A Digital Twin System for the Integration of Railway Infrastructure Data

by
Gerhardus Christiaan Doubell

Thesis presented in partial fulfilment of the requirements for the degree
of Master of Engineering (Mechatronic)
in the Faculty of Engineering at Stellenbosch University

Supervisor: Dr K Kruger
Co-supervisors: Prof AH Basson, Mr PDF Conradie

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at are those of the author and are not necessarily to be attributed to the NRF.

December 2022

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 6 September 2022

Abstract

A Digital Twin System for the Integration of Railway Infrastructure Data

G.C. Doubell

Department of Mechanical and Mechatronic Engineering
Stellenbosch University
Thesis: MEng (Mechatronic Engineering)
December 2022

The integration of data within large complex systems, such as railway infrastructure, forms a key component of enabling data-led decision making. Data can exist in numerous different data silos, provided by different data sources, with different interfaces and encapsulated data properties. The Passenger Rail Agency of South Africa (PRASA) currently faces this challenge.

Research on digital twins (DTs) highlight the integration of data as one of the key advantages of DTs. However, very little research has been published on the *implementation* of DTs within *railway infrastructure* and, in particular, for the *integration of data*.

Considering the above, this thesis presents the development of a DT system technology demonstrator for the integration of data for PRASA’s railway infrastructure. This technology demonstrator is evaluated based on a case study implementation for PRASA maintenance management in the Western Cape province.

The DT system architecture utilises a layer with various “wrapper” components to provide specific interfaces to each of the data sources to be included in the system. In addition to the wrapper layer, the DT system architecture utilises a DT aggregation hierarchy to store data that represents physical reality, where each DT can also provide services that rely purely in the DT's scope of reality. The DT hierarchy is complemented with a services network to provide the required services that, as dictated by its purpose, integrates information from various DTs and, potentially, external sources. The DT system architecture is intended to be vendor-neutral and can therefore be broadly implemented.

The developed DT system promises to be a valuable tool for maintenance management, by providing a holistic view of the railway infrastructure. The view can be reconfigured to adapt to changing infrastructure, data sources and user needs. Having been tested with diverse data sources and interfaces, complex relationships between infrastructural elements, and changing requirements and infrastructure, the DT system technology demonstrator can be considered suitable for PRASA's railway infrastructure network, but also for other contexts where a similar environment (with regards to data sources and interfaces) is encountered.

The integration of data through the DT system provides a means through which the number of data sources that need to be consulted for decision making, can be reduced. At the same time, the DT system provides a platform from which additional services can be offered, both now and in the future.

Uittreksel

‘n Digitale Tweeling-Stelsel vir die Integrasie van Data van Spoorweginfrastruktuur

G.C. Doubell

Departement Meganiese en Megatroniese Ingenieurswese
Universiteit Stellenbosch
Tesis: M.Ing (Megatroniese Ingenieurswese)
Desember 2022

Die integrasie van data binne groot komplekse stelsels soos spoorweginfrastruktuur, vorm 'n kern element van om data-geleide besluitneming moontlik te maak. Data kan in talle verskillende datasilo's, wat deur verskillende databronne voorsien word, met verskillende koppelvlakke, en met verskillende data-eienskappe, bestaan. Die Passasierspooragentskap van Suid-Afrika (*Passenger Rail Agency of South Africa*, PRASA), staar tans hierdie uitdaging in die gesig.

Navorsing oor digitale tweelinge (DT'e) beklemtoon die integrasie van data as een van die belangrikste voordele van DT'e. Baie min navorsing is egter gepubliseer oor die implementering van DT'e binne *spoorweginfrastruktuur* en, in besonder, vir die *integrasie* van data.

Met inagneming van die bogenoemde, bied hierdie tesis die ontwikkeling van 'n tegnologiedemonstrator van 'n DT-stelsel vir die integrasie van data vir PRASA se spoorweginfrastruktuur. Hierdie tegnologiedemonstrator word geëvalueer op grond van 'n gevallestudie-implementering vir PRASA se instandhoudingsbestuur in die Wes-Kaap provinsie.

Die DT-stelselargitektuur gebruik 'n laag met verskeie "omhulsel"-komponente om spesifieke koppelvlakke te verskaf aan elk van die databronne wat in die stelsel ingesluit moet word. Saam met die laag omhulsel-komponente, gebruik die ontwikkelde DT-stelselargitektuur, 'n samevoegingshiërargie van DT’e om data te stoor wat die fisiese werklikheid voor stel, waar elke DT ook dienste lewer wat suiwer van die DT se omvang van realiteit afhang. Die DT hierargie word gekomplementeer deur 'n dienstenetwerk wat vereiste dienste verskaf wat, wanneer daarvoor gevra word, inligting van verskeie DT’e en, potensieel, eksterne bronne integreer. Die DT-stelselargitektuur is bedoel om verkoper-neutraal te wees en kan dus breedweg geïmplementeer word.

Die ontwikkelde DT-stelsel beloof om 'n waardevolle hulpmiddel vir instandhoudingsbestuur te wees, deur 'n holistiese siening van die spoorweginfrastruktuur te verskaf, wat herkonfigureer kan word om aan te pas by veranderende infrastruktuur, databronne en gebruikersbehoeftes. Nadat dit getoets is met diverse databronne en koppelvlakke, komplekse verhoudings tussen infrastruktuurelemente, en veranderende vereistes en infrastruktuur, kan die DT-stelsel tegnologiedemonstrator as geskik beskou word vir PRASA se spoorweginfrastruktuurnetwerk, maar ook vir ander kontekste waar 'n soortgelyke omgewing (m.b.t. databronne en koppelvlakke) teëgekom word.

Die integrasie van data deur die DT-stelsel bied 'n manier waardeur die aantal databronne wat geraadpleeg moet word vir besluitneming, verminder kan word. Terselfdertyd bied die DT-stelsel 'n platform vanwaar addisionele dienste aangebied kan word, beide nou en in die toekoms.

Acknowledgements

No man is an island, and neither is his work. I am filled with gratitude for the numerous people who have supported, shaped, and guided me in the process of working on this thesis. Although not everyone is mentioned, I hope to honour some in this section.

Firstly, Jesus Christ, to whom my life is dedicated out of thankfulness, is really the one who provided me with the vision, the heart, and ability to perform the work presented in this thesis. I am extremely grateful for His reliable and consistent presence, grace, and guidance that He has so often given to me in love. This work has been done for Him.

My parents, brother, and sister have played a massive role in supporting and encouraging me throughout. Knowing and experiencing their love, and the perspective that they often gave, has been a reliable source of strength during times when I was tired, discouraged, or demotivated. I also thank them for their genuine joy and celebration in the small, as well as large victories - it has brought me great encouragement throughout. My girlfriend, Danielle, has also been such a God-sent source of encouragement, help, and support without whom I wouldn't have been able to be where I am today.

My friends and the other MAD Research Group members have inspired me and kept me accountable to excellence in all areas of life. It has been a privilege to live out the different areas of life together with them, and it has shaped and formed me in many ways that I am very grateful for.

I also want to acknowledge the PRASA Research chair for their significant contributions to this research in the form of providing the initial significant financial support, and facilitating interactions between us, as researchers and PRASA. I also want to specifically acknowledge the staff at Salt River – obtaining the necessary relevant data for my case study was a key element of this research, and their friendliness, assistance, and willingness to share information with me really made this process an insightful and enjoyable one.

Finally, I want to acknowledge my supervisors – Pieter Conradie, Prof. Basson, and Dr. Kruger, for the tremendous role that they have played in my thinking and work on this thesis. The numerous hours of their time, and their perspective, guidance, and willingness to wrestle with ideas and concepts *with* me, has been integral to the work presented here. I am truly grateful for them and all that I could have learnt from them.

Table of contents

Page

List of figures

List of tables

List of abbreviations

API	Application programming interface
AMQP	Advanced message queuing protocol
BIM	Building information modelling
CBM	Condition-based maintenance
CSV	Comma-separated values
CUI	Central user interface
DC	Design constraint
DT	Digital twin
FPI	Fault path indicator
FR	Functional requirement
IIS	Interaction information set
IoT	Internet of things
ML	Machine learning
MS	Microsoft
MTS	Metrorail technical services
MQTT	Message queuing telemetry transport
NFR	Non-functional requirement
OEM	Original equipment manufacturer
OHTE	Overhead track equipment
PLC	Programmable logic controller
PLM	Product lifecycle management
PRASA	Passenger Rail Agency of South Africa

RTF	Run-to-failure
SG	Service gateway
SLADT	Six-layer architecture for digital twins
SLADTA	Six-layer architecture for digital twins with aggregation
SN	Services network
SOA	Service orientated architecture
SOC	State owned company
SQL	Structured query language
SR	Stakeholder requirement
TFR	Transnet Freight Rail
TGIV	Track geometry inspection vehicle
UI	User interface
USB	Universal serial bus
WL	Wrapper layer

1 Introduction

This chapter provides a background on digital twins (DTs) and railway infrastructure in South Africa (Section 1.1). This will guide the readers' understanding of the research objectives, presented in Section 1.2. These objectives are subsequently motivated in Section 1.3, after which the chapter concludes in Section 1.4 with the methodology with which this research has been conducted. The methodology is accompanied by an overview of the structure of the thesis.

1.1 Background

The South African rail transport sector is recovering from a period of financial neglect (Japan International Cooperation Agency, 2013; Papatolios, 2021) where obsolete facilities and aging rolling stock cause frequent failures, safety problems and a generally unreliable service to commuters (Japan International Cooperation Agency, 2013). Despite population growth, the World Bank reports a decline of 68% in passenger trips per annum in South Africa from 2009 to 2019. This situation has been exacerbated by widespread theft and damage to assets during the national lockdown in 2020. (World Bank, International Finance Corporation and Multilateral Investment Guarantee Agency, 2021). In an effort to recover the damage, the South African government is investing a sum of roughly R170 billion over a period of 10 years into the transport sector (Ngqakamba, 2019) to restore and "modernise" the rail infrastructure of the country.

The management of rail infrastructure in South Africa is governed by two state-owned enterprises, namely the Passenger Rail Agency of South Africa (PRASA), and Transnet SOC Ltd or, more specifically, the Transnet Freight Rail (TFR) division. PRASA is responsible for all *passenger transport operations* other than that of the luxurious Blue Train, for which TFR is responsible. TFR is also responsible for all *freight transport operations* via rail in the country. Furthermore, Transnet owns and maintains the *intercity* rail infrastructure, while PRASA owns and maintains most of the *urban* (*intracity*) rail infrastructure.

In 2009, PRASA established a research chair in the Department of Industrial Engineering at Stellenbosch University, to help solve the complex and pressing issues faced by the rail industry. One such issue is the present need to *integrate and manage the masses of data* collected from the infrastructure. In their most recent Corporate Plan, PRASA published the need to "Strengthen and secure the *management* of information, documents and records" [emphasis added] (PRASA, 2021). Addressing this need is a challenge that is exacerbated by the digitisation performed by the modernisation program – adding new, different, and often proprietary data acquisition systems to the network. PRASA acknowledges this by stating the need to "Shift from silos to the enterprise-wide services model (central ICT unit)" (PRASA, 2021). Single measurement instances can amass data of the order of gigabytes, which are not exempt from subsequent manual processing. This makes the effective digestion of data an extremely time-consuming process.

While there are various models and frameworks available for dealing with data management and integration, a specific tool that has seen great interest in recent years is the concept of a digital twin (DT). Its value for specifically the integration of data has seen numerous publications (Fernando *et al.*, 2018; van der Valk *et al.*, 2020). This study will focus on the development of such DTs. It should be noted that, with research on DTs increasing significantly in recent years, the definition of DTs has taken on multiple variations (Fuller *et al.*, 2020). The definition used in this thesis, will be a combination of the definitions used for DTs in the context of the built environment (Johnson *et al.*, 2021), and for road and rail infrastructure (Vieira *et al.*, 2022)**:** a DT is seen as *a digital representation of a physical entity (asset or asset system) that integrates different static and dynamic data, as well as different support tools such as physical models, data analytics, simulation, and prediction capabilities. This can then be used to generate insights for physical asset management decision-making.*

This research considers a DT *system,* drawing strongly from the DT research of Human (2022) and Redelinghuys (2020) who advocated for the use such approaches to DT implementation. *A DT system is taken to refer to a hierarchy of DTs with supporting components, that work together to represent a complex physical system for a user*.

Application domains such as healthcare, manufacturing and smart city environments (Fuller *et al.*, 2020) has seen extensive DT research, but research on the application of DTs to railway infrastructure is still in its infancy (Doubell *et al.*, 2021). Together with this, very little research has been published to provide guidelines and details for DT *implementation* (Sharma *et al.*, 2020).

The Mechatronics, Automation and Design Research Group in the Department of Mechanical and Mechatronic Engineering at Stellenbosch University has conducted several research projects towards the design, development and implementation of DTs in recent years (Redelinghuys, Basson and Kruger, 2018; Redelinghuys, Kruger and Basson, 2020; Human, Basson and Kruger, 2021; Kruger, Human and Basson, 2022). This study forms part of a collective effort from the research group towards the implementation and evaluation of DTs for complex systems.

1.2 Objectives

With the given background in mind, the research objective of this thesis is:

To develop a technology demonstrator of a DT system for PRASA railway infrastructure, where the infrastructure is in its operational lifecycle phase.

The technology demonstrator is designed and evaluated with the focus on use cases that address PRASA's needs relating to *maintenance management* for a representative section of their railway infrastructure network. Although the design and development focuses on the mentioned use cases, the technology demonstrator's functionality should also illustrate its suitability for the rest of PRASA's railway network.

The DT system technology demonstrator is not a commercial solution, but should demonstrate the ability to facilitate the *reconfigurable integration* of different (possibly even temporary) infrastructural elements' digital representations, each with diverse (and possibly changing) data sources. It should also demonstrate the ability to support end users with different data needs. In other words, the technology demonstrator is aimed at being able to facilitate the adaptation of the DTs within the software system, so as to reflect the changes that occurred to the physical system over time (relating to infrastructure elements or data sources being added or removed) and provide the user with access to the data so that it can be utilised in different ways.

1.3 Motivation

Section 1.1, elaborated in Chapter 3, mentions the integration of data as a major need for PRASA's railway infrastructure management. DT literature often refers to the integration of data as a major advantage of DTs (Melesse, Di Pasquale and Riemma, 2020; Barricelli, Casiraghi and Fogli, 2019). However, as mentioned in Section 1.1, there is a current need for literature on the actual implementation of DTs. These two arguments provide the basis for motivating why this research is conducted – to provide a tool for PRASA to perform their data integration, and to contribute to the literature on the implementation details of DTs.

A railway infrastructure network is a complex system with numerous components and subsystems. At PRASA, these systems are managed largely independently of one another. As indicated in the Section 1.1, a system of DTs with software services could reflect the reality of the entire railway infrastructure network and, by integrating information from various physical subsystems, better support decisions required during operation.

The value of DTs for integrating data has mostly been discussed conceptually in reported literature. This thesis contributes to realising the data integration of DTs in practice. By developing a DT system technology demonstrator for PRASA, the value of such a system of DTs, particularly in the context of integrating data from various sources, can be validated. The implementation would also provide PRASA with an indication of the value that they can expect from such a system.

Maintenance management is, as will be shown in Chapter 3, the main use that PRASA has for data from the railway infrastructure (practically all of which is currently in its operational lifecycle phase). Effective maintenance management processes require data, leading to maintenance management being one of the most cited use cases for DTs (Szpytko, Yorlandys and Duarte, 2021). This use case is, therefore, a sensible choice to provide PRASA with an indication of the value of DT technology for their context.

With the railway infrastructure finding itself in the middle of a modernisation program, it is to be expected that the infrastructure, or at least the data sources and management systems, will change or additional elements would be added, such as new sensors, new infrastructural inspection services, new signalling systems with their corresponding data logging systems, etc. The DT system would have to adapt accordingly, to maintain the representation of the physical infrastructure network as it exists.

Given the nature of a DT, it would also be difficult to accurately predict all the future use cases that one might have for the system. The reconfigurability of the system to its changing context – to be able to provide new and different services to the users as their requirements evolve – is therefore a desirable characteristic of which the implementation will be explored in this research. This will shed some light on how one can develop a DT system to facilitate these types of reconfigurations.

1.4 Methodology and thesis overview

The development of the DT system technology demonstrator is performed in four phases, namely: design requirement identification, design, implementation, and evaluation through a case study implementation. Recommendations given by the design framework of Human (2022) (mentioned upon in Section 2.5) were strongly considered throughout this process.

Subsequent to this introductory chapter, the thesis presents a literature review of the research on DT technology for railway infrastructure, as well as implementations in industry to date. This provides perspective on the concept of a DT and a DT system, as well as the application and implementation thereof to the context of railway infrastructure.

The needs analysis, reported in Chapter 3, was based on a series of interviews and discussions with PRASA infrastructure management and technical staff at the regional management office for Western Cape railway infrastructure. The needs analysis considered PRASA's current context in terms of the properties of the digital infrastructural data sources (availability, acquisition methodologies, data storage and transmission characteristics, etc.), and subsequently also reports the expressed and implied stakeholder requirements from stakeholders in this context. The expressed and implied requirements provide the basis for the design requirements for a DT system in the context of PRASA railway infrastructure maintenance management.

Given the design requirements, a general software architecture for a DT for this context is developed in Chapter 4. The architectural development forms the design phase of the development of the technology demonstrator, addressing the requirements of Chapter 3, within the PRASA railway infrastructure context. The architectural development starts from a reference architecture for DT systems, presented by Human (2022). This architecture is an extension of the six-layer architecture for DTs with aggregation (SLADTA), presented by Redelinghuys, Kruger and Basson (2020), and was shown to be suitable to complex systems by Human (2022).

Having established the general software architecture, a specific implementation of that architecture is reported in the case study in Chapter 5. The scope of the DT system technology demonstrator is first determined, leading to a description of the specific implementation architecture and other details. The DT system is then evaluated in Chapter 6, based on the case study implementation of the technology demonstrator in Chapter 5. The technology demonstrator is subjected to three tests in the evaluation, to verify its functionality and evaluate its design. The tests use real data obtained from PRASA, but since no real-time data was available from the actual infrastructure, the "real-time" nature of the updated data was simulated.

Finally, the thesis concludes in Chapter 6 with a discussion that reflects on the objectives set out in Section 1.2.

2 Literature review

This chapter presents the literature review, to establish the research context for this study. The literature review is structured to firstly explore the current understanding of the *concept* of a DT – clarifying the concept in the given context of railway infrastructure in Section 2.1. The *application* of DTs within this context is then discussed in terms of value presented by DTs within this context (in Section 2.2). The latter discussion explores the conceptual value, as well the implementations of DTs within the context of railway infrastructure.

This then leads to a brief exploration of literature on the role of DTs to integrate data, and how DTs for the integration of data from various sources (specifically "data silos") have been implemented (in Section 2.3).

Finally, the two DT system reference architectures that influenced the development of the DT system presented in this thesis, are presented in Sections 2.4 and 2.5, respectively.

2.1 The digital twin concept

One of the first mentions of the concept of a DT was by Grieves in his presentation about product lifecycle management (PLM) in 2002. It was introduced as the "Conceptual Ideal for PLM", with Figure 1, below illustrating the concept presented. (Grieves and Vickers, 2017)

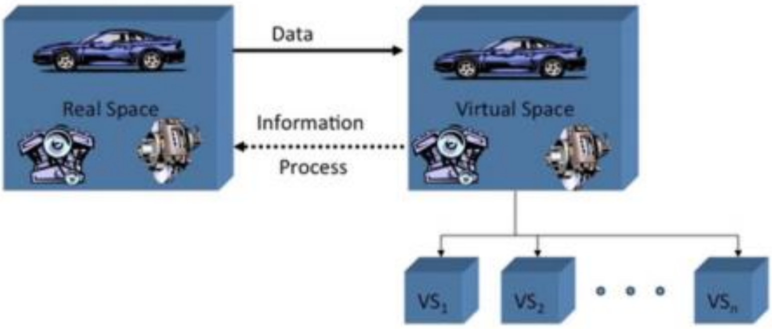


Figure 1: The conceptual ideal for PLM - a front-runner for the digital twin concept (Grieves and Vickers, 2017)

The concept presented all the elements that would come to be associated with a DT: the "real space", representing the physical entity or "twin"; the "virtual space", containing a virtual (digital) representation of the physical entity in the "real space", as well as data and information flows from the real space to the virtual space and back. Information flow is also described from the virtual space to "virtual sub-spaces" (Grieves and Vickers, 2017). A more prominent initial use of the concept was by NASA technology roadmaps in 2010 (Shafto *et al.*, 2010), where multiple uses of the term can be found. NASA defined a DT as "an integrated multi-physics, multiscale, probabilistic simulation of a system that uses the best available physical models, sensor updates, fleet history, etc. to mirror the life of its flying twin" (Shafto *et al.*, 2010). This definition was used in the context of astronautics and aerospace, where the DT referred to a virtual representation of satellites – inaccessible for inspection or scenario investigation once launched.

As the application context of DTs broadened, the initial conceptual ideal of DTs for PLM became more feasible, with DTs in different domains emphasizing the different roles and functions that it can be used for - from production processes and systems in the manufacturing industry (Overton and Brigham, 2017) through to product design (Tao *et al.*, 2019). However, almost every publication on DTs currently mention the disparate definitions of DTs being used, with the contexts and use of the DTs greatly influencing how authors describe and use the term (Vieira *et al.*, 2022).

With DT research growing, the typical application of DTs has been formed by an “internet of things” (IoT) perspective (Fuller *et al.*, 2020), with high data velocity applications. However, with the move towards “smart infrastructure”, research towards the application of DTs in the context of domains such as civil infrastructure and facilities management are increasing (Callcut *et al.*, 2021; Marocco and Garofolo, 2021; Villa *et al.*, 2021). Railway infrastructure, as will be illustrated in Chapter 3, is a good example of this - where measurements are taken in periodic intervals that can span up to 6 months between samples.

However, in the context of civil infrastructure and facilities management, where more matured technologies such as Building Information Modelling (BIM) exist, the concept and definition of DTs are broadened even further and it becomes difficult to distinguish between DTs and technologies such as BIM. Kritzinger, *et al.* (2018) assists with the differentiation of these concepts by presenting the concept of a DT, in contrast to what they referred to as the “Digital Shadow” and “Digital Model”. Figure 2 below, illustrates the difference as determined by the nature of data flow between the physical and digital entities.

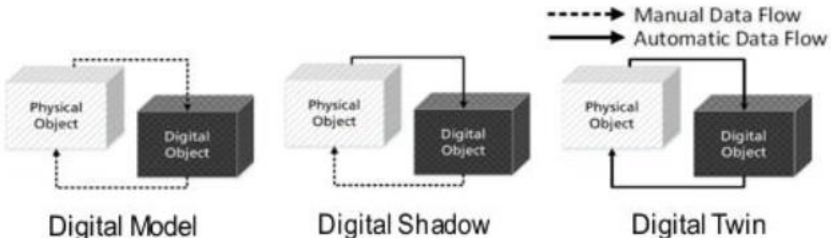


Figure 2: Evolution of the digital twin (Tchana, Ducellier and Remy, 2019, as adapted from Kritzinger *et al.*, 2018)

BIM technology is typically implemented as an example of what can be seen as a “Digital Model”. Here the information model (digital entity) is manually updated with data entries. Appropriate personnel can then subsequently act upon the data in the information model (a “manual” data flow back to the physical entity). A form of DT can be created by combining BIM technology with IoT technologies (such as wireless sensor networks) that enable automatic data transmission from the physical to the digital entity (Marzouk and Abdelaty, 2014; Gao *et al.*, 2021; Marocco and Garofolo, 2021). According to Kritzinger, *et al.* (2018), these would be classified as “Digital Shadows”. Further evolution, with automatic data transmission back to the physical entity as well, will then result in what can be called a “Digital Twin”.

This definition of automatic data flow from the physical to the digital and back corresponds well the original idea as presented by Grieves (2017). However, very few definitions of DTs found in literature require this automatic data flow from the digital back to the physical. Most literature would define both the “Digital Twin” and “Digital Shadow”, as presented by Kritzinger, *et al.* (2018), as a “Digital Twin” (Fuller *et al.*, 2020; Xiaodong *et al.*, 2020). For instance, the integration of BIM with IoT technologies are typically considered as “Digital Twins” (Marzouk and Abdelaty, 2014; Gao *et al.*, 2021; Kaewunruen, Sresakoolchai and Lin, 2021; Marocco and Garofolo, 2021). However, BIM itself is typically *not* regarded the same as DT technology (Boje *et al.*, 2020; Callcut *et al.*, 2021; Villa *et al.*, 2021; Yu *et al.*, 2021).

Among the little research that has been published on the implementation details of DTs (Chokshi and McFarlane, 2008; Sharma *et al.*, 2020), different reference architectures, paradigms and implementation considerations can be seen. The shortage of implementation details for DTs, highlight the possibility of even more diverse implementations.

Finally, it is worth noting that research towards DTs is continuing to increase (Fuller *et al.*, 2020), with more than 500 publications having been made on DTs in industrial operations by 2019 (Melesse, Di Pasquale and Riemma, 2020). This illustrates the increasing interest in the concept.

2.2 Digital twins in railway infrastructure

2.2.1 The value of digital twins for railway infrastructure

Research towards the application of DT technology to rail infrastructure is still in its infancy, with limited research on the subject available to date (Doubell *et al.*, 2021; Vieira *et al.*, 2022). However, the available literature makes a case for the value that it can offer (Vieira *et al.*, 2022).

Rail infrastructure is considered critical to economic growth (Love *et al.*, 2017). With this, it is also considered a complex system (Gurdur Broo and Schooling, 2021), with diverse data sources (Johnson *et al.*, 2021), stakeholders (Kaewunruen, Sresakoolchai and Lin, 2021), and assets (that often also overlap with other sectors of activity such as power, telecommunications networks) (Vieira *et al.*, 2022). It's complex nature is also seen in the fact that it is a "system of systems" (Gurdur Broo and Schooling, 2021). Adding to this, railway infrastructure is also managed by humans, which, according to Vickers and Grieves (2017), adds another layer of complexity. However, DTs are conceptually ideally suited to such (complex) systems (Vickers and Grieves, 2017).

Moving from the characteristics to the use of railway infrastructure, it is noted that the demand for these networks are increasing, leading to maintenance backlogs and a need for increased investment (OECD, 2020). Deficiencies and aging infrastructure are also typically seen (OECD, 2019). As mentioned in Section 1.1, this is especially true for the South African context. Effective management of these issues, as noted by Vieira *et al.* (2022), require objective data and tools to use that accurately assess the cost, risk, and performance of the infrastructure (Costin *et al.*, 2018).

Kaewunruen, Sresakoolchai and Lin (2021) consider the DT as an information platform that can improve *asset management effectiveness* by presenting stakeholders with the opportunity to collaborate, co-create, and view the latest information. Research presented by Vieira *et al.* (2022) agrees with this – concluding that the application of DTs, in the context of rail and road infrastructures, provide a means to address challenges related to *sustainability* and *resilience* of the infrastructure. This is due to their ability to monitor the condition and performance of the infrastructure, and the resulting quality of long-term decision making that is enabled through DTs (Vieira *et al.*, 2022).

2.2.2 Application of digital twins within railway infrastructure

The application of DTs within the railway context is diverse. The following discussion highlights the focus and extent to which DTs have been applied to railway infrastructure.

BIM forms an integral component of current literature on the application of DTs. Many DT applications within this context consider BIM as the basis for constructing DTs, where DTs are formed through the integration of cloud computing, BIM, and IoT technologies (Marocco and Garofolo, 2021). The role of BIM within the DT context in railway infrastructure is for data *visualisation* (Marzouk and Abdelaty, 2014; Ye *et al.*, 2019), for *storing* the relevant data (Marzouk and Abdelaty, 2014), and for *providing services* based on the data (such as calculating the "energy performance" of the assets) (Love and Matthews, 2019).

Marzouk and Abdelaty (2014) consider BIM as a *data source*, distinct from the DT, while other publications use the terms BIM and DT interchangeably (Gao *et al.*, 2021; Kaewunruen, Sresakoolchai and Lin, 2021). In the latter case, the BIM software used, such as Navisworks or Bentley BIM, is responsible for the integration of data from the monitored data sources. In all of these cases, BIM is considered integral to the DTs themselves.

Asset management support services are therefore offered to users through BIM, but can also be offered to users through software modules that just draw the necessary data from a common database. Two examples of the latter will be mentioned for illustration. The first is the high-level systems architecture for smart infrastructure proposed by Gurdur Broo and Schooling (2021) where data from DTs are shared in a "interoperable data layer" (common database) from where it can then be used for various services – enabling "data integration, abstraction, analytics, artificial intelligence applications, visualisation and similar functions" (Gurdur Broo and Schooling, 2021). As another example, Boschert, Heinrich and Rosen (2018) refer to a network of semantically connected elements that form the respective DTs. From this foundation, data analytics services that offer physics-based insights can be implemented using similarity search algorithms (Boschert, Heinrich and Rosen, 2018).

Studies that describe implementations of DTs within railway infrastructure can be classified as having done either an actual implementation (Marzouk and Abdelaty, 2014; Love and Matthews, 2019; Ye *et al.*, 2019; Meixedo *et al.*, 2022) or a conceptual implementation (Boschert, Heinrich and Rosen, 2018; Gurdur Broo and Schooling, 2021; Jiang *et al.*, 2021). Most of these implementations do not use just a single DT, but a system of DTs, with various digital entities, representing the corresponding physical entities, and interacting with one another through information exchange.

DT applications are generally centred around maintenance management (Kaewunruen, Sresakoolchai and Lin, 2021; Vieira *et al.*, 2022), with specific points of interest that can be grouped into three categories: firstly, *prognostics* and health *monitoring* (including simulation and condition monitoring) (Ye *et al.*, 2019; Kochan, 2020; Xiaodong *et al.*, 2020; Ahmadi *et al.*, 2021; Febrianto *et al.*, 2021; Kaewunruen, Sresakoolchai and Lin, 2021; Kampczyk and Dybeł, 2021); secondly, closed *lifecycle management* (Boschert, Heinrich and Rosen, 2018; Tchana, Ducellier and Remy, 2019; Jiang *et al.*, 2021); and, finally, *sustainable* smart infrastructure (Gurdur Broo and Schooling, 2021). These are naturally all related, but the categorisation done above, highlights the focus of the DT applications.

