The Importance of System Engineering Competences and Knowledge for Big Data Science and Research Infrastructure Projects

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

Abstract*—Big Data Science projects are an essential part of the modern data driven science and require extensive use of digital technologies and are typically built using large scale Research Infrastructures (RI) that combine computational, storage and data management facilities to support data collection, processing and scientific workflow management. The success of realising Big Data Science projects and effective use of Big Data Infrastructures (BDI) 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 BDIs and data driven scientific instruments require a wide spectrum of competences and knowledge related to modern technologies, system and software engineering, including also 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 summarises the Sustainable Architecture Design Principles (SADP) that can provide both guidance for the technical design of RI/BDI and a basis for targeted training for engineering and scientific personnel. The paper also refers to ongoing research and developments on ensuring the environmental sustainability of RIs where the proposed SADP plays an important role. The presented work is based on the author's experiences of developing and teaching courses on Big Data Infrastructure Technologies for Data Analytics (BDIT4DA) and DevOps for Software Engineering (DevOps4SE) 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 been adapted to the continuous technology development and specific needs of the developer teams.

Keywords*—*Data Science Projects, Research Infrastructure, Sustainable Architecture Design Principles, Environmental Sustainability, System and Software Engineering Competences and Skills, EDISON Data Science Framework (EDSF), Education and Training Methodology.

I. INTRODUCTION

Modern science is data driven, growingly digitalized and extensively uses digital and Big Data and Data Science Analytics technologies. It requires large scale Research Infrastructures (RI) to combine computational, storage and data management facilities to support data collection, processing and scientific workflow management. Europe has well established 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-Infrastructure that serves all European research community such as GEANT, EGI, 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 wide spectrum of competences and knowledge related to modern technologies, system and software engineering (SSE).

The success of realising Big Data Science projects and effective use of Big Data Infrastructures (BDI) will depend on adopting modern technologies and well-defined architectures that support the sustainability and long term evolution of technical solutions, contributing also to the environmental sustainability of BDI/RI through their lifecycle. The paper proposes sustainable architecture design principles that can provide guidance for RI technical design and can be used for targeted training for engineering and scientific personnel training. The suggested principles are based on important architectures and standards that define the modern Internet, information and communication technologies (ICT), and a wide range of system and software engineering practices.

Growth of the complexity of cloud based data centric applications requires new skills from research engineers that span beyond just research software engineering or scientific programming. 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. This is another area where the developer teams need to have the necessary expertise.

This paper refers to and effectively uses the EDISON Data Science Framework (EDSF) methodology, initially developed in the EDISON Project (2015-2017) and currently maintained by the EDISON community [3]. The EDSF provides a general framework for the Data Science related curriculum design and customized course development what has been discussed in the author's previous works [5, 6]. The Data Science Engineering Body of Knowledge (DSENG-BoK) and Model Curriculum (MC-DSENG), described in detail below, are defined as a part of EDSF.

The paper presents the author's experiences in developing and teaching courses on Big Data Infrastructure Technologies for Data Analytics (BDIT4DA) and DevOps for Software Engineering (DevOps4SE) 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 developer teams

The paper is organised as follows. Section II provides background information on the large European RI and e-Infrastructure projects. 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 introduces the Sustainable Architecture Design Principles (SADP) and their component groups that provide a basis for defining necessary competences for sustainable Big Data and RI design and operation. Section V introduces the EDISON Data Science Framework (EDSF) and provides brief information about the Data Science Engineering Body of Knowledge (DSENG-BoK) and the DSENG Model Curriculum. Section VI describes two courses Big Data Infrastructure Technologies for Data Analytics (BDIT4DA) and DevOps and Cloud based Software Engineering (DevOps4SE) taught by the author in different education environments and formats, which could provide necessary knowledge for sustainable research infrastructure and services development. Conclusion in Section VII explains the author's motivation in sharing experience and describes ongoing efforts to support professional approach in research infrastructure services and applications developments.

II. European Initiatives to Support Research

A. *European Open Science Cloud (EOSC)*

Europe traditionally supported the creation of Research Infrastructures (RI) and e-Infrastructures to support research in different thematic domains. EOSC is an initiative and program 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 [7]. 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 [8], which will technically require creating Federated Research Data Management Infrastructure.

Looking retrospectively at ERA and EOSC development since 2016, we can refer to the author's paper from 2020 [9] 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.

