Sustainable Architecture Design Principles for Large Scale Research Infrastructure Projects

Yuri Demchenko
Complex Cyber Infrastructure Lab
University of Amsterdam
Amsterdam, The Netherlands
email: y.demchenko@uva.nl

***Abstract*—Modern data driven science extensively uses digital technologies and requires large scale Research Infrastructures (RI) combining computational, storage and data management facilities to support data collection, processing and scientific workflow management. The success of building dedicated RIs and their efficient operation will depend on adopting modern technologies and well-defined architectures that support the sustainability and long term evolution of technical solutions. Building and operating modern data driven RIs and scientific instruments require a wide spectrum of competences and knowledge related to modern technologies, system and software engineering (SSE), including Big Data infrastructure engineering. The paper provides background information on large European RI and e-Infrastructure projects and analyses different cases/scenarios where the availability of necessary system and software engineering competences and knowledge are critical for the success of RI. The paper proposes sustainable architecture design principles that could provide both guidance for RI technical design and targeted training for engineering and scientific personnel. The paper presents the author's experiences in implementing the proposed design principles in two education courses.**

Keywords— Data Driven Science, Research Infrastructure, Sustainable Architecture Design Principles, System and Software Engineering Competences and Skills, Education and Training Methodology.

I. INTRODUCTION

Modern science is data driven, growingly digitalized, and extensively uses digital infrastructure, Big Data and Data Science Analytics technologies. It requires large scale Research Infrastructures (RI) combining computational, storage and data management facilities to support data collection, processing and scientific workflow management. Europe has well established policies and practices in building and operating dedicated domain specific RIs coordinated by ESFRI (European Strategy Forum on Research Infrastructures), which publishes a tri-annual Roadmap document that analyses the status and trends in European RIs development and successful operation. Traditionally, demand for computing, network and data storage infrastructure has been supported by e-Infrastructures that serve all European research community such as GEANT, EGI, EuroHPC, PRACE, EUDAT. Recent ESFRI Roadmap 2021 [1] included a new domain DIGIT to support research and experimentation with digital infrastructure technologies, where the SLICES-RI is dedicated to a wide range of digital technologies from 5G/RAN to IoT/Edge and cloud computing, and AI powered systems [2].

Many scientific domains require dedicated RIs that can be created as virtual digital RI on top of existing e-Infrastructures. Efficiency of such RIs will depend on adopting modern technologies and well-defined architecture that support the sustainability and long term evolution of technical solutions. Building and operating modern data driven RIs and scientific instruments require a wide spectrum of competences and knowledge related to modern technologies, system and software engineering (SSE).

The paper proposes sustainable architecture design principles (SADP) that could provide both guidance for RI technical design and targeted training for engineering and scientific personnel training. The suggested principles are based on important architectures and standards that define the modern digital infrastructure, Internet, information and communication technologies (ICT), and a wide range of system and software engineering practices. The paper also provides an overview of the key standardization frameworks that constitute the proposed SADP and are important to know and use by RI and application development teams.

The paper discusses the importance of addressing necessary competences and skills for successful RI development and operation throughout the whole RI lifecycle. This is motivated by continuous growth of the complexity of cloud based data centric applications that require new skills from research engineers that span beyond just research software engineering or scientific programming. The paper analyses a few cases and scenarios where the availability of necessary system and software engineering competences and knowledge are critical for the success of RI.

As another example, it is a common practice to use for scientific programming, such developer-friendly platforms as Jupyter Notebook for Python. However, when it comes to the full scale deployment of the debugged scientific workflows

or trained ML models, the developers or engineers need to port them to cloud based Big Data platforms in the production environment. So, they need to be familiar with Big Data technologies and tools. In the past time, it was a task of the release team, but now, with the adoption of the DevOps practice and a variety of CI/CD tools, deployment is done as a part of the DevOps process by the same developer team. This is another area where the developer teams need to have the necessary expertise.

The paper presents the author's experiences in addressing necessary SSE competences in both professional training for the SLICES-SC [3] and SLICES-PP [4] projects and in university education. The paper refers to SLICES Summer School [5] and two inter-connected courses, the Big Data Infrastructure Technologies for Data Analytics (BDIT4DA) [6] and the DevOps for Software Engineering (DevOps4SE) [7], that include necessary competences and learning outcomes required for building modern data-driven infrastructures and applications. The paper shares the experience of how both courses have evolved and adapted to the continuous technology development and specific needs of the developers' teams.

