UNICO I+D Project
6G-DATADRIVEN-02

6G-DATADRIVEN-02-E20

Initial design of orchestration tools

Abstract

Industrial scenarios present myriad wireless networking and device capacity challenges. In order to ensure the effective deployment and life-cycle management of network applications for industry 4.0, orchestration tools that are fit for purpose are required. In this document we highlight the most significant of these challenges and how they can be addressed. We then propose a design that leverages the state of the art in compute and network virtualization to realize a robust orchestrator for the connected industry.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia

Document properties

Document number	6G-DATADRIVEN-02-E20
Document title	Initial design of orchestration tools
Document responsible	Carlos J. Bernardos (UC3M)
Document editor	Constantine Ayimba, Carlos J. Bernardos (UC3M)
Editorial team	Constantine Ayimba, Carlos J. Bernardos (UC3M)
Target dissemination level	Public
Status of the document	Final
Version	1.0
Delivery date	31/12/2023
Actual delivery date	31/12/2023

Production properties

Reviewers	Antonio de la Oliva (UC3M)

Disclaimer

This document has been produced in the context of the 6G-DATADRIVEN Project. The research leading to these results has received funding from the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union-NextGenerationEU through the UNICO 5G I+D programme.

Contents

List of Figures 4
List of Acronyms 5
Resumen Ejecutivo 6
Executive Summary 7
1. Introduction 8
2. State of the art in orchestration for Industry 4.0 10
3. Orchestrator design considerations for Industry 4.0 11
3.1. Diverse device technologies 11
3.2. Process and service interaction 11
3.3. Operational security 12
3.4. Realtime monitoring 12
3.5. Autonomous operation 12
4. Initial orchestrator design 13
4.1. Service Discovery 13
4.2. Virtualization 14
4.3. Virtual Infrastructure Management 14
4.4. Intelligent placement algorithms 15
5. Managed Resources 16
5.1. Compute resources 16
5.2. Network resources 16
6. Summary and conclusions 18
7. References 19

List of Figures

Figure 1: Orchestration tools schema 13

List of Acronyms

AI/ML:	Artificial Intelligence / Machine Learning
API	Application Programming Interface
BPMN:	Business Process Model and Notation
CNF:	Cloud-Native Network Function
CPU:	Central Processing Unit
DNS:	Domain Name System
GPU:	Graphics Processing Unit
IIoT:	Industrial Internet of Things
MANO:	Management and Orchestration
NFV:	Network Function Virtualization
NPN	Non-Public Network
PNI-NPN:	Public Network Integrated NPN
QoS:	Quality of Service
SFC:	Service Function Chain
TLS:	Transport Layer Security
TSN:	Time Sensitive Network
VIM:	Virtual Infrastructure Manager
VNF:	Virtual Network Function
VPN:	Virtual Private Network
VLAN:	Virtual Local Area Network
VPN:	Virtual Private Network
ZSM	Zero Touch Management

Resumen Ejecutivo

Este documento presenta un diseño inicial de un orquestador que sirve a la industria conectada en el contexto del proyecto 6G-DATADRIVEN-02. Se aprovecha de los principios de diseño más significativos en el estado del arte y propone un enfoque modular que hace que la herramienta de orquestación sea adaptable a la futura Industria 4.0. Las principales contribuciones en este entregable son:

- una revisión concisa del estado del arte en orquestación para la Industria 4.0;
- un análisis exhaustivo de los desafíos de la orquestación en un entorno industrial;
- un diseño modular de un marco de orquestación integral capaz de gestionar tanto recursos informáticos como de red.

Nuestro diseño allana el camino para la gestión autónoma sin intervención (ZSM) de la industria conectada mediante la explotación de herramientas que exponen APIs abiertas que pueden integrarse en bucles cerrados coordinados.

El resto del documento está redactado en inglés, de cara a maximizar el impacto del trabajo realizado en este proyecto.

