SLICES, a scientific instrument for the networking community

Serge Fdida [a],[*], Nikos Makris [b], Thanasis Korakis [b], Raffaele Bruno [c], Andrea Passarella [c], Panayiotis Andreou [d], Bartosz Belter [e], Cédric Crettaz [f], Walid Dabbous [g], Yuri Demchenko [h], Raymond Knopp [i]

[a] *Sorbonne Université, CNRS, LIP6, Paris, France*
[b] *University of Thessaly, Volos, Greece*
[c] *IIT, CNR, Pisa, Italy*
[d] *UCLan Cyprus, Larnaca, Cyprus*
[e] *Poznan Supercomputing and Networking Center, Poznan, Poland*
[f] *Mandat International, Geneva, Switzerland*
[g] *Université Côte d'Azur, Inria, Sophia Antipolis, France*
[h] *University of Amsterdam, Amsterdam, The Netherlands*
[i] *EURECOM, Communication Systems Dept., Sophia Antipolis, France*

ARTICLE INFO

ABSTRACT

A science is defined by a set of encyclopedic knowledge related to facts or phenomena following rules or evidenced by experimentally-driven observations. Computer Science and in particular computer networks is a relatively new scientific domain maturing over years and adopting the best practices inherited from more fundamental disciplines. The design of past, present and future networking components and architectures have been assisted, among other methods, by experimentally-driven research and in particular by the deployment of test platforms, usually named as testbeds. However, often experimentally-driven networking research used scattered methodologies, based on ad-hoc, small-sized testbeds, producing hardly repeatable results. We believe that computer networks needs to adopt a more structured methodology, supported by appropriate instruments, to produce credible experimental results supporting radical and incremental innovations. This paper reports lessons learned from the design and operation of test platforms for the scientific community dealing with digital infrastructures. We introduce the SLICES initiative as the outcome of several years of evolution of the concept of a networking test platform transformed into a scientific instrument. We address the challenges, requirements and opportunities that our community is facing to manage the full research-life cycle necessary to support a scientific methodology.

1. Introduction

Prototyping a scientific instrument to explore the design space of future and emerging digital infrastructures has often been considered as impossible or irrelevant. The reasons are manifold, (i) it is hard to predict the future landscape and challenging scientific questions, (ii) the technology is evolving too quickly, and (iii) the community is fragmented. Unfortunately, the recent COVID episode demonstrated that providing evidence is a difficult task, often grounded on experimental research, and that this process is long and complex but timely and absolutely necessary.

Up to now, networking test platforms have tried to capture a variety of demands. Academia represents the first target group, necessitating tools to support the validity of the assumptions and provenance of results and data published in scientific papers. However, very little has been done to cover the entire research and data lineage lifecycle to ensure the longevity of such data as well as its access to the wider research and innovation community. It is for that reason that the FAIR (Findable, Accessible, Interoperable, and Reusable) and Open Science Principles were developed promoting interoperability and reproducibility of the results. Industry forms a second target group that often emphasizes the value of solutions to support conformance or interoperability testing, or the importance to make test platforms available for SMEs or startup companies, because otherwise they will never have the opportunity to access such instruments. As a consequence, different target groups impose unique requirements and expectations that need to be addressed by current and future testbeds.

The field has matured quite a lot over the last decades. The first phase of test platforms can be illustrated by facilities such as PlanetLab[1] and Orbit.[2] In 2005, the concept of testbed federation was introduced and applied to PlanetLab with the deployment of PlanetLab Europe in 2007 [1]. Since then, this concept has developed quite a lot [2,3]. Orbit's success [4] has been due to addressing the need for realistic environments to test wireless protocols that were becoming essentials. The second phase of test platforms was initiated in 2007 with the ambition of the NSF GENI [5] (120M\$ 2008/2016) and EU FIRE [6] initiatives (200M€ 2007/2022). This corresponds somehow to a more structured approach to build a research infrastructure for this domain. GENI's approach was meant to design a nationwide test platform, composed of GENI Nodes and racks that the experimenters could program. In Europe, the ambition was to federate testbeds with very heterogeneous resources. Both initiatives were nicely articulated and produced SFA [7], the Slice-based Federation Architecture, proposing a practical solution to federate the facilities managed by independent authorities. The third phase already started with initiatives such as NSF PAWR in the US [8], CENI in China and ICT 17/19/52 in Europe [9–11]. Those are developed in parallel without much cooperation and collaboration at present. The novelty comes from two new types of stakeholders. The tech giants are developing their own facilities (experimental or production), which is providing a risk with respect to the competition with academic research as these private platforms are not open, neither are the data that they use to produce their results. Second, other initiatives have emerged supported by the open-source community, such as ONAP [12], ORAN [13], OpenAirInterface [14] enabling new and unique opportunities to deploy fully programmable and virtualised network infrastructures.

As a scientific community, our first message is that we have to continue to raise global awareness and promote the importance of a scientific instrument, because it is a community responsibility. All the efforts highlighted above address the demand for the networking field, including the demand for the experimental validation of research results. This experimental validation constitutes a cornerstone of any sound scientific methodology. Both for historical and practical reasons, though, experimentally-driven research in the networking field has been so far quite fragmented, characterized by the development of "in-house" testbeds, with the purpose of validating specific innovations (of specific research groups). This has (i) limited the scale at which experiments can be executed, (ii) limited the reproducibility of results (another pillar of scientific research) due to customizations at the individual testbed level, and thus (iii) limited the credibility of results produced. Because digital infrastructures are rapidly becoming a fundamental technological basis of our society, we think this gap needs to be urgently filled, such that next generation networks (starting from beyond-5G and 6G) can be developed based on reference, large-scale experimental infrastructures that could act as reference point for the wide community of computer networks and distributed systems researchers.

In Europe, there exists a framework called ESFRI — European Strategy Forum on Research Infrastructures [15] that supports the design, implementation and operation of scientific instruments. ESFRI is fully driven by science and organized in phases that should guarantee the feasibility and sustainability of the instrument. This framework is used by all scientific domains with a similar vision about the objectives and methodology. Targeting a scientific instrument for our community means that we have to align and adopt the principles promoted by ESFRI. It starts with a clear statement about the scientific question that this particular instrument will address. For instance, does the

Higgs boson exist? Formulating such a question, which is easily understandable by other disciplines, the stakeholders and citizens, is not straightforward in our domain.

In this paper, we present the SLICES ESFRI initiative that is meant to support the discovery process related to the future, emerging digital infrastructures. In Section 2, we focus on the ESFRI framework defining the requirements to enter into the roadmap. It relates to the ability and value of the future facility as well its sustainability. Section 3 highlights some of the design issues that are still being debated. We illustrate these foundations with the example of a 5G network in Section 4. From this analysis, we derive preliminary architecture guidelines for SLICES in Section 5. The research lifecycle dimension is of utmost importance. We introduce EOSC as a valuable target to be articulated with SLICES and discuss the components of the full-research lifecycle in Section 6. The interoperability with EOSC is presented in Section 7. We illustrate this with an example borrowed from another discipline. Finally, we conclude in Section 8 and list topics for future investigations.

2. The ESFRI framework, a scientific instrument

ESFRI, the European Strategy Forum on Research Infrastructures, established in 2002, brings together national governments, the scientific community, and the European Commission, to support a coherent and strategy-led approach to policy making on Research Infrastructures (RIs) in Europe. The ESFRI Roadmap[3] contains the best European science facilities based on a thorough evaluation and selection procedure. The Strategy Report on Research Infrastructures 2021 includes the Roadmap 2021 and the ESFRI vision of the evolution of Research Infrastructures in Europe, addressing the mandates of the European Council, and identifying strategy goals. Since 2002, within the framework of ESFRI and the ESFRI Roadmap process, national governments have worked in close partnership with the European Commission and the scientific community to catalyze the establishment of over 50 European Research Infrastructures, mobilizing investments of approximately €20 billion across the EU.

ESFRI applies a lifecycle approach to the development and implementation of RIs as presented in Fig. 1. The lifecycle concept describes the different milestones in the development, implementation and operation of a Research Infrastructure over time, specifying minimal key requirements that must be met to enter each stage. Application of this concept allows for a coherent assessment of the scientific and organizational maturity of Research Infrastructures across all fields of science.

The concept of a new RI typically emerges bottom-up from the scientific communities clustering around well identified scientific needs and goals. For RIs to remain relevant throughout the entire RI lifecycle, scientific excellence is the *conditio sine qua non*, which becomes, together with adequate human resources, crucial when it comes to long-term persistence in the operational phase. Effective governance and sustainable long-term funding (public and private) are other key elements for ensuring long-term sustainability of RIs at every stage in their lifecycle.

We observe that until 2018, ESFRI was organized in 5 WGs related to energy, environment, health and food, physical sciences and engineering, social and cultural innovation. We had to wait until 2018 to applaud the creation of a working group dealing with Data, Computing and Digital Research Infrastructures that clearly differentiates digital *research* infrastructures (i.e., addressing the needs of the *digital research* communities) from *production* research e-infrastructures defined as ICT support to other sciences.

SLICES is a distributed research infrastructure. It is organized with a central node and a set of distributed nodes. The central node hosts

---

[1] PlanetLab, https://planetlab.cs.princeton.edu/.

[2] Open-Access Research Testbed for Next-Generation Wireless Networks (ORBIT), https://www.orbit-lab.org/.

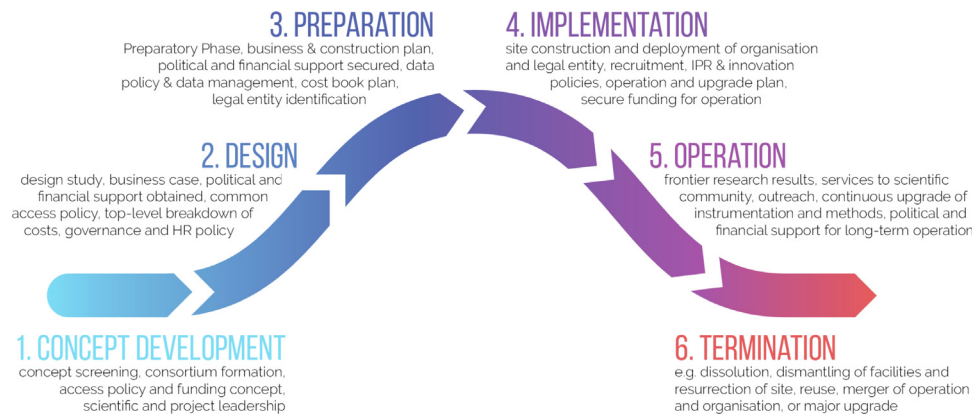[3] ESFRI Roadmap 2021, https://www.esfri.eu/esfri-roadmap-2021.

Fig. 1. ESFRI lifecycle approach.

the governance and resources needed to steer and run the facility. As a European facility, the distributed nodes are hosted by different member states, covering different types of needs. At the time of writing, we have 15 countries supporting the effort.[4] Industry is supporting but not directly involved in contributing to a common good.

