applied sciences

MDPI

Article

Edge Intelligence Service Orchestration with Process Mining

Yong Zhu [1,2,3,*], Zhihui Hu [2] and Zhenyu He [3,4]

[1] School of Computer Engineering, Jinling Institute of Technology, Nanjing 211169, China
[2] School of Computer Science & Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
[3] Information Department, HOHAI University, Nanjing 210098, China
[4] Department of Land Surveying and Geo-Informatics, Hong Kong Polytechnic University, Hong Kong 999077, China
* Correspondence: zhudz@jit.edu.cn; Tel.: +86-1816-809-2326; Fax: +86-25-86188957

Abstract: In the post-cloud computing era, edge computing as a distributed computing paradigm, integrating the core capabilities of computing, storage, network, and application, provides EIS (edge intelligence service), such as real-time business, data optimization, intelligent application, security, and privacy protection. The EIS has become the core value driver to promote the IoE (Internet of Everything), to dig deeply into data value and create a new ecology of application scenarios. With the emergence of new business processes, EIS orchestration has also become a hot topic in academic research. A design methodology based on a complete "describe-synthesize-verify-evaluate" process was established to explore executable design specifications for EIS by means of model validation and running instances. As proof of concept, a CPN (colored Petri net) prototype was simulated and its operational processes were discovered by process mining from event data available in EIS for behavior verification. The instances running on WISE-PaaS demonstrate the feasibility of the research methodology, which aims to optimize EIS through service orchestration.

Keywords: EIS; process mining; service orchestration; CPN

check for updates

Citation: Zhu, Y.; Hu, Z.; He, Z. Edge Intelligence Service Orchestration with Process Mining. *Appl. Sci.* **2022**, *12*, 10436. https://doi.org/10.3390/app122010436

Academic Editor: Miguel García-Pineda

Received: 4 September 2022
Accepted: 9 October 2022
Published: 16 October 2022

1. Introduction

New computing environments, such as cloud computing and IoT (Internet of Things) provide convenient services, such as data sharing and fusion computing, greatly making use of the data processing capacity as well as computing and storage resources [1]. According to Cisco, the global network traffic reached 278 EB/month in 2021. Moreover, according to the forecast by Machina research, the number of global IoT connections will increase to 27 billion and cellular connections will reach 22 billion in 2025 [2]. The number of networked devices is increasing rapidly, resulting in an increasing amount of data and traffic bringing great pressure on network resources. It is difficult for traditional cloud computing to respond to requests of terminal devices in real-time in the case of insufficient bandwidth [3]. Moreover, the challenges are as follows [4]:

- Cloud centralized-based services might become bottlenecks when distributed queries and updates are frequent. Meanwhile, malicious attacks can also lead to a single point of failure;
- With the diversification of business processes and dynamic scalability of functions, cloud centralized-based services, and data processing cannot adapt to the flexibility and real-time metrics;
- DOs (data owners) and DCs (data consumers) relying on data centers may have data risks of security and privacy, which may damage the rights and interests of DOs [5].

In order to respond to the existing challenges, meet business needs, and improve user experiences, the edge computing paradigm is proposed to "sink" the cloud computing power and improve the processing capacity of edge devices, which provides edge services near the data source to meet the key requirements in terms of agile connectivity and real-time optimization and intelligent application and security and privacy [6]. The rise of edge intelligence of IIoT will promote the manufacturing industry to dig deeply into data value, to create a new ecology of industrial production. As a combination of edge computing and AI, edge intelligence uses computing resources from near data sources to complete intelligent tasks; it is a feasible solution for industrial intelligence applications. Thus, edge computing has gradually become an effective supporting platform [7] for new IoE applications, which have three-tier architecture, including "cloud-edge-end" [8]. The core layer edge supports the downward (southward) access to various field devices and upward (northward) connection with the cloud. EIS provides core value drivers for the implementation of IoT applications, helping customers improve operational efficiency and business transformation. In the post-cloud computing era, edge computing, as a distributed computing paradigm integrating the core capabilities of computing, storage, network, and application, providing edge intelligent services, such as real-time business, data optimization, intelligent application, security, and privacy protection nearby, enabling developers to quickly develop and deploy edge applications. Industry application scenarios are constantly enriched, such as autonomous driving [9], IIoT, VR/AR, smart medical care, etc. At present, the fusion of EIS, AI, and blockchain has also become a hot topic in academic research. It is indicated by searching for the keywords: blockchain, MEC (multi-access edge computing), ML (machine learning), and IoT [10], which, on the one hand, blockchain introduces security, privacy, and trust to MEC, and on the other hand, MEC improves the scalability of blockchain in a distributed and efficient manner. With various intelligent services showing great vitality, such as XaaS derived from cloud native computing, edge computing, and blockchain, such as BaaS and FaaS (functions as a service), as well as SDX defined by software, such as SDN (software-defined network) and SDEc (software defined edge computing), researchers continue to put forward new models and algorithms for solving the increasingly complex requirements. Data service servers [11] monitor and manage data flow and enable them to interact with users. DIaaS (data integrity as a service) [12] is implemented by smart contracts on the blockchain, which is fully decentralized. EIS supports the business expansion and operation of developers, taking advantage of the deployment and management of cloud service infrastructure. Services on edge computing rely on the adequate decomposition of the system into ECN (edge computing node) or SMCs (state machine components), which form a service composition/orchestration, where particular operations are executed successively. The model described by a Petri net can be decomposed into two nets that are able to exchange messages upon the occurrence of observable events. An SPN (stochastic Petri net) model represents and evaluates the performance of an edge computing architecture. Service composition for edge intelligence, particularly for an AI subtask composition, is carried out on an EdaaS (edge device as a service), which has a particular functionality and a set of non-functional features.

Data science and big data signify the growing importance of data-based approaches. Real behavior can be reconstructed from data event logs [13]. Process mining techniques provide a wide range of data-driven methods that are process-centric at the same time, which support discovering process models as well as behavior verification and process discovery in past executions of processes [14]. Since the generated simulation model is supported by historical data (event data), which is based on the DES (discrete event simulation) technique, the generated event data are similar to the behavior of the real process. In all of the proposed tools for simulation in process mining, interaction with the user and their knowledge is an undeniable requirement for designing and running the simulation models. Motivated by the above innovative paradigm (edge computing, EIS, process mining), design methodology (XaaS, SDX), and toolchain (WISE-PaaS, CPN Tools, ProM), the fine-grained EIS orchestration with process mining is proposed. The purpose of this study is to establish a design methodology and explore executable design specifications to optimize EIS.

The key contributions of this article can be summarized as follows:

- The methodology (mixing the modeling and verifying for structure, behavior, and data of EIS) provides a constructive way to realize the service's flexible scalability and process automation, which opens up space for optimizing fine-grained service orchestration through DSE (design space exploration).
- Behavior logic, as well as architecture, are extracted from the event logs available in EIS application scenarios by the operational process discovery of process mining, to improve the service orchestration. The cross of CPN Tools and ProM provides deeper insight into the knowledge of EIS orchestration.