Recent developments focused on building sustainable EOSC infrastructure include the launch of the EOSC EU Node that will provide a backbone for future federated infrastructure of national and thematic EOSC Nodes [10]. This offers several key benefits for the research community, as it serves as a foundational component for bridging gaps between different research communities, enhancing resource sharing, and extending international cooperation. The EOSC nodes approach will help building effective researcher-centric infrastructure providing a similar level of usability as public cloud service providers. It needs to further incorporate the powerful user driven "crowd wisdom" demonstrated in the numerous EOSC related projects funded in EU Horizon 2020 and Horizon Europe programs. Adopting industry experience in sustainable infrastructure engineering and operation would further facilitate RI and EOSC development, however 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, 11] 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.

C. *GreenDIGIT Project for Greening Future RIs*

The GreenDIGIT project [12] aims to develop and validate a comprehensive framework including innovative technical solutions, models, and tools to enhance the environmental sustainability of digital RIs by reducing their environmental and climate impact throughout their entire lifecycle, ensuring that RIs operate sustainably while maintaining high standards of scientific excellence.

The GreenDIGIT recent developments include the Shared Responsibility Model for Sustainability (SRM4S) that defines multiple roles in ensuring the environmental sustainability of the whole ecosystem, including RI providers and operators, virtual research environment providers (typically operated by thematic RIs or scientific domain groups), research applications vendors, and researchers or research project. SRM4S implementation will be supported by sustainability-by-design approach and framework that will be based on the proposed in this paper Sustainable Architecture Design Principles.

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 the 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 simple github based Scrum management tool. All teams members were involved in all aspects of the project development, although at different levels and responsibility: 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, that 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 as 5G, IoT, high performance and optical networking, data management, with strong experience in building domain specific testbeds and applications.

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 using 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 processes 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. Also, in the conditions of fast technologies development, core architecture and design decisions must be revisited and verified with the new technologies development and evolution.

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 stand-alone 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*

Critical analysis of the above scenarios, we can identify the following key knowledge and competences that are required for successful architecture design and infrastructure projects 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 details 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 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, requirements to the core components and to other or external components that 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, 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 and possible 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

A. *Sustainable Architecture Design Principles Structure*

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

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, 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. At least two major cloud providers recently published their “Well Architected” cloud services design recommendations. The proposed recommendations are also supported by the research community’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.
- Other architecture styles may be used for specific tasks and applications: web-queue-worker, event-driven, Big Data, data-centric and data-driven as an alternative to compute-centric.
- 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 platform, distributed and highly scalable systems, in particular Hadoop ecosystem and NoSQL databases.
- Data management infrastructure and services that should cover two domains: services data (mostly related to management plane) and business or research data produced as a result of business operation 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 to 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 are 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 interested readers to explore):

- The 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, architecture style, 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 inside and between developer teams at the different stages of a project.
- 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.

It takes some time for all cooperating team members to come to a common understanding of terminology 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 retained and what not retained.

V. Data Science Engineering (DSENG) BoK as a basis for Addressing New Skills Demand

This section provides conceptual background for developing education and training courses on modern data driven infrastructure and applications development. This includes the authors experience in developing EDSION Data Science Framework and its application in teaching Data Science Engineering and DevOps courses.

A. *EDISON Data Science Framework (EDSF)*

The EDISON Data Science Framework (EDSF) [5] provides a basis for Data Science education and training, curriculum design and competences management that can be customised for specific organisational roles or individual needs. Defined in the EDISON project for the Data Science professions family, EDSF includes/reflects/addresses all necessary competence and knowledge areas for modern data driven science and can be extended to modern System and Software Engineering. EDSF is regularly updated (since its first publication in 2016) to reflect continuous technologies development. Recent EDSF Release 4 (EDSF2022) includes the following EDSF parts, which are published in separate documents [4]: Part 1 - CF-DS – Data Science Competence Framework; Part 2 - DS-BoK – Data Science Body of Knowledge; Part 3 - MC-DS – Data Science Model Curriculum; Part 4 - DSPP - Data Science Professional profiles and occupations taxonomy; Part 5 – EDSF-UCA – Use Cases and Applications.

The CF-DS provides the overall basis for the whole framework and includes the core competence groups required for the successful work of a Data Scientist in different work environments in industry and in research and through the whole career path (refer to the CF-DS specification for details):

- Data Science Analytics (including Statistical Analysis, Machine Learning, Data Mining, Business Analytics, others) (DSDA)
- Data Science Engineering (including Software and Applications Engineering, Data Warehousing, Big Data Infrastructure and Tools) (DSENG)
- Data Management and Governance (including data stewardship, curation, and preservation) (DSDM)
- Research Methods and Project Methods (DSRMP)
- Domain Knowledge and Expertise (Subject/Scientific domain related)

Data Science competences must be supported by knowledge that are defined primarily by education and training, and skills that are defined by work experience correspondingly.