Finally it should be noted that Vieira *et al.* (2022) developed an eligibility criterion for DTs in the context of railway infrastructure (used in their literature review on the subject). Since this is specific to the context of this thesis as well, it will be used in the development of the functional requirements for a DT in Section 3.3. Summarising their requirements, DTs should:

- Serve a specific purpose
- Provide a digital representation of an entity and its context
- Automate data acquisition
- Provide insights, to support asset management decision making, at different levels of aggregation (e.g. component level and network level)

2.2.3 Implementations in industry

Commercial applications of DTs for railway infrastructure are not exempt from the DT definition debate mentioned in Section 2.1. Commercial applications of DTs include Royal HaskoningDHV's DT for ProRail, enabling network analysis and *simulation* on the electrical network (Augusteijn, 2021); the Transpennine Route Upgrade (TRU) programme by Network Rail in Great Britain, that utilised a route-wide DT that provides a central *source of information* from various stakeholders, disciplines and geographical regions (Cockerell, 2022); Siemens' DT of Singapore's downtown line signalling system that provided *simulation* capabilities to enable the testing and integration of the signalling system before deployment, while also aiding in the training signalling operators and other staff (Siemens Mobility, 2020); and Almaviva's DT of the *observed state* of the infrastructure as observed by *visual* sensors (Mangan, 2020).

The ambiguity of the DT definition leads to companies offering "DT solutions" without the consumers necessarily fully understanding the scope of functionality that the DTs can deliver (Callcut *et al.*, 2021). The applications of Siemens and Almaviva will be described further, to illustrate how the interpretation of "digital twin" can differ in industrial applications.

One of the relatively recent applications in the rail industry is Siemens' projects in Singapore – to establish the *Signalling simulation centre* and *Rail Enterprise Asset Management System (REAMS)* for Singapore's downtown line. These projects were implemented so that the DT can aid with faster and more in-depth analyses of the network state, enhanced testing (on the DT - without interrupting rail operations), increased capability for staff training, and improved operational efficiency of the network (Siemens Mobility, 2018, 2020).

Another example of DT implementation in rail industry, is AlmavivA's massive DT project in Italy. The project entails the creation of a DT for the infrastructure of the entire Italian rail network which includes more than 16,000 km of track. The DT of the network will contain visual component information obtained through

GPS and camera sensors. The DT will be able to provide *"back-and-forth feedback between the twin and its real-world counterpart"* (Mangan, 2020), thereby fitting in well with the definition of a DT presented by Kritzinger, *et al.* (2018).

These two industrial implementations show how different the implementations of a "DT" can be, with the AlmavivA implementation having a greater focus on the visual representation, while the Siemens implementation incorporates network and asset behaviours.

2.3 Digital twins to integrate data

Before a discussion on the integration of data from disparate data sources, it is necessary to differentiate between DTs' ability to integrate from sources that are *heterogeneous in type* (such as temperature, acceleration and GPS data), and their ability to integrate *data silos* (segregated groups of data, stored in multiple enterprise applications (Patel and Member, 2019)). Though similar to one another, the issue of data silos revolves around the accessibility of the data for various use cases, instead of the nature (or degree of heterogeneity) of the data itself.

DTs claim to be able to integrate both heterogeneous data and data silos (Xiaodong *et al.*, 2020; Malakuti, Borrison, *et al.*, 2021), but the distinction is generally not made clear in literature. It should be noted, however, that integration of data from various sources in general, is one of the most cited requirements for DTs in literature (Fernando *et al.*, 2018).

As discussed by Doubell *et al.* (2022), the early authors on the concept of the DT present organisational data silos as "probably the biggest obstacle to the Digital Twin", since a DT "requires a homogeneous perspective of this information that persists across functional boundaries" (Vickers and Grieves, 2017). This homogenisation (or integration) of data silos is a general issue, with literature offering solutions such semantic profiling (Mazayev, Martins and Correia, 2017), Big Data Lakes (Patel and Member, 2019), and mobile data management platforms (Brodt *et al.*, 2011).

The integration of data silos through DTs are often implied, but this is mostly from the perspective of DTs as a means to *prevent the creation of more data silos in the future,* such as in Kaewunruen, Sresakoolchai and Lin (2021), and Gurdur Broo and Schooling (2021). The alternative, of DTs as a means to *integrate existing data silos,* as implemented by Malakuti, Juhlin, *et al.* (2021), is rare.

The data of many industrial systems are already stored and maintained in silos (Malakuti, Juhlin, *et al.*, 2021). This is also the case for the context of this study, as shown in Chapter 3. The challenge faced by DT implementations in this context, is to avoid creating another (larger) data silo, or a so-called “solution silo” (Malakuti, 2021; Malakuti, Juhlin, *et al.*, 2021).

A relatively recent literature review on DTs, by Sharma *et al.* (2020) suggest that one of the barriers to the advancement of the DT applications, is the lack of publicly available DT implementation details (Sharma *et al.*, 2020). This led to research on the *implementation* of a cloud-based architecture for DTs, based on “a common information meta-model” (Malakuti, 2021; Malakuti, Borrison, *et al.*, 2021; Malakuti, Juhlin, *et al.*, 2021). Here the DTs essentially provide a single point of access, with a known interface (HTTP/REST was used in their case studies) for data to which it is connected. This reduces the maximum number of communication paths required for *m* applications (services) and *n* data sources, from m*n (without the DT) to *m+n* (with the DT). The former requires every application to potentially communicate with each data source that it will require data from. In the latter case, everything just communicates with the DT. Figure 3 below, illustrates the latter case.

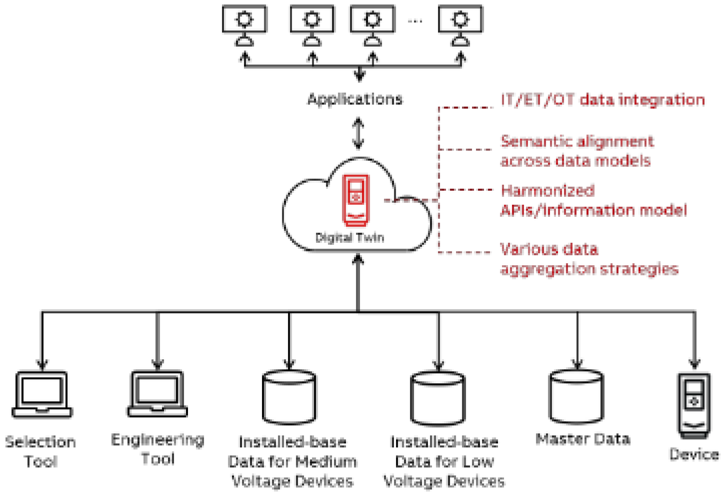


Figure 3: Role of the Digital Twin, as proposed by Malakuti, Juhlin, *et al.* (2021)

The DT implemented by Malakuti, Juhlin, *et al.* (2021), stores and maps the information from the data sources to parameters for which it (the DT) can subsequently be queried for. It therefore filters data from the sources, for use in other applications.

The DT implementation was based on Microsoft (MS) Azure services, with MongoDB and CosmosDB providing storage to data from the data sources, and data processing services being implemented as .NET microservices. Data ingestion into the DT, from devices, happens through advanced message queuing protocol (AMQP) or message queuing telemetry transport (MQTT) protocols and the Azure

IoT Hub service. External to the DT, were applications that execute simulation and machine learning (ML) models were implemented as microservices using FastAPI (Malakuti, Borrison, *et al.*, 2021).

2.4 Six-layer architecture for digital twins with aggregation

DT aggregation presents several advantages, such as separation of concerns, reconfigurability, scalability (Ciavotta, Bettoni and Izzo, 2018; Lutze, 2019; Redelinghuys, 2020), and the ability to present different perspectives of the system (Ciavotta, Bettoni and Izzo, 2018; Borangiu *et al.*, 2020; Villalonga *et al.*, 2021). However, from a recent literature review by Juarez, Botti and Giret (2021) only the reference architectures presented by Borangiu *et al.* (2020) and Redelinghuys, Basson and Kruger (2018) can be seen to utilise aggregation. The latter architecture is referred to as the "six-layer architecture for DTs with aggregation" (SLADTA). A case study, implemented by Human (2022) later verified the subsequent suitability of SLADTA to complex systems. This provides motivation for using SLADTA in the context of this study. Therefore, this section will provide a description of SLADTA, to inform further discussion and use thereof.

The six-layer architecture for digital twins (SLADT) was developed as a reference architecture for DT development within the context of complex manufacturing (Redelinghuys, Basson and Kruger, 2018). The architecture is based on different "Layers" (1 to 6) from the physical to the digital twin. These layers communicate with one another in a bi-directional manner, as illustrated by the connecting lines in Figure 4. The layers (numbered 1 to 6), represent the different high-level components (or component groups) needed for a DT. SLADTA, illustrated in Figure 4, provides then a means to integrate and organise information between these DTs that adhere to SLADT (Redelinghuys, Kruger and Basson, 2020).

SLADTA comprises of digital twin *instances* (DTIs) and *aggregates* (DTAs), as adapted from the definitions of Vickers and Grieves (2017). The DTIs are responsible for data acquisition from the sensors and devices on the physical twin (Layer 1), via physical twin-specific data sources (Layer 2), through local data repositories (Layer 3), to both cloud-based storage (Layer 5) and the user (through services (Layer 6) and a dashboard). DTAs aggregate information from different DTIs and other DTAs and do not obtain information from the data sources themselves. DTAs can therefore be seen to have no Layers 1 and 2 in Figure 4.

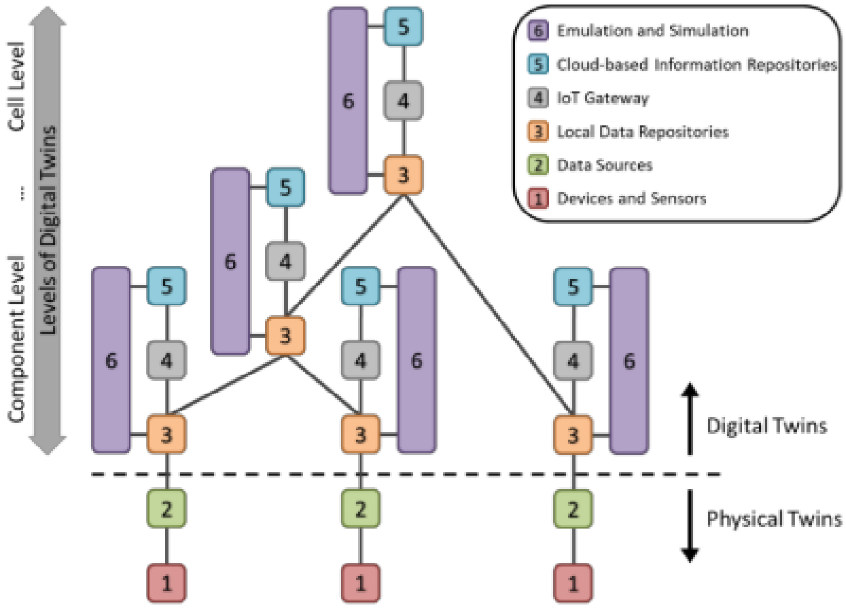


Figure 4: Connection architecture for SLADTA (Redelinghuys, Kruger and Basson, 2020)

Layer 1 in SLADT represents physical devices that can provide or consume signals that are exchanged with the local controller (such as sensors or actuators). The local controllers (such as a programmable logic controller (PLC)) serve as data sources to the DT and constitute Layer 2. Components categorised in Layer 2, may also be used to provide some functionality specific to the physical twin (such as sending control and actuation signals). Data is then typically communicated from here to Layer 3, which is the local data repositories, while Layer 4 is the IoT Gateway. The IoT Gateway constitutes custom-developed software that links Layers 3 and 5 and serves as the data-to-information conversion layer. It receives data, adds context if necessary, and sends the information to the cloud repository (Layer 5). Layer 5 stores the information in a long-term database. Layer 6 is responsible for providing the user with the required services based on the available information contained in the other layers. Layer 6 can connect to multiple layers in SLADT to obtain the information. Typical examples of services that can be provided here are system monitoring, data visualization, data analysis and simulation (Minerva, Lee and Crespi, 2020).

It is interesting to note that SLADTA exhibits a "fractal nature" – with different DTs forming part of one another, with similar internal architectures (SLADT). This fractal nature allows for a "separation of concerns" by encapsulating decision-making in the DT(s) that have access to that information. It also exhibits desirable characteristics such as modularity, flexibility, and reconfigurable aggregation.

2.5 Reference architecture for digital twins with aggregation and services

Human (2022) developed a “design framework to guide the detailed design of a DT aggregation architecture, or a system of DTs, to reflect complex systems” as a result of the value seen in DT aggregation (mentioned in Section 2.4). This design framework (independent to the domain of application) is intended to guide DT system developers to “design a detailed architecture for a DT aggregation hierarchy that reflects a complex system within a given domain”. This also provides a means through which various software quality attributes, and trade-offs for DT system design can be considered.

The design framework uses a general reference architecture (shown in Figure 5) proposed by Human (2022) to implement the different design decisions. The reference architecture is not limited only to DTs (in an *aggregation hierarchy*) but includes the supporting software components needed to provide services that address user needs. These are provided in the form of a *services network* (SN) and *management services*. Other major supporting components are the *gateway service* (G), and the *central user interface* (CUI). At the time of the study no comparable reference architectures that accommodate DT aggregation with services, could be found.

Human (2022) proposed the use of SLADT for the DTs’ internal architecture, with the DT aggregation hierarchy taking the form of SLADTA. The “Data”, “Models” and “Services” noted in the DTs in Figure 5 refer to the data capturing performed by the DTs, and system modelling and services performed by SLADT’s Layer 6.

The SN is a set of services distinct from those provided by the DTs in the aggregation hierarchy. SN services utilise the information provided by DTs (through Layer 6), by other services within the SN, and/or by external data sources and services (that do not originate from the physical system, but influences the management of the physical system, such as weather services, financial data, or equipment inventory). Human (2022) expects the SN to follow a microservices or service-orientated architecture (SOA).

The services within the SN might require orchestration when interacting with one another (where multiple functions need to be performed consecutively to achieve a certain outcome). This leads to the necessity of an *Orchestration service*. The *Orchestration service* forms part of the third major grouping, namely management services. These services were derived from published research on services in the context of distributed computing environments (Ciavotta *et al.*, 2017, 2020; Sanjay Gadge and Vijaya Kotwani, 2017; Taibi, Lenarduzzi and Pahl, 2018; Kuhn, Schnicke and Oliveira Antonino, 2020).

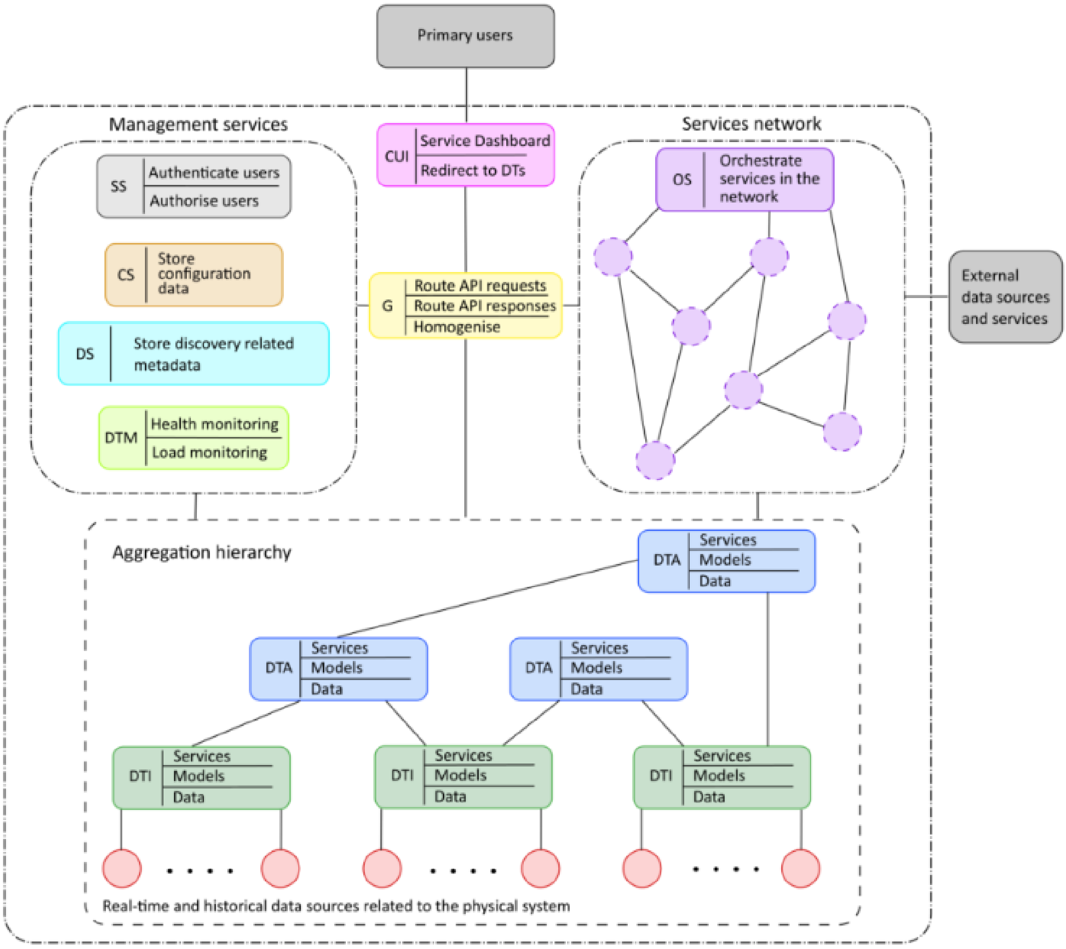


Figure 5: Digital twin system reference architecture proposed by Human (2022)

The *central user interface* (CUI) provides the means through which users of the system will interact with and obtain services from the respective DT(s) and associated service(s). It therefore serves as a flexible dashboard to display data for other services (from the SN or DTs).

The *Gateway* service (G) is responsible for enabling and directing interactions between components in the DT system. The gateway is intended to simplify communications, reducing the number of requests made by client components by negating the need for service discovery logic. Serving as the central point for all interactions, the gateway also enables service interoperability, but presents the risk of becoming a communications bottleneck.

The *directory service* (DS) provides a central repository for component metadata. This service can be queried for discovery information (such as an IP address or related messaging topic) about DTs and SN services within the system. DT and service components must register themselves when they start up or a log must be manually maintained that provides all the relevant metadata of available components in the system.

The *security service* (SS) performs all authentication and authorisation functionality within the DT system. This reduces the need to implement network security in each individual service, as authentication and authorisation services can all be delegated to this one, central service. The *DT monitoring* service (DTM) monitors the health and computational load experienced by individual components (not necessarily just DTs). The purpose is to ensure maximum availability of components within the system.

Finally, the *configuration server* (CS) contains start-up and operational configuration settings for the DT system components. This allows for “automatic deployment and redeployment of digital components and it makes the reconfiguration of distributed systems much easier”.

2.6 Concluding discussion

The literature review presents a conceptual understanding of DTs in Section 2.1 – clarifying the concept in the given context of railway infrastructure where BIM is an established technology. BIM itself should be seen as a “Digital Model”. However, when combined with IoT technologies (such as wireless sensor networks), the combination is seen as a DT, since these would include automatic data transmission from the physical entity to its digital representation in the BIM application. Even though differences on the definition of DTs abound, its use as a digital representation of a physical entity, with automated data flow from the former to the latter is often seen in literature. However, the lack of a universally agreed-upon definition for DTs is seen to permeate into the different facets of its use and design, leading to a wide range of interpretations and applications of the concept.

Section 2.2 highlights that research on the application of DTs to railway infrastructure is noted to still be in its infancy. However, the value of DTs for railway infrastructure is emphasised throughout current literature on the subject. Therefore, authors anticipate an increase in research in this domain. One reason for this is that the effective management of infrastructural health issues, caused by deficiencies and aging infrastructure, require objective data and tools that can accurately assess the cost, risk, and performance of the infrastructure. The integration of data through DTs presents a promising possibility of just such a tool.

DT research in this context generally focuses on the value to be obtained (conceptually), rather than the means through which it ought to be implemented. This highlights a need for research that elaborates on the implementation of DT systems within the context of railway infrastructure, e.g. through case studies.

Other than the effective management of infrastructural health issues, potential value for DTs in the context of railway infrastructure are seen to be the improved sustainability and resilience of the infrastructure, and the opportunity for stakeholders to collaborate, co-create, and view the latest information of the infrastructure.

A significant discovery from the literature review is that the application of DTs to railway infrastructure generally do not use just a single DT, but rather a system of DTs. It is also seen that the application of DTs are generally focused on maintenance management. The overlap between DTs and BIM is specifically observed in the railway infrastructure context, where BIM forms an integral part of nearly all DT literature. BIM is seen as the basis for constructing DTs in this context.

The role of DTs in integrating data from various sources are subsequently discussed in Section 2.3, with specific attention to implementations where integration occurs between different data silos. The major contributions identified in literature on the implementation of DTs for the integration of data silos, utilise a common information meta-model. By implementing these, the authors illustrate how DTs can remain use-case agnostic by providing services in the form of microservices while the DT just maintains the semantics of its data. These implementations also show that services such as ML and simulation can be implemented as microservices in such a scenario and do not need to be built into the DT itself.

Finally, two vendor-neutral reference architectures for DT systems are presented in Sections 2.4 and 2.5. These architectures capitalise on the value of aggregation in the context of complex systems (such as railway infrastructure) and have been validated through case studies. The latter reference architecture (discussed in Section 2.5) expands on SLADTA (Section 2.4) and utilises the idea of implementing services as microservices. The latter architecture will provide the basis for the development of the DT system presented in this thesis, as discussed in Chapter 4.

3 Needs analysis

This chapter serves to identify the requirements for a DT system in the context of PRASA's railway infrastructure. This is accomplished through a contextual analysis in Section 3.1, wherein the infrastructural data and the current use thereof is described. This leads to the stakeholder requirements (discussed in Section 3.2), from where the design requirements for a DT system is developed in Section 3.3.

3.1 Contextual analysis

This section will focus the context within which the case study will be performed – the properties and use of PRASA's railway infrastructure data. The methodology with which information for this section was obtained, is first described in Section 3.1.1, after which a brief explanation of the different infrastructural domains is presented in Section 3.1.2, to provide background for the rest of the discussions. The infrastructural data that is available to PRASA is then described in Section 3.1.3, followed by a discussion of how the data is currently being used, in Section 3.1.4.

3.1.1 Scope of the analysis

Most of the information presented in this section was obtained through an exploratory analysis that entailed a series of ad-hoc, semi-structured interviews and discussions with PRASA infrastructure management and technical staff at PRASA's regional management office for Western Cape railway infrastructure (in Salt River, Cape Town). The interviews were conducted during the period of September 2021 to November 2021. To honour the confidentiality agreement between PRASA and Stellenbosch University, no names are referenced in the observations made.

Information for the signalling systems were obtained from PRASA's most recent specification of required operational capability for their signalling systems, specified in July 2019. The signalling system for the Western Cape province is currently being replaced by third-party contractors as part of the modernisation program. The required operational capability stipulate the general, operational, and functional requirements that the newly installed signalling systems should adhere to.

The interviews and study of the relevant documentation were used to determine:

1. Existing data sources on the network
2. Data storage and communication of data from these data sources

3. Typical use of railway infrastructure data

It should be noted that the deployment context described in the following sections is specifically for the Western Cape context. This context was selected due to the ready assistance and accessibility of personnel at the Salt River office, as well as the proximity to Stellenbosch University – easing the process of discussions, interactions, and inspections considerably. However, differences between regions, such as different third-party data providers, different infrastructural network layouts, different suppliers of infrastructural components, etc. are not considered significant for the purposes of this study. This is due to the fact that these differences are accounted for through the requirement of the technology demonstrator to enable “reconfigurable integration” of data (stipulated in Section 1.2). Satisfactory demonstration of the technology demonstrator for the Western Cape, would therefore imply that it's functionality can be extended to other regions as well by reconfiguring the DT system to incorporate the new region’s assets and data sources. Therefore the needs and requirements derived from the Western Cape context are used to develop the design requirements for (Section 3.3) that will guide the implementation of a DT system that will suit other regions as well.

3.1.2 Infrastructural domains

Railway infrastructure can be divided into five domains, namely electrical, signalling, track, civil, and telecommunications infrastructure. However, from the perspective of managing the infrastructure, the track and civil infrastructure are combined to form the “permanent way” or “perway”. This is also how PRASA’s railway infrastructure management division is structured – with Perway, Electrical, Signalling and Telecommunications being the names of the different infrastructural management departments.

Main components of the perway infrastructure are the tracks, sleepers, and ballast. The electrical infrastructure comprises of the electrical distribution network, the substations, and the overhead track equipment (OHTE) that then provide power to the trains through the catenary wires. Signalling infrastructure includes all the components that are involved with directing railway traffic, such as the interlocking system, occupancy detection devices (e.g. axle counters) signals, point machines, etc.

Telecommunication systems specifically refer to the railway telecommunications infrastructure, and include the rail-bound fibreoptic communications, station-bound communications, closed-circuit television, radio, alarm, and similar systems. The fibre-optics form the backbone of all communications along the railway network - communicating commands within the signalling system and to the circuit-breakers in the electrical substations.

3.1.3 Infrastructural data sources

Nearly all PRASA's data on railway infrastructure is provided by external or third-party contractors. The provision of the data is, therefore, subject to proprietary implementations and agreements between these entities and PRASA. The observations made regarding third party data providers in this study, is therefore only applicable for providers during the period of September 2021 to November 2021.

Section 3.1.4 will describe PRASA's maintenance management processes in more detail. However, for now it is necessary to know that PRASA uses a condition-based maintenance (CBM) tactic. CBM is performed in all infrastructural domains except for the telecommunications infrastructure, which is treated in accordance with a run-to-failure (RTF) maintenance tactic. CBM generates data through feedback from detailed and routine (surface-level) condition inspections that are performed annually and quarterly respectively. Data from these inspections is logged onto a custom-developed structured query language (SQL) database called Empac. Empac keeps a detailed record of the feedback from past condition inspections as well as from corrective (follow-up) work. The purpose of corrective work is to restore or repair the condition of the inspected infrastructure when it gets below an acceptable threshold. In addition to the feedback from *past* (completed) inspections and corrective work, the planned *future* inspections and corrective work for each asset is also stored on Empac.

Data from Empac is obtained through a custom-developed software application called Hyperion. Further details regarding the use and interface to Hyperion were unavailable at the time of the research.

3.1.3.1 Perway domain data sources

Additional to the feedback from condition inspections and corrective work, data for the perway infrastructure is currently also recorded though automated measurements with a track geometry inspection vehicle (TGIV), and telephonic fault reports reported to the Metrorail technical services (MTS). MTS operates similar to a call centre – receiving calls and responding to the reported faults, theft and vandalism immediately by dispatching available maintenance teams telephonically.

The TGIV is operated by third-party contractors who is generally expected to perform quarterly measurements on the tracks. It should be noted that the TGIV also records information on the electrical infrastructure domain's assets (discussed in Section 3.1.3.2). The perway-related data from the TGIV include information on track geometry, rail wear and ballast profile. Measurements are made every 250 mm from Cape Town station on the main railway lines in each corridor. The raw data obtained from these measurements are provided to PRASA on a physical, external hard drive in the form of MS Excel or comma-separated values (CSV) files within a month after the measurements. This is accompanied by more MS Excel files that summarise the key discrepancies and provide use-case specific information.

It should be noted that PRASA engineering staff typically take another 1.5 weeks to determine the appropriate response to this data. A list of roughly 100 perway-related defects per corridor that are furthest outside their respective tolerances (as reported by the TGIV data) is compiled. Note that these are not necessarily the most urgent defects to be attended to, but rather an inherited method of identifying "the worst" defects. As an example of why this is not necessarily the most critical, consider a defect that has a lower deviation, but exist in an operationally critical area. It should be more urgent to repair the defect that is critical to train operations, than it is to repair the other (possibly larger) defects that exist in areas where, for example, trains currently do not operate any way.

Through the course of every day, the staff at MTS take physical note of all the reported faults. The faults are then manually recorded in a book (paper-based) and subsequently formalised into a MS Excel file that is sent monthly as a report to the regional perway maintenance management department. These faults are attended to immediately (as will be discussed in Section 3.1.4). Table 1 summarises the characteristics of the perway-related data sources.

3.1.3.2 Electrical domain data sources

Similar to MTS, the Electrical department at PRASA have an office that receives and responds to fault notifications or theft and vandalism reports on electrical infrastructure – called "Electrical Control". Electrical Control compiles and sends their MS Excel reports daily (as opposed to MTS's monthly reports).

The electrical power characteristics of the distribution network are monitored by a third-party contractor. This is done with "fault path indicators" (FPIs) installed at 7 – 8 km intervals in certain areas. FPIs are devices clamped onto the electrical lines, that monitor these parameters and communicate it to a cloud-based server. The data is available as information provided through an authenticated web-based user interface (UI) as a license-based service to PRASA. When the license expires, data provision through the UI will also terminate, leaving PRASA with only the data up until their last paid period.

The substations at Bellville, Dieprivier, and Fishoek substations each have an energy meter installed and operated by a third-party contractor. This meter measures the electrical power characteristics of the alternating and direct current power in-and-out of the substation, respectively. The energy meters can sample at user-selected intervals. As with the FPI system, this information is communicated to a cloud-based server, from where it can then be viewed through an authenticated web-based UI as a licence-based service. As a miscellaneous point to note for future reference, the Cape Town substation is referred to as “Traction-Main”, where the Eskom power is fed into PRASA’s distribution network.

For the OHTE, the electrical maintenance teams have taken initiative to measure the thickness of the contact wire at least once a year. The contact wire thicknesses are measured right next to the mastpoles, and then again halfway between the two mastpoles (approximately 33.5m apart on typical straight sections, and 25m apart around curves). The results are tabulated on paper. Some of these paper records have been digitized into MS Excel sheets and stored on a local computer.

Finally, just as with perway, the electrical infrastructure is also inspected with the TGIV, and the measurement data is handled in a similar way as with the perway data. Measured parameters include, but are not limited to stagger, clearance, slope, and contact force induced by the contact wire on the pantograph. Table 2 summarises the characteristics of the electrical-related data sources.

