream and send it to a streaming platform (as mentioned in the Introduction, the Kafka streaming platform is preferable). The data must be translated into a pivot-table format and displayed on a dashboard after it has been stored on the platform (ideally Desktop Tableau). Based on the tableau's conclusions, one should be able to make specific business decisions based on the analysis' results. The study focuses on the development and implementation of an artefact that can receive sensory input, store data as streams, and show the results in a tableau for analytics. Another important aspect of this study is that it should contribute to a body of knowledge on how to develop and implement a real-time IoT system. Experiences and observations should be documented and used as recommendations for designing and implementing an IoT system, as well as providing insights into how to approach difficulties from a different perspective. This study falls under the Critical Realism paradigm, and the study will also focus on the events occurring within the IoT system (artefact). Documenting events, experiences and observations is an important part of this study. This study is objective by nature, even though there are events and causes which cannot easily be explained by humans. The methodology that will be implemented to conduct this study is Design Science Research, which is empirical by nature. Field studies, experiments and simulations will have to be conducted in order to design and implement a real-time IoT system.

Chapter 2: LITERATURE REVIEW

# Conducting the Literature Study

## Introduction

This chapter introduces the literature review. Thorough research will be conducted on the research methodology and research paradigm applicable to this study, as well as a more in-depth examination of Arduino sensors, streaming databases, the Kafka streaming platform and Tableau Desktop. Each section is listed as a subheading in this chapter, and the main points of discussion of each section will be listed as further subheadings. Apart from the research methodology (DSR) and research paradigm (CR), the main points of discussion of each section will be constructed as follows: an introduction to the section, followed by the applications of each section, and finally how Design Science Research will be implemented in each section. Subheadings relevant to each section will be listed in between the main points of discussion. The objective of this chapter is to introduce the crucial components of the artefact, to examine the essence of each component to the artefact as well as standard practice design principles that should be considered when planning to design and develop the artefact. The literature study should serve as a component of the knowledge base on how to design and implement an IoT system. The pertinent question is how to stream data from sensory input to the Kafka streaming platform via an Arduino and convert the data into a pivot-table format (or any other streaming processing format for that matter). Research should be conducted on how to transfer the formatted data to a dashboard (Tableau Desktop). Data analytics in the Tableau Desktop environment must be studied, such as data visualisation techniques and data transfers from Kafka to the dashboard environment. The tools and development environments necessary to design and implement the IoT system should be investigated and utilized. This chapter should serve as the basis on which future generations of developers can rely to design and develop efficient/effective IoT systems using streaming databases to process real-time data. This research aims to fill gaps in the literature by providing a new perspective on how to implement and design an IoT system. This study should enable junior developers/individuals with little to no experience in the IoT or software development fields to quickly establish a fully functional IoT system. The study's goal is to provide overly simplistic documentation, tools, and architecture for designing and implementing an IoT system.

## Research Paradigm

### Critical Research (CR)

The paradigms of positivism and interpretivism are being challenged by critical realism (CR), which is a rival paradigm to the aforementioned (Wynn & Williams, 2008:2). CR-based research methodologies can be particularly useful for conducting research to examine complex organizational occurrences in a comprehensive way in order to establish deep contextual and causal explanations that account for the wide range of environmental and organizational elements that contribute to their occurrence (Wynn & Williams, 2008:2). CR-based research responds to current calls for systems-oriented MIS theories, which can be used to discover the processes that link a series of unpredictably occurring events and complicated interactions (Wynn & Williams, 2008:2). Theorists and researchers can use Critical Realism to develop more precise explanations for a group of occurrences or events. (Wynn & Williams, 2008:2). It is not necessary to use procedures that are more appropriate for the natural sciences, thus CR can be seen as the preferred paradigm for approaching complex phenomena, such as occurrences often observed in information systems (Wynn & Williams, 2008:2). CR allows for the expansion and improvement of existing research methods. (Wynn & Williams, 2008:2).

Regarding social scientific research, critical realism became a powerful alternative paradigm to conduct research, opposing positivism and interpretivism (Wynn & Williams, 2008:3). Contemporary CR is ontologically based and aims to answer what reality should be like for science to become achievable (Wynn & Williams, 2008:3). The last-mentioned is based mostly on Roy Bhaskar's ideas (Wynn & Williams, 2008:3). The core assumption of Critical Realism is that hypotheses developed by a scientific study must centre around the objective fact that makes up the world, even when humans are sometimes unable to completely comprehend or experience this reality, and our existing knowledge regarding objective reality is far from perfect (Wynn & Williams, 2008:3). Consequently, CR has been dubbed as "ontologically bold but epistemologically cautious” by W. Outhwaite (Wynn & Williams, 2008:3). Hence, CR research focuses on determining what reality should be so as to explain the happening of a specific set of events (Wynn & Williams, 2008:3).

Critical realism claims that the world is not falsifiable to the conditions through which humans get access to it (Wynn & Williams, 2008:4). To rephrase: humans only see a small part of the objective world, and nature of the objective world is difficult to grasp, characterize, and measure (Wynn & Williams, 2008:4). The ideas that emerge through scientific practice are part of a transitive dimension of science that includes knowledge generated by humans with the aim of explaining and understanding the objective nature of the universe, which is an intransitive dimension in and of itself (Wynn & Williams, 2008:4). These theories can't be directly compared to the real world; instead, humans can only perform comparisons to see how well the explanation of the observable occurrences under investigation happens to be (Wynn & Williams, 2008:4). The contrast between experiences, events, and structures distinguishes Critical Realism from empiricist theories like positivism (Wynn & Williams, 2008:5). Critical Realism understands that phenomena occurring within an environment can be quantified, such as experiences, and they are likely to only make up a small portion of the actual events that occur inside a given social system (Wynn & Williams, 2008:6). As a result, ontologically, events that occur within a given structure are unrelated to the experiences that we can objectively observe and measure (Wynn & Williams, 2008:6). Critical Realism sees reality as an open system that humans cannot directly influence (Wynn & Williams, 2008:6).

Critical realism aims to explain objective reality through participants' observations and interpretations, as well as the examination of additional objective data (Wynn & Williams, 2008:7). As a result, knowledge claims generated by Critical Realism researchers are centred on identifying components of reality, specifically mechanisms and structures, that must exist in order for the events/experiences that are being investigated to take place (Wynn & Williams, 2008:7). These knowledge claims are predicated on a number of CR-specific epistemological assumptions: explanation rather than prediction or comprehension, socially mediated knowledge and explanation by mechanisms, being unable to observe mechanisms, as well as various plausible mechanisms (Wynn & Williams, 2008:7). The belief in the presence of a mechanism that is warranted is typically hampered with in Critical Realism by the reality that these processes are rarely observable or measurable (Wynn & Williams, 2008:8). The impacts manifested in the subsequent events and experiences are sometimes the only way to identify these processes and the structural components from which they are created (Wynn & Williams, 2008:8). Actual occurrences our experiences of them, as well as the underlying mechanisms and structures that interact to create them, are separate, which is consistent with stratified ontology principles. (Wynn & Williams, 2008:8). Consequently, the observant’s knowledge assertions may be based on both observable and unobservable elements, such as structures and mechanisms (Wynn & Williams, 2008:8).

Following the discussion of the essence of Critical realism, the following figure illustrates this paradigm’s methodologies/approaches:

Diagram

Description automatically generated

Figure 3: Key Relationships between Ontological, Epistemological and Methodological Principles (Wynn & Williams, 2008:9)

The table below gives a summary of the methodologies typically implemented in a CR study:

Table 2: Summary of Methodological Principles for Conducting CR-based Case Study Research

Table

Description automatically generated

Source: (Wynn & Williams, 2008:10)

The figure below indicates research objects along with their research modes and purpose in a Critical Realism study:

Table

Description automatically generated

Figure 4: A taxonomy of information systems research approaches (Dobson, 2001:203)

One can speculate from the discussion of the essence of Critical Realism, as well as the figures above, Critical Realism would be the suited paradigm for this study. CR is perfect for researching IS areas where complex events occur within an open system, and only explanations and descriptions are necessary to provide for the current events occurring. Understanding how and why these events occur is insignificant within a Critical Realism paradigm. CR is also perfect for contributing to current knowledge and expanding on a subject. There is a clear distinction between experiences, events, and structures within a CR paradigm, and those elements should be objectively examined while considering that there are factors that those elements cannot influence. Critical Realism is suited for this study, since a researcher can observe and interpret events occurring within a system, while keeping objectivity in mind and considering that the truth/reality are relative. The main aim of CR is to explain why certain events in a system occur, rather than a prediction or comprehension of why these events occur in a system. Critical Realism also takes into consideration that there are events within a system that can be unobservable. Experiences and observations are an important aspect of Critical Realism, as a result, it is critical to keep track of your experiences/observations in order to get a more objective/partial picture of reality. Another important part regarding the paradigm is the methodologies that will be followed to gain knowledge and research a subject area. An empirical approach will be followed to gain knowledge and conduct the literature study via experiments and testing. By examining [**Figure 3**](#Figure)**,** the applicable ontology will most likely be an Objective Reality or Open Systems Perspective. The applicable epistemology will likely be explanations rather than prediction or explanations via mechanisms. The most suited methodology for this study will be an empirical corroboration approach. When examining [**Figure 4**](#FigureMethod)**,** one can speculate that the appropriate modes to investigate in this literature study will be laboratory experiments, field experiments and possibly surveys. Due to the nature of the study, using surveys to collect data might not be the best possible approach, due to the fact that the research is based on designing and implementing an IoT system. Surveys provide more quantitative approach to sample data based on a hypothesis or answering a theoretical research question. To conclude: Critical Realism is the most effective research paradigm to use to conduct the literature study, since one of the main objectives of the study is to build a knowledge base on how to design and implement an IoT system. Experiences should be documented thoroughly and events occurring should be analysed without understanding the deeper cause of all events occurring in the research environment.