Executive Summary

This document provides an initial design of an orchestrator that serves connected industry in the context of the 6G-DATADRIVEN-02 project. It leverages the most significant design principles in the state of the art and proposes a modular approach that make the orchestration tool adaptable to future Industry 4.0. The key contributions in this deliverable are:

- a concise review of the state of the art in orchestration for Industry 4.0;
- a thorough analysis of the challenges of orchestration in an industrial environment;
- a modular design of a holistic orchestration framework capable of managing both compute and network resources.

Our design paves the way for the autonomous Zero Touch Management (ZSM) of the connected industry by exploiting tools exposing open APIs that can be integrated to coordinated closed loops.

1. Introduction

Standard orchestration solutions are broadly based on the ETSI NFV MANO [1] framework. They address the challenge of integrating edge resources by extending MANO components, data models, and procedures. Specialized Virtualized Infrastructure Managers (VIMs) are introduced to handle edge infrastructures and their virtualization techniques, all still centrally controlled by an orchestrator. Adapting to edge environments involves expanding traditional network service models to incorporate Cloud-native Network Functions (CNFs). The latter are akin to legacy Virtual Network Functions (VNFs), statically declared in the Service Function Chain (SFC) and interconnected from a networking-focused perspective rather than a more flexible service-based approach. Placement algorithms typically regard edge resources as a unique type of constrained cloud resources located close to the network edge.

With regard to connected industry, these resources constitute a particular form of Industrial IoT (IIoT) presenting myriad orchestration challenges including:

- *Heterogeneity*: Industrial networks comprise diverse devices and protocols from different vendors.
- *strict latency constraints*: Many industrial processes require real-time monitoring and control. Administering network applications in IIoT must ensure low-latency communication and timely data processing to meet the demands of real-time applications.
- *Tight security requirements*: IIoT networks are attractive targets for cyber threats due to their critical nature. Administering network applications must prioritize robust security measures, including data encryption, authentication, access control, and intrusion detection, to protect against cyber threats.
- *Tough wireless communication environment*: Factory floors are replete with machines, moving elements and reflective surfaces. As a consequence, wireless signals experience blockage, scattering and significant Doppler shifts that hamper reliable wireless communication.

Achieving comprehensive orchestration continuity at the extreme edge management level during the initial phase may pose challenges. Service models must incorporate support for resources at the extreme edge, and the orchestration framework should be compatible with the serverless paradigm. It is essential to expose the capabilities of hardware acceleration to services to facilitate flexible function placement, especially for high-performance requirements. A technology-agnostic approach to resource orchestration for Network Functions (NFs) is necessary, accompanied by information and data models that support comprehensive lifecycle management.

In order to adequately address these challenges, the design of orchestration tools for Industry 4.0 will need to employ the following principles:

- *Standard protocols*: This tackles the challenges of heterogeneity

[1] https://www.packetcoders.io/what-is-etsi-mano/

- *Lightweight virtualization*: This is particularly important considering the constraints on the edge and extreme edge devices as well as the stringent low latency requirements.
- *Simple data models*: Given the non-ideal communication environment, low data rates are preferable to ensure reliable communication. As a consequence, packets need to be small with only the essential minimal data to run the service.
- *Encrypted communication*: Transport Layer Security (TLS) and such like protocols should be supported by all services and the corresponding orchestrators for connected industry.

2. State of the art in orchestration for Industry 4.0

In order to address the challenges presented by IoT, several approaches leverage the serverless computing paradigm. (Du, et al., 2020) propose Catalyzer, a sandbox solution providing both low-latency application deployment applications and strong security. Key to the performance of their solution is the use of a checkpoint image to restore a container thereby avoiding the long instantiation delay.

The authors of (Guo, et al., 2022) propose a modified lightweight container that is suitable for dense deployments. It leverages *Kata* [2]Runtime and implements *rootfs* [3]with read/write splitting resulting in sub-second instantiation time.

Workflow managers which orchestrate industry processes using a micro-service-based architecture have also gained prominence in Industry 4.0. In (Larrinaga, et al., 2022), the authors propose an orchestration mechanism suitable for deployment in embedded systems. It achieves the latter by exploiting BPMN [4]recipes and Node.js which is not resource hungry.

