POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

LOG8430E: Software Architecture and Advanced Design

Fall 2021

Instructor

- Marios Fokaefs – Assistant Professor
- Research
  - DevOps
  - Software Evolution and Maintenance
  - Cloud computing, IoT, SOA
  - Software Engineering Economics
- https://www.polymtl.ca/expertises/en/fokaefs-marios-eleftherios
- marios.fokaefs@polymtl.ca
- Office M-4119 (Tuesdays, Wednesdays, Thursdays)
  - Virtual Webex: https://polymtl.webex.com/meet/marios.fokaefs

Instructor – Marios Fokaefs

- BSc in Computer Science – Greece
- MSc in Software Engineering – Alberta
  - Identification of Extract Class refactoring opportunities in object-oriented systems.
- PhD in Software Engineering – Alberta
  - Web service evolution
  - Economic management of web services
- Research Assistant (postdoc) – York
  - Self-adaptive cloud systems
  - Economic management of cloud and distributed systems.

Mohammadreza Rasolroveicy - TA for Software Architecture

- PhD Student at Polytechnique
- Research interests: DevOps - AutoScaling - Blockchain - IoT Security - Cloud Computing
- Email : mohammadreza.rasolroveicy@polymtl.ca
- Office : 4218 - Department of Computer & Software Engineering , Polytechnique Montréal

Course Objectives

- Present the styles and patterns of software architecture and design putting an emphasis on distributed systems.
- Measure and maintain the quality of design and architecture.
- Explore and master the design of distributed systems in real case studies: frameworks, cloud and service-oriented systems, and systems for big data analytics.

At the end of the course, the student will be able to:

- Design the architecture of software by choosing the styles of architecture and design patterns according to the requirements and justifying these decisions.
- Design the components of the architecture by using special models and novel technologies (event-driven architectures, service-oriented and object-oriented architectures, cloud)
- Evaluate the quality of an architecture or a software design and maintain it by applying changes.

Course Philosophy

- This is a fourth year course.
  - You will not be asked to design or develop code.
  - It is assumed that you already know the principles of software design (which we will remember here).
- Can you recognize, recover, evaluate a software design or architecture that you were not involved in?
- Scenario: You have just been hired by a company, you inherit an existing system and you are integrated in an existing team.
- You are asked to help others understand a system that you did not design.
- Learn to communicate and present your findings to technical and non-technical personnel.

Situation of the course

- LOG8430 ends a chain of courses:
  - INF1005C – procedural programming
  - INF1010 – object-oriented programming
  - LOG2410 – software design
  - LOG3210 – Elements of languages and compilers
  - LOG3430 – Methods of software testing and validation

Situation of the course

- LOG8430 is also a graduate course
- Be at ease to read and write in English.
- Conduct research
  - There is not always one single correct answer.
  - Justify and verify your responses.
  - Critical thinking
  - Collaborate!
- Work well in a team.
- Advanced concepts on:
  - Object-oriented design
  - SOA
  - Cloud
  - Distributed Systems

In-class format of the course

- 2 hours of lecture – 1 hour of in-class practice (M-2101)
  - Exercises on the course of the day.
- Every class will start at 9am sharp with two short breaks.
- 3 hours of lab every second Tuesday (L-3818)
  - Work on the assignments.
  - We follow the calendar for even lab weeks (B2).
  - First lab on September 7th
- Safety measures
  - Keep physical distance (seat with two seats between you and you can keep your masks off, or with one seat and try to keep your masks on)
  - Do not approach the podium! As I will not approach the audience. If you have a question you can ask from your seat, even after the end of the class.
  - A slack channel will be maintained, to further facilitate collaboration. Bring your laptops or portable devices.

Virtual format of the course

- Pre-recorded lectures on YouTube.
  - You are invited to watch the videos before the actual day of the course and prepare your questions for discussion.
  - The videos will not be kept in-sync with the actual lectures, but they will still provide a good summary/approximation of the course material.
- A Slack space will be maintained for direct interaction.
  - There will be channels for lectures, labs, assignments and exams.
  - You can post your questions any time, but the space will be closely monitored during course hours.
- In-class sessions *may* be available over WebEx for those that cannot attend in person.
  - If there are a lot of technical difficulties, we may have to adjust these sessions.
  - Recordings *may* be available after the sessions.