Table 1: Perway data sources

Data type	Data source	Update frequency	Data storage	Data acquisition method	Means of accessing the data
Perway geometry	TGIV	Quarterly	Local hard drive (Requires up to 80 GB storage space per measurement run).	Uploaded to the hard drive as measurements are being made.	Through universal serial bus (USB) from the local hard drive.
Visual inspection and maintenance data	Empac	Daily	Empac	Uploaded onto Empac (using Hyperion) by maintenance teams.	Available for access through the Hyperion custom developed web service.
Faults	MTS report	Daily	Local computer	Created by MTS staff and mailed to all relevant managerial staff.	Available from all the people on the mailing list

Table 2: Electrical data sources

Data type	Data source	Update frequency	Data storage	Data acquisition method	Means of accessing the data
Contact wire thicknesses	Contact wire measurement records	At least once per year (preferably twice)	Mostly paper-based with some digitized and stored on a local computer	Manually entered into an MS Excel file that is then kept on the local computer.	On the local computer.
Distribution network power properties	FPI system	Continuously	Service provider stores the data	From a web-service interface (requires authentication)	Data available through web-interface. New data is subject to license payment for that month.
Substation power properties	Substation energy meters	Upon user-set OR every 24 hours (if no exceedance is reported)	Service provider stores the data	From a web-service interface (requires authentication)	Data available through web-interface. New data is subject to license payment for that month.
Visual inspection and maintenance data	Empac	Daily	Empac	Uploaded onto the Empac (using Hyperion) by maintenance teams.	Through Hyperion
Faults	Electrical Control report	Daily	Local computer	Created by Electrical Control staff and mailed to all relevant managerial staff.	Available from all the people on the mailing list
OHTE geometry	TGIV	Quarterly inspections	Local hard drive (Requires 80 GB storage per measurement run)	Uploaded to the hard drive as measurements are being made	Through universal serial bus (USB) from the local hard drive.

3.1.3.3 Signalling domain data sources

Data regarding the signalling system is obtained from the proprietary signalling system (installed by third-party contractors) and MTS reports. The signalling system is required to have a man-machine interface (or UI) from which the signalling system can be operated. Commands from the UI, and responses to those commands (indicating the state of the different infrastructural components, such as the point machines, signals, and occupancy detection sensors) should be logged.

Data is logged to long-term memory (stored on cloud based systems), from where it should be available, though proprietary server-computers, for at least 6 years after it is logged.

The logged data can be accessed by PRASA operators through an authenticated UI installed on a designated, proprietary computer. The proprietary software through which the signalling infrastructure is managed from this computer, has export functionality. This can be used to export fault logs to MS Excel files. Signalling operators provide these files to the signalling maintenance managers on a daily basis (similar to the MTS reports).

Signalling faults are also reported to MTS, who records the faults in a MS Excel file, following the same procedure as for the other domains, reporting to the signalling maintenance managers on a daily basis.

3.1.3.4 Telecommunications domain data sources

Since telecommunications infrastructure is subject to an RTF maintenance tactic, it does not receive routine inspections from where data can continuously be obtained as with the other infrastructural domains. Data on this infrastructural domain is obtained when an asset fails or becomes dysfunctional (referred to as a “fault”). This fault will then either be logged and reported in the proprietary signalling system (which also monitors the telecommunications network through which it sends commands to the signalling infrastructure), or be reported telephonically to MTS, from where the standard reporting process will be followed.

3.1.3.5 Summary of data sources

In summary, the TGIV, UMC, signalling system, FPI system and substation energy meters are all sources of data that provide PRASA with data through third party contractors or companies. Therefore, PRASA generally does not obtain raw data, but rather obtains data as a service – data converted to information either through display on a UI, or through processing (subsequently stored in MS Excel files).

Consequently, PRASA generates and manages very little of its own infrastructural data.

3.1.4 PRASA's typical usage of infrastructural data

The main use of data from the railway infrastructure (apart from train operations) is for *maintenance management*. Maintenance management at PRASA is planned, scheduled, and executed within each of the four respective infrastructural departments respectively (discussed in Section 3.1.2). The functionality of the infrastructure is maintained through the combination of planned ("tactical") and unplanned ("non-tactical") maintenance tasks.

PRASA primarily adheres to a CBM tactic for its planned maintenance. This is with the exception of telecommunications infrastructure which, due to telecommunications equipment's tendency to fail without warning, is attended to by means of a RTF tactic. CBM is performed through routine condition inspections, which might prompt corrective (follow-up) work, also referred to as corrective maintenance tasks, to restore or repair the condition of the inspected infrastructure when it reduces to below an acceptable threshold.

Unplanned maintenance, include responses to asset failure, malfunctions, theft, vandalism or other damage. This type of maintenance is performed through "breakdown" maintenance tasks. It should be noted that corrective work might be required after breakdown maintenance (as is often the case after inspections in CBM).

The following discussion describes the general process within which maintenance teams would be required to execute CBM and breakdown maintenance respectively. It is important to note that there are three sets of people responsible for assigning tasks to maintenance teams: the Maintenance Planning Office (MPO), MTS, and Electrical Control. The first is responsible for CBM-related maintenance tasks, and the other two are responsible for assigning breakdown maintenance tasks to maintenance teams.

For both breakdown as well as CBM tasks, maintenance teams are grouped and deployed primarily according to department and, secondarily, according to region. As an example - there would be a signalling maintenance team that is designated to perform signalling-related maintenance tasks in the area of Cape Town's central business district, while another signalling maintenance team would be responsible for signalling-related maintenance tasks in Simonstown.

3.1.4.1 The condition-based maintenance process

The routine condition inspection tasks for CBM are pre-emptively scheduled by the planning office at the start of an asset's lifecycle, for the whole lifecycle. This is done in consultation with PRASA engineers, and in accordance with the maintenance plan stipulated by the original equipment manufacturer (OEM). These tasks are then uploaded onto Empac, from where it will be used as reference for the CBM of the asset over its lifecycle. PRASA's management of these tasks is described in Figure 6 below.

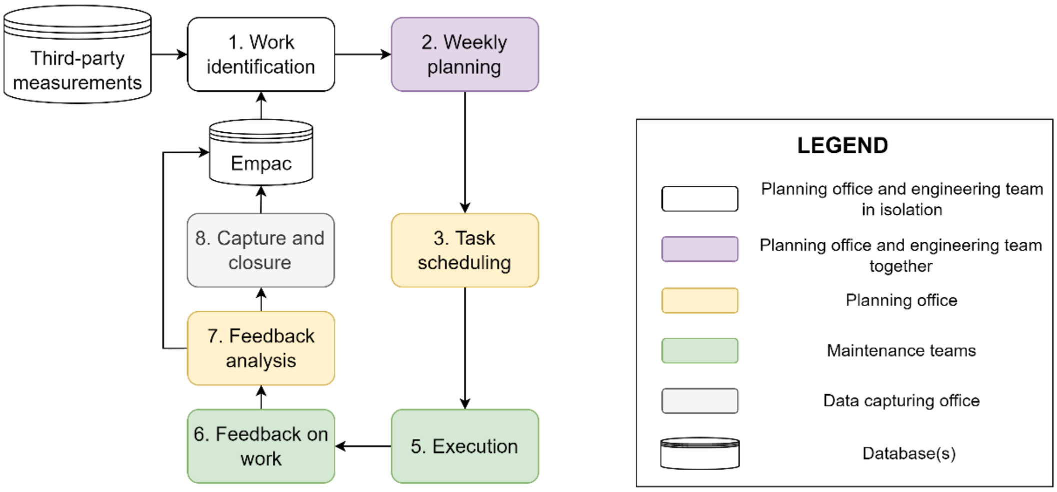


Figure 6: Condition-based maintenance planning, scheduling and execution at PRASA

The maintenance management process depicted in Figure 6 is repeated on a weekly basis. It starts with the MPO and the departmental engineering team each identifying the maintenance tasks considered most urgent for that week. The MPO considers the scheduled condition inspections and corrective work, as reported on Empac. The engineering team considers the third-party measurements (discussed in Section 3.1.3), and derives the required corrective work, from that data.

The MPO, departmental engineering team, *and* technical officer in charge of materials management for that department, then gather for a maintenance planning meeting during which all these tasks are considered. The maintenance planning meetings are purposed to plan and schedule the required maintenance tasks while prioritising the tasks with the greatest effect on train operations. The technical officer is there to provide insight into the current availability of materials for maintenance.

During these meetings, a schedule for the maintenance tasks is established. These tasks are subsequently handed to the relevant maintenance team(s) via paper-based documents called "job charts". The number of job charts that need to be printed over the course of every month, can easily reach up to more than 1000, and the process of printing these job charts can take up to a full week.

After receiving their job charts for the week, the maintenance teams are responsible to request relevant material and tools from the material and tool store, respectively. It is important to note that, even though the materials are known at the scheduling meeting, the availability of the tools are not known. It can therefore happen that two maintenance teams require the same tools on the same day. This complicates matters further since maintenance teams would then have to rearrange their schedules themselves in accordance with the availability of the tools. The tools store manager keeps track of where the tools are and which maintenance team is using it, on a personal MS Excel file.

Having collected the necessary tools, the maintenance teams go out to execute the tasks they were assigned. Maintenance tasks are typically a combination of routine inspection tasks and corrective work. Routine inspection tasks include the specific inspection routes that daily patrolmen should take, monthly visual safety inspections, “trolley inspections” (during which the contact wire measurements are made), and an annual complete condition assessment for each asset.

When, upon inspection, an asset is found to be defective, and maintenance personnel have the appropriate equipment on hand, the asset is repaired through immediate corrective maintenance work. The work performed should be reported as technical details on the job charts. If they are unable to repair the asset, or if they can repair the asset superficially, but know that further maintenance would be required, it should be reported as such on the job charts.

It is important to note here, that the contact wire measurements are made while performing the scheduled trolley inspections. The electrical maintenance teams monitor, record, and respond to these measurements themselves – it is not planned, scheduled, or provided to them through the planning office.

Subsequent to the completion of maintenance tasks, the paper-based job charts are brought back to the planning office. The planning office then analyses the feedback from the maintenance teams and logs the results to Empac. The MPO then uploads the technical details reported, as well as any required corrective maintenance tasks on Empac for execution in a following week. Future corrective maintenance tasks might require the immediate enforcement of temporary interventions, such as to impose speed limits for the trains, until all the defects are dealt with. The MPO should therefore report these corrective actions to the departmental engineering team, who is responsible for making the decisions on these temporary interventions.

After the planning office’s analysis and upload of the feedback through the job charts, the job charts are transferred to the data capture office where all the remaining (non-technical, administrative) details of the completed maintenance tasks are uploaded to Empac.

3.1.4.2 The breakdown maintenance process

As discussed in Section 3.1.3, the deployed maintenance teams or other external sources such as train drivers, safety officers, passengers, pedestrians, etc. report faults to Electrical Control and MTS telephonically. These faults are by nature unexpected, and dealing with them require unplanned ("non-tactical") breakdown maintenance that is not included in the planned CBM activities. The breakdown maintenance process is described in Figure 7 below.

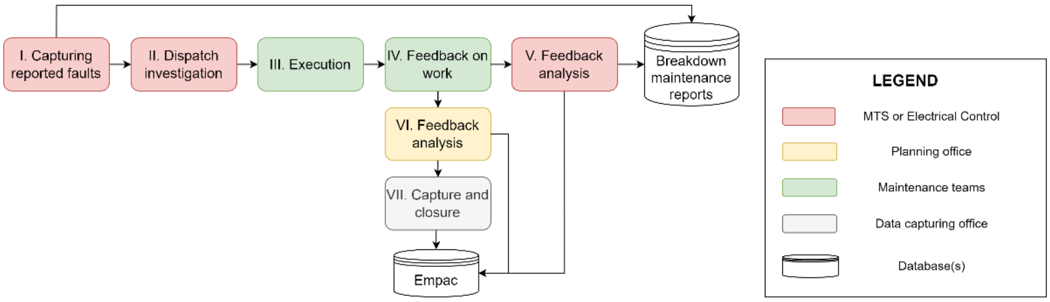


Figure 7: PRASA's breakdown maintenance execution process

After the reporting call, the receiving entity (either MTS or Electrical Control) is responsible to take note of what is reported, for which specific asset, by whom it is reported, etc. A maintenance team would then be ordered to conduct an immediate investigation to assess the criticality of the fault and, if possible, also attend to the issue. Maintenance teams subsequently report their actions and findings to MTS or Electrical Control telephonically, who then captures/logs the feedback on the MS Excel file discussed in Section 3.1.3.

Sometimes the investigation team cannot attend to the fault immediately (e.g., because it requires specialised tools or, upon closer inspection, they find that the initial fault diagnosis was inaccurate). Upon being informed of this, Electrical Control or MTS will log the fault on Empac to be scheduled as future corrective maintenance tasks.

Maintenance teams have a dedicated book within which they record their feedback on the investigations, supplementary to the telephonic reports. This book is delivered to the MPO at the end of each month, to be processed in a similar way as the CBM job charts.

It should be noted that attending these (unscheduled) investigations would often delay maintenance teams from completing their scheduled CBM tasks (inspections or corrective work) for the day. This creates a backlog of maintenance tasks.

3.1.4.3 Summary and observed challenges

The overwhelming number of breakdown maintenance tasks currently encountered, presents a serious concern. The risk is that infrastructure that is critical to train operations may possibly not receive the required attention, i.e. that breakdown maintenance tasks will continually postpone the required CBM tasks. This will lead to degrading infrastructure that in turn leads to more failures and more breakdown maintenance. This unfortunately produces a misalignment between the net maintenance output and the spent resources (financial, human, time, etc.).

Effective management and prioritisation of maintenance tasks would require each respective infrastructural maintenance department to have a holistic view of their infrastructural domain, with insight into all the relevant data sources' data. The weekly maintenance planning meetings provide a platform where this can be achieved through verbal discussion, but the amount of data sources that need to be consolidated in these meetings present a significant challenge.

In identifying and compiling the required corrective maintenance tasks for the week, the respective departments' engineering teams should consider data from 2 sources for perway, 4 sources for electrical, 2 sources for signalling, and 2 sources for telecommunications infrastructure. During the maintenance planning meetings, the scheduled inspections and corrective work that is already loaded on Empac should also be considered. Added to this is the availability of materials and tools for these maintenance tasks that must be taken into account. From all of these separate sources, the different maintenance tasks required should then be prioritised.

Another point to note about the current use of the infrastructural data, is that all of the data sources exist in silos and are used as such – with data only being integrated in discussion between different parties during the weekly maintenance planning meetings. This affects breakdown maintenance especially. For example, if a fibre-optic cable is cut, it will lead to faults in the signalling, telecommunications, as well as electrical infrastructure departments. All three departments will therefore respond, when the only response required was the telecommunications team. Table 3 summarises the use of infrastructural data for maintenance management.

Table 3: Data sources' current use(s) in maintenance management

Data source	Use in maintenance management	Relevant infrastructural domain(s)
Empac	• Scheduled condition inspections and corrective work • Contains remarks on asset-condition as reported by maintenance teams	All
MTS report(s)	Instructing breakdown and corrective maintenance tasks	Perway, signalling, telecommunications
TGIV	Corrective maintenance tasks for the +/-100 largest proportional exceedances	Perway
Contact wire measurements	Corrective maintenance tasks for where critical thicknesses are reached	Electrical
Fault path indicators	Currently not used	Electrical
Substation energy meters	Currently not used	Electrical
Electrical control report(s)	Instructing breakdown and corrective maintenance tasks	Electrical
Proprietary signalling (software) system	Instructing breakdown and corrective work tasks	Signalling and telecommunications
Daily signalling fault logs	Instructing breakdown and corrective work tasks	Signalling

The current maintenance management system, therefore, requires extensive effort to obtain a holistic view of the current condition of the infrastructure. This is even more so if a knowledge of the current condition of *a specific asset* is desired - all the different data sources would have to be considered. After finding the relevant data in each, these different data sources' data would then have to be manually consolidated. This is even more challenging for *historic* data, where measurement dates, will also need to be determined and consolidated across the different data sources.

Therefore, the determination of long-term degradation trends will require immense effort, and this is before the analysis thereof can be done. This presents a barrier to validation and re-evaluation of the current maintenance management practices.

Finally, the whole maintenance management process involves hundreds of pages of job charts, which need to be manually searched through and sorted to distribute to the relevant maintenance teams. This is a tedious and time-consuming process that keeps management staff occupied, reducing their effectiveness in the roles that they are supposed to fulfil.

3.2 Stakeholder requirements

Given the context described in Section 3.1, this section describes the stakeholders' expressed and implied infrastructural maintenance management requirements (in Sections 3.2.1 and 3.2.2 respectively). These requirements describe the stakeholders' expectation for a DT (or similar software) system.

3.2.1 Stakeholder requirements for maintenance management

In light of the context as described in the preceding sections, the following stakeholder requirements ensue. Table 4 summarises the requirements expressed by persons in PRASA's executive and maintenance management positions.

Table 4: Expressed stakeholder requirements

ID	Need
SR1	Insight into which maintenance tasks are most critical for the restoration and maintenance of passenger-ready railway infrastructure
SR2	A means for maintenance planners to view the available materials and tools at the time of maintenance task scheduling
SR3	A central database that stores (to prevent the loss of data), and provides remote access to, railway infrastructure data
SR4	More value to be extracted from the data available on the condition of the railway infrastructure through data analytics and/or other software services
SR5	Digitised reporting systems (less manual/paper work) in the areas of, for example, job chart management and data capturing

The issue of numerous data sources having to be considered when maintenance planning is done, can be reformulated as a need for <u>a single source of information that integrates all the data sources on the condition of the different infrastructural domains</u>, for the engineering team and maintenance planners. This addresses SR1, SR2, and SR3.

The issue of asset-specific information being difficult to obtain, and historic information on assets' condition even more so, can be reformulated as a need for <u>asset-specific information (both current and past) that can easily be obtained to be used in analysis</u>. This addresses with SR1, SR3, as well as SR4.

The ineffective maintenance output as a result of the continual, immediate responses in the form of breakdown maintenance, without prioritisation of when and where these tasks are performed, highlight the need for a perspective of the condition of *all* the network infrastructure domains' assets. This broadens the scope of integration mentioned in the first derived need (addressing SR1, SR2, and SR3).

The difficulty in effectively managing breakdown maintenance, because of a lack of perspective on the other infrastructural domains' condition in the area of the reported fault(s), highlights the need for a holistic view of the condition of the infrastructure across the network, to be able to gain insight into which maintenance tasks are operationally-critical. This also addresses SR1, SR2, and SR3.

Finally, the digitisation of the current inefficient paper-based process encountered at PRASA, is addressed through an expressed stakeholder requirement in SR5.

3.2.2 Implicit stakeholder requirements for a software system

Some stakeholder needs were not explicitly mentioned but given the context, these requirements are implied to address the inherent risks of creating a software system that will integrate data for PRASA. Table 5 lists these implicit stakeholder requirements.

Table 5: Implicit stakeholder requirements

ID	Need
SR6	System availability should be ensured
SR7	Data should be secure
SR8	Integrating data from different data silos
SR9	Compatible with computational resources available at PRASA
SR10	Adhere to PRASA organisational data-handling policies
SR11	Compatible with PRASA's network constraints

3.3 Digital twin system design requirements

In this section, design requirements for a DT system are provided. For future development and extension of the DT technology demonstrator, the requirements are stated broadly and are not limited to the given context. The requirements are divided into functional requirements (FRs), non-functional requirements (NFRs), and design constraints (DCs). These are listed in Table 6, , and Table 8, respectively. For traceability, Table 9 shows a mapping of the stakeholder requirements to the design requirements and constraints.

Some requirements are considered to involve more than the functional scope specified for this study (as described in Section 1.2) and are therefore excluded for the rest of the design process. These (marked with "**" in the tables below) would, therefore, require specific further research. Due to time constraints, some other requirements (marked with "*") were also not implemented. However, these requirements should be straight-forward to adopt, given the experience with the case study implementation (discussed in Chapter 5).

However, the requirements that *are* included are considered sufficient to address the stakeholder requirements, and are also those that are typically used for DTs in the contexts of railway infrastructure (Vieira *et al.*, 2022) and data silos (Malakuti, Juhlin, *et al.*, 2021). The excluded requirements would therefore not inhibit the objective of the study as outlined in Section 1.2.

Table 6: Digital twin functional requirements

ID	Description
FR1	Present user-selected information to the user
	a. …in the form of dashboards
	b. …in the form of digital documents *
FR2	Affect the physical infrastructural elements a. ...though human decisions that inform manual operations b. ...though human decisions that inform automated operations ** c. ...though automated decisions that inform automated operations **
FR3	Autonomously acquire data
	a. …from data repositories
	b. …from sensors *
	c. …from humans **
	d. …from paper-based sources *
FR4	Autonomously convert data to information
	a. …using domain-knowledge to enhance value of information provided to the user
	b. …using domain-knowledge to homogenise acquired data from disparate sources (e.g., convert to SI units, and relate location designations to latitude and longitude coordinates)
	c. …using domain-knowledge to aggregate acquired data
	d. …by hosting machine-learning (ML) **
	e. …using domain-knowledge to "clean" acquired data *
	f. …using domain-knowledge to verify and validate acquired data *
	g. …using domain-knowledge to fuse acquired data *
FR5	Store data autonomously
FR6	Adapt DT system to changing context
	a. …[to changing] physical infrastructure configuration
	b. …[to changing] user information needs
	c. …[to changing] data repositories (changing data acquisition requirements)

Table 7: Digital twin non-functional requirements

ID	Description
NFR1	Reliability**
NFR2	Maintainability**
NFR3	Secure data storage, transfer, and access**
NFR4	Acquire data from data silos without compromising the performance of existing applications that are associated with that data silo
NFR5	Create and maintain the relationship(s) between data records found in different data sources (silos)
NFR6	Integrate data from different data sources with inconsistent data types
NFR7	Integrate data from different data sources with inconsistent fieldnames for the same parameter (e.g. “Track Gauge” and “TRACK_G” – referring to the same parameter, but with different labels).
NFR8	Obtain data from different data sources through various interfaces (e.g., REST API, versus SQL)
NFR9	Adapt to unpredictable changes to the data source(s), e.g., the proprietary interface changes, or the internal structure/field names changes, etc

Table 8: Digital twin design constraints

ID	Description
DC1	Computational resources available for DT system to use*
DC2	PRASA organisational data-handling policies*
DC3	Compatibility with PRASA’s network constraints*

Table 9: Stakeholder requirement mapping

Stakeholder requirement	Relevant design requirement(s)
SR1	FR1.a, FR3, FR4, FR6
SR2	FR1.a, FR3, FR4, FR6
SR3	FR3, FR5, FR6
SR4	FR1.a, FR4
SR5	FR1.b, FR3
SR6	NFR1, NFR2
SR7	NFR3
SR8	FR3, FR4, NFR5, NFR6, NFR7, NFR8, NFR9
SR9	DC1
SR10	DC2
SR11	DC3

4 Digital twin system architecture and operation

This chapter discusses the developed of a DT system architecture that addresses the functional requirements listed in Table 6 (noting that, as indicated there, certain functions will not be addressed in this study). The architecture presented in here can be used for the development of a DT system for the integration of railway infrastructure data. The technology demonstrator developed in the case study in Chapter 5, will therefore utilise this architecture. It should be noted that, even though not all the FRs listed in Table 6 are considered in the design of the architecture, it is expected that the DT system should still be suitable to meet most of these (excluded) requirements, but that will have to be evaluated in future work.

The chapter starts with an overview of the DT system architecture in Section 4.1, followed by the mapping of the functional requirements to the areas of significance within the DT system architecture (Section 4.2). Thereafter, the main elements of the architecture are described in greater detail in Section 4.3. The chapter then concludes in Section 4.4 with a description of system operation.

This chapter elaborates on the work of Doubell *et al.* (2022).

4.1 Architecture overview

The proposed DT system architecture for railway infrastructure is illustrated in Figure 8. The architecture is an adaptation of the reference architecture by Human (2022) (discussed in Section 2.5), and the main differences are the addition of the wrapper layer, the omission of the distinction between DTIs and DTAs, and the omission of some management services. Otherwise, the roles, characteristics, and functionality of components remain the same as discussed for the reference architecture (in Section 2.5). The vendor-neutral nature of this architecture ensures that it can be used in a wide range of contexts, and prevents it from being excluded from consideration by organisations who already implement a certain software, e.g. Navisworks or Bentley BIM software.

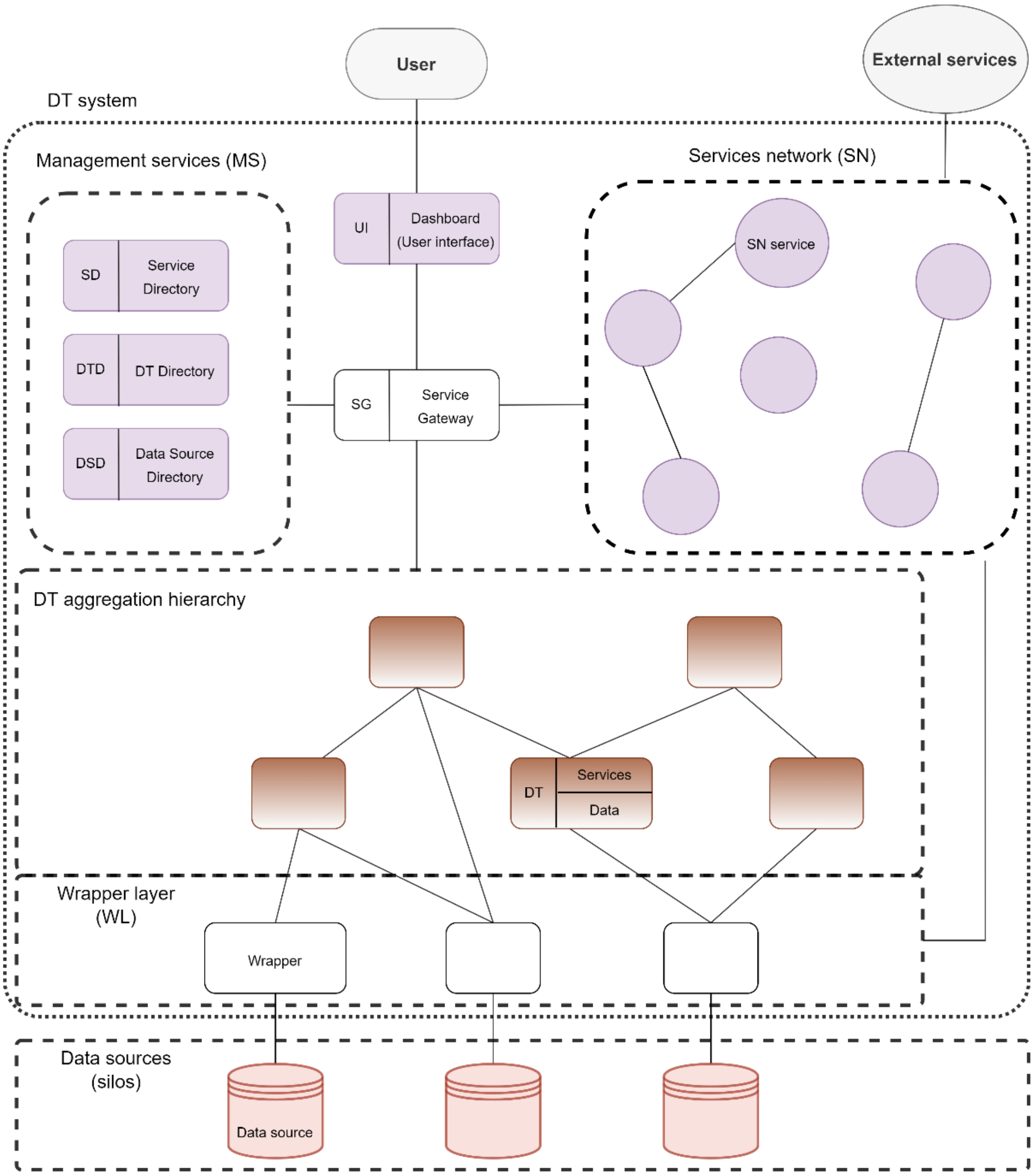


Figure 8: General digital system architecture for integrating railway infrastructure data

4.2 Functional mapping

As with the reference architecture, the architecture in Figure 8 has three major groupings – the SN, the aggregation hierarchy, and the management services. Other major supporting components are the services gateway (SG), and the UI.

The discussions below map each of the functional requirements considered in this study to a grouping or key component of the reference architecture. The layer in

SLADT is also specified for cases where the function needs to be performed by a DT.

Presenting user-selected data to the user (FR1.a)

The role of data presentation can be fulfilled by the ***user interface***.

Autonomously acquire data from data repositories (FR3.a)

Data acquisition from data repositories is allocated to the DT ***aggregation hierarchy***. However, the data sources encountered in this context typically encapsulate the functions of the first few layers of SLADT already. Therefore, a DT system must obtain data, not from the sensors in Layer 1, but rather from sources that have already stored and/or aggregated that data (representing Layer 3 or Layer 5). The resulting variations of SLADT, to be used as internal architectures for the DTs in this study, are discussed in Section 4.3.3.

The interfacing requirements for the data sources also differ greatly - many do not even have the capability to transmit their data over the internet (such as a MS Excel file). Human's reference architecture (2022) is therefore modified by adding a "***Wrapper Layer***" (WL) (elaborated upon in Section 4.3.2). The WL provides an interface to the data sources through various "wrapper" components, making the information from the data sources available to the rest of the system.

Autonomously convert data to information, using domain-knowledge to enhance the value of information provided to the user (FR4.a)

Conversion from data to information is performed through software services that are performed either by DTs (in the DT aggregation hierarchy) or by SN services. FR4.a would therefore be performed by either or both the DT ***aggregation hierarchy*** and the ***SN***. DTs provide services through DT services, located in Layer 6 of SLADT.

Autonomously convert data to information, using domain-knowledge to homogenise acquired data from disparate sources (FR4.b)

The relevant sources can either relate to the infrastructural data repositories described in Chapter 3, or it can refer to the DTs, providing information to other DTs or SN services. The data to information conversion can therefore be done by the DT, in the ***aggregation hierarchy***, through the IoT gateway (SLADTA Layer 4) with the possible assistance of the WL, or by a SN service in the ***SN***.

