

Article

A Microservices-Based Approach to Designing an Intelligent Railway Control System Architecture

Ivaylo Atanasov ¹, Vasil Vatakov ² and Evelina Pencheva ^{2,*} ¹ Faculty of Telecommunications, Technical University of Sofia, 1756 Sofia, Bulgaria; iia@tu-sofia.bg² Faculty of Telecommunications and Electrical Equipment in Transport, Todor Kableshev University of Transport, 1574 Sofia, Bulgaria; vatakov84@abv.bg

* Correspondence: evelina.nik.pencheva@gmail.com

Abstract: The symmetry between customer expectations and operator goals, on one hand, and the digital transition of the railways, on the other hand, is one of the main factors affecting green transport sustainability. The European Train Control System (ETCS) was created to improve interoperability between different railway signaling systems and increase safety and security. While there are a lot of ETCS Level 2 deployments all over the world, the specifications of ETCS Level 3 are under development. ETCS Level 3 is expected to have a significant impact on automatic train operation, protection, and supervision. In this paper, we present an innovative control system architecture that allows the incorporation of artificial intelligence (AI)/machine learning (ML) applications. The architecture features control function virtualization and programmability. The concept of an intelligent railway controller (IRC) is introduced as being a piece of cloud software responsible for the control and optimization of railway operations. A microservices-based approach to designing the IRC's functionality is presented. The approach was formally verified, and some of its performance metrics were identified.

Keywords: railways; control systems; automation; microservices; discrete event systems



Citation: Atanasov, I.; Vatakov, V.; Pencheva, E. A Microservices-Based Approach to Designing an Intelligent Railway Control System Architecture. *Symmetry* **2023**, *15*, 1566. <https://doi.org/10.3390/sym15081566>

Academic Editors: Crescenzo Pepe and Silvia Maria Zanolini

Received: 10 July 2023

Revised: 5 August 2023

Accepted: 8 August 2023

Published: 11 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

One of the most important aspects of transport development, which coincides with the global challenges, is sustainability. Among the different transport modes that make a mobile society sustainable, rails travel represents the most environmentally oriented area due to their small carbon footprint. The sustainable development of railway transport depends on the possibilities of symmetry between the requirements for highly reliable, safe, and secure services, as well as efficient and productive operation, on one hand, and digitalization, which drives new technologies in the rail industry, on the other hand.

The European Railway Traffic Management System (ERTMS) is a key enabler of the digitalization and sustainable transition of railway transport. ERTMS is a European standard designed to achieve interoperability throughout Europe and provide higher performance, increase efficiency, and improve track utilization and customer experience [1,2]. ERTMS has two components, namely the European Train Control Systems (ETCS), which comprise the core signaling and train control systems, and GSM-R, which will be inherited by the Future Railway Mobile Communication System to provide stable, secure, and reliable connections.

ERTMS/ETCS Level 3 is a train control system wherein movement authorities are generated at the track side and transmitted to the train via radio communication. This model enables continuous supervision and control of train speeds through communication with the trackside ERTMS subsystem. This process makes it possible for trains to run in moving blocks closer together while maintaining safety requirements and, thus, increasing the track capacity [3]. ETCS Level 3 has the potential to allow considerable infrastructure saving and address capacity constraints. ETCS Level 3 is still under development, and

multiple issues have to be addressed before it can be operationally implemented. Highly reliable radio communications and train virtual coupling are two problems, the solutions to which will enable capacity increases and open the door to further automation.

The application of artificial intelligence (AI) in railways can foster the deployment of ERTMS in several areas, including predictive trackside maintenance, traffic management, energy efficiency, etc. [4]. The integration of AI and machine learning (ML) with Internet of Things (IoT) sensors in engines, brakes, wheelsets, and coaches has the potential to improve safety and reliability, and deploying sensors across trackside systems can empower proactive maintenance [5,6].

This paper presents an innovative intelligent architecture of the ETCS system, which features programmability, virtualization, and automation. The ETCS system consists of heterogeneous distributed components installed onboard and trackside within several control centers. The proposed ETCS control system architecture applies a disaggregated approach to control railway assets and software-based functions and enables intelligent control of smart railway operations. The main concept is the intelligent railway controller (IRC), which is responsible for controlling and optimizing the railway's operation. Incorporating AI/ML in IRC overcomes the issues associated with complex railway management and facilitates the delivery of high bandwidth, high quality, and low latency services.

The initial version of this work was published in [7]. The additional contributions presented here include the following:

- The architecture was refined to cover more functional details. In [7], the idea of IRC was presented, and key functions and interfaces were identified, with the focus being on the interaction between time-tolerant and time-sensitive functionalities. In this work, the IRC functionality was elaborated, stressing the functions that expose the required services to AI/ML applications and the AI/ML model workflow function.
- To enable open and interoperable interfaces, railway control system virtualization, and big data intelligence, a microservice-based approach to designing the IRC functionality is presented. In the proposed intelligent architecture, logical functions of time-tolerant and time-sensitive control and optimization, as well as the AI/ML workflow, including model training and updating, are presented as separate microservices instead of as a monolithic design. The well-known benefits of the service-based approach include modularity, extensibility, discoverability, composability, reusability, and loose coupling.
- In [7], a microservice for policy management and enrichment information was designed. In this work, microservices for the management of IRC's applications and ML model management were synthesized. The inherent IRC framework functionality, including authentication and authorization functions, service registration and discovery functions, AI/ML workflow functions, and AI/ML monitoring functions, is made up of modular, reusable, and loosely coupled service bricks.
- Modeling of discrete event systems, formal methods, and symmetry properties was used to prove the approach's feasibility.

This paper is structured as follows. Section 2 provides a brief review of the related works. Section 3 describes the IRC idea. Section 4 presents RESTful services for application management and services related to ML model management and performance monitoring. In Section 5, the feasibility of the idea is illustrated via the modeling of the ML model lifecycle. The estimation of the IRC key performance indicators in terms of latency is provided in Section 6. Section 7 presents some security considerations related to the proposed intelligent railway control system architecture. The concluding section discusses the benefits and limitations of the proposed approach and depicts some future research directions.

2. Related Works

The challenges facing the development of the ERTMS, including its implementation, safety, communication interoperability, human factors, and the diversity of formal meth-

ods, languages, and tools for modeling, verifying, and validating ERTMS products, were discussed in [8].

Signaling systems play an essential role in the control, supervision, and protection of safe train movements, and their availability influences the railway system's performance. The railway networks have a reserve for lower maintenance costs, more availability, and capacity if non-centralized signaling systems are considered. Bearing in mind that the decentralized solutions used for railway signaling systems increase their complexity and inherent safety requirements, it becomes evident that safety validation, which is carried out using a system of methods, is necessary. The approach that is widely adopted by the industry is scenario-based testing, though its sufficiency to assure the necessary safety level of the complex signaling systems is in question. An alternative means of verification, which is both rigorous and already used in the railway domain, is formal verification. However, despite the successful applications of formal methods for decentralized railway signaling, the steps taken in this regard have been limited.

A formal model that validates the principles of ETCS Level 3 was presented in [9]. The impact of the capacity of different signaling systems was investigated in [10], where the comparative analysis showed that the implementation of hybrid ETCS Level 3 solutions can improve the capacity of high-density commuter lines. In [11], the authors proposed a methodology that could be used for formal modeling, verification, and performance evaluation of moving block systems. In [12], a modular and extensible architecture for testing a moving block signaling system was presented, wherein trains received instructions to move to a specific position on the track, in contrast to the fixed block signaling method. In [13], the authors presented an analysis of the railway's capacity using high-performance ERTMS signaling systems, considering the effects of route congestion conflicts at the railway stations and delay propagation. The effects of an ERTMS speed profile filtering on the train driver's braking behavior, running time, and workload were studied in [14]. In [15–17], the authors presented approaches that enable the formal modeling and verification of a moving block system in ERTMS Level 3, which preserves the safety properties. The experience gained from the above-mentioned studies enables the identification of future research goals to improve the formal specification and verification of real-time systems, as well as the recognition of some limitations concerning the usage of formal methods and tools in the railway industry. A formal method for stepwise development and model checking of state transition systems that represent the behavior of interlocking system models was presented in [18]. A control scheme for distributed multiple high-speed train control, which was based on an event-trigger mechanism, was presented in [19]. In [20], a restructuring scheme of railway signaling systems that may be used to improve the process of engineering, construction, commissioning, and operational safety was described. In [21], the authors analyzed the principles of railway signaling system design and applied a comprehensive approach that considers railway stock parameters and infrastructure facilities. In [22], the author proposed a multi-agent technique to optimize the scheduling of the virtual coupling of trains. In [23], an IoT device was proposed as part of a signaling system, which may be used to monitor and log data related to train movements. The problem with virtual coupling trains concerning capacity performances and potential gains over traditional signaling systems was addressed in [24]. The results of a comparative analysis showed that the biggest capacity improvements of virtual coupling relate to scenarios in which the trains use different routes. A model for the safety evaluation of railway traffic under particular conditions of uncertainty was proposed in [25]. In [26], a stochastic analysis of the safety of train movement during an earthquake was performed. An edge-computing-based platform for testing signaling systems on site was described in [27]. A virtual reality environment that assists in installing, updating, and maintaining railway signaling systems was presented in [28].

AI has the potential to play an important role in all areas of railway transport, including safety, security, autonomous control and driving, sustainability, transport planning, and passenger mobility. AI/ML applications may be used for both real-time control and

non-real-time control. AI/ML applications may be used for automatic train protection (continuous train control to keep the speed restrictions), automatic train operation (speed regulation, station stopping, and train and platform door control safety), and automatic train supervision (supervision of train status, automatic routing selection, automatic schedule creation, and automatic system status monitoring) [29–31]. A method for AI-based automated train operation was described in [32]. The use of AI/ML could revolutionize predictive maintenance in railway transport by detecting equipment issues before they become critical [33,34]. An integrated method for the predictive maintenance of railway infrastructure, which is based on deep reinforcement learning and digital twins, was proposed in [35]. The adoption of AI is also well-suited to crowd control, customer service, delay prediction, freight and infrastructure monitoring, etc. [36,37]. A method for the intrusion detection of railway events in distributed vibration sensing, which was based on deep learning, was presented in [38]. An algorithm for railway traffic planning that may improve the system performance was proposed in [39]. An optimization method that used rescheduling strategies for freight railway operations and considered train delay times and priorities was proposed in [40]. The application of AI in scheduling high-speed train operations was illustrated in [41], where the authors studied passenger flow characteristics, train load rates, and train service quality. In [42], a project that studied the methods and models involved in the safe use of AI/ML in train movements, which is called safetrain, was presented in order to improve the safety and reliability of train operations. AI-based methods for reliable railway engineering that consider robustness and transparency have been investigated. The application of the concept of digital twins in railways was investigated in [43], where the authors proposed a workflow of digital twin design that considered specific requirements that lead to high reliability and safety.