3. Design foundation principles

The future generation of digital infrastructures is designed to be programmable, extendable and scalable. Key enabling technologies are now available to empower this important transformation. An important concept is network disaggregation whereby networking software is separated from the switching and/or routing hardware and broken down into functional components that can be more efficiently operated. Software Defined Networking (SDN) assumes programmable network devices in which the forwarding plane is decoupled from the control plane. In addition, the control plane is logically centralized in a software-based controller ("network brain"), while the data plane is composed of network devices ("network arms") that forward packets. Network Function Virtualization (NFV) will deliver the promises of a software framework to the network, creating the need for an efficient and effective orchestration of the network resources.

The primary technologies and solutions to address the requirements for the SLICES facility can be classified as described in the next subsections.

3.1. Software defined network and network function virtualization

Following the major evolution of telecommunications networks with the adoption of the internet technology and the emergence of cellular networks, we are now facing a paradigm shift in the way that digital infrastructures are designed and operated. Indeed, recent advances in networking such as SDN and NFV [16] are changing the way network operators deploy and manage Internet services. SDN and NFV, together or separately, bring to network operators new opportunities for reducing costs, enhancing network flexibility and scalability, and shortening the time-to-market of new applications and services. On the one hand, SDN introduces a logically centralized controller with a global view of the network state. On the other hand, NFV allows to fully decouple network functions from proprietary appliances and to run them as software applications on general-purpose machines. It is a scalable approach as it gives the operators the ability to scale their network architecture across multiple servers to adapt quickly to the changing needs of their customers.

3.2. Network slicing

Another disruptive concept that should help in realizing the vision is network slicing, which allows a single physical network to be segmented into multiple isolated logical networks of varying sizes and structures tailored to different types of services and customers [17]. It is a multi-tenant virtualization technique in which the various network functionalities are extracted from the hardware and/or software components and then offered in the form of slices to the different users of the infrastructure (tenants). Basically, each slice includes a number of dedicated physical resources and network functions, which are isolated from other slices and provide specific functionalities including RAN and core network. Network slicing aims to offer operators the possibility of creating, in real-time and on-demand, various levels of services for different enterprise verticals, enabling them to customize their operations. In particular, it allows service differentiation with different QoS levels, reliability and security. Network slicing requires a continuous reconciliation of customer-centric service level agreements (SLAs) with infrastructure-level network performance capabilities. However, one of the main issues to solve is how to meet the requirements of different network services programmed from a single physical infrastructure. Autonomic (AI-empowered) and self-optimized management is needed to dynamically create, scale down or up, and reconfigure according to application demands [18,19]. It is important to remark that the slicing concept is one of the key innovations in 5G since Release 15, supported by NFV and SDN techniques. Gradually, this concept is evolving from supporting primarily core-network resource provisioning, to more advanced (and forward-looking) paradigms, whereby also edge and far-edge devices' resources can be virtualised and provided as dynamic, on-demand components of the network infrastructure [20,21]. However, an end-to-end network slice composed of sub-slices that belong to different technological domains (RAN, core, edge/cloud), requires hierarchical and distributed management solutions to cope with the heterogeneity of the orchestration systems of different technological domains [22,23].

3.3. Network disaggregation

Legacy aggregated networking devices have been developed and commercialized by vendors for decades. The term aggregation refers here to the vertical integration of software and specialized hardware components, bundled into a proprietary networking device. Network device disaggregation is the ability to source switching hardware and network operating systems separately. The term white box switches refer to switches built on commodity hardware that run different possible Network Operating Systems (NOS). This approach is putting pressure on the legacy aggregated networking vendors, but requires talented developers to build and grow the solution. This concept has

---

been extended to the radio access network: RAN disaggregation [24,25] was specified by 3GPP [26] and detailed by the Open Networking Foundation (ONF) [27] as an important step allowing for dynamic creation and lifecycle management of use-case optimized network slices. The idea here is to split the RAN protocol stack so that the individual components can be developed independently by different vendors. This horizontal disaggregation also enables distributed deployment of RAN functions in the network.

3.4. Distributed platform

All the aforementioned techniques SDN/NFV, network slicing and disaggregation can be combined in a distributed platform to test advanced networking scenarios in realistic large-scale environments. This could be done by leveraging virtualized computing and networking resources in a flexible way to provide support for solutions based on the use-case, geography and experimenter choice. In such a distributed platform, the functions of the RAN nodes (the base stations) may be deployed as a "Central Unit", centralizing the packet processing functions and executed as Virtual Network Functions (VNFs) on commodity hardware in edge cloud locations. One or more "distributed units" performing the baseband processing functions as VNFs on commodity hardware with possible hardware acceleration and several "radio units" running the radio functions with specialized hardware on antenna sites. In a more general setting, different functions can be deployed on different sites in the network in order to realize the required flexibility and assess performance of the different split options.

3.5. Control and user-plane separation

Another vertical disaggregation consists in the separation of Control and User Planes (CUPS) [28]. In fact, with the densification of the next generation radio access networks, and the availability of different spectrum bands, it is more and more difficult to optimally allocate radio resources, perform handovers, manage interfaces, and balance load between cells. It is therefore necessary to adopt centralized control of the access network in order to increase system performance. This approach can be realized by decoupling the intelligence from the underlying hardware in all parts of the network.

3.6. Research data management

SLICES wants to fully endorse and adopt the Open Science and FAIR principles, acting as a catalyst to enable and foster cutting edge research, data-driven science and scientific data-sharing. Several design considerations should be taken into consideration, including: (i) *easy and open access* to scientific data to facilitate further knowledge discovery and research transparency ensuring the longevity of the data and access to the wider research and innovation community. (ii) *scalable architecture* to efficiently leverage a large number of storage resources to support efficient data storage and compute, including highly parallelized data workflows to support experiments; (iii) *privacy preservation* methods for ensuring end-to-end security and privacy in compliance with relevant legal frameworks; and (iv) *data quality assurance* methods to ensure data quality across multiple dimensions, such as accuracy, completeness and integrity, in order to improve data utility. To address this, SLICES requires to carefully design and develop efficient and scalable data management, analysis and reporting mechanisms, supported by appropriate metadata profiles to cater for access and reuse of FAIR data and services. These tools need to capture and report the entire data lineage/provenance across the data management lifecycle, while also providing the systematic means for secure and trustworthy interoperability of data and services ensuring the authenticity and immutability of the shared data.

4. Illustration with 5G

As an illustration, we are presenting how these concepts are transforming the design and operation of cellular networks. The evolution of 5G networks introduces architectural changes in the Radio Access Network (RAN) [29] and the Core Network (CN) [30] that will have adverse impact on how research infrastructure testbeds are designed as to support a variety of use cases.

4.1. Disaggregation of 5G RAN

The 5G NR [31] defines a fully distributed Radio Access Network (RAN), by breaking traditional radio components into Radio Units (RUs), Distributed Units (DUs), and Centralized Units (CU). On the contrary, 4G provides a small level of disaggregation between Remote Radio Units (RRUs) and the centralized Baseband Units (BBU).

5G RAN has evolved from 4G with significant improvements in capabilities and functionalities. With the usage of a wider range of carrier frequencies that includes part of millimeter wave (mmWave) frequency spectrum, and flexible frame structure with variable number of symbols per subframe, 5G NR can utilize up to 400 MHz of bandwidth per carrier. Several platforms exist that implement the 5G stack fully in software. By making use of Software Defined Radios (SDR), such platforms can turn commodity equipment (e.g., with General Purpose Processors) to fully functional base stations. The two most prominent solutions in open source to implement such functionality are: (1) the OpenAirInterface5G platform (OAI) [14], and (2) the srsRAN platform [32]. Both platforms support the basic operations for the 5G NR, though OAI has a wider user base and implements more features, such as disaggregated operation for the RAN, several different supported SDRs, etc. From an architecture perspective, 3GPP Release 15 has introduced CU/DU split (3GPP Option 2 split [26]) along with a Virtualized RAN architecture. Splitting the higher layers of 3GPP software stack (SDAP, PDCP and RRC) and lower layers (RLC, MAC and PHY) into separate logical units, known as Centralized Unit (CU), Distributed Unit (DU) and Radio Unit (RU), enables to deploy them at separate locations. Further split of gNB-CU is induced by separation between the Control Plane (CP) and User Plane (UP) named as gNB-CU-CP and gNB-CU-UP. The NG-RAN Network Resource Manager (NRM) [33] was designed to enable "separate" provisioning of CU, DU, CU-CP, CU-UP.

Possible options for decomposition of the RAN environment are studied, resulting in the identification of eight options (3GPP Options 1–8). Building on top of the different disaggregation options, and especially delving into the CP/UP separation (CUPS), Open RAN (O-RAN) architecture defines open and standardized interfaces among the different elements of the disaggregated RAN. Through the use of such standardized interfaces, interoperability of functions between different vendors is made possible, while programmability of the RAN through dedicated interfaces is enabled [34]. O-RAN Alliance is responsible for an additional split of the CU-CP into Radio Intelligence Controller (RIC) and remaining part of CU-CP. O-RAN defines the specifications for interface definitions between CU, DU, RU and RAN intelligent controller (RIC) that can be deployed at the edge of the network. Depending on the operation of the RIC and the programmable functions in the gNB, the RIC can operate in real-time mode (< 1 ms latency for programming the different functions, e.g., for Radio Resource Management) or near-real-time/non-real time mode (e.g., for the application and integration of Machine Learning models to the operation of the RAN). The OAI community works closely with the O-RAN ecosystem to ensure interoperability of key interfaces for experimentation with such disaggregated RAN topologies. Moreover, OAI aims for interoperability with upcoming O-RAN compliant radio-units to allow experimental infrastructure initiatives such as SLICES to make use of industry-grade radio solutions.