The paper is organised as follows. Section II provides background information on the large European RI and e-Infrastructure projects, and also introduces the SLICES-RI. Section III is devoted to the analysis of the different cases and scenarios where the availability of necessary system and software engineering competences and knowledge are critical for the success of RI projects. Section IV presents the proposed Sustainable Architecture Design Principles and related issues. Section V discusses the importance of addressing data management issues and provides example requirements to SLICES Data Management Infrastructure. Section VI describes how key SSE competences are addressed in two courses, BDIT4DA and DevOps4SE, taught by the author in different education environments and formats. The conclusion in Section VII explains the author's motivation in sharing experience and describes ongoing efforts to support professional approaches in research infrastructure services and applications developments.

II. European Initiatives to Support Research

A. European Open Science Cloud (EOSC)

Europe traditionally supported creating Research Infrastructures (RI) and e-Infrastructures to support research in different thematic domains. EOSC is an initiative and programme by the European Union to provide European researchers, innovators, companies and citizens with a federated and open multi-disciplinary environment where they can publish, find and re-use data, tools and services for research, innovation and educational purposes [8]. The EOSC Strategic Research and Innovation Agenda (SRIA) provides a roadmap to achieve the EOSC vision and objectives, namely to deliver an operational "Web of FAIR data and services" for science [9], which will technically require creating Federated Research Data Management Infrastructure. Ongoing developments aim at providing a customisable research environment for researchers and research projects using services provided by the EOSC Portal Catalog and Marketplace [10].

Looking retrospectively at ERA and EOSC development since 2016, we can refer to the author's paper from 2020 [11] that provided the analysis of the European RI evolution stages in the context of the core Internet, networking and cloud technologies development. This paper identified the adoption of some key competences that would facilitate EOSC development as an effective instrument for European research and for science-industry cooperation. Currently, EOSC has advanced in creating an effective data management framework, but its potential is limited by a lack of highly effective researcher-centric infrastructure providing the same level of usability as public cloud service providers. EOSC building has been funded by dedicated EU Horizon 2020 and Horizon Europe projects that consolidate experience from multiple research and technology domains and communities. Adopting industry experience in sustainable infrastructure engineering and operation would further facilitate RI and EOSC development, however this will require the necessary competences and skills in infrastructure and applications engineering.

B. SLICES Research Infrastructure

The Scientific Large-scale Infrastructure for Computing/ Communication Experimental Studies (SLICES) [2, 12] is a distributed Digital Infrastructure designed to support large-scale experimental research focused on networking protocols, radio technologies, services, data collection, parallel and distributed computing and, in particular, cloud and edge-based computing architectures and services. This encompasses the full range of network, computing, and storage functions required for on-demand services across many verticals and addresses new complex research challenges, supporting disruptive science in IoT, networks and distributed systems. SLICES will integrate multiple experimental facilities and testbeds operated by partners, providing a common services access and integration platform. SLICES will allow academics and industry to experiment and test the spectrum of digital technologies whereby the computing, network, storage, and IoT resources can be combined to design, experiment, operate, and automate the full research lifecycle.

SLICES-RI will use EOSC experience and infrastructure services supporting Research Data Management for data sharing and access. SLICES-RI Data Management Infrastructure is being designed to be federated with EOSC Federated Research Data Management Infrastructure.

SLICES-RI related projects SLICES-SC and SLICES-PP recognized the importance of addressing necessary competences and skills both for RI developers and operators and for researchers by establishing SLICES Summer School, which is intended to evolve to SLICES Academy. Addressing practical and staffing aspects, bridging necessary competences and skills gaps may be challenging in conditions of the strong competition for the top talents with industry and business. The primary goal of developing

proposed Sustainable Architecture Design principles is to facilitate building the necessary capacity among SLICES staff and researchers.

III. Role of Architect and System Engineer in Successful Scientific Projects

A. User stories and team operation in complex RI projects developments

This section will describe some real life scenarios and stories of complex RI development that illustrate the importance of proper team composition (containing key competences, knowledge, experience), leadership, and architecture and technology selection. All examples are "distilled" from the real projects that the author actively contributed or cooperated, however project names are not provided for privacy protection reasons.

Case 1 (reference or successful). Development of cloud based applications for research and industry (Innovation Action).