Based on a comprehensive literature review, the authors of [44] concluded that future research into the applications of AI in railway operations must focus on the optimization of AI applications in railways, decision making in conditions of uncertainty, and dealing with cybersecurity challenges.

Table 1 summarizes some of the main works related to the area of control systems and intelligent control and operation in railways.

Table 1. Summary of some main works in the area of control systems and intelligent control and operation in railways.

Research Area	Research Subarea	Related Papers	Contributions
Signaling systems in railways	Formal modeling and verification	[9–18]	Models and methods for formal testing of railway control systems
	Architecture and control schemes	[19–28]	New control schemes for automatic train operation and automatic train protection
AI/ML in railways	Train control	[29–32]	AI/ML models and methods for autonomous control and driving
	Predictive maintenance	[33–35]	AI/ML models and methods for trackside maintenance
	Operation optimization	[36–43]	AI/ML models and methods for transport planning, passenger mobility, safety, and security

The literature review showed that there are some issues and challenges that must be considered in the context of embedding AI/ML applications in railway operations to maximize their potential, including the following topics:

- Interoperability is essential for performing analyses and real-time data exchange between heterogeneous systems in the context of AI/ML railway applications. The heterogeneity of management tools and devices may result in non-integrated data.

- Embedding sensors that make trains more sensible and enable predictive trackside maintenance requires ultra-high-quality connectivity with low latency. Any delays or connectivity disruptions between the managed railway assets and the control system may result in incorrect operation and undesired consequences.
- The lack of standards and frameworks for deploying AI/ML in train control systems, track maintenance, and passenger traffic flow control is an open issue that must be considered. Well-defined frameworks must consider data gathering and data analytics techniques, as well as other AI/ML enablers. Standardized procedures enable seamless and well-defined information flows between phases of ML model development and implementation. The development of such standardized frameworks and architecture is essential to ensure interoperability, security, and consistency in implementing AI/ML applications in the railway sector.
- Data privacy and security are critical to safety-critical railway operations. Any intrusion into data exchange could cause damage to railway assets and human casualties. The technologies that enable the application of AI/ML in railway operations must conform to existing security policies.
- The possibility of integration and execution of multiple ML models at the same time is referred to as scalability. In this context, the capability of instantiating and running multiple virtual machines is an important feature of future virtualized control systems.

Research has yet to determine a practical way to link AI/ML techniques while adhering to the requirements and approval processes that exist in the railway domain. Considering the different requirements of AI/ML applications, we propose a disaggregated approach to deploying AI/ML applications in the ETCS Level 3 architecture built entirely on cloud-native principles. In the proposed architecture, the control functionality is disaggregated into time-tolerant and time-sensitive functions, aiming to determine multivendor interoperability, agility, and programmability. The proposed intelligent architecture enables the onboarding of third-party applications to automate and optimize railway operations at scale.

3. The Concept of an Intelligent Railway Controller

The proposed control system architecture defines the railway management automation and orchestration (RMAO) platform that is responsible for the orchestration, operation, management, and automation of managed railway elements, such as trains and trackside equipment. The RMAO hosts the trackside functions of ETCS and automatic train service (ATS), also known as the traffic management system (TMS), as defined in the ERMTS/ETCS architecture. The functions related to the security and safety of all trains and monitoring of trackside equipment are the responsibility of the intelligent railway controller (IRC) and reside in the RMAO layer. The railway edge cloud is a cloud computing platform that provides an environment in which to run virtualized managed functions (IRC, trains, and trackside equipment).

The concept of IRC is introduced to enable the exposure of data and analytics to facilitate automation and improved resilience of railways. The programmability of IRC allows the onboarding of third-party applications to implement different automation and management use cases. The proposed innovative intelligent architecture defines two kinds of IRC: one type that operates in non-real-time in more than 1 s, which is named time-tolerant IRC (TT-IRC), and another type that works in a control loop from 10 ms up to 1 s, which is named time-sensitive IRC (TS-IRC). The TT-IRC is a part of the RMAO and provides functionality that leverages data-driven approaches and analytics to improve railway operations. It controls the railway elements through the TS-IRC via policy guidance and manages the ML model workflow. The R1 interface between TT-IRC and TS-IRC is used for policy management and provisioning of enrichment information. The TT-IRC runs applications (ttApp) that provide value-added services for the inspection of railway lines, damage detection, predictive maintenance, and passenger flow analysis. The TS-IRC hosts applications (tsApp) used in driver assistance systems, such as driving and braking

control, collision protection systems, and the enforcement of TS-IRC policies. More details about the R1 interface between TT-IRC and TS-IRC can be found in [7]. In this paper, we focused on the functionality of the TT-IRC, which, as a logical entity, is responsible for the support of applications such as ttApp service exposure and ttApp conflict mitigation for the AI/ML model workflow, the AI/ML model's monitoring functions, and R1 functions.

The TT-IRC can access external data (enrichment information) that can be used for train control and track monitoring. It uses ttApps to analyze different information and generate policies, such as policies for the control and optimization of train movements, generation of information, performance of data analytics, AI/ML model monitoring, and AI/ML workflow support. The TT-IRC exposes services, such as data sharing and access to data for ttApp applications, via an internal interface, e.g., to perform ttApp management functions (mitigation of ttApps conflicts) and service exposure functions (service registration and discovery, authentication, authorization, etc.).

Figure 1 shows the overall view of the service-based TT-IRC architecture.

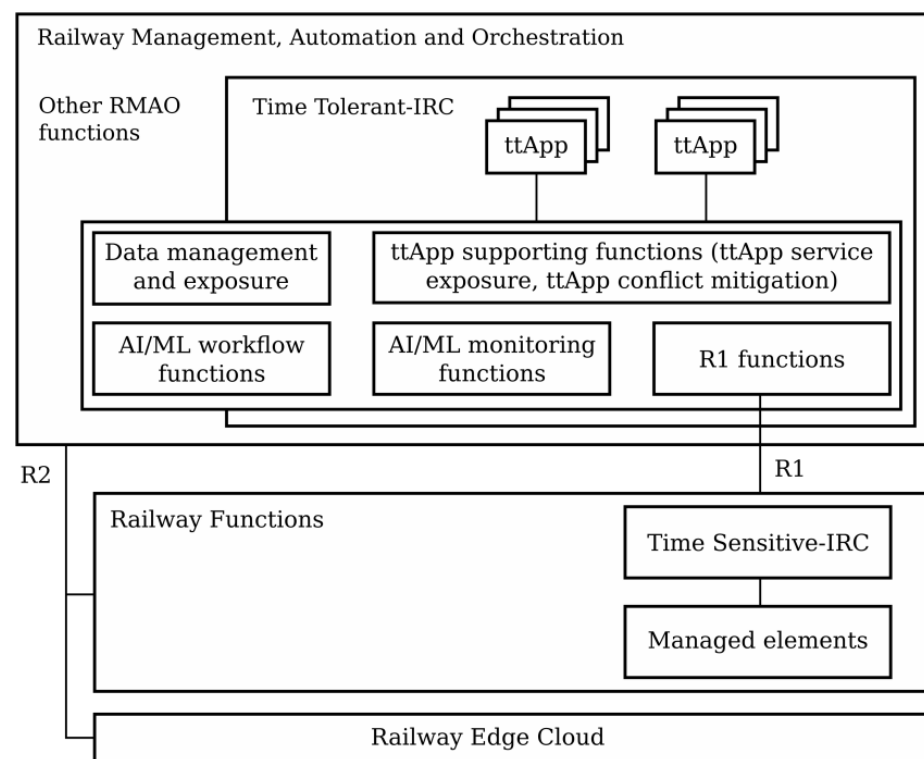


Figure 1. Overall view of the service-based TT-IRC architecture.

The design of the TT-IRC may follow the principles of the microservice architecture, whereby the TT-IRC functions are designed as RESTful services. REST stands for representational state transfer, which is an architectural style used in distributed systems. The main concept in REST is the resource, which represents any physical or logical entity. The resource is uniquely identified based on its uniform resource identifier (URI), and it is manipulated using HTTP methods: GET is used to retrieve information about the resource, POST is used to create a new resource, PUT is used to update the resource information, and DELETE is used for resource removal.

4. An Approach to Designing IRC Services

4.1. Services for ttApp Management

The TT-IRC exposes the services' capabilities to ttApps. ttApps are modular applications that leverage the exposed functionality to provide value-added services. Examples of exposed capabilities include the following examples:

- ttApp package management;
- ttApp instance life cycle management;
- Automatic Train Protection;
- Automatic Train Operation;
- Automatic Train Supervision;
- Track monitoring.

ttApps may be provided by the railway operator or third parties.

The TT-IRC framework exposes infrastructure capabilities, such as authentication and the discovery of exposed capabilities, that can be implemented as CapabilityMgmt services and for the management of ttApp packages that can be implemented as a ttAppPackageMgmt service. The services can be published in a service directory.

The CapabilityMgmt service provides functions for the following issues:

- Registration of a new capability;
- Capability discovery;
- Notification about the registration of a new capability.

In addition, the CapabilityMgmt service supports integrity management functions, such as load balancing, fault management, and heartbeat, which are beyond the scope of this paper.

Figure 2 shows the structure of the URIs of resources related to the CapabilityMgmt service.

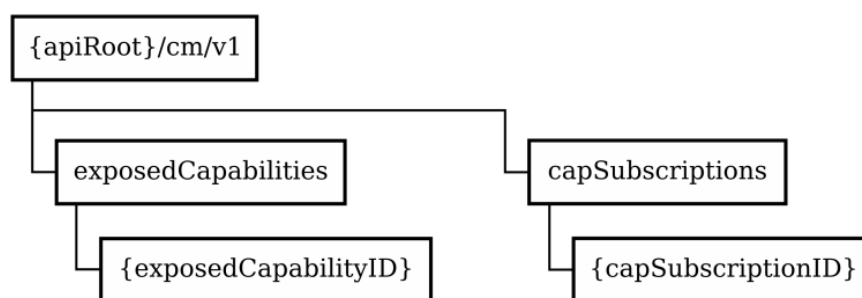


Figure 2. URI structure of the resources related to exposed capabilities.

The exposedCapabilities resource is a container of all capabilities of the TT-IRC exposed by the railway operator. An individual exposed service capability is represented by the {exposedCapabilityID} resource. Applying the HTTP GET method to the exposed Capabilities resource retrieves the list of all exposed capabilities, while an HTTP POST method is used to register a new exposed capability. The registration of a new capability requires authentication. The HTTP GET, PUT, and DELETE methods are applied to the {exposedCapabilityID} resource to retrieve information about, update, or delete an individual exposed capability, respectively. Some resources also represent all active subscriptions for changes in the exposed capabilities (capSubscriptions resource) and an individual subscription ({capSubscriptionID}). A new subscription is created by applying the POST method to the capSubscriptions resource. Information about individual subscriptions can be retrieved using the GET method, updated using the PUT method, or removed using the DELETE method.