Course schedule

Course	Content	Date	Deadline /Quiz
01	Introduction Basic concepts SOLID principles Design patterns	2 September	
02	Distributed architectures Architectural styles and patterns Frameworks Ecosystems	9 September	
03	Design quality Quality attributes Quality metrics	16 September	
04	Bad design Antipatterns / design smells Refactoring, refactoring to patterns	23 September	

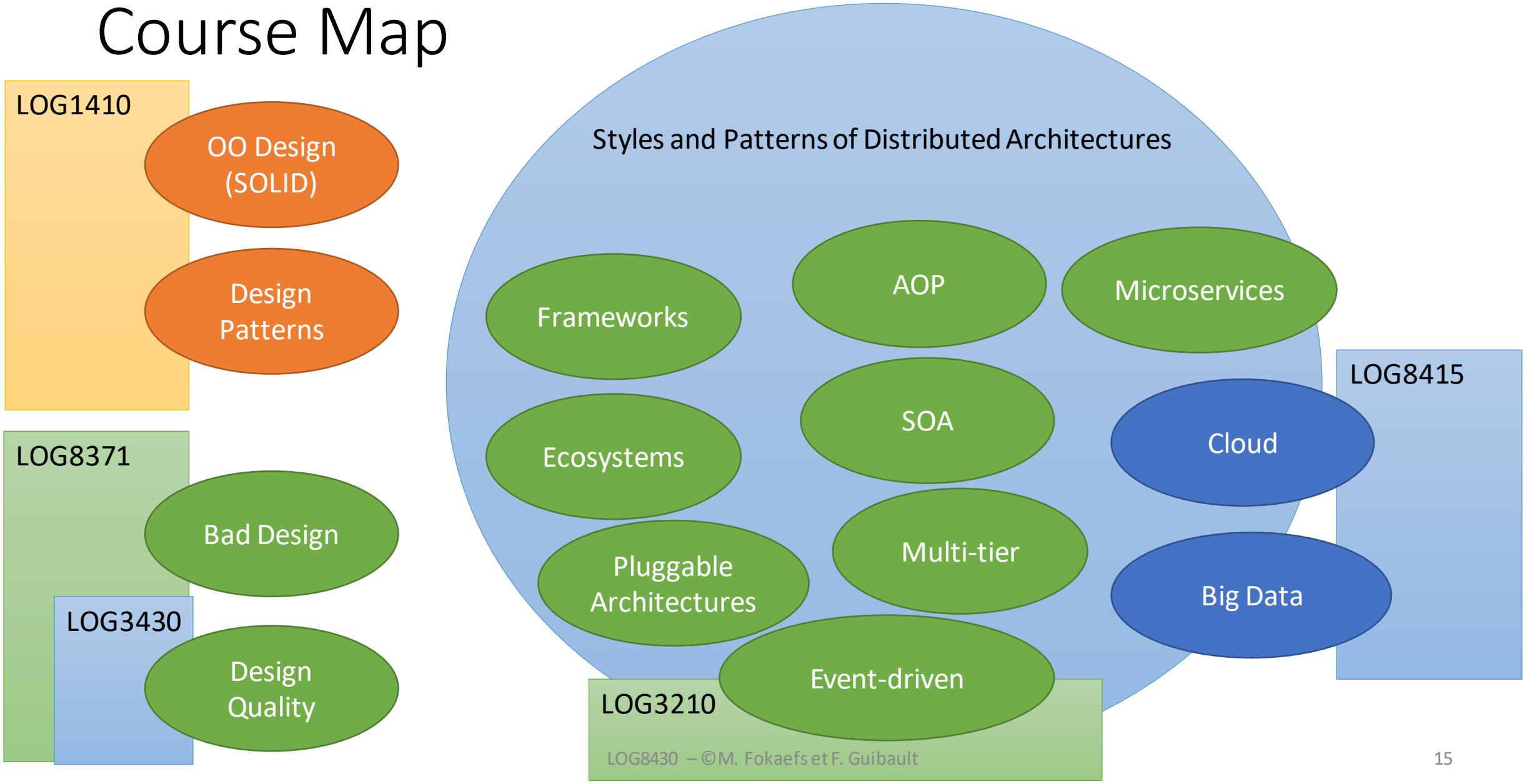
Course schedule

Course	Content	Date	Deadline/Quiz
05	Software Architecture for big data: Insertion and analytics	30 September	Quiz 1
06	Software Architecture for big data: Databases	7 October	TP1 (4 October)
--	*Reading Week*	14 October	
07	Blockchain	21 October	
08	Event-driven architectures Concurrency patterns	28 October	Quiz 2