It is reasonable to presume that the information model in O-RAN, as presented in Fig. 2, will be the extension of the 3GPP NRM, with

Table 1
Open Source frameworks and projects that can be utilized to implement RAN and MEC infrastructure.

| Name	Network domain	Description	References/links
OAI [14]	RAN	eNodeB, gNodeB and UE software	https://openairinterface.org
srsLTE [32]	RAN	eNodeB, gNodeB and UE software	https://openairinterface.org/
SD-RAN [37]	RAN and Edge	Framework for RAN components and RAN intelligence controller	https://github.com/srsran/srsRAN
AETHER [38]	RAN and Edge	5G/LTE, Edge-Cloud-as-a-Service (ECaaS)	https://opennetworking.org/sd-ran/
FlexRIC [35]	RAN	Real-time controller for software-defined RAN	https://gitlab.eurecom.fr/mosaic/flexric
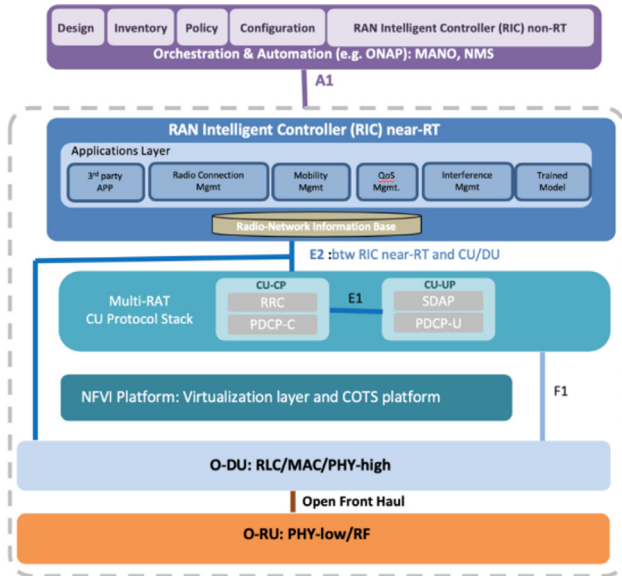
Fig. 2. Open-RAN deployment and programmable interfaces.

additional Managed Element object classes for RIC and possibly with extension of the information models for CU, CU-CP and DU. Similar to the O-RAN programmable interfaces, dedicated solutions for specific platforms exist, that open up the programmability of the RAN functions in practice. For example, the FlexRIC platform (also called as FlexRAN) [35,36], developed by Eurecom for OAI, allows the programmability of the OAI RAN in real-time, by exposing a REST interface. The interface can be used for retrieving statistics from the network as well, allowing for the advanced monitoring of the RAN in real-time. The FlexRAN controller is under further extension for becoming compatible with the O-RAN interfaces for programming the network. Similar to the FlexRAN platform, the SD-RAN platform developed by the Open Networking Foundation (ONF) is complementing O-RAN's focus on architecture and interfaces by building and trialing O-RAN compliant open-source components. SD-RAN [37] is developing a near-real-time RIC (nRT-RIC) and a set of exemplar applications that run on top (xApps) for controlling the RAN. Towards integrating all the above efforts for the end-to-end deployment of the cellular network with extended use of virtualized services, the AETHER framework is currently under development by ONF [38]. AETHER combines three main elements, namely, a control and orchestration interface to the RAN, an edge cloud platform (the AETHER edge), with support for cloud computing APIs, and a central cloud (the AETHER core), for orchestration and management. The AETHER project integrates several ONF efforts, including SD-RAN, ONOS [39], CORD [40] and OMEC [41], for providing a fully-fledged solution for the deployment of the cellular network in an end-to-end manner.

Table 1 lists out different open-source frameworks and projects that can be utilized to implement RAN and MEC infrastructure.

4.2. Disaggregation of 5G core

In 5GC, one of the most important characteristics is the separation of the User Plane (UP) functions from the Control Plane (CP)

functions (3GPP TS 23.501 [42]). UP functions mainly take care of traffic forwarding while the CP functions manage the authentication, network slice selections, etc. The principal advantage of such separation is being able to flexibly scale the CP functions independently on UP functions in case of traffic peak and vice versa. Another benefit lies in the flexibility to separately deploy CP functions so that some functions can be deployed, according to the requirement of the use case, in a centralized datacenter or a distributed one close to the RAN. The flexibility in scaling and deployment makes 5G networks more complex than previous generations of the telecommunication networks.

The Core Network (CN) is the central element of a network that provides services to customers who are connected to the access network. The 5G core network is referred as 5GC, and is an evolved version of EPC (LTE Evolved Packet Core network) as a cloud-native and service-based-architecture (SBA) [43]. The main components of the 5GC are the Access and Mobility Function (AMF), Session Management Function (SMF), User Plane Function (UPF), Unified Data Management (UDM), Authentication Server Function (AUSF), Policy Control Function (PCF), Network Exposure Function (NEF), Network Repository Function (NRF) and Network Slicing Selection Function (NSSF). These 5G network functions are cloud-native by design, thanks to the Service Based Architecture (SBA) design of the 5GC. Therefore, their instantiation can take place as Virtual Network Functions (VNFs) or Container Network Functions (CNFs) in any of the available virtualization platforms. (See Fig. 3.)

The main goal is to adapt the 5GC functions independently when the load increases for any specific service or set of services, which is a major advancement from previous mobile network generations. To promote flexibility and reduce cost, it is possible to adopt COTS hardware at the NFV Infrastructure (NFVI) layer. These hardware resources are managed by open-source Virtual Infrastructure Management (VIM) software such as Openstack [49], OpenVIM [50] or Kubernetes [51]. The following Table 2 summarizes the available open-source solutions.

4.3. Softwarization, orchestration, virtualization and programmability

SDN is designed to make networks more flexible, controllable and agile. As a consequence, SDN enables network control to become directly programmable that makes its ability to provide network virtualization, automation, and create new services on top of virtualized resources. There exists a plethora of open source SDN solutions for mobile networks, including Open Networking Operating System (ONOS), Central Office Rearchitectured as a Datacenter (CORD), O-RAN, Open Network Automation Platform (ONAP) [12], AETHER and SD-RAN. (See Fig. 4.)

Management and Orchestration (MANO) frameworks [52] build on top of the network programmability and extended softwarization for network functions, and are being used to meet the agile and flexible management solutions for virtual network services in the 5G and beyond era. There are popular open source NFV MANO projects, namely OSM [53] and ONAP [12].

ETSI introduces the NFV MANO architecture, which comprises three main functional blocks, as further detailed below. MANO is an important component in managing the lifecycle of VNFs (including CFNs and PNFs) and hence managing overall infrastructure with agility and flexibility. The NFV MANO system entities, such as the Network Function Virtualization Orchestrator (NFVO), the Virtual Network Function
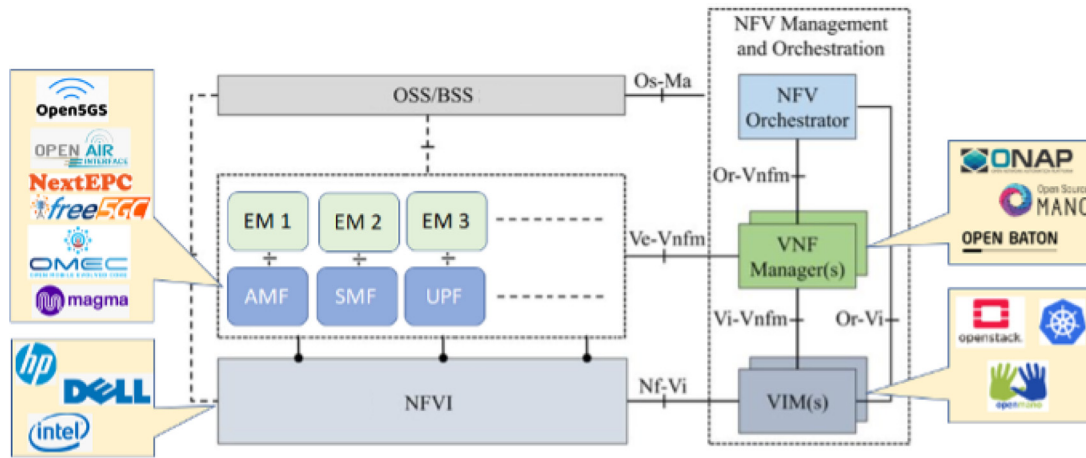
Fig. 3. Cloud-native instantiation of the 5G Core Network.

Table 2
Available open-source solutions.

| Name	Network domain	Description	References/links
Open5GS [44]	CN	5G/LTE software	https://open5gs.org
OpenAirInterface CN (OAI-CN) [45]	CN	5G/LTE software	https://openairinterface.org/
NextEPC [46]	CN	LTE EPC software	https://nextepc.org
srsEPC [32]	CN	LTE EPC software	https://github.com/srsran/srsRAN
Free5GC [47]	CN	5G software	https://free5gc.org
OMEC [41]	CN	LTE EPC software	https://opennetworking.org/omec/
Magma [48]	CN	LTE/5G software	https://docs.magmacore.org/docs/basics/introduction.html
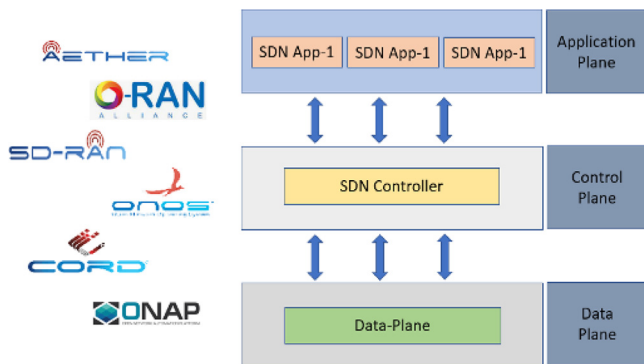
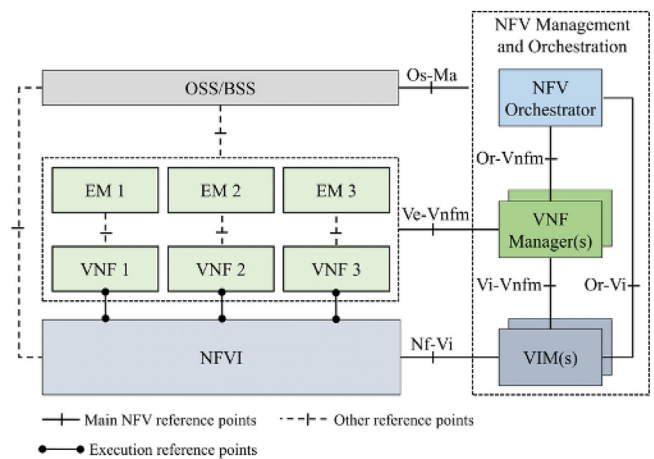**Fig. 4.** Key ONF SDN platforms.

Fig. 5. ETSI NFV-MANO architecture.

Manager (VNFM) and the Virtual Infrastructure Manager (VIM), coordinate with each other over well-defined reference points to manage entities such as Network Functions Virtualization Infrastructure (NFVI), VNFs, CNFs, Physical Network Functions (PNFs) and Network Services (NSs). In the context of research testbeds, MANO framework provides efficiency by bringing network functions to several experimenters (tenants/users) at the same time. Fig. 5 illustrates these blocks with the reference points that connect them.

The three main components of the NFV-MANO architecture are detailed below:

1. *Virtual Infrastructure Manager (VIM)* performs controlling mechanisms for the NFV Infrastructure (NFVI) resources within an infrastructure provider. VIM is also responsible for receiving fault measurement and performance information of NFVI resources. Consequently, VIM can supervise NFVI resources allocation to the available VNFs;

2. *VNF Manager (VNFM)* conducts one or several VNFs and does the lifecycle management of VNFs. VNF lifecycle management involves establishing/configuring, preserving, and terminating VNFs;

3. *NFV Orchestrator (NFVO)* implements resource and service orchestration in the network. NFVO is split up into Resource Orchestrator (RO) and Network Service Orchestrator (NSO). First, the RO collects the current information regarding possible physical and virtual resources of NFVI through the VIM. Following this, the NSO applies a complete lifecycle management of multiple network services. In this way, the NFVO keeps updating the information about the available VNFs running on top of NFVI. As a result, the NFVO can initiate multiple network services. As part of the lifecycle management, the NFVO can also terminate a network service whenever no longer a service request is received for that specific service. In several solutions, NFVO and VNFM are integrated into the MANO section.

Table 3

Frameworks for VNF lifecycle management.