The ttApp package contains files related to the ttApp descriptor, which contains the ttApp rules and requirements, a virtual machine image, the manifest file, and other optional files. The ttAppPackageMgmt service enables ttApp lifecycle management, ttApp rules, and requirement management. It also manages the ttApp images. The ttAppPackageMgmt service provides the following functions:

- Registering the ttApp package (making a ttApp package available to the RMAO/TT-IRC);
- ttApp instance life cycle management;
- Querying the ttApp package information (providing the information contained in the package);

- Enabling/disabling a ttApp package (enabling a ttApp package in the RMAO/TT-IRC for further application initiation or disabling a ttApp package);
- Deleting a ttApp package (removing a ttApp package from the RMAO/TT-IRC);
- Fetching a ttApp package (retrieving a ttApp package or selected files in it).

Figure 3 shows the URI structure of resources supported by the ttAppPackageMgmt service.

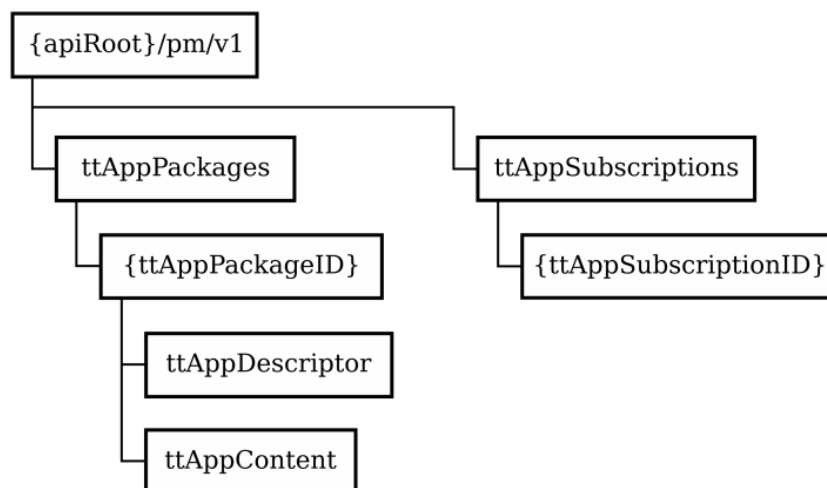


Figure 3. URI structure of resources related to ttApp package management.

The ttAppPackages resource represents all registered ttApp packages. The resource supports the GET method, which provides a list of all registered ttApp packages, and the POST method, which registers a new ttApp package. The {ttAppPackageID} resource represents an individual ttApp package, and the HTTP methods supported by it are GET, PUT, and DELETE, which retrieve, update, and delete information about the ttApp package, respectively. The ttAppDescriptor resource represents the ttApp descriptor of the onboarded ttApp package, and it supports the GET method, which is used to read the ttApp package descriptor. The ttAppContent resource represents the content of the ttApp package and, by applying the GET method, fetches the registered ttApp package content, while applying the PUT method uploads the ttApp package content.

The ttAppSubscriptions resource represents subscriptions for registered ttApp packages. It supports the POST method, which creates a new subscription to notifications related to onboarding/changing ttApp packages. Applying the GET method to the ttAppSubscriptions resource retrieves the list of active subscriptions. The {ttAppSubscriptionID} resource represents an individual subscription and supports GET and DELETE methods, which read and terminate an individual subscription, respectively.

Figure 4 shows the flow of registering a new ttApp package and the discovery of exposed capabilities.

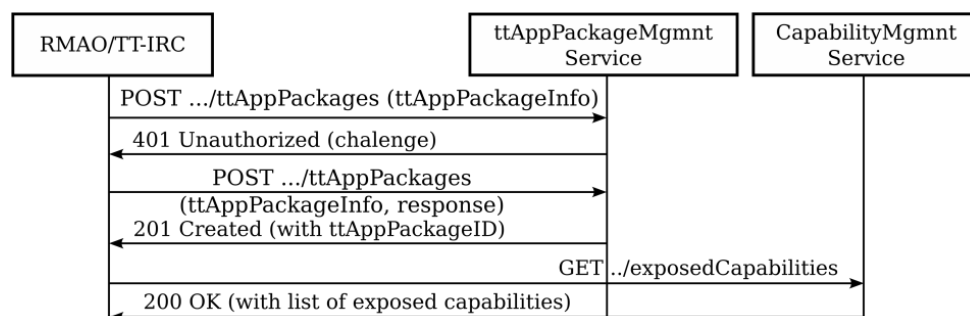


Figure 4. The flow of registration of a new ttApp package and the discovery of exposed capabilities.

When a ttAppPackage has to be registered, a POST method is applied to the ttAppPackages resource. The 401 Unauthorized response of the first POST request contains a

challenge that has to be used for authentication, and the second POST method sends the calculated authentication response. If the authentication is successful, the identifier of the newly onboarded ttAppPackage is returned. The new ttApp package may discover exposed capabilities.

The TT-IRC framework also supports the functionality of ttApp lifecycle management, which can also be implemented as a service (ttAppLCMgmt service).

Figure 5 shows the URIs structure of resources related to ttApp lifecycle management.

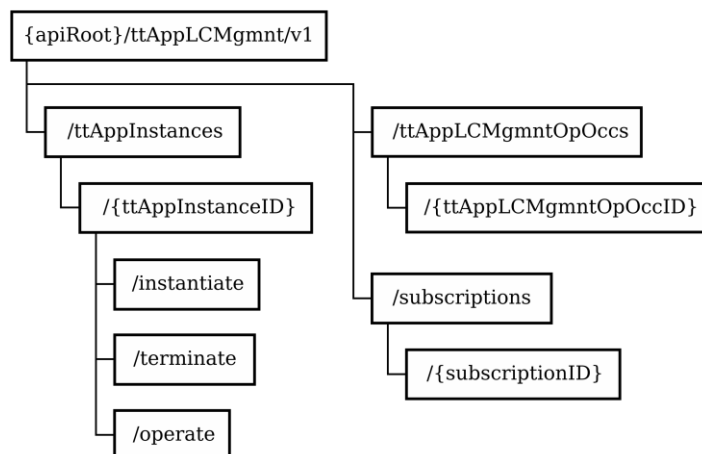


Figure 5. Structure of resource URIs related to ttApp lifecycle management.

The ttAppInstances resource represents all application instances and supports the POST method, which creates a new ttApp instance resource, and GET, which reads the list of ttApp instance resources. The {ttAppInstanceID} resource represents individual ttApp instances and supports the GET and DELETE methods, which read and delete the ttApp instances, respectively. The instantiate resource represents the task of instantiating a ttApp instance, which includes ttApp instance authentication and authorization, initial configuration, and resource assignment. The terminate resource represents the task of ttApp instance termination, and the operate resource represents the task of starting or stopping the ttApp application. These resources support the POST method, which instantiate, terminate, and start/stop the ttApp instance, respectively.

The ttAppLCMgmtOpOccs resource is used for the operation occurrence of the ttApp lifecycle management, and applying the GET method queries multiple individual ttApp lifecycle operation occurrences. An individual ttApp lifecycle management operation occurrence is represented by the {ttAppLCMgmtOpOccID} resource, and it can be read by applying the GET method. There are also resources representing subscriptions, as well as an individual subscription's notifications related to the ttApp instance's lifecycle.

Figure 6 shows the flow of ttApp instance initiation, and Figure 7 shows the flow of ttApp instance termination.

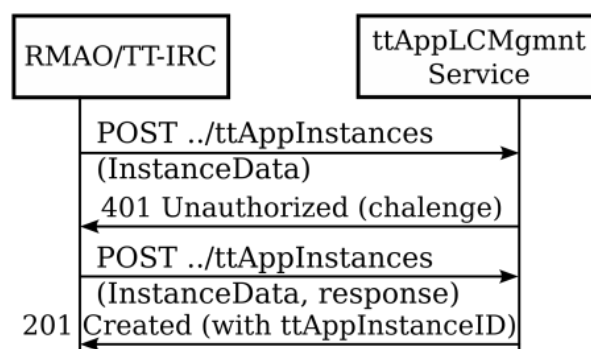


Figure 6. The flow of ttApp instantiation.

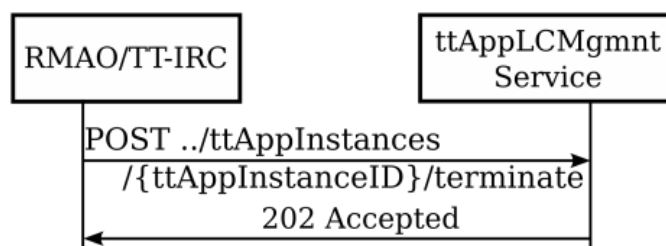


Figure 7. The flow of ttApp instance termination.

4.2. Services Related to ML Model Management

The TT-IRC is also responsible for the process of the ML model workflow, which consists of data processing, model training and refinement, model evaluation, and deployment. Specific use case can be served by applying ML algorithms in a ML model. The lifecycle of the ML model includes deployment, instantiation, and termination.

The TT-IRC can train a ML model using data collected from the managed elements. It also may be an inference host, which hosts the ML model during the model's execution and online training. The TT-IRC needs to provide the ML model designer with the following functions:

- ML model onboarding for training;
- Notifications about published trained ML models;
- Discovery of inference host capabilities;
- Selection of trained and published ML models and their deployments;
- Notification about ML model termination.

The InferenceHostCapability service provides information about the capabilities of the host in which the ML model is executed.

Figure 8 shows the structure of resource URIs related to inference host capabilities.