Course schedule

Course	Content	Date	Deadline/Quiz
09	SOA	4 November	
10	Microservices	11 November	
11	AOP	18 November	
12	Revision	25 November	Quiz 3
13	Assignment Presentations	2 December	TP3 (presentations: 1 December papers: 9 December)

Course Map


LOG1410
OO Design (SOLID)
Design Patterns
LOG8371
Bad Design
LOG3430
Design Quality
Styles and Patterns of Distributed Architectures
Frameworks
AOP
Microservices
Ecosystems
SOA
Cloud
Pluggable Architectures
Multi-tier
Big Data
Event-driven
LOG8415
LOG3210

Resources

- There is no principal textbook, but many resources to study and understand the material:
  - Links
  - Tutorials
  - Textbooks
  - Articles
  - Software
- Available on Moodle for every class.

Tools

- IDE
  - Android Studio - https://developer.android.com/studio
  - Eclipse - https://www.eclipse.org/
  - IntelliJ IDEA - https://www.jetbrains.com/idea/
- Git
  - GitHub - https://github.com/
  - Bitbucket - https://bitbucket.org/
- Design and modeling
  - Papyrus - https://www.eclipse.org/papyrus/
  - ArgoUML - http://argouml.tigris.org/
  - Enterprise Architect - https://sparxsystems.com/products/ea/
  - Understand - https://scitools.com/features/
  - PlantUML Integration - https://plugins.jetbrains.com/plugin/7017-plantuml-integration
  - PlantUML Diagram Generator - https://plugins.jetbrains.com/plugin/15991-plantuml-diagram-generator
  - Sketch It! - https://plugins.jetbrains.com/plugin/10387-sketch-it-
- Source code analysis
  - JDeodorant - https://marketplace.eclipse.org/content/jdeodorant
  - Ptidej - http://www.ptidej.net/
  - SonarCloud - https://sonarcloud.io/
  - Android Studio
- Metrics
  - Understand
  - Eclipse Metrics 3 - https://github.com/qxo/eclipse-metrics-plugin
  - MetricsReloaded - https://plugins.jetbrains.com/plugin/93-metricsreloaded
  - MetricsTree - https://plugins.jetbrains.com/plugin/13959-metricstree

Important

- All the material presented during the lectures and the labs can be the object of an evaluation.
- The content of the course is dynamic!
- The instruction language of the course for the Fall term is English.
  - Slides and videos for the lectures are available both in English and in French (older versions).
  - All supplementary resources are mainly in English.
  - You may ask for special dispensation to submit an assignment or take a quiz/exam in French at least one week before the respective deadline.
- All slides are currently available in Moodle.
  - Changes may occur in the slides.
- If there are updates on the material, there will be announcements on the forum.
- For as long as the course takes place in-person the pre-recorded videos will not be updated.
- Late submission of assignments will receive 10% penalty per day of delay.
- Any request to differ an exam needs to pass through the office of academic affairs.
- Collaboration is permitted for the assignments, but all rules and regulations relevant to plagiarism apply at all time.

Evaluation of the course

- Assignments 60%
  - TP1 15%
  - TP2 15%
  - Discussion assignment 30%
- Quizzes 3 18% (6% each)
- Final exams 22%
- At least 40% in the final exams to pass the course.

Assignments

- In teams of 3 people
  - Use the link for “Group Formation for Assignments” in Moodle by September 10.
- Objectives
  - Master the theory
  - Understand and analyse
  - Design and decisions
- 2 practical assignments + 1 discussion assignment (paper)
- 2 lab sessions per assignment: 1st will show tutorials, 2nd will answer questions and solve problems.
- Practical assignment: Analysis of a system (architecture recovery, problems in design and so on).
  - Prepare a report of the analysis.
- Discussion assignment: Compare some solution alternatives, exercise some systems to evaluate them.
  - Prepare a paper describing a core concept, the comparison of alternatives and the experimental evaluation.

Final exams and Quizzes

- Individual
- 3 Quizzes throughout the term
  - Possible dates: Sept 30, Oct 28, Nov 25
  - Each 6% of the total course grade
- Final exams
  - 22% of the total course grade
  - You need at least 40 out of 100 in the final exams to pass the course.
- Theory and practice
  - Like the exercises in class.
- Examples of systems discussed in class, in lab or in the assignments.
- Depending on the situation of the course at the date of each exam:
  - In-person: Written exams in classroom (or where Polytechnique specifies)
  - Virtual: Quizzes and exams will be performed as Moodle Tests online.