Autonomously convert data to information, using domain-knowledge to aggregate acquired data (FR4.c)

This functional requirement also relates to the handling of data upon acquisition from an unspecified source (which can either be a data repository, or a DT). It can therefore be addressed either through the DT ***aggregation hierarchy***, or the ***Service Network***. DTs will perform the function as part of the IoT gateway functionality (SLADT Layer 4) with the possible assistance of the WL.

Store data autonomously (FR5)

SLADT prescribes two layers for data storage (Layer 3 and 5). However, the storage prescribed here is long-term and will therefore be stored in Layer 5 of the DTs in the DT ***aggregation hierarchy***.

Adapt DT system to changing physical infrastructure configuration (FR6.a)

The physical infrastructure configuration is represented by a hierarchy of, and interconnection between, DTs in the ***aggregation hierarchy***. Therefore, changing infrastructure (such as new assets, stolen or vandalised assets, etc.) would need to be reflected in the relation between the different DTs that comprise the aggregation hierarchy. This means that new DTs must be able to be added to the system, existing DTs must be removable, and their relation to other DTs within the aggregation hierarchy must be reconfigurable.

Since the requirement is not something that is inherent to the infrastructural elements, but rather corresponds to the relation between different infrastructural elements (or DTs), this requirement is not met through a specific part of either of the two reference architectures. Rather, it highlights the need for a person (or additional program) that can fulfil a "***system configurator***" role, which would have the responsibility of performing this function. The aggregation hierarchy, however, significantly simplifies the work of such a system configurator.

Adapt DT system to changing user information needs (FR6.b)

User information needs are met through the software services, that (as has already been discussed) are performed through the DTs in the ***aggregation hierarchy*** and the SN services in the ***SN***.

However, this requirement also highlights the need for the DT system operation to be independent of the software services provided or, in other words, it should be possible to add new services or adapt old ones without affecting the rest of the system.

Adapt DT system to changing data repositories (changing data acquisition requirements) (FR6.c)

Data acquisition from data repositories is facilitated through the IoT gateway (Layer 4) of the DTs in the aggregation hierarchy, in combination with the WL. However, adaptation to changing data repositories will be facilitated by the wrappers in the ***WL***, described in Section 4.3.2.

4.3 Architecture components

This section discusses the significant components of the DT system architecture presented in Section 4.1. Since many of the components fulfil the same function as the original reference architecture, reference should be made to Section 2.5, for the components that are not discussed here.

4.3.1 Data sources

The *data sources* exist outside the scope of the system to be implemented. This is because these data sources already exist in the form of either proprietary systems or established organisational systems, which are managed and maintained for the purpose they were created for. It is often impossible or undesirable to alter these data sources. The architecture does not require any modifications to these data sources, but only a known interface to them. If, for example, a data source provides data through a web API, the IP address, authentication information and the message ontology for that web API would be required.

Examples of data sources can be seen in Section 3.1.3 and include the MS Excel files generated by the TGIV measurements, contact wire measurement records, the substation energy meters, and the signalling system’s proprietary software system.

4.3.2 Wrappers

The WL is the main enabler for the interfacing with data repositories. The WL contains wrapper components that provide a consistent interface between the DT system and the data repositories. Each wrapper has a designated data source, for which it translates the unique API (or interface) of that data source, to a common, stable API for the DT system (similar to the idea of “harmonized APIs” as implemented by Malakuti, Juhlin, *et al.* (2021)). The wrapper is custom developed for its designated data source. This enables SN services and DTs to make standardised data requests to any data repository (regardless of interface specifications), as long as there is a wrapper that provides that interface. The wrappers' internal architecture is illustrated in Figure 9.

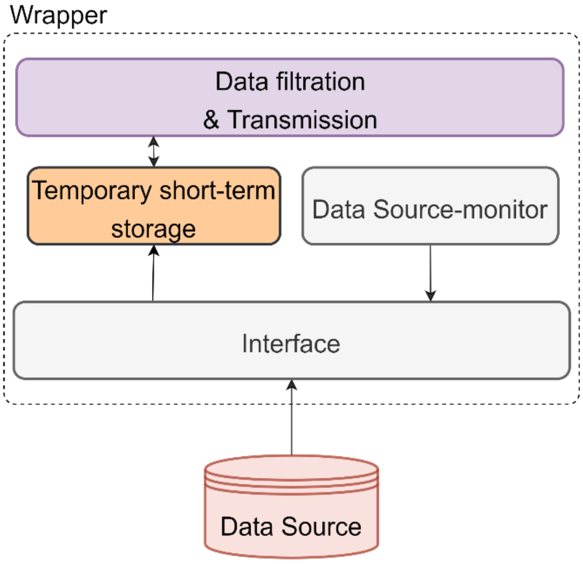


Figure 9: Internal architecture of the wrapper component

The wrappers play a key role in addressing the integration challenges (NFR4, NFR7, NFR8, and NFR9) because they are the only parts of the architecture that interface directly with the data sources. The sources can differ substantially and, therefore, each data source's wrapper is likely to require custom software development. To limit the custom development, the wrappers are kept simple: with functionality limited to monitoring its data source for new data, broadcasting a notification of new data to all listening components when the data source, and responding to data requests from other components. Changes in a specific data source, would therefore only require changes to that data source's wrapper component, instead of having to change the entire DT system, or all the DTs that require data from that data source.

Data integration is performed through the co-operation between the wrapper and the DT components (addressing the integration challenges, NFR4 – NFR9, together). The way in which this integration is performed, can be seen in Section 4.4.2.

Queries to the data source often involve large data transfers, incurring significant latencies. The wrapper therefore contains a temporary, local storage component to store a mirror of the data source every time it is checked for new data. This reduces the number of repeated queries that need to be made to the data source, thereby reducing the overall latency of data acquisition.

Wrappers also present a means through which to interface with external sources or services that do not necessarily provide data purely for the purpose of the DT aggregation hierarchy. Examples are weather services or financial data (typically connected with directly from the SN). In a case where multiple services in the SN would therefore require data or services from external providers, they can all just query the wrapper, instead of each making the connection to the external service or data provider.

4.3.3 Digital twin internal architecture

The DT components in this architecture are responsible for providing the digital representation of the physical entities. A DT should maintain knowledge of its own scope of reality and the relevant domain-knowledge associated with its physical counterpart. The DT representation in Figure 8 illustrate two high-level components - data and services. The DT contains the data of its physical counterpart in Layers 3 and/or 5 of SLADT (depending on the internal DT configuration used, discussed hereafter), while the services are provided by Layer 6.

SLADT (Redelinghuys, Basson and Kruger, 2018), described in Section 2.4, is used as internal architecture for the DTs. However, the data sources to the DT system in this context already perform the functionality of Layers 1 and 2 of SLADT. Therefore, these two layers are not required for the DTs in this architecture. For the remaining layers of SLADT, the three configurations shown in Figure 10 are used.

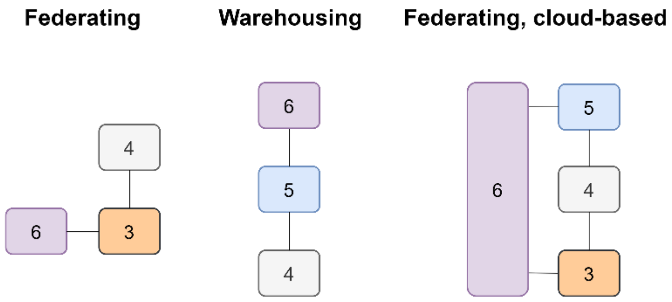


Figure 10: Possible SLADT configurations for digital twins

The configurations are adapted to the purpose and context within which they are used. Two general approaches to gathering data are data warehousing and data federation (Pathak *et al.*, 2006). The former refers to when the relevant data is collected, transformed, and stored in a common format that can subsequently be queried. The latter refers to where the data is not stored, but rather collected and transformed upon request. These two methods of data acquisition are the distinguishing factors between the three variations.

A DT without Layer 5 is best suited for data federation – obtaining data through Layer 4, where it is transformed and then temporarily stored using Layer 3, from where it is provided to the requesting entity through Layer 6. This configuration of the different layers of SLADT, is therefore termed the "*Federating DT"* (F-DT).

Sometimes the long-term availability of the external data is not certain. An example of this is where data is provided as a service and access to data is terminated once the payment for such access ceases. In such a scenario, the organisation would want to duplicate the data on its own hardware to ensure the long-term availability thereof. SLADT's Layer 5 would then have to be implemented. In this scenario, low-latency (near real-time) data access might not be a priority (as is the case with PRASA's railway infrastructure – highlighted by the need for only Layers 4, 5 and 6). In this case it would be sensible to opt for a data warehousing approach. The "*Warehousing DT*" (W-DT) is a suitable configuration here, completely omitting Layer 3.

If instead, low-latency (near real-time) data access is indeed a priority, another variation termed the "*Federating, cloud-based DT*" (FCB-DT) would be appropriate. This DT has all the components of a typical SLADT implementation, except for Layers 1 and 2. The FCB-DT enables both data federation and data warehousing, but at greater computational and storage resource expenses.

Any combination of these three configurations can be used in the implementation of DTs for the DT system architecture. The selection is subject to stakeholder data requirements, computational and storage resource capacity, and the purpose (use cases) of the DTs for the organisation. Selection can be guided by design patterns, such as that presented by Human (2022), that translate these priorities into architectural design choices.

4.3.4 Digital twin aggregation hierarchy

SLADTA makes a distinction between DTIs and DTAs - considering DTIs as DTs that obtain data from Layers 1 and 2, and are therefore found at the lowest level in the aggregation hierarchy. DTAs are considered "higher" in the aggregation hierarchy and draw data from DTIs or other DTAs. This distinction assumes that DTAs would not also obtain data from data sources or repositories. This constrains the design of the system by forcing all DTs to obtain data from either a data source (as a DTI), or from other DTs (as a DTA), but not both. In the case where a specific data source is of interest in the representation of a DTA, a DTI would have to be developed that is not of direct interest to the stakeholders.

An example is the railway track – a combination of two parallel rails, kept in place by (typically) concrete or wooden sleepers, upon which the train travels. The individual rails are highly likely to possess different properties at a given point in time. The stakeholder might require a digital representation (DT) for the individual rails, as well as a distinct DT for the track. The track possesses additional characteristics such as slope, track gauge, etc. Now these parameters mentioned for the track can be argued to belong to the sleepers, but since DTs for the sleepers are not required, the data repositories that contain this information, can provide it directly (through a wrapper) to the DT of the track.

Strictly adhering to SLADTA would require an additional DTI to be developed for the sleepers, which can then provide this data to the DTA of the track. However, if this restriction is removed, the data can be provided to the track DT directly, thereby reducing the need for (time-consuming) DT development for the sole purpose of providing data to another DT. Therefore, this restriction is removed in the proposed system architecture – DTs can obtain data from any combination of data repositories and DTs, regardless of their position in the aggregation hierarchy.

DTs in the given DT system architecture are configured to obtain data from specific wrapper(s). These wrappers also notify the DT when new data is received. Because of the diversity expected in the data sources providing data to the DT (possibly varying greatly in update frequencies) the role of checking for new data is not performed by the DTs.

As in the original design of SLADT, data from the data sources is gathered into the DT through the IoT Gateway (Layer 4 in SLADT). In contrast to the original SLADT, the rest of the DT system can only gain the information from a DT through its services on Layer 6 and no provision is made in this architecture for aggregation data flows through Layer 3 or 4. Both Layers 4 and 6 would most probably manipulate data – Layer 4 for calculating new parameters to store, performing data cleaning operations, etc. before storage; Layer 6 to perform the services requested.

DTs provide their information to requesting entities through services performed in Layer 6 (subsequently referred to as *DT services*). This information can also include static data, such as its attributes (e.g., the date of entity installation, the entity's OEM, etc.). The static information is allocated to the DT upon start-up from a configuration file. DT services (as opposed to SN services) transform the data stored in, or acquired by the DT, to information (e.g., by adding context) and provide a means through which this information can be retrieved.

DTs can be removed from the DT system ("deleted") by removing their details from the configuration file – thereby terminating access to the DT and its representation of the physical infrastructure element. However, this would rarely occur., since it is preferrable to maintain the DTs within from the system, so as to maintain the long-term availability of the data and configuration of data sources relevant to this DT, for future services. For example, if a physical entity is removed from the actual system (because of e.g., theft or routine replacement), that DT component should still be available to provide insight into the historic data of the physical asset, even though the latter is not there anymore.

Instead of completely removing the DT from the system, it is therefore recommended to rather just reconfigure it so that the DT does not receive new data. This way it will still be available to provide services related to the data it obtained during its lifetime. If so desired, the architecture allows for these "old" DTs to be run on a separate server and indicated to the user in a different manner (though an additional service on the UI).

4.3.5 Management services

The role of the management services in the general architecture (Figure 8) is to manage the interactions between components in the DT system. This architecture employs distinct "directory services" for the DTs, wrappers (data sources) and SN services in the DT system, through the DT Directory (DTD), Data Source Directory (DSD), and SD (Services Directory) services respectively. These directory services keep track of the components that are present in the system.

4.3.6 Configuration file and entry-point program

The components seen in Figure 8, are launched with an entry-point program in accordance with specifications given in the system configuration file. These two form a similar idea to that of the configuration server component, presented by Human (2022) and briefly described in Section 2.5.

The entry-point program is used to run the entire system, in other words the .exe that will start all the components (as described in Section 4.4.1) and execute ("run") the DT system on the operating hardware. The entry-point program "builds" the DT system based on the configuration file. The main purpose of the use of a configuration file is to maximise the reconfigurability of the system, without having to change any source code. The configuration file therefore specifies the component-specific properties such as the wrappers from whom the DT needs to obtain its data, the socket addresses where the different components will be hosted (in the case where TCP/IP communication is used), the filtration parameters that the DTs need to communicate to the wrappers, etc. The configuration file also specifies the SN and DT service scripts to be used in the system.

4.4 System operation

This section describes the system operation through which the system architecture satisfies the functional requirements stipulated in Table 6. Three typical scenarios encountered in the use of such a system are discussed: system start-up, new data being logged in the data repositories, and data requests by the user.

Before discussing the different scenarios, a general note on component interactions. User information requests, together with data transmissions, are the most prevalent communications/interactions that need to be facilitated by the system. The different components in the system should have the ability to interact with other components, both as senders/initiators/requestors, and as receivers/responders/requestees.

All interactions are based on the initiator’s knowledge of where and how to contact the responder. Registration of each of the SN, WL, and DT hierarchy components at their respective directory services is, therefore, the backbone of any interaction and is described as part of the “system start-up” scenario in Section 4.4.1.

4.4.1 System start-up

System start-up is directed by a configuration file that is set up by the system configurator. The configuration file ensures, firstly, that the correct data repositories (sources) are utilised and mapped to the relevant assets (or asset groups, also subsequently referred to as infrastructural elements) and, secondly, that the physical infrastructure configuration is represented accurately by the DT hierarchy. Finally, the SN and DT services to be activated within the system must also specified by the configurator.

This approach provides the system configurator with control over the configuration of the DT aggregation hierarchy (FR6.a), the use cases (services offered) (FR6.b), and the data sources available to the system (FR6.c).

When the entry-point program is run, all the components start up individually and independently of one another. The first step of any non-management service component (after it has been initialised to a satisfactory degree) is to register itself with the relevant directory service.

Component registration ensures that the relevant directory services are aware of the *Interaction Information Set* (IIS) of the components in the system. The IIS contains the information required for interaction with the component in question (such as socket address and authentication information). The information in the IIS would thus be highly correlated to the choice of messaging mechanism implemented for the system (e.g., with TCP/IP, not considering security and authentication, the IIS would contain the IP address and port number).

Registration involves a fairly simple interaction, illustrated in the sequence diagram illustrated in Figure 11 (next page). During registration, the registering component sends a message to the SG, with the instruction to register itself. This message is accompanied by its own IIS. Based on the type of component (DT, SN service, or wrapper) the SG will route this registration request to the appropriate directory service, where the IIS, with the component name, is stored. Other components can now request the relevant directory service (through the SG) for the IIS of a component (explained in Section 4.4.3).

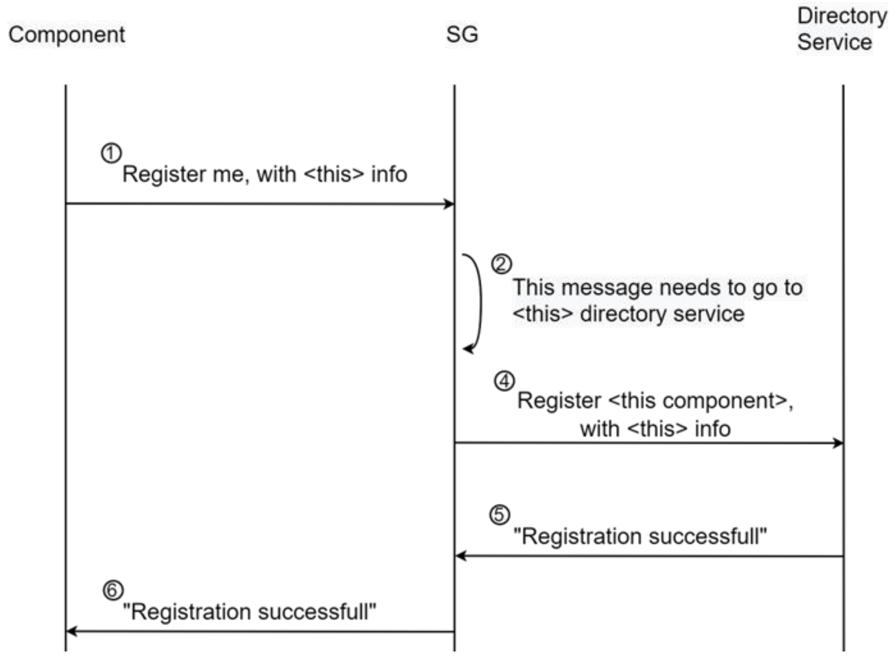


Figure 11: Sequence diagram for component registration

4.4.2 Update data repositories

This section considers the operational sequence when new data has been logged in the data repositories. This sequence provides insight into how the presented DT system achieves NFR4 to NFR9.

When new data is logged to a data repository, that data repository’s wrapper will broadcast a message to all listening DTs (that rely on this wrapper for their data), that it has new data to share. The DTs will subsequently request the wrapper for a specific parameter for a specific time range. The wrapper will then withdraw or filter the requested data from its source, homogenize the data to the format expected by the DT, and pass the data to the DT.

The DT receives the data and integrates it (provides semantic links between data records) using the domain knowledge of how the parameters that is encapsulated within itself, relate to one another. The information is then subsequently stored in DTs’ long-term repository. DTs do not have to handle inconsistent features and data types, since the wrappers homogenised the data.

In the sequence described above, the wrappers are responsible for filtering the requested data from the source, but not for integrating different parameters. The wrappers, therefore, address NFR4, NFR7, NFR8, and NFR9, while DTs address NFR5 and NFR6.

This process serves the purpose of autonomously acquiring data from data repositories (FR3.a) and converting data to information, to homogenise and aggregate the data using domain-knowledge (FR4.b and FR4.c respectively). This data is then stored in the DTs' long-term storage (FR5).

4.4.3 Information requests

The information exchange that does not interact with wrapper components, will here be referred to as the "*information exchange mechanism*". This mechanism enables the functionality of presenting information to the user on the UI (FR1.a).

The service requested from the UI will determine the information presented. Therefore, development of the DT and SN services, and the UI displays required by them, should occur concurrently. This enables the integration of domain-knowledge with the presentation of information to the user (FR4.a), thereby ensuring sensible display of the information (also meeting FR1.a).

The information exchange mechanism utilises a basic interaction sequence as a building block for interactions between a component requesting information (the requestor) and the component responding to that request (the respondent). The interactions that form the basic interaction sequence are illustrated in Figure 12.

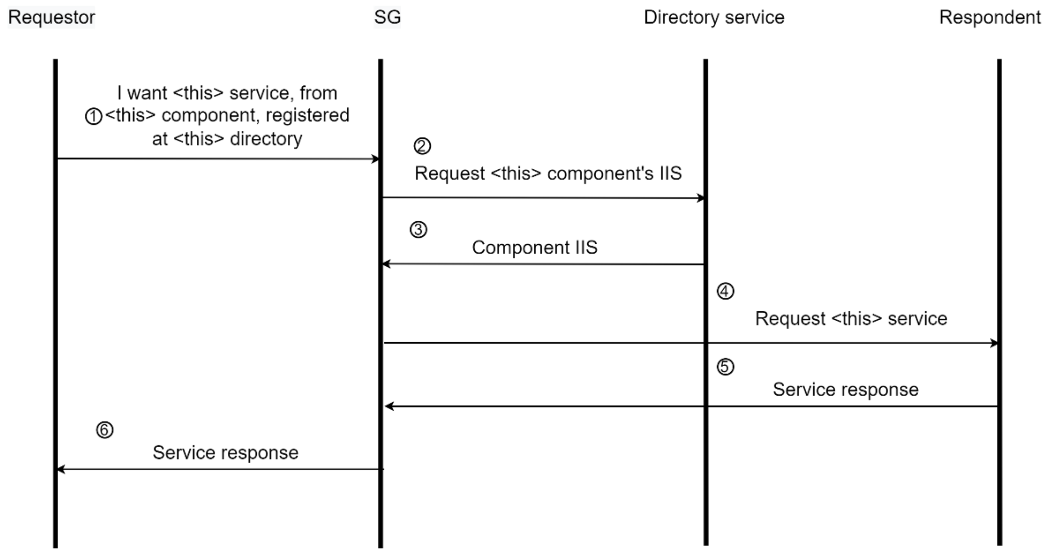


Figure 12: The basic interaction sequence for information exchange

Every component (other than the SG itself) will, upon start-up, be provided with both its own IIS and the IIS of the SG by the configuration file. The SG is initiated with the IIS of the management services.

For a requestor to obtain information from a respondent, a message must be sent to the SG that specifies the *identity of the requestee* (i.e. the name/id of the SN service or DT), the *management service* where the IIS for this component can be found, and the *service* requested from it.

The SG then obtains the IIS for this component from the management service specified, at which point it (the SG) requests the respondent to provide information through the specified service. Respondents cannot always provide all the information to the requestor directly upon request – it may be necessary to first obtain data from other components in the DT system before it can perform its own service. In that case there are two types of interactions between non-wrapper components: inter- and intragroup interactions.

With *intergroup interactions*, the interaction(s) occur between components in different areas of the system, such as between the DT hierarchy and management services. *Intragroup interactions* occur when the interaction is between components within the same group, such as between components in the SN, or between components in the DT hierarchy.

Both types of interactions are initiated by the requestor, with a request for the IIS of the responding component. Intragroup communications would only need to do this *once*, after which they can store the IIS of that component, whereas intergroup interactions will have to perform this step *repeatedly* for each interaction.

The two types of interactions are differentiated based on the intensity of interactions. Intragroup interactions generally occur much more frequently and are more predictable (the same components interacting with one another repeatedly), than intergroup interactions. This distinction reduces the total latency of the system, as storing the corresponding component's IIS, removes the need to request it from the directory service again. This reduces the latency for a single instance of such an interaction. If frequently used, this accumulates to a significant decrease in latency for the entire system.

An example of intragroup interactions is where a given SN service requests information from the same SN services every time. This is opposed to the intergroup interactions required when "SN service A" requests data from "DT A" for one query, but for another query requires data from "DT B" instead.

5 Case study – a digital twin technology demonstrator

This chapter discusses the case study – applying the DT system architecture described in Chapter 4, to a section of PRASA railway infrastructure as a DT system technology demonstrator, i.e. demonstrating the proposed DT system's ability to meet the FRs discussed in Section 3.3. The case study first discusses the scope delineation in Section 5.1, after which the implementation details of the reference architecture is discussed in Section 5.2. This case study will serve as basis for the evaluation presented in Chapter 6.

5.1 Scope delineation

This section discusses the selected scope for the case study in terms of the physical scope (referring to the infrastructural elements and their data sources), as well as the services required to meet the stakeholder requirements discussed in Section 3.2.

5.1.1 Physical scope

The physical scope of the case study is considered to comprise of two aspects: the infrastructural elements that should be represented by the DT system, and the data sources from where these DTs obtain their data.

The DT system technology demonstrator is developed for a subgroup of railway infrastructure between the Cape Town and Claremont train stations (a 10.16 km stretch of infrastructure). This subgroup will comprise of the *perway (track and ballast)*, as well as the *OHTE (electrical)* infrastructure for this section.

Unfortunately, most of the data sources described in Section 3.1.3, could not be used in the case study due to issues regarding accessibility, authorisation, and a lack of domain-knowledge to interpret the data at the time of the contextual analysis done in Chapter 3. This rendered only the TGIV data, the MTS monthly reports, the daily reports from Electrical Control and the Signalling fault logs, available for use.

The TGIV datasets (MS Excel files) that describe parameters related to the *geometry of the track*, *the layout (coordinates) of the track*, the *wear on the rails*, the *condition of the ballast*, and the *threshold exceedances measured on the catenary wires*, were used for the case study.

Though it is generated from a single origin, the data found in the different files of the TGIV, differ in terms of data structure, and data types encapsulated. Some store strings, others store integers, and still others store floats. The unavailable data sources' interfaces will be simulated, since the TGIV presents only one interface (accessing data from a local data repository, such as a MS Excel file). The simulated interfaces are between the DT system and a web server (through a web-API, such as the proprietary systems' interfaces), and between the DT system and an SQL database (such as Empac). Future work could explore the use of more and other data sources for the DT system.

The dataset on the layout of the track is simulated as data that is hosted on a web server, obtained using a web-API. This is done by creating a python Flask web app that, upon a REST-API, GET request, responds by loading the file from local storage and sending it as a JSON string.

The rest of the datasets are all hosted as separate data tables within a PostgreSQL database. This simulates the data acquisition through SQL queries with authentication. The ability of the system to use data from local data repositories, is illustrated through the third DT system test, described in Section 6.1.3.

Data collected during the measurement run of 6 March 2021 was used across all the data sources. However, for the Electrical infrastructure, this was elaborated upon with data from 3 July 2019, and 30 January 2020. Test 1 (described in Section 6.1.1) uses the data from 20 August 2021 to illustrate the automatic update functionality enabled by the DT system (which is explained in Section 4.4.2).

5.1.2 Service scope

As described in Chapter 4, information is presented to the user through the combination of the DT and SN services of the DT system. These services are dictated by the user needs. The stakeholder requirements, listed in Section 3.2, provide a good indication as to what these services should be, in order to provide value to the stakeholders.

The services that are implemented for the technology demonstrator developed in this study, as derived from the expressed stakeholder requirements presented in Table 4, are therefore the following:

S1. Viewing the operational state of an asset

S2. Viewing the usability of an asset(-group) for train operations, subsequently only referred to as *usability* (so that assets that impede the usability of otherwise operational assets, can be prioritised in maintenance)

S3. A map of the locations where measured (condition) parameters exceed their thresholds by a user-defined margin (so that, for the user-selected asset(-group), the largest exceedances, or localised exceedances that correspond to other tasks, can be prioritised in maintenance operations)

S4. A chart of the number of measured (condition) parameters that exceed their thresholds by a user-defined margin, over a user-defined span of time (for analysis of asset degradation trends)

S5. A comparison between the number of threshold exceedances of different assets (so that maintenance managers can have an accurate comparison of the work required on different asset(-groups))

5.2 Implementation

The case study implemented the DT system architecture, presented in Chapter 4, to the scope outlined in Section 5.1. This yielded a technology demonstrator that could then subsequently be evaluated in Chapter 6.

5.2.1 Case study software architecture

Figure 13, illustrates the result of applying the general DT system architecture of the technology demonstrator (described in Chapter 4) to the case study. The DT system components are implemented on two computers for the case study. This illustrates the ability of the architecture to run as a distributed system. The distribution of components between the two computers is shown in Figure 14.

The four rail DTs, the track and ballast DTs, and the two catenary DTs, are hosted on Computer A, while the DTs describing the perway and electrical infrastructure domains as a whole, are hosted on Computer B.

The *Rail*, *Ballast*, and *Catenary Data* as well as the *Track Geometry* wrappers, together with their respective data sources, are hosted on Computer A, while the *Coordinates* wrapper is hosted on Computer B and the server that hosts the data is hosted on Computer A.

All the management services, together with the UI server, are hosted on Computer A, while the SN service, named *ExceedCountComparator*, and the SG are hosted on Computer B. The case study is implemented in such a way that the UI hosted by the server, can be interacted with by client devices anywhere on the same local area network. The client devices should be able to display a web-browser such as Google Chrome.