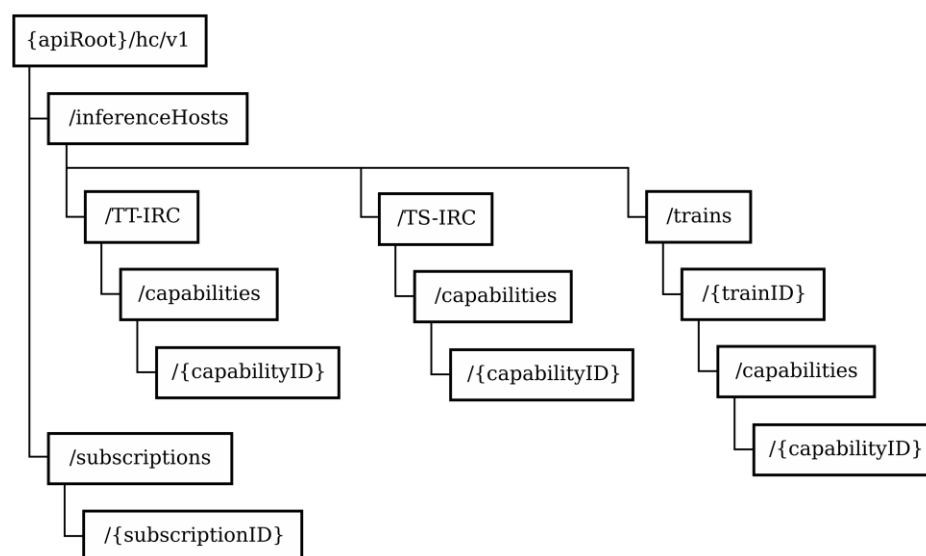


Figure 8. Structure of resource URIs related to inference host capabilities.

The capabilities and properties of the inference host include processing capacity, supported ML model formats and engines, and the requirements of the controlled use case, such as execution time and delay sensitivity, available data sources, and virtualized infrastructure. An inference host may be the TT-IRC or TS-IRC for supervised ML, unsupervised ML, and reinforcement ML, while for federated ML, the inference host may be the train's onboard equipment.

All of the resources support the GET method, which is used to retrieve the inference host capabilities. The subscription resources represent subscriptions to notifications about changes in the inference host's capabilities.

MLModelMgmt service enables the ML model designer to onboard a new ML model for training and select a published trained ML model for deployment. The service sends notifications to the ML model designer about the trained and published ML models and the ML model's termination. The structure of resource URIs related to the ML model management is shown in Figure 9.

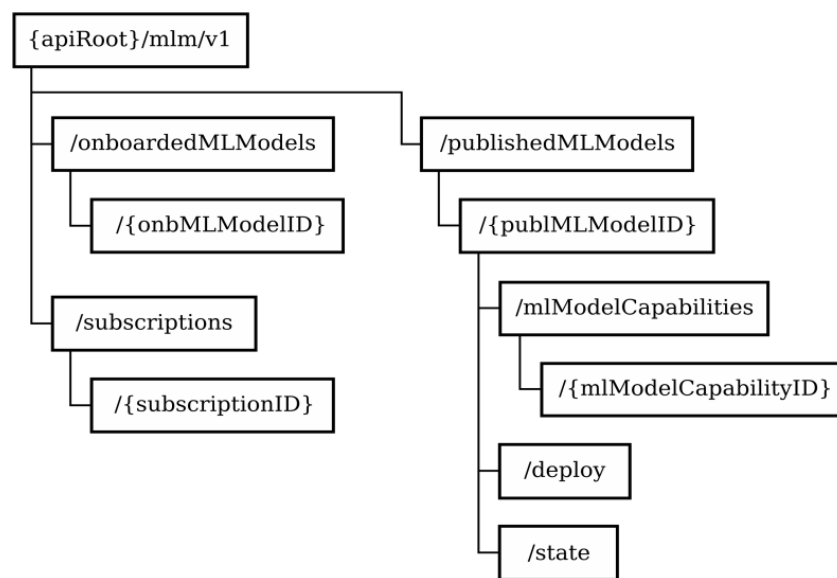


Figure 9. Structure of resource URIs related to the ML model management.

The onboardedMLModels resource represents all onboarded ML models, and applying the GET method to it retrieves the list of all onboarded ML models, while applying the POST method creates a new {onbMLModelID} resource representing an individual onboarded ML model. The resource representing an individual onboarded ML model supports the GET method, which queries information about the ML model, and the DELETE method, which is used to remove the ML model. When the TT-IRC completes the ML model training, it publishes it in the RMAO directory and notifies the ML model designer to select the model for deployment.

The publishedMLModels resource is the catalog for all trained and published ML models. Applying the GET method to the resource returns the list of published ML models. The {publMLModelID} resource represents an individual published ML model. This resource possesses sub-resources that describe the ML model's capabilities and requirements, which are available for reading (GET method). The deploy resource represents the deployment task, and applying the POST method deploys the ML model. The state resource represents the state of the ML models, and applying the GET method on the resource returns one of the following outcomes: initiated, running, or terminated. The subscription resources represent subscriptions for notifications related to ML models.

The model training requires access to model training data collected from managed elements. During ML model execution, model inference data are collected and used to update the ML model's configuration. The TT-IRC functionality for ML model training, initiating, starting, updating, and terminating also may be synthesized as services through access to respective data.

4.3. A Use Case of a ML Model's Lifecycle

Any use case that addresses a specific ML algorithm application during operation (e.g., automatic train control) includes the following steps:

1. The discovery of capabilities of both the ML model and the inference host takes place when a new ML model has to be executed or an existing ML model has to be updated. The considerations that have to be taken into account include the inference host's processing capability, the requirements of the ML model, the support of the virtualized infrastructure, and available data sources. This step is required to check whether the ML model can be executed on the target inference host. The InferenceHostCapability service is used during this step.
2. The ML model training is related to the specific use case for which the ML model is applicable. The ML training host initiates the model training using the ML training data collection. Model training data are collected from the TS-IRC and managed entities. The available enrichment information may be used by the TT-IRC, which has been collected or derived from non-control system data sources or managed entities themselves. Being trained and validated, the model is published into the RMAO/TT-IRC catalog. The MLModelMngmnt service is used during this step to manipulate resources representing the onboarded ML models.
3. The ML designer is notified that the trained model is published, and they need to check whether the trained model can be deployed in the inference host for the given use case, i.e., the ML model requirements are met. This step is the ML model selection step. The MLModelMngmnt service is used during this step in order to notify the model designer.
4. At the deployment and inference step, the ML designer informs the RMAO/TT-IRC to initiate model deployment. Once the model is deployed and activated, online data are used for inference in the ML use case. The MLModelMngmnt service is used during this step for the manipulation of resources representing published ML models.
5. During ML model execution, feedback about the ML model's performance is gathered in the RMAO/TT-IRC. The feedback and reports are required to monitor the model accuracy, running time, and key performance indicators. Based on the ML model's performance evaluation, a notification may be sent that suggests that model retraining is required or another model has to be used. The functionality related to ML model performance monitoring also can be synthesized as a microservice.
6. The preceding steps are related to ML model retraining, updating, and ML model reselection. In some scenarios, the ML model may be terminated, e.g., in the case of severe ML model performance degradation.

5. Formal Verifications of the IRC Design Based on Behavioral Symmetry

Symmetry is very useful in distributed system analysis [45]. It is related to a distributed system's robustness because it identifies behavioral equivalent entities that communicate with each other. As the entities serve the same aims with regard to system operation, their communication style has to be symmetric. In the proposed intelligent IRC functionality, all communications between identified functions must be synchronized, which means that the interacting entities must expose symmetric behavior.

Formal verification is used to prove the approach's feasibility and the inherent behavioral symmetry.

The ML model lifecycle may be considered as a discrete event system, that is, as a dynamic process with discrete states and transitions that are triggered by events. The events that cause leaving one state and the transition into another state are related to receiving/emitting HTTP requests/responses, which manipulate the service resources.

As a part of TT-IRC service design, the models of the discrete event systems that represent the ML model lifecycle from the points of view of the designer and the RMAO/TT-IRC are developed. The models consider the case in which the TT-IRC is chosen as an inference host. Formal methods are used to prove the correctness of the TT-IRC functionality with respect to the defined services.

Figure 10 shows the abstract view of the ML model lifecycle supported by the ML designer.

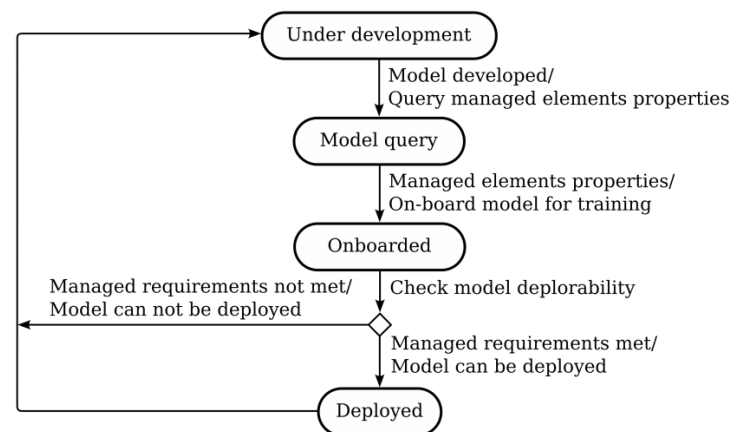


Figure 10. The abstract state machine of the ML model lifecycle supported by the ML designer.

In the UnderDevelopment state, the model is under design and composition. In the ModelQuery state, the model has to be used for specific use cases, and the various capabilities and properties of the ML inference host are discovered. In the Onboarded state, the ML model and the relevant metadata are onboarded into the training host, and the ML model is training. In the Deployed state, the trained and validated ML model is deployed and running.

Figure 11 shows an abstract view of the ML model’s lifecycle, as supported by the RMAO/TT-IRC.

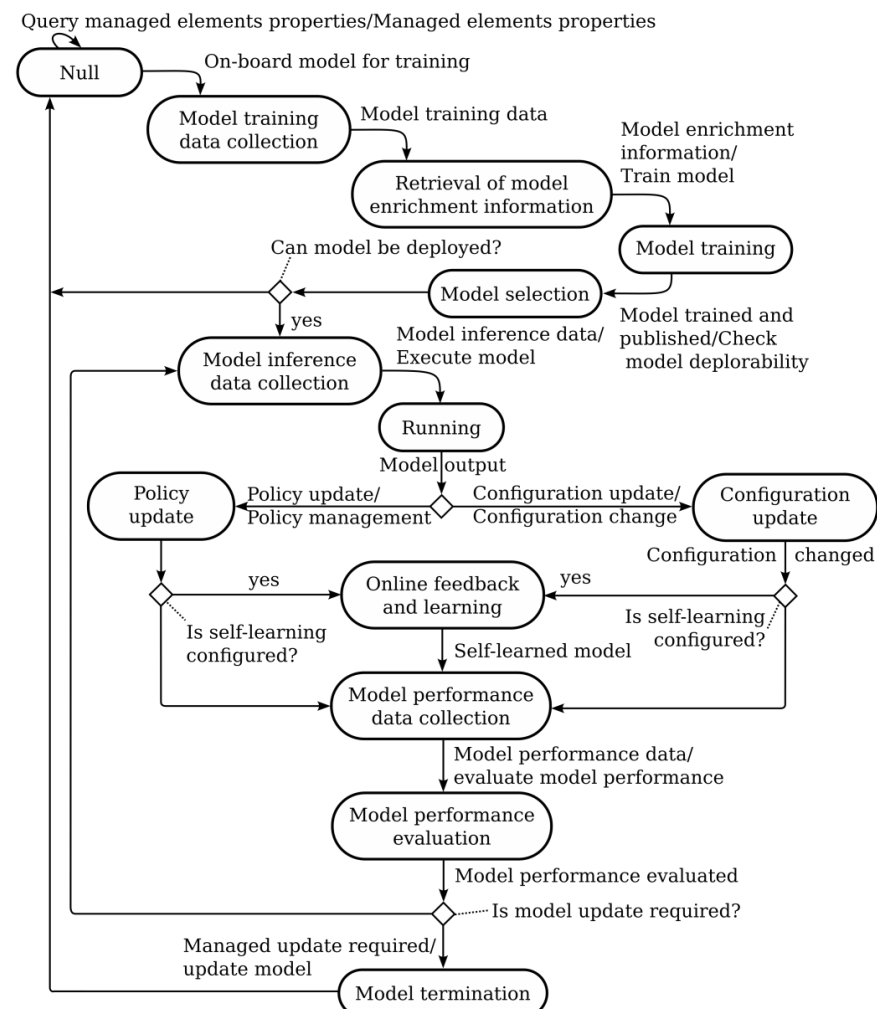


Figure 11. The abstract state machine of the ML model lifecycle from the RMAO/TT-IRC’s point of view.