Interactive sessions

- Exercises and quizzes on the subject of the week.
- Learning by practice.
- Improve and evaluate your ability to make, explain, and justify decisions on problems around software design and architecture.
- There will be no evaluation for these exercises.
- BUT! Examples of such exercises may be used in the final exam.

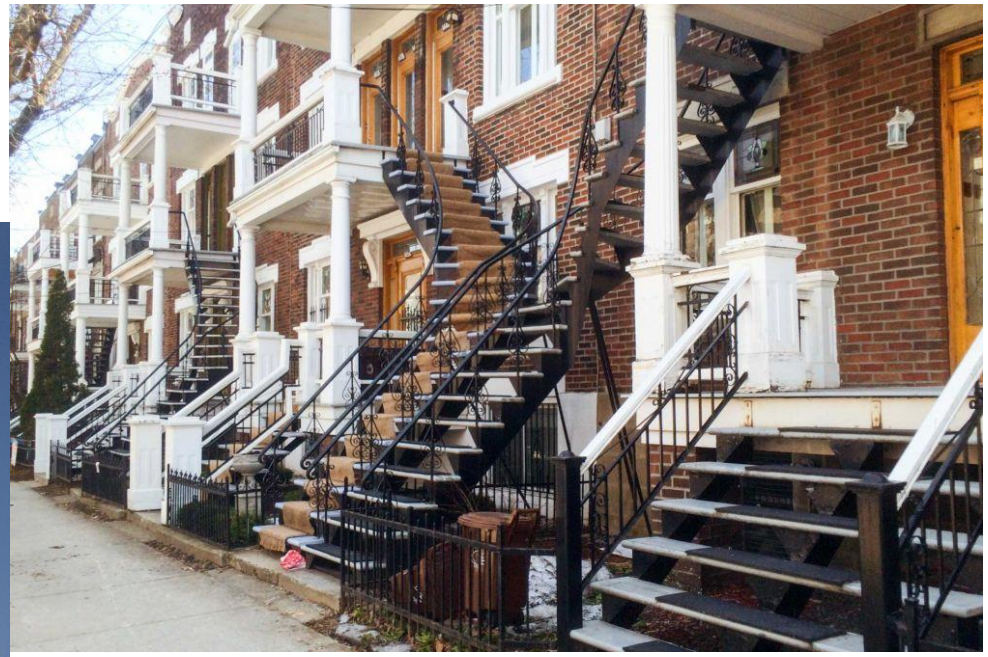
3 Questions on software architecture

1. What is a software architecture?
2. What are the criteria that guide a software architect to choose a specific architectural style?
3. What impact does the choice of architecture have on the development of a software system?