## Research Methodology

The study design revolves around Design Science Research (DSR), as well as experimentations and documenting experiences. When designing objectives using this methodology, the available knowledge base is usually insufficient, and designers must rely on intuition, experiences, as well as trial-and-error methods (Hevner *et al.*, 2004:99). The artefact produced by the study is an experiment, and usually one makes use of new and upcoming technological methods/tools to design and build the artefact and finally being able to answer a research question (Hevner *et al.*, 2004:99). Using DSR, researchers can explore the nature of the problem, the environment(s) it occurs in, and formulate solutions as a result of the artefact’s execution, which emphasizes the need of creating and implementing prototype artefacts (Hevner *et al.*, 2004:99). Within the Information Systems context, it so happens that the design-science research paradigm is technologically proactive (Hevner *et al.*, 2004:98). It focuses on developing and analysing cutting-edge IT artefacts that help organisations tackle critical information driven activities (Hevner *et al.*, 2004:98). The behavioural science research paradigm is reactive to technology (Hevner *et al.*, 2004:98). It focuses on the development and justification of theories that tries to explain and predict strange occurrences connected to the possession, deployment, management, and application of such technologies (Hevner *et al.*, 2004:98). Problems within Design Science Research includes an exaggeration of technology artefacts and failing to maintain a sufficient conceptual framework, which could result in a artefacts that is well designed, but happens to be ineffective in real-world organizational settings (Hevner *et al.*, 2004:98). This study aims to provide an artefact that provides value to society and can be deployed real-time, therefore eliminating abovementioned problems while contributing to a knowledge base on how to design and implement an effective IoT system. The figure below describes the framework to conduct DSR for a given subject area:

Diagram

Description automatically generated

Figure 5: Information Systems Research Framework (Hevner et al., 2004:80)

One can conclude from this figure that experiments, field studies and simulations will have to be conducted in order to design and implement a real-time IoT system. The research methodology should be able to provide vital information and resources to the knowledge base's foundations, such as theories, frameworks, constructs and methods. The figure below describes evaluation methods in order to conduct research (and most of them will be implemented in this study):

Table

Description automatically generated

Figure 6: Design Evaluation Methods (Hevner et al., 2004:86)

To conclude: the study will be more empirical by nature and will contribute to a knowledge base on how to design and implement an IoT system, while documenting the experiences in designing and implementing the solution as well as to document obstacles or errors that occurred during the design/implementation of the artefact. The documentation on the obstacles/errors should be able to guide the user to quickly setup their own IoT system, and when the user does have a problem to implement the system according to guidelines, they should be able to get an alternative perspective on solving a problem based on the documentation.

### Design Science Research (DSR)

According to Peffers *et al.* (2007), Design Science is a methodology or paradigm which can be used to develop and assess IT artefacts, the artefacts themselves aimed at resolving identified organizational issues. DSR entails a thorough process of designing artefacts to solve observable problems, as well as contributing to research, evaluating the planned designs, and communicating the results from observance and experiences to relevant audiences (Peffers *et al.*, 2007:49). Typical artefacts produced using the DSR methodology includes instantiations, methods, models and constructs, to name a few (Peffers *et al.*, 2007:49). Social innovations or unique properties of technological, social, or informational resources may also be included. (Peffers *et al.*, 2007:49). To sum up: any product created with an incorporated solution in mind to an extensively researched topic is included in this description (Peffers *et al.*, 2007:49). The methodology that will be used to conduct this study is Design Science Research that falls under the Critical Realism paradigm. Before diving into the lifecycle of the project using DSR, q quick recap is needed on what the study is about. The goal of this research is to learn how to read sensory input as a stream from an Arduino R3 Uno and transfer it to a streaming platform (as mentioned in the Introduction, the Kafka streaming platform is preferable). After the data has been stored on the platform, it must be converted into a pivot-table format and shown on a dashboard (ideally Desktop Tableau). From the result based on the analysis, one should be able to make specific business decisions based on the tableau's conclusions. The research focuses on the design and implementation of an artefact that can collect sensory input, store data as streams, and display the findings in a tableau so that analytics may be performed. Another important aspect of this study is to contribute to a knowledge base on how to design and implement an effective IoT system that can be deployed real-time. Experiences and observations should be documented, and they should serve as guidelines on how to successfully design and setup an IoT system, while also providing insights on how to tackle problems by looking at a different angle. The following diagram illustrates the lifecycle of a project using Design Science Research (the steps that will be used to design and implement this project):

Diagram

Description automatically generated

Figure 7: Design Science Research Methodology Process Model (Peffers et al., 2007:54)

A further elaboration on each step in the above process model will follow:

1. **Identifying the problem and motivation**

Define the research question in detail and justify the importance of a solution (Peffers *et al.*, 2007:54). The research question will be used to create an artefact that can provide a functional solution, breaking down the problem piece by piece can allow the solution to capture the complexity of the problem, and proves to be beneficial in the long run (Peffers *et al.*, 2007:54). Being able to motivate why a solution can be used to solve a problem contributes to both the researcher and intended audience to pursue the answer and to accept results flowing out of the solution, while also visualising the researcher’s reasoning regarding how to solve the research question (Peffers *et al.*, 2007:55). Emphasis is placed on understanding the problem and how to resolve it (Peffers *et al.*, 2007:55).

1. **Defining the objectives of a solution**

From the problem definition and knowledge of what is achievable, deduce the goals of a solution (Peffers *et al.*, 2007:55). Objectives might be quantitative, such as the terms in which a desirable solution would be preferable to current ones, or qualitative, such as a description of how a new artefact is meant to provide answers to previously unsolved problems (Peffers *et al.*, 2007:55). The objectives should be logically deduced from the problem description (Peffers *et al.*, 2007:55). Observance of the current state of problems, as well as existing solutions (if they exist) and their effectiveness, are essential to define the objectives of a solution (Peffers *et al.*, 2007:55).

1. **Design and development of the artefact**

During this step the researcher starts to build the artefact (Peffers *et al.*, 2007:55). Artefacts include constructs, models, procedures, or instantiations (all of them defined rather broadly), and they novel qualities of social, technological, or any other informational resources (Peffers *et al.*, 2007:55). Any constructed object that incorporates a research input into the design is referred to as a design research artefact (Peffers *et al.*, 2007:55). This activity entails identifying the desired functionality or purpose and the architecture of the artefact (Peffers *et al.*, 2007:55). Finally, the researcher has to build the artefact (Peffers *et al.*, 2007:55). One of the resources required for changing from aims to design and development is knowledge of theory that may be used to design and implement a solution (Peffers *et al.*, 2007:55).

1. **Demonstration of the artefact**

The researcher has to demonstrate that an artefact works or contributes to solving a problem (Peffers *et al.*, 2007:55). The researcher might have to simulate several instances of a problem to verify that the artefact actually does its job (Peffers *et al.*, 2007:55). Methods to test whether an artefact’s functionality follows user requirements could be done via experimenting with it, running simulations, performing case studies, documenting experiences and observations, or any other activity that is relevant (Peffers *et al.*, 2007:55). Effective understanding of how to use the artefact to address the problem is one of the most important principles required for the demonstration (Peffers *et al.*, 2007:55).

1. **Evaluation of the artefact**

Examine and assess how efficient the artefact supports a solution that contributes to solving a problem (Peffers *et al.*, 2007:56). This exercise entails comparing a solution's objectives to actual results observed from the demonstrations of the artefact (Peffers *et al.*, 2007:56). It necessitates an understanding of key measurements and analysis methods (Peffers *et al.*, 2007:56). Depending on the nature of the problem and the type of artefact, evaluation can take on many various forms. (Peffers *et al.*, 2007:56). It may include a comparison between the functioning of the artefact or the goals of the solution from [**step 2**](#ProcessModel) in the process model, objective quantitative performance measurements like budgets or items produced, survey results based on customer/participant satisfaction, client feedback, or simulations, among others (Peffers *et al.*, 2007:56). It could include measurable system performance indicators like response times or availability (Peffers *et al.*, 2007:56). In theory, such an assessment may incorporate any relevant logical proof or empirical evidence (Peffers *et al.*, 2007:56).

1. **Communicating the results to peers or intended audience**

If necessary, one should make researchers and other target audiences, such as practicing professionals, aware about the problem and its significance, the artefact itself, its value and distinctiveness, the thoroughness of its design, and its effectiveness (Peffers *et al.*, 2007:56). The formal structure of an empirical research process (identifying and describing a problem, literature review, the development of a hypothesis, data sampling, analysis, observations and results, debates or discussions, and finally the conclusion) is also a required framework for empirical research papers, and researchers may use the framework of this process to structure the paper in scholarly research publications (Peffers *et al.*, 2007:56). Communication necessitates familiarity with the presumptions, point of views, approaches, methodological analysis, and core beliefs members of an academic disciplinary community have (Peffers *et al.*, 2007:56).

### DSR implementation of artefact

Following the Design Science Research approach/processes as indicated in [Figure 7](#DSR_Process) on page 20, the design and development approach of the IoT system and streaming database (artefact) will be conducted as follow:

1. **Identify the problem and provide a motivation**

The research problem is how to develop an IoT system by using a streaming database. How would one go about to stream data from an Arduino sensor to the Kafka streaming platform, and pipeline the converted data from Kafka to a dashboard environment (Tableau Desktop)? How would one utilize the Tableau Desktop environment to perform data analytics and present the data to make an informed business decision? The motivation for the research problem is to contribute to a new/existing knowledge base that will allow future generations of developers to design and implement IoT systems via using a streaming database. The knowledge base and documentation provided should allow individuals with the minimum IoT and streaming databases knowledge to set up and deploy smart systems at the fastest possible rates, while also being able to provide high quality and effective IoT systems. A proper use case for the IoT system will be developed at a later stage, as the research is more of a generic nature at this stage.

**Sensors**

The research question pertaining to this section is what type of Arduino sensor will be suitable to stream data to the Kafka streaming platform (given a specific use case that will be determined at a later stage). Another important aspect is to identify optimal circuitry for the Arduino microcontroller in order to efficiently stream data to the Kafka streaming platform. The motivation behind the research question is to develop an IoT system that can stream data and perform analytics in order to provide feasible results to make an informed business decision, provided a specific use case. It is also important to select a sensor that is applicable to the use case and to study the functionalities and properties of said sensor.

**Kafka**

The research problem pertaining to this section is how to stream sensory input from the Arduino microcontroller to the Kafka streaming platform. The motivation behind the research problem is to design a IoT system by means of using an Arduino microcontroller with a selected sensor that is suitable to a specific use case, the IoT system should be able to stream data from the Arduino microcontroller to the Kafka streaming platform. It is also necessary to identify additional tools and platforms that can streamline the streaming process from a sensor to Kafka.

**Tableau Desktop/Dashboards**

The research problem in this case is how to pipeline data from the Kafka streaming platform to Tableau Desktop. What data visualization techniques will be suitable for a given IoT sensor use case? How will the developer communicate the results from analysis and data visualization to the relevant stakeholders? What informative decision can be made based on the results from the Dashboard? Is Tableau Desktop the most efficient/effective Dashboard software to use? What are the key performance indicators or key performance risks for the given use case? The motivation behind the research problem is to create a knowledge base, with regards to creating and implementing a solution to read sensory input as a stream from an Arduino Uno R3 to the Kafka streaming platform, transferring the converted data to Tableau Desktop in order to perform analytics in the dashboard environment and make an informative business decision based on the results from the dashboards. Future generations must be able to learn how to construct a successful Internet of Things system via following guidelines from the knowledge base.

1. **Define the objectives of the solution**

In order to design and develop the artefact, efficient and effective tools and development platforms will be necessary to ensure a simplistic yet intuitive IoT system design. One of the main objectives is to find suitable tools and platforms that allows streamlined development of a smart system that utilizes streaming databases. The design and development of the IoT system must be documented as the development process moves along, and any faults/errors detected in the IoT system must be carefully documented to prevent future developers from making the same mistakes.

**Sensors**

The IoT system (specifically the Arduino microcontroller) should have a sensor that can stream real-time data to the Kafka streaming platform. Another possible objective is to include a sensor, such as light-emitting diodes, to provide noticeable output to indicate that the primary sensor is busy streaming real-time data. The Arduino microcontroller should also have efficient circuitry to prevent short-circuits in the system and to provide the sensors with optimal power.