The DS-BoK defines the Knowledge Areas (KA) for building Data Science curricula that are required to support identified Data Science competences. DS-BoK is organised by Knowledge Area Groups (KAG) that correspond to the CF-DS competence groups. The DS-BoK is based on ACM/IEEE Classification Computer Science (CCS2012) [15], incorporates best practices in defining domain specific BoK's and provides reference and mapping to existing classifications and BoKs: ACM Computer Science BoK (CS-BoK) selected KAs [16], Software Engineering BoK (SWEBOK) [17], and related scientific subjects from CCS2012: Computer systems organization, Information systems, Software and its engineering.

The MC-DS is built based on DS-BoK and linked to CF-DS where Learning Outcomes are defined based on CF-DS) competences and Learning Units (LU) are mapped to Knowledge Units in DS-BoK.

The formal DS-BoK and MC-DS definitions create a basis for Data Science educational and training programmes compatibility and consequently Data Science related competences and skills transferability.

B. *Data Science Engineering BoK and Model Curriculum*

Data Science Engineering KAG helps develop the ability to use engineering principles to research, design, develop and implement new instruments and applications for data collection, analysis and management. It includes Knowledge Areas that cover: software and infrastructure engineering, manipulating and analysing complex, high- volume, high-dimensionality data, structured and unstructured data, cloud based data storage and data management.

Data Science Engineering Model Curriculum includes LUs related to system and software engineering, infrastructure design and operations, and data driven applications design. The following are commonly defined Data Science Engineering Knowledge Areas (as part of KAG02-DSENG):

- KA02.01 (DSENG/BDI) Big Data infrastructure and technologies, including NOSQL databased, platforms for Big Data deployment and technologies for large-scale storage;
- KA02.02 (DSENG/DSIAPP) Infrastructure and platforms for Data Science applications, including typical frameworks such as Spark and Hadoop, data processing models and consideration of common data inputs at scale;
- KA02.03 (DSENG/CCT) Cloud Computing technologies for Big Data and Data Analytics;
- KA02.04 (DSENG/SEC) Data and Applications security, accountability, certification, and compliance;
- KA02.05 (DSENG/BDSE) Big Data systems organization and engineering, including approached to big data analysis and common MapReduce algorithms;
- KA02.06 (DSENG/DSAPPD) Data Science (Big Data) application design, including languages for big data (Python, R), tools and models for data presentation and visualization;
- KA02.07 (DSENG/IS) Information Systems, to support data-driven decision making, with a focus on data warehouse and data centers.

C. *DSENG/BDIT - Big Data Infrastructure Technologies course content*

Big Data infrastructures and technologies shape many Data Science applications. Systems and platforms behind Big Data differ significantly from traditional ones due to specific challenges of volume, velocity, and variety of data that need to be supported by data storage and transformation. Data Lakes and SQL/NoSQL databases must be included in the DSENG curriculum.

Deployment of Data Science applications is usually tied to one of the most common platforms, such as Hadoop or Spark, hosted either on private or public clouds. The application workflow must be linked to a whole data processing pipeline, including ingestion and storage for a variety of data types and sources. Data Science Engineers and Research Software Engineers should have a general understanding of data and application security aspects in order to properly plan and execute data-driven processing in the organization.

D. *Importance of Data Management Competences*

Data Management and Governance (DMG) [18, 19] must be included in the DSENG courses and Big Data Engineering course to ensure a better link between infrastructure technologies and required DMG functionality and practices by the research community. This, in particular, is implemented in the practical BDIT curriculum. DMG topics and LUs should include the FAIR data principles (data must be Findable, Accessible, Interoperable, Reusable) [20, 21], that are growingly adopted by the research community and necessary infrastructure technologies to support them. FAIR compliant Data Management Infrastructure is the core component of the experimental and general research reproducibility.

VI. Experience of designing and teaching BDIT4DA and DevOps4SE Courses

This section describes the organisation and content of two courses that can provide the foundation for the System and

Software Engineering competences required for the design, deployment and operation of modern research infrastructure and services: Big Data Infrastructure Technologies for Data Analytics and DevOps and cloud based Software Engineering. Both courses use similar structure and organization but are targeted for different academic programs and primary groups of practitioners. They 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.