In the Null state, the capabilities of the ML model and inference host can be discovered. In the ModelTrainingDataCollection state, the model is onboarded for training, and model training data are collected from the managed elements. In the RetrievalOfModelEnrichmentInformation state, additional information about the model is retrieved, e.g., for the trains. In the ModelTraining state, the ML model is undergoing training. In the ModelSelection state, the model is trained, validated, and published into the catalog, and the RMAO waits for the designer's decision regarding whether the model can be deployed. In the ModelInferenceDataCollection state, the model inference data are collected from the managed elements. In the Running state, the ML model is executed. Based on the output, policy guidance may be needed (PolicyUpdate state), or configuration changes may be required (ConfigurationUpdate state). The model may be optionally configured to perform self-learning (OnlineFeedbackAndLearning state). In the ModelPerformanceData state, feedback and reports on the performance of the ML model are collected by the RMAO/TT-IRC to monitor the way in which the ML model works. In the ModelPerformanceEvaluation state, the ML model's performance is evaluated. As a result of ML model performance evaluation, either advice to use another ML model is sent to the designer or a retraining procedure takes place. In the ModelTermination state, there is severe degradation of the ML model performance, the model is terminated, and a backup solution is activated.

Both state machines representing the ML model lifecycle are run as parallel processes, and there must be symmetry in their behavior, that is, the state machines have to expose symmetric behavior. To prove behavioral models' symmetry, the state machines are formally described as labelled transition systems (LTS), and the mathematical tool of bi-simulation is used.

An LTS is a frequently used mathematical formalism that captures the event-triggered transitions between the discrete states of a system. An LTS is a quadruple of a set of states, a set of actions, a set of transitions, and a set of initial states [46]. In the following definitions, short notations given in brackets are used to represent the names of states and transitions.

Definition 1. Let $L^{des} = (S^{des}, A^{des}, T^{des}, s_0^{des})$ be an LTS representing the model of a ML model lifecycle that is supported by the application designer, where:

- $S^{des} = \{UnderDevelopment [s_1^a], ModelQuery [s_2^a], Onboarded [s_3^a], Deployed [s_4^a]\};$
- $A^{des} = \{ModelDeveloped [a_1^a], ManagedFunctionsProperties [a_2^a], ModelRequirementsNotMet [a_3^a], ModelRequirementsMet [a_4^a], UpdateModel [a_5^a];$
- $T^{des} = \{(s_1^a a_1^a s_2^a), (s_2^a a_2^a s_3^a), (s_3^a a_3^a s_1^a), (s_3^a a_4^a s_4^a), (s_4^a a_5^a s_1^a)\};$
- $s_0^{des} = UnderDevelopment.$

Definition 2. Let $L^{irc} = (S^{irc}, A^{irc}, T^{irc}, s_0^{irc})$ be an LTS representing the model of a ML model lifecycle that is supported by the RMAO/TT-IRC, where:

- $S^{irc} = \{Null [s_1^r], ModelTrainingDataCollection [s_2^r], RetrievalOfModelEnrichmentInformation [s_3^r], ModelTraining [s_4^r], ModelSelection [s_5^r], ModelInferenceDataCollection [s_6^r], Running [s_7^r], PolicyUpdate [s_8^r], ConfigurationUpdate [s_9^r], OnlineFeedbackAndLearning [s_{10}^r], ModelPerformanceDataCollection [s_{11}^r], ModelPerformanceEvaluation [s_{12}^r], ModelTermination [s_{13}^r]\};$
- $A^{irc} = \{QueryManagedElementProperties [a_1^r], OnboardModelForTraining [a_2^r], ModelTrainingData [a_3^r], ModelEnrichmentInformation [a_4^r], ModelTrainedAndPublished [a_5^r], ModelCanNotBeDeployed [a_6^r], ModelCanBeDeployed [a_7^r], ModelInferenceData [a_8^r], PolicyUpdate [a_9^r], ConfigurationUpdate [a_{10}^r], SelfLearningConfigured [a_{11}^r], SelfLearningNotConfigured [a_{12}^r], SelfLearnedModel [a_{13}^r], ModelPerformanceData [a_{14}^r], ModelUpdateRequired [a_{15}^r], NoModelUpdateRequired [a_{16}^r], TerminationProcedureCompleted [a_{17}^r]\};$
- $T^{irc} = \{(s_1^r a_1^r s_1^r), (s_1^r a_2^r s_2^r), (s_2^r a_3^r s_3^r), (s_3^r a_4^r s_4^r), (s_4^r a_5^r s_5^r), (s_5^r a_6^r s_1^r), (s_5^r a_7^r s_6^r), (s_6^r a_8^r s_7^r), (s_7^r a_9^r s_8^r), (s_7^r a_{10}^r s_9^r), (s_8^r a_{11}^r s_{10}^r), (s_9^r a_{11}^r s_{10}^r), (s_{10}^r a_{13}^r s_{11}^r), (s_8^r a_{12}^r s_{11}^r), (s_9^r a_{11}^r s_{11}^r), (s_{11}^r a_{14}^r s_{12}^r), (s_{12}^r a_{15}^r s_{13}^r), (s_{12}^r a_{16}^r s_6^r), (s_{13}^r a_{17}^r s_1^r)\};$
- $s_0^{irc} = Null.$

The symmetry in the behavior of both state machines is via by the use of the well-known mathematical tool known as weak bi-simulation [47,48]. Bi-simulation can be equivalently defined as a symmetric relationship between the states of the entities involved.

Proposition 1. L^{des} and L^{irc} have a bi-simulation relationship and expose symmetric behavior.

Proof. In order to prove the bi-simulation between L^{des} and L^{irc} , it is necessary to identify a bi-simulation relationship between their states. Let $R = \{(s_1^a, s_1^r), (s_3^a, s_2^r), (s_4^a, s_6^r)\}$. It will be proven that R is characterized by a weak bi-simulation relationship. The following bijective function between the states in R may be identified:

1. The ML model is designed and composed, the properties of the managed elements are discovered to determine the inference host, and the ML model is onboarded for training: $\forall (s_1^a a_1^a s_2^a) \wedge (s_1^a a_2^a s_2^a) \exists (s_1^r a_1^r s_1^r) \wedge (s_1^r a_2^r s_2^r)$.
2. The model training data are collected from the managed elements, enrichment information is retrieved, and the ML model is trained and published, but the model requirements are not met, and the model cannot be uploaded: $\forall (s_3^a a_3^a s_1^a) \exists (s_2^r a_3^r s_3^r) \wedge (s_3^r a_4^r s_4^r) \wedge (s_4^r a_5^r s_{45}^r) \wedge (s_5^r a_6^r s_1^r)$.
3. The ML model training data are collected from the managed elements, enrichment information is retrieved, the ML model is trained and published, the model requirements are met, and the ML model is uploaded and run. The output of the model execution is policy management. The ML model undertakes self-training online if configured. Model performance data are collected, the ML model performance is evaluated, and the model continues to run: $\forall (s_3^a a_4^a s_4^a) \exists (s_5^r a_7^r s_6^r) \wedge (s_6^r a_8^r s_7^r) \wedge (s_7^r a_9^r s_8^r) \wedge ((s_8^r a_{11}^r s_{10}^r) \wedge (s_{10}^r a_{13}^r s_{11}^r) \vee (s_8^r a_{12}^r s_{11}^r)) \wedge (s_{11}^r a_{14}^r s_{12}^r) \wedge (s_{12}^r a_{16}^r s_6^r)$.
4. The ML model training data are collected from the managed elements, enrichment information is retrieved, the ML model is trained and published, the model requirements are met, and the ML model is uploaded and run. The output of the model execution is configuration management of the managed elements. The ML model undertakes self-training online if configured. Model performance data are collected, the ML model performance is evaluated, and the model continues to run: $\forall (s_3^a a_4^a s_4^a) \exists (s_5^r a_7^r s_6^r) \wedge (s_6^r a_8^r s_7^r) \wedge (s_7^r a_{10}^r s_9^r) \wedge ((s_9^r a_{11}^r s_{10}^r) \wedge (s_{10}^r a_{13}^r s_{11}^r) \vee (s_9^r a_{12}^r s_{11}^r)) \wedge (s_{11}^r a_{14}^r s_{12}^r) \wedge (s_{12}^r a_{16}^r s_6^r)$.
5. The ML model training data are collected from the managed elements, enrichments information is retrieved, the ML model is trained and published, the model requirements are met, and the ML model is uploaded and run. The output of the model execution is policy management. The ML model undertakes self-training online if configured. Model performance data are collected, the ML model performance is evaluated, and the ML model requires an update and is, therefore, terminated: $\forall (s_3^a a_4^a s_4^a) \exists (s_5^r a_7^r s_6^r) \wedge (s_6^r a_8^r s_7^r) \wedge (s_7^r a_9^r s_8^r) \wedge ((s_8^r a_{11}^r s_{10}^r) \wedge (s_{10}^r a_{13}^r s_{11}^r) \vee (s_8^r a_{12}^r s_{11}^r)) \wedge (s_{11}^r a_{14}^r s_{12}^r) \wedge (s_{12}^r a_{15}^r s_{13}^r) \wedge (s_{13}^r a_{17}^r s_1^r)$.
6. The ML model training data are collected from the managed elements, enrichment information is retrieved, the ML model is trained and published, the model requirements are met, and the ML model is uploaded and run. The output of the model execution is configuration management of the managed elements. The ML model undertakes self-training online if configured. Model performance data are collected, the ML model performance is evaluated, and the ML model requires an update and is, therefore, terminated: $\forall (s_3^a a_4^a s_4^a) \exists (s_5^r a_7^r s_6^r) \wedge (s_6^r a_8^r s_7^r) \wedge (s_7^r a_{10}^r s_9^r) \wedge ((s_9^r a_{11}^r s_{10}^r) \wedge (s_{10}^r a_{13}^r s_{11}^r) \vee (s_9^r a_{12}^r s_{11}^r)) \wedge (s_{11}^r a_{14}^r s_{12}^r) \wedge (s_{12}^r a_{15}^r s_{13}^r) \wedge (s_{13}^r a_{17}^r s_1^r)$. \square

The identified bijection function between the states in R proves that R is a weak bi-simulation relationship, and the L^{des} and L^{irc} , therefore, expose symmetry in their behavior when running as parallel processes.