This paper is organized as follows. After introducing the research significance, the objectives and contributions in Sections 1 and 2 explore edge intelligence, process mining technology, and its relationship with the Petri net model and our research route. The prototype of intelligent cloud edge in CPN is put forward in Section 3, including color set attributes and logic assignment with attribute predicate. Section 4 elaborates on the simulation verification of EIS with CPN Tools, and its service instance runs on WISE-PaaS. Section 5 discusses the process behavior and event data by process mining as proof of concept. Section 6 concludes the paper. Finally, future research problems are discussed.

2. Related Work

For edge intelligence, AI provides technologies and methods for edge computing, and edge computing provides scenarios and platforms for AI. "AI on edge" focuses on how to build an AI model on the edge computing platform, mainly including model training and inference; "AI for edge" focuses on providing better solutions to key problems in edge computing with the help of advanced AI technology, mainly including task unloading and edge caching [15]. From the dimension of edge computing-enabled AI, an on-demand acceleration framework based on edge-end collaboration is proposed, aimed at the deployment of deep learning model on the edge of the network. From the dimension of edge computing empowered by AI, an adaptive edge service placement mechanism based on online learning and an edge service migration method based on the factor graph model are proposed, aimed at the placement of edge computing services [16]. Edgent [17] regards the deep neural network as a directed graph and optimizes it from two aspects of segmentation and simplification for a deep learning model. The optimization strategy is a trade-off between the model inference speed and accuracy. Edgeke [18] balances resource consumption and inference performance on resource-constrained edge devices. The neural network is compressed by knowledge distillation to reduce the demand for computing resources, and the early exit technology is used to provide a flexible computing method for the neural network; through the EdgeMI algorithm, the distributed computing of a deep neural network is realized on heterogeneous edge clusters, and the final performance acceleration ratio reaches 1.84x–3.57x.

Software-defined computing gives birth to cloud computing, which allows the flexible allocation of computing resources. There are software-defined radios, software-defined networks, software-defined data centers, and so on. SDEC [19] involves abstracting (the logical relationship is abstracted into rules, and inferencing is carried out according to the real-time state), virtualizing (the device model is virtualized to realize the decoupling and separation of software, hardware, and services) and pooling (unified management, sharing, reuse, and cooperation) the physical equipment resources on the edge by using semantic description modeling, knowledge graphs, and other technologies. When the edge computing gateway software [20] processes a variety of IoT communication protocol messages (MQTT, OPC-UA, etc.), it forwards them to local computing for data collection, analysis, and AI processing, according to the message routing rule engine, and responds quickly. SDN and virtualization technology optimize the edge computing architecture by defining the management and monitoring of computing, storage, data center, security, and other resources through software, to realize resource integration and management collaboration within multiple networks. The collaboration capability of SDN-based cloud-edge network framework [21] is reflected in ① EC-IaaS resource collaboration; ② EC-PaaS management collaboration; ③ and EC-SaaS application service collaboration, in other words, the application layer, control layer, and infrastructure layer of SDN correspond to the three fields of application services, platform management, and resource facilities in the cloud-edge collaboration. In SDIoT architecture [22], combining SDN and IoT, the resource-limited applications can be efficiently managed at the edge, collecting sensor data from terminal nodes through SBI (southbound interface) and providing connectivity to the cloud service through the NBI (northbound interface).

The embedded devices in the IIoT constitute the edge computing layer in the industrial internet. Heterogeneity is their most significant characteristic, low latency is the main motivation for industrial scenarios, and security is another driving force for the application of edge computing in industries. The challenges of edge computing [23], such as system modeling with deterministic time delay, industrial task design, and unloading, real-time container technology, resource management in the heterogeneous computing environment and constrained resource environment, and broader forms of resource sharing are proposed. OT experts believe that it is necessary to master the industrial mechanism model so that AI can be applied a really work on edge; IT experts emphasize the role of AI in building a common computing architecture [24]. Edge intelligence in industries needs to closely combine the AI algorithm and model with industrial knowledge and mechanisms in order to give full play to its effect. The value of the cloud edge combination provides users with cloud-consistent functionality and applications and experiences on infrastructure. Kubedge integrates cloud, management, edge, and end to solve five key problems of edge computing [25]: cloud-edge collaboration, heterogeneous support, large-scale device access, lightweight edge and end, and consistent experience. The edge-cloud project OpenYurt [26] of CNCF extends the native Kubernetes to the intelligent open platform of edge computing, which ensures non-invasive standardization to realize edge autonomy and central cloud control.

The decomposition technique based on the Martinez-Silva is put forward, intended for obtaining all the minimal support p-invariants of a Petri net. Among such invariants, the sets of places corresponding to the SMCs will be found [27]. Reference [28] proposes an SPN model to represent the MEC architecture and evaluate the trade-off between the MRT (mean response time) and resource utilization. The architecture is composed of three parts: mobile devices, front end, and edge computing; the resources of single servers are parallelized using containers. The main objective is to minimize resource costs and maximize performance. A blockchain-based decentralized solution for service composition [29] has been introduced in the scope of complex multimedia service delivery. Reference [30] proposed a collaborative framework at the network edge, aiming for a swift composite service delivery system and service composition models. A simulation-based optimization method using the SPN model for service composition [31] was put forward. Reference [32] highlights both QoS (Quality of Service) and QoE (Quality of Experience) and solves the service composition in an edge computing environment with a special focus on fault tolerance. The work in [33] focuses on workflow processes with concurrent behavior, which uses an extended version of the α algorithm to incorporate timing information [34]. Process mining can be seen as a tool in the context of BPI (business process intelligence) [35], and transition systems can be represented by compact Petri nets [36]. PAIS (process-aware information systems) by process mining discovers process models from event logs, analyzes performance characteristics of processes, and establishes behavioral relations between Petri nets and BPMN (business process model and notation) models in order to visualize metrics within a BPMN diagram. The Petri net (free-choice workflow nets) and workflow graphs deal with "high-level" process models (e.g., BPMN models, control flow models, data, and resource perspectives) [37]. The conversion algorithms help to construct flat control flow skeletons of the target BPMN models from the discovered Petri nets and other "low-level" models, covering both the control flow perspective and the resource perspective [38]. A framework [39] is proposed with core components (constraint formula) enabling the constraint monitor. More tailor-made rules can be formalized into Petri-net patterns [40] or linear temporal logic to evaluate whether process executions comply with them. The model-based and clustering-based approaches learn the rules by analyzing event data. The constraint formula can be extended to predict and evaluate future violations. A process mining tool generates signals by analyzing the event data and executing the actions.

The attributes of related work are shown in Table 1:

Table 1. Comparison of Related Work Attributes.

Categories	References	Attributes		
		AI	**Service Management**	**Service Orchestration**
edge intelligence	[15–18]	strong	weak	weak
SDx	[19–22]	average	average	average
application	[23–26]	weak	strong	average
decomposition/ composition	[27–32]	average	average	strong
Petri Net and process mining	[33–40]	weak	average	strong

Edge intelligence integrates AI into edge computing and is deployed on the edge, focusing on AI algorithms, model training, and inference. The core of EIS is about task migration and offloading, resource allocation, heterogeneous computing, and cloud-edge collaboration. EIS orchestration arranges (intelligent/automatic invocation) multiple services according to service requirements to realize complex applications. For example, embedded machine learning services can monitor product quality in real-time based on sensor data, orchestrating related software services to complete the business processes. SOA (service-oriented architecture) is loosely coupled, and the services between nodes provide information interaction, which lays the foundation for EIS orchestration.