**Kafka**

The objective of the solution is to design/develop the Arduino microcontroller to stream data to the Kafka streaming platform. Additional tools and platforms should be identified that can assist in the design and implementation process of the IoT system.

**Tableau Desktop/Dashboards**

* Transfer data from the Kafka streaming platform to Tableau Desktop or another dashboard environment.
* Identify the data visualization techniques that will be the most suited to present the results from analysing the real-time data to stakeholders.
* Identify the KPIs or KPRs for a given real-time data streaming IoT system.
* Set benchmarks for dashboard results (wat the developer/stakeholders want to know from a dashboard).
* Plan a presentation on how to communicate the final results to stakeholders.
* One should be able to make an informed business decision based on the results from the dashboards.
* Provide a knowledge base on how to design and implement a functional real-time data streaming IoT system.

1. **Design and develop the artefact**

During this stage, the artefact will be physically designed and implemented. Research must be conducted on following standard design principles of streaming databases. A plan must be developed to oversee data flows from the Arduino sensors to the Kafka streaming platform and to the Tableau Desktop environment. Pipelining of the data should be carefully planned. The artefact must be developed, connecting sensors to the Arduino, transferring real-time data from the sensors to the Kafka platform, converting data streams on the Kafka platform and transferring it to Tableau Desktop. Data visualisation techniques and data analysis strategies must be employed to conclude a valid business decision.

**Sensors**

During this stage, the design and development of the Arduino microcontroller starts. The relevant sensors, provided a specific use case, should be connected to the microcontroller. One should ensure that enough power supply terminals and digital I/O pins are available in order to connect additional sensors, should it become necessary to. Optimal circuitry should be designed to avoid short-circuits and for ideal power supply to sensors. The process of connecting the sensors and designing the circuitry should be documented. Any errors or faults detected in connecting the sensors or designing the circuitry should be documented as well to contribute to a knowledge base and preventing future developers from making the same mistakes when designing and developing a real-time streaming IoT system.

**Kafka**

During this stage, one should already have a functional Arduino microcontroller that can receive sensory inputs. This is the stage where one needs to setup and install the Kafka streaming platform and connect the Arduino microcontroller to the streaming platform in order to pipeline data. Data pipelining should be carefully planned via creating data flow diagrams. It would be an excellent idea to design a diagram that indicates the streaming database architecture. As the design and development of the artefact goes along, one needs to document each step of the processes. Any faults or errors that is detected in the data pipelining system or Kafka architecture must be properly documented. Experimentations, such as prototyping, with a variety of different tools and platforms to stream data from sensors to the Kafka platform might be necessary to determine which of the tools/platforms are the most efficient and simple to stream real-time data. One should create a script that generates high-velocity and high-volume data to determine how effectively the Kafka streaming architecture manages real-time data.

**Tableau Desktop/Dashboards**

During this stage, the design and development of the Dashboard section begins. One should pipeline the data from the Kafka streaming platform to the Tableau Desktop environment. Data analysis should be performed in the dashboard environment and data visualization techniques (such as area or bar charts, scatter plots, heat maps etc.) should be employed to display the results from the data analysis. Key performance indicators should be properly defined during this stage in order to ensure that the stakeholders receive relevant information from the dashboards. One should write a script that generates sufficient real-time data in order to make correct inferences from results on the dashboard. It might be necessary to design the dashboards with alternative dashboard software, such as Power BI or IBM Cognos, as Tableau Desktop is licensed software and only provides a two-week free trial period. The design and development process should be documented as time goes by. Any errors detected in the dashboard environment should pe properly documented as well. A presentation should be designed to communicate results to stakeholders.

1. **Demonstrate the artefact**

After designing and developing the artefact, the IoT system must be properly tested to determine whether it is functional. Sprints of several test will have to be conducted in order to gain a respectable overall perspective whether the system adheres to user requirements. It should also be taken into consideration to assess the IoT system by providing erroneous or incomplete stream data to test how the IoT system manages faulty data. In cases where the IoT system is provided with erroneous data, the system should provide clear error messages as to why the system cannot manage/pipeline the data.

**Sensors**

Sprint test runs should be conducted to determine whether the microcontroller’s sensors can stream data. The results and observations should be documented. The test runs can be conducted by printing the received data on a console on the Arduino IDE platform. It would be optimal to display error messages if the Arduino IDE does not receive any streamed data, and indicators as to why the sensors failed to stream any data. In some instances, circuitry might be the cause for a sensor to fail to stream data. The circuity should either then be redesigned, or one should ensure that all pins are properly connected to the microcontroller. The demonstration serves the purpose of determining whether the system adheres to user requirements.

**Kafka**

The demonstration is used to evaluate how well the technology complies with user needs.Once again, a series of sprint test runs should be conducted in order to determine how the Kafka streaming platform manages streamed data. All results should be properly documented and stored. It is necessary to record any errors that occurred during the demonstrations of data streaming. Keeping logs of why errors occurred, and why Kafka failed to pipeline data from the Arduino microcontroller can prove to be invaluable to design a more efficient and optimal IoT system, as well as preventing future developers from making the same mistakes when building similar systems.

**Tableau Desktop/Dashboards**

Using the script that generates sufficient real-time data, one should observe the results from data analysis and dashboards. It might be necessary to run the script a few times in order to generate enough data for the dashboards to display relevant information. The dashboards can be screenshotted as evidence of the test runs.

1. **Perform an evaluation of the artefact**

After demonstrating the artefact, an evaluation session should occur to determine how the IoT system performed overall. The evaluation can be done by documenting the results and drawing a valid conclusion based on the data analysis from the sprint test runs. A comparison must occur between solution objectives and the actual demonstration data results.

**Sensors**

An evaluation session should follow the demonstration of the microcontroller's capability to stream data from sensors to ascertain how well the data streaming went overall. Documenting the outcomes and coming to a sound conclusion based on the data analysis from the sprint test runs are two ways to conduct the review. The solution goals and the actual demonstration data outcomes must be compared.

**Kafka**

Following the demonstration of the artefact, an evaluation session should be held to determine the real-time data streaming performance from the Arduino microcontroller to the Kafka streaming platform. Documented results from the demonstration will assist the developer in drawing valid conclusions (did the artefact meet user requirements/did the artefact provide the intended functionalities?). There must be a comparison between the solution objectives and the actual demonstration data results. It may be necessary to evaluate key performance indicators in order to determine the efficiency of the IoT system. It is of the utmost importance to determine how the IoT system manages high-velocity and high-volume data. Did the tools and additional platforms used to design the stream processing system contribute to an efficient IoT system, or did it lead to worse data streaming performance/cause the system to crash?

**Tableau Desktop/Dashboards**

During this stage it is important to evaluate the results from the dashboards. Did the dashboards provide enough information in order to make an informative business decision? Did the data visualization techniques provide enough information as to inform users what the purpose of the artefact is? A comparison must occur between the solution objectives and the actual demonstration data results, to determine the efficiency of the IoT solution and whether it adheres to user requirements. Did the Kafka-dashboard architecture efficiently pipeline data, or would it be necessary to redesign the architecture and use different dashboard tools and software?

1. **Communicate the results of demonstration/evaluation to audience**

After conducting the evaluation session, the results should be communicated to the audience that have an interest in this artefact. The results can be presented in a written form or presentation (either visually or orally). One should communicate the research problem to the audience and why/how the solution contributes to resolving the problem. It is also a good idea to walk one’s audience through the design and implementation process, to gain a deeper understanding of the artefact. Questions that the audience have about the artefact must be addressed professionally and being prepared for the questioning session is of utmost importance.

**Sensors**

The audience that are interested in this artefact should be informed about the evaluation session's findings. The findings may be given orally or in writing. It would be a good idea to present the Arduino microcontroller’s data streaming capabilities live to the audience. One should explain to the audience the research problem and why or how the solution helps to solve the problem. For a deeper understanding of the artefact, it is also a good idea to walk the audience through the design and execution process. The audience's inquiries about the artefacts must be managed properly, and it is crucial to be ready for the question-and-answer period.

**Kafka**

The findings of the evaluation session should be shared with the audience members who are interested in this artefact. The Arduino microcontroller's data streaming capabilities should be demonstrated to the audience in real time. One should explain the research problem to the audience and why or how the solution contributed to solve the problem. The streaming process system should also be demonstrated to the audience via the use of a poster or an electronic visual presentation. One should be properly prepared to answer any doubts or uncertainties the audience may have.

**Tableau Desktop/Dashboards**

During the final stage one should communicate the results of the dashboards to the intended audience or stakeholders. The final presentation should contain the dashboards and indicate what informative business decisions can be made from the analysed data. It might be a promising idea to walk the stakeholders through the Kafka-dashboard architecture to explain how data is pipelined from Kafka to the dashboard environment. One should be prepared for any questions the stakeholders might have during the presentation session, and it is important to focus on key metrics to discuss the relevance of information derived from the dashboards.

## Streaming Databases

### Introduction

A real-time, continuous, ordered (implicitly by arrival time or explicitly by timestamp) succession of items is known as a data stream. It is not possible to save a complete stream locally, nor is it possible to regulate the sequence in which the items arrive (Hébrail, 2008:90). Data streams are processed using continuous queries by DSMSs, which are specialized systems based on stream processing engines (Surdu, 2011:69). These systems were developed to meet the needs of applications that monitor streams of data in order to deliver useful results (Surdu, 2011:69). Benchmarks demonstrated that a DSMS can perform at least a factor of four better than a DBMS when processing heavy load data streams and doing continuous and one-off queries (Surdu, 2011:69). To maximize throughput and speed up systems, Data Stream Management systems (DSMS) handle data without storing it (Zimanyi, 2017:4). Combinations of values are delivered by data sources in the form of data streams throughout time (Surdu, 2011:67). Some applications will find it easier to use the data thanks to the stream processing programming paradigm, which is comparable to data-flow programming, event stream processing, and reactive programming (Zimanyi, 2017:2). Software systems that manage data streams by providing flexible concurrent processing are known as data stream management systems (DSMS), more commonly known as streaming databases (Zimanyi, 2017:2). These systems list processing and computation of data as a higher priority than storing it (Zimanyi, 2017:2). Data streams are instantly processed, after which they are discarded (Surdu, 2011:70). An element cannot be retrieved after it has been viewed and managed by the processing engine (Surdu, 2011:70). Although some data stream history may be saved, ultimately it will be discarded (Surdu, 2011:70). The variety of data sources may significantly grow as time goes on (Surdu, 2011:70). The rate at which tuples enter the stream, or stream tuple rate, can also be adjusted (Surdu, 2011:70). When a system is overloaded, it might no longer be able to respond to queries promptly (Surdu, 2011:70). High tuple rate can be addressed using a range of methodologies (Surdu, 2011:70). Real-time analytics and continuous processors are powered by continuous queries (Zimanyi, 2017:6). A continuous query is essentially a query support that refreshes previously emitted results (Zimanyi, 2017:6). In other words, a continuous query produces results in a dynamic table that is continuously updated and may be queried in the same way as a static table (Zimanyi, 2017:6). Continuous queries run continuously (hence the name) and produce a dynamic table updated on the fly, in contrast to a conventional query that terminates and produces a static table (Zimanyi, 2017:6). This idea is comparable to the one that materialized view maintenance in SQL attempted to implement (Zimanyi, 2017:6).