6. Evaluation of Functional Metrics of the Proposed Microservice-Based Approach

Functional metrics of the proposed microservice-based approach, also known as so-called key performance indicators, impact users' perceptions and include parameters such as latency, energy efficiency, throughput, and loss rate. Each of these parameters has to be estimated on a per service basis. Non-functional metrics, such as service lifecycle, service reliability, and service computational load, are related to the service performance and deployment and are functions of the proposed RMAO framework.

Future digitalized railways will rely on the seamless connectivity, high speeds, reliability, and low delays of fifth-generation (5G) mobile networks. As the inference host in the proposed control center architecture can be the TS-IRC or the train's onboard equipment, which has strong latency requirements, an experiment was set up to estimate the latency introduced by the microservices.

In the proposed microservice architecture, the communications are based on HTTP.

The component diagram depicting the experiment is shown in Figure 12.

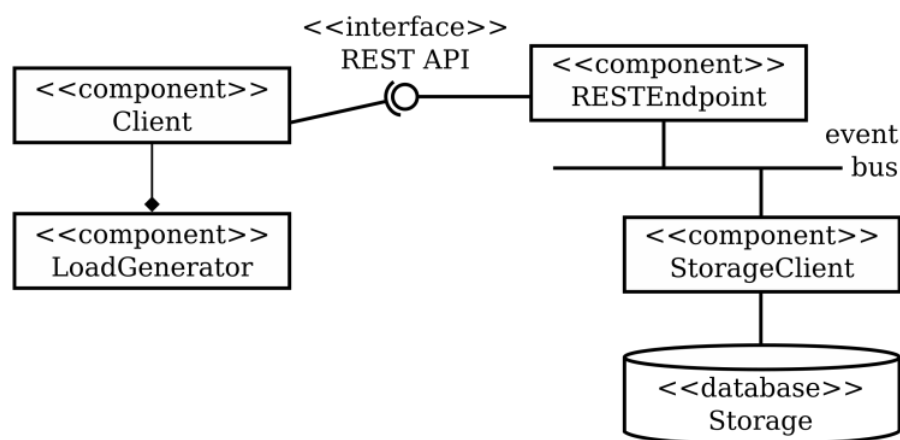


Figure 12. The component diagram depicting the experiment setup.

The experiment is conducted via emulation, which requires components to implement both server and client functionality. The RESTful load consists of POST requests generated via a Java-based HTTP multi-threaded client. Each request contains a JSON payload in order to deliver the domain-specific data, and it is marked by adding an extra header that holds the submission instant in nanoseconds. The server component consists of two Docker instances: one instance enables the REST endpoint and the Cassandra client, and the other function enables the Cassandra server, providing a lightweight virtualized storage service. Containers are deployed onto two nodes with eight cores, and each node is 32 GB in size. The nodes are connected via 1Gb Ethernet, and the serving side containers are bridged, though to separate as much as possible from the adjacent traffic, IPv6 link local addressing is used. On the serving side, there are a couple of docker instances, dedicated to (a) a lightweight virtualized keystore, i.e., the Apache Cassandra service, and (b) the REST endpoint backed by a Cassandra client. At the REST endpoint, the time-marker header is fetched out of the request and copied into the response, and, thus, possessing the response arrival instant and the initial instant that is passed back, the client aggregates the latency-related information.

The offered load consists of 20,000 operations. The time series, as shown in Figures 13 and 14, are formed based on the differences between the response arrival time and the request submission time for each operation in a time window of a thousand consecutive values, where the frame numbers are ninth and nineteenth. The raw latency time series are less expressive when the question is related to estimating the shape and limits of the most frequent latency values, and because of that, using the probability density functions (PDF) might be better and more appropriate. By forming the length of the bins within a

sub-millisecond scale, we can observe most of the mass and its dynamics when comparing different frames. Figures 15 and 16 illustrate the numerical results.

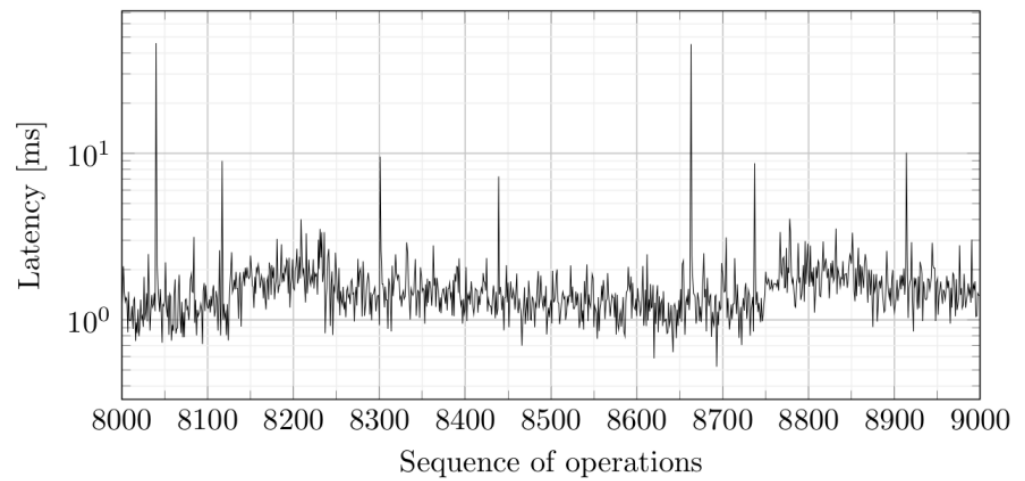


Figure 13. Latency values for the ninth frame of 1000 operations.

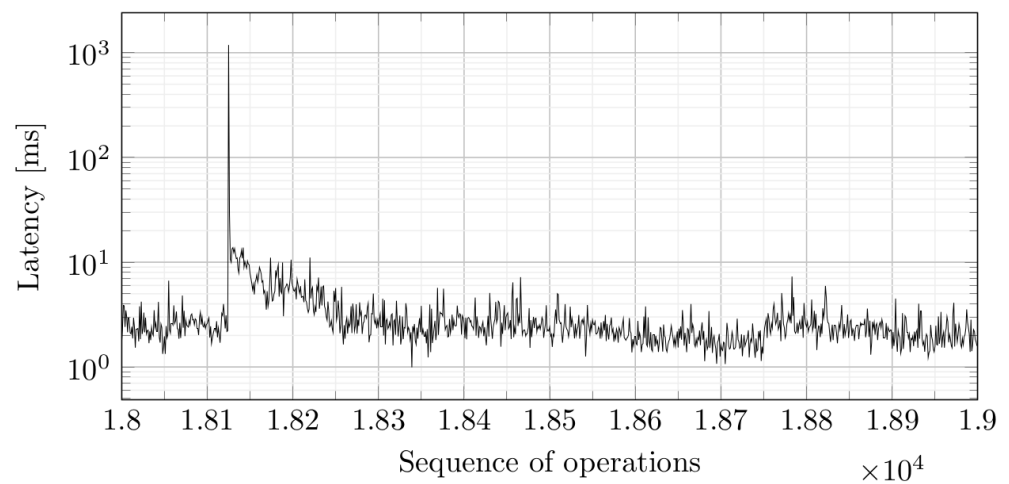


Figure 14. Latency values for the nineteenth frame of 1000 operations.

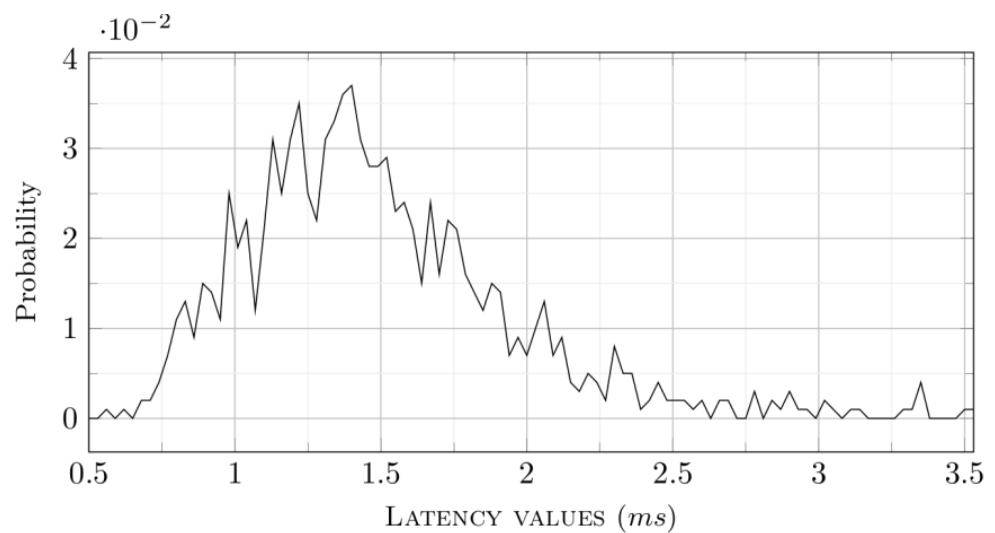


Figure 15. PDF values for the ninth frame of 1000 operations.