Management and Orchestration framework	OSM [53]	ONAP [12]	CORD [40]	OpenBaton [55]
Ease of Installation	X	✓	✓	✓
Resource Footprint	High	High	Medium	Medium
Multi VIM support	✓	✓	X	✓
VNF, CNF & PNF Support	✓	✓	✓	✓
Multi-user Support (multi-tenancy)	✓	✓	X	✓
Multi-site Support (multi-domain)	✓	✓	✓	X
Network Slicing support	✓	✓	✓	✓
NFV-MANO compliance	✓	Partial	X	✓

Different frameworks have been developed in accordance with the NFV-MANO architecture, mainly aiming at providing fully-fledged solutions for the virtualized services lifecycle management. Such frameworks include multi-tenancy aspects, providing isolated slices of the infrastructure to each tenant, initially aiming at the execution of different vertical services on top of shared 5G infrastructure [54]. Such multi-tenancy aspects and isolation of traffic flows between each tenant of the infrastructure can be directly projected to the use of the same testbed infrastructure from multiple users concurrently, while providing guarantees for their performance. In Table 3, we list the different open source MANO frameworks that are currently widely utilized by the researchers as well as industry players like AT&T, Telefonica and others. This table also showcase the comparison between major open-source frameworks for VNF lifecycle management in terms of capabilities, multi-tenancy support, compliance or not with the NFV-MANO architecture, etc.

In the NFV world, containers are an emerging technology and the paradigm is standing between virtual machines and containers now. Containers show high utilization of computing resources and better performance than virtual machines. Multiple containers can be executed on the same host and share the same Operating System (OS) with other containers, each running isolated processes within its own secured space. Because containers share the base OS, the result is being able to run each container using significantly fewer resources than if each was a separate virtual machine (VM). Along with this trend, NFV industry has also been interested in the option of Containerized Network Functions (i.e., CNFs) instead of conventional Virtualized Network Functions (i.e., VNFs) due to its scalability and efficiency for operation and management. CNF-based solutions are also more appropriate for real-time networking functions. For those benefits, various mobile operators are trying to replace conventional VM-based NFV platforms with container-based platforms. Each VM includes a full copy of an operating system, the application, necessary binaries and libraries — taking up tens of GBs. VMs can be slow to boot, while Containers share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size), and thus handle more applications. Because they do not include the operating system, containers require fewer system resources and less overhead. They also tend to be faster to start/stop and they are ultra-portable across environments.

For low-latency use cases, 5G Core Network (CN) and RAN components are motivated to run as Containerized Network Functions (CNFs), instead of VMs in the case of Virtual Network Functions (VNFs), supported by tools like Kubernetes, that can deploy the services directly on bare-metal. Integration with the aforementioned NFVO tools like e.g., OSM is also possible. Open-source projects are moving towards cloud-native design, but until they become a reality, a mix of VNFs and CNFs could be adopted. Edge computing will have requirements for low-latency, cost-efficient infrastructure, secure with AI/ML capabilities. CNFs will be widely considered for the cases of Edge/Fog computing, due to the low complexity and fast instantiation of cloud-native services that can be achieved. However, simply fork-lifting existing 5G RAN software to a COTS platform is not enough. To realize the value of Cloud RAN, one needs to embrace cloud native architecture. Cloud native architecture facilitates RAN functions to

be realized as microservices in containers over bare metal servers, supported by technologies such as Kubernetes. The Table 4 lists some of the widely used open-source container solutions.

For high-speed user-plane networking in the 5GC and advanced signal and information processing in the RAN, hardware accelerators are commonly used in industrial solutions to ensure real-time operation. For experimental network deployments, several open solutions in the context of AETHER and OAI can be now leveraged to integrate P4 [59], FPGA [60,61] or GPU [62] based hardware accelerators in computing clusters. From a research perspective there are many challenges related to efficiently integrating such solutions with CNFs on generic computing platforms.

4.4. Multi-access edge computing

MEC (Multi-access Edge Computing) has been developed as a solution for network operators, enabling the extension of telecommunications infrastructure with servers offering computing resources to users and service providers (including cloud service providers). The relevant standards are developed by the ETSI standardization organization within the ISG MEC (Industry Specification Group on Multi-access Edge Computing) working group. The ETSI MEC solution is now an integral part of the 5G network infrastructure, but it can also be used in LTE networks and other access networks. It should be emphasized that the MEC technique is a significant step in the development of telecommunications infrastructure towards a future, integrated communication and computing infrastructure.

Parallel to the standardization work, research is carried out on the specification of mechanisms and algorithms for MEC systems. In particular, as part of projects related to the implementation of NFV systems or orchestration in cloud systems, working groups were established to implement extensions of these systems and to support the computing technique at the network edge. Examples of such initiatives are: Open NFV Edge [63], Edge Automation through ONAP [64], OpenStack Edge [65], or LinuxFoundation Edge [66]. The solutions proposed by the above projects are usually extensions of the architecture of orchestration systems developed for NFV backbone networks or cloud applications that offer the possibility of orchestrating applications at the edge of the network. Therefore, these solutions are not fully compatible with the ETSI MEC architecture.

In addition, research projects aimed at developing prototypes of MEC systems fully compliant with ETSI standards have been developed [67,68]. In particular, the SYMEC [69], implemented an ETSI-compliant MEC platform on the low-energy ARM architecture.

5. Architecture guidelines for SLICES

The initial guidelines on the various hardware building blocks for different types of SLICES facilities can be categorized into four basic sub-systems:

- Inter-Facility Interconnections and Intra-Facility Switching Fabric;
- Real-time and Non-real-time Computing;
- Radio Infrastructure;

Table 4

Widely used open-source container solutions.

Container solution	Description	References/links
Kubernetes [51]	Developed by Google, most widely used	https://kubernetes.io/
Docker [56]	Software platform that allows you to build, test, and deploy applications quickly	https://www.docker.com/
Openshift [57]	Container management tool based on Kubernetes created by RedHat	https://www.redhat.com/en/technologies/cloud-computing/openshift
Apache Mesos [58]	Apache Mesos is an open-source cluster management system	http://mesos.apache.org/
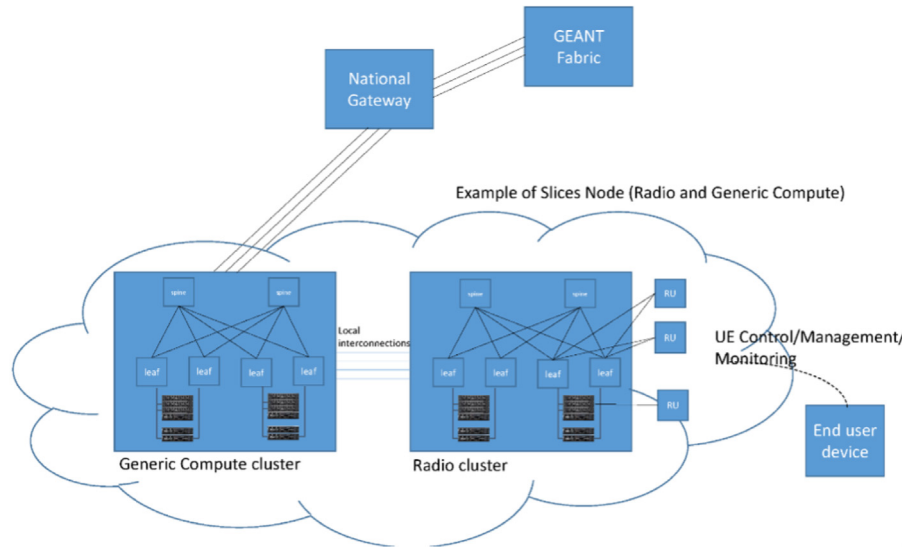
Fig. 6. A high-level view of a SLICES node from an equipment standpoint.

• End-user devices.

The example of a SLICES node is shown in Fig. 6. It demonstrates two interconnected clusters in the same geographic region, one of which is equipped with Radio Units and the other is a more generic computing platform. The left cluster has a long-distance interconnection with the national gateway, which itself is interconnected with the GEANT fabric and the rest of the SLICES network. In the following subsections we provide some initial guidelines for the architecture of the various components. As a general rule for hardware and network topologies, SLICES nodes should aim to mutualize as much as possible the types of computing and networking equipment in order to reuse deployment and configuration methods and to be able to share and establish common best practices. This follows the spirit of similar large-scale platform projects such as the Linux Networking Foundation OPNFV [70], Cloud-Native Computing Foundation [71] and the Open Compute Foundation [72]. Because of the lack of space and its diversity, we will not describe further the hardware components envisaged in SLICES.

As SLICES aspires to provide fully programmable remotely accessible infrastructure to the Digital Infrastructure community, the respective frameworks shall be developed for ensuring seamless and easy access to the experimental resources. The different site facilities will form an integrated single pan-European facility, adopting common tools for managing and orchestrating experiments over the infrastructure, as well as providing single access credentials to users. A first attempt to sketch our reference architecture, with respect to the tools used for its management, is described in Fig. 7.

Towards achieving this integration, the sites will adopt network virtualization for their disaggregated resources. Each node will be considered as a single domain for experimentation [73], while the overall orchestration of experiments will be performed through a centralized infrastructure. Site and node selection frameworks will be developed in the context of SLICES, towards ensuring the optimal use of resources among the sites.

Moreover, and towards ensuring the smooth operation of the infrastructure, tools for facilitating access will be developed and deployed. Open-source software shall be employed, based on the paradigms of existing testbed access schemes, user authentication and authorization. This software will be appropriately tailored with new modules for managing the new equipment described in the previous section. Table 5 provides a comparison between the existing tools for the experimentation plane of the experiments, and the progress beyond them.

In terms of integration of the various components, the software tools shall encompass single-sign in procedures, with access certificates issued by a single authority. The resource discovery, reservation, and allocation shall comply with the access policies for SLICES (following the ESFRI principles) and be interchanged with the respective facility authorities through a standardized process. For this purpose, the SFA protocol [7] has been extensively used in past and present solutions and could inspire a future candidate together with new complementary or alternative proposals that will be considered as well.

Based on the automation tools for accessing the infrastructure, we intend to equip new experimenters with a store in order to easily deploy services with a single click manner over the infrastructure. This can be achieved with these frameworks by using pre-compiled versions of services, and by supporting different methods for virtualization of resources (e.g., Virtual Machines, docker containers, Linux Containers). For example, public docker repositories provide different images that can be used to deploy commonly used services (e.g., databases, web services, applications and application servers) through a friendly interface. Moreover, the entire architecture will be augmented with the appropriate tools for experiment monitoring, experiment data and results visualization and cross-correlation analysis and inference with previous experiments executed over the infrastructure.