The team is composed of specialists from all relevant domains (infrastructure, network, DevOps/Agile, security, software development, use cases domain) with sufficient experience in system and software engineering at different levels (architecture, requirement engineering, integration).

The team adopted DevOps Agile Scrum methodology for distributed teams coordination: two months sprints, weekly standups, and regular demo sessions, - including a simple github based Scrum management tool. All team members were involved in all aspects of the project development, although at different levels and responsibilities: from leading and direct responsibility to awareness and notification. The team used active knowledge transfer and experience sharing, all team members were willing to learn new domains, technologies and tools what helped team cohesion. The leadership was supported by effective collaboration and team members' initiative.

The development process started with the first set of use cases analysis and requirements specification, followed by the design and development stage, driven by CI/CD process.

The project developed multilayer architecture with well defined services at each layer what allowed easy integration of available services implementation and faster development of new specialist services.

The project developed products that became a part of the service offering of the SME partner. All component products/applications developed by partners were re-used in other projects. Use cases implemented and used by respective communities.

Case 2 (not sufficient system engineering expertise). Development of the complex research infrastructure to support experimentation on digital infrastructure technologies.

The team is composed of researchers in specific domain technologies, such high performance and optical networking, data management, with strong experience in building domain specific applications and testbeds.

The team effectively performed in developing domain specific testbeds and infrastructure services, however experienced difficulties in delivering consistent project architecture addressing short term and long term goals.

Such a project would require, besides domain specific expertise, also strong expertise in general system engineering, software development, that would facilitate the use of best practices in the consistent architecture definition and further project development. The expertise in project management and development practices such as DevOps, CI/CD, and Agile scrum or Kanban will facilitate the whole project development.

Case 3 (lack of system engineering and infrastructure expertise). This is the often case when the project requires delivering digital infrastructure to support specific domain research, but the project consortium partnership lacks partners in computer or infrastructure technologies, system and software engineering.

The project development team invites computer or infrastructure specialists who often have the necessary practical applications development expertise but limited knowledge in system engineering and (large) project development. The consortium configuration doesn't allow gathering a critical mass of infrastructure or applications developers to successfully deliver operational services. However, this should not excuse using standard project or architecture development process that should include use cases analysis, requirements engineering, design, development and deployment. The outcome of such a project may be limited to demonstration or services delivered at TRL3. Building further development to higher TRL4-TRL7 on such results would be problematic if the presented results are not based on the best practices in architecture design and system implementation. Services and infrastructure might need to be redesigned based on the well defined architecture, operational model and blueprint.

Case 4 (not sufficient knowledge of background technologies). This scenario can be a variation of scenarios 2 or 3. The project architect (or person responsible for the architecture development) has experience from the previous successful RI or e-Infrastructure projects of 5-8 years ago when performing in a junior position or specific task leader. Possessed experience is very important, but acting in the role of system architect requires knowledge of all background technologies that contribute to the architecture definition. The risk is if the project architect approaches the overall project architecture development from the point of specific research tasks, in contrary to designing the system or service for a wider community of users and contributors. Sustainable architecture decisions will help organize and coordinate the contribution of many developers with different domain expertise. Furthermore, in the conditions of fast technology development, core architecture and design decisions must be regularly revisited and verified with the development and evolution of new technologies.

Knowledge of the ongoing standardization in key technology areas is essential. This is especially related to the

cloud/edge/IoT technologies that have evolved from standalone solutions to whole ecosystems that potentially can provide an integrated development environment and interoperable solutions for specific use cases.

B. *Essential Knowledge and Competences in Architecture design and System engineering*

Based on a critical analysis of the above scenarios, we can identify the following essential knowledge and competences that are required for successful architecture design and infrastructure project implementation. This is further presented as sustainable architecture design principles summarized in the next section. The following knowledge, competences and experience are essential in modern digital infrastructures and applications:

For Architects and team managers

- System and Software Engineering principles and best practices, in particular for data driven and user facing services.
- Cloud based and cloud native services design methods and familiarity with popular Open Source and public cloud platforms
- Knowledge of standards in the area of ICT and computer technologies and those related to architecture design.

For application developers and scientific programmers:

- Working experience with popular programming languages (Python, Java, C, others) and web application development frameworks.
- Highly beneficial knowledge and experience with the DevOps Agile Scrum or Kanban
- DevOps CI/CD tools to support automated deployment (GitHub, Ansible, Terraform).