### Management and processing of real-time data

**Figure 8** describes a stream processing system's architecture for sensor applications. The data source manager, query processing engine, catalog manager, scheduler, QoS (Quality of Service) manager, and ECA (event-condition-action) manager are the six parts that comprises the system (Jiang & Chakravarthy, 2006). Continuous data streams are accepted by the data source manager, which adds input tuples to the appropriate input queues of query plans (Jiang & Chakravarthy, 2006). Additionally, it monitors several stream input characteristics as well as data stream characteristics (such as categorization) (Jiang & Chakravarthy, 2006). These features offer valuable data for query scheduling, query optimization, and QoS control (Jiang & Chakravarthy, 2006). The task of creating query plans and dynamically improving them falls under the purview of the query processing engine (Jiang & Chakravarthy, 2006). Both continuous queries and one-off ad-hoc queries are supported (Jiang & Chakravarthy, 2006). The system's stream meta data, comprehensive query strategies, and resource data are all stored and managed by the catalog manager (Jiang & Chakravarthy, 2006). Due to the system's continuous nature, the scheduler chooses which operator or query to run during each time period (Jiang & Chakravarthy, 2006).  A few scheduling tactics exists to handle the scheduling and execution of queries (Jiang & Chakravarthy, 2006). The majority of stream-based applications have different QoS requirements, thus in order to ensure that the QoS requirements of diverse queries are met, the QoS manager uses a variety of QoS delivery mechanisms (such as load shedding, admission control, and so on) (Jiang & Chakravarthy, 2006). Monitoring changes through CQs (continuous queries) is essential and crucial in a sensor environment to record and comprehend the physical world (Jiang & Chakravarthy, 2006). It is important to note that it is equally crucial to respond to such changes right away (Jiang & Chakravarthy, 2006). Within a sensor application containing a stream processing architecture, complicated events are detected, and predefined actions are taken in response utilizing the ECA manager.

Diagram

Description automatically generated

Figure 8: Architecture of Continuous Query Processing System (Jiang & Chakravarthy, 2006)

The figure below provides a simple generalized architecture of a data stream management system:

Diagram

Description automatically generated

Figure 9: Abstract reference architecture for a data stream management system (Golab & Ozsu, 2003:3)

If the system can't keep up, an input monitor may reject packets as a means of controlling the input rates (Golab & Ozsu, 2003:2). Three partitions are commonly used to store data: summary storage for stream synopses, temporary working storage for window queries, and static storage for meta-data (such as the actual location of each source) (Golab & Ozsu, 2003:2). Although one-time questions over the status of the stream may also be presented, long-running queries are registered in the query repository and grouped for shared processing (Golab & Ozsu, 2003:2). In response to shifting input rates, the query processor may re-optimize the query plans in communication with the input monitor (Golab & Ozsu, 2003:2). Users either stream or temporarily buffer the results. Finally, based on the most recent results, users can refine their queries (Golab & Ozsu, 2003:2).

### Applications

Sensor networks can be used to monitor geophysical conditions, traffic jams, mobility, vitals in the medical field, and manufacturing procedures (Golab & Ozsu, 2003:3). These programs use intricate filtering, and when odd patterns in the data are found, an alarm is set off (Golab & Ozsu, 2003:3). While aggregation over a single stream may be required to make up for individual sensor failures (due to physical damage or low battery power), aggregation and joins over several streams are necessary to analyse data from multiple sources (Golab & Ozsu, 2003:3). Access to some historical data may be necessary for sensor data mining (Golab & Ozsu, 2003:3).

Ad-hoc technologies are already in use to monitor Internet data usage in almost real-time (Golab & Ozsu, 2003:3). Similar to sensor networks, it is necessary to combine data from many sources, monitor and filter packets, and look for unexpected circumstances (such congestion or denial of service) (Golab & Ozsu, 2003:3). It is also necessary to support historical inquiries and web mining, maybe to compare current traffic patterns with patterns that correspond to well-known occurrences like a DoS attack (Golab & Ozsu, 2003:3). Other criteria include keeping track of recent/popular URL queries or identifying the users who use the most bandwidth (Golab & Ozsu, 2003:3). These are particularly significant because it is a general conception that Internet traffic patterns follow the Power Law distribution, which results in a selected few users consuming a significant portion of the available bandwidth (Golab & Ozsu, 2003:3).

Low latency requirements and real-time responses are provided by continuous queries (Surdu, 2011:69). A driver must receive a real-time notification whenever a new toll is imposed for his or her vehicle in a variable tolling system that computes highway tolls depending on dynamic criteria like accident proximity or traffic congestion, for example (Surdu, 2011:70). It would be useless to provide such a response in the future (Surdu, 2011:70).

The online mining of call logs, call history, and transactions from Automated Teller Machines all adopt the data stream approach (Golab & Ozsu, 2003:3). Finding intriguing customer behavioural patterns, spotting questionable expenditure patterns that can point to fraud, and predicting future data values are main objectives of transaction log analysis systems (Golab & Ozsu, 2003:3). This necessitates merging several streams, intricate filtering, and statistical analysis, as with most other streaming applications (Golab & Ozsu, 2003:3).

Regardless of the industry, current Information Systems (IS) manage an increasing amount of data to enable human activity (Hébrail, 2008:91). Analysis of ever-increasing amounts of data is necessary for the monitoring of these systems, such as the supervision of a telecommunication network (Hébrail, 2008:91). The "store then process" approach, which was formerly utilized to maintain massive data warehouses, is no longer an option (Hébrail, 2008:91). Real-time supervision applied to specific data is strongly preferred to batch supervision applied to aggregate data (Hébrail, 2008:91). This can only be accomplished using data stream processing, an approach that was devised to process data as it is being created (Hébrail, 2008:91).

The primary function of many operational information systems is to process streams of data that arrive at a rapid intervals (Hébrail, 2008:91). For instance, in the financial industry, computerized systems help traders by automatically assessing the development of stock markets (Hébrail, 2008:91). In the same way that original Information Systems were created using files to store data before database technologies were available, such systems are being created without any generic tools to process streams (Hébrail, 2008:91). Data stream processing solutions will promote the approach to enable generic software to handle stream queries, mine streams for data, and publish the results (Hébrail, 2008:91).

The first industry to employ commercial data stream management technologies on an industrial scale today is the Financing industry (Hébrail, 2008:91). The primary use of these type of systems in Finance is to support trading choices by tracking the evolution of stock price and sales volume as time progresses (Hébrail, 2008:91). These programs fall within the second group of operational systems, whose main objective is the control of data streams (Hébrail, 2008:91). Finding stocks whose price has climbed by 3% over the past hour and whose volume of sales has increased by 20% over the past 15 minutes are typical stream queries for these type of applications (Hébrail, 2008:91).

### Benefits and Drawbacks

According to Zimanyi (2017:5), some of the benefits provided by data stream management systems are as follow:

* Analytics respond immediately to events (Zimanyi, 2017:5). There is no delay between the event's occurrence, how it is handled, and the actions that is executed (Zimanyi, 2017:5).
* Larger volumes of data can be handled by streaming databases than other processing systems (Zimanyi, 2017:5).
* In the real world, data happens to be more continuous in nature. Such systems are suitable for handling real-time data containing huge volumes and rapid rates of transfer (Zimanyi, 2017:5).
* Infrastructure decentralization. Moving toward a microservices architecture means phasing out large, expensive data centres (Zimanyi, 2017:5).

The table below points out the main differences between a traditional database management system and a data stream management system:

Table 3: DBMS and DSMS comparisons

Table

Description automatically generated

**Source: (Zimanyi, 2017:7)**

Streaming databases are not feasible in all situations, however, and there are some issues/disadvantages, such as the following:

* Backtracking over a data stream is not possible due to performance and storage limitations. Online stream algorithms can only make one pass over the input data (Golab & Ozsu, 2003:2).
* Due to the fact that it is impossible to save an entire stream, approximate summary structures like synopses and digests are recommended (Golab & Ozsu, 2003:2). Queries about the summaries may therefore not have precise responses/results (Golab & Ozsu, 2003:2).
* Blocking operators, which requires the entire input stream before being able to produce any results, may not be permitted in some streaming query plans (Golab & Ozsu, 2003:2).
* It is rather difficult to gather, analyse, and consume dynamic, diverse, and unbounded observation streams.

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 10: Big Data Challenges (Gürcan & Berigel, 2018:4)

## Arduino Sensors

### Introduction

A sensor is a device that changes a physical quantity's resistance or produces a voltage or current by converting it to an electrical signal (Ziemann, 2018:5). Various types of electrical signals, such as currents or voltage, are represented by the output signals (Patel *et al.*, 2020:1). The sensor is an apparatus that receives various signals, such as physical, chemical, or biological signals, and transforms them into an electric signal (Patel *et al.*, 2020:1). Based on the applications, input signal, conversion method, material utilized, and sensor properties like cost, accuracy, or range, the sensors are divided into many types (Patel *et al.*, 2020:1). Active and passive sensors are the two primary categories of sensors (Patel *et al.*, 2020:2). A passive sensor does not need an additional energy source, and it responds to external stimuli by producing an electric signal (Patel *et al.*, 2020:2). Accordingly, the sensor transforms input energy into output signal energy (Patel *et al.*, 2020:2). Photographic, thermal, electric field sensing, chemical, infrared, and seismic sensors are a few examples of passive sensors (Patel *et al.*, 2020:2). The response of the active sensors, or excitation signal, requires external energy sources (Patel *et al.*, 2020:2). Sensors make the appropriate adjustments to these input signals to produce the output signals (Patel *et al.*, 2020:2). Due to their own features that can change in reaction to an external effect and then transform into electric signals, the active sensors are also known as parametric sensors (Patel *et al.*, 2020:2). There are several applications for active sensors in meteorology and observation of the atmosphere and surface of the Earth (Patel *et al.*, 2020:2). The table below indicates diverse types of sensors and their properties:

Table 4: Sensors based on their detection properties

Text, table

Description automatically generated with medium confidence

**Source: (Patel *et al.*, 2020:3)**

In the past ten years, there have been significant advancements in sensor technology in terms of sensitivity, intelligence, and compactness (Patel *et al.*, 2020:19). Traditional sensors including photosensors, optical sensors, capacitive sensors, and nearly all other types of sensors have been supplanted by integrated circuit versions such as MEMS (microelectromechanical systems) (Patel *et al.*, 2020:19). Since the sensors are compactly integrated into all contemporary computer and navigational devices, a typical smartphone typically contains around twenty-two sensors that serve a variety of functions (Patel *et al.*, 2020:19). The development of sensor technology has led to the creation of smart sensors that are intelligent and wearable (Patel *et al.*, 2020:19). In major applications like self-driving cars, where hundreds of smart sensors are used for seamless and smooth driving without aid from a driver, this can be observed in smart watches, smart gadgets, or other electronic devices (Patel *et al.*, 2020:19). This also applies to the use of AI in these fields, as well as others like robotics, medical diagnosis, and brain-computer interfaces (BCIs) (Patel *et al.*, 2020:19). The sensors now possess the intelligence and smarts thanks to artificial intelligence for new and cutting-edge uses in the business world, the medical field, and complex automation (Patel *et al.*, 2020:19).