The SLICES architecture, illustrated in Fig. 8, can be designed considering the limitations and challenges of existing federation-based architectures such as SFA. For example, the SLICES architecture could be designed by advancing the Slice-based Federation Architecture (SFA) and further enhancements are required to overcome the limitations
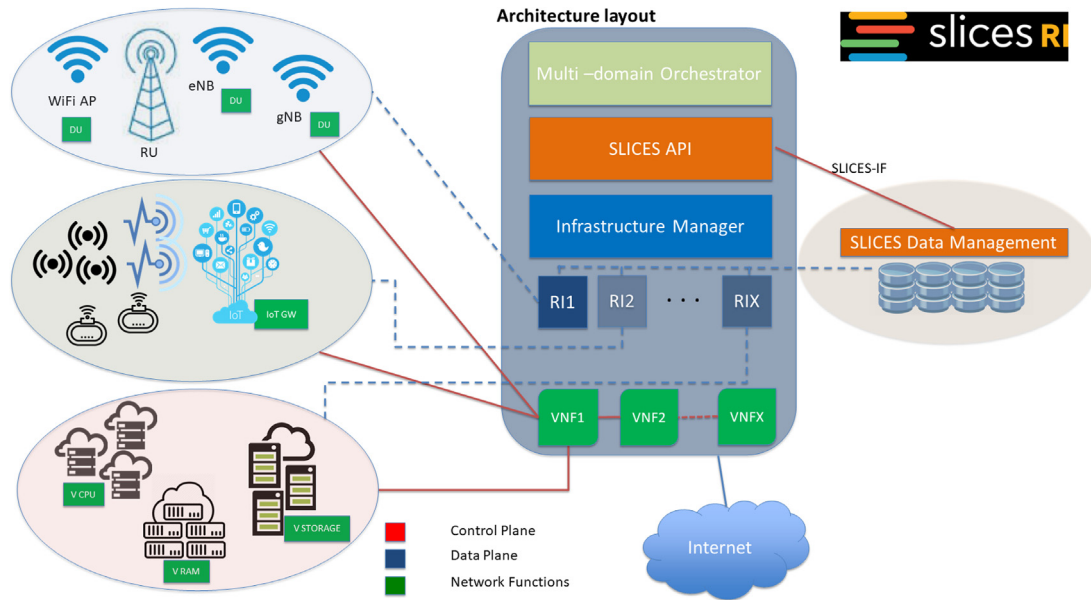
Fig. 7. SLICES Infrastructure conceptual architecture.

Table 5
Comparison of different proposed frameworks vs existing ones for the SLICES architecture.

Existing tools	Proposed solution(s)	Benefits
Control and Management Framework (OMF) [74]	NFV-based orchestration solution	Current tools provide metal as a service access to the testbed resources, or in some cases virtualized access by interacting with the respective VIM interface of a testbed. On top, the experiments can be orchestrated by using a publish/subscribe scheme for the communication between a centralized controller and the actual resources. Adopting an NFV-based solution will allow the orchestration of experiments as Virtual Network Functions (virtualized access) or Physical Network Functions (Metal as a Service access), through the adoption of industry-grade tools. These shall allow higher utilization of the testbed resources, increasing the user capacity of each testbed, more secure end-to-end experiments, end-to-end network configuration and experiment reliability.
SDN programmability	SDN Assist	Current tools aim in providing a programmable interface for users that shall use their own controller for managing the flows in the network. In some cases, isolation of flows between different users on a switch is possible, through the adoption of tools like FlowVisor . Moving to an NFV-based orchestration solution supporting features like SDN Assist enables the programming of flows for an experiment during the instantiation time. Based on an end-to-end programmable SDN plane (based on Open-vSwitch or hardware OpenFlow/P4 switches) programmability extends to the entire datapath used for the experiments, isolating users and providing multi-tenancy over the infrastructure.
Wireless programmability	Open-RAN (ORAN)	Current tools for programming the wireless components rely on specific interfaces dedicated to specific equipment for the RAN. As such interfaces become standardized, through efforts like O-RAN alliance, adopting such APIs can increase the supported equipment, and open-up more programmability for the RAN. As such tools use standardized interfaces, they integrate with several NFV based orchestration solutions, allowing a truly end-to-end experiment configuration and instantiation.
Edge and Core Configuration	NFV-based orchestration solution	Current tools provide Virtualized access to the core and cloud network configuration, or in some cases metal-as-a-service access. Switching to the same NFV based orchestration solution as the rest of the nodes will enable the seamless network configuration, and move the edge/core cloud configuration to supporting a different number of settings (such as cloud-native 5G network configuration).

and complexities to integrate wireless, edge and other experimental resources. We can consider a layer-based architecture as shown in Fig. 8.

In this architecture, every component of SLICES testbed falls under a certain layer:

1. *Resource Layer*: It includes experimental resources such as CPU's, RAM, storage, containers, VM's, network, wireless, HPC and IoT devices;
2. *Virtualization Layer*: This layer includes cloud computing platforms (e.g., Openstack) that virtualize the underlying hardware resources and provide interfaces to the higher layers for programming/instantiating services over them. Examples of such programming interfaces are the ones defined by the O-RAN alliance (e.g., A1/E2 interfaces), or the P4 programming abstractions for wired networks;
3. *Orchestration Layer*: It includes tools that orchestrate and instantiate services over the infrastructure equipment. Examples

of such tools are OSM, ONAP and Kubernetes, mainly involved in NFV Management and Orchestration. It provides Network-Function-as-a-service and exposes northbound interfaces (NBI) APIs to be used by external entities;

4. *NBI Layer*: This Layer defines the Open APIs that can be used by the SLICES application framework. Examples of such interfaces are the SOL005, the SOL004 from the ETSI NFV-MANO architecture [75] that can be found as the NBI interface of several MANO compliant tools, or even more generic ones, like TM-Forum based APIs for service lifecycle control;
5. *Application Layer*: This Layer will host the SLICES-Core application, located at the SLICES central hub. It is responsible for managing all experimental resources that are exposed by lower layers, saved in the database and is further exposed to experimenters as a Service-Catalog. It also exposes NBI API's that can
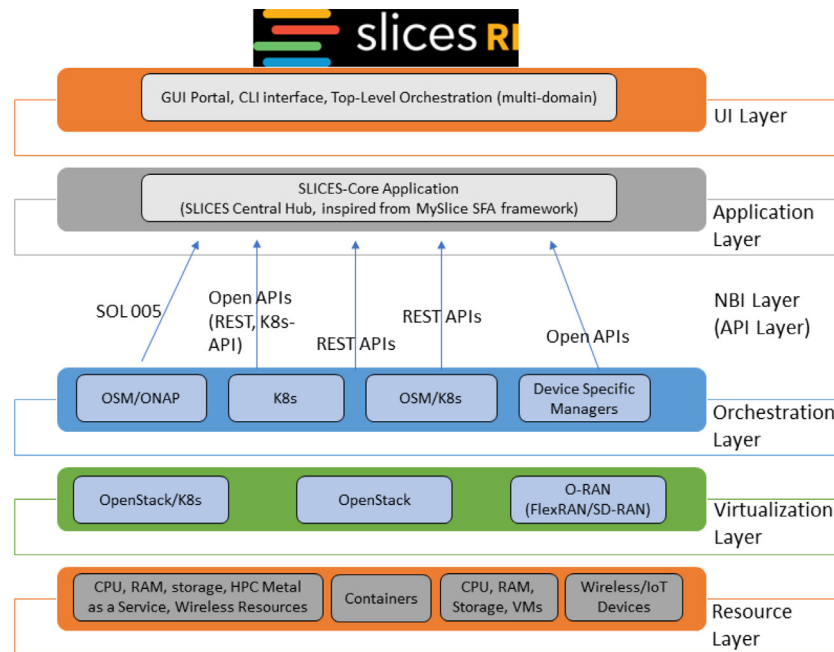
Fig. 8. Layered architecture for SLICES.

be used by a 3rd party orchestrator. The architecture of SLICES-Core application will start from components similar to MySlice V2 [76], and will be further enhanced at later stages;

6. *UI Layer*: This Layer defines the User Interface for the experimenters. It should abstract the experiments enough to make them more user friendly as possible.

The operation of the central-hub relies on the control of multiple-domains through the SLICES core application. Its operation resembles the functionality of a multi-domain orchestrator, that brings together different domains (in different locations, managed from different authorities) under the supervision of a single authority. The multi-domain orchestrator glues NFV, MEC and Cloud-Native orchestrators using API abstraction layers. Different groups of experimental resources on any of those testbeds might be virtualized and managed based on different technologies (e.g., VMs and containers). This in turn requires that multi-domain orchestrators use different orchestrators, managing different types of resources. For example, an NFVO and a MEC (Multi-access Edge computing Application Orchestrator-MEAO) are likely to be required in individual testbeds services, both managed concurrently through similar APIs from the same central entity.

The central SLICES core application shall include an abstraction API, used to trigger the required API invocation chains on different domain orchestrators when a high-level action is performed. A set of southbound clients is used in order to connect to NFV, MEC and Cloud-native local domain orchestrators.

It is worth noticing that the different levels of access provided over the facility will also correspond to finer or coarser grain control over the deployment of experimental resources. More experienced users would be willing to control exactly which experimental resource to use from which facility at which site, while less-experienced users might not even know that their experiment is actually using heterogeneous resources composed out of different facilities spread across Europe. In the latter case, SLICES, through dedicated management components, will automatically assign resources to the users' experiments, in order to optimize the overall utilization of the RI's resources or simplify the work of the experimenter.

6. The research life cycle

Experimentally-driven research should be grounded on a solid methodology that is understood and implemented by other disciplines.

This is somehow the ambition of the European EOSC initiative. As a consequence, SLICES does not target only the deployment of the instrument/facility but as importantly, addresses the full research life-cycle, including open data, data management and reproducibility.

Researchers and research stakeholders nowadays require that research data is made available for other researchers to examine, experiment and develop further. Additionally, preserving the data in conjunction with how conclusions from the data were drawn, accelerates the discovery process, enable easier reproducibility of the results and thus supports evidence. It is then necessary to develop policies and procedures for regulating the management and publication of research data in order to make them interoperable and widely available.

In Europe, it is recommended to conform with the European Open Science [77] and Open Access policy [78], Open Research Data Pilot [79] and FAIR [80] principles in producing and managing research data. This requires defining appropriate metadata (including compatible experiment description) on the data produced by or integrated into the infrastructure with the objective to ensure eventually data accessibility, trustworthiness, reusability and interoperability with data produced by similar infrastructures/experiments for enabling complex experiments and multi-domain research. Alignments with the relevant recommendations such as the ones published by EOSC FAIRsFAIR [81] project, GO FAIR initiative [82] and RDA for FAIR data management [83], and general European Open Access to research publications and Open Research Data Pilot policies, are of utmost importance.

The FAIR (Findable, Accessible, Interoperable, and Reusable) [84] Data Principles were developed to be used as guidelines for data producers and publishers, with regards to data management and stewardship. One important aspect that differentiates FAIR from any other related initiatives is that they move beyond the traditional data and they place specific emphasis on automatic computation, thus considering both human-driven and machine-driven data activities. Since their publication, FAIR principles became widely accepted and used. To this end, SLICES fully endorses and adopts the FAIR principles, acting as a catalyst to enable and foster the data-driven science and scientific data-sharing in this area.