C. *Applying Sustainable Architecture Design Principles for Infrastructure Project Development*

In this section, we provide a few scenarios that may happen if the project development involves multi-disciplinary teams with different backgrounds and experiences, in cases that require additional knowledge or expertise. We put these examples immediately after the cases description and summary of essential competences and skills to show links between them, however it also uses some concepts described in detail in the next session on the Sustainable Architecture Design principles.

Scenario 1 (Structural architecture driven design). The project team has sufficient experience in domain specific systems from the previous work or projects, but the new project requires wider expertise. The team starts with implementing available solutions that address the main goals of the system or infrastructure, defines the overall architecture, and requirements to the core components and to other or external components for which the team doesn't possess the necessary expertise. To solve the problem of the expertise gap, the team (actually the team leader) looks for cooperation with other projects and invites experts to share expertise, A dedicated team member is assigned to gain new expertise and take over necessary components implementation in the future.

Scenario 2 (Designing Up, Down, and Out): System implementation is started without consistent and grounded architecture definition but with a large availability of legacy components and sufficient expertise in ICT systems design from the previous project. The project or team starts with implementing some infrastructure and services islands, often siloed services, based on available expertise and already existing components. The project or team should proceed with the development and implementation but, at the same time, start the structural analysis of the services being developed and put them in the context of the future system development and integration with other components based on the general sustainable architecture design principles explained above, for example, split siloed services into layered components, apply multi-tier design, define API between layers, tiers and components that may be distributed or external third party.

Scenario 3 (ad-hoc services piloting). The project is started by the community with identified needs for information digital infrastructure services but without prior experience in building such services, for example, building research infrastructure for social or environmental sciences, humanities. The project may succeed in user needs studies and defining user requirements, but it may fall short of transforming user requirements to technical system requirements and corresponding architecture and functional design. The project may end up with the pilot services for demonstration of the proof-of-concept and stand-alone tools but further successful development will require a professional approach in infrastructure design, what in its turn may also require services re-design.

In all cases and scenarios, familiarity with the DevOps and Agile Scrum or Kanban practices would be highly beneficial to facilitate development and better organize teamwork. It is important here to adopt the concept of the Minimum Viable Product (MVP) and DevOps methodology to progress from MVP to full scale implementation.

In general and as a demand of time, the projects will benefit from using design templates and learning the cloud-based and cloud-native design approaches that effectively use composable services and design templates that are supported by a variety of deployment automation tools such as AWS CloudFormation, Ansible, Terraform, others.

IV. Sustainable Architecture Design Principles and Implementation Aspects

Architecture provides a blueprint for systems and applications development and should ensure staged development and future sustainable system evolution.

A. *Sustainable Architecture Design Principles*

Sustainable architecture design principles provide recommendations and guidance for the evolutional approach in designing and implementing complex infrastructure projects. The complexity of modern systems and applications

requires knowledge and competences in multiple technology and computer domains. The sustainable architecture intends to achieve lowering infrastructure or services resources, energy and waste along the whole service or infrastructure lifecycle, including supply chain, upgrade, replacement, and decommissioning.

The following principles are derived from the existing architecture frameworks for the main structural and infrastructure components comprising modern digital and data infrastructures such as Internet architecture (in particular TCP/IP architecture), Telecom OSI model and related ITU-T standards, TeleManagement Forum, related ISO and IEEE standards, 5G/6G related standards, and others. This is also supported by the author's experience from different research and development projects as well as university teaching and professional training.

General architecture design principles

- Layered architecture design for services and mechanisms, including inter-layer interfaces, including cross-layer services and mechanisms definition that are typically defined as service planes, for example, management plane, security plane, data management plane.
- Multi-tier services and infrastructure design, including combined multi-layer and multi-tier systems that may use or apply different architectural and layered solutions.
- Application Programming Interfaces (API) for composable services that must be supported by consistent (and fully qualified API metadata and namespaces definition).

Service architecture related

- Service Oriented Architecture (SOA) and Microservices Architecture (MSA) that is supported with the different VM and container solutions and/or platforms.
- Cloud powered, cloud based and cloud native design principles that require knowledge of the modern cloud architecture and cloud platform, both Open Source and public clouds (at least Amazon Web Services, Microsoft Azure, and Google Cloud Platform). This also includes such powerful cloud based mechanism as Virtual Private Cloud (VPC) that provide VPN based secure environment for multi-tenant customer applications.
- Service lifecycle management model that should include all necessary services to support lifecycle stages in the context of specific services. This also includes services composition and orchestration for services deployment and operation.