Referring to related work and following the “describe—synthesis—verify—evaluate” methodology, the research is as follows:

1. The synthesizable CPN model was firstly constructed for theoretical verification, taking cloud-edge intelligent entities, such as nodes, gateways, interfaces, data centers, etc., as the places, and taking behavior and logic as the transition function and guard to imitate service orchestration for specific business processes.
2. After simulation verification, the behavior synthesis was carried out, based on executable specification. The standard definition of the function module, interface protocol, and hierarchical network provided strong support for the abstraction of data interaction and business processes. Each element could be composed of multiple business logics, each of which could be individually listed as a service entry in the service contract, providing rich service orchestration with structured modules and a hierarchical network.
3. The evaluation instances were run on WISE-PaaS for DSE. With the help of a toolchain (DeviceOn, NodeRED, MQTT, Dashboard) and management protocols to dynamically deploy and execute multiple parallel applications, the efficiency of the system development has been greatly improved.
4. With that, the knowledge of fine-grained service orchestration was extracted by process mining from event data available in EIS for behavior verification and process discovery, so as to improve the service orchestration.

3. Intelligent Cloud-Edge Model

Following the design methodology of "description-synthesis-verification-optimization", the intelligent cloud-edge architecture was built at first, and the corresponding using-view and deployment view were explained to ensure the implementation of the system. Based on this, the model in CPN is described, which focuses on the hierarchical interface and highlights cloud-edge collaboration and intelligent services. It supports the executable design specifications using CPN Tools. The model simulation verification and its instance activity combined with typical services are presented in the following two sections.

Edge computing is a distributed IT architecture [41], and the platform services are mainly based on EIS and cloud component services. Cloud-edge intelligent service has the ability to bridge and coordinate the two (cloud, edge) to guide the partition and execution of tasks, so that time-sensitive data can be preprocessed by the local or nearest EIS, and time-insensitive data can be sent to the cloud for big data intelligence analysis and global long-term storage. It also manages and optimizes the network from the overall view, and has a certain openness to flexibly deploy new applications and services. Intelligent cloud-edge architecture is mainly composed of three parts: ① edge layer: As PaaS in cloud-edge cooperation, it is composed of many edge domain environments such as virtualization, data storage, edge device management, and data processing. For each domain, EIS local resources are defined by software, and SDN is enabled to solve the heterogeneity and scalability of edge devices, so as to schedule edge computing network resources more effectively; ② cloud center: It is responsible for managing, scheduling, integrating, and optimizing various resources distributed on the network by cloud computing technology. The dynamic network system based on SDN enables real-time programming and large-scale management for a value-added cloud computing network, to further enhance its scalability and dynamics and provide new IT service; ③ Hub/Gateway: based on the idea of step-by-step control and global/local control, it ensures the overall consistency to realize the reasonable access to computing tasks and the optimal allocation of computing resources, quickly responding to computing requests and feedback, computing results in real-time, and building a service-oriented network open capability interface to meet the different requirements of services.

Intelligent cloud-edge service involves the following elements: infrastructure, Hub/Gateway, SDN/MQTT, data center, AIFS (AI framework service), service management, orchestration, security, etc. Its architecture is shown in Figure 1:

Using view describes the activities that need to be coordinated between different elements involved in edge computing to guide the implementation of reliable and complex applications, which give the key in each stage of system design, implementation, deployment, operation, and development. The function module and implementation module among using-view provide the basis for the subsequent stage of detailed function design and implementation. The executable module tasks realize the business logic, which has two ways: service orchestration and simple business invocation. User roles involve a variety of user types in the whole life cycle of edge applications, which have corresponding access permissions to ensure the security and reliability of the system. The core elements emphasized in the edge computing scenario include time, space, trigger conditions, results, and constraints. Deployment view describes the structure and techniques of an edge computing architecture, where the former refers to the topology of component distribution and their interconnection, and the latter includes interfaces, protocols, behaviors, and other attributes to ensure that the operations and activities identified in the using-view are correctly mapped to functional components and business processes meet service orchestration and management.

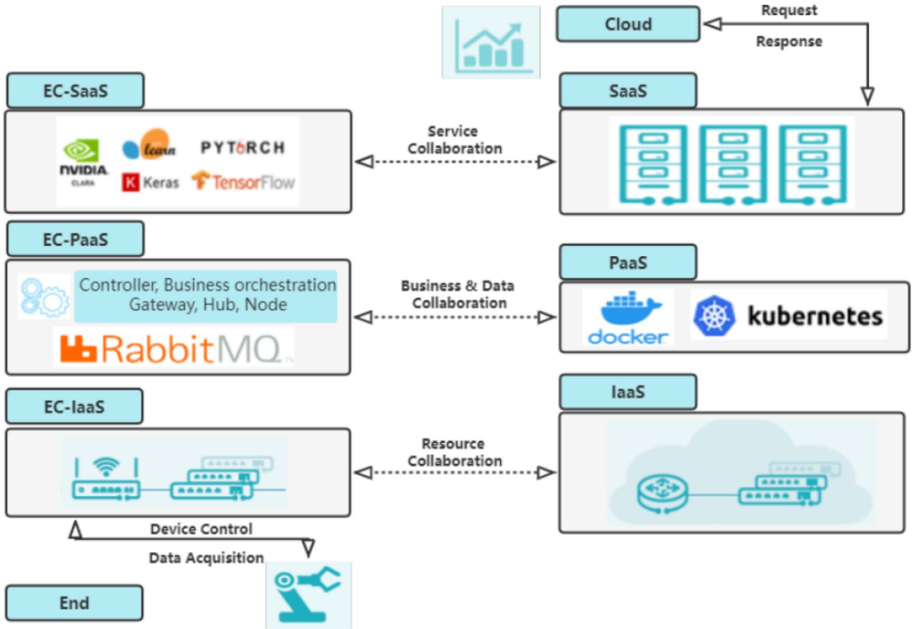


Figure 1. Architecture of Intelligent Cloud-Edge Service.

The preliminaries of CPN and color set are as follows:

Definition 1. *A CPN is a 9-tuple:* $CPN = (P, T, A, \sum, V, C, G, E, I)$ *where:* ① P *is a finite set of places;*② T *is a finite set of transitions such that* $P \cap T = \Phi$; ③$A \subseteq P \times T \cup T \times P$ *is a set of directed arcs;* ④ $\sum$ *is a finite set of non-empty color sets.*⑤ V *is a finite set of typed variables such that* $Type[v] \in \sum, \forall v \in V$. ⑥ $C : P \to \sum$ *is the color set function that assigns a color set to each place.* ⑦ $G : T \to EXPRv$ *is a guard function that assigns a guard to each transition t, such that* $Type[G(t)] = Bool$. ⑧ $E : A \to EXPRv$ *is an arc expression function that assigns an arc expression to each arc a, such that* $Type[E(a)] = C(p)_{MS}$. ⑨ $I : P \to EXPR_{\varnothing}$ *is an initialization function that assigns an initialization expression to each p, such that* $Type[I(a)] = C(p)_{MS}$.