Understanding the data collected and processed within SLICES becomes essential to understand data usage from the target user groups. This should allow to develop an appropriate information model that represents the data collected from the SLICES testbeds, experimental

equipment and applications. We consider that the datasets generated by the usage of the SLICES hardware and software infrastructure can be roughly organized into five main categories:

- **Observational Data:** collected using methods such as surveys (e.g. online questionnaires) or recording of measurements (e.g. through sensors). The data include mostly data related to signal or performance measurements, and network or service log data that allow for experiment evaluation and reproducibility.
- **Experimental Data:** where researchers introduce an intervention and study the effects of certain variables, trying to determine their impact.
- **Simulation Data:** is generated by using computer models that simulate the operation of a real-world process or system. These may use observational data.
- **Derived Data:** involves the analysis (e.g. cleaning, transformation, summarization, predictive modeling) of existing data, often coming from different datasets (e.g. the results of two experiments), to create a new dataset for a specific purpose.
- **Metadata:** concerns data that provides descriptors about all categories of data mentioned above. This information is essential in making the discovery of data easier and ensuring their interoperability.

SLICES, as an open platform, promotes interoperability, thus nonproprietary, unencrypted, uncompressed, and commonly used by the research community formats should be adopted. In addition, SLICES end users should have the ability to decide on a suitable license and attach it to their data.

Our preliminary estimations for SLICES include up to 5000 users and their data, accounting for up to 50 GB per user on the individual nodes and up to 1 TB on the cloud. This provides us with a preliminary estimation of 0.25PB–1PB of data storage for all datacenters residing on SLICES nodes, and 5PB for the cloud-based datacenter.

As a consequence, SLICES will setup a data management framework to support the efficient and effective operation of the SLICES infrastructure. To accomplish this, the data management framework sets its own design goals, which are summarized below.

- **Data Governance:** A systemic and effective Data Governance structure to support the data management operations through a hierarchical structure with appropriate roles (e.g. Data Manager, Data Protection Officer and Metadata administrator), implement all related policies and processes, and adopt standards and leading practices.
- **Data Architecture:** An agile Data Architecture that can perform efficiently to fulfill the SLICES infrastructure requirements, scales gracefully to accommodate for increased workloads, is flexible to integrate new processes and technologies, and is open to interact with other systems and infrastructures.
- **Data Quality:** Appropriate data transformation mechanisms to ensure Data Quality across multiple dimensions (e.g. accuracy, completeness, integrity), in order to improve data utility (e.g. further processing, analysis).
- **Metadata:** Appropriate metadata management mechanisms to facilitate collaboration between users by providing the means to share their data and also support FAIR data.
- **Interoperability:** Facilitate seamless interaction with other systems and infrastructures.
- **Analytics:** Deployment of statistical, machine learning and artificial intelligence techniques to draw valuable insights from data and appropriate visualization techniques to interpret them.
- **Data Security:** Mechanisms to protect data from unauthorized access and protect its integrity.
- **Privacy:** Strict controls to manage the sharing of data, both internally and externally.

7. Interoperability with EOSC and external systems

Since SLICES aims to provide a pan-European experimental research platform by jointly utilizing the geographically dispersed computing, storage and networking RIs, it is highly important that the different RIs interacting in the experimental workflow are interoperable with each other. Similarly, existing research needs to be accessible and directly pluggable to SLICES services and sites. For example, considering a MEC use case, compute, storage and networking resources from different RIs can be used. In such a scenario, it is necessary that resource description, availability, execution and data exchange are smooth. This can only be assured if a common interoperability framework is adopted across the SLICES ecosystem so that different subsystems have a common understanding of resources and data/metadata are on the same page with respect to the licensing, copyright and privacy requirements. The SLICES infrastructure is designed to ensure compatibility and integration with EOSC and existing ESFRI infrastructures, and be ready to offer advanced ICT infrastructure services to other RIs and projects, with the special focus on the FAIR data management and exchange. (See Fig. 9.)

EOSC [85] has established itself as an important pillar in the implementation of Open science concept by accelerating the adoption of the FAIR data practices among researchers in the European Union. Integration of SLICES into European Research Infrastructure via EOSC will facilitate data sharing and reuse among SLICES partners and the larger European researchers' community. Interoperability-focused integration of SLICES with EOSC will make it easier for SLICES users to reap the benefits of many services and tools pertaining to diverse scientific domains that are being developed around the EOSC ecosystem.

Therefore, it is of utmost importance to design the integration framework of SLICES with EOSC in such a way that the data exchange between SLICES and EOSC is interoperable for scientific workflow management for data storage, processing and reuse. To this end, the recommendations of the EOSC interoperability framework are considered in great detail for the design of the SLICES interoperability framework.

Interoperability is an essential feature of EOSC ecosystem as a federation of services and data exchange is unthinkable without interoperability among different EOSC constituents. The meaningful exchange and consumption of digital objects is necessary to generate value from EOSC, which can only be realized if different components of the EOSC ecosystem (software/machines and humans) have a common understanding of how to interpret and exchange them, what are the legal restrictions, and what processes are involved in their distribution, consumption and production. To facilitate this, EOSC interoperability framework (EOSC-IF) [86] is defined as a generic framework for all the entities involved in the development and deployment of EOSC.

To achieve this, a dedicated interface, coined SLICES-Interoperability Framework (SLICES-IF) shall be developed. The interface will be built upon the foundations led by the European Interoperability Reference Architecture (EIRA) [87], where interoperability is classified at four layers, namely: (i) technical, (ii) semantic, (iii) organizational; and (iv) legal. Although the target audience for EIRA (governance and administration) was very different from the SLICES stakeholders, core principles and objectives are similar. Additionally, the different components (in particular technical and semantic) of SLICES-IF would be chosen in such a way that SLICES is fully interoperable with EOSC for uninterrupted data exchange pertaining to use of EOSC services and research data by SLICES as well as to enable the publications of SLICES infrastructure, services and data through EOSC portal. More details about the SLICES-IF interface to EOSC and external RIs is provided in SLICES-Design Study Deliverable D4.2 [88].

SLICES aims to allow its users (and interoperating platforms) to uniformly find, and access any object, such as data, services and software. To accomplish this, SLICES defines a hierarchical metadata structure, where each digital object is first described using a select set of common
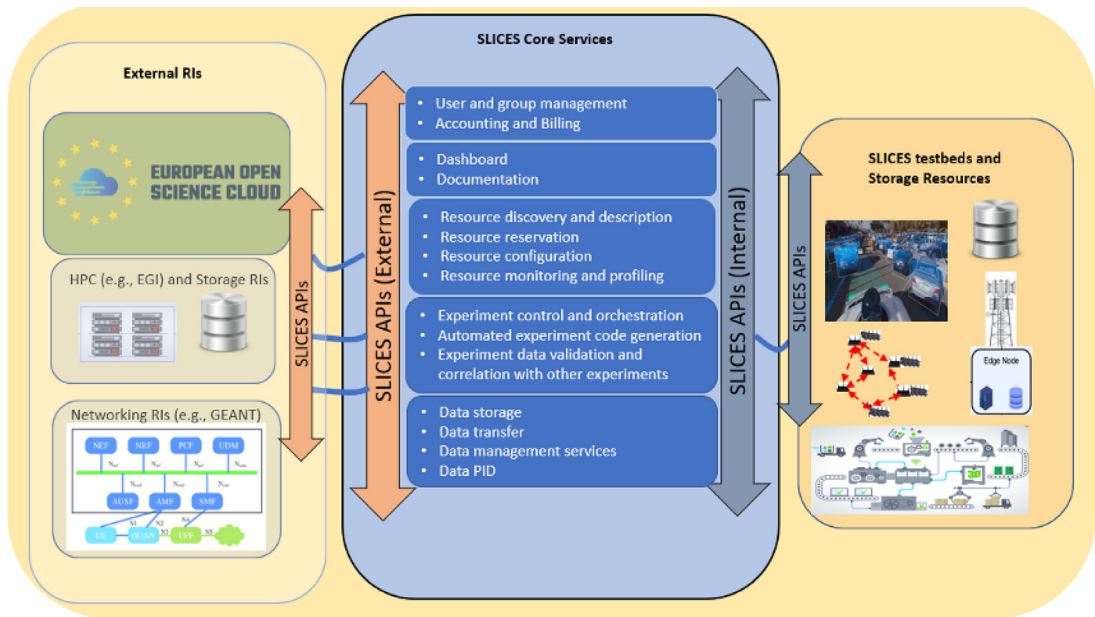
Fig. 9. SLICES interconnection with European e-Infrastructures and digital infrastructures.
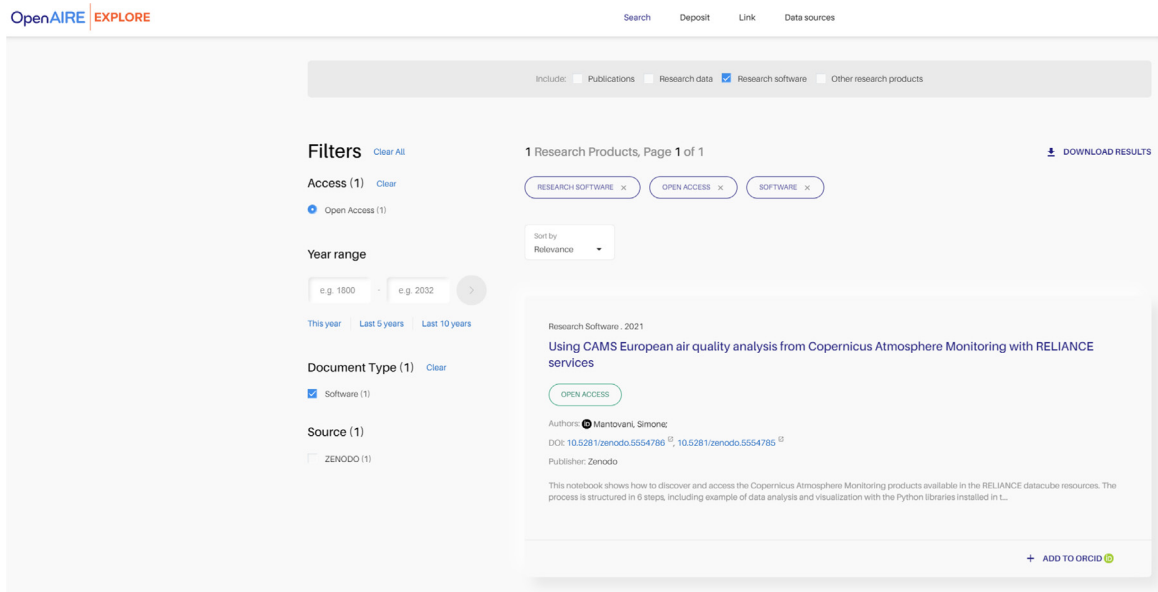

Fig. 10. OpenAIRE Explore.

metadata attributes and then according to its type, the description is extended with a set of type-specific attributes. The relevant information can be then accessed using SLICES authentication and authorization mechanisms.

In order to realize the vision of FAIR research, supporting the full research lifecycle, let us consider a simple example borrowed from another field of research and illustrated by the Reliance project.[5] Reliance delivers a suite of innovative and interconnected services that extend EOSC's capabilities to support the management of the research lifecycle within Earth Science Communities and Copernicus Users. Consider core services provided to the research community, that could be data, software publications, others. These core services are named after research objects that are for use by the experimenters and share by the experimenters. As an illustration, assume that you are

doing some research related to the Copernicus air quality. You can go to the OpenAIRE.[6] explorer and search for Copernicus quality. And you will find all the associated resources as described in Fig. 10

You are looking for a software, because someone has produced a software taking research data as input and producing a map of the air quality in a given region as an output. You find the software and with the software comes a set of additional metadata. So for instance, it could be a Jupyter Notebook as in Fig. 11.