### Applications

 Militaries make use of a   system called RADAR (Radio Detection and Ranging), which utilizes radio waves to detect objects and determine their range, altitude, direction, and speed (Kaswan *et al.*, 2020:1114). Radars consists of different sizes and performance requirements (Kaswan *et al.*, 2020:1114). Typical applications include long-range surveillance, early-warning systems in ships, and air traffic control at airports (Kaswan *et al.*, 2020:1114). A missile guidance system's core is made up of these types of systems (Kaswan *et al.*, 2020:1114). In times of conflict, numerous small portable radar equipment as well as systems that take up several large rooms are maintained and operated (Kaswan *et al.*, 2020:1114). Arduinos are utilized in many industries due to the simplicity of the programming environment, signal kinds, and simple adaptability in new setups (Kaswan *et al.*, 2020:1114). In order to add remote control and monitoring features to minor legacy industrial systems, Arduino boards provide a flexible, low-cost alternative to the typical industrial gadgets (Kaswan *et al.*, 2020:1114). Due to the expansion of wireless technologies like Wi-Fi and cloud services in recent years, the use of wireless devices in daily life has become a standard routine (Kaswan *et al.*, 2020:1114). Nowadays, Arduinos are used to monitor traffic lights, as well as real-time control systems with configurable timings, pedestrian illumination, as well as other applications (Kaswan *et al.*, 2020:1114). In a traffic control system, the junction timing is automatically adjusted to accommodate smooth vehicle movement and prevent traffic jams at junctions (Kaswan *et al.*, 2020:1114). The number of heartbeats in a minute can be counted by using an Arduino-based heartbeat monitor (Kaswan *et al.*, 2020:1114). The Arduino possesses an integrated heartbeat sensor module that detects the heartbeat when a finger is placed on the sensor (Kaswan *et al.*, 2020:1114). Many other medical devices can be designed using an Arduino microcontroller (Kaswan *et al.*, 2020:1114). Arduinos can be utilized to operate a variety of bodily parts, including handSight gloves, Breathalyzer microphones, heart rate monitors, and other medical devices (Kaswan *et al.*, 2020:1114). A heart rate monitor powered by an Arduino is more sophisticated than one that merely measures the user's heart rate (Kaswan *et al.*, 2020:1114). Each button verbally explains what it does while also displaying the measurements on the screen. The latest four readings will be saved, displayed, averaged, and even some uplifting quotes can be provided via a monitor (Kaswan *et al.*, 2020:1114). This sensor is used to monitor activity levels and patterns, fevers, and hypothermia (Kaswan *et al.*, 2020:1114).

Facial expressions can be detected by using this equipment (Kaswan *et al.*, 2020:1114). We can determine breathing rate, breathing depth, activity level, and arousal levels with the aid of this Arduino contraption (Kaswan *et al.*, 2020:1114). Some Arduino medical devices can also be used to identify both the frequency and intensity of muscular contractions, and this provides the functionality to monitor bodily movements (Kaswan *et al.*, 2020:1114). These are but a few of the examples of fields that employs Arduino microcontrollers and their sensors. The table below indicates the applications of Arduino microcontrollers:

Table 5: Summary of Arduino Application areas

Table

Description automatically generated

**Source: (Kaswan *et al.*, 2020:1115)**

## Kafka Streaming Platform (Apache Kafka)

### Introduction

A distributed streaming platform called Kafka has three key features: The first step is to publish, distribute, and subscribe to records streams using a message queue or real-time messaging system (Gürcan & Berigel, 2018:3). Secondly, a reliable strategy is to store data streams with fault tolerance (Gürcan & Berigel, 2018:3). Process log and event streams are also a main capability of this streaming platform (Gürcan & Berigel, 2018:3). In order to connect streaming data systems and support real-time applications, Kafka provides dependable and low latency responses (Gürcan & Berigel, 2018:3). Kafka is designed to provide a consistent, high throughput, low latency infrastructure for processing feeds of real-time data and can be used to store and process data streams (Fernandes *et al.*, 2020:428). Kafka can be used to handle events in real-time and integrate different software system components (Gürcan & Berigel, 2018:3). Kafka offers superior scalability and message consistency compared to other streaming platforms such as Flume (Gürcan & Berigel, 2018:3). Data flow from one application to another is managed by a messaging system, which focuses on data. Reliable message queuing serves as the foundation for distributed messaging (Fernandes *et al.*, 2020:428). Two different messaging patterns are available, namely a Point to Point Messaging System, during which messages (streaming data) is transferred as a queue, the messages in the queue can be consumed by several consumers (Fernandes *et al.*, 2020:428). The Publish-Subscribe Messaging System utilizes topics to contain messages. Each message in a given topic can be consumed by consumers from many topics (Fernandes *et al.*, 2020:428). Kafka provides streaming applications with benefits such as load balancing and data replication, as well as high-velocity management of data (Fernandes *et al.*, 2020:428). The limitations of Kafka include a decrease in performance due to brokers and consumers compressing and decompressing data flows, this can also affect the throughput, which can cause problems for the Kafka broker (Fernandes *et al.*, 2020:428). When a message requires some tuning, the Kafka broker may occasionally have issues since Kafka's performance is decreased (Fernandes *et al.*, 2020:428).

### Applications

* **Activity tracking**

Kafka was originally designed to monitor transactions and activities on web pages (Franklin, 2022). LinkedIn had to redesign its user activity tracking pipeline as a collection of publish-subscribe feeds that are updated in real-time (Franklin, 2022). Due to the enormous volume of activity messages, also known as events, generated by each user page view, activity tracking is frequently quite intensive (Franklin, 2022). Clicks on web pages, user registrations, likes on social media, amount of time spent on web pages, orders, and environmental changes are some examples of events that can be tracked by Kafka (Franklin, 2022). Events can be published or produced under specific Kafka topics (Franklin, 2022). Each feed can be loaded into a data lake or warehouse for offline processing and reporting, among a variety of other use cases (Franklin, 2022). Additional applications can subscribe to topics, obtain the data, and use it as necessary for monitoring, analysis, reports, newsfeeds or personalization purposes (Franklin, 2022).

* **Real-time data processing**

Data must be processed as soon as it becomes available for many systems (Franklin, 2022). Kafka transports data with an extremely low latency from producers to consumers (measured in milliseconds) (Franklin, 2022). Financial institutions can collect and process payments/transactions as soon as they are received, this provides to be very useful when institutions try to counter fraudulent transactions and stop the transactions immediately (Franklin, 2022). Financial institutions are also provided with the ability to update dashboards with current market values (Franklin, 2022). Predictive maintenance (used in IoT systems) makes use of models that continuously examine streams of measurements from out-of-the-box equipment and provides sound alerts as soon as they spot deviations that might be signs of impending breakdowns (Franklin, 2022). Mobile devices must be able to process real-time data for the use of certain applications, such as navigation (a user can use Google Maps to travel from a starting location to any specified location) (Franklin, 2022). Logistics and the supply chain industries can monitor and update their tracking software in order to continuously track shipments/cargos and estimate the time of arrival in real-time to clients (Franklin, 2022).

* **Messaging systems**

Since Kafka has superior throughput, built-in segmentation, replication, and fault-tolerance, as well as improved scaling characteristics, it works well as a replacement for conventional message brokers (Franklin, 2022).

* **Monitoring of operational metrics or key performance indicators (KPIs)**

Kafka provides the frequent monitoring of operational data. Statistical data from remote applications are combined in this process to create consolidated feeds of operational data (Franklin, 2022).

* **Log Aggregations**

Kafka is a popular log aggregation tool for companies (Franklin, 2022). In order to process log files, log aggregation often entails gathering actual log files from servers and storing them in a centralized location, such as a file server or data mart (Franklin, 2022). Kafka encapsulates the data as a stream of messages and filters file specifics (Franklin, 2022). This makes it possible to process data with lower latency and to accommodate distributed data consumption and numerous data sources more easily (Franklin, 2022). Kafka provides comparable speed to log-centric systems like Scribe or Flume while also providing substantially reduced end-to-end latency and higher durability assurances thanks to replication (Franklin, 2022).

### Streaming data from sensors to Kafka

Sources that are generated continuously streams data to the Kafka platform (Wu *et al.*, 2020:207). Examples of these sources includes IoT device sensors, user activities stored from websites via the use of cookies or local storage, and payment requests received from mobile apps (Wu *et al.*, 2020:207). In most of these use cases, streaming data needs to be managed real-time due to the fact that the value of these data becomes useless over time (Wu *et al.*, 2020:207). To provide an example: a fraud detection system should be able to detect a fraudulent transaction before it is completed, thus ensuring that financial losses are circumvented in the process (Wu *et al.*, 2020:207). Given that Apache Kafka serves as a modern distributed messaging system, offering elevated durability and scalability functionalities, it makes sense that corporate companies such as Twitter, Spotify, Uber and Netflix employ this platform to handle real-time data streaming (Wu *et al.*, 2020:207). Kafka can process streamed data as an uninterrupted series of batch jobs on small batches of streaming data (Wu *et al.*, 2020:207). Throughput as well as end-to-end latency in the stream processing system can be severely impacted by larger batch sizes (Wu *et al.*, 2020:207). It does however provide the utilization of network bandwidth. The main purpose of Kafka is to transfer data from one platform to another in order to process data for multiple reasons (Wu *et al.*, 2020:208).

Components of Kafka in a stream processing system can be described as follow:

* Messaging schema (Publish/Subscribe, also known as pub/sub): built on top of the producer, topic, broker and consume components (Wu *et al.*, 2020:208).
* Producer: application that sends streaming data, also known as messages, to their respective topics (different logical groups of messages) (Wu *et al.*, 2020:208).
* Consumer: application that subscribes to a specific topic and reads messages from said topic (Wu *et al.*, 2020:208).
* Kafka cluster: serves as the core part of the messaging system. This is essentially a distributed storage system that consists of various brokers (Wu *et al.*, 2020:208).
* Brokers: servers that receives messages from a Kafka producer, provides disk storage for messages and duplicate the messages in order to provide fault tolerance (Wu *et al.*, 2020:208). Brokers distribute all received messages in a specific topic among each other and they are stored in sequential write-ahead logs, also known as partitions (Wu *et al.*, 2020:208). Enterprise applications usually have a predetermined number of partitions per topic, each broker can host multiple partitions under a specific topic (Wu *et al.*, 2020:208).