Definition 2. *Let* $S = \{s_1, s_2, s_3, ...\}$ *be a non-empty set, and the multi-set function is* $m : s \to N, \forall s \in S$, *where N is a non-negative integer. The multi-sets are in the following form:* $\sum_{s \in S} m(s)`s = m(s_1)`s_1 + +m(s_2)`s_2 + +m(s_3)`s_3 + +...$*Their basic operations are defined as follows:* ① $\forall s \in S : (m_1 + +m_2)(s) = m_1(s) + m_2(s)$. ② $\forall s \in S : (m_1 - -m_2)(s) = m_1(s) - m_2(s)$. ③ $\forall s \in S : (n * *m)(s) = n * m(s)$. ④ $m_1 \leq m_2 \Leftrightarrow .\forall s \in S : m_1(s) \leq m_2(s)$. ⑤ $|m| = \sum_{s \in S} m(s)$.

By introducing associating data structure (color set) to every token, the modeling ability of CPN is enhanced. Obviously, hierarchical intelligent cloud-edge architecture has an infrastructure layer, intelligent edge layer, cloud service layer, etc., in which there is data and information exchange between entities. The business state is transitioned through running tasks to realize service orchestration. To build the CPN model of hierarchical intelligent cloud-edge architecture, the main entities, such as nodes, gateways, EIS, dedicated function modules and their interfaces, cloud services, data centers, asset resources and other entities are described as places. Moreover, the behavior logic of different states, such as application deployment, task migration, service orchestration, etc., is mapped to transition reflecting the related place and token. The CPN network structure can be divided into multiple functional blocks to form a hierarchical network by substitution transition. The parent-child logical association is implemented by the fusion-set place in CPN Tools. The places in the sub-page are called port-place and those in sup-page are called slot-places, providing powerful support for the decomposition of complex business networks into sub-models through substitution transition and fusion-set, such as southbound heterogeneous data access and northbound RESTFul access through typical layer interfaces of cloud-edge intelligent services.

The key to CPN modeling lies in color set design. The combination of its data structure and guard can describe complex state transitions to represent polymorphic behavior. The basic color set (limited to space, only some examples are listed) is designed as follows:

$colset\ Usr\ =\ STRING$ // User information; The same applies to other entities

$colset\ Usr\_L\ =\ listUsr$ // User list

$colset\ Serve\ =\ with\ Gateway\ |\ AIFS\ |\ Dashboard\ |\ ...\ |\ SSO$

// Enumeration of various services; The same applies to process status

$colset\ DataX\ =\ product\ Data0\ *\ Rule$

// Product data structure that contains attributes, such as rules-based data stream

$colset\ EISAddr\ =\ record\ IP1\ :\ Port1\ *\ IP2\ :\ Port2\ *\ IP3\ :\ Port3$

// Record data structure that maps respective EIS address into the "IP: Port"

Model behavior attribute is assigned by APre (Attribute Predicate). It is a triple of (Entity, ∝, Entity) or (Entity, ∝ <Value>) representing the place object, transitions behavior and color set respectively, where $\propto \in \{\wedge, \vee, \neg, \cup, \cap, \subseteq, \supseteq, \in, \forall, \exists\}$ is an operator to limit the value range of the attribute. Typical APre logic examples are as follows:

$AD \subseteq A \times D$ // Attribute-Data assignment

$ADObj \subseteq AD \times Obj$ // Attribute-Data-Object assignment

$NS(fun, srv) = fun \in N|(fun, srv) \in NS$ // Function map to service set

$RuleD(rule, data) = rule \in Rule|(rule, data) \in RuleD$

// Protocol and rule map to communication, interface, data stream engine

$regEle(addr : AD, src, tgt, rule, prot) :$

$Src \cup \{srcsrc/\ \in\ Src\}, Tgt \cup \{tgt|tgt/\ \in\ Tgt\},\ Rule \cup \{rule|rule\ \in\ Rule\}, Prot \cup \{prot|prot \in Prot\}$

// Register runtime service process elements, such as source, target, rule, protocol, etc

$checkS(name : S, data, attr) : \forall fun(\cap datai \subseteq D) \cap (\cap attri \subseteq A)$

// Check, audit, and verification services

The above notations are based on the color set of the CPN model and logical representation. The former represents object characteristics, which uses the prefix symbol "colset" as a token to apply to CPN Tools software; the latter uses predicate logic and set paradigm to describe object behavior and its relationship with attributes, which can be mapped to transition functions and guards of the CPN model. The corresponding entities and behaviors of the cloud-edge are first mapped to the nine-tuple $(P, T, A, \sum, V, C, G, E, I)$ to construct the model prototype in the CPN modeling process. color set design is the key to representing model data structures to achieve complex data streams. APre assignment logic ensures high-level entity behavior, which in turn supports automated business and intelligent services.

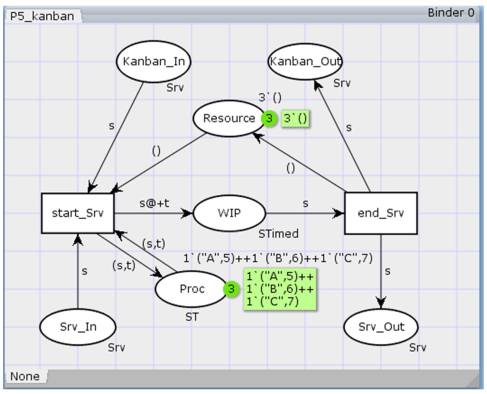
4. Fine-Grained Service Orchestration for EIS

Cloud-edge intelligence is implemented based on the micro-service framework. Services can be called through RESTful or RPC, with the characteristics of business decoupling, decentralization, and "self-service". The EIS at the core layer completes the data acquisition and monitoring of heterogeneous devices southbound, and realizes cloud connection and intelligent device management northbound, which with open standard architecture follows MQTT protocol and docker container technology as the basic framework. Edge interface provides customers with the development application of RESTful API, MQTT communication, and NodeRED data stream. Data infrastructure services provide database and data network services. The data analysis framework integrates a variety of machine learning function libraries and back-end distributed computing resources. The system supports data security (SSL/TSL protocol communication), application security (container deployment) and platform security (based on RBAC, UAA, SSO, and JSON tokens).

WISE-PaaS [42] has functional modules, such as security management, SCADA (Supervisory Control and Data Acquisition), HMI (human and machine interface), AIFS, etc., for end devices to provide comprehensive development tools and standard protocol SDKs with various IoT connectivity. The sensor network connection at the bottom layer of the architecture is responsible for collecting data, managing the sensor hub, converting the sensor protocol into MQTT, and then transmitting the data to the MQTT server or proxy. The management and interface layer uses Webmin for configuration. Whether the service is deployed on edge devices or in the cloud, SSL/TLS communication security features are configured. Cloud-edge intelligent services support elastic capacity expansion to manage complex platform environments and resource expansion. It has the 'penness' ability, enabling users to quickly create, deploy, and manage cloud applications. More software modules can be flexibly added through WISE-PaaS Marketplace to help build, implement, and start cloud-edge intelligent innovative applications, and provide users with vertical industry SRP (solution-ready package) and DFSI (domain focus system integrator).