You now have access to the software that will execute exactly what has produced this research data. What you are willing to do, at first, is to reproduce the results. On the other hand, you would like to take your own data, use the same process and produce your own new results. The last step is that you go to the service, which is named Rohub.[7] And then

---

5 Reliance, https://www.reliance-project.eu/.

6 OpenAIRE, https://explore.openaire.eu/.

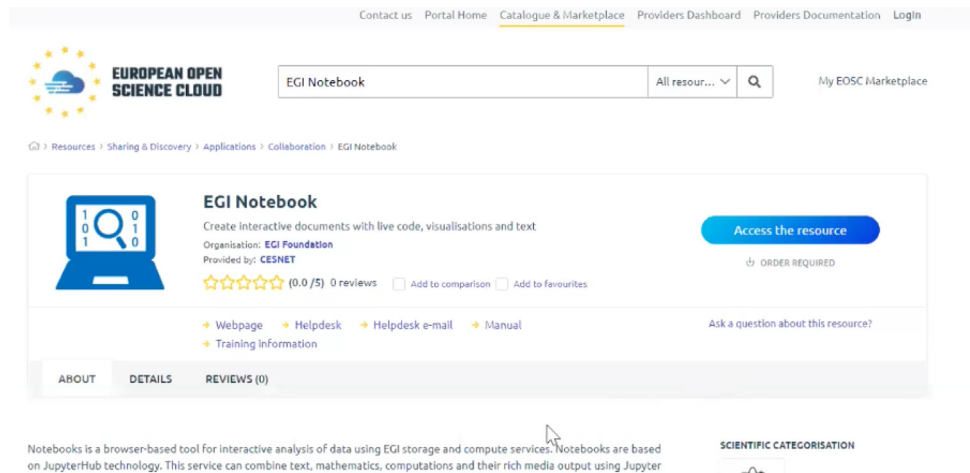7 Rohub, https://reliance.rohub.org/.

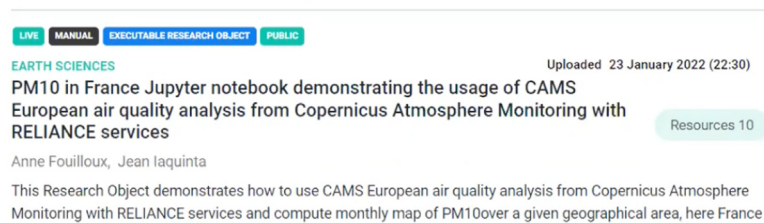**Fig. 11.** EGI Notebook.

Fig. 12. Jupyter notebook.

you bundle your different resources, like the Jupyter notebook that you have used, the data that you have exploited, and the output that you have produced. You can now publish this research outcome as your own contribution made available to the community, defined as PM 10 in Fig. 12. This full research-life cycle is really important, otherwise, the result that you produce cannot be published, because it simply cannot be reproduced. There exists data initiatives in our field, like the ACM Artifact Review level [89], it is nice and ambitious. But it does not yet fully align with the best practices in other fields of research.

8. Conclusion

It is a best practice in fundamental sciences to think about thought experiments that will validate the scientific assumptions. It is indeed a challenging endeavor to design a test platform to support networking and distributed research. Up to now, networking test-beds have tried to capture a variety of demands. However, very little has been done to cover the entire research and data lineage life-cycle. SLICES is the outcome of an effort to align the methodology to build such a platform in order to satisfy the key requirements of a scientific instrument. In Europe, ESFRI provides such a framework where most of the large research infrastructures are incubated, deployed and operated. The paper describes our continuous work aiming at designing the SLICES end-to-end reference architecture. It emphasized the analysis of the current demand from relevant ICT stakeholders, and the foundational principles on which it will be grounded. These principles, alongside with the current trends in resource management (resource programmability, network virtualization, resource disaggregation) have resulted in the wide adoption of several Management and Orchestration (MANO) frameworks for deploying experiments and applications over distributed infrastructures. The paper also discusses major open-source software that is used by the research community for networking at large, experiments that provides opportunities for SLICES. The integration and

interoperability with EOSC infrastructure are also presented and the research life-cycle is illustrated. This work is based on the long experience of the participating members in managing and operating test platforms infrastructures to best serve our research community.

CRediT authorship contribution statement

Serge Fdida: Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Nikos Makris:** Writing – original draft, Writing – review & editing, Visualization. **Thanasis Korakis:** Writing – review & editing, Project administration, Funding acquisition. **Raffaele Bruno:** Writing – review & editing. **Andrea Passarella:** Methodology, Writing – review & editing, Project administration, Funding acquisition. **Panayiotis Andreou:** Methodology, Writing – review & editing, Project administration. **Bartosz Belter:** Writing – review & editing, Project administration. **Cédric Crettaz:** Methodology, Project administration. **Walid Dabbous:** Writing – review & editing, Project administration. **Yuri Demchenko:** Methodology, Writing – review & editing, Project administration. **Raymond Knopp:** Writing – review & editing, Methodology, Project administration.

Declaration of competing interest

Acknowledgment

References

[1] P. Antoniadis, S. Fdida, T. Friedman, M. Mishal, OneLab: An open federated facility for experimentally driven future internet research, in: Proceedings of Conext 2010, Springer, 2010, pp. 1–12.

[2] T. Wauters, B. Vermeulen, W. Vandenberghe, P. Demeester, S. Taylor, L. Baron, M. Smirnov, Y. Al-Hazmi, A. Willner, M. Sawyer, et al., Federation of internet experimentation facilities: Architecture and implementation, in: European Conference on Networks and Communications, EuCNC 2014, 2014.

[3] P. Antoniadis, S. Fdida, T. Friedman, V. Misra, Federation of virtualized infrastructures: Sharing the value of diversity, in: Proceedings of the 6th International Conference, Co-NEXT '10, Association for Computing Machinery, New York, NY, USA, 2010, http://dx.doi.org/10.1145/1921168.1921184.

[4] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, M. Singh, Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols, in: IEEE Wireless Communications and Networking Conference, 2005, vol. 3, IEEE, 2005, pp. 1664–1669.

[5] C. Elliott, GENI-global environment for network innovations, in: LCN, 2008, p. 8.

[6] M. Lemke, The European FIRE future internet research and experimentation initiative, in: 2009 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops, 2009, pp. 2–3, http://dx.doi.org/10.1109/TRIDENTCOM.2009.4976186.

[7] J. Auge, T. Parmentelat, N. Turro, S. Avakian, L. Baron, M.A. Larabi, M.Y. Rahman, T. Friedman, S. Fdida, Tools to foster a global federation of testbeds, Comput. Netw. 63 (2014) 205–220.

[8] A. Gosain, Platforms for advanced wireless research: Helping define a new edge computing paradigm, in: Proceedings of the 2018 on Technologies for the Wireless Edge Workshop, 2018, p. 33.

[9] F. Moggio, M. Boldi, S. Canale, V. Suraci, C. Casetti, G. Bernini, G. Landi, P. Giaccone, 5G EVE a European platform for 5G application deployment, in: Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization, 2020, pp. 124–125.

[10] H. Koumaras, D. Tsolkas, G. Gardikis, P.M. Gomez, V. Frascolla, D. Triantafyllopoulou, M. Emmelmann, V. Koumaras, M.L.G. Osma, D. Munaretto, et al., 5GENESIS: The genesis of a flexible 5G facility, in: 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD, IEEE, 2018, pp. 1–6.

[11] C. Kalogiros, G. Zois, G. Darzanos, H.K. Hallingby, H. Lønsethagen, M.B. Weiss, A. Gavras, The potential of 5G experimentation-as-a-service paradigm for operators and vertical industries: The case of 5G-VINNI facility, in: 2019 IEEE 2nd 5G World Forum, 5GWF, IEEE, 2019, pp. 347–352.

[12] Linux Foundation, ONAP – Open network automation platform, [Online], http://onap.org/.

[13] L. Gavrilovska, V. Rakovic, D. Denkovski, From cloud RAN to open RAN, Wirel. Pers. Commun. (2020) 1–17.

[14] N. Nikaein, M.K. Marina, S. Manickam, A. Dawson, R. Knopp, C. Bonnet, OpenAirInterface: A flexible platform for 5G research, ACM SIGCOMM Comput. Commun. Rev. 44 (5) (2014) 33–38.

[15] European Strategy Forum on Research Infrastructures (ESFRI), [Online], https://www.esfri.eu/.

[16] M.S. Bonfim, K.L. Dias, S.F.L. Fernandes, Integrated NFV/SDN architectures: A systematic literature review, ACM Comput. Surv. 51 (6) (2019) 1–39.

[17] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, H. Flinck, Network slicing and softwarization: A survey on principles, enabling technologies, and solutions, IEEE Commun. Surv. Tutor. 20 (3) (2018) 2429–2453, http://dx.doi.org/10.1109/COMST.2018.2815638.

[18] X. Cheng, Y. Wu, G. Min, A.Y. Zomaya, Network function virtualization in dynamic networks: A stochastic perspective, IEEE J. Sel. Areas Commun. 36 (10) (2018) 2218–2232, http://dx.doi.org/10.1109/JSAC.2018.2869958.

[19] W. Wu, C. Zhou, M. Li, H. Wu, H. Zhou, N. Zhang, X.S. Shen, W. Zhuang, AI-native network slicing for 6G networks, IEEE Wirel. Commun. 29 (1) (2022) 96–103, http://dx.doi.org/10.1109/MWC.001.2100338.

[20] F. Rebecchi, M. Dias de Amorim, V. Conan, A. Passarella, R. Bruno, M. Conti, Data offloading techniques in cellular networks: A survey, IEEE Commun. Surv. Tutor. 17 (2) (2015) 580–603, http://dx.doi.org/10.1109/COMST.2014.2369742.

[21] F. Dressler, C.F. Chiasserini, F.H.P. Fitzek, H. Karl, R. Lo Cigno, A. Capone, C.E. Casetti, F. Malandrino, V. Mancuso, F. Klingler, G. Rizzo, V-Edge: Virtual edge computing as an enabler for novel microservices and cooperative computing, IEEE Netw. (2022).

[22] K. Han, S. Li, S. Tang, H. Huang, S. Zhao, G. Fu, Z. Zhu, Application-driven end-to-end slicing: When wireless network virtualization orchestrates with NFV-based mobile edge computing, IEEE Access 6 (2018) 26567–26577, http://dx.doi.org/10.1109/ACCESS.2018.2834623.

[23] H. Chergui, A. Ksentini, L. Blanco, C. Verikoukis, Toward zero-touch management and orchestration of massive deployment of network slices in 6G, IEEE Wirel. Commun. 29 (1) (2022) 86–93, http://dx.doi.org/10.1109/MWC.009.00366.

[24] L. Bonati, M. Polese, S. D'Oro, S. Basagni, T. Melodia, Open, programmable, and virtualized 5G networks: State-of-the-art and the road ahead, Comput. Netw. 182 (2020) 107516.