(Laskaratos, et al., 2022) propose MESON, which extends the ETSI NFV MANO framework creating an orchestrator that enables cross slice communication. In this way the orchestrator is made aware of services running on other slices and can improve its placement mechanism by choosing the most suitable edge resources which are not unduly occupied.

Another promising approach to orchestration for Industry 4.0 is the one proposed in (Wang, et al., 2022) which exploits TSN in the link layer to achieve reliable runtimes for network services. It uses collaborative intelligence at the edge network to orchestrate micro-service-based applications. In this way it exploits distributed resources to achieve low latencies while guaranteeing reliable performance.

[2] https://katacontainers.io/
[3] https://refspecs.linuxfoundation.org/FHS_3.0/fhs/ch03.html
[4] https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation

3. Orchestrator design considerations for Industry 4.0

Beyond the considerations of resource constraints at the extreme edge where devices such as robotic arms and production lines operate, orchestration tools for Industry 4.0 network services must take into account myriad considerations. Some of these are:

1. Diverse device technologies:
2. Process and service interaction
3. Operational security
4. Realtime monitoring
5. Autonomous operation

In what follows we look into each of these in more detail

3.1. Diverse device technologies

To support diverse services effectively, a model-based service specification is crucial. This includes enabling multi-connectivity for the devices by integrating fixed and wireless access technologies. This integration enhances performance and introduces new functionalities, such as reducing latency through parallel channel access. For the orchestrator to work effectively, optimizing interface selection for integrated access networks is essential. The current static translation between communication service definitions and network descriptors needs to evolve into a dynamic process, accounting for infrastructure specifics. This dynamic translation would automatically adapt to the deployment environment, customizing service instantiation based on varying network capabilities, resource availability, and virtualization technologies.

Given the variability of capabilities in different environments and the dynamic nature of service runtime, particularly in the volatile extreme edge resources, supporting mechanisms for dynamic and data-driven translation of service descriptors is crucial. This ensures automatic readaptation and optimization of service deployment, enhancing Quality of Service (QoS) across diverse and potentially changing network infrastructures

3.2. Process and service interaction

The tight coupling of the manufacturing process and the control of the corresponding factory floor machinery means that any unexpected misalignment can have grave consequences. An apt orchestrator operating in connected industry environments must be TSN capable and ensure that commands for the deployment and placement of services are executed with timely precision.

3.3. Operational security

In the context of orchestration, it is imperative to flexibly combine both internal and external functionalities of an enterprise to create use case-specific services tailored to various service provision scenarios. Efficient management of private networks by personnel lacking expertise in telecommunications requires the orchestration of private networks through abstractions relevant to the application domain. In the transition to B5G/6G, there is a focus on achieving higher levels of automation for the seamless integration of external capabilities into private networks. The development of management APIs is essential to empower independent developers to introduce novel solutions in the context of PNI-NPN scenarios. Orchestrators should implement these APIs, allowing Non-Public Network (NPN) owners to effectively oversee their networks. This oversight includes provisioning, fault supervision, and ensuring performance assurance through secure and auditable mechanisms.

3.4. Realtime monitoring

We envision that B5G/6G-specific service orchestration and platforms must facilitate data-driven orchestration driven by automated data-driven decisions. To realize this, management and orchestration systems in B5G/6G should not only provide access to metrics from the infrastructure, as commonly seen in current orchestration systems (such as VNFs' CPU and RAM metrics or network usage metrics) but also from the data plane, encompassing metrics of the deployed services. For example, in a B5G/6G orchestration application, one could correlate conventional infrastructure metrics with an image recognition system within the network service. To enable this functionality, appropriate interfaces must be activated to grant access to the specific data of each network service

Aside from aiding in the detection of network faults and addressing issues promptly, this advanced monitoring system would also contribute to supplying essential data for training AI/ML models. It also enables the proactive identification, diagnosis, and resolution of service quality degradations before impacting end-users.