The figure below describes the stream processing pipeline using Kafka as the messaging system:

Diagram

Description automatically generated

Figure 11: Kafka in a streaming system

A Kafka cluster consists of three brokers, and the Filter, Map, reduce and Aggregate stream processors executes specific operations on the received streaming data (Wu *et al.*, 2020:208). The messages can be delivered from multiple type of external sources, such as IoT sensors, payment terminals as well as mobile devices (Wu *et al.*, 2020:208). The messages are then transferred via a stream processor (Wu *et al.*, 2020:208). From the above figure, filter processing is performed, and messages are published to an explicit topic, such as Topic A from the example (Wu *et al.*, 2020:208). Other processors from the streaming system transfer required messages from the Kafka cluster by means of subscription to respective topics (Wu *et al.*, 2020:208). To provide an example: the Map processor subscribes to topic A, ingests messages, mapping operations are then performed on the messages and send the results to Topic B for other processors to subscribe (Wu *et al.*, 2020:208). Furthermore, downstream applications may make use of user-defined functions to retrieve results for different services in order to make decisions (Wu *et al.*, 2020:208). That is, every processor uses the producer and consumer application programmable interface to communicate with the Kafka cluster, in order to transfer data flows between different processors in the streaming system (Wu *et al.*, 2020:208).

### Processing real-time data and queries

This section is a follow-up from the previous section. As mentioned previously, Kafka is a distributed streaming platform that allows the streaming of high-velocity and high-volume data. Refer to [Figure 11](#Kafka_Architecture) for the standard Kafka architecture. Every topic category has several partitions, and each partition has an ordered, immutable sequence of messages that are continuously added (Liu *et al.*, 2014:359). Kafka maintains these message feeds (Liu *et al.*, 2014:359). Distinct sequential ids are provided to each message in order to identify it within a partition (Liu *et al.*, 2014:359). A partition in a Kafka cluster is distributed over several nodes for fault tolerance (Liu *et al.*, 2014:359). The published messages are stored in a Kafka cluster for a specified amount of time (Liu *et al.*, 2014:359). Regardless of whether they have been read or not, messages are discarded when the time comes. Each subscriber to a message is identified by the name of the group they are a part of (Liu *et al.*, 2014:359). To guarantee the proper message order, a partition of a topic can only be sent to one consumer in a consumer group (Liu *et al.*, 2014:359).

User activity tracking pipelines was originally rebuilt using Kafka as a collection of real-time publish/subscribe feeds (Liu *et al.*, 2014:359). Site activities, such as page browsing, searching, and other user actions, are published to a Kafka cluster as topics, one topic is provided for each type of activity (Liu *et al.*, 2014:359). For a variety of use cases, including real-time processing, real-time monitoring, data loading into Hadoop or a data warehouse, those topics are available via subscription (Liu *et al.*, 2014:359). The Kafka data pipelines are observed for monitoring activities, such as keeping track of the statistics of aggregations from distributed applications that produce centralized feeds of operational data (Liu *et al.*, 2014:359). Kafka is thus well adapted for circumstances where users must manage and analyse real-time data (Liu *et al.*, 2014:359). LinkedIn currently supports dozens of subscribing systems and sends more than fifty-five billion messages to users each day. This is all possible through the use of streaming data with Kafka (Liu *et al.*, 2014:359).

## Dashboards: Tableau Desktop

### Introduction

A dashboard is a program or user interface that aids in understanding the organizational structures and business procedures of a company as well as being a useful analytical tool that measures the performance of an organisation (Ioana *et al.*, 2014:852). Dashboards address the intensifying complex nature of market data that senior management faces in the age of information (Pauwels *et al.*, 2009:176). Key performance indicators (KPIs) and key risk indicators (KRIs) are displayed in connected charts, maps, and scorecards in modern dashboards so that a company can concentrate on the most crucial performance operations (Ioana *et al.*, 2014:852). The dashboard's goal is to clearly present information on a single screen so that it can be understood by all stakeholders (Ioana *et al.*, 2014:852). Four factors are mentioned by managers as driving the need for dashboards: the need for cross-departmental integration in performance reporting practices and for resource allocation, poorly organized pieces of potentially decision-relevant data, managerial biases in information processing and decision making, the increasing demands for marketing accountability, given the dual objective of companies to grow the top-line while keeping costs down for a healthy bottom-line (Pauwels *et al.*, 2009:176). The purpose of a dashboard is to summarize important performance metrics with underlying factors, and they should be integrated in order to communicate performance levels throughout a company (Pauwels *et al.*, 2009:177). A dashboard can be described as a reasonably compact group of linked KPIs and underlying performance factors that represents both immediate and long-term objectives and can be accessed by all members of the company (Pauwels *et al.*, 2009:177). Tableau Desktop is software that provides analytics and data visualization using drag and drops (Sleeper, 2021:9). One can connect to a range of data sources, analyse data of all sizes, perform ad hoc studies instantly, integrate various components into coherent dashboards for simpler consumption, share views with others, and far more using Tableau Desktop (Sleeper, 2021:10). Tableau Desktop and Tableau Public are comparable, except Tableau Desktop allows one to load a worksheet onto the Tableau platform (Akhtar *et al.*, 2020:31). It has a two-week trial period and is a licensed version (Akhtar *et al.*, 2020:31). By directly linking to data from a data warehouse, one can take full advantage of real-time data analytics (Akhtar *et al.*, 2020:31). One can quickly import data from various sources into Tableau's data engine and combine it by merging different views into an interactive display (Akhtar *et al.*, 2020:31). Files created by Tableau desktop have twb and twbx extensions (Akhtar *et al.*, 2020:31). Server as well as online sources can be used by Tableau Desktop users as a high-performance data store (Akhtar *et al.*, 2020:31). Excel, Tableau Public, Reporting Services, SharePoint, Microsoft SQL Server, and Performance Point are all programs that can be used to construct dashboards (Ioana *et al.*, 2014:852). All of these tools allow for the breakdown of business objectives, and the development of a plan to incorporate the required data (Ioana *et al.*, 2014:852). Nevertheless, the majority of dashboards are generated in Excel because it is inexpensive to implement and can be used to track business performance on timely intervals (daily, weekly, monthly) (Ioana *et al.*, 2014:852). Dashboards can be used as a tool to alter the corporate culture at all levels of the organization (Ioana *et al.*, 2014:855). One of the most significant advantages of adopting dashboards is that managers are able to analyse data on a single screen where KPIs and KRIs are tracked, the decision-making process are streamlined, and actions can be taken to reduce risks and enhance corporate performance (Ioana *et al.*, 2014:855).

### Applications

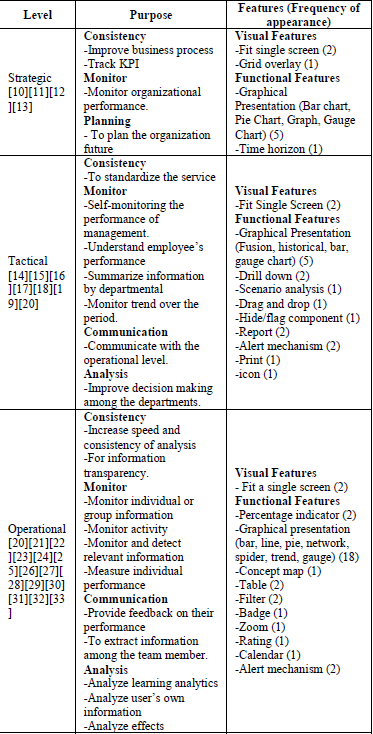
Dashboards are frequently associated with corporate organizations, where they are used to optimize decision-making, improve operational efficiency, increase data visibility, drive strategy and transparency, lower costs, and improve communication (Sarikaya *et al.*, 2018:686). A dashboard ensures that measurements and measurement practices are uniform across all departments and business units (Pauwels *et al.*, 2009:179). To provide an example: Avaya employs a variety of marketing strategies and conducts business in over 50 different countries and markets (Pauwels *et al.*, 2009:179). Prior to the dashboard project, the company lacked a global system of records (which restricted data collection), had disparate definitions of qualified leads (a crucial performance metric in the transition from marketing to sales), and had a lack of regional interest in metrics collection (Pauwels *et al.*, 2009:179).

Performance levels can be tracked via the use of a dashboard (Pauwels *et al.*, 2009:179). The monitoring process itself can be both developmental (what the company have learned from the performance measures) and/or evaluative (which employees or departments within the company performed well?) (Pauwels *et al.*, 2009:179). Google employs dashboard measures which serve as early performance indicators, and the organization will take corrective action if, for instance, the "trust and privacy" statistic indicates a deterioration (Pauwels *et al.*, 2009:179).

A dashboard can be used to plan the goals and strategies of a company (Pauwels *et al.*, 2009:179). To illustrate this point, Ameritrade developed a dashboard that links into the planning cycle and is connected to quarterly bonuses using corporate scorecards from the strategic planning department (Pauwels *et al.*, 2009:179).

A dashboard could be used to inform stakeholders about essential company information and statistics (Pauwels *et al.*, 2009:179). By selecting the metrics on the dashboard, it specifically communicates not only what the performance is but also what a business regards as performance measures (Pauwels *et al.*, 2009:179). Vanguard provides to be a fantastic example of a company that effectively communicates dashboard metrics to their Board and translates their business focus on client loyalty, feedback, and word-of-mouth into their dashboard measurements (Pauwels *et al.*, 2009:179). The dashboard is extensively utilized for many different reasons in sectors including education, health, and logistics (Rahman *et al.*, 2017:1). For instance, dashboards are used in the educational environment to monitor performance and ensure consistency in order to increase student awareness of their own and their peers' learning activities (Rahman *et al.*, 2017:1). It also aids in reflection, clarity, and the impact of the learner's traces on behaviour (Rahman *et al.*, 2017:1). Even more diverse dashboard use cases and requirements can be found outside of the Business Intelligence area (Sarikaya *et al.*, 2018:686). As an illustration, health organizations have adapted dashboards at both the large-scale (hospital management) and patient-care levels, with the main goal to promote collaboration and awareness across various roles, timespans, and skills (Sarikaya *et al.*, 2018:686). Urban informatics and community organizations/non-profits, and large/diverse group of stakeholders all encounter obstacles with integrating various data sources, adapting to multiple platforms, including mobile devices, and developing metrics and representations to represent abstract results like community awareness, stakeholder engagements, as well as trust (Sarikaya *et al.*, 2018:686).