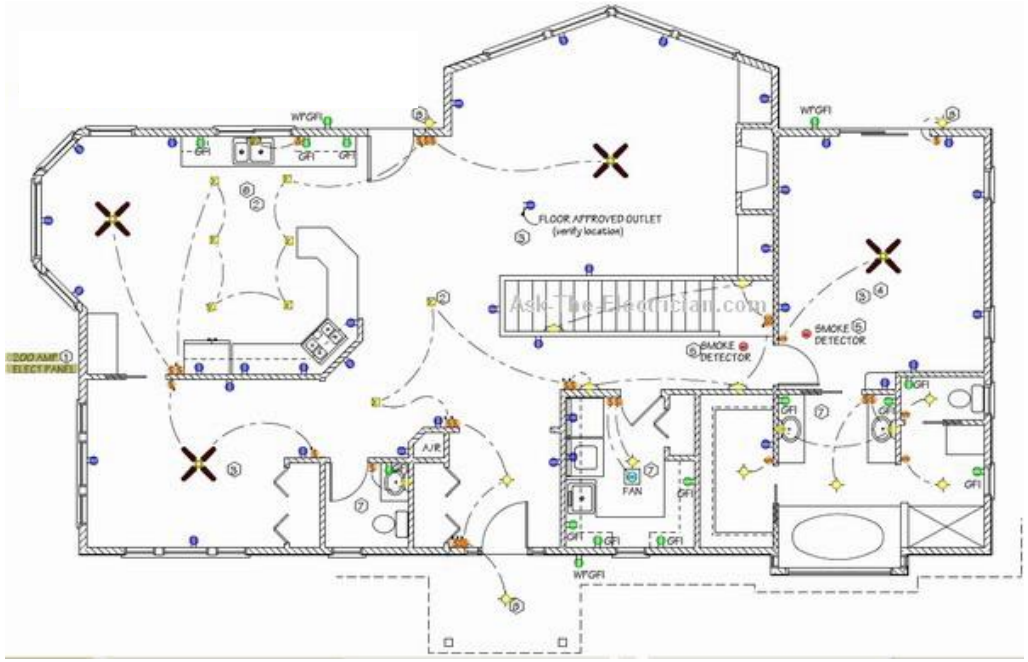
3 Questions on software design

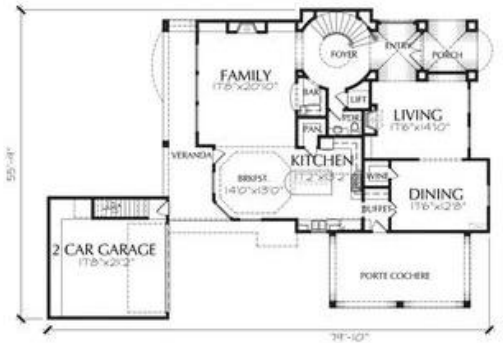
1. What is the difference between software architecture and software design?
2. Are the decisions on architecture and design independent?
3. How can we evaluate the quality of an architecture and the quality of a design?

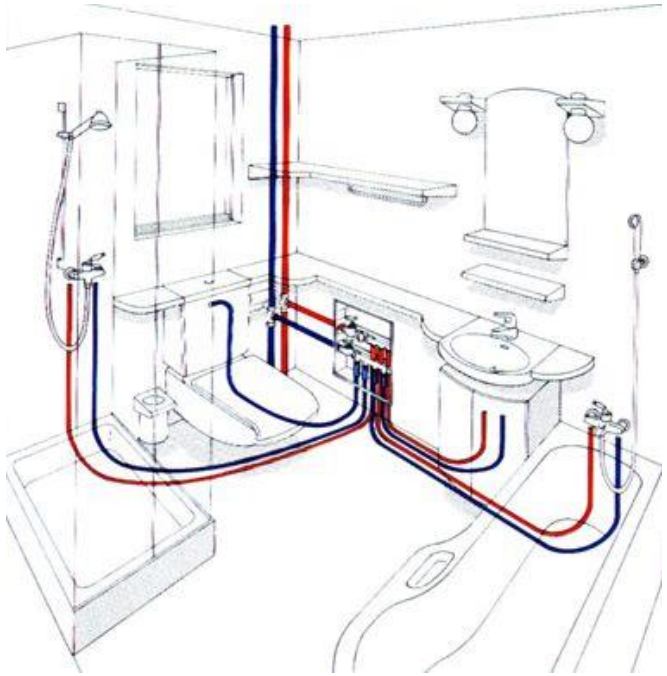
Architecture and design in the building domain



Architecture and design in the building domain







3 Questions on software architecture

1. What is a software architecture?
2. What are the criteria that guide a software architect to choose a specific architectural style?
3. What impact does the choice of architecture have on the development of a software system?

3 Questions on software design

1. What is the difference between software architecture and software design?
2. Are the decisions on architecture and design independent?
3. How can we evaluate the quality of an architecture and the quality of a design?

What is architecture?
What is design?

Software Architecture (ANSI/IEEE Std 1471-2000)

The architecture of a software system is defined as the fundamental organization of the system, embodied in its components, their relationships with each other and their environment, and the principles that guide its design and evolution.

Software Architecture (SEI definition)

The architecture of a software system is the set of structures necessary to reason about the system, which include the software elements, the relations between them and the properties of each one of them.

Software Architecture (Sangwan, 2015)

The architecture of a software system is fundamentally concerned with the organization of a system into its constituents and their interrelations in order to achieve a given objective.

Architecture

- It answers to “what?”
- On a model level
  - Packages, components and relationships
- A more abstract level of decisions
  - The decisions that need to be made in advance.
  - The decisions which are very expensive to change during development.
- Relevant to the operational objectives and the non-functional requirements.
- It includes:
  - Modules
  - Databases and data technologies
  - Execution and deployment environments

Design

- It answers to “how?”
  - How are we going to implement the architecture?
- More concrete elements:
  - Classes
  - Methods/Functions
  - Data tables
- Relevant to the functional requirements.
- Elements that can change during development (maintenance phase)
  - Reengineering (refactoring)
  - Evolution
  - Adaptation