3.5. Autonomous operation

Given the inevitable complexity of Industrial IoT, orchestration systems capable of full closed loop autonomy will be less error-prone compared to those reliant on human control. To this end, the tools so developed should be driven by AI. Specifically, a comprehensive AI orchestration system is required to oversee AI/ML services, facilitating collaborative approaches among distributed agents. This system should not only facilitate interactions among agents within the same federated services but also potentially among multiple services. Additionally, it should effectively manage communication resources for these services and feature APIs that can be used to integrate to pertinent smart manufacturing components.

4. Initial orchestrator design

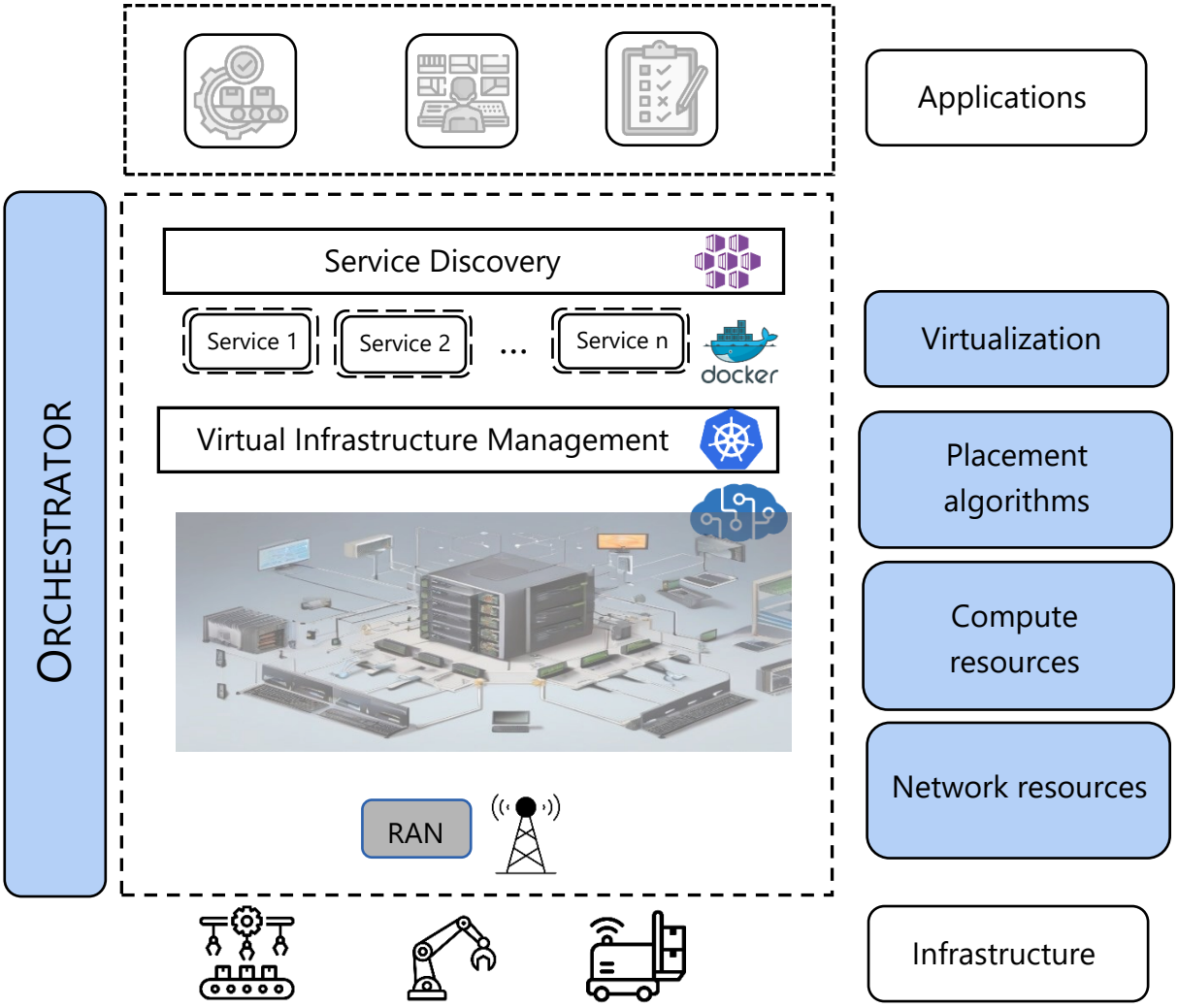

FIGURE 1: ORCHESTRATION TOOLS SCHEMA

Figure 1 shows the design layout of the orchestrator highlighting the tools required to effectively provision services in the connected industry. In what follows we delve into the roles of each and consider technologies that can be used to realize the functions.