The model based on service orchestration supports the visual presentation of model business processes through model definitions, such as architecture, interface, functions, and requirements, to generate executable specifications in the form of multi-language version code/script. The edge computing domain model is integrated with the vertical industry domain model through the development platform and toolchain to support model library version management. Service orchestration is based on a three-layer architecture. ① The orchestration service located in the cloud defines business organization processes, provides visual workflow tools, supports CRUD operations, and develops a service framework based on reuse. ② The strategy controller deployed on edge realizes the local nearby control to ensure the real-time performance of business scheduling. In order to better integrate the requirements of edge computing and vertical industry, high-quality and efficient service templates and policy templates can be predefined. ③ The strategy executor module built in each edge computing node is responsible for translating the strategy into the device command and scheduling the execution locally. The strategy focuses on high-level service requirements, and the specific intelligent algorithm realizes fine-grained control of edge computing nodes.

The event with tokens fire state transition in CPN, representing different service behaviors. The subsequently transitional color set represents the attributes of their respective services, so as to realize the fine-grained EIS orchestration mode, i.e., service/business → DBaaS (database as a Service), CaaS (container as a Service) → FaaS, such as CRUD, URL call, Lambda → task process, such as start, stop, update, and message passing. The EIS service orchestration model is both a hierarchical (orchestration service in the cloud–strategy controller on the edge–strategy executor in the node) architecture and a process (data stream and business flow) pattern. The former realizes the hierarchical interface through substitution transitions and fusion-set places, while the latter is described by Kanban information flow, as shown in Figure 2:



CPN Tools Simulation Performance Report
Net: /P5_kanban2.cpn

Marking_size statistics				
Resource	**Count**	**Avrg**	**Min**	**Max**
2	202	0.104575	0	2
3	202	0.456897	0	3
4	202	1.703704	0	4

Data collection statistics					
Processing times	**Count**	**Sum**	**Avrg**	**Min**	**Max**
(2, 3, 4)	100	6929	69.290000	2	153
(5, 6, 7)	100	10197	101.970000	5	232
(8, 9, 1)	100	8010	80.100000	1	270

Generated: Fri Mar 4 16:31:45 2022

Figure 2. EIS Model with Kanban Information Flow.

EIS with fine-grained service orchestration consists of a series of business processes. It's assumed that there are three work nodes (excluding input and output interfaces) in this model, which also provides the Kanban information flow in the opposite direction corresponding to each node to provide feedback on the service status. One of the working nodes is shown in Figure 2, whose structure and behavior are consistent, only relevant parameters are different, such as processing times, resources per work node, and Kanban in between work nodes. The performance and resources of service orchestration are optimized by configuring parameters. The rationality of CPN model behavior can be verified by SSA (state space analysis). DSE is realized through parameter configuration. Performance is evaluated by the monitor of CPN Tools 4.0 in timed color set tokens. Among them, although the color set processing times significantly affect the efficiency of service orchestration. However, the color set resources per work node have little impact, because the conflict between nodes is not fully considered. Kanban's mode tuning can reasonably arrange service orchestration.

The EIS state behavior in the CPN model represented by the color set is verified by CPN Tools, and then its process instance is about to run on WISE-PaaS to evaluate the data stream engine performance and communication proxy QoS (quality of service). WISE-PaaS is built into EIS. Among them, WISE-Agent provides abundant interaction-friendly and intelligent interfaces. Its application is connected to the WISE-PaaS/RMM cloud server, which is not only used to communicate and exchange information between edge devices and cloud platforms but also has the function of a small database and lightweight computing and analysis. The plug-in with a special protocol is configured to meet the requirements of flexible and extended services. The instance of service orchestration with information flow on WISE-PaaS is shown in Figure 3:

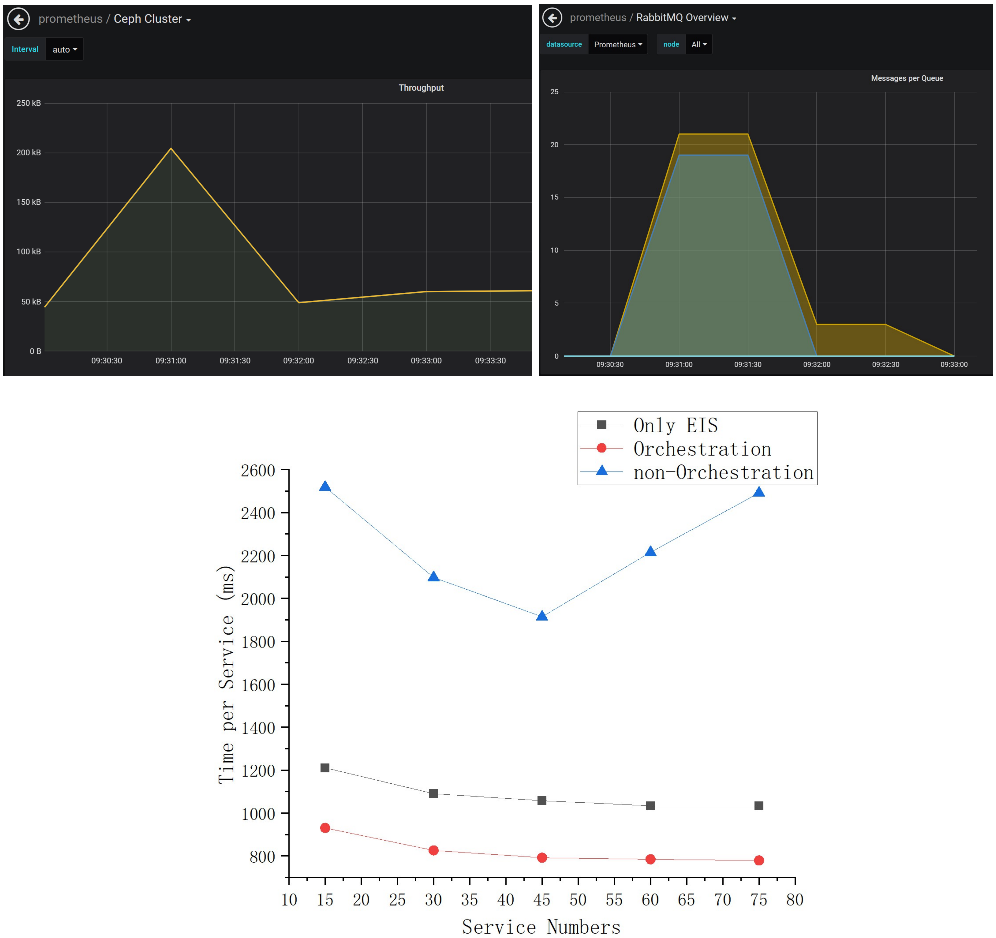


Figure 3. Instance of Service Orchestration Running on WISE-PaaS.