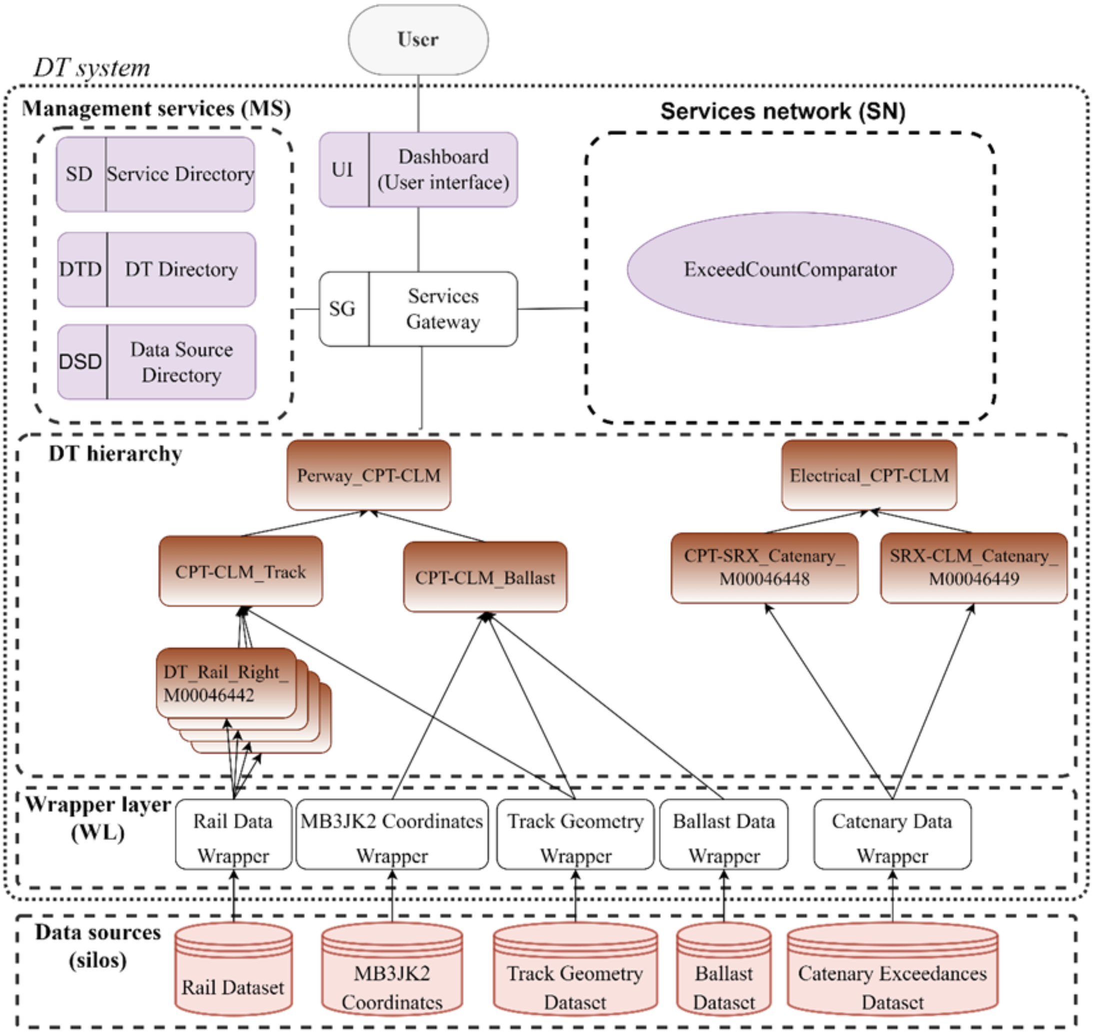


Figure 13: Case study digital twin system architecture

5.2.2 General implementation notes

Before an in-depth discussion of the case study implementation, some general remarks can be made regarding the implementation of the technology demonstrator. All the components of the technology demonstrator, except for the UI component, are implemented as a Python program, packaged with all its dependencies into a Windows executable file.

The entry point program reads the configuration file (implemented as a JSON file) and runs each of the DT system components as a separate thread on the computer used for execution. All components are implemented as instances of their respective python classes. The parameters that specify their attributes are provided in the configuration file, as discussed in Section 4.3.6.

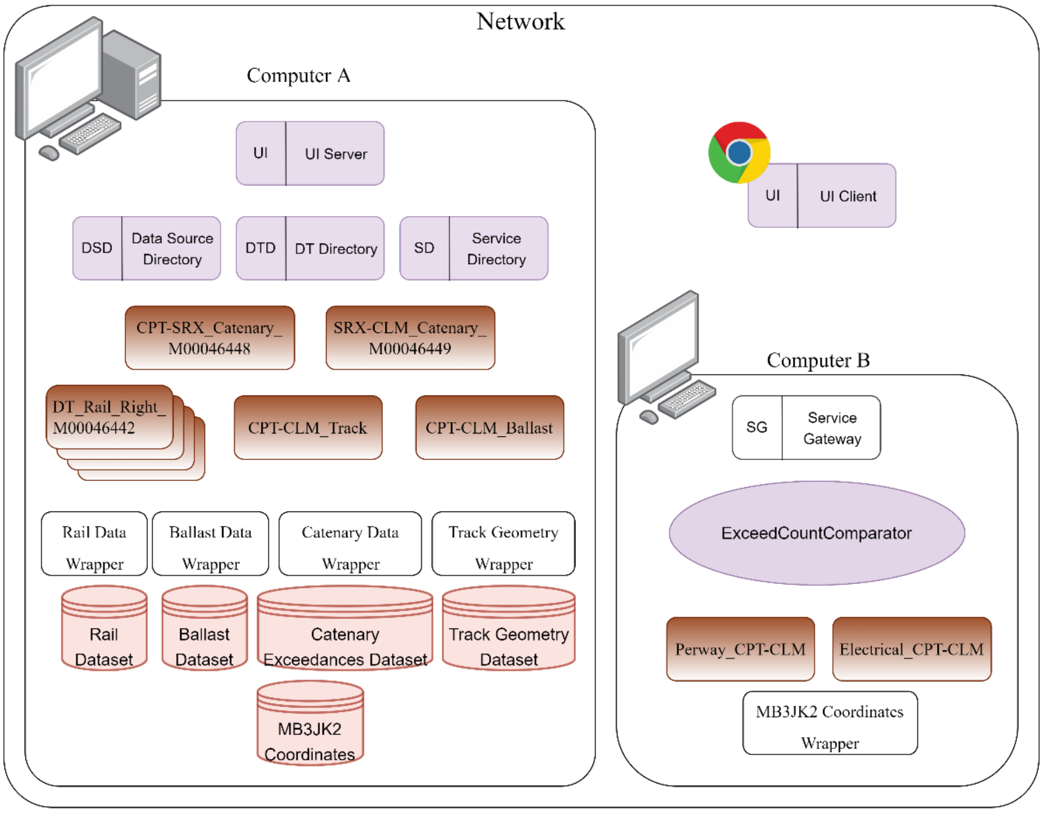


Figure 14: Distribution of DT system components across hardware

The implemented system does not accommodate "hot-updating" of the DT system configuration, i.e. updating the DT system configuration during runtime. Therefore, in the case of required reconfiguration, the DT system should be restarted with the updated configuration file. Data will not be lost, since the data storage locations would stay the same. Future work should consider the implementation of "hot-updating" functionality.

The functionality of the different SLADT layers are encapsulated in functions within the DT python class. Layer 5, providing functionality to interface with the long-term storage component of the architecture, is selected to be implemented as a PostgreSQL database. Interactions with the database is done through SQL queries using the psycopg2 python module. The directory services also use this python module since each of them also utilise a PostgreSQL database to keep their respective directories of active components.

Interactions between the different architectural components are all implemented using the event-based TCP/IP-based Socket.io framework. Messages are communicated as JSON strings.

5.2.3 Digital twin specifications

Given the physical scope with its specific infrastructural elements identified in Section 5.1.1, certain choices need to be made with regards to which infrastructural elements will be represented by which DTs. This is done in Section 5.2.3.1, after which the implementation of these DTs are discussed (Section 5.2.3.2).

5.2.3.1 Digital twin identification

DTs in continuous systems such as railway networks, water distribution networks or oil and gas pipelines, should either be demarcated according to the respective concerns of the end-users, or according to their manageable parts (Human, 2022). Since maintenance management is done according to infrastructural domains, the perway and electrical infrastructure domains would therefore work well as DTs – providing stakeholders with the required holistic view of the infrastructure as expressed in Section 3.2.1.

Rails, track, and ballast are elements that are given distinct maintenance tasks. Rail maintenance activities include grinding, lubricating, and bending the rails. Further, track geometry is maintained through lifting, levelling, aligning, or replacing the critical components, and ballast is maintained through processes of mechanised ballast cleaning and management. Because these infrastructural elements also have data sources that measure parameters specific to their condition, it can also be expected that the maintenance planning meetings will include specific discussions regarding their maintenance. It should be noted here that, although rails form part of the track infrastructural element, they will be considered as separate (constituent) DTs, to illustrate how a DT (the track DT in this case) can obtain data from both wrappers as well as DTs. Rail infrastructure elements are also maintained distinct from one another, so by providing a distinct DT for each rail components makes sense from the perspective of the separation of concerns, as well.

The catenary wires are set up between substations. A key concern for stakeholders to the maintenance process, is the operational condition and usability for railway operations of these assets. If one part of the catenary wires between substations experiences a critical fault (is broken, stolen or vandalised), the entire catenary wire up to the next substation is rendered effectively useless. This leads to the selection of the allocation of DTs to two catenary wires between "*Traction Main*" (or TM, which is the substation that obtains electricity from the municipal grid) and the *Salt River* substation (SRX), and between the *SRX* and *Claremont* (CLM) substations respectively.

5.2.3.2 Digital twin implementation

The DTs that interact directly with the wrapper layer are implemented as W-DTs (described with the other DT configurations in Section 4.3.3). These DTs are for Rail, Ballast, and Catenary from TM-SRX and SRX-CLM respectively.

This decision is made based on the requirement for long-term storage and the speed of data acquisition not being a critical factor. Long-term storage (Layer 5) is required to ensure the longevity of the data that is obtained from third party contractors but not yet backed up to cloud storage for PRASA (as discussed in Section 3.1). The current lack of cloud storage, presents a danger of data loss in the case of theft, data corruption, damage, etc. The design decision was therefore made to duplicate the data from the respective data sources within the DT system.

Furthermore, the speed of data acquisition and service delivery is not a critical concern for the stakeholders. This is exemplified by the data acquisition processes of the contractors who are allowed to take several days after their measurement, before delivering data/information to PRASA. There is therefore no need for the local storage (Layer 3) of data to enhance data acquisition and service delivery speeds of the DTs.

Since the electrical domain DT does not obtain data from a wrapper, the design decision is made not to commit its data to long-term storage, as was done for the catenary DTs. The electrical domain DT is simply used to provide services related to the entities that it represents. This DT is therefore implemented with an F-DT configuration.

The perway DT is implemented as an FCB-DT, since it needs to be able to both federate data from constituent DTs, and store data from the database that provides data on the geometry of the track.

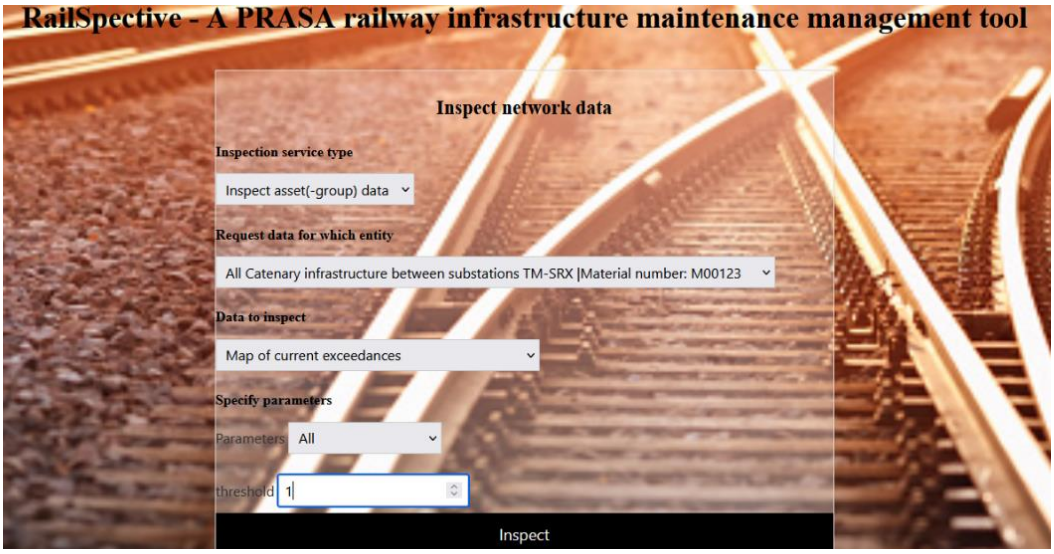
Significant information provided to DTs from the configuration file include the labels (if they have any) of their constituent DTs (so that they may know where to federate data from), how their constituent DTs relate to one another in terms of usability (i.e. which DTs' operational condition, directly affect the usability of which other DTs), as well as the operational condition of the infrastructural element (given that the infrastructural element is an asset and not an artificial grouping such as perway - which cannot be considered to have an operational condition apart from the operational condition of its constituent elements).

Upon start-up, the DTs are to register themselves at the DTD. However, this is only done after interaction with their constituent DTs (that provide them with data) have been established – ensuring its ability to perform the required services at the time of registration (visibility to the user and other components).

The JSON string that must be communicated to the wrappers to obtain the appropriate data from their sources, is also included in the configuration file of the DTs. This can be seen in an example of the JSON configuration file, given in Appendix C.

5.2.4 User interface

The UI is implemented as a reactive web application, created with the Svelte web framework. The UI server is hosted and run independently to the rest of the DT system. This enables developers to adapt the UI dynamically - changing the way that data is displayed, or inputs received, without any changes to rest of the DT system. The case study implemented the UI to be visible through a regular web browser, such as Google Chrome. Figure 15 presents an example of a fully populated UI, just before the user queries the information.



RailSpective - A PRASA railway infrastructure maintenance management tool

Inspect network data

Inspection service type: Inspect asset(-group) data

Request data for which entity: All Catenary infrastructure between substations TM-SRX |Material number: M00123

Data to inspect: Map of current exceedances

Specify parameters

Parameters: All

threshold: 1

Figure 15: A fully populated UI ready to query data from the DT system

The UI is dynamically updated as the user enters/selects the desired information. The JSON string that the UI sends to the SG, is populated through these entries and selections. The process that is followed by the user in specifying a specific service, is illustrated in the flow diagram in Figure 16. The template for the messages that is populated as the user selections occur, is illustrated in Figure 17. The multicoloured blocks correspond in Figure 16 and Figure 17 - indicating places where the user's selection (in Figure 16) determines an outcome or value in the JSON string (in Figure 17). The grey blocks indicate the actions and fields that the UI will perform and populate automatically. The keys of the JSON message template are set (constant) and are indicated in purple. With the exception of requests to the wrappers, this template is used when any two components request services from one another (i.e. for all messages in the basic interaction sequence described in Section 4.4.3).

The first selection asks the user whether the desired service relates to a specific asset(-group) or not. If the user wants to know, e.g. the operational state of a specific asset, then they would select the asset(-group) option. Otherwise, they would select the other option. The back-end of the UI would configure the JSON message accordingly, as indicated by the first conditional block in the flow diagram presented in Figure 16.

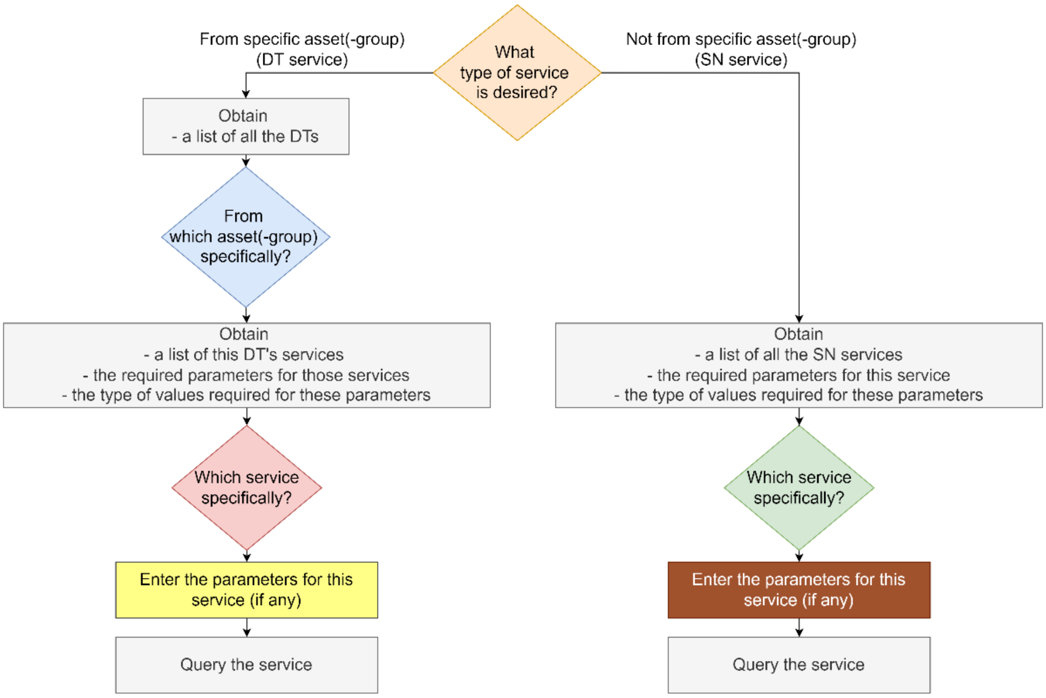


Figure 16: Flow diagram illustrating the process of specifying a service to be performed

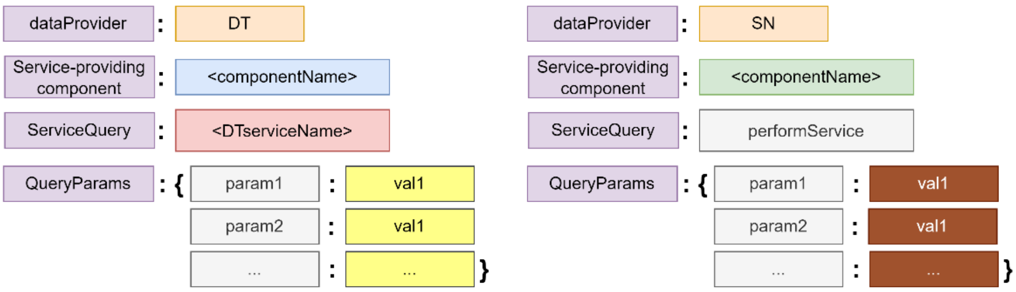


Figure 17: Service request message template for a DT service (left) and a SN service (right)

This initial differentiation between the user's request for a DT or SN service are as a result of the fact that the former can provide numerous services, while the latter only provides a single service. This means that in the SN service case, the UI can immediately be populated with the parameters required to perform that service, whereas in the DT case, these parameters are determined by the DT service selected, therefore, the user should first select which DT service is desired.

The information that the UI obtains automatically (indicated in the grey blocks in Figure 16) are attributes that are specified in the DT system configuration file. These are requested from the components themselves and offered as services in the same way as the basic interaction sequence A, described in Section 5.2.5.4 (without the component having to subsequently make any further queries).

The UI is developed to have three display options – text, map, and chart. Since the type and meaning of the information that the services respond with, will vary greatly, each service responds with a JSON string. The first key of such a string defines the service it provides. Based on this, the UI is developed to interpret the means to display the information. For example, the service will respond with a JSON string that has the term "Map" as its first key. Based on this, the UI will then know that the remaining key-value pairs in the JSON string refer to the coordinates, and values to be displayed at these coordinates, on a map. Development of the UI for the display of information from the DT system, therefore, requires a detailed knowledge of the format of the incoming information.

It should be noted that there are various alternatives to the implementation of the UI functionality – both in terms of how the service request message is created, as well as how information from services is received, interpreted, and displayed. Future work can evaluate alternative implementations such as altering the sequence with which the service request message is created - possibly first asking for all the available services before specifying the component(s) that should provide them.

5.2.5 Service provision

This section discusses how the services described in Section 5.1.2 is implemented for the case study.

5.2.5.1 Service demarcation

In accordance with the design principles of the architecture described in Chapter 4, the single source of information on the condition of the railway infrastructure will be the DT hierarchy. Therefore, the information of the infrastructure that can be obtained from the DT system, is subject to that which can be offered by the DT services.

The services specified in Section 5.1.2 should be addressed either through specific DT services or through SN services. The DT services that were implemented to provide the information necessary for all the other services to function are:

DT-S1. Provide the operational state of the queried DT.

DT-S2. Provide with the usability of the DTs that form a part of the queried DT.

DT-S3. Provide the locations (coordinates) where the most recent measurement of selected parameters exceed their typical thresholds.

DT-S4. Provide the number of measured instances for which the queried parameter(s) exceed their typical thresholds, for a user-defined timeframe.

Only one SN service is implemented for the technology demonstrator, and that service is:

SN-S1. Provide a comparison between asset- or asset groups, of the latest number of measured instances where the queried parameter(s) exceed their typical thresholds.

Section 4.4.3 discusses how the SN services utilise the DT-services to perform their functions. Table 10 provides a summary of how the different services are used to provide the services listed in Section 5.1.2 (that address the stakeholder needs).

Table 10: Technology demonstrator service offering

Stakeholder requirement	DT system service	DT-service	SN-service
SR1	S1	DT-S1	-
SR1	S2	DT-S2	-
SR1	S3	DT-S3	-
SR4	S4	DT-S4	-
SR1	S5	DT-S3	SN-S1

It should be noted that in addition to the services listed above, both the SN service components, as well as the DTs, include a service that provides a list of names of the available DT-services, together with their required parameters. This is for the dynamic update functionality of the UI.

5.2.5.2 Relevant domain-knowledge

The implementation of the services listed above requires knowledge of what the specific thresholds are, an understanding of how the operational state of assets affect their usability for train operations, and how condition-based maintenance tasks are delegated. The latter is required to inform sensible communication of the condition of the assets, to the maintenance managers.

The relevant domain-knowledge for these services can be explained with a single scenario.

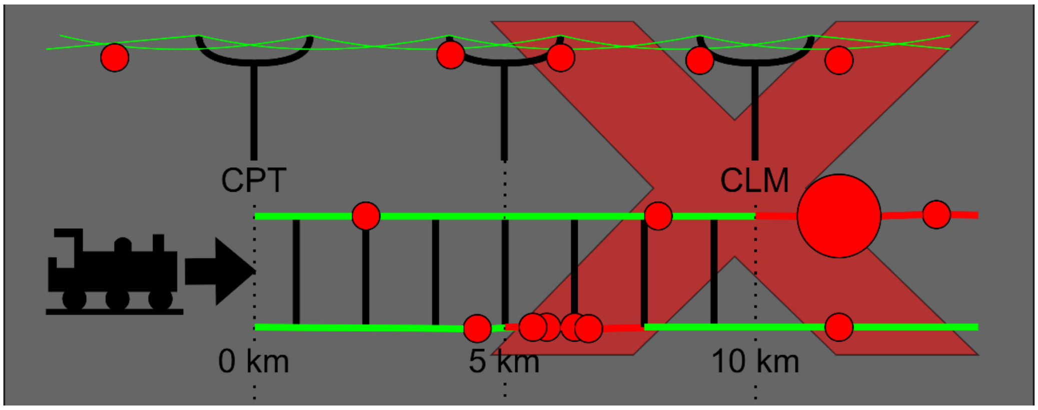


Figure 18: Scenario to illustrate infrastructure usability

Figure 18 provides a visual illustration of a hypothetical scenario where there is a 10 km stretch of infrastructure from Cape Town to Claremont station (Track and Electrical, OHTE and substations, indicated in the figure). Trains depart from Cape Town station in the mornings, from where they commute to Claremont (and beyond) and back. The red dots indicate areas on the track where some aspect of their condition requires maintenance. Larger dots indicate critical faults, where some aspect of an asset's condition exceeds its critical threshold. The colour of the assets indicates their operational state (i.e. whether that piece of infrastructure, in and of itself, can fulfil its intended function). All the OHTE in the figure is therefore still operational, but the pieces of rail on right-hand side rail just after 5 km and on left-hand side just after 10 km, respectively, are not.

For an asset to be usable for train operations, it needs to both be operational, as well as *accessible* by the train. A train cannot pass over the "broken" piece of right-hand side rail just after 5 km, and therefore the entire section of infrastructure thereafter, can be considered as *not usable* – a train will not be able to pass beyond this point (indicated by the red cross).

It is important to note here that fixing the critical fault just after 10 km on the lefthand side rail will not improve the operational state of the CPT-CLM line in any way – the concentrated group of maintenance tasks (accumulating into a critical error) will still exist and therefore the latter half of the line will still *not be usable* for train operations.

This scenario explains the stakeholder need noted as SR1 in Table 4, where the attendance to critical faults does not necessarily guarantee passenger-ready infrastructure to be in a usable state.

The resolution (level of detail) at which parameters should be investigated are also heavily dependent on their use. For example, the TGIV discussed in Chapter 3 measures at a resolution of every 250 mm, but maintenance managers might only require the information at a resolution of 1 m (i.e., be able to notice if there is an exceedance on any 1 m stretch of infrastructure). Providing them with *all* the information (at every 250 mm) might, therefore, reduce the value of the information provided.

5.2.5.3 Integrating domain-knowledge into the services

The integration of domain-knowledge can be broken down into two areas: the *facets* that influence the result, and the *relation* between those different facets. The result itself is also a facet. For example, the usability of a piece of rail is a facet of that piece of rail that is determined by two other facets: its own operational state, and the operational state of the rails leading up to it. These facets relate to one another in the manner described in the scenario above – the rail itself, *as well as* the rails leading up to it (both facets) should be operational, for the rail to be considered usable.

Apart from the *operational state* of an asset, the facets that are considered in this case study, are the *usability* of asset(-groups), the *locations* (from the central station) of measured parameters, the *threshold values* of those parameters, the *number of threshold exceedances* for a certain asset(-group), the *dates* at which measurements were taken, and the *resolution of information* required by maintenance personnel.

Facets are either available internally already (e.g. the operational state of a DT is already contained within itself) or need to be obtained from other components (through services). An example of where a component does not have the information it requires, is in the case of the comparison service which needs to obtain the information of the number of threshold exceedances of the DTs that it will be comparing, from those DTs. The relationships between the different facets are use-case specific and are therefore contained in the services that apply them.

The operational state of an asset is included as an attribute of the asset – assigned via the configuration file and obtained by querying DT-S1. In the implementation of the technology demonstrator for this case study, the system configurator is responsible for changing the operational state of the DT to match that of the physical entity. Future work can improve the DT-service (DT-S1) to provide an automatic evaluation of the operational state of the asset, based on its condition data, instead of just conveying the attribute.

The threshold values of the measured parameters, and the resolution of information required by maintenance personnel, are programmed into the services that use them (DT-S3 and DT-S4) for the concept demonstrator, but it would only require simple changes to read the thresholds from a configuration file instead.

5.2.5.4 Interactions for service-provision

Each of the services is provided by a series of basic interaction sequences (described in Section 4.4.3) - from the UI to the service-providing component (from where more basic interaction sequences can be made), and back to the UI. To illustrate this, the component interactions required to perform the service of comparing the number of threshold exceedances between the electrical infrastructure and the ballast, are illustrated in Figure 19.

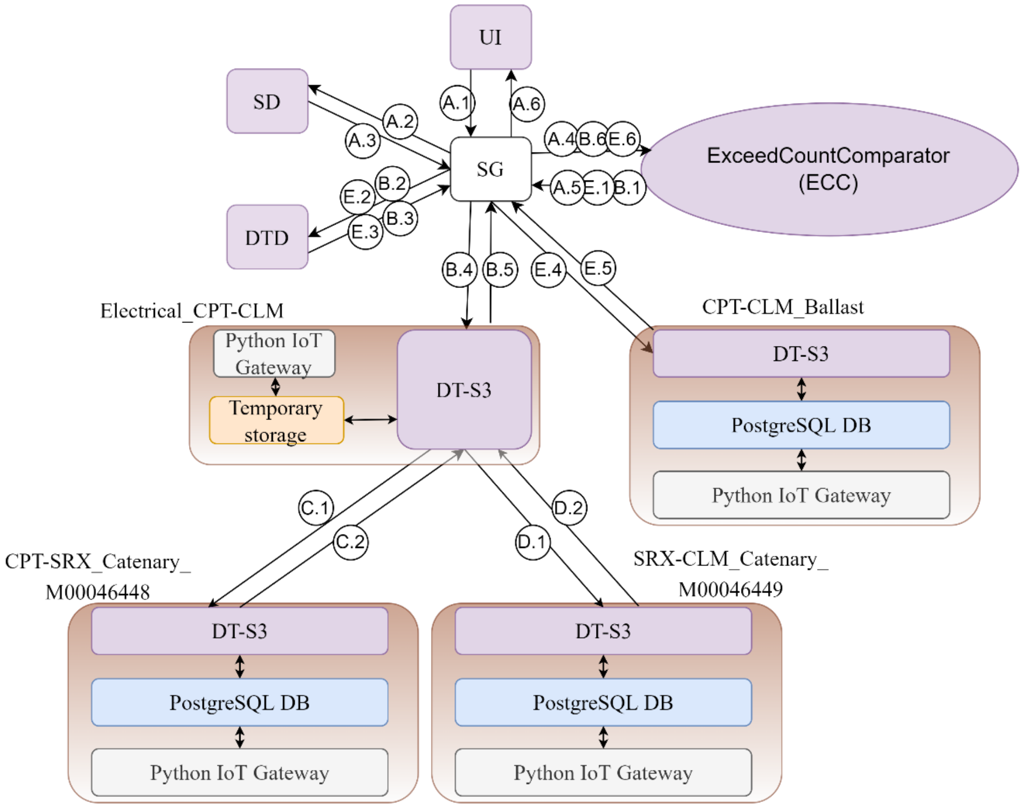


Figure 19: Interactions for comparison service implementation

The sequence of interactions required to provide this service illustrates the implementation of all the principles communicated in Section 4.4.3. Note that Figure 19 purposely omits the wrappers (and, by extension the data sources) that the DTs also interact with, because they are not queried as part of this interaction sequence – the data for the DTs is assumed to already exist in the PostgreSQL databases.

A total of five basic interaction sequences can be seen in Figure 19. The letters depict distinct basic interaction sequences, and the numbers next to the letters depict the step in each of the respective basic interaction sequences, as explained in Section 4.4.3. For example, A.3 is the third step in the first basic interaction sequence that was initiated in the sequence. Interactions A, B, and E are intergroup interactions, whereas interactions C and D depict intragroup interactions. Instead of having to ask the directory services for the IISs of the catenary DTs again, the intragroup basic interaction sequences query the other DTs directly from Layer 6 of the F-DT, after it obtains the IISs of the two catenary DTs from its own temporary storage layer (Layer 3 in SLADTA, where they have been stored subsequent to previous interactions).

Basic interaction sequence A is initiated by the user, from the UI as the user requests for the information of the comparison. The back-end of the UI will know that it is the SN-service labelled "*ExceedCountComparator*" (or ECC) that is responsible for providing this service, and will route this request to the SG (A.1). The SG will obtain the IIS of the *ECC* service from the SD (A.2-A.3), create a TCP/IP socket connection to the service, and request the comparison from the specified DTs from the *ECC* service (A.4).

Upon obtaining this request, the *ECC* service needs to first obtain the number of exceedances from the *Electrical* DT (B.1-B.6), thereafter it should also obtain the number of exceedances of the *Ballast* DT (E.1-E.6) before it can perform its own function of comparing the two and providing the UI with a resulting string (A.5).

The number of threshold exceedances are provided to the requesting component, through the DT-S3 service within the DTs' Services Layer (Layer 6). The processes of obtaining the total number of exceedances from the two DTs are the same (basic interaction sequences C and D), with the former only requiring the additional two intragroup interactions for it to know what its own total is.

The DT-services themselves are implemented as methods within the DT python class, whereas the *ECC* is implemented as a separate class with its own methods and initialisation sequences (among which is its registration, as discussed in Section 4.4.1).