[25] M. Polese, L. Bonati, S. D'Oro, S. Basagni, T. Melodia, Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges, 2022, arXiv preprint arXiv:2202.01032.

[26] L.M.P. Larsen, A. Checko, H.L. Christiansen, A survey of the functional splits proposed for 5G mobile crosshaul networks, IEEE Commun. Surv. Tutor. 21 (1) (2018) 146–172.

[27] D. Wypiór, M. Klinkowski, I. Michalski, Open RAN—Radio access network evolution, benefits and market trends, Appl. Sci. 12 (1) (2022) 408.

[28] J. Kim, D. Kim, S. Choi, 3GPP SA2 architecture and functions for 5G mobile communication system, ICT Express 3 (1) (2017) 1–8.

[29] A. Ghosh, A. Maeder, M. Baker, D. Chandramouli, 5G evolution: A view on 5G cellular technology beyond 3GPP release 15, IEEE Access 7 (2019) 127639–127651.

[30] S. Rommer, P. Hedman, M. Olsson, L. Frid, S. Sultana, C. Mulligan, 5G Core Networks: Powering Digitalization, Academic Press, 2019.

[31] 3GPP, 3GPP TS 38.470 V17.0.0 (2022-04), 3rd generation partnership project; Technical specification group radio access network; NG-RAN; F1 general aspects and principles (release 17), 2022.

[32] srsRAN: A 4G/5G software radio suite, [Online], https://github.com/srsran/srsRAN.

[33] 3GPP, 3GPP TS 28.541 V17.6.0 (2022-03), 3rd generation partnership project; Technical specification group services and system aspects; Management and orchestration; 5G network resource model (NRM); Stage 2 and stage 3 (release 17), 2022.

[34] L. Bonati, S. D'Oro, M. Polese, S. Basagni, T. Melodia, Intelligence and learning in O-RAN for data-driven NextG cellular networks, IEEE Commun. Mag. 59 (10) (2021) 21–27.

[35] R. Schmidt, M. Irazabal, N. Nikaein, FlexRIC: An SDK for next-generation SD-RANs, in: Proceedings of the 17th International Conference on Emerging Networking Experiments and Technologies, 2021, pp. 411–425.

[36] X. Foukas, N. Nikaein, M.M. Kassem, M.K. Marina, K. Kontovasilis, FlexRAN: A flexible and programmable platform for software-defined radio access networks, in: Proceedings of the 12th International on Conference on Emerging Networking Experiments and Technologies, 2016, pp. 427–441.

[37] ONF, ONF software defined RAN (SD-RAN), 2022, [Online], https://opennetworking.org/sd-ran/.

[38] ONF, ONF AETHER - 5G connected edge platform, 2022, [Online], https://opennetworking.org/aether/.

[39] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, et al., ONOS: Towards an open, distributed SDN OS, in: Proceedings of the third Workshop on Hot Topics in Software Defined Networking, 2014, pp. 1–6.

[40] L. Peterson, A. Al-Shabibi, T. Anshutz, S. Baker, A. Bavier, S. Das, J. Hart, G. Palukar, W. Snow, Central office re-architected as a data center, IEEE Commun. Mag. 54 (10) (2016) 96–101.

[41] ONF, ONF OMEC - open mobile evolved core, 2022, [Online], https://opennetworking.org/omec/.

[42] 3GPP, 3GPP TS 23.501 V17.5.0 (2022-06), 3rd generation partnership project; Technical specification group services and system aspects; System architecture for the 5G system (5GS); Stage 2 (release 17),, 2022.

[43] O. Arouk, N. Nikaein, 5G cloud-native: Network management & automation, in: NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2020, pp. 1–2.

[44] Open5GS, Open source project of 5GC and EPC (Release 16), [Online], https://open5gs.org/.

[45] OAI, OpenAirInterface Core Network, [Online], https://gitlab.eurecom.fr/oai/cn5g.

[46] NextEPC, NextEPC: Open Source EPC, [Online], https://nextepc.org/.

[47] Free5GC, Open source implementation of 5G core network (3GPP release 15 and beyond), [Online], https://www.free5gc.org/.

[48] Meta, Magma: Communications service providers leverage Magma's open network core solution to connect people using LTE, 5G, Wi-Fi, and beyond, [Online], https://www.facebook.com/connectivity/solutions/magma.

[49] T. Rosado, J. Bernardino, An overview of openStack architecture, in: Proceedings of the 18th International Database Engineering & Applications Symposium, 2014, pp. 366–367.

[50] R. Mijumbi, J. Serrat, J.-L. Gorricho, S. Latré, M. Charalambides, D. Lopez, Management and orchestration challenges in network functions virtualization, IEEE Commun. Mag. 54 (1) (2016) 98–105.

[51] D. Bernstein, Containers and cloud: From LXC to docker to Kubernetes, IEEE Cloud Comput. 1 (3) (2014) 81–84.

[52] M. Ersue, ETSI NFV management and orchestration-An overview, in: Proc. of 88th IETF Meeting, 2013.

[53] ETSI, Open source MANO, 2016, OSM home page - https://Osm.Etsi.Org/.

[54] A. Mayoral, R. Vilalta, R. Casellas, R. Martinez, R. Munoz, Multi-tenant 5G network slicing architecture with dynamic deployment of virtualized tenant management and orchestration, MANO instances, in: ECOC 2016; 42nd European Conference on Optical Communication, 2016, pp. 1–3.

[55] G.A. Carella, T. Magedanz, Open baton: A framework for virtual network function management and orchestration for emerging software-based 5G networks, Newsletter 2016 (2015) 190.

[56] C. Anderson, Docker [software engineering], IEEE Softw. 32 (3) (2015) 102–c3.

[57] S. Picozzi, M. Hepburn, N. O'Connor, DevOps with Openshift: Cloud Deployments Made Easy, O'Reilly Media, Inc., 2017.

[58] M. Frampton, Apache mesos, in: Complete Guide to Open Source Big Data Stack, Springer, 2018, pp. 97–137.

[59] R. MacDavid, P. Lin, B. Padmanabhan, A. ThakuR, L. Peterson, J. Rexford, O. Sunay, A P4-based 5G user plane function, in: SOSR '21: Proceedings of the ACM SIGCOMM Symposium on SDN Research, SOSR, 2021, pp. 162—168.

[60] F. Kaltenberger, H. Wang, S. Velumani, Performance evaluation of offloading LDPC decoding to an FPGA in 5G baseband processing, in: 25th International ITG Workshop on Smart Antennas, 2021, pp. 1–4.

[61] XILINX demos at MWC 2022, [Online], https://www.xilinx.com/about/events/2022/mwc-2022.html.

[62] NVIDIA AERIAL SDK - Build and deploy GPU-accelerated 5G virtual radio access networks (vRAN), [Online], https://developer.nvidia.com/aerial-sdk.

[63] OPEN NFV edge project , [Online], https://wiki.opnfv.org/display/EC/Edge+cloud.

[64] ONAP Project: Edge automation through ONAP, [Online], https://wiki.onap.org/display/DW/Edge+Automation+through+ONAP.

[65] OpenStack project: Edge computing, [Online], https://www.openstack.org/edge-computing/.

[66] LF EDGE project: Building an open source framework for the edge, [Online], https://www.lfedge.org/.

[67] 5G City Project, [Online], https://www.5g.eu/.

[68] Openness Project, [Online], https://www.openness.org/.

[69] SYMEC Project, [Online], https://www.symec.com.pl/.

[70] Linux foundation projects: OPNFV/Anuket, [Online], https://www.opnfv.org/.

[71] Cloud Native Computing Foundation (CNCF), [Online], https://www.cncf.io/.

[72] Open Compute Project Foundation (OCP), [Online], https://www.opencompute.org/.

[73] G. Bernini, P.G. Giardina, S. Spadaro, F. Agraz, A. Pagès, J. Cabaça, P. Neves, K. Koutsopoulos, A. Matencio, Multi-domain orchestration of 5G vertical services and network slices, in: 2020 IEEE International Conference on Communications Workshops, ICC Workshops, IEEE, 2020, pp. 1–6.

[74] T. Rakotoarivelo, M. Ott, G. Jourjon, I. Seskar, OMF: A control and management framework for networking testbeds, Oper. Syst. Rev. 43 (4) (2010) 54–59.

[75] G. ETSI, 013: Network functions virtualisation, NFV; Management and orchestration; Os-Ma-Nfvo reference point–Interface and information model specification.

[76] L. Baron, R. Klacza, P. Gaudet-Chardonnet, A. Bradai, C. Scognamiglio, S. Fdida, Next generation portal for federated testbeds MySlice V2: From prototype to production, in: Fed4FIRE Engineering Conference - FEC2, 2017, poster URL https://hal.archives-ouvertes.fr/hal-01804013.

[77] EU's open science policy, [Online], https://ec.europa.eu/info/research-and-innovation/strategy/strategy-2020-2024/our-digital-future/open-science_en.

[78] EU support for open access, [Online], https://ec.europa.eu/info/research-and-innovation/strategy/strategy-2020-2024/our-digital-future/open-science/open-access.

[79] Open Research Data Pilot of the European Commission, 2017, [Online], https://www.openaire.eu/what-is-the-open-research-data-pilot.

[80] B. Mons, C. Neylon, J. Velterop, M. Dumontier, L.O.B. da Silva Santos, M.D. Wilkinson, Cloudy, increasingly FAIR; Revisiting the FAIR data guiding principles for the European open science cloud, Inf. Serv. Use 37 (1) (2017) 49–56.

[81] EOSC FAIRsFAIR project, [Online], https://www.fairsfair.eu/.

[82] EOSC GO FAIR initiative, [Online], https://www.go-fair.org/.

[83] RDA FAIR Data Maturity Model Working Group, et al., FAIR Data Maturity Model: Specification and Guidelines, 10, Research Data Alliance, 2020.

[84] M.D. Wilkinson, M. Dumontier, I.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L.B. da Silva Santos, P.E. Bourne, et al., The FAIR guiding principles for scientific data management and stewardship, Sci. Data 3 (1) (2016) 1–9.

[85] P. Ayris, J.-Y. Berthou, R. Bruce, S. Lindstaedt, A. Monreale, B. Mons, Y. Murayama, C. Södergård, K. Tochtermann, R. Wilkinson, Realising the European Open Science Cloud, European Union, 2016.

[86] O. Corcho, M. Eriksson, K. Kurowski, M. Ojsteršek, M. van de Sanden, F. Coppens, EOSC Interoperability Framework: Report From the EOSC Executive Board Working Groups FAIR and Architecture, European Commission, 2021.

[87] M.A. Wimmer, R. Boneva, D. Di Giacomo, Interoperability governance: A definition and insights from case studies in Europe, in: Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age, 2018, pp. 1–11.

[88] K. Joshi, Y. Demchenko, P. Andreou, et al., SLICES design study D4.2: SLICES infrastructure and services integration with EOSC and open science (initial proposal), 2021, [Online], http://slices-ds.eu/wp-content/uploads/2021/12/SLICES-DS_D4.2.pdf.

[89] ACM Artifact Review and Badging, [Online], https://www.acm.org/publications/policies/artifact-review-badging.