Importance of System Engineering Competences and Knowledge in Large Scale Digital Research Infrastructure Projects

Yuri Demchenko (University of Amsterdam)

Architecture design thinking in system and software engineering

Employing System and Design thinking for the structured design and development of complex computer/IT services and applications

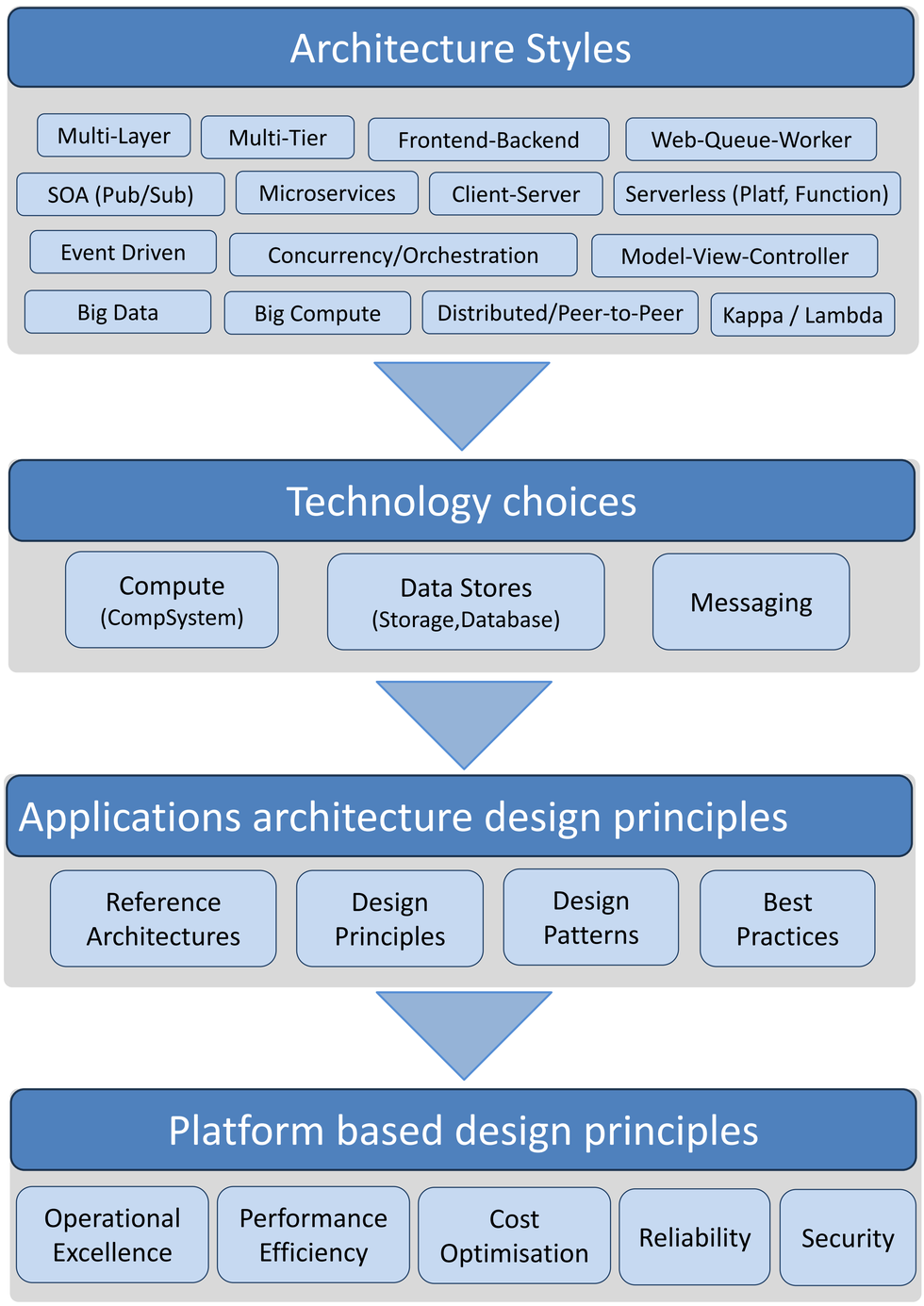
1. **Problem-Solving Orientation**: Deep understanding of the problem to be solved, considering both functional requirements (what the system is supposed to do) and non-functional requirements (qualities such as performance, scalability, reliability, and usability).
2. **Holistic View**: Holistic view of the system, considering all components and their interactions.
3. **Iterative and Incremental Development**: Iterative and incremental development processes.,
4. **Stakeholder Engagement**: Engaging with all stakeholders, including end-users, developers, business owners, and others.
5. **Design Principles and Patterns**: Leverage established design principles and patterns to guide decision-making.
6. **Trade-off Analysis**: Analise trade-offs, such as performance versus cost or functionality versus security.
7. **Sustainability and Evolution**: Consider the long-term sustainability and evolution of the system.