4.1. Service Discovery

Service discovery plays a crucial role in application deployment by automatically identifying and providing information about available services within a network, enabling seamless communication and integration between different components and ensuring dynamic scalability and efficient resource utilization. Common protocols and tools used for service discovery include DNS-based

Service Discovery (DNS-SD), mDNS (Multicast DNS), *etcd*[5], *Consul*[6], and *Kubernetes*' [7]service discovery mechanisms. In essence, service discovery simplifies the complexity of managing distributed systems by providing a dynamic and automated way for services to find and communicate with each other in a networked environment.

4.2. Virtualization

Virtualization in microservice deployment provides isolation and resource efficiency by enabling the creation of lightweight, independent containers or virtual machines for each microservice, facilitating easier scaling, management, and deployment, while minimizing dependencies and conflicts between microservices. Once services are exposed, the orchestrator works to encapsulate them in containers or other such lightweight virtual elements. The latter form inter-dependent service function chains that work together to deliver the application. To achieve the lightweight virtualization required to deploy industry 4.0 services, such tools as *Docker* [8]or *LXC* [9]containers can be used. Some level of customization might be required to optimize their operation given the latency sensitive nature of industrial applications.

4.3. Virtual Infrastructure Management

Virtual Infrastructure Managers (VIMs) are responsible for managing and controlling virtualized resources within a data centre. They work by interfacing with the underlying physical infrastructure and providing a layer of abstraction to virtualized environments. VIMs allocate compute, storage, and network resources to the containers or virtual machines that constitute a Service Function Chain. The main functions of VIMs are:

- *Resource Abstraction*: VIMs create an abstraction layer between the physical hardware and the virtualized resources. This abstraction allows users or higher-level management systems to interact with virtual resources without directly dealing with the complexities of the underlying physical infrastructure.
- *Resource allocation*: VIMs are responsible for allocating and managing virtual resources, including computing power (CPU and memory), storage, and network resources. They ensure efficient distribution and utilization of these resources among various virtual machines or containers.

[5] https://etcd.io/
[6] https://www.consul.io
[7] https://kubernetes.io/
[8] https://www.docker.com/
[9] https://linuxcontainers.org/

- *Hypervisor communication*: In virtualized environments, VIMs often interact with hypervisors, which are responsible for creating and managing virtual machines. The VIM communicates with the hypervisor to deploy, monitor, and control virtual instances on the physical hardware.
- *Resource Monitoring*: VIMs continuously monitor the performance and health of virtualized resources. This involves tracking resource usage, identifying bottlenecks, and ensuring that virtual machines or containers receive the necessary resources to operate optimally.
- *Security and Isolation:* They security enforce isolation between different virtual instances thereby providing a level of security for the virtualized environments. They ensure that resources are partitioned securely and that one container/virtual machine cannot compromise the integrity or performance of others.

Kubernetes is a popular VIM that works especially well with Docker containers encapsulated in pods ensuring that an orchestrated service keeps running despite the failure(s) of containerised instances.

4.4. Intelligent placement algorithms

Most commercial VIMs carry out scaling and load balancing operations based on resource utilisation thresholds. Such operations are by thereby reactive to prevailing conditions and might lead to unwanted delays which are detrimental to latency sensitive services. Incorporating data driven AI algorithms has been shown to greatly improve the performance of orchestrators by carrying out pre-emptive migrations before the failure of a virtualized instance or scaling a service to multiple nodes before existing resources are saturated. These algorithms exploit the data obtained by the monitoring capabilities provided by VIMS to train AI agents that learn to predict traffic patterns and leverage these to inform the placement mechanisms of VIMs.