Data infrastructure and services related

- Big Data computation models and supporting platforms, distributed and highly scalable systems, in particular, Hadoop ecosystem and NoSQL databases.
- Data management infrastructure and services that should cover both: services data (related to the management plane) and business or research data produced as a result of business operations or scientific research.
- Services and data management continuity in IoT/sensor networks, edge, cloud, data-driven applications that also include 5G/6G Radio Access Network (RAN), edge and cloud convergence.

Security and compliance design principles

- Security architecture and security services lifecycle management which are well defined by numerous standards and supported by the major infrastructure development frameworks; also security services have their own multi-layer architecture (can also be referred as security plane), their integration with the main infrastructure services, including data infrastructure) is realized via API calls and consistent definitions of the security roles, access control policies and credentials and secrets management.
- Compliance frameworks that define requirements to and recommendations for secure services and infrastructure design and operation. Cloud Security Alliance (CSA) and Compliance Assessment Initiative Questionnaire (CAIQ) provide the best overview of all important standards and regulations to ensure systems security and compliance, and data protection.

Project Management and DevOps

- DevOps and SRE (Site Reliability Engineering) practices applied to system and services engineering and operation. This should also include continuous monitoring and optimisation on multiple user -centric and business-centric SLI/KPI (Service Level/Key Performance Indicators).
- DevSecOps that extended the DevOps model and practices by addressing security aspects during the whole system/services lifecycle, intending to address "Security by Design" concept (however not yet fully developed)
- General compliance with the project management principles, models and procedures applied to infrastructure, services, and data handling and analytics.

A wider scope of architecture design principles can be found in standards and recommendations related to enterprise architecture design, such as the NIST Enterprise Architecture Model (EAM) [13], which divides the architecture description into domains, layers, views, and offers perspectives models. A similar approach is used in TOGAF architecture definition [14]. This provides a framework and a tool for the systemic design approach and decisions on the different components of the system. This also paves a way for making long-term decisions about design requirements, sustainability, and system or services evolution. Guidelines are provided in both documents/frameworks.

B. Important overloaded terms and concepts

It is important to maintain consistent terminology and definition of all architecture-defining components in modern converged systems, which, to a large extent, is ensured by modern standardization system (refer to the standard bodies listed above). Common understanding and correct/unambiguous use of domain related terminology is important for effective communication between developers and researchers communication.

Based on our experience, the following are examples of some overloaded and domain or context specific terms that

may create confusion and misunderstanding between developers with different technical (and educational) backgrounds (we don't provide references to listed below concepts and terms, leaving it to the readers to explore):

- Architecture concept is itself often understood in different ways by researchers and developers with different backgrounds and experiences. The best way to achieve homogeneity in understanding architecture and its design principle is to learn the architecture examples and templates in Internet, telecom and web based technologies that proved their efficiency in guiding technology development and progress. A wider understanding and vision of the architecture concepts can be gained with TOGAF, which provides a recipe for general enterprise architecture design and links technical design with the business and mission goals.
- Architecture, reference architecture, framework, reference model: all these terms are in many cases interchangeable, often used together and in combination, but have their own meanings in specific contexts. Understanding this may help avoid confusion between developers and teams at the different stages of projects.
- Architecture diagram, architecture model, functional diagram, process or sequence diagram: these are very useful design and presentation tools but their use should not be taken out of context or replace structural architecture definition.
- Multi-layer and multi-tier systems in system and applications engineering, and multi-level systems in security engineering.
- Blueprint and Bill of Materials (BOM) as general concepts and similarly named concepts in DevOps and CI/CD in software engineering.
- Metadata and data modeling definitions as they are defined in industry and research data management domains against information models and metadata in telecom and Internet service management.
- Security as a general concept and those related to different security domains such as computer security, network/ Internet security, application security, cryptography, hardware security, operational security, access control, identity management and trust management. This is also related to the specifics of the security models: host-centric, service-centric, and data-centric.

It may take some time for all cooperating team members to come to a common terminology understanding and domain context, but this process can be accelerated with introductory training. When adopting or introducing specific terminology, it is important to verify the definition with the corresponding standards, and document internally what original concepts are retained and what are not retained.