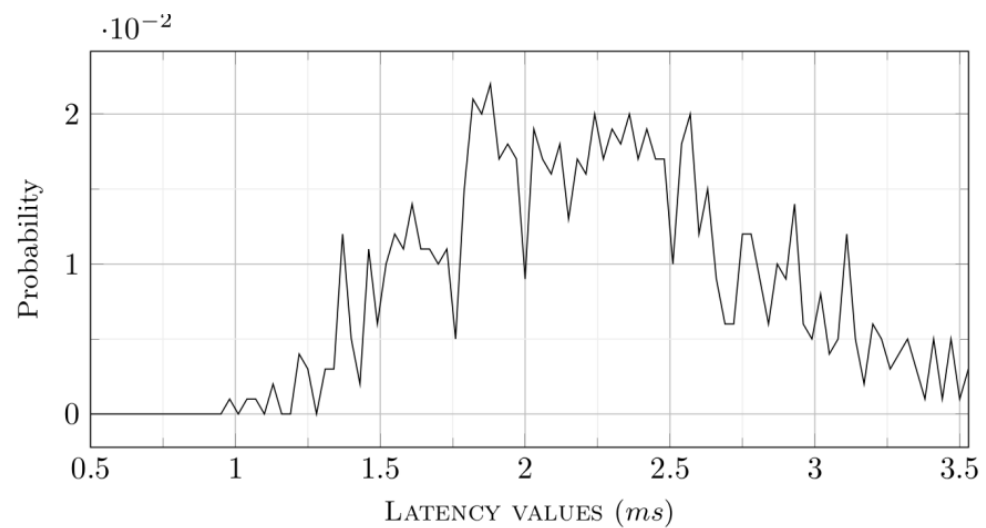


Figure 16. PDF values for the nineteenth frame of 1000 operations.

The results show that the average latency injected via the interface is about two milliseconds, which is acceptable for the design aims. Such latency values in communication between the TT-IRC and the TS-IRC, which are related to submission of AI/ML-based instructions, enable on-time control on the train propulsion and braking systems without the delay in human reaction and the changeability and possibility of misinterpretation that is inherent in manual train operation.

Bearing in mind the eventual increase in average latency values, which are caused by topology changes for further implementations, e.g., load balancing, etc., we expect that the question will remain open to improvements.

7. Security Considerations of the Proposed Intelligent Architecture

The deployment of the proposed open railway control system architecture introduces multiple security considerations. As an open ecosystem, the disaggregated architecture requires a specific focus on security threats at the interfaces between components that may be provided by multiple vendors and the threats related to open-source applications. Additionally, common security considerations related to cloud infrastructure, virtualization, and distributed denial of service attacks have to be taken into account.

The disaggregated architecture implies components that vary in their specific functions or use cases. While the inherent openness fosters interoperability, the compatibility between components and functions from different vendors (e.g., delays in the device updates) is crucial to the control system security. In case of vulnerability, it may be difficult to identify which party is responsible.

The key security objective for the open interfaces is to provide the following safeguards:

- Confidentiality and integrity of data;
- Availability of transport network interface connectivity;
- Authenticity of the functions related to time-sensitive communications.

Confidentiality and the integrity of data can be protected by implementing the security control mechanisms offered by fifth-generation mobile networks over the air interface. The availability of open interfaces requires security control to manage the potential denial of service attacks and unauthorized device access, such as access control to IRC management functions and managed elements. Appropriate cryptographic security mechanisms may be used in the open interfaces in real time. The authenticity may be based on mutual TLS (transport link security), including certificates based on public key infrastructure.

To ensure ttApp application security and mitigate the threats associated with ttApp development, the following practices may be useful:

- Use of stable AI/ML models and data sets;

- Implementation of mutual authentication, which is provided by the IRC framework;
- Protection against malicious snooping, modifying, or injected messages may be achieved via confidentiality and integrity protection;
- Policies have to be defined to mitigate the conflicts in case of a multivendor environment.

AI/ML models may expose networks to unpredictable or malicious behavior when subject to data poisoning attacks (e.g., changes in the input data that can be considered as random noise). The training, deploying, and updating AI/ML models require approaches that harden them against such attacks.

The proposed intelligent railway control system architecture may be deployed in a cloud at the edge of the railway communication network and become a point of intrusions and attacks. The security of the cloud is one of the main challenges in cloud computing. Studies of cloud computing security challenges, issues, threats, and possible solutions were discussed in [49,50]. In [51,52], the authors presented security challenges related to microservices applications and described different security solutions and practices.

The identified security considerations, which are related to the proposed railway control system architecture, are inherent in open systems and require the adoption of standards and best practices.

8. Conclusions

This paper presented a means of linking AI/ML techniques in the railway domain to improve service reliability, availability, efficiency, safety, and security. The main contribution of the research is the application of a disaggregated approach to the design of the ETCS Level 3 control system, which enables the incorporation of AI/ML. The proposed architecture enables railway operators to provide railway operation with self-optimization capabilities, which use automation to manage railway services more efficiently. Automation can simplify railway operations and management. One advantage of the architecture is the incorporation of ML framework, where the intelligent railway controller enables operators to programmatically control the railway network in both near-real time and non-real time. The support of ML models, which automate operations and make data-driven decisions, enables the deployment of railway operations and third-party applications. Predictive AI/ML models, e.g., for track monitoring, use algorithms to process track state data and analyze previous and current events to find patterns. Incorporating such tools and automation helps to increase safety and minimize human errors. The proposed architecture promotes the virtualization of ETCS control system functions, in which the disaggregated components are connected via open interfaces and optimized using IRC. Bringing programmability into the ETCS control system is one of the greatest benefits of virtualization. Programmability enables the development of applications that allow the creation of more sustainable railways that improve safety, increase capacity, and reduce operating costs.

This paper applied the principles of microservice architecture to the design of IRC functionality. The microservices architecture exposes well-known benefits of increased scalability, improved productivity, fault tolerance, and better resilience and capabilities for function optimization.

Along with these benefits, the microservice architecture has some disadvantages. Microservice flexibility and agility introduce operational complexity, meaning that strong service-level separation and composition are required. The design of microservices increases communication and coordination, and even though services can be deployed in isolation, they must work together, and any interaction failures can lead to brittleness, auditing difficulty, and debug deployments.

Based on the appropriate approach to mitigating the drawbacks, the shift to the cloud-based and as-a-service model can reduce the costs of implementing new features and railways operations. New technologies can be deployed more quickly thanks to the adoption of microservices and application programming interfaces, and artificial intelligence can contribute to economic efficiency and environmental compatibility.

In a future study, the data types of the defined service interfaces will be defined, and the service logic will be developed.

Author Contributions: I.A. contributed to the software and methodology. E.P. contributed to the conceptualization and writing—original draft preparation. V.V. contributed to the formal analysis and verification. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Bulgarian National Science Fund grant number KP-06-H57/12.

Data Availability Statement: This study did not report any data.

Conflicts of Interest: The authors declare no conflict of interest regarding the publication of this paper. The funders had no role in the design of this study; the collection, analysis, or interpretation of data; the writing of the manuscript; or the decision to publish the results.

References

1. Rosberg, T.; Cavalcanti, T.; Thorslund, B.; Prytz, E.; Moertl, P. Driveability analysis of the european rail transport management system (ERTMS)—A systematic literature review. *J. Rail Transp. Plan. Manag.* **2021**, *18*, 100240. [\[CrossRef\]](#)
2. Rosberg, T.; Thorslund, B. Radio communication-based method for analysis of train driving in an ERTMS signaling environment. *Eur. Transp. Res. Rev.* **2022**, *14*, 18. [\[CrossRef\]](#)
3. Knutsen, D.; Olsson, N.O.E.; Fu, J. ERTMS/ETCS Level 3: Development, assumptions, and what it means for the future. *J. Intell. Connect. Vehi.* **2023**, *6*, 34–45. [\[CrossRef\]](#)
4. Mulongo, N.Y.; Mnkandla, E.; Kanakana-Katumba, G. Artificial Intelligence as Key Driver for Competitiveness in the Railway Industry: Review. In Proceedings of the 2021 62nd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS), Riga, Latvia, 14–15 October 2021; pp. 1–6. [\[CrossRef\]](#)
5. Yang, N.; Chen, M. Design and Application of Big Data Technology Management for the Analysis System of High Speed Railway Operation Safety Rules. In Proceedings of the IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS), Raichur, India, 24–25 February 2023; pp. 1–6. [\[CrossRef\]](#)
6. Gesmann-Nuissl, D.; Kunitz, S. Auditing of AI in Railway Technology—A European Legal Approach. *Digit. Soc.* **2022**, *1*, 17. [\[CrossRef\]](#)
7. Pencheva, E.; Atanasov, I.; Trifonov, V. Towards Intelligent, Programmable, and Open Railway Networks. *Appl. Sci.* **2022**, *12*, 4062. [\[CrossRef\]](#)
8. Ranjbar, V.; Olsson, N.O.E. Towards Mobile and Intelligent Railway Transport: A Review of Recent ERTMS Related Research. *WIT Trans. Built Environ.* **2020**, *199*, 65–73. [\[CrossRef\]](#)
9. Hansen, D.; Leuschel, M.; Körner, P.; Krings, S.; Naulin, T.; Nayeri, N.; Schneider, D.; Skowron, F. Validation and real-life demonstration of ETCS hybrid level 3 principles using a formal B model. *Int. J. Softw. Tools Technol. Transf.* **2020**, *22*, 315–332. [\[CrossRef\]](#)
10. Ranjbar, V.; Olsson, N.O.; Sipilä, H. Impact of signalling system on capacity—Comparing legacy ATC, ETCS level 2 and ETCS hybrid level 3 systems. *J. Rail Transp. Plan. Manag.* **2022**, *23*, 100322. [\[CrossRef\]](#)
11. Saddem-Yagoubi, R.; Sanwal, M.U.; Libutti, S.; Benerecetti, M.; Beugin, J.; Flammini, F.; Ghazel, M.; Janssen, B.; Marrone, S.; Mogavero, F.; et al. Toward Usable Formal Models for Safety and Performance Evaluation of ERTMS/ETCS Level 3: The PERFORMINGRAIL Project. In Proceedings of the AIIT 3rd International Conference on Transport Infrastructure and Systems (TIS ROMA 2022), Rome, Italy, 15–16 September 2022; pp. 321–327.
12. Mazini, A.; Samra, M.; Chen, L.; Blumenfeld, M.; Nicholson, G. Specification and Design of a Modular and Extensible Architecture for Testing Moving Block Systems. *WIT Trans. Built Environ.* **2022**, *213*, 147–158. [\[CrossRef\]](#)
13. Cansu, U.; Atieh, K.; Stefano, R. Influence of Signalling Systems on the Capacity of Railways by Lines and Nodes Assessment Methods. *Transp. Res. Procedia* **2023**, *69*, 321–327. [\[CrossRef\]](#)
14. Rosberg, T.; Thorslund, B. Impact on driver behavior from ERTMS speed-filtering. *J. Rail Transp. Plan. Manag.* **2023**, *26*, 100386. [\[CrossRef\]](#)
15. Dghaym, D.; Dalvandi, M.; Poppleton, M.; Snook, C. Formalising the Hybrid ERTMS Level 3 specification in iUML-B and Event-B. *Int. J. Softw. Tools Technol. Transf.* **2020**, *22*, 297–313. [\[CrossRef\]](#)
16. Basile, D.; ter Beek, M.H.; Ferrari, A.; Legay, A. Exploring the ERTMS/ETCS full moving block specification: An experience with formal methods. *Int. J. Softw. Tools Technol. Transf.* **2022**, *24*, 351–370. [\[CrossRef\]](#)
17. Stankaitis, P.; Iliasov, A.; Kobayashi, T.; Ait-Ameur, Y.; Ishikawa, F.; Romanovsky, A. A refinement-based development of a distributed signalling system. *Form. Asp. Comput.* **2021**, *33*, 1009–1036. [\[CrossRef\]](#)
18. Geisler, S.; Haxthausen, A.E. Stepwise development and model checking of a distributed interlocking system using RAISE. *Form. Asp. Comput.* **2021**, *33*, 87–125. [\[CrossRef\]](#)
19. Zhang, T.; Li, Y. Distributed Multiple High-Speed Trains Consensus Control Based on Event-Triggered Mechanism. *Symmetry* **2022**, *14*, 1846. [\[CrossRef\]](#)