6 Evaluation

This chapter evaluates the developed technology demonstrator described in Chapter 5. The evaluation is based on three sets of tests, discussed in Section 6.1. Based on these tests, the DT system functionality is verified in Section 6.2 (whether it meets the FRs specified in Section 3.3). The generality of the case study results is then evaluated in Section 6.3 by validating the representative nature of the technology demonstrator. Finally, the DT system *architecture* is evaluated in Section 6.4, by discussing the observed benefits.

6.1 Tests

The tests performed in this section evaluates the system operation, as well its ability to adapt to changing physical infrastructure, data sources and user requirements. Each test is described, and the results subsequently discussed.

6.1.1 Performance metrics

A selected set of metrics is used to indicate the performance of the DT system technology demonstrator. One must keep in mind that the purpose of the case study is to verify the proposed DT system’s ability to meet the selected set of FRs discussed in Section 3.3. Therefore, the metrics given here are not accompanied by any target values; rather, they will be used to provide some measure of the performance of the implemented system, for comparison in future work. Future work can also provide and measure additional performance metrics.

The performance metrics are:

1. The system response-time – from UI query to display of the results
2. Data ingestion time (latency) (i.e. the time from when the data source is populated with new data, until the data is available in the system)
3. Configurator time required to adapt system functionality
4. Lines of configuration code manipulated to adapt system functionality
5. Developer time required to adapt system functionality
6. Lines of source code manipulated to adapt system functionality
7. Number of components changed to adapt system functionality
8. Maximum system downtime when adapting system functionality

PM1 provides a performance measure for the user experience (FR1); PM2 evaluates the performance of the system's automated data acquisition (FR3); and PM3-PM8 provide an indication of effort required to adapt the system to changing context (FR6). The tests performed in the rest of Section 6.1, will determine the values of these metrics for the developed technology demonstrator.

6.1.2 System operation tests

6.1.2.1 Test description

The purpose of this test is to verify system functionality (with regards to service provision) by querying a variety of DTs, for the services they offer (summarised in Table 11). There are three parts to this test. For the first part, the DTs are queried for their services, and the response times for those services are calculated. The second and third parts verify the system's ability to automatically recognize, acquire, and store new data for the appropriate DTs, and the system's ability to handle removed data sources. These two aspects were tested by adding new data to the simulated SQL data source labelled "*Catenary Exceedances"* in Figure 13, and removing another simulated data source, labelled "*MB3JK2_OHTE_Exceedances"* in Figure 13 from the system by deleting it.

Table 11: Services performed to test system functionality

Service	Viewpoint(s)/Query target(s)		
	Case 1	**Case 2**	**Case 3**
Operational state	Left Rail 0 - 5 km	Catenary SRX-CLM	Ballast
Usability state	Track	Perway	Electrical
Map of Exceedances (>1%)	Track	Perway	Electrical
Chart of Exceedances (1/01/2019-30/07/2022; >1%)	Track	Perway	Electrical
Exceedance comparison (>1%)	Track & Ballast	Catenary SRX-CLM & Catenary CPT-SRX	Perway & Electrical

Table 11 summaries how the stored condition parameters for a specific DT are filtered for where they exceed 1% of their respective threshold. The historical data inspected is for the period of 1 January 2019 to 30 July 2022. Each of the 3 cases are queried 6 times. The resulting response times are indicated in Figure 21.

For the second part of the test, new data is added to the simulated SQL database of the *Catenary Exceendances*. This is done by executing an SQL query on the *Query Tool* in pgAdmin4 (the PostgreSQL GUI), thereby migrating the batch of data recorded for the OHTE on 5 August 2021, and stored on a MS Excel file, to its simulated SQL database. This adds 133 new rows/records to the PostgreSQL database. It is expected that the *Catenary Data* wrapper will recognise that new data has been added to the database, inform the two catenary DTs that obtain data from it this data source that there is new data, upon which they will request their relevant data fields to then be ingested and stored in their respective long-term storage facilities.

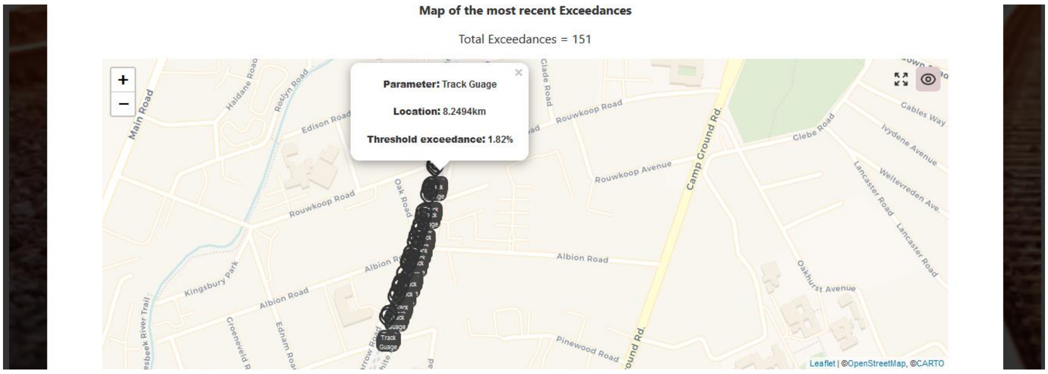
This is verified by asking the *catenary SRX-CLM* DT to provide a chart of the total number of threshold exceedances between 1 January 2019 and 30 July 2022 that exceed a threshold by more than 1% before adding the new data, and then again after adding new data. It is expected that a new datapoint should appear on the chart after new data is added.

The third part of this set of tests reflects the scenario where the original data source becomes unavailable while the DT system is in operation (such as when a server is stolen, data is corrupted, or a proprietary data provider terminates their contract with PRASA). This is simulated by deleting the PostgreSQL data table hosting the database for the *MB3JK2_OHTE_Exceedances* from the server while the system is running for the test. This makes the “original” data (in the source) unavailable to any entity that didn’t duplicate it somewhere else.

The continued availability of the data in the system can be shown by performing the same query as the one performed for the previous part: showing the trend of the total number of measurements for the *catenary SRX-CLM* DT, between 1 January 2019 and 30 July 2022 that exceed their respective threshold by more than 1%. If the DTs do indeed store their data, the data should *still* be available and ready to be viewed by the user (even though the original data is deleted).

6.1.2.2 Test results and discussion

Figure 20 illustrates the output of the “Map” service for the track DT, as specified above, as an example of the output to the UI. The rest of the visual results, as displayed on the UI, can be seen in Appendix A.



Map of the most recent Exceedances

Total Exceedances = 151

Parameter: Track Guage

Location: 8.2494km

Threshold exceedance: 1.82%

Figure 20: Result of the “map” service test, as applied to the perway assetgroup

The response times for the first part of this set of tests, are plotted in the graph shown in Figure 21, below. The response times for the operating condition and usability of the infrastructure (which were less than 1 second), were omitted for clarity of the graph.

The x-axis of the chart is labelled according to the convention of <Service_queried> - <Target_case>, where the Service_queried specifications of ‘Map’, ‘Chart’, and ‘Comp’ are the abbreviations used for the Map of exceedances, Chart of exceedances, and Exceedance comparison services, respectively. Target_case refers to the three different cases of Viewpoint(s)/Query target(s) specified in Table 11.

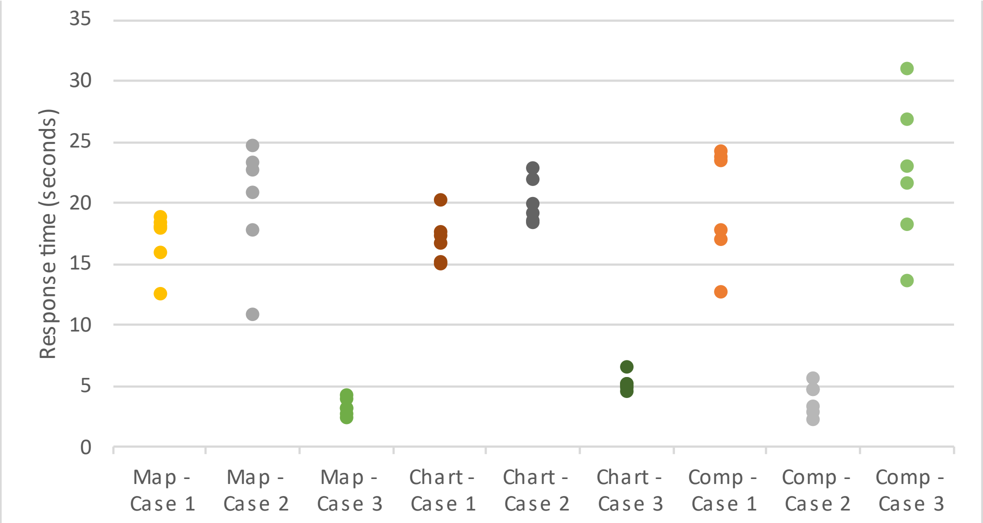


Figure 21: Service response times

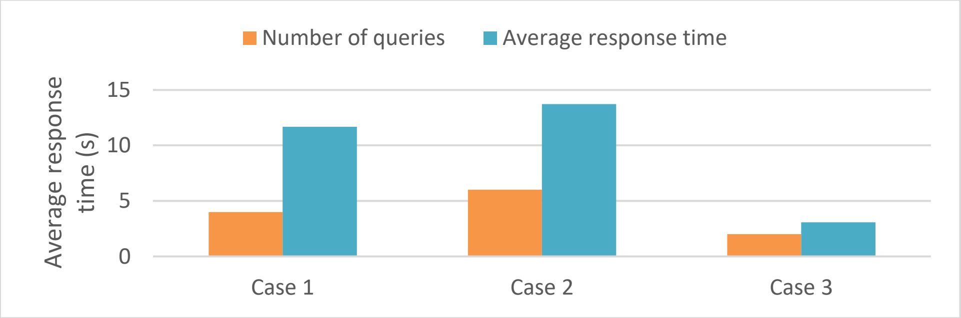


Figure 22: Number of queries' effect on response time

The response times for the technology demonstrator are large – with only the operational state and usability services performing in the millisecond range. The cause of these large response times might involve the number of queries that are needed to perform the service, given the results shown in Figure 22. However, even though the service to provide the operational state of an asset did not have to make any queries to other DTs, the usability service was still required to request the operational state from other DTs as part of its service.

Since both services did not have to process any measurement data, it is more likely that the significant (time-consuming) part in the process of service provision, is as a result of the large volumes of data that had to be processed.

An observation that favours this statement, is that the size of the data that had to be processed by the electrical DT was significantly less than that of the other two DTs: the average number of records ingested at a time by the catenary DTs is 153, while that of the track, ballast and rail DTs are all more than 47 000 each. The relative size of the data that had to be processed for the electrical DT that obtained its data from the catenary DTs, were therefore much smaller. Future work should evaluate the effect of storing the calculations so that services do not need to recalculate or process through all of the data every time that they are queried.

The second part of the tests displayed the following results before and after data ingestion – Figure 23 and Figure 24 respectively. A new set of measurements (taken on 2021-08-05) can be seen in Figure 24 (after data has been added to the data source). It took 14 milliseconds to upload and ingest the data into the DT system. For the third part of the test, the system was able to still obtain data from its own database, even after the source of the data was deleted – reflecting the same output as that seen in Figure 24.

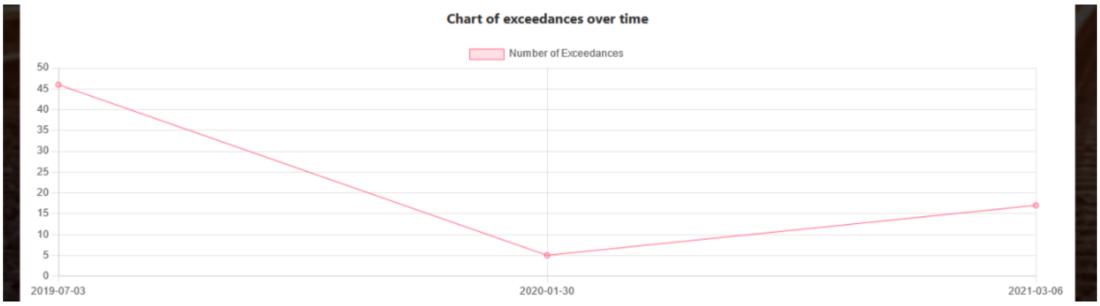


Figure 23: Number of exceedances *before* new data in the source

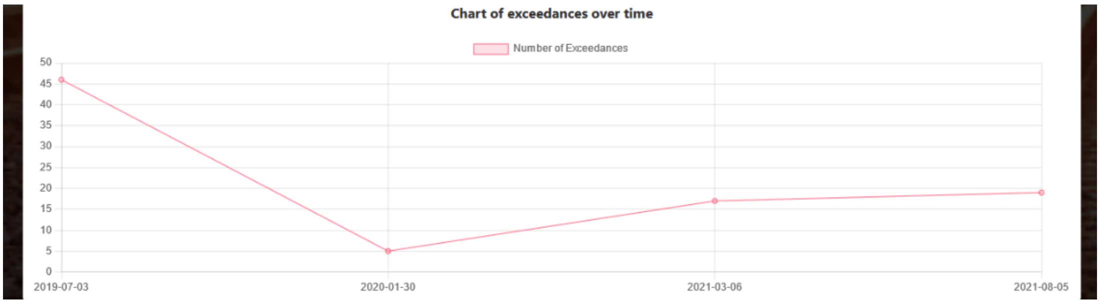


Figure 24: Number of exceedances *after* new data in the source

6.1.3 System reconfiguration test - changing physical infrastructure

6.1.3.1 Test description

This test simulates a scenario where a piece of the catenary wire from Salt River substation to Claremont substation is stolen, vandalised or broken, and the entire catenary wire need to be replaced. In the simulated scenario, the stakeholders desire for the data from the old catenary wire to still be available (for prognostics and health monitoring applications that might be built later). The test determines the process and effort required by the system configurator and developer to adapt the DT system accordingly.

6.1.3.2 Test results and discussion

The system configurator will be responsible for adapting the DT hierarchy to reflect the change in physical infrastructure brought about by the removal and replacement of the catenary wire. Because the data should still be available, the only change that is required to the old DT, is to remove its data acquisition links to the data sources in the system configuration file. This DT is therefore now a representation of the old asset, where the asset does not obtain any new information. For the reconfiguration to take effect, the DT system will now be restarted (as discussed in Section 5.2.2) with the new configuration file.

A new DT should be specified in the configuration file (for the new catenary wire). This can be done by essentially reusing the configuration code of another catenary DT. This test reused the configuration code of the original DT (in its working condition). This required the subsequent adaption of only the DT label, and the date from which the DT should obtain data from its data sources.

Having performed these modifications in the configuration file of the program on Computer A, the system configurator will restart the system.

The results of the reconfiguration tests are listed in Table 12. These reflect the time taken and lines of code changed for such a reconfiguration to have occurred.

Table 12: Reconfiguration test results - changing physical system configuration

Reconfiguration performed	"Removing" old catenary DT		Adding new catenary DT	
Configurator time	2 minutes		3 minutes	
Lines of configuration code manipulated	Copied Modified Removed	0 1 10	Copied Modified Added	70 4 0
Developer time	0 minutes		0 minutes	
Lines of source code manipulated	0		0	
Number of components changed	1		1	

It should be noted that this reconfiguration requires no changes to the source code of the system. The only component that required changes was the configuration file on the one computer. The configurator took 2 and 3 minutes for the reconfiguration. The system downtime is counted from when the system is restarted, till after all the DTs have registered themselves. This was taken to be just under 3 minutes for both of these reconfigurations. Since the DTs need to wait for its constituent DTs to register themselves, the time taken till all of the DTs are registered, is about 3 minutes.

6.1.4 System reconfiguration test - changing user information needs

6.1.4.1 Test description

System reconfiguration to meet changing user needs can be separated into two scenarios: new user needs in terms of the perspective on the data desired, or in terms of the type of service provided. This third test simulates a scenario for each of these. The first scenario is where a new perspective on the data is required, i.e. executive management requires a combined view of the perway and electrical railway infrastructure between Cape Town and Claremont (with the same services as is available for the other DTs). The second scenario is where a completely new service is required, with information from a new (external) database having to be incorporated: the planning office desires a service with which they can view the currently available maintenance tools/materials for any entity in the DT hierarchy. The available materials database is manually updated and kept on a MS Excel file on Computer A. The test determines the process and effort required by the system configurator and developer to adapt the DT system accordingly.

6.1.4.2 Test results and discussion

For the first simulated scenario, a similar process to the one described in Section 6.1.3.1 is required, where the system configurator inserts the configuration code for the new DT in the system configuration file on Computer B, and the program is restarted. This DT is configured to have as constituent DTs the perway and electrical DTs. The fact that the usability of the perway infrastructure, is directly influenced by the usability of the electrical infrastructure and vice versa, is also indicated here.

In the second simulated scenario, a new SN service will need to be created to perform the service required. This new SN service will, for the test, be run on its own on a third computer (C). The materials database, located on Computer A, communicates the available materials per infrastructural domain, each on a respective sheet. A new DT service is therefore added to provide requestors with the infrastructural domain that it forms a part of. This will now be an additional parameter in each of the DTs’ configuration file. A new wrapper component is also created to interface with the materials database – drawing the available materials from the sheet indicated to it. Note that this was optional – a SN service could also have interfaced with the database directly. However, to illustrate the effort required to add a wrapper to the system for a new data source (that might as well provide data to the DT system) this option is selected. The new SN service will query the selected DT to ask it for its infrastructural type, and subsequently query a newly developed wrapper for the available materials for that infrastructural domain and provide the result to the UI.

For the first scenario no source code changes, or new development was required – only additions to the configuration file. However, the second scenario required the development of three new components – the SN service, the DT service, and the wrapper. All these changes had to then also be reflected in the configuration file. The results of the two scenarios are summarised in Table 13, below. Source code manipulations are given as a total – including copied, modified, and added lines of source code. System downtime (from restart till completed registration of all components) were 225 seconds (or slightly under 4 minutes).

Table 13: Reconfiguration test results - changing user needs

Reconfiguration performed	Adding section DT	Whole new service		
		Adding a Wrapper	Adding a SN service	Adding a DT-service
Configurator time	5 minutes	7 minutes		
Lines of configuration code manipulated	Copied 28 Modified 6 Added 0	Copied 9 Modified 4 Added 0	Copied 7 Modified 4 Added 0	Copied 10 Modified 0 Added 1
Developer time	0 min	15 min	135 min	1 min
Lines of source code manipulated	0	102	126	4
Number of components changed	1	1	1	1

6.2 Functional verification

The functional requirements used to develop the reference architecture, described in Chapter 4, were taken as a subset of the larger set of requirements listed in Section 3.3. This subset of requirements was selected based on the objective of this project provided in Section 1.2 and therefore only includes requirements relevant to the scope of the objective.

Section 4.4 provides a detailed discussion of system operation. It is through these operational procedures that the FRs for the DT system are met. In other words, DT system functionality is enabled through DT system operation. This section (6.2) serves to evaluate the functionality of the DT system, by verifying the performance of the selected set of FRs (in the logic of the "back-end" system operation). It highlights how the functions described in Section 4.4, has been met in Section 6.1.1 (the latter serving as verification of the functions' output).

The DT system provides a means through which the physical infrastructural elements can be affected (FR2.a). Specifically, the DT system provides a perspective on the infrastructure that informs human decisions that then lead to inform manual operations (in this case the execution of maintenance tasks). The visual outputs seen in Section 6.1.3 and in Appendix A provide this perspective.

Functional requirements FR6.a, FR6.b and FR6.c all pertain to the adaptability of the DT system. Sections 6.1.3 and 6.1.4 discuss tests that verify that these adaptations can be achieved. Adaptation to changing physical infrastructure (FR6.a) and changing data sources (FR6.c) can be achieved through configuration file changes, or the development of new wrapper components in the case of new data sources being presented to the system. Adaptation to changing user needs (FR6.b) can be achieved through configuration file changes to, e.g., include a new entity of interest in the system, or expand the insights (services) that are offered for the infrastructure's data. These adaptations to the technology demonstrator require relatively little effort – with the longest reconfiguration experiment taking only roughly 2.5 hours. These adaptations are implemented upon system restart, which results in a system downtime of 2-4 minutes for the tested configurations.

Autonomous acquisition of data from data repositories (FR3.a) is performed by the data warehousing process that subsequently stores the data in the DTs' PostgreSQL database, which serves as long-term storage (FR5). This process is described in Section 4.4.2 and is verified with the first test, reported in Section 6.1.1. It should be noted that the case study only considered data from the TGIV - an automatically generated database, typically stored as MS Excel files on a local hard drive. However, different interfaces (REST-API and SQL) were simulated, with a local file also being loaded in the second reconfigurability test (Section 6.1.4); and the data contained within the different TGIV files that were used as data sources, also varied substantially. The second reconfiguration test (Section 6.1.4) illustrated that a new wrapper could be written for a completely new type of data source (with manually updated data, storing completely different kinds of data in a local MS Excel file), within 15 minutes. Since this wrapper exposed the same API to the system as the other wrappers that provide data to the DTs, it can be argued that this could also easily be done for data sources that provide data to the DTs. However, further work will need to verify this.

The data warehousing and federation processes converts data to information. This happens either upon data ingestion – with the wrapper homogenising the data from the data source (FR4.b), or upon data federation for service provision – with the DT or SN services aggregating the information (FR4.c) provided to them by other DT or SN services, to perform their own functions. These processes involve inconsistent features and data types.

The UI is subsequently responsible for presenting the data/information, that the DT system provides it with, in a sensible manner to the user – through charts, maps or text messages (FR4.a). These are subsequently displayed on a dashboard (FR1.a) that is accessible to any user with network access to the DT system implemented by the technology demonstrator. Screenshots of the dashboard can be seen in Appendix B.

6.3 Generality of case study results

The case study results can be extrapolated to be generally applicable to the rest of PRASA's railway infrastructure if the physical scope selection of the case study is representative of a typical case for PRASA's railway infrastructure anywhere on the railway network, and if the architecture is scalable in terms of DTs and services. For scalability, the architecture should be able to adapt to accommodate more DTs, and more services that are implemented for the case study.

It is clear from Section 4.3.4 that the physical *infrastructure* to be represented actually has very little influence on the design and configuration of the DTs within the system. Factors that influence the design and configuration of the DTs are the *relation of each element to other elements in the system*, and the *data sources* that need to be integrated for the DT of that specific infrastructural element, specifically the number of data sources, their interfaces, and the nature (structure and types) of the data they provide. For the technology demonstrator to be applicable to areas of PRASA's infrastructure network other than the Western Cape case study, the case study scope should be representative of the relationship that infrastructural elements within PRASA's railway infrastructure might have to one another, and the variety of data sources that can be expected on PRASA's network (discussed in Section 3.1.3).

The selected physical scope of the case study, and the subsequent DTs that was implemented for the technology demonstrator, illustrated various DT aggregation pathways and relationships between DTs (infrastructural elements). The track FCB-DT aggregates data from both a wrapper and the four rail W-DTs (all run on the same computer); the perway F-DT aggregates data from the FCB-DT of the track, as well as from the W-DT of the ballast (where the latter two are run on a different computer than the perway F-DT). The electrical F-DT aggregates data from two catenary W-DTs (run on a different computer than the electrical F-DT), and the section F-DT (implemented as part of the test in Section 6.1.4 and located on the same hardware as the perway and electrical DTs) aggregates data from another two F-DTs that pertain to different infrastructural domains. The technology demonstrator also illustrates different DTs that pertain to the same asset, but are found at different geographical locations, such as the rail and catenary infrastructure between Cape Town (Traction Main) and Claremont respectively, that are divided into four and two DTs respectively. The case study thus illustrated different relationships between infrastructural elements, as well as different types of data sources and configurations to data sources.

Three types of interfaces were implemented in the DT system technology demonstrator and its tests – authenticated SQL queries to an SQL database, REST-API queries to a simulated web-server and reading from a local MS Excel file. Wrappers were developed for MS Excel files that were both manually and automatically created and updated. These different wrappers also often provide data to the same DTs, as in the case of the ballast DT – where it integrates data from a wrapper that obtains data through SQL queries, as well as from a wrapper that obtains data through a REST-API. These are all the different interfaces that were implemented for the case study. All but the proprietary, authenticated web servers Section 3.1.3, were therefore implemented for the technology demonstrator. However, this statement is based on the knowledge of the different interface types encountered on PRASA's railway network in the Western Cape, as discussed in Section 3.1. Future work will need to verify the fact that the infrastructural domains across the different provinces do not differ significantly.

The both the catenary DTs obtain their data from the same wrapper. The rail DTs also obtain their data from the same wrapper (different from that of the catenary DTs'). In contrast to this, the track DT obtains location data (latitude, longitude and distance from Cape Town station) from its constituent rail DTs, while the condition (or geometry) of the track is provided by the wrapper. Therefore, the track DT needs to integrate not only data from different interfaces and different type of data sources, but also with different uses and implications (it is not all just condition data). By illustrating that the DT system technology demonstrator can integrate these different data sources' data, it is shown that the proposed DT system can integrate data from various other data sources and/or DTs as well.

Given all this information, it can thus be stated that the DTs in the technology demonstrator provide a representative case for PRASA's railway infrastructure network. A diverse set of relationships between infrastructural elements, as well as a data source interactions, are represented. These observations illustrate the suitability of the technology demonstrator to a wide variety of contexts. The scalability of the technology demonstrator is illustrated through the reconfiguration tests performed in Sections 6.1.3 and 6.1.4. These illustrate the capability of the technology demonstrator to adapt and include new infrastructural elements (DTs) as well as services (both DT as well as SN services). The results obtained with the technology demonstrator, can therefore be extrapolated to the broader PRASA infrastructural network, or any context where data from diverse data sources need to be consolidated in the creation of a DT system.

6.4 Architectural evaluation

Data sources encountered in the context of railway infrastructure are often in the form of data silos. This presents a challenge to the effective utilisation of all the data for different use cases. Integration of these data silos, without creating another “solutions silo”, is a particular challenge.

The DT system architecture presented by this study overcomes the technical challenges of integrating data from silos (NFR4-NFR9) through a combination of a DT hierarchy and a WL. The DT hierarchy provides an intuitive subdivision of concerns related to the physical infrastructural network – with DT components providing the possibility to introduce a new “area” of concern. Also, because a strictly one-to-many relationship between DTs in the DT hierarchy is not enforced, different aggregation pathways are possible. This allows for different stakeholders' concerns to be easily managed. The second reconfiguration test performed on the case study, reported in Section 6.1.4, provides an excellent example – the DT hierarchy implemented for the case study was initially based on infrastructural domain type and physical proximity (as presented in Section 5.2). The test’s addition of the section DT provided a new perspective - reflecting all the infrastructural elements represented in the system, irrespective of the infrastructural domain they belong to.

The provision of software services through both the DT hierarchy and the SN allows the service to be hosted where it is the easiest to maintain. If the service only requires the data of a particular DT, then it is best to let that DT host the service. Otherwise, the service can be placed in the SN which allows it the flexibility to draw information from multiple sources – both DTs and others. The value of separating service into these two categories is exemplified in the second reconfiguration test performed on the case study. In the test, a service to provide the available materials for a specific infrastructural element could be split into two – with the DT service providing the infrastructural domain that its specific component belongs to, and the SN service having the flexibility to query both the DT for its domain, as well as then being able to query the materials database (through a wrapper) to establish the available materials for that infrastructural domain at the time of the query. Similarly, keeping the functions that gather and prepare information for the users separate from the dashboard generation, simplifies changing either the one or the other, as the needs of users or the users themselves change.

Another important feature to note is that each of the components (particularly the DTs and the services in the SN) operate as separate holons (Koestler, 1967) –

autonomous and independent of the other components. These holons are loosely coupled and co-operative by sending or responding to asynchronous messages containing data, information, requests, etc.

The holonic behaviour can also be exploited by letting the different components of the DT system run as different processes (on the same or different computational hardware) or as different threads on the same hardware. The system components can therefore easily be distributed across computational resources. This is proven in the second reconfiguration test (reported in Section 6.1.4). In the case study, the DT system was distributed across two computers, with different components running as separate threads on the respective computers. In the reconfiguration test, a new SN service was implemented and included - running as its own process, on a computer that was not included in the system before.

This holonic nature simplifies the reconfiguration of the DT system to reflect changes to the physical reality or the users' needs. Changes to user needs will be catered for by providing new services in the SN, or by adding DT services to the DTs in the hierarchy, as illustrated in the second reconfiguration test reported in Section 6.1.3. Physical infrastructural changes, such as the addition/removal of infrastructure, can easily be implemented by adding/removing DTs from the system, as illustrated in the first reconfiguration test, reported in Section 6.1.3. Changes in data sources occur when, e.g. a company is contracted to provide new data on the condition of a piece of infrastructure that already has a DT component in the system. Such changes primarily affect the WL and the DT configuration files.

The reconfiguration tests reported in Sections 6.1.3 and 6.1.4 illustrate how the modular nature of the architecture enables significant reconfigurations of the DT system through changing only configuration data or files (without any software/”source” code changes). Even though the technology demonstrator required a restart of the DT system when such reconfigurations had to be made, the time taken to implement the reconfigurations were minimal. Reconfiguration through the configuration file therefore minimises the system downtime and the effort required to maintain an up-to-date digital representation of the physical infrastructure.

7 Conclusion and recommendations

This thesis presents the development of a DT system technology demonstrator that was developed for the integration of railway infrastructure data. The technology demonstrator illustrates the value of a DT system architecture that was developed for this purpose, through a case study of a subset of PRASA’s railway infrastructure in the Western Cape province of South Africa.

The research presented here is in many ways along the same vein as recent literature on DTs in the context of railway infrastructure, in that it also highlights the value of DTs for maintenance management. This is because, at the time of the study, maintenance management was the main use of PRASA’s railway infrastructure data, and therefore the services offered with the DT system were focused on the use case of maintenance management. However, this research validates the conceptual claims through the actual *implementation* of such a DT system. This fills a specific gap in literature regarding implementation details for DT systems and adds to the growing body of knowledge in the field of DTs in the context of railway infrastructure. Contrary to other implementations in literature, this implementation uses vendor-neutral software and does not incorporate BIM. This minimises possible barriers to the adoption of the proposed DT system architecture.