A service consists of several businesses according to the verification model. The service instance is assumed to include five segments of businesses, whose performance (service delay) and QoS (communication queue) are measured on WISE-PaaS for EIS, service orchestration, and non-orchestration. The test platform supports the data stream design engine (Node-RED), message passing (MQTT), function and service management, and output visualization. Service orchestration is based on business loads and node resources, and tasks are dispatched via MQTT topics. The tests show that service orchestration achieves the best results. Only EIS results come in second place because of reduced communication overhead on a single EIS node. The result of non-orchestration warns that the result may deteriorate due to mismatches between nodes and tasks if node resources are not properly utilized for service orchestration. This also basically shows that service delay can be significantly reduced by EIS and its service orchestration compared with centralized cloud computing. If interactive and AI algorithms are used, an optimal performance can be achieved in large-scale and complex environments. However, the benefit gained comes at a cost for edge computing. Therefore, there is a trade-off in cost performance.

5. Process Mining for EIS

Process mining insights support process discovery; operational processes monitor transform into automated actions execution, aim to discover, monitor, and improve business processes by extracting knowledge from event logs available in EIS. It also offers techniques for automatic discovery of process models from event logs, which enhance discovered processes with event data. Discovered process models become more available and understandable, which can be imported/exported from/to most modeling tools to simulate. Moreover, process mining techniques can be easily integrated into the existing suites, and process discovery models with different representations, such as Petri net, transition systems, and BPMN, allow for the combination of different perspectives varying from control flow to the perspective of resources.

The model with process mining for EIS mainly includes three parts: the event logs, rule monitor (similar to Kanban information in the previous section) and service engine (as work nodes in the previous section), which are defined as follows:

- Event logs: They are considered a starting point in the context of process mining. Each event may refer to different objects from different object classes. A conventional event log (trace) is a special case of this event data notion. Their formal definition is defined as follows:
  Let $A \subseteq \mu_A$ be a set of activity labels. Where A trace $\sigma \in A^*$ is a sequence of activity labels. $L \in \beta(A^*)$ is an event log, i.e., a multi-set of traces. Event projection can be expressed as $e = (Trace, name, org, time, state) \in \mu_{event}$(See Table 2).
- Rule Monitor: Abstracted from context, formula, rule, and instance, it is able to analyze future events. Let $R \subseteq \mu_r$ be a set of rules to be used for monitoring. $rm_R \in \mu_{log} \to \mu_{RL}$ is the rule monitor, $\forall L \in \mu_{log}$.
- Service Engine: Based on actions (activity/operation), formula, and instance, it is able to assess future rule instances. Let $A \subseteq \mu_a$ be a set of actions used by the service engine. $se_A \in \mu_{log} \to \mu_{AL}$ is the service engine, $\forall L \in \mu_{log}$.

The architecture of process mining for EIS is shown in Figure 4:

There are two major components of rule monitoring (with the model, knowledge, and rule) and service engine (with a series of actions) defined above. The sound structure of an SWF-net reflects its behavior and vice versa, which includes routing constructs (sequential, parallel, conditional, iterative) and workflow building blocks (AND-split, AND-join, OR-split, OR-join). The mining algorithm, Algorithm α, is able to discover a large class of sound WF-nets/SWF-nets on the basis of complete event logs, whose basic relations are expressed as follows:

- Direct succession: $x > y$ iff for case x is directly followed by y;
- Causality: $x \to y$ iff $x > y$ and not $y > x$;

- Parallel: $x||y$ iff $x > y$ and $y > x$;
- Choice: $x\#y$ iff not $x > y$ and not $y > x$.

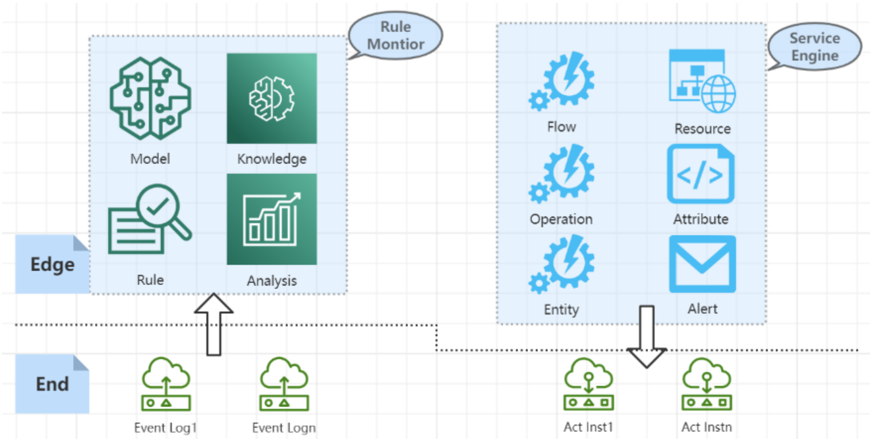


Figure 4. Architecture of Process Mining for EIS.

Notation A, σ, L, e, and R are the element symbols of process mining, which constitute the basis for describing the model. The service business processes with the basic relations $(>, \rightarrow, ||, \#)$ can be mined by importing set L with σ and e into ProM software, combining it with EIS rule R and applying algorithms (such as α Miner).

The process mining approach starts with a subset of model formal semantics (based on token includes activities, start and end events, exclusive and parallel gateways), applying the α mining algorithm on the original event log L (a concrete case for EIS given in Table 2). Therewith, a process model is discovered in the form of a well-defined control flow modeling formalism, such as Petri nets, transition system, and causal nets process.

$$L = [trace1^i, trace2^j, trace3^k \ldots] \tag{1}$$

An event log is represented as a multi-set of traces, in which one trace can appear multiple (n) times. Multi-sets are also used to present the states of Petri nets and BPMN models.

The performance information for a Petri net can be visualized in the initial BPMN model to show the behavioral properties of process models discovered from an event log, and language relations between Petri nets and corresponding BPMN. Subsequently, performance analyses provide the metrics, such as case arrival rate and average duration of the activities, which aim to provide a simulation model and the corresponding simulated event log as close to reality as possible.

Existing process mining tools provide users with a visual representation of process discovery and performance analyses using event data in the form of event logs, which include three main modules: process mining using mining algorithm (Algorithm α), simulation representation (BPMN), and transformation of the generated events into an event log. BPMN packages architecture and their functionality in the tool, such as ProM or Disco, provides the ability to convert Petri nets, transition, and causal nets to BPMN. Moreover, the resource and data flow perspectives can be discovered as well: Process trees with resources can be converted to the BPMN model with lanes, and data Petri nets by the data-aware process mining algorithm can be used to create BPMN models with data perspective.

The EIS is instantiated in a ProM plug-in, whose input objects are event logs shown in Table 2:

Table 2. Event Log.

Trace	Name	Org	Time	State
1	A	JIT	2022-01-08T08:20:01	complete
1	B	JIT	2022-01-08T08:21:01	complete
1	C	JIT	2022-01-08T08:22:01	complete
1	D	JIT	2022-01-08T08:23:01	complete
2	A	JIT	2022-01-10T08:20:01	complete
...				
3	A	JIT	2022-01-12T08:20:01	complete
...				