Table 6: Dashboard purposes and features



**Source: (Rahman *et al.*, 2017:3)**

### Transferring data from Kafka to dashboard environment

This section suggests a hybrid architecture that strives to make the most of available tools in order to create an environment that can support the processing and real-time visualization of information. The data sources can be of various forms presented in the first section of this architecture (Sousa *et al.*, 2021:15). In accordance with the suggested architecture, these data sources communicate with Solace directly (Sousa *et al.*, 2021:15). The company's own APIs as well as open protocols should be used, according to the guideline. This approach has no technological limitations placed on the architecture (Sousa *et al.*, 2021:15). Since Solace PubSub+ excels in both event broadcasting and event management, Solace viewed other streaming platforms as obsolete (Sousa *et al.*, 2021:15). This is only accurate in instances of transactional and operational use, though (Sousa *et al.*, 2021:15). For the storage and administration of analytical use cases in event flows, Kafka continues to shine (Sousa *et al.*, 2021:15). Kafka is a necessary component in this architecture, due to the fact that the objective is to build an architecture that is as resilient and covers as many use cases as possible (Sousa *et al.*, 2021:15). Although still in development, this new Solace PubSub+ version is not yet as stable as hoped. Additionally, a number of businesses that have invested in using these technologies in production use Kafka (Sousa *et al.*, 2021:15). This technology may be utilised with the architecture suggested here, and it also fosters synergy between all of its parts (Sousa *et al.*, 2021:15). Furthermore, this proposal makes Kafka utilization optional for those who like simpler architectures and are only beginning to invest in these areas (Sousa *et al.*, 2021:15). Furthermore, it is suggested that in this instance of incorporating Kafka, connectors created by Solace should be used to enable data connectivity between Solace and Kafka (Sousa *et al.*, 2021:15). MQTT, AMQP, REST, WebSocket, and JMS are just a few of the basic protocols that Solace supports for connecting directly to Kafka (Sousa *et al.*, 2021:15). Pipelining data to Apache Spark is the next step after storing it in Solace and/or Kafka (Sousa *et al.*, 2021:15). By doing this, we are allowing consumers to select the technology they want to employ to access the data (Sousa *et al.*, 2021:15). The data analysis will subsequently be carried out in a parallel and distributed fashion using the data present in Apache Spark (Sousa *et al.*, 2021:16). The two data access zones on Tableau (suitable tool for data analysts) and Power BI (more suitable for less skilled employees) forms the final component of the architecture (Sousa *et al.*, 2021:16). By employing this approach, expenses can be decreased due to the fact that Tableau is the most expensive software in the suggested design, and staff can use Power BI more efficiently because it is highly flexible and simple to use (Sousa *et al.*, 2021:16). The provided architecture is unique since it incorporates hybridization on top of Kafka in addition to offering a well-researched pipeline option for information processing (Sousa *et al.*, 2021:16). Users have the choice to remain invested because Solace no longer supports the connection between Kafka and the suggested architecture (Sousa *et al.*, 2021:16). Additionally, as depicted, the use of Kafka is optional since Solace supports all tasks designed for Kafka (Sousa *et al.*, 2021:16). **Figure 12** below demonstrates the architecture described in this section:

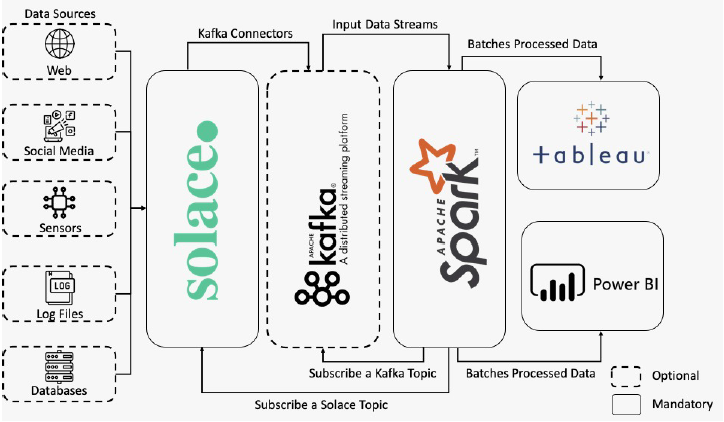


Figure 12: A hybrid architecture supporting Kafka and Tableau integration (Sousa *et al.*, 2021:15)

### Data visualization techniques/Data analysis

Data visualization is the ability to increase the effectiveness of data organization and processing (Akhtar *et al.*, 2020:29). The depiction of data or information in a graph, chart, or other visual format is known as data visualization (Akhtar *et al.*, 2020:29). People can process more complicated information and improve memory through visualization. It conveys how data and images relate to one another (Akhtar *et al.*, 2020:29). This is crucial since it makes it easier to spot trends and patterns (Akhtar *et al.*, 2020:29). We need to be able to comprehend progressively larger quantities of data as Big Data becomes more relevant in the modern era (Akhtar *et al.*, 2020:29). Data analysis is the study of examining raw data to draw inferences from it (Patel, 2022:7). Numerous methods and strategies for data analysis are automatically transformed into machine operations and algorithms that process raw data for human utilization (Patel, 2022:7). A corporation can enhance its performance through the use of data analysis (Patel, 2022:7).

In a typical business analytics dashboard, important components include:

* **Data aggregation:** the process of gathering, organizing, and sorting data in preparation for analysis (Patel, 2022:9).
* **Data mining:** the process of analysing large amounts of data for business statistics utilizing statistics, machine learning, and big data sets to find patterns and establish connections (Patel, 2022:9).
* **Association and Sequence Identification:** recognizing unexpected behaviours carried out in tandem with other behaviours or patterns (Patel, 2022:9).
* **Text Mining:** large-scale, unstructured data sets are scanned and organized for optimal quality/quantity analysis (Patel, 2022:9).
* **Forecasting:** the process of periodically analysing previous data to provide accurate predictions of future actions or events (Patel, 2022:9).
* **Optimization:** after patterns have been discovered and forecasts have been made, companies can use simulation techniques to determine the ideal circumstances (Patel, 2022:9).
* **Data Visualization:** the visual presentation of data in the form of graphs and charts allows for quick and simple data interpretations (Patel, 2022:10).

To efficiently and simply portray the key points of the data, graphs and charts covering enormous amounts of information can be edited into simple formats (Patel, 2022:10). Consider the goal of the graph or chart as well as the information one wants to provide before deciding how to present the data in the best possible way (Patel, 2022:10). The following type of charts are available using Tableau software:

* **Line charts**

A series of data points connected by a straight line are included this type of chart (Patel, 2022:11). Each point plotted on the chart explains how the graph's horizontal and vertical axes relate to one another (Patel, 2022:11). The X-axis displays the important measurements or time dimensions, such as month, quarter, or year, while the Y-axis displays the numerical value (Patel, 2022:11). Typically, temporal patterns and correlation, as well as data over a specific time period, are represented using line and area charts (Patel, 2022:11).

* **Area charts**

By colouring the region between the line segment and the x-axis, area charts are used to jointly measure data patterns across timely periods (Patel, 2022:12). A line chart is simply a continuation of an area chart (Patel, 2022:12).

* **Column line charts**

A combination of a line chart and a column chart is commonly known as a column line chart (Patel, 2022:13). One measure is shown as a column and another as a line in this sort of chart (Patel, 2022:13). These two metrics are displayed under several time units, including months, quarters, and years (Patel, 2022:13). This graph works far better to indicate the relationship between two measurements over time (Patel, 2022:13).  Sales revenue, net income, and gross margins are examples of common measurement on a column line chart (Patel, 2022:13).

* **Bar charts**

Among all chart kinds, the bar chart is the one that is used the most (Patel, 2022:15). Bar charts are frequently used to organize data, filter out information that is unimportant given a particular use case, and position data by value in descending or ascending order (Patel, 2022:15). A bar chart is typically used to compare several categorical data values, which includes grouping the data by fusing numbers together in a chart (Patel, 2022:15). The less important categorical data should be placed into another group if there are multiple categories available (Patel, 2022:16).

* **Stacked area charts**

The base area chart is expanded upon the stacked area chart (Patel, 2022:16). Since the values of each group are placed on top of one another, it is possible to see both the sum of a numerical variable and the importance of each group from the same image (Patel, 2022:16).

For example, the lines can be made to show how the population of different states has changed through time (Patel, 2022:16). A graph representing population trends and showing statistics of each state in order from least to most populous state can be created by colouring the region beneath each line to symbolize the state it represents (Patel, 2022:16).

* **Horizontal Charts**

The most effective graphical tool for showing comparisons between data categories frequently is a horizontal bar chart (Patel, 2022:17). With the use of a horizontal bar chart, the presenter can easily exhibit huge data labels because the horizontal rectangles offer space for text presentations (Patel, 2022:17).

* **Waterfall Charts**

A waterfall chart is a visual representation of data that makes it easier to comprehend the impact of adding up positive or negative values (Patel, 2022:17). These intermediary values may be based on phase or on time (Patel, 2022:17). Due to what appears to be a formation of columns (bricks) in the midst of the air, the waterfall chart is also known as the flying brick chart or Mario chart (Patel, 2022:17). It is typically referred to as a bridge in the Financing industry (Patel, 2022:18). Using this kind of chart, one can observe how values fluctuate with regards to positive and negative values (Patel, 2022:18). Typically, waterfall diagrams are used to show how a series of intermediate values rise and decrease a starting value to produce a final outcome (Patel, 2022:18).

Abovementioned points are but a few of the visualization techniques used within Tableau software. It is important to identify key performance indicators in order to determine which type of visualization technique to employ to present statistics/results to stakeholders. Several dashboards allow numerous coordinated views (Sarikaya *et al.*, 2018:685). Data faceting using slicers and filters, cross-highlighting by choosing data items within views, and drilling up and down the layers of a data hierarchy can all be used to interact between views (Sarikaya *et al.*, 2018:685). These dashboards enable users to narrow their analysis to the data points that matter to them (Sarikaya *et al.*, 2018:685). Cross-highlighting (such as darkened visual elements) or the presence of standard interactive elements (such as drop-down menus or slicers), are deemed as interactive capabilities of a dashboard (Sarikaya *et al.*, 2018:685). The figure below is how a typical Tableau Desktop dashboard looks:

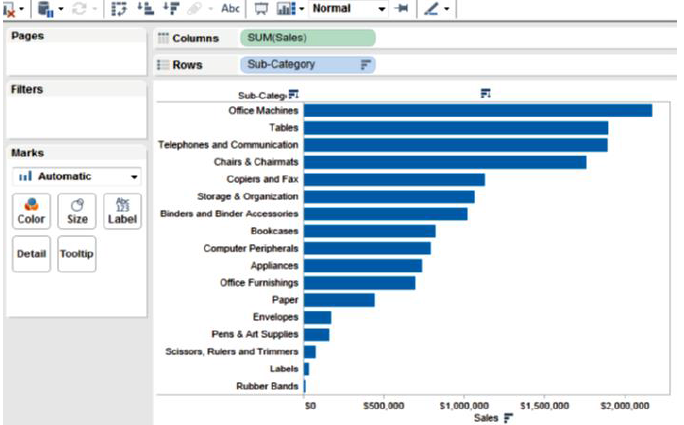


Figure 13: A typical Tableau Dashboard (Akhtar *et al.*, 2020:37)