By adding a WL to the reference architecture of Human (2022), and using different configurations of SLADT, presented by Redelinghuys, Basson and Kruger (2018), the developed DT system could facilitate the integration of data from different data silos. This functionality could then be used for the purposes of digitally representing the physical infrastructure by semantically connecting (and possibly storing) the data for each DT, and by providing services to users, either through DTs themselves or through services situated in the SN of the DT system. The DT system thus provides a single source of information for the railway infrastructure, leading to a holistic view of the railway infrastructure, and thereby meeting major stakeholder requirements expressed by PRASA for such a software system.

The case study illustrated how a significant amount of PRASA’s recorded requirements could be met. The requirements that were not met, were considered either outside the scope of this project, or is argued to be relatively straight-forward to adopt, given more time. The services of the DT system technology demonstrator, implemented for the case study, presented the operational state of infrastructural elements, the usability of infrastructural elements for train operations, the number of threshold exceedances over time, as well as the locations of the latest exceedances of measured parameters on the infrastructure. However, the response-times for these services were longer than would be desired. It is suspected that this can be ascribed to the large volumes of data that needs to be processed during the execution of these services.

The case study also illustrated the DT system technology demonstrator's ability to be reconfigured by a system configurator, to adapt to changing infrastructure, user needs and data sources. Also, because infrastructural data can be stored and integrated for the asset(s) represented, the data would continue to be available even if the data from third-party data providers are lost.

Even though the technology demonstrator was implemented for only a small part of the infrastructure that exists on PRASA's network, the developed DT system can be considered suitable for the rest of PRASA's railway infrastructure network, and even other contexts with data silos as well. This is due to the selected scope of the case study, which is representative of the different scenarios that the technology demonstrator might be exposed to in such a broader context: diverse data sources and interfaces, complex relationships between infrastructural elements, and changing requirements and infrastructure.

Having accomplished and highlighted all the above, this research also inspired thoughts for future work. Future work should evaluate the effect of *storing intermediate results* of the services, thereby reducing the processing time of this implementation. Future work should also evaluate the given DT system architecture when *integrating other types of data sources* such as from IoT devices and sensors, humans, and paper-based data sources, and whether this architecture can be *expanded to include automatic decision-making* (e.g. with regards to the prioritisation of maintenance tasks). The *inclusion of "hot-updating" functionality* of the DT system configuration would be valuable in reducing the required time and effort for reconfiguration of the DT system even further. To do this, the adaption of the entry-point file in the given implementation, to more closely resemble the "configuration server" presented by Human (2022), could be considered. In terms of services, the application of ML and other data analytics services as *microservices in the service network*, as has been done with vendor-specific solutions in literature, can add tremendous value to the insights obtained from the DT system. The suitability of the architecture to contexts other than PRASA's railway infrastructure should also be evaluated.

Some key considerations for PRASA in implementing this system further, would be to *investigate the adequacy of the available computational resources*, *ensure adherence to their data-handling policies*, and *compatibility with their internet network constraints*. Research on the *safety, security and reliability* of the DT system, together with the *establishing debugging processes and tools for the system configuration file* (to identify logical errors in the configuration of the DT system, would also be critical for any further development of such a system in an industrial/commercial setting.

In conclusion, the contributions of this thesis and the proposed future work, indicate that the application of DT systems to railway infrastructure networks present promising opportunities to add value through the integration of data. Exciting and impactful research and application lies ahead.

References

Ahmadi, M., Kaleybar, H.J., Brenna, M., Castelli-Dezza, F. and Carmeli, M.S. (2021) ‘Adapting Digital Twin technology in Electric Railway Power Systems’, *2021 12th Power Electronics, Drive Systems, and Technologies Conference, PEDSTC 2021* [Preprint]. Available at: https://doi.org/10.1109/PEDSTC52094.2021.9405876.

Augusteijn, N. (2021) *Digital Twins help analyse the future traction energy supply*, *RailTech.com*. Available at: https://www.railtech.com/infrastructure/2021/08/11/using-digital-twins-for-traction-energy-analyses/?gdpr=deny (Accessed: 28 October 2022).

Barricelli, B.R., Casiraghi, E. and Fogli, D. (2019) ‘A survey on digital twin: Definitions, characteristics, applications, and design implications’, *IEEE Access*, 7(Ml), pp. 167653–167671. Available at: https://doi.org/10.1109/ACCESS.2019.2953499.

Boje, C., Guerriero, A., Kubicki, S. and Rezgui, Y. (2020) ‘Towards a semantic Construction Digital Twin: Directions for future research’, *Automation in Construction*, 114. Available at: https://doi.org/10.1016/J.AUTCON.2020.103179.

Borangiu, T., Oltean, E., Răileanu, S., Anton, F., Anton, S. and Iacob, I. (2020) ‘Embedded digital twin for ARTI-type control of semi-continuous production processes’, *Studies in Computational Intelligence*, 853, pp. 113–133. Available at: https://doi.org/10.1007/978-3-030-27477-1_9/COVER.

Boschert, S., Heinrich, C. and Rosen, R. (2018) ‘Next generation digital twin’, in I. Horváth, J.P.S. Rivero, and P.M.H. Castellano (eds) *Implementation, application and utilization of smart systems*. TMCE.

Brodt, A., Schiller, O., Sathish, S. and Mitschang, B. (2011) ‘A mobile data management architecture for interoperability of resource and context data’, in *12th IEEE International Conference on Mobile Data Management*. IEEE. Available at: https://doi.org/10.1109/MDM.2011.81.

Callcut, M., Agliozzo, J.-P.C., Varga, L. and Mcmillan, L. (2021) ‘Digital Twins in Civil Infrastructure Systems’. Available at: https://doi.org/10.3390/su132011549.

Chokshi, N. and McFarlane, D. (2008) ‘A distributed architecture for reconfigurable control of continuous process operations’, *Journal of Intelligent Manufacturing*, 19(2), pp. 215–232. Available at:

https://doi.org/10.1007/s10845-008-0075-7.

Ciavotta, M., Alge, M., Menato, S., Rovere, D. and Pedrazzoli, P. (2017) ‘A Microservice-based Middleware for the Digital Factory’, *Procedia Manufacturing*, 11, pp. 931–938. Available at: https://doi.org/10.1016/J.PROMFG.2017.07.197.

Ciavotta, M., Bettoni, A. and Izzo, G. (2018) ‘Interoperable meta model for simulation-in-the-loop’, *Proceedings - 2018 IEEE Industrial Cyber-Physical Systems, ICPS 2018*, pp. 702–707. Available at: https://doi.org/10.1109/ICPHYS.2018.8390793.

Ciavotta, M., Maso, G.D., Rovere, D., Tsvetanov, R. and Menato, S. (2020) ‘Towards the digital factory: A microservices-based middleware for real-to-digital synchronization’, *Microservices: Science and Engineering*, pp. 273–297. Available at: https://doi.org/10.1007/978-3-030-31646-4_11/COVER.

Cockerell, S. (2022) *Network Rail Uses Digital Twin for Upgrade*, *ARC Advisory Group*. Available at: https://www.arcweb.com/blog/network-rail-uses-digital-twin-transpennine-route-upgrade (Accessed: 28 October 2022).

Costin, A., Adibfar, A., Hu, H. and Chen, S.S. (2018) ‘Building Information Modeling (BIM) for transportation infrastructure – Literature review, applications, challenges, and recommendations’, *Automation in Construction*, 94, pp. 257–281. Available at: https://doi.org/10.1016/J.AUTCON.2018.07.001.

Doubell, G.C., Basson, A.H., Kruger, K. and Conradie, P.D.F. (2022) ‘A Digital Twin System to Integrate Data Silos in Railway Infrastructure’, in *Under review for Proceedings of SOHOMA 2022*.

Doubell, G.C., Kruger, K., Basson, A.H. and Conradie, P. (2021) ‘The Potential for Digital Twin Applications in Railway Infrastructure Management’, *World Congress on Engineering Asset Management*, pp. 241–249. Available at: https://doi.org/10.1007/978-3-030-96794-9_22.

Febrianto, E., Butler, L., Girolami, M. and Cirak, F. (2021) ‘A Self-Sensing Digital Twin of a Railway Bridge using the Statistical Finite Element Method’, *arXiv preprint arXiv:2103.13729*, pp. 1–15.

Fernando, L., Durão, C.S., Haag, S., Anderl, R., Schützer, K. and Zancul, E. (2018) ‘Digital twin requirements in the context of Industry 4.0’, *IFIP international conference on product lifecycle management*, (July), pp. 204–214.

Fuller, A., Fan, Z., Day, C. and Barlow, C. (2020) ‘Digital Twin : Enabling

Technologies , Challenges and Open Research', *IEEE Access* [Preprint], (June). Available at: https://doi.org/10.1109/ACCESS.2020.2998358.

Gao, Y., Qian, S., Li, Z., Wang, P., Wang, F. and He, Q. (2021) 'Digital twin and its application in transportation infrastructure', *Proceedings 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence, DTPI 2021*, pp. 298–301. Available at: https://doi.org/10.1109/DTPI52967.2021.9540108.

Grieves, M. and Vickers, J. (2017) 'Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems', *Transdisciplinary Perspectives on Complex Systems*. Edited by F.-J. Kahlen, S. Flumerfelt, and A. Alves, (August 2017), pp. 85–113. Available at: https://doi.org/10.1007/978-3-319-38756-7.

Gurdur Broo, D. and Schooling, J. (2021) 'A Framework for Using Data as an Engineering Tool for Sustainable Cyber-Physical Systems', *IEEE Access* [Preprint]. Available at: https://doi.org/10.1109/ACCESS.2021.3055652.

Human, C. (2022) *A Design Framework for Aggregation in a System of Digital Twins. Stellenbosch University*.

Human, C., Basson, A.H. and Kruger, K. (2021) 'Digital Twin Data Pipeline Using MQTT in SLADTA', *Studies in Computational Intelligence*, 952, pp. 111–122. Available at: https://doi.org/10.1007/978-3-030-69373-2_7.

Japan International Cooperation Agency (2013) 'Data Collection Survey on Railway Sector in the Republic of South Africa', (February).

Jiang, F., Ding, Y., Song, Y., Geng, F. and Wang, Z. (2021) 'An Architecture of Lifecycle Fatigue Management of Steel Bridges Driven by Digital Twin', *Structural Monitoring and Maintenance*, 8(2). Available at: https://doi.org/10.12989/smm.2021.8.2.187.

Johnson, A., Heaton, J., Yule, S., Luke, S., Pocock, D., Parlikad, A.K. and Schooling, J. (2021) 'Informing the information requirements of a digital twin: a rail industry case study', *Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction*, pp. 1–13. Available at: https://doi.org/10.1680/jsmic.20.00017.

Juarez, M.G., Botti, V.J. and Giret, A.S. (2021) 'Digital twins: Review and challenges', *Journal of Computing and Information Science in Engineering*, 21(3). Available at: https://doi.org/10.1115/1.4050244/1100428.

Kaewunruen, S., Sresakoolchai, J. and Lin, Y. (2021) 'Digital twins for managing railway maintenance and resilience', *Open Research Europe*, 1. Available at: https://doi.org/10.12688/openreseurope.13806.2.

Kampczyk, A. and Dybeł, K. (2021) 'The Fundamental Approach of the Digital Twin Application in Railway Turnouts with Innovative Monitoring of Weather Conditions'. Available at: https://doi.org/10.3390/s21175757.

Kochan, A. (2020) 'Digital Twin of the ECTS application', *WUT Journal of Transportation Engineering*, 131(December 2020), pp. 87–98. Available at: https://doi.org/10.5604/01.3001.0014.8199.

Koestler, A. (1967) 'The Ghost in the Machine', in *The Ghost in the Machine*. New York: The Macmillan Company, pp. 45–58.

Kritzinger, W., Karner, M., Traar, G., Henjes, J. and Sihn, W. (2018) 'Digital Twin in manufacturing: A categorical literature review and classification', 51(11), pp. 1016–1022. Available at: https://doi.org/10.1016/J.IFACOL.2018.08.474.

Kruger, K., Human, C. and Basson, A. (2022) 'Towards the Integration of Digital Twins and Service-Oriented Architectures', *Studies in Computational Intelligence: Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future*, 1034, pp. 131–143. Available at: https://doi.org/10.1007/978-3-030-99108-1_10.

Kuhn, T., Schnicke, F. and Oliveira Antonino, P. (2020) 'Service-Based Architectures in Production Systems: Challenges, Solutions Experiences', *2020 ITU Kaleidoscope: Industry-Driven Digital Transformation, ITU K 2020* [Preprint]. Available at: https://doi.org/10.23919/ITUK50268.2020.9303207.

Love, P.E.D. and Matthews, J. (2019) 'The "how" of benefits management for digital technology: From engineering to asset management', *Automation in Construction*, 107. Available at: https://doi.org/10.1016/j.autcon.2019.102930.

Love, P.E.D., Zhou, J., Edwards, D.J., Irani, Z. and Sing, C.P. (2017) 'Off the rails: The cost performance of infrastructure rail projects', *Transportation Research Part A: Policy and Practice*, 99, pp. 14–29. Available at: https://doi.org/10.1016/J.TRA.2017.02.008.

Lutze, R. (2019) 'Digital Twins in eHealth - : Prospects and Challenges Focussing on Information Management', *Proceedings - 2019 IEEE International Conference on Engineering, Technology and Innovation, ICE/ITMC 2019* [Preprint]. Available at: https://doi.org/10.1109/ICE.2019.8792622.

Malakuti, S. (2021) 'Emerging Technical Debt in Digital Twin Systems', *2021 26th*

IEEE International Conference on Emerging Technologies and Factory Automation (ETFA) [Preprint]. Available at: https://doi.org/10.1109/ETFA45728.2021.9613538.

Malakuti, S., Borrison, R., Kotriwala, A., Kloepper, B., Nordlund, E. and Ronnberg, K. (2021) ‘An Integrated Platform for Multi-Model Digital Twins’, in *IoT ’21: 11th International Conference on the Internet of Things*. Available at: https://doi.org/10.1145/3494322.3494324.

Malakuti, S., Juhlin, P., Doppelhamer, J., Schmitt, J., Goldschmidt, T. and Ciepal, A. (2021) ‘An Architecture and Information Meta-model for Back-end Data Access via Digital Twins’, *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* [Preprint]. Available at: https://doi.org/10.1109/ETFA45728.2021.9613724.

Mangan, T. (2020) *How Digital Twin Technology is Helping Build a Smart Railway System in Italy*. Available at: https://www.nutanix.com/theforecastbynutanix/industry/how-digital-twin-technology-is-helping-build-a-smart-railway-system-in-italy (Accessed: 16 February 2021).

Marocco, M. and Garofolo, I. (2021) ‘Integrating disruptive technologies with facilities management: A literature review and future research directions’. Available at: https://doi.org/10.1016/j.autcon.2021.103917.

Marzouk, M. and Abdelaty, A. (2014) ‘Monitoring thermal comfort in subways using building information modeling’, *Energy and Buildings*, 84, pp. 252–257. Available at: https://doi.org/10.1016/J.ENBUILD.2014.08.006.

Mazayev, A., Martins, J.A. and Correia, N. (2017) ‘Interoperability in IoT Through the Semantic Profiling of Objects’, *IEEE Access*, 6, pp. 19379–19385. Available at: https://doi.org/10.1109/ACCESS.2017.2763425.

Meixedo, A., Santos, J., Ribeiro, D., Calçada, R. and Todd, M.D. (2022) ‘Online unsupervised detection of structural changes using train–induced dynamic responses’, *Mechanical Systems and Signal Processing*, 165. Available at: https://doi.org/10.1016/J.YMSSP.2021.108268.

Melesse, T.Y., Di Pasquale, V. and Riemma, S. (2020) ‘Digital twin models in industrial operations: A systematic literature review’, *Procedia Manufacturing*, 42(2019), pp. 267–272. Available at: https://doi.org/10.1016/j.promfg.2020.02.084.

Minerva, R., Lee, G.M. and Crespi, N. (2020) ‘Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models’, *Proceedings of the IEEE*, 108(10), pp. 1785–1824. Available at: https://doi.org/10.1109/JPROC.2020.2998530.

Ngqakamba, S. (2019) *Ramaphosa visits Prasa about rail improvements | News24*. Available at: https://www.news24.com/news24/SouthAfrica/News/ramaphosa-visits-prasa-about-rail-improvements-20190409 (Accessed: 18 June 2021).

OECD (2019) 'OECD Regional Outlook 2019: Leveraging Megatrends for Cities and Rural Areas', in *OECD Urban Studies*. OECD Publishing. Available at: https://doi.org/10.1787/9789264312838-EN.

OECD (2020) 'Transport Bridging Divides'. OECD Publishing. Available at: https://doi.org/10.1787/55AE1FD8-EN.

Overton, J. and Brigham, J.-C. (2017) *The Digital Twin. Data-driven simulations innovate the manufacturing process, White Paper*. Available at: https://assets1.dxc.technology/analytics/downloads/DXC-Analytics-Digital-Twin.pdf.

Papatolios, N. (2021) *Article - Rail freight SA - private investments against neglect.pdf*, *Rail freight South Africa: private investments against neglect*. Available at: https://www.railfreight.com/railfreight/2021/03/23/rail-freight-south-africa-private-investments-against-obscurity/?gdpr=accept (Accessed: 21 July 2021).

Patel, J. and Member, S. (2019) 'Bridging Data Silos Using Big Data Integration', *International Journal of Database Management Systems*, 11(3). Available at: https://doi.org/10.5121/ijdms.2019.11301.

Pathak, J., Jiang, Y., Honavar, V. and Mccalley, J. (2006) *Condition Data Aggregation with Application to Failure Rate Calculation of Power Transformers*. Available at: https://doi.org/10.1109/HICSS.2006.93.

PRASA (2021) *Corporate Plan 2021/2024*. Available at: https://doi.org/10.3726/978-3-653-06145-1/2.

Redelinghuys, A.J.H. (2020) *An Architecture for the Digital Twin of a Manufacturing Cell*, *PhD Dissertation. Stellenbosch University*. Available at: http://scholar.sun.ac.za/handle/10019.1/108283.

Redelinghuys, A.J.H., Basson, A.H. and Kruger, K. (2018) 'A Six-Layer Digital Twin Architecture for a Manufacturing Cell', in *Service Orientation in Holonic and Multi-Agent Manufacturing: Proceedings of SOHOMA 2018*.

Redelinghuys, A.J.H., Kruger, K. and Basson, A. (2020) ‘A six-layer architecture for digital twins with aggregation’, *Studies in Computational Intelligence*, 853, pp. 171–182. Available at: https://doi.org/10.1007/978-3-030-27477-1_13.

Sanjay Gadge and Vijaya Kotwani (2017) *Microservice Architecture: API Gateway Considerations*. Available at: http://mainlab.cs.ccu.edu.tw/presentation/pdf/(2017)Microservice-Architecture-API-Gateway-Considerations.pdf (Accessed: 18 August 2022).

Shafto, M., Conroy, M., Doyle, R., Glaessgen, E., Kemp, C., LeMoigne, J. and Wang, L. (2010) ‘Modeling, Simulation, information Technology & Processing Roadmap’, *National Aeronautics and Space Administration*, p. 27.

Sharma, A., Kosasih, E., Zhang, J., Brintrup, A. and Calinescu, A. (2020) *Digital Twins: State of the Art Theory and Practice, Challenges, and Open Research Questions*.

Siemens Mobility (2018) *for Singapore mass transit system*. Available at: https://assets.new.siemens.com/siemens/assets/api/uuid:06d4e3a8-f880-48f3-ba8f-4a9f2cbb9c7d/PR2018090305MOEN.pdf (Accessed: 16 February 2021).

Siemens Mobility (2020) *Downtown Line signalling simulation center*. Available at: https://assets.new.siemens.com/siemens/assets/api/uuid:3d8a7cc5-f530-4b00-bac8-854c999294d1/PR201910145606en.pdf (Accessed: 16 February 2021).

Szpytko, J., Yorlandys, · and Duarte, S. (2021) ‘A digital twins concept model for integrated maintenance: a case study for crane operation’, *Journal of Intelligent Manufacturing*, 32, pp. 1863–1881. Available at: https://doi.org/10.1007/s10845-020-01689-5.

Taibi, D., Lenarduzzi, V. and Pahl, C. (2018) ‘Architectural Patterns for Microservices: A Systematic Mapping Study’, *CLOSER 2018: Proceedings of the 8th International Conference on Cloud Computing and Services Science; Funchal, Madeira, Portugal, 19-21 March 2018*, 2018-Janua, pp. 136–147. Available at: https://doi.org/10.5220/0006701101360147.

Tao, F., Sui, F., Liu, A., Qi, Q., Zhang, M., Song, B., Lu, S.C. and Nee, A.Y.C. (2019) ‘Digital twin-driven product design framework’, *International Journal of Production Research*, 7543, pp. 1–19. Available at: https://doi.org/10.1080/00207543.2018.1443229.

Tchana, Y., Ducellier, G. and Remy, S. (2019) ‘Designing a unique Digital Twin for linear infrastructures lifecycle management’, *Procedia CIRP*, 84, pp. 545–

549. Available at: https://doi.org/10.1016/j.procir.2019.04.176.

van der Valk, H., Haße, H., Möller, F., Arbter, M., Henning, J.L. and Otto, B. (2020) ‘A taxonomy of digital twins’, *26th Americas Conference on Information Systems, AMCIS 2020* [Preprint], (August).

Vickers, J. and Grieves, M. (2017) ‘Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems’. Available at: https://doi.org/10.1007/978-3-319-38756-7_4.

Vieira, J., Poças Martins, J., Marques de Almeida, N., Patrício, H. and Gomes Morgado, J. (2022) ‘Towards Resilient and Sustainable Rail and Road Networks: A Systematic Literature Review on Digital Twins’. Available at: https://doi.org/10.3390/su14127060.

Villa, V., Naticchia, B., Bruno, G., Aliev, K., Piantanida, P., Antonelli, D. and Tagliabue, C. (2021) ‘IoT Open-Source Architecture for the Maintenance of Building Facilities’. Available at: https://doi.org/10.3390/app11125374.

Villalonga, A., Negri, E., Biscardo, G., Castano, F., Haber, R.E., Fumagalli, L. and Macchi, M. (2021) ‘A decision-making framework for dynamic scheduling of cyber-physical production systems based on digital twins’, *Annual Reviews in Control*, 51, pp. 357–373. Available at: https://doi.org/10.1016/J.ARCONTROL.2021.04.008.

World Bank, International Finance Corporation and Multilateral Investment Guarantee Agency (2021) *Country Partnership Framework for the Republic of South Africa for the Period FY22-FY26*, *Country Partnership Framework with the Republic of South Africa for the period of FY22 - FY26*. World Bank, Washington, DC.

Xiaodong, W., Feng, L., Junhua, R. and Rongyu, L. (2020) ‘A survey of digital twin technology for PHM’, *Advances in Intelligent Systems and Computing*, 1031 AISC, pp. 397–403. Available at: https://doi.org/10.1007/978-981-13-9406-5_48.

Ye, C., Butler, L., Bartek, C., Iangurazov, M., Lu, Q., Gregory, A. and Girolami, M. (2019) ‘A Digital Twin of Bridges for Structural Health Monitoring’, *12th International Workshop on Structural Health Monitoring 2019* [Preprint].

Yu, G., Wang, Y., Mao, Z., Hu, M., Sugumaran, V. and Wang, Y.K. (2021) ‘A digital twin-based decision analysis framework for operation and maintenance of tunnels’, *Space Technology*, 116, p. 104125. Available at: https://doi.org/10.1016/j.tust.2021.104125.

Zheng, H. (2021) ‘Research and Analysis on the Application of Digital Twin Technology in Urban Rail Transit’, pp. 1067–1070. Available at: https://doi.org/10.1109/ipec51340.2021.9421186.

Appendix A Services' display results

This Appendix contains screenshots of the different outputs as seen on the UI in response to the services. The outputs were obtained with the DT system configuration, as presented in Chapter 5 of the thesis.

A.1 Viewing the operational state of an asset

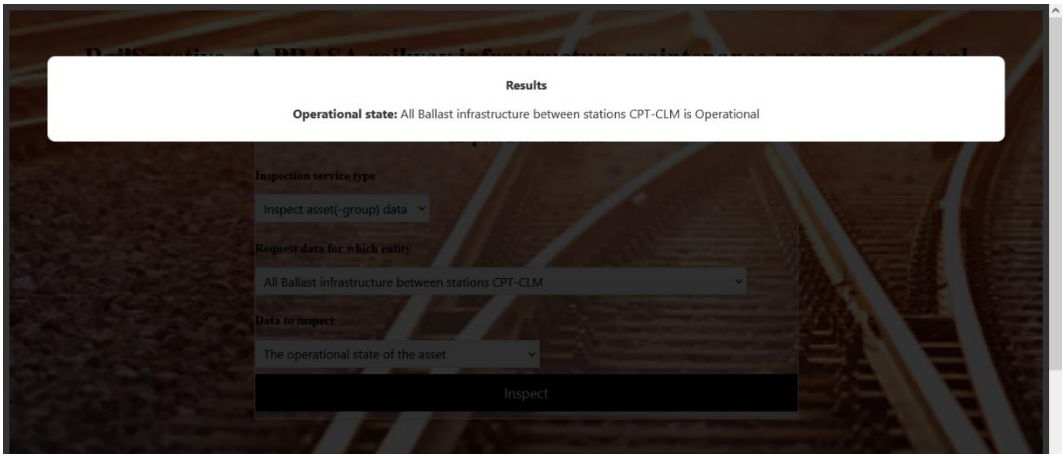
The screenshots seen below are before and after viewing the operational state of the ballast infrastructure between Cape Town and Claremont.

Inspection service type: Inspect asset(-group) data

Request data for which entity: All Ballast infrastructure between stations CPT-CLM

Data to inspect: The operational state of the asset

Figure A-1: Dashboard before querying the operational state of the ballast infrastructure



Operational state: All Ballast infrastructure between stations CPT-CLM is Operational

Inspection service type: Inspect asset(-group) data

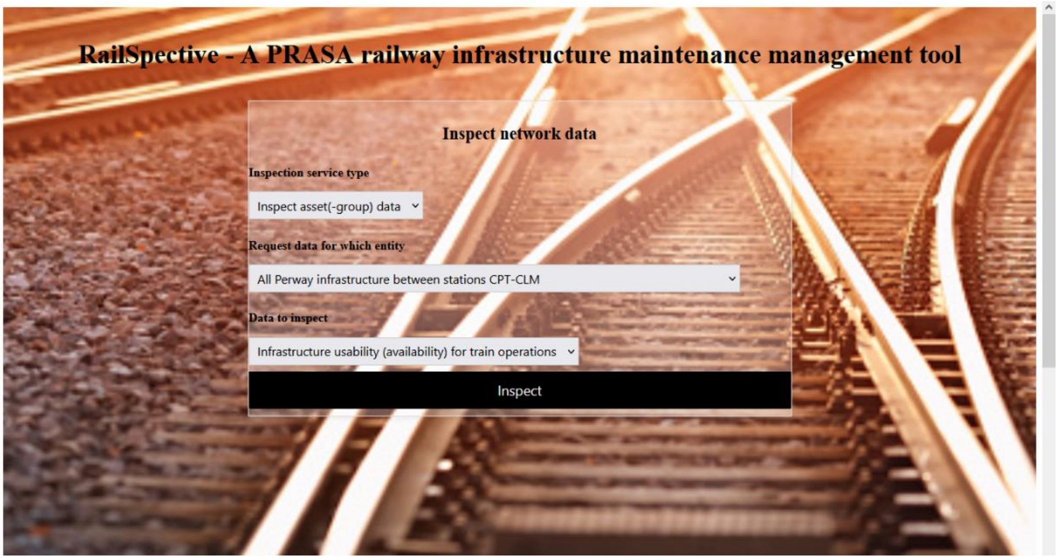
Request data for which entity: All Ballast infrastructure between stations CPT-CLM

Data to inspect: The operational state of the asset

Figure A-2: Dashboard after querying the operational state of the ballast infrastructure

A.2 Viewing the usability of an asset(-group) for train operations

The screenshots seen below are before and after viewing the usability of the perway DT.



RailSpective - A PRASA railway infrastructure maintenance management tool

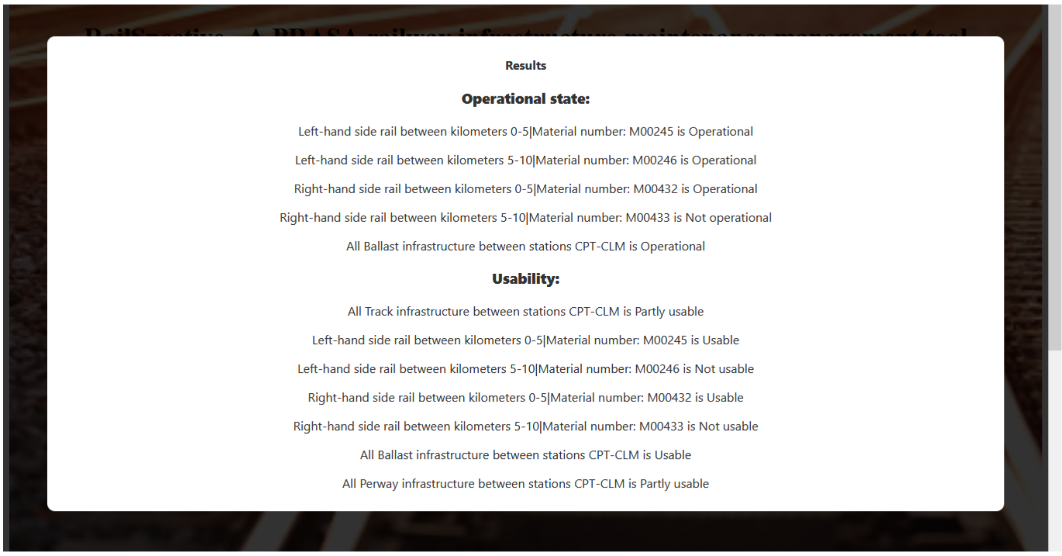
Inspect network data

Inspection service type: Inspect asset(-group) data

Request data for which entity: All Perway infrastructure between stations CPT-CLM

Data to inspect: Infrastructure usability (availability) for train operations

Figure A-3: Dashboard before querying the usability of the perway infrastructure



Results

Operational state:

Left-hand side rail between kilometers 0-5|Material number: M00245 is Operational

Left-hand side rail between kilometers 5-10|Material number: M00246 is Operational

Right-hand side rail between kilometers 0-5|Material number: M00432 is Operational

Right-hand side rail between kilometers 5-10|Material number: M00433 is Not operational

All Ballast infrastructure between stations CPT-CLM is Operational

Usability:

All Track infrastructure between stations CPT-CLM is Partly usable

Left-hand side rail between kilometers 0-5|Material number: M00245 is Usable

Left-hand side rail between kilometers 5-10|Material number: M00246 is Not usable

Right-hand side rail between kilometers 0-5|Material number: M00432 is Usable

Right-hand side rail between kilometers 5-10|Material number: M00433 is Not usable

All Ballast infrastructure between stations CPT-CLM is Usable

All Perway infrastructure between stations CPT-CLM is Partly usable

Figure A-4: Dashboard after querying the usability of the perway infrastructure

A.3 A map of the threshold exceedance locations

The screenshots seen below are before and after viewing the locations where perway has measured parameters that exceed its threshold by more than 1%. With the images indicating different parameters.