The process miner includes the following three aspects: ① log summary: learn about process event classes and their related properties in the dashboard, inspector, and summary; ② discovering processes: the model is mined into a Petri net by process mining algorithm; ③ modeling for visualization: replays a log to obtain performance information, and enhances the process. The event logs in Table 1 are expressed as $L = [ABCD^3, ACBD^3, AED^4 \ldots]$ according to Formula (1). Its event occurrences (absolute/relative) are as follows: A = 10/27.8%, B = 6/16.7%, C = 6/16.7%, D = 10/27.8%, E = 4/11.1%. The results mined by Alpha Miner and the inductive visual miner for an EIS instance are shown in Figure 5:

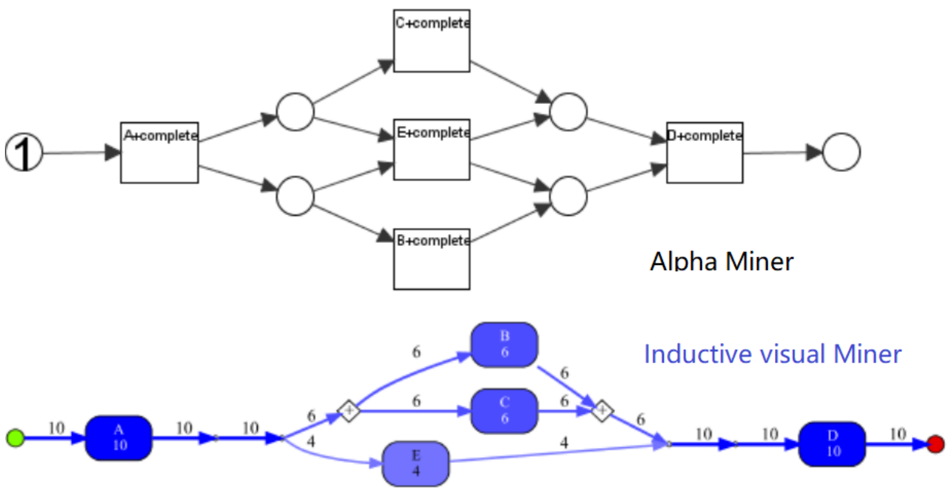


Figure 5. Models mined in ProM.

The above process mining implementation for EIS can discover processes such as proof-of-concept, to enhance and optimize the business processes. The experimental results highlight the research:

- Rule monitors effectively detect violations to analyze future events through context, formula, and rule;
- Service engine effectively generates corresponding actions to assess future rule instances, with business (activity/operation) and formula;
- The application from event data improves business processes by process discovery.

Process mining can achieve structural models preserving behavior recorded in an event log on the control flow perspective; Petri nets, causal nets, and the transition system are generated from an event log by various control flow discovery algorithms.

6. Conclusions

As an extension of cloud computing, edge computing has the advantages of IT decentralization, infrastructure autonomy, edge hosting, and so on. Both of them complement and depend on each other. The value of an organic whole of cloud-edge is that the functions and experiences users obtain on any infrastructure are consistent with those on the cloud, for cloud-edge-end applications. The container-based isolation ensures the business security running on the edge, and the decoupling or loose coupling between resources and services well supports the adaptation of heterogeneous resources. Edge computing fuses with AI to carry out AI processing close to data sources (such as IoT sensors and devices), which can effectively reduce time delay in IIoT.

With the increasing of the system scalability and process complexity in edge computing, on the one hand, it is necessary to effectively manage many internal infrastructures and their massive data, combine them with heterogeneous objects (downward or southward) in different communication protocols and standards, make rational use of cloud computing service capabilities (upward or northward), and expand the integrated system (eastward–westward); On the other hand, innovative BPI should be adopted to deal with EIS, service orchestration, edge real-time computing and analysis.

Following the design methodology, physical objects are abstracted into structure and behavior models to generate executable specifications. The standard function block interface definition and business process logic are provided to map into software service. The efficiency of development and deployment for EIS is improved on WISE-PaaS. Process mining extracts service orchestration knowledge from event data available in EIS to solve process-related problems of modeling, analysis, and monitoring for diverse industries. Our conclusion is that in the post-cloud computing era, the key drivers provided by EIS will become the core values in the implementation of the IoE application. EIS can be effectively improved by constructing special attributes of model behavior, decoupling cloud-edge intelligent services, deploying fine-grained service orchestration. Process mining obtains a structural model in which the behavior is preserved in event data, and discovers various process models (Petri net, BPMN) from event logs, to reinforce, analyze, and visualize EIS processes.

There are still two issues that are worth further research:

- There are the limits on the methodology of the CPN model and process mining for complex EIS, based on the control (process) behavior. In future research, it is necessary to provide the methodology with "value-added" for service orchestration, such as data attributes, which can adapt to well-established control/data structures and behaviors.
- Service orchestration has to go through multiple rigorous model validations, as well as an instance running. In order to solve the above deficiencies, it is necessary to design an efficient executable specification for comprehensive experimental settings and performance comparison and to ensure the consistency of the entire methodology.