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 platform and development environment. The BDIT4DA course actually implements recommendations of the Data Science Engineering (DSENG) Body of Knowledge and Model Curriculum which are part of the EDSF [4]. 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 to Big Data technologies development and availability of the cloud based education platforms.

Currently, the course includes 10 basic modules that include both lectures and labs or practice. Since 2021 term, the course has included a new Module 8 on Data Science Projects Management and DataOps/MLOps [22]. 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 discussed in this paper Sustainable Architecture Design principles and 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 SQL and NoSQL Databases. CAP Theorem. Modern large scale databases AWS Aurora, Azure CosmosDB, Google Spanner.
Module 5 Data Streams and Streaming Analytics. Kafka, Flume. Spark architecture and popular Spark platforms, DataBricks.
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 split 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 workshops such as summer schools or conference tutorials.

B. *DevOps and Cloud based Software Development (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 and development are used as examples.

The technology related modules follow the main topics (Knowledge Units) of the proposed DevOpsSE Body of Knowledge [22, 23] which includes the following topics/modules:

1. DevOps fundamentals, Continuous Integration (CI), Continuous Delivery (CD), Continuous Testing. Relation to other agile development technologies Lean, CAMS, and ITSM. Organisational impact of DevOps, key performance indicators.
2. Agile software development: Scrum, Kanban, Kaizen; Agile Scrum process and team management; Role of multi-disciplinary feature teams. Agile Manifesto.
3. DevOps Tools and Processes: CI/CD pipeline; DevOps Toolchain; Coding, versioning, collaboration and team based development; Git, Automated testing. Software packaging, Container technologies, Kubernetes.
4. Site Reliability Engineering (SRE) and user or business centric large scale services operation and monitoring.
5. Cloud Computing Architectures, service and deployment models, Cloud IaaS and Infrastructure as Code, Cloud economics
6. Cloud monitoring and tools, AWS CloudWatch
7. Cloud powered software development: AWS and Azure example and tools; Cloud based Architecture design patterns
8. Cloud Automation processes and tools to support CI/CD: Cloud based tools (e.g. AWS CloudFormation, Azure ARM); multicloud tools Ansible, Terraform, others.
9. Quality Assurance (QA) in cloud based software development systems. in clouds.

10. DevSecOps; Secure Software Development Lifecycle Management (SDLM); cloud based tools for secure software development and testing.
11. Cloud Security Architecture and Models, Cloud compliance, CSA Consensus Assessment Initiative Questionnaire (CAIQ) and Certification.

Following positive experience from the BDIT4DA course and responding to the students' interest, two new topics/lectures will be added in the new term 2023/2024:

12. Data Science project management and DataOps/MLOps processes and platforms;
13. Sustainable Architecture Design principles and cloud based design patterns which will incorporate discussed in this paper competences.

C. *Practice/lab and project development environment*

Both courses from the beginning of their establishing in 2027-2018, used real cloud platforms, primarily AWS and/or Azure, benefiting from the educational credits provided by both providers. Since August 2021, educators can benefit from the AWS Academy resources and courses [24] that include a foundational course on Cloud Computing Foundation and Machine Learning Foundation. These courses are provided for all educators without passing certification training. More advanced courses such as for example Cloud Architecting, Cloud Developing, Cloud Operations, Data Analytics, require the teachers to pass the corresponding course themselves. AWS Academy uses Canvas for course delivery and management.

Although AWS Academy courses are professionally developed, they constitute only part of both courses, also the resources provided may not be sufficient for performing the educational course projects. The students are advised to either create their own starter AWS accounts or simply use the Free Tier [25] resources that are quite extensive and sufficient for typical projects in both courses DevOps4SE and BDIT4DA.

D. *Lessons Learned (in education and training)*

The highly demanded BDIT4DA course has been taught by the author in different programs and different installations: campus face-to-face teaching, part-time evening lecturing and practice, and online lecturing. Experience confirms that the lectures provide essential knowledge for the students and professionals to understand the modern Big Data and data driven technologies, lecture material can be slightly adapted to different audiences, given that there is no single textbook for the course. Recently established AWS Academy provides free of charge entrance level courses on the Cloud Computing Foundation and Machine Learning Foundation allow both courses BDIT4DA and DevOps4SE to use them as a single platform for learning basics and for educational projects. Using public cloud platforms for educational projects allows the students to become familiar with the public cloud platform as well as the cloud based development environment that can be reused in their future projects and professional activity.