V. Data Management and Governance Aspects

A. Data Management Infrastructure

Data Management Infrastructure is an essential component of the modern RIs (as it is being implemented in SLICES-RI) must support all stages of the research data lifecycle [12, 15] that typically include data collection from experiments (including experiment description and measurement data), data storage, data preparation, data lineage and quality assurance, data publication, and data sharing.

DMI Architecture definition includes hierarchical service layers (allowing horizontal and vertical composition and integration) and cross-layer services defined as planes. Such architecture definition allows separating data management and governance functions, concerns and actors/roles. The following service layers and planes are defined:

Layer 5 – Virtual Research Environment (VRE) and researcher portal or dashboard.

Layer 4 - Experiment configuration and management.

Layer 3 - Experimental data collection/recording that applies data models and metadata for experimental data.

Layer 2 - Data processing that performs data analysis, allows ML model building for processes and systems-under-test, and ensures the computation workflow scalability and portability.

Layer 1 - Data Storage, Archiving, Exchange that represents the physical or virtual infrastructure resources for data or metadata storage, archiving and publication. This layer supports FAIR Digital Object (FDO), PID registries and gateway/proxy.

Data Management Plane includes (cross-layer) Data Management Services and Tools that can be used by each of the DMI layers:

- Data Management Plan and Data Quality Assurance, FAIR compliance,
- Metadata registries and tools,
- Data Governance Policy, Data Security, GDPR compliance.

B. SLICES DMI Requirements

This section provides a practical example of defining requirements to SLICES Data Management Infrastructure to support efficient experimental data management. The following requirements are derived from the best practices and use cases analysis in the SLICES-DS project [15, 16]:

RDM1. Distributed data storage and experimental data(set) repositories should support common data and metadata interoperability standards, in particular, common data and metadata formats. Outsourcing of data storage to the cloud must be protected with appropriate access control and compliant with the SLICES Data Management policies.

RDM2. SLICES DMI should support the whole research data lifecycle. It should provide interfaces to experiment workflow and staging.

RDM3. SLICES DMI shall provide PID (Persistent IDentifier) and FDO (FAIR Digital Object) registration and resolution services to support linked data and data discovery that should be integrated with EOSC services.

RDM4. SLICES DMI must support (trusted) data exchange and transfer protocols that allow policy-based access control to comply with the data protection regulations.

RDM5. SLICES DMI must enforce user and application access control and identity management policies adopted by

the SLICES community that can be potentially federated with the EOSC Federated AAI.

RDM6. Procedures and policies must be implemented for data curation and quality assurance.

RDM7. Certification of data and metadata repositories should be considered at some maturity level following certification and maturity recommendations by RDA.

VI. SADP IN EDUCATION AND TRAINING

This section provides information on how the proposed SADP are implemented in two interconnected courses that are designed to provide a strong foundation for the System and Software Engineering competences as they are required for the design, deployment and operation of modern research infrastructure and services: Big Data Infrastructure Technologies for Data Analytics (BDIT4DA) [6] and DevOps and cloud based Software Engineering (DevOps4SE) [7]. Both courses have similar structure and organization but are targeted at different academic programs and primary groups of practitioners. The courses include lectures, practice, labs, group projects, and literature study and seminars, and use one of the popular public cloud platforms AWS, Azure, or Google Cloud for educational purposes and educational project development what allows the students to gain important experience for better workplace integration.

A. Big Data Infrastructure Technologies for Data Analytics (BDIT4DA)

Big Data Infrastructure Technologies for Data Analytics (BDIT4DA) is the original course developed by the authors for Data Science masters and has the main goal to provide the students with sufficient knowledge for implementing data analytics projects using cloud based Big Data platforms and development environments. The BDIT4DA course implements recommendations of the Data Science Engineering (DSENG) Body of Knowledge and Model Curriculum, which are part of the EDISON Data Science Framework (EDSF) [17, 18]. The detailed description of BDIT4DA course is given in the author's earlier publication from 2019 [6]. Since that time, the course has undergone continuous development adopting Big Data technologies development and availability of the cloud based education platforms. On the content side, the course implemented most of the SADP principles both in lecturing and course project guidelines.