Author Contributions: Conceptualization, Y.Z.; methodology, Y.Z.; software, Z.H. (Zhengyu He); validation, Z.H. (Zhengyu He); formal analysis, Z.H. (Zhihui Hu); investigation, Z.H. (Zhengyu He); resources, Z.H. (Zhihui Hu); data curation, Z.H. (Zhihui Hu); writing—original draft preparation, Y.Z.; writing—review and editing, Z.H. (Zhihui Hu); visualization, Z.H. (Zhihui Hu); supervision, Y.Z.; project administration, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the State Key Laboratory of Computer Architecture fund(ICT, CAS) grant number CARCHA202010 and the JIT fund grant number 40620025.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to reason that the study not involving humans or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to acknowledge the State Key Laboratory of Computer Architecture fund (ICT, CAS), the JIT fund, and Jian Ju for WISE-PaaS.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, F.; Tang, G.; Li, Y.; Cai, Z.; Zhang, X.; Zhou, T. A survey on edge computing systems and tools. *Proc. IEEE* **2019**, *107*, 1537–1562. [CrossRef]
2. Cao, B.; Li, Y.; Zhang, L.; Zhang, L.; Mumtaz, S.; Zhou, Z.; Peng, M. When Internet of Things meets blockchain: Challenges in distributed consensus. *IEEE Netw.* **2019**, *33*, 133–139. [CrossRef]
3. Ren, Y.D.J.; Qi, Z.W.; Guan, H.B.; Chen, L. Development and prospect of edge intelligence for industrial Internet. *Strateg. Study CAE* **2021**, *23*, 104–111. [CrossRef]
4. Novo, O. Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT. *IEEE Int. Things J.* **2018**, *5*, 1184–1195. [CrossRef]
5. Cheng, G.J.; Huang, Z.J.; Deng, S.G. Data management based on blockchain and edge computing for Internet of things. *J. Internet Things* **2020**, *4*, 1–9.
6. Qi, Q.; Tao, F. A smart manufacturing service system based on edge computing, fog computing, and cloud computing. *IEEE Access* **2019**, *7*, 86769–86777. [CrossRef]
7. Shi, W.S.; Sun, H.; Cao, J.; Zhang, Q.; Liu, W. Edge Computing: An emerging computing model for the Internet of everything era. *J. Comput. Res. Dev.* **2017**, *54*, 907–924.
8. Edge Computing Reference Architecture 3.0. 2018. Available online: http://www.ecconsortium.org/Lists/show/id/334.html (accessed on 3 September 2022).
9. NVIDIA DRIVE PX2. NVIDIA DRIVE PX2: Scalable AI Platform for Autonomous Driving [EB/OL]. 2019. Available online: https://www.nvidia.com/en-us/self-driving-cars/drive-platform (accessed on 3 September 2022).
10. Du, Y.; Wang, Z.; Leung, V. Blockchain-enabled edge intelligence for IoT: Background, emerging trends and open issues. *Future Internet* **2021**, *13*, 48. [CrossRef]
11. Data Service Server User's Manual(V2.1.0). Available online: www.advantech.com.tw (accessed on 3 September 2022).
12. Liu, B.; Yu, X.L.; Chen, S. Blockchain based data integrity service framework for IoT data. In Proceedings of the 2017 IEEE International Conference on Web Services (ICWS), Honolulu, HI, USA, 25–30 June 2017; pp. 468–475.
13. Kalenkova, A.A.; van der Aalst, W.M.; Lomazova, I.A.; Rubin, V.A. Process mining using BPMN: Relating event logs and process models. *Softw. Syst. Model.* **2017**, *16*, 1019–1048. [CrossRef]
14. Pourbafrani, M.; Vasudevan, S.; Zafar, F.; Xingran, Y.; Singh, R.; van der Aalst, W.M. A python extension to simulate petri nets in process mining. *arXiv* **2021**, arXiv:2102.08774.
15. Qiao, D.W.; Guo, S.T.; He, J.; Zhu, Y.D. Edge intelligence: Research progress and challenges. *Radio Commun. Technol.* **2022**, *48*, 34–45.
16. Zhou, Z.; Yu, S.; Chen, X. Edge intelligence: A new nexus of edge computing and artificial intelligence. *Big Data Res.* **2019**, *5*, 53–63.
17. Li, E.; Zhou, Z.; Chen, X. Edge intelligence: Edge computing driven deep learning acceleration technolohy. *Autom. Panor.* **2019**, *1*, 48–52.
18. Xue, F. *Research on Neural Network Computing Method for Edge Intelligence*; Beijing Jiaotong University: Beijing, China, 2020.
19. Yang, K.; Ding, X.L.; Liu, J.P. Research and application of smart edge computing in IoT. *Telecommun. Sci.* **2019**, *35*, 176–184.
20. Zhang, X.; Tong, X.P.; Zhang, Y.; Wang, P. Intelligent edge computing for the Internet of Things. *AI-View* **2019**, *1*, 28–37.
21. Li, B.; Hou, P.; Niu, L.; Wu, H.; Ding, H.W. Survey of cloud-edge collaborative architecture research based on software defined network. *Comput. Eng. Sci.* **2021**, *43*, 242–257.
22. Rafique, W.; Qi, L.; Yaqoob, I.; Imran, M.; Rasool, R.U.; Dou, W. Complementing IoT services through software defined networking and edge computing: A comprehensive survey. *IEEE Commun. Surv. Tutorials* **2020**, *22*, 1761–1804. [CrossRef]
23. Wang, Q.C.; Jin, G.S.; Li, Q.; Wang, K.; Yang, Z.Y.; Wang, H. Industrial edge computing: Vision and challenges. *Inf. Control* **2021**, *50*, 257–274.
24. Song, H.Z. Industrial edge intelligence. *Autom. Panor.* **2021**, *38*, 62–66.
25. Ruan, Z.P.; She, W.K.; Li, K.; Zhou, P. Research of the edge intelligent device management method based on KubeEdge. *Electr. Power Inf. Commun. Technol.* **2020**, *18*, 63–68.
26. OpenYurt. Available online: https://openyurt.io/zh/ (accessed on 3 September 2022).
27. Wiśniewski, R.; Karatkevich, A.; Wojnakowski, M. Decomposition of distributed edge systems based on the Petri nets and linear algebra technique. *J. Syst. Archit.* **2019**, *96*, 20–31. [CrossRef]
28. Carvalho, D.; Rodrigues, L.; Endo, P.T.; Kosta, S.; Silva, F.A. Mobile edge computing performance evaluation using stochastic petri nets.In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–6.

29. Al Ridhawi, I.; Aloqaily, M.; Boukerche, A.; Jaraweh, Y. A blockchain-based decentralized composition solution for IoT services. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 July 2020; pp. 1–6.
30. Al Ridhawi, I.; Aloqaily, M.; Kotb, Y.; Al Ridhawi, Y.; Jararweh, Y. A collaborative mobile edge computing and user solution for service composition in 5G systems. *Trans. Emerg. Telecommun. Technol.* **2018**, *29*, E3446. [CrossRef]
31. Huang, J.; Liang, J.; Ali, S. A simulation-based optimization approach for reliability-aware service composition in edge computing. *IEEE Access* **2020**, *8*, 50355–50366. [CrossRef]
32. Hayyolalam, V.; Otoum, S.; Özkasap, Ö. Dynamic QoS/QoE-aware reliable service composition framework for edge intelligence. *Clust. Comput.* **2022**, *25*, 1695–1713. [CrossRef] [PubMed]
33. Van der Aalst, W.; Weijters, T.; Maruster, L. Workflow mining: Discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1128–1142. [CrossRef]
34. Van Der Aalst, W.; Van Hee, K.M.; van Hee, K. *Workflow Management: Models, Methods, and Systems*; MIT Press: Cambridge, MA, USA, 2004.
35. Sayal, M.; Casati, F.; Dayal, U.; Shan, M.C. Business process cockpit VLDB'02. In Proceedings of the 28th International Conference on Very Large Databases, Hong Kong, China, 20–23 August 2002; pp. 880–883.
36. Ehrenfeucht, A.; Rozenberg, G. Partial (set) 2-structures. *Acta Inform.* **1990**, *27*, 343–368. [CrossRef]
37. Van der Aalst, W.M.P. The application of Petri nets to workflow management. *J. Circuits Syst. Comput.* **1998**, *8*, 21–66. [CrossRef]
38. Conforti, R.; Dumas, M.; García-Bañuelos, L.; Rosa, M.L. Beyond tasks and gateways: Discovering BPMN models with subprocesses, boundary events and activity markers. In Proceedings of the International Conference on Business Process Management, Haifa, Israel, 7–11 September 2014; Springer: Cham, Switzerland, 2014; pp. 101–117.
39. Park, G.; van der Aalst, W.M. A general framework for action-oriented process mining. In Proceedings of the International Conference on Business Process Management, Seville, Spain, 13–18 September 2020; Springer: Cham, Switzerland, 2020; pp. 206–218.
40. Ramezani, E.; Fahland, D.; van der Aalst, W.M. Where did I misbehave? Diagnostic information in compliance checking. In Proceedings of the International Conference on Business Process Management, Tallinn, Estonia, 3–6 September 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 262–278.
41. The Biggest Challenge of IIoT Implementation—Industrial Internet of Things Software Platform and Edge Intelligence. Available online: https://www.zhuanzhi.ai/document/5d65d6eb47571c24809f6948e29e5085 (accessed on 3 September 2022).
42. ADVANTECH. WISE-PaaS IIoT Cloud Platform Architecture and Service. 2021. Available online: https://docs.wise-paas.advantech.com/en (accessed on 3 September 2022).