Software Architecture styles

- Multi-layer architecture style
- Multi/N tiers models
- Web-Queue-Worker
- Services Oriented Architecture (SOA)
- Microservices Architecture (MSA)
- Client-Server Architecture
- Serverless (Platform, Function)
- Frontend – Backend and Model-View-Controller (MVC)
- Event Driven Architecture (EDA)
- Concurrency/Orchestration/ Pipeline
- Realtime Kappa & Batch/Speed Lambda
- Big Data (large scale distributed/ stream): Hadoop, NoSQL, Spark, Cloud
- Big Compute (HPC, Batch, Scalable, Parallel, Cloud)

Importance of Layered Architecture

- Separation of Concerns (SoC)
- Scalability (Horizontal and vertical)
- Maintainability and Manageability
- Interchangeability and Flexibility, Multi-provider/multi-vendor
- Reusability
- Improved Security (multilayer measures)
- Simplified Testing
- Easier Deployment and Integration
- Development Parallelism
- Standardization



Sustainable (Long-term) Architecture Design Principles

General architecture design principles

- **Layered architecture design** for services and mechanisms, including inter-layer interfaces, including cross-layer services and mechanisms that are typically defined as service planes.
- **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).
- **Service lifecycle management** that includes services to support all lifecycle stages, in particular, services modelling, composition, deployment and orchestration for operation.
- **Operations and management support**: SLA, monitoring, audit, certification compliance

Data management infrastructure and related services

- Data management services: service data (management & operation) and research, industrial or business data produced in operations; **FAIR data principles for data/metadata**.
- Big Data computation models and supporting platform, distributed and highly scalable systems, in particular Hadoop and NoSQL databases.
- Services and data management in the **continuum** of IoT/sensor networks, edge, cloud, data-driven applications that also includes 5G/6G RAN (Radio Access Network), edge and cloud convergence.

Security and compliance design principles

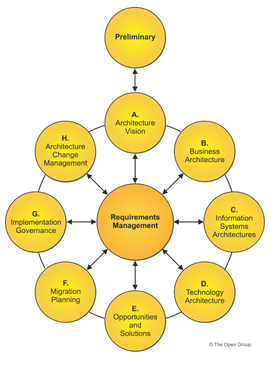
- **Security architecture and security services lifecycle management** including trust bootstrapping and secrets management; integration with the main infrastructure services via security API and policy definition.
- **Compliance frameworks** that define requirements and recommendations for services and infrastructure security design and operation. Refer to CSA and Compliance Assessment Initiative Questionnaire (CAIQ).
- **Industry best practices**: Federated AuthN, AuthZ, Identity Management

Project Management and DevOps based Agile principles

- **DevOps and SRE** (Site Reliability Engineering) practices applied to system and services engineering and operation. Includes continuous monitoring and optimisation on multiple user - centric and business-centric SLI/KPI (Service Level/Key Performance Indicators).
- **DevSecOps** that addresses security design aspects during the whole system/services lifecycle, intending to address “Security by Design“ concept
- General compliance with the **project management principles, models and procedures** applied to infrastructure, services, and data handling and analytics (ITIL, PMI, Scrum. etc).

Architecture design principles and best practices by Microsoft Azure and AWS

Ten design principles for Azure applications
https://learn.microsoft.com/en-us/azure/architecture/guide/design-principles/
Microsoft Azure Well-Architected Framework – Pillars: **Reliability – Security – Cost optimization – Operational Excellence – Performance Efficiency**
https://learn.microsoft.com/en-us/azure/well-architected/
AWS Architecture Center - https://aws.amazon.com/architecture/
AWS Well-Architected Framework – Pillars: **Operational Excellence – Security – Reliability – Performance Efficiency – Cost Optimisations - Sustainability**
https://docs.aws.amazon.com/wellarchitected/latest/framework/the-pillars-of-the-framework.html



TOGAF Architecture development steps

References

Yuri Demchenko, The Importance of System Engineering Competences and Knowledge in Large Scale Digital Research Infrastructure Projects, Proceedings EDUCON 2024 Conference, 8-11 May 2024, Kos, Greece
Yuri Demchenko, Sustainable Architecture Design Principles for Large Scale Research Infrastructure Projects, Proc. the 25th Int. Conf. on High Performance Computing and Communications (HPCC2023), 13-15 December 2023, Melbourne, Australia
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
EDISON Data Science Framework, EDSF Community Wiki, https://github.com/EDISONcommunity/EDSF/wiki/EDSFhome

Contact: Yuri Demchenko <y.demchenko@uva.nl>

EDISON building the data science profession
UvA Universiteit van Amsterdam
slices RI Grant ID: 951850
GreenDIGIT Grant ID: 101131207