20. He, X.; Li, H.; Jiang, Y.; Shi, J. Analysis of Technical Schemes for Restructuring of Signaling Systems in Urban Rail Transit. In Proceedings of the IEEE 7th International Conference on Intelligent Transportation Engineering (ICITE), Beijing, China, 11–13 November 2022; pp. 500–504. [\[CrossRef\]](#)
21. Efanov, D.V.; Khóroshev, V.V.; Osadchy, G.V. Principles of Safety Signalling and Traffic Control Systems Synthesis on Railways. In Proceedings of the International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Sochi, Russian, 15–19 May 2023; pp. 634–638. [\[CrossRef\]](#)
22. Zhang, J. Simulation-Based Schedule Optimization for Virtual Coupling-Enabled Rail Transit Services with Multiagent Technique. *J. Adv. Transp.* **2023**, *2023*, 3196066. [\[CrossRef\]](#)
23. Ambati, M.; Rao, L.S.; Prathipati, P.S.; Kumar, A.S. IOT Based Event Logger for Railway Signaling. In Proceedings of the International Conference on Smart and Sustainable Technologies in Energy and Power Sectors (SSTEPS), Mahendragarh, India, 7–11 November 2022; pp. 159–162. [\[CrossRef\]](#)
24. Quaglietta, E.; Wang, M.; Goverde, R. A multi-state train-following model for the analysis of virtual coupling railway operations. *J. Rail Transp. Plan. Manag.* **2020**, *15*, 100195. [\[CrossRef\]](#)
25. Blagojević, A.; Stević, Ž.; Marinković, D.; Kasalica, S.; Rajilić, S. A Novel Entropy-Fuzzy PIPRECIA-DEA Model for Safety Evaluation of Railway Traffic. *Symmetry* **2020**, *12*, 1479. [\[CrossRef\]](#)
26. Tan, J.; Xiang, P.; Zhao, H.; Yu, J.; Ye, B.; Yang, D. Stochastic Analysis of Train Running Safety on Bridge with Earthquake-Induced Irregularity under Aftershock. *Symmetry* **2022**, *14*, 1998. [\[CrossRef\]](#)
27. Gao, P.; Zhao, M.; Xie, S.; Qiu, K.; Wang, T.; Yang, Z. An Edge Computing-Based Platform of Railway Signalling System On-site Digital Smart Test. In Proceedings of the IEEE 7th International Conference on Intelligent Transportation Engineering (ICITE), Beijing, China, 11–13 November 2022; pp. 511–516. [\[CrossRef\]](#)
28. Vanichchanunt, P.; Tanmalaporn, T.; Suthamvijit, C.; Noisri, S.; Wuttisittikulkij, L.; Pongyart, W.; Paripurana, S. Virtual Reality for Railway Signaling System Training. In Proceedings of the 20th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Nakhon Phanom, Thailand, 9–12 May 2023; pp. 1–4. [\[CrossRef\]](#)
29. Sinha, R.; Jagadisha, T.; Spandana, S.; Mahan, S.R.; Patil, A.P. Design of a Real-Time Train Control and Management System. In Proceedings of the IEEE Bangalore Humanitarian Technology Conference (B-HTC), Vijayapur, India, 8–10 October 2020; pp. 1–6. [\[CrossRef\]](#)
30. Luo, J.; Peng, Q.; Wen, C.; Wen, W.; Huang, P. Data-driven decision support for rail traffic control: A predictive approach. *Expert Syst. Appl.* **2022**, *207*, 118050. [\[CrossRef\]](#)
31. Silva-Rodríguez, J.; Salvador, P.; Naranjo, V.; Insa, R. Supervised contrastive learning-guided prototypes on axle-box accelerations for railway crossing inspections. *Expert Syst. Appl.* **2022**, *207*, 117946. [\[CrossRef\]](#)
32. Wang, H.; Hao, L.; Sharma, A.; Kukkar, A. Automatic control of computer application data processing system based on artificial intelligence. *J. Intell. Syst.* **2022**, *31*, 177–192. [\[CrossRef\]](#)
33. Putra, H.G.P.; Supangkat, S.H.; Nugraha, I.G.B.B.; Hidayat, F.; Kereta, P. Designing Machine Learning Model for Predictive Maintenance of Railway Vehicle. In Proceedings of the International Conference on ICT for Smart Society (ICISS), Bandung, Indonesia, 2–4 August 2021; pp. 1–5. [\[CrossRef\]](#)
34. Daniyan, I.; Mpofu, K.; Muvunzi, R.; Uchegbu, I.D. Implementation of Artificial intelligence for maintenance operation in the rail industry. *Procedia CIRP* **2022**, *109*, 449–453. [\[CrossRef\]](#)
35. Sresakoolchai, J.; Kaewunruen, S. Railway infrastructure maintenance efficiency improvement using deep reinforcement learning integrated with digital twin based on track geometry and component defects. *Sci. Rep.* **2023**, *13*, 2439. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Shi, P.; Hu, H. Short-time Passenger Flow Prediction Model based on Combined Model for Large Events in and Out of Rail Stations. In Proceedings of the 4th International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), Hamburg, Germany, 7–9 October 2022; pp. 208–212. [\[CrossRef\]](#)
37. Zhang, Z.; Wang, C.; Gao, Y.; Chen, Y.; Chen, J. Passenger Flow Forecast of Rail Station Based on Multi-Source Data and Long Short Term Memory Network. *IEEE Access* **2020**, *8*, 28475–28483. [\[CrossRef\]](#)
38. Yang, J.; Wang, C.; Yi, J.; Du, Y.; Sun, M.; Huang, S.; Zhao, W.; Qu, S.; Ni, J.; Xu, X.; et al. Railway Intrusion Events Classification and Location Based on Deep Learning in Distributed Vibration Sensing. *Symmetry* **2022**, *14*, 2552. [\[CrossRef\]](#)
39. Ghute, M.; Barhate, A.; Dhengle, S.; Bakal, Y.; Kawale, S.; Rathod, D. Railway Signalling System using Encoder and Decoder. In Proceedings of the 7th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 11–13 April 2023; pp. 248–250. [\[CrossRef\]](#)
40. Bai, Z.; Wang, H.; Yang, L.; Li, J.; Lu, H. A Rescheduling Approach for Freight Railway considering Equity and Efficiency by an Integrated Genetic Algorithm. *J. Adv. Transp.* **2023**, *2023*, 8989644. [\[CrossRef\]](#)
41. Li, J.; Peng, Q.; Wen, C. Statistical Analysis of Train Operation and Passenger Distribution Based on Real Records: A Case Study of Wuhan-Guangzhou HSR. *J. Adv. Transp.* **2023**, *2023*, 8923716. [\[CrossRef\]](#)
42. Zeller, M.; Rothfelder, M.; Klein, C. safe.trAIIn—Engineering and Assurance of a Driverless Regional Train. In Proceedings of the IEEE/ACM 2nd International Conference on AI Engineering—Software Engineering for AI (CAIN), Melbourne, Australia, 15–16 May 2023; p. 197. [\[CrossRef\]](#)
43. De Donato, L.; Dirnfeld, R.; Somma, A.; De Benedictis, A.; Flammini, F.; Marrone, S.; Azari, M.S.; Vittorini, V. Towards AI-assisted digital twins for smart railways: Preliminary guideline and reference architecture. *J. Reliab. Intell. Environ.* **2023**, 1–15. [\[CrossRef\]](#)

44. Tang, R.; De Donato, L.; Besinović, N.; Flammini, F.; Goverde, R.M.; Lin, Z.; Liu, R.; Tang, T.; Vittorini, V.; Wang, Z. A literature review of Artificial Intelligence applications in railway systems. *Transp. Res. Part C: Emerg. Technol.* **2022**, *140*, 103679. [[CrossRef](#)]
45. Sánchez-García, R.J. Exploiting symmetry in network analysis. *Commun. Phys.* **2020**, *3*, 87. [[CrossRef](#)]
46. Rao, L.; Liu, S.; Peng, H. An Integrated Formal Method Combining Labeled Transition System and Event-B for System Model Refinement. *IEEE Access* **2022**, *10*, 13089–13102. [[CrossRef](#)]
47. Xu, X. On Bisimulation in Absence of Restriction. Cornell University, Computer Science. *arXiv* **2022**, arXiv:2210.10574.
48. Wu, H.; Long, H. Probabilistic weak bisimulation and axiomatization for probabilistic models. *Inf. Process. Lett.* **2023**, *182*, 106399. [[CrossRef](#)]
49. Kaur, M.; Kaimal, A.B. Analysis of Cloud Computing Security Challenges and Threats for Resolving Data Breach Issues. In Proceedings of the International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 23–25 January 2023; pp. 1–6. [[CrossRef](#)]
50. Alrasheed, S.H.; Alhariri, M.A.; Adubaykhi, S.A.; El Khediri, S. Cloud Computing Security and Challenges: Issues, Threats, and Solutions. In Proceedings of the 5th Conference on Cloud and Internet of Things (CIoT), Marrakech, Morocco, 28–30 March 2022; pp. 166–172. [[CrossRef](#)]
51. Raj, P.; Vanga, S.; Chaudhary, A. Microservices Security. In *Cloud-Native Computing: How to Design, Develop, and Secure Microservices and Event-Driven Applications*; IEEE: Piscataway, NJ, USA, 2023; pp. 289–298. [[CrossRef](#)]
52. Kalubowila, D.C.; Athukorala, S.M.; Tharaka, B.A.S.; Samarasekara, H.W.Y.R.; Arachchilage, U.S.S.S.; Kasthurirathna, D. Optimization of Microservices Security. In Proceedings of the 3rd International Conference on Advancements in Computing (ICAC), Colombo, Sri Lanka, 9–11 December 2021; pp. 49–54. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.