## Chapter Summary

A streaming database provides the means to store and process real-time data. Streaming databases can manage multiple data streams from disparate data sources. These type of data repositories allows one to analyse data real-time. Streaming databases are perfectly suited for IoT applications, automated business processes and real-time analytics since they can process and analyse queries immediately as they are received. A streaming database provides several advantages when developing real-time applications, such as reduced development costs and faster application development times. Managers can market their products a lot faster and improve the decision-making process via using the real-time analytics provided by streaming databases. IoT sensors allows streaming databases to receive data in real time. Sensors are used in modern day applications to detect changes in an environment and respond accordingly. Sensors communicate with other devices in a network in order to transfer data between devices and systems. Sensors are well suited for modern real-time applications as they can warn one of events occurring, and businesses can make an informed decision on how to respond to the events based on the warnings and data provided by sensors. Data provided by sensors are often used for real-time analysis purposes and allows business managers to effectively manage business processes based on the real-time data. Sensors have a variety of purposes, and are often used in temperature measuring applications, detecting health vitals of patients in a hospital, humidity detection in an environment and detecting movements in a room. Apache Kafka is a distributed streaming platform that publishes/subscribes to real-time data streams and makes use of message queueing to pipeline data. A main feature of Kafka is to provide fault tolerance for data streams and to process data streams as soon as they arrive on the platform. Kafka allows developers to build real-time streaming data pipelines to transfer data between sensors and applications, as well as assisting in the development of real time applications that transforms or reacts to streams of data. Kafka also provides a level of scalability that most other streaming platforms fail to offer. A dashboard is a tool that allows one to visualize data and present the most valuable information so managers can make informed business decisions and to plan company objectives and strategies. Dashboards should be arranged on a single screen so stakeholders can gain a comprehensive overview of key performance metrics. Dashboards provides the means to display vast amounts of data on a single screen and streamlines the decision-making process. A dashboard can also indicate what type of data is essential to business process, and managers can re-evaluate business strategies and determine new key performance indicators to monitor the performance of business processes. A variety of data visualization techniques are provided by dashboards, such as charts and graphs. It is important for managers to decide which type of visualization technique to use, as they support various kinds of performance measurements. Tableau Desktop is a dashboard software solution that allows the rapid creation of dashboards. It can also manage enormous amounts of data sufficiently and is thus well suited for real-time data analytics. The main objective of this study is to determine how stream sensory input from an Arduino microcontroller and pipeline real-time data to a streaming database/platform, such as Apache Kafka. The data must be processed on the streaming platform and finally get pipelined and displayed on a dashboard. Tableau Desktop is the preferred dashboard software at the moment. Based on the data analysis from the provided dashboards, one should be able to make a strategic business decision. This study's potential to contribute to a body of information on how to design and deploy a real-time IoT system is another crucial feature. It is important to keep track of experiences and observations since they may be used to provide suggestions for creating and trying to implement an IoT system as well as to shed light on alternative ways to tackle challenges observed during the design and implementation process. An essential component of this study is the documentation of incidents, interactions, and observations.

Chapter 3: Name of chapter (Chapter)

# Heading 1 won’t print. Don’t delete – doing so will lead to incorrect numbering.

## Heading 2

### Heading 3

## Heading 5

#### Heading 6

##### Heading 5

Chapter 4: Name of chapter (Chapter)

# Heading 1 won’t print. Don’t delete – doing so will lead to incorrect numbering.

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

Bibliography

Akhtar, N., Tabassum, N., Perwej, A. & Perwej, Y. 2020. Data analytics and visualization using tableau utilitarian for covid-19 (coronavirus). *Global Journal of Engineering and Technology Advances*, 3(2):28-50. <https://hal.archives-ouvertes.fr/hal-03226643/document> Date of access: 3 July 2022. <https://doi.org/10.30574/gjeta.2020.3.2.0029>

Calvillo-Arbizu, J., Román-Martínez, I. & Reina-Tosina, J. 2021. Internet of things in health: Requirements, issues, and gaps. *Computer Methods and Programs in Biomedicine*, 208:10. Date of access: 30 March 2022. 0.1016/j.cmpb.2021.106231

Fernandes, E., Salgado, A.C. & Bernardino, J. 2020. Big data streaming platforms to support real-time analytics*.* In. ICSOFT. pp. 426-433.

Franklin, J. 2022. *Apache kafka use cases: When to use it & when not to*. <https://www.upsolver.com/blog/apache-kafka-use-cases-when-to-use-not> Date of access: 2 July 2022.

Golab, L. & Ozsu, M.T. 2003. *Data stream management issues–a survey*. <https://cs.uwaterloo.ca/~tozsu/ddbms/publications/stream/streamsurvey.pdf> Date of access: 27 June 2022.

Gürcan, F. & Berigel, M. 2018. Real-time processing of big data streams: Lifecycle, tools, tasks, and challenges*.* In. 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). IEEE. pp. 1-6.

Hébrail, G. 2008. Data stream management and mining. *Mining Massive Data Sets for Security*:89-102. <https://hal-imt.archives-ouvertes.fr/file/index/docid/479571/filename/NICSP19-0089.pdf> Date of access: 27 June 2022. 10.3233/978-1-58603-898-4-89

Hevner, A.R., March, S.T., Park, J. & Ram, S. 2004. Design science in information systems research. *MIS Quarterly*, 28(1):75-105. <https://wise.vub.ac.be/sites/default/files/thesis_info/design_science.pdf> Date of access: 27 Apr. 2022.

Ioana, B., Claudia, S.-P.D. & Ioan, B.M. 2014. Using dashboards in business analysis. *THE ANNALS OF THE UNIVERSITY OF ORADEA*:851-856. <https://web.s.ebscohost.com/ehost/pdfviewer/pdfviewer?vid=0&sid=c74defdf-48ba-484c-aa68-59f35d1120b5%40redis> Date of access: 3 July 2022.

Jiang, Q. & Chakravarthy, S. 2006. Anatomy of a data stream management system. *ADBIS Research Communications*, 215, <http://www.ceur-ws.org/Vol-215/paper12.pdf> Date of access: 28 June 2022.

Kaswan, K.S., Singh, S.P. & Sagar, S. 2020. Role of arduino in real world applications. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, 9(1):1113-1116. <http://www.ijstr.org/final-print/jan2020/Role-Of-Arduino-In-Real-World-Applications-.pdf> Date of access: 30 June 2022.

Liu, X., Iftikhar, N. & Xie, X. 2014. Survey of real-time processing systems for big data*.* In. Proceedings of the 18th International Database Engineering & Applications Symposium. pp. 356-361.

Patel, A. 2022. *Data visualization using tableau.* Metropolia University of Applied Sciences. <https://www.theseus.fi/bitstream/handle/10024/652129/Patel_Ashwin.pdf?sequence=4> Date of access: 3 July 2022.

Patel, B.C., Sinha, G. & Goel, N. 2020. Introduction to sensors. In. *Advances in modern sensors*: IOP. p 367.

Pauwels, K., Ambler, T., Clark, B.H., LaPointe, P., Reibstein, D., Skiera, B., ... Wiesel, T. 2009. Dashboards as a service: Why, what, how, and what research is needed? *Journal of service research*, 12(2):175-189. <https://journals.sagepub.com/doi/pdf/10.1177/1094670509344213?casa_token=FrpHUiyXtiAAAAAA:nj4dZjEFjHX7dphCXWCK3PfhkRMVbK0iR4P78PJ62HcslSOcMdedQ2Kxx1T3vwZeNlvKZkkwXKcVAQ> Date of access: 3 July 2022.

Peffers, K., Tuunanen, T., Rothenberge, M.A. & Chatterjee, S. 2007. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45-77. <https://moodle.ufsc.br/pluginfile.php/4227088/mod_resource/content/1/PEFFERS%20A%20Design%20Science%20Research%20Methodology%20JMIS%20%282007%29.pdf> Date of access: 27 Apr. 2022. 10.2753/MIS0742-1222240302

Pries-Heje, J., Venable, J. & Baskerville, R. 2014. Rmf4dsr: A risk management framework for design science research. *Scandinavian Journal of Information Systems*, 26(1):57-82. <http://iris.cs.aau.dk/tl_files/volumes/volume26/Pries-Heje-etal-26-1.pdf> Date of access: 27 Apr. 2022.

Rahman, A.A., Adamu, Y.B. & Harun, P. 2017. Review on dashboard application from managerial perspective*.* In. 2017 International Conference on Research and Innovation in Information Systems (ICRIIS). IEEE. pp. 1-5.

Sarikaya, A., Correll, M., Bartram, L., Tory, M. & Fisher, D. 2018. What do we talk about when we talk about dashboards? *IEEE transactions on visualization and computer graphics*, 25(1):682-692. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8443395&casa_token=f5CoGxGpxhQAAAAA:9T1eqWrD0op9Av4HS-LIm4EYDMwzaT2-bcU6pyKWGHZT-wmfYr-gZ3ajc_yL-oL03BVPPrYs2R4Xcw&tag=1> Date of access: 3 July 2022.

Sleeper, R. 2021. *Tableau desktop pocket reference*. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc.

Sousa, R., Miranda, R., Moreira, A., Alves, C., Lori, N. & Machado, J. 2021. Software tools for conducting real-time information processing and visualization in industry: An up-to-date review. *Applied Sciences*, 11(11):4800. <https://d1wqtxts1xzle7.cloudfront.net/78138627/pdf-libre.pdf?1641404135=&response-content-disposition=inline%3B+filename%3DSoftware_Tools_for_Conducting_Real_Time.pdf&Expires=1656859789&Signature=QuwBX9lXDMDyCzojMzE7eSMIhJQ7uIdpDGLjTtih0j-d~DwrlWJhjG1-cFCfGPVkUfF7bvBzj0S3ditvdL00Iwu0o8NrunwuwKgU0rjA7PMfFexZshIpH2tM0rLR1vHLtjQjHFM-cPsnwvpwdb~Blt61-ZXKe6tfcZa8xPEqm-I2XGD2~Q75eFNSbhf2Iu79CClhbUgE65zkVgbWikGRY-UKGxu0fq4R3SoqS3K1U01m6MXKU7vugqEjvcjBuw4iqZuPd7Ie93LDBRiPWXEl2BaE6cNGBuj54CTcLxaTkm4pbqA04LqcdzNbBupu2ZaniTufwUwe~zmx2o2ZY34YWA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA> Date of access: 3 July 2022. <https://doi.org/10.3390/app11114800>

Surdu, S. 2011. Data stream management systems: A response to large scale scientific data requirements. *Annals of the University of Craiova-Mathematics and Computer Science Series*, 38(3):66-75. <http://inf.ucv.ro/~ami/index.php/ami/article/view/438/380> Date of access: 27 June 2022.

Watt, A. 2014. *Project management*. 2. British Columbia: BCcampus Open Education. Available from: <https://opentextbc.ca/projectmanagement/> Date of access: 28 Apr. 2022.

Wu, H., Shang, Z., Peng, G. & Wolter, K. 2020. A reactive batching strategy of apache kafka for reliable stream processing in real-time*.* In. 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE). IEEE. pp. 207-217.

Wynn, J., Donald E. & Williams, C.K. 2008. *Critical realism-based explanatory case study research in information systems*. Paper presented at the Association for Information SystemsAIS Electronic Library (AISeL), Paris. <https://core.ac.uk/download/pdf/301346934.pdf> Date of access: 26 Apr. 2022.

Ziemann, V. 2018. *A hands-on course in sensors using the arduino and raspberry pi*. 6000 Broken Sound Parkway NW, Suite 300: CRC Press. (Series in sensors).

Zimanyi, E. 2017. *Streaming databases & pipelinedb*. <https://cs.ulb.ac.be/public/_media/teaching/pipelinedb_2017.pdf> Date of access: 27 June 2022.

Annexures (TOC\_Heading)