Currently, the course includes 10 basic modules that include both lectures and labs or practice. The first four modules provide the foundation of Big Data technologies and other modules are strongly oriented on the practical infrastructure and applications design. Since 2021 term, the course includes a new Module 8 on Data Science Projects Management and DataOps/MLOps [19]. Two new modules were added in 2022: Module 9 on AWS SageMaker platform for Data Analytics and ML projects development, and Module 10 on Cloud based Architecture design patterns, which will be extended with the discussed in this paper SADP and cloud based design patterns.

The following are modules included and taught in the BDIT4DA course:

Module 1: Introduction to the course. Cloud Computing foundation. Cloud service models, cloud resources.

Module 2: Big Data architecture framework, cloud based Big Data services and platforms.

Module 3: Big Data Algorithms: MapReduce, Pregel. Hadoop platform and components for Big Data analytics: HDFS, YARN, MapReduce, HBase, Pig, Hive, others.

Module 4: Data Streams and Streaming Analytics. Kafka, Flume. Spark architecture and popular Spark platforms, DataBricks.

Module 5: SQL and NoSQL Databases. CAP Theorem. Modern large scale databases AWS Aurora, Azure CosmosDB, Google Spanner.

Module 6: Data Management and Governance. Research Data Management and FAIR data principles in data management.

Module 7: Big Data Security and Compliance. Cloud compliance standards and cloud provider services assessment.

Module 8: Managing Data Science Projects. Research methods and project organisation. Data Science Process Models, DataOps and MLOps.

Module 9: Platforms and tools for Data Analytics and NL pipeline automation (such as AWS SageMaker or Azure MLOps, supported with Data Lakes)

Module 10: Sustainable Architecture Design principles and cloud based design patterns, which will be extended with discussed in this paper and design patterns.

Depending on the program configuration and scheduling, the necessary subset and configuration of modules can be selected. Modules can be delivered in the form of sessions that can combine lectures (2-3 hrs), practice (2-4 hrs) which can be slit on smaller 2-3 lessons. and interactive activities such as literature review, project progress presentation. The modules are developed in such a way that they can be re-used in other courses or for targeted training or workshop such as summer schools or conference tutorials.

B. DevOps and Cloud based Software Engineering (DevOps4SE)

The DevOps4SE course includes two general types of modules: (1) technology related that provide systematic information about DevOps technologies, design principles, tools and extended information about selected cloud platforms that are supported by practice and labs; (2) use cases, and case studies, practices, where real life projects are used as examples.

The technology related modules follow the main topics (Knowledge Units) of the proposed DevOpsSE Body of Knowledge [7, 19]. Following positive experience from the BDIT4DA course and responding to the students' interest, the following new topics/modules will be added in the new term 2023/2024:

- DevSecOps; Secure Software Development Lifecycle Management (SDLM); cloud based tools for secure software development and testing.

- Data Science project management and DataOps/MLOps processes and platforms;
- Sustainable Architecture Design principles and cloud based design patterns.

C. Importance of Dedicated Data Management Training

Data Management and Governance (DMG) and corresponding infrastructure services are an integral part of modern RIs [20, 21]. Corresponding courses and training modules must be included in the SSE related educational courses, special training needs to be provided to the RI personnel and researchers. DMG and RDM topics are included in both courses and provided as recurrent training in SLICES Summer School, in particular, the following topics are included: FAIR data principles (data must be Findable, Accessible, Interoperable, Reusable) [22], Data Management Plan (DMP), data and metadata publication, data modelling and metadata definition [23].

VII. Conclusion

This paper presents the author's long-time experience in designing and developing different infrastructure components and services for research infrastructures in the framework of multiple projects funded by European research programs. The author's experience developed from applications developer to architect and project coordinator, changing the research domain as the technologies evolve from Internet and web applications to collaborative systems, computer grids and clouds, Big Data, data centric technologies, and research data management. Personal experience in mastering new technologies, provided a solid foundation for developing educational materials for university teaching and professional training on the technologies mentioned above.

The author believes that the presented paper will provide a basis for discussion on how to address the growing complexity of modern digital infrastructures and the increased demand for new competences and skills to ensure sustainable development and operation of modern and future Research Infrastructures.

Acknowledgment

The research leading to these results has received funding from current Horizon Europe projects SLICES-DS (951850), SLICES-PP (951850), GreenDIGIT.