5. Managed Resources

The distributed nature of applications for connected industry requires that parts of the software are hosted on the network edge servers. The latter provide compute, storage and memory resources to the server-side application. The orchestrator is also responsible for ensuring that network resources such as radio channels are availed to allow for client-side applications residing in the Industrial IoT devices can communicate to their server-side counterparts. The variable nature of the wireless channel means that the orchestrator must constantly revise modulation scheme allocations or the number of time-frequency radio units allocated to a running application to maintain reliable service times. Aside from these radio access network resources are the data network resources detailed below.

5.1. Compute resources

By exploiting virtualization of the physical infrastructure, the VIM can more effectively ensure that resources are fairly allocated to network services. Some of the main resources managed by the orchestrator are:

- *Virtual CPU/GPU*: VIMs allocate the requisite number of virtual CPU or GPUs needed to a container or VM depending on the type of service that each run.
- *Memory*: The amount of memory required by a container or VM is also in the purview of an orchestrator.
- *Virtual images*: In order to instantiate a pre-defined network function or service, the orchestrator maintains the template of the service as an image to allow for quick deployment.
- *Network interfaces*: Virtual network interfaces connect containers/VMs to virtual networks, enabling communication between them and external networks. Virtual switches and routers are often used to manage virtualized network resources.

5.2. Network resources

Network orchestration involves the coordinated management and automation of various network resources to ensure efficient and scalable network operations. The key network functions that are managed by the orchestrator are:

- *Switches and Routers*: Network orchestration can automate the configuration and management of switches and routers, ensuring proper routing, VLAN configurations, and Quality of Service (QoS) settings.
- *Firewalls*: Orchestration enables the automated deployment and configuration of firewalls to control and monitor network traffic, implement security policies, and protect against unauthorized access.
- *Load Balancers*: Orchestration can automate the provisioning and configuration of load balancers to distribute incoming network traffic across multiple servers, ensuring optimal resource utilization and high availability.

- *Virtual Private Networks (VPNs)***:** Network orchestration facilitates the automated setup and management of VPN connections, allowing secure communication over public networks.
- *IPv4 and IPv6 Addressing***:** Orchestration tools can automate the assignment and management of IP addresses, ensuring proper addressing schemes and avoiding conflicts.

6. Summary and conclusions

In this deliverable, we have presented the key principles to consider in the design of orchestration tools for the connected industry. We have given a brief overview of the most recent and pertinent state of the art and considered how each addresses the challenges of orchestration in industrial environments. Finally, we have presented our proposed design of an articulated orchestrator with several interacting components.

7. References

Du, D., Yu, T., Xia, Y., Zang, B., Yan, G., Qin, C., & Chen., Q. W. (2020). Catalyzer: Sub-millisecond Startup for Serverless Computing with Initialization-less Booting. *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '20)*, 467–481.

Guo, Z. L., Cheng, J., Chen, Q., Guan, E., Bian, Z., Tao, Y., . . . Minyi, W. H. (2022). RunD: A Lightweight Secure Container Runtime for High-density Deployment and High-concurrency Startup in Serverless Computing. *2022 USENIX Annual Technical Conference (USENIX ATC 22)*, 53-68.

Larrinaga, F., Ochoa, W., Perez, A., Cuenca, J., Illarramendi, & M., J. L. (2022). Node-RED Workflow Manager for Edge Service Orchestration. *IEEE/IFIP Network Operations and Management Symposium*. Budapest: IEEE.

Laskaratos, D., Dimolitsas, I., Papathanail, G., Xezonaki, M.-E., Pentelas, A., Theodorou, V., . . . Papavassiliou, S. (2022). MESON: A Platform for Optimized Cross-Slice Communication on Edge Computing Infrastructures. *IEEE Access*, 49322-49336.

Wang, Y., Yang, S., Ren, X., Zhao, P., Zhao, C., & Yang, X. (2022). IndustEdge: A Time-Sensitive Networking Enabled Edge-Cloud Collaborative Intelligent Platform for Smart Industry. *IEEE Transactions on Industrial Informatics*, 2386-2398.