It is also important to admit that besides the widely recognized importance of consistent data management in research and industry, this topic is rarely addressed in Computer Science curricula. The lectures and practice on Enterprise Data Management and Research Data Management are included in the BDIT4DA course (Module 6) and continuously updated with the ongoing development by Research Data Alliance (RDA) [26] and EOSC community, in particular data and metadata management tools [27].

An important aspect of the BDIT4DA course and any other course related to modern digital and data technologies is to facilitate the development by the students a kind of data-centric approach and thinking (refer to the author's recent paper [28]).

VII. CONCLUSION AND FUTURE DEVELOPMENTS

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 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. The author's personal experience in mastering new technologies, provided a solid foundation for university teaching and professional training on the technologies mentioned above.

The presented work is also based on the author's long experience in teaching cloud computing technologies and cloud based Software Engineering courses [22, 23, 29]. It is important to admit that providing students with knowledge of the architectural design of the modern cloud and Big Data platforms is essential for working with modern and future technologies. Clouds and Big Data are driving modern infrastructure development and provide examples of well-designed systems that could be re-used by the developers.

The presented work intends to provide a basis for wider discussion by the research and academic community on developing a System and Software Engineering competence framework, including a common body of knowledge, targeted for project technical coordinators and architects. Special training for project teams can be provided at the beginning of the project. This is especially important if the project or research team is composed of researchers or developers with different education and technical backgrounds.

The academic education or professional training must provide a strong basis for graduates and trainees to continue their further self-study and professional development in conditions of the fast developing technologies and agile business environment adopted by the majority of modern companies. To achieve this, the Data Science and Systems and Software Engineering curricula needs to be supported by professional skills development courses such as workplace skills that are also referred to as the 21st Century skills [29], which should develop an important attitude to continuous professional (self-)learning and understanding of wider technologies and competences scope in any specific professional activity.

ACKNOWLEDGMENT

The research leading to these results has received funding from the Horizon2020 projects EDISON (grant n. 675419), and current Horizon Europe projects SLICES-DS (951850), SLICES-SC (101008468), SLICES-PP (951850), GreenDIGIT (101131207).

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] EDISON Community wiki. [online] https://github.com/ EDISONcommunity/EDSF/wiki/EDSFhome

[4] 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

[5] Yuri Demchenko, Mathijs Maijer, Luca Comminiello, Data Scientist Professional Revisited: Competences Definition and Assessment, Professional Development and Education Path Design, International Conference on Big Daya and Education (ICBDE2021), February 3-5, 2021, London, United Kigndom

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

[8] EOSC Strategic and Research Innovation Agenda (SRIA) [online] https://eosc.eu/wp-content/uploads/2023/08/SRIA-1.1-final.pdf

[9] European Open Science Cloud – EU Node [online] https://open-science-cloud.ec.europa.eu/

[10] 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.

[11] 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.

[12] GreenDIGIT Project [online] https://greendigit-project.eu/

[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] The 2012 ACM Computing Classification System [online] http://www.acm.org/about/class/class/2012

[16] ACM and IEEE Computer Science Curricula 2013 (CS2013) [online] http://dx.doi.org/10.1145/2534860

[17] Software Engineering Body of Knowledge (SWEBOK) [online] https://www.computer.org/web/swebok/v3

[18] 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

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

[20] 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.p df

[21] 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, Austria

[22] 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.

[23] Yuri Demchenko, Zhiming Zhao, Spiros Koulouzis, Jayachander Surbiryala, Zeshun Shi, Xiaofeng Liao and Jelena 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), part of the eScience 2019 Conference, September 24 – 27, 2019, San Diego, California, USA

[24] AWS Academy [online] https://www.awsacademy.com/AcademyHome

[25] AWS Free Tier [online] https://aws.amazon.com/free/

[26] Research Data Alliance (RDA) [online] https://www.rd-alliance.org/

[27] EOSC Core Components, FAIRCORE4EOSC Project [online] https://faircore4eosc.eu/eosc-core-components

[28] Yuri Demchenko, Viktoriya Degeler, Ana Opresu, Steve Brewer, Professional and 21st Century Skills for Data Driven Digital Economy, In Proc. Data Science Education (DSE), Special Session, EDUCON2023 – IEEE Global Engineering Education Conference, 1-4 May 2023, Kuwait

[29] Demchenko, Yuri, David Bernstein, Adam Belloum, Ana Oprescu, Tomasz W. Wlodarczyk, Cees de Laat, New Instructional Models for Building Effective Curricula on Cloud Computing Technologies and Engineering. Proc. The 5th IEEE International Conference and Workshops on Cloud Computing Technology and Science (CloudCom2013), 2-5 December 2013, Bristol, UK.