References

[1] ESFRI Roadmap 2021 [online] https://roadmap2021.esfri.eu/media/1295/esfri-roadmap-2021.pdf

[2] SLICES-RI [online] https://www.slices-ri.eu/

[3] SLICES-SC [online] https://slices-sc.eu/#

[4] SLICES-PP [online] https://slices-pp.eu/#

[5] SLICES Summer Schoul 2023. Slices Academy [online] https://moocs-academy.slices-ri.eu/

[6] Big Data Platforms and Tools for Data Analytics in the Data Science Engineering Curriculum, Proc 2019 3rd International conference on Cloud and Big Data (ICCBDC 2019), August 28-30, 2019, Oxford, UK

[7] Yuri Demchenko, Z.Zhao, S.Koulouzis, J.Surbiryala, Z.Shi, X.Liao, J.Gordiyenko, Teaching DevOps and Cloud based Software Engineering in University Curricula, Proc. 5th IEEE STC CC Workshop on Curricula and Teaching Methods in Cloud Computing, Big Data, and Data Science (DTW2019), the eScience 2019 Conference, September 24 – 27, 2019, San Diego, California, USA

[8] EOSC (European Open Science Cloud) Association [online] https://eosc.eu/

[9] EOSC Strategic and Research Innovation Agenda (SRIA) [online] https://www.eosc.eu/sites/default/files/EOSC-SRIA-V1.0_15Feb2021.pdf

[10] EOSC Portal [online] https://eosc-portal.eu/

[11] Yuri Demchenko, Cees de Laat, Wouter Los, Future Scientific Data Infrastructure: Towards Platform Research Infrastructure as a Service (PRIaaS), Proc. The International Conference on High Performance Computing and Simulation (HPCS 2020), 10-14 Dec 2020, Virtual.

[12] Serge Fdida, Nikos Makris, Thanasis Korakis, Raffaele Bruno, Andrea Passarella, Panayiotis Andreou, Bartosz Belter, Cedric Crettaz, Walid Dabbous, Yuri Demchenko, Raymond Knopp, SLICES, a scientific instrument for the networking community, Computer Communications, 2022, ISSN 0140-3664, https://doi.org/10.1016/j.comcom.2022.07.019.

[13] NIST Federal Enterprise Architecture Framework [online] https://obamawhitehouse.archives.gov/sites/default/files/omb/assets/egov_docs/fea_v2.pdf

[14] TOGAF Enterprise Architecture [online] https://www.opengroup.org/togaf

[15] Demchenko, Y., S. Gallenmüller, S. Fdida, P. Andreou, C. Cretaz, M. Kirkeng, Experimental Research Reproducibility and Experiment Workflow Management. TASIR Workshop, Proc. COMSNETS 2023 Conf. 3-8 January 2023, Bengaluru, India

[16] Deliverable D4.5 SLICES infrastructure and services integration with EOSC, Open Science and FAIR: Recommendations and design patterns (final report)/. SLICES-DS Project. 31 August 2022.

[17] EDISON Data Science Framework (EDSF). [online] Available at https://github.com/EDISONcommunity/EDSF

[18] The Data Science Framework, A View from the EDISON Project, Editors Juan J. Cuadrado-Gallego, Yuri Demchenko, Springer Nature Switzerland AG 2020, ISBN 978-3-030-51022-0, ISBN 978-3-030-51023-7

[19] Yuri Demchenko, From DevOps to DataOps: Cloud based Software Development and Deployment, Proc. The International Conference on High Performance Computing and Simulation (HPCS 2020), 10-14 Dec 2020, Virtual.

[20] Data Management Body of Knowledge (DM-BoK) by Data Management Association International (DAMAI) [online] http://www.dama.org/sites/default/files/download/DAMA-DMBOK2-Framework-V2-20140317-FINAL.pdf

[21] Data Management Maturity Model (DMM), CMMI Institute, 2018 [online] https://cmminstitute.com/data-management-maturity

[22] Turning FAIR into reality. Final Report and Action Plan from the European Commission Expert Group on FAIR Data, 2018 [online] https://ec.europa.eu/info/sites/info/files/turning_fair_into_reality_1.pdf

[23] Yuri Demchenko, Lennart Stoy, Research Data Management and Data Stewardship Competences in University Curriculum, In Proc. Data Science Education (DSE), Special Session, EDUCON2021 – IEEE Global Engineering Education Conference, 21-23 April 2021, Vienna