RailSpective - A PRASA railway infrastructure maintenance management tool

Inspect network data

Inspection service type: Inspect asset(-group) data

Request data for which entity: All Perway infrastructure between stations CPT-CLM

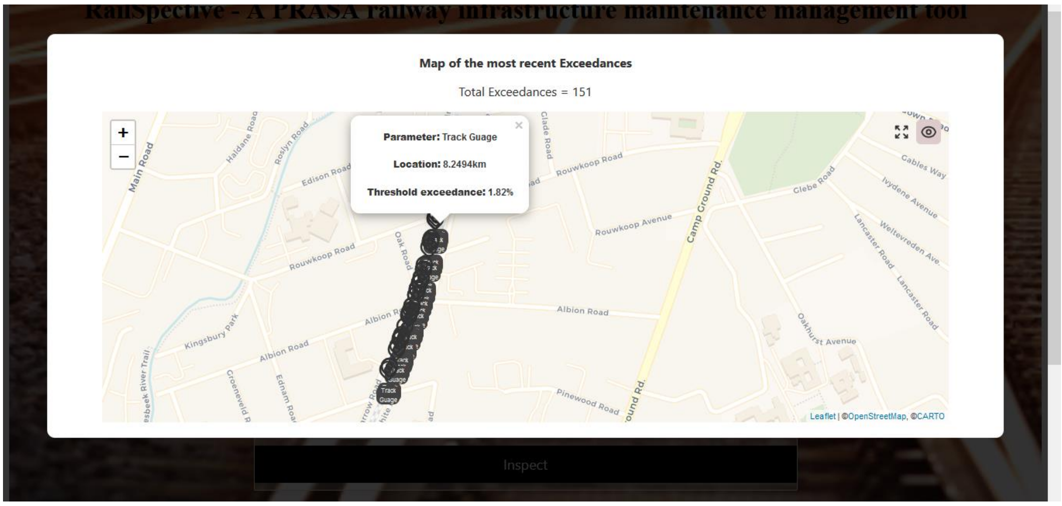
Data to inspect: Map of current exceedances

Specify parameters

Parameters: All

threshold: 1

Figure A-5: Dashboard before querying the map of exceedances



RailSpective - A PRASA railway infrastructure maintenance management tool

Map of the most recent Exceedances

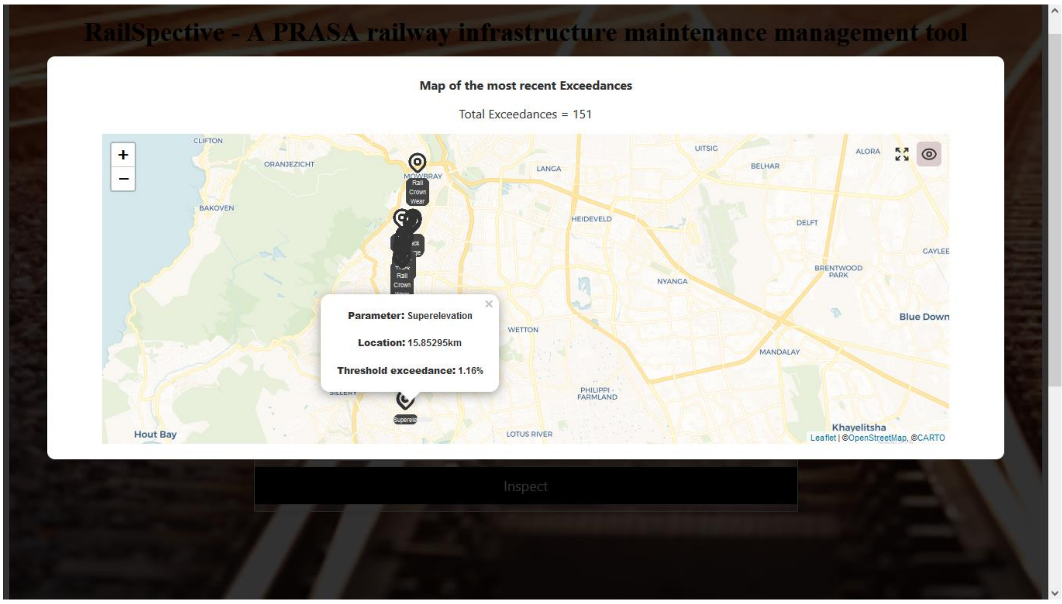
Total Exceedances = 151

Parameter: Track Guage

Location: 8.2494km

Threshold exceedance: 1.82%

Figure A-6: Dashboard highlighting track guage exceedances on map after perway query



RailSpective - A PRASA railway infrastructure maintenance management tool

Map of the most recent Exceedances

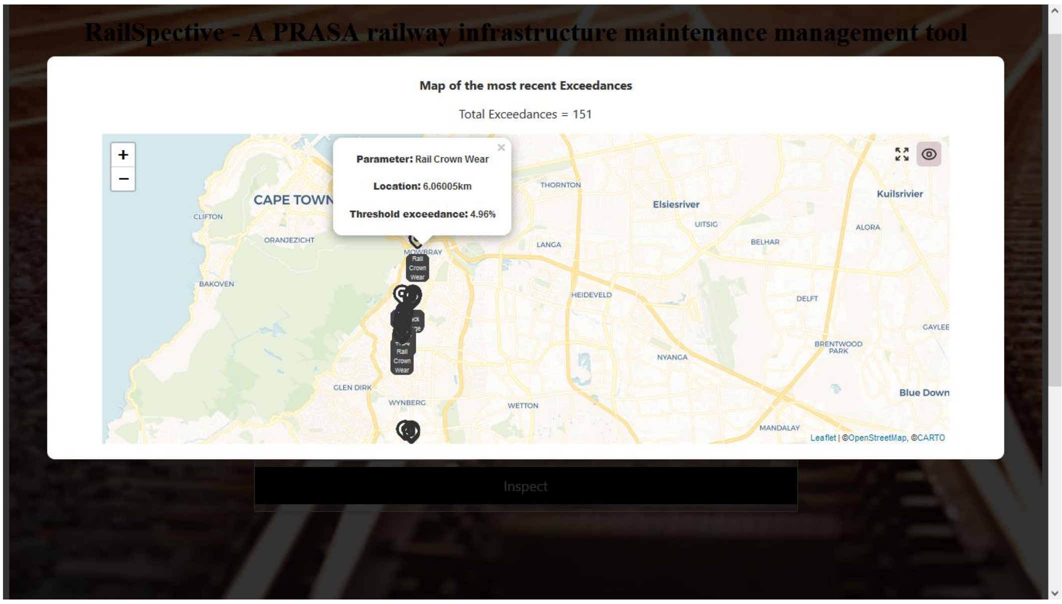
Total Exceedances = 151

Parameter: Superelevation

Location: 15.85295km

Threshold exceedance: 1.16%

Figure A-7: Dashboard highlighting superelevation (ballast) exceedances on map after perway query



RailSpective - A PRASA railway infrastructure maintenance management tool

Map of the most recent Exceedances

Total Exceedances = 151

Parameter: Rail Crown Wear

Location: 6.06005km

Threshold exceedance: 4.96%

Figure A-8: Dashboard highlighting rail crown wear (rail) exceedances on map after perway query

A.4 A chart of the threshold exceedances over time

The screenshots seen below are before and after viewing the threshold exceedances over time, for the electrical infrastructure.

RailSpective - A PRASA railway infrastructure maintenance management tool

Inspect network data

Inspection service type: Inspect asset(-group) data

Request data for which entity: All Electrical infrastructure between substations TM-CLM

Data to inspect: Chart of the number of exceedances over time

Specify parameters

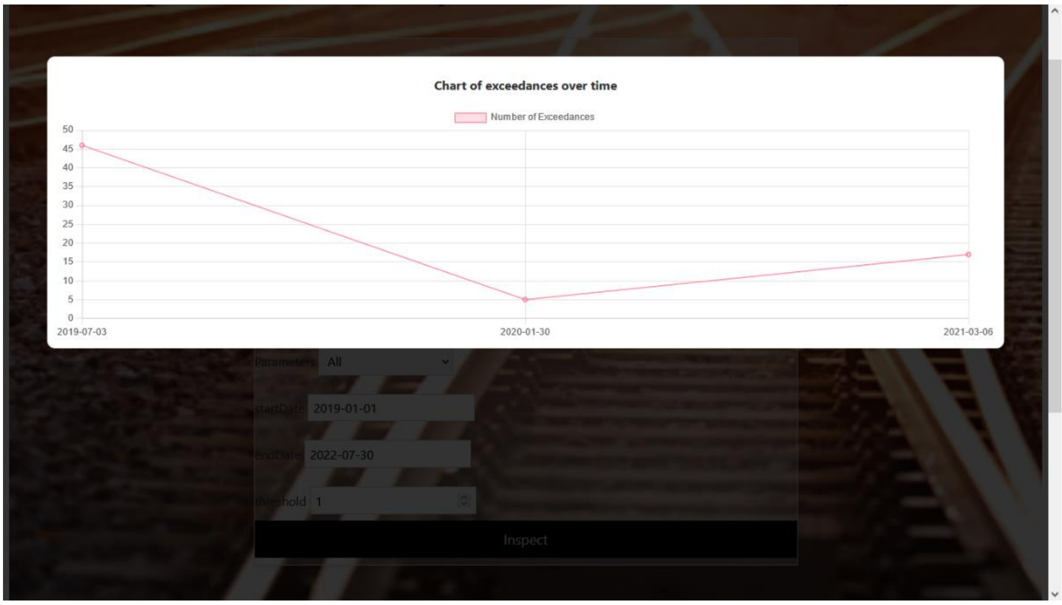
Parameters: All

startDate: 2019-01-01

endDate: 2022-07-30

threshold: 1

Figure A-9: Dashboard before querying exceedances over time for electrical infrastructure



Parameters: All

startDate: 2019-01-01

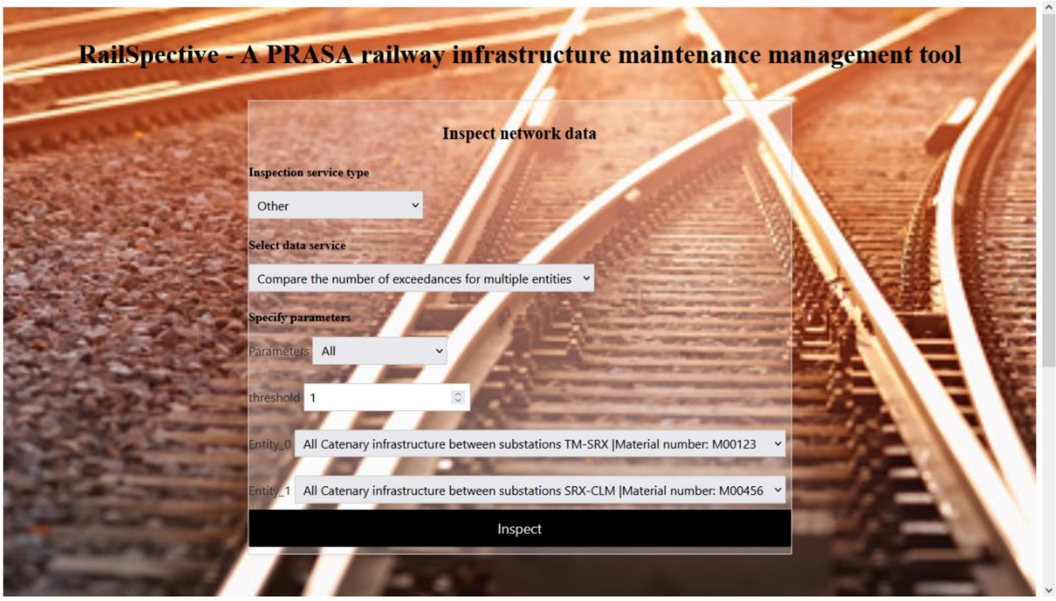
endDate: 2022-07-30

threshold: 1

Figure A-10: Dashboard after querying exceedances over time for electrical infrastructure

A.5 A comparison between the number of threshold exceedances of different assets

The screenshots seen below are before and after inspecting the comparison between two catenary wire assets.



RailSpective - A PRASA railway infrastructure maintenance management tool

Inspect network data

Inspection service type: Other

Select data service: Compare the number of exceedances for multiple entities

Specify parameters

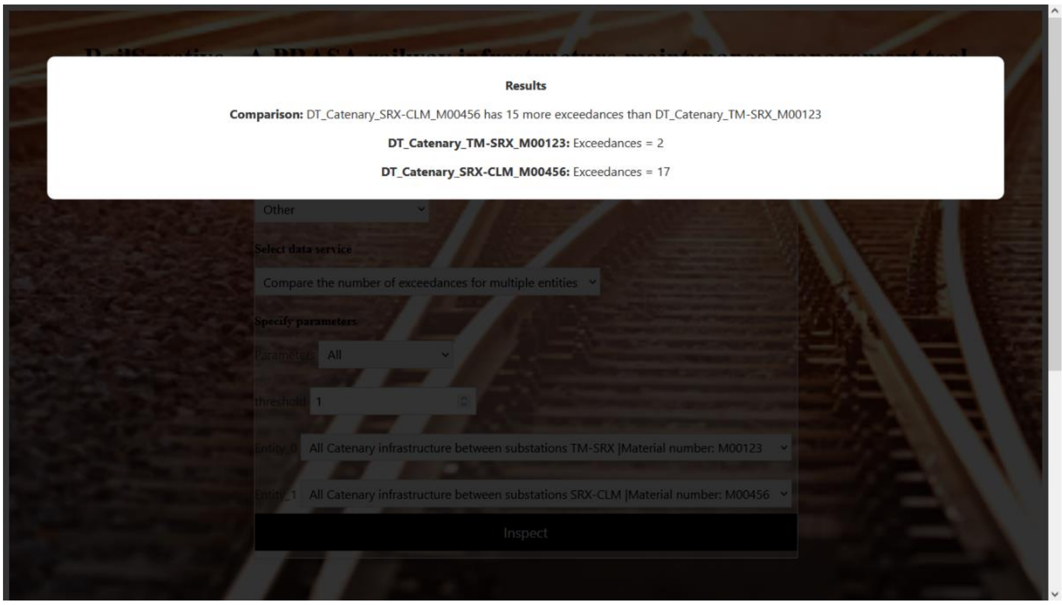
Parameters: All

threshold: 1

Entity_0: All Catenary infrastructure between substations TM-SRX |Material number: M00123

Entity_1: All Catenary infrastructure between substations SRX-CLM |Material number: M00456

Figure A-11: Dashboard before comparison service queried



Results

Comparison: DT_Catenary_SRX-CLM_M00456 has 15 more exceedances than DT_Catenary_TM-SRX_M00123

DT_Catenary_TM-SRX_M00123: Exceedances = 2

DT_Catenary_SRX-CLM_M00456: Exceedances = 17

Other

Select data service: Compare the number of exceedances for multiple entities

Specify parameters

Parameters: All

threshold: 1

Entity_0: All Catenary infrastructure between substations TM-SRX |Material number: M00123

Entity_1: All Catenary infrastructure between substations SRX-CLM |Material number: M00456

Figure A-12: Dashboard after comparison service queried

Appendix B Case study railway map

Figure B-1, below indicates the map of the section of the MB3JK subline used for the case study. Substations along the network are indicated by the houses. The names along the red line indicate the stations.

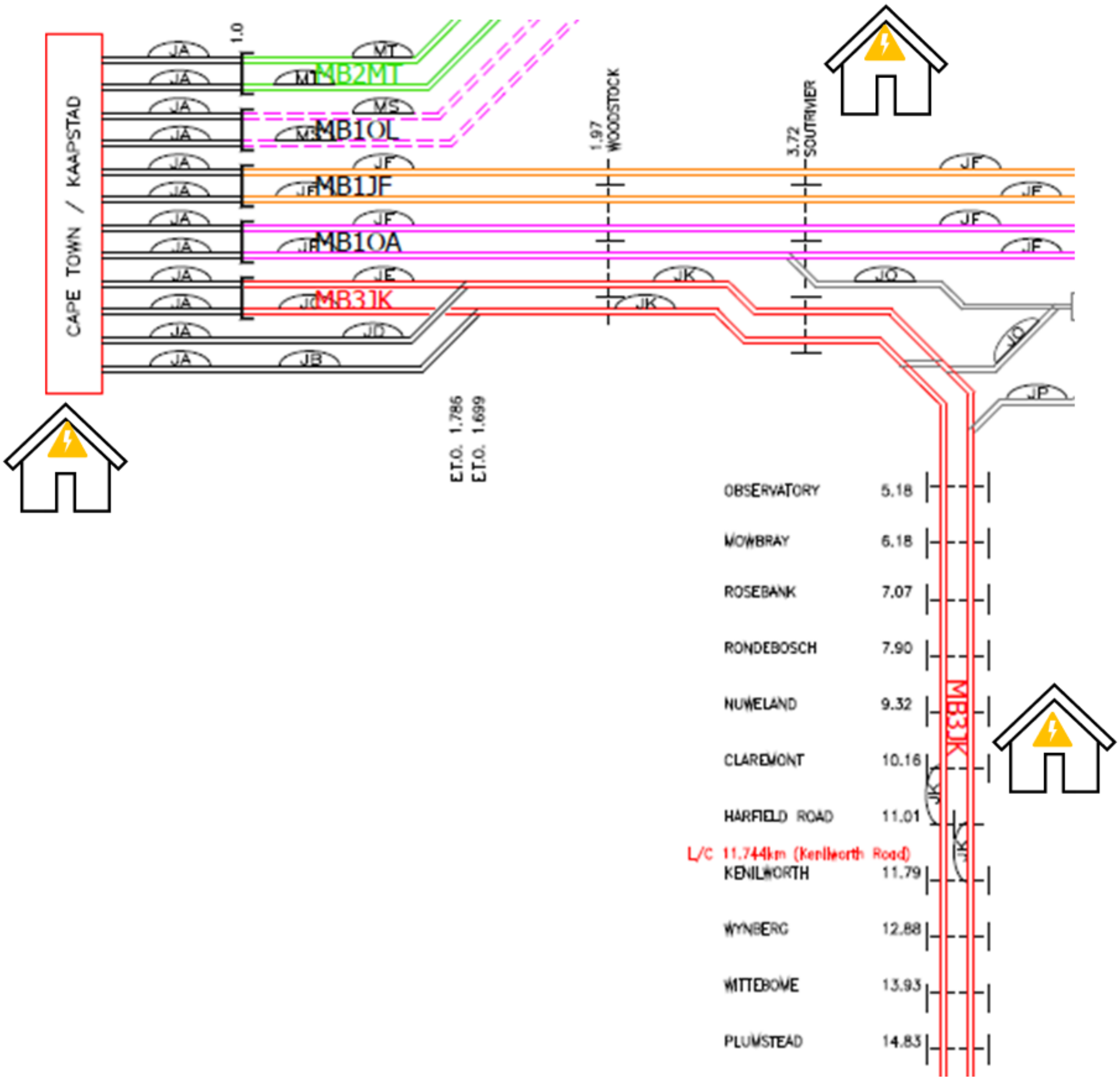


Figure B-1: Case study infrastructure map

Appendix C Case study configuration file

Below is an illustration of the JSON configuration file as would be used for a DT system that has all the management services, two wrappers (*"API_MB3JK2_Layout_wrapper"* and "*SQL_MB3JK2_IMDATA_wrapper*"), one SN service (*"ExceedCountComparator*"), and two DTs (*"DT_Ballast_CPT-CLM" and "DT_Catenary_SRX-CLM_M00456"*).

```
{
    "Mngmt_services": {
        "GATEWAY": {"label":"GATEWAY", "ip":"127.0.0.1", "port":7001,
"dtd_url":"http://127.0.0.1:7002", "dsd_url":"http://127.0.0.1:7003",
"sd_url":"http://127.0.0.1:7004","log_choice":false},

        "DIR_Services":{
            "DTD": {"label":"DTD", "ip":"127.0.0.1", "port":7002,
"storage_ip":"127.0.0.1", "storage_port":5432, "storage_username":"postgres",
"storage_pw":"ekPostgreSQL01", "database":"prasa_dt_db",
"dataTable":"dtd_registry","log_choice":false},
            "DSD": {"label":"DSD", "ip":"127.0.0.1", "port":7003,
"storage_ip":"127.0.0.1", "storage_port":5432, "storage_username":"postgres",
"storage_pw":"ekPostgreSQL01", "database":"prasa_dt_db", "dataTable":"dsd_registry",
"log_choice":false},
            "SD": {"label":"SD", "ip":"127.0.0.1", "port":7004,
"storage_ip":"127.0.0.1", "storage_port":5432, "storage_username":"postgres",
"storage_pw":"ekPostgreSQL01", "database":"prasa_dt_db", "dataTable":"s_registry",
"log_choice":false}
        }
    },


"DS_wrappers":{
        "API_MB3JK2_Layout_wrapper":{
            "label":"API_MB3JK2_Layout_wrapper",
            "ip":"127.0.0.1",
            "port":7005,
            "gateway_url":"http://127.0.0.1:7001",
            "wrapper_type":"API",
            "dateFirstElement":"year",
            "dateDelimiter":"-",
            "refresh_T":300,
            "api_source_address":"http://127.0.0.1:8001",
            "log_choice":false
        },
        "SQL_MB3JK2_IMDATA_wrapper":{
            "label":"SQL_MB3JK2_IMDATA_wrapper",
            "ip":"127.0.0.1",
            "port":7006,
            "gateway_url":"http://127.0.0.1:7001",
            "wrapper_type":"SQL",
            "dateFirstElement":"year",
            "dateDelimiter":"-",
            "startDate":"2019-01-01",
            "refresh_T":100,
            "src_ip":"127.0.0.1",
            "src_port":5432,
            "src_username":"postgres",
            "src_pw":"ekPostgreSQL01",
            "source_db":"data_db",
            "source_table":"MB3JK2_IMDATA",
```

```
            "log_choice":false
        }
    },

    "UI_services":{
        "ExceedCountComparator":{
            "label":"ExceedCountComparator",
            "ip":"127.0.0.1",
            "port":7010,
            "gateway_url":"http://127.0.0.1:7001",
            "serviceParams":{
                    "No_Entities2Compare":2,
                    "Parameters":["All", "Track Guage", "Rail Crown Wear",
"Superelevation", "Stagger", "Vertical Force Total"],
                    "threshold":"intInput"
            },
            "log_choice":false
        }
    },

    "DTs": {
        "DT_Ballast_CPT-CLM": {
            "label":"DT_Ballast_CPT-CLM",
            "infra_domain":"PW",
            "ip":"127.0.0.1",
            "port":7012,
            "gateway_url":"http://127.0.0.1:7001",
            "storage_ip":"127.0.0.1",
            "storage_port":5432,
            "storage_username":"postgres",
            "storage_pw":"ekPostgreSQL01",
            "database":"prasa_dt_db",
            "data_storage":{
                "condition_data":{
                    "columns":["datetime", "location", "params", "param_vals",
"exceedance_percentages"],
                    "uniqueCol":"all",
                    "preStoreServices":["ExcPercCalc"],
                    "sources":{
                        "my_wrappers":{
                            "SQL_MB3JK2_Ballast_wrapper":{
                                "datetime":[{"Values_SourceField":"DATE",
"filterParam":"all"}],
                                "location":[{"Values_SourceField":"LOCATION",
"filterParam":"FloatBetween",
"qualifiers":{"col_to_filter_in":"LOCATION","startVal":0, "endVal":17.004}}],
                                "param_vals":[{"Values_SourceField":"SUP",
"filterParam":"all", "store_fieldname":{"col_to_store_in":"params",
"store_as":"Superelevation"}}]
                            }
                        },
                        "child_DTs":[]
                    }
                },
                "coord_data":{
                    "columns":["location", "latitude", "longitude"],
                    "uniqueCol":"location",
                    "preStoreServices":[],
                    "sources":{
                        "my_wrappers":{
                            "API_MB3JK2_Layout_wrapper":{
                                "location":[{"Values_SourceField":"KM_FROM",
"filterParam":"FloatBetween", "qualifiers":{"col_to_filter_in":"KM_FROM",
"startVal":0, "endVal":17.004}}],
```

```
                                "latitude":[{"Values_SourceField":"LATITUDE",
"filterParam":"all"}],
                                "longitude":[{"Values_SourceField":"LONGITUDE",
"filterParam":"all"}]
                            }
                        },
                        "child_DTs":[]
                    }
                }
            },
            "asset_operational_state":true,
            "startDate":"2019-01-01",
            "child_DTs":[],
            "DT_services_to_UI":{
                "Asset_operational_status":"None",
                "ExceedanceLocs_over_threshold":{
                    "Parameters":["All", "Superelevation"],
                    "threshold":"intInput"
                },
                "ExceedanceGraph":{
                    "Parameters":["All", "Superelevation"],
                    "startDate":"dateInput",
                    "endDate":"dateInput",
                    "threshold":"intInput"
                }
            },
            "DT_service_specs_forBackend":{
                "Asset_operational_status":"None",
                "ExceedanceLocs_over_threshold":{
                    "params_source_table":"condition_data",
                    "params_colNames":{"exceedCol":"exceedance_percentages",
"locationCol":"location", "dateCol":"datetime", "paramCol":"params"},
                    "coords_source_table":"coord_data",
                    "coords_colNames":{"locationCol":"location",
"latCol":"latitude", "longCol":"longitude"}
                },
                "ExceedanceGraph":{
                    "params_source_table":"condition_data",
                    "params_colNames":{"exceedCol":"exceedance_percentages",
"locationCol":"location", "dateCol":"datetime", "paramCol":"params"},
                    "coords_source_table":"coord_data",
                    "coords_colNames":{"locationCol":"location",
"latCol":"latitude", "longCol":"longitude"}
                }
            },
            "log_choice":false
        },

        "DT_Catenary_SRX-CLM_M00456": {
            "label":"DT_Catenary_SRX-CLM_M00456",
            "infra_domain":"E",
            "ip":"127.0.0.1",
            "port":7020,
            "gateway_url":"http://127.0.0.1:7001",
            "storage_ip":"127.0.0.1",
            "storage_port":5432,
            "storage_username":"postgres",
            "storage_pw":"ekPostgreSQL01",
            "database":"prasa_dt_db",
            "data_storage":{
                "condition_data":{
                    "columns":["datetime", "location", "params", "param_vals",
"exceedance_percentages"],
                    "uniqueCol":"all",
```

```
                    "preStoreServices":["ExcPercCalc"],
                    "sources":{
                        "my_wrappers":{
                            "MB3JK2_TGIVCatenaryExceptions_wrapper":{
                                "datetime":[{"Values_SourceField":"DATE",
"filterParam":"all"}],
                                "location":[{"Values_SourceField":"KM_FROM",
"filterParam":"FloatBetween", "qualifiers":{"col_to_filter_in":"KM_FROM",
"startVal":3.74, "endVal":10.16}}],
                                "param_vals":[{"Values_SourceField":"MAX",
"filterParam":"TextEquals", "qualifiers":{"col_to_filter_in":"PARAMETER",
"strings_to_filter_for":["Stagger", "Vertical Force Total"],
"columns_to_store_qualifiers":"params", "store_qualifiers_as":["Stagger", "Vertical
Force Total"]}}]
                            }
                        },
                        "child_DTs":[]
                    }
                },
                "coord_data":{
                    "columns":["location", "latitude", "longitude"],
                    "uniqueCol":"location",
                    "preStoreServices":[],
                    "sources":{
                        "my_wrappers":{
                            "MB3JK2_TGIVCatenaryExceptions_wrapper":{
                                "location":[{"Values_SourceField":"KM_FROM",
"filterParam":"FloatBetween", "qualifiers":{"col_to_filter_in":"KM_FROM",
"startVal":3.74, "endVal":10.16}}],
                                "latitude":[{"Values_SourceField":"LATITUDE",
"filterParam":"all"}],
                                "longitude":[{"Values_SourceField":"LONGITUDE",
"filterParam":"all"}]
                            }
                        },
                        "child_DTs":[]
                    }
                }
            },
            "startDate":"2019-01-01",
            "asset_operational_state":true,
            "child_DTs":[],
            "DT_services_to_UI":{
                "Asset_operational_status":"None",
                "ExceedanceLocs_over_threshold":{
                    "Parameters":["All", "Stagger", "Vertical Force Total"],
                    "threshold":"intInput"
                },
                "ExceedanceGraph":{
                    "Parameters":["All", "Stagger", "Vertical Force Total"],
                    "startDate":"dateInput",
                    "endDate":"dateInput",
                    "threshold":"intInput"
                }
            },
            "DT_service_specs_forBackend":{
                "Asset_operational_status":"None",
                "ExceedanceLocs_over_threshold":{
                    "params_source_table":"condition_data",
                    "params_colNames":{"exceedCol":"exceedance_percentages",
"locationCol":"location", "dateCol":"datetime", "paramCol":"params"},
                    "coords_source_table":"coord_data",
                    "coords_colNames":{"locationCol":"location",
"latCol":"latitude", "longCol":"longitude"}
```

```
                },
                "ExceedanceGraph":{
                    "params_source_table":"condition_data",
                    "params_colNames":{"exceedCol":"exceedance_percentages",
"locationCol":"location", "dateCol":"datetime", "paramCol":"params"},
                    "coords_source_table":"coord_data",
                    "coords_colNames":{"locationCol":"location",
"latCol":"latitude", "longCol":"longitude"}
                }
            },
            "log_choice":false
        }

    }
```