M.EIC, 2024-25

Large Scale Software Development

Ademar Aguiar

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

DEI
DEPARTAMENTO DE
ENGENHARIA INFORMÁTICA

What is Software Architecture?

- Well, we know what Building Architecture is:
  - The Structure of the Building
  - The Overall Partitioning – Rooms and Such
  - Guidelines for Construction
  - Some notion of the use of the building

- It is similar in Software Architecture
  - The gross structure of the software
  - The overall partitioning – Subsystems and Such
  - Guidelines for construction, e.g., technologies to be used
  - The highest-level mapping of the problem space to the solution space
  - Some notion of the use of the software

… more

- A model of the finished system
  - Related to mapping of the problem space to solution space
- A vision
  - Of how the system should work
  - Of how it should be implemented
- A guide for implementation
  - (see above)

Why Software Architecture?

- The bits don’t care!
- Why is software architecture important?

Why Architecture?

- Every system has an architecture
  - But some are better than others
  - A systematic architecture makes life much easier for future as well as present developers

Architecture helps:

- People understand the system
  - Programmers, managers, **and users**
- Divide up the work
- Accomplish quality requirements
  - Important: more on this later
- Maintenance and enhancements
- Guide product families

Architectural Knowledge

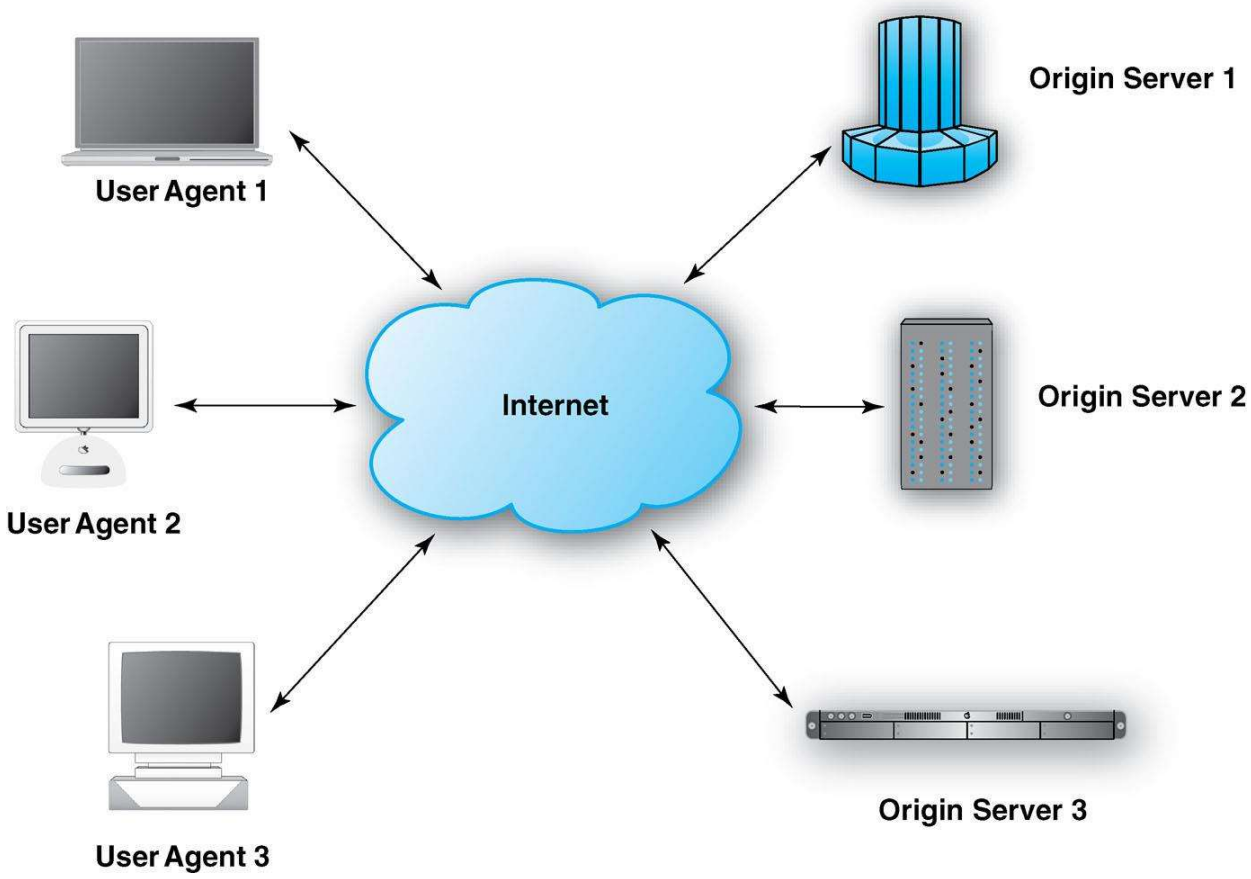
- In system maintenance & enhancement, how much effort is devoted to discovery?
  - HALF
  - A good architecture can help a lot
    - Structure
    - Implementation conventions

The Power of Big Ideas:
The Architecture of the Web

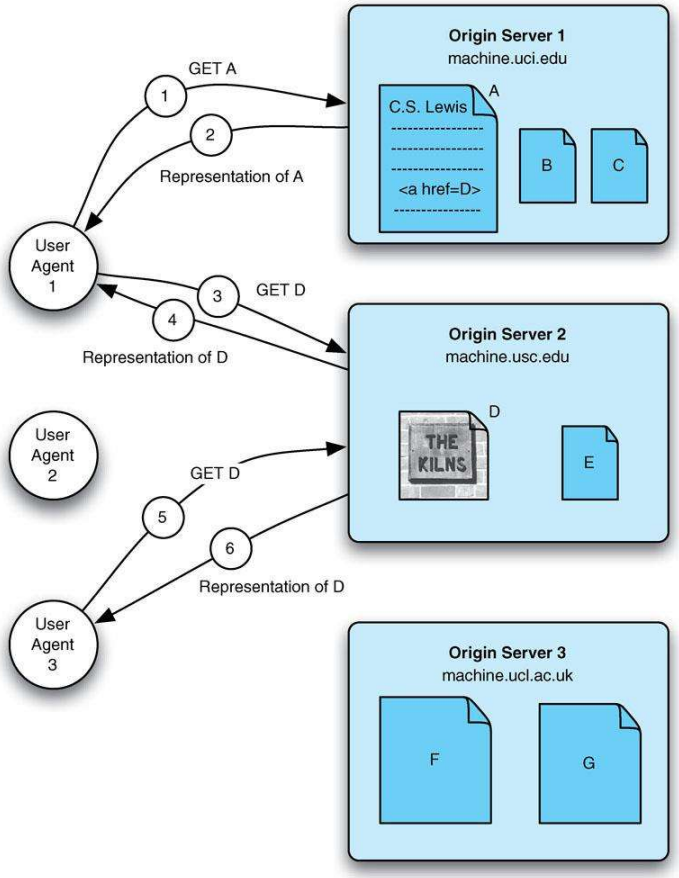
- Suppose you want to build the World Wide Web …
- Where do you start?
- For that matter, what *IS* the Web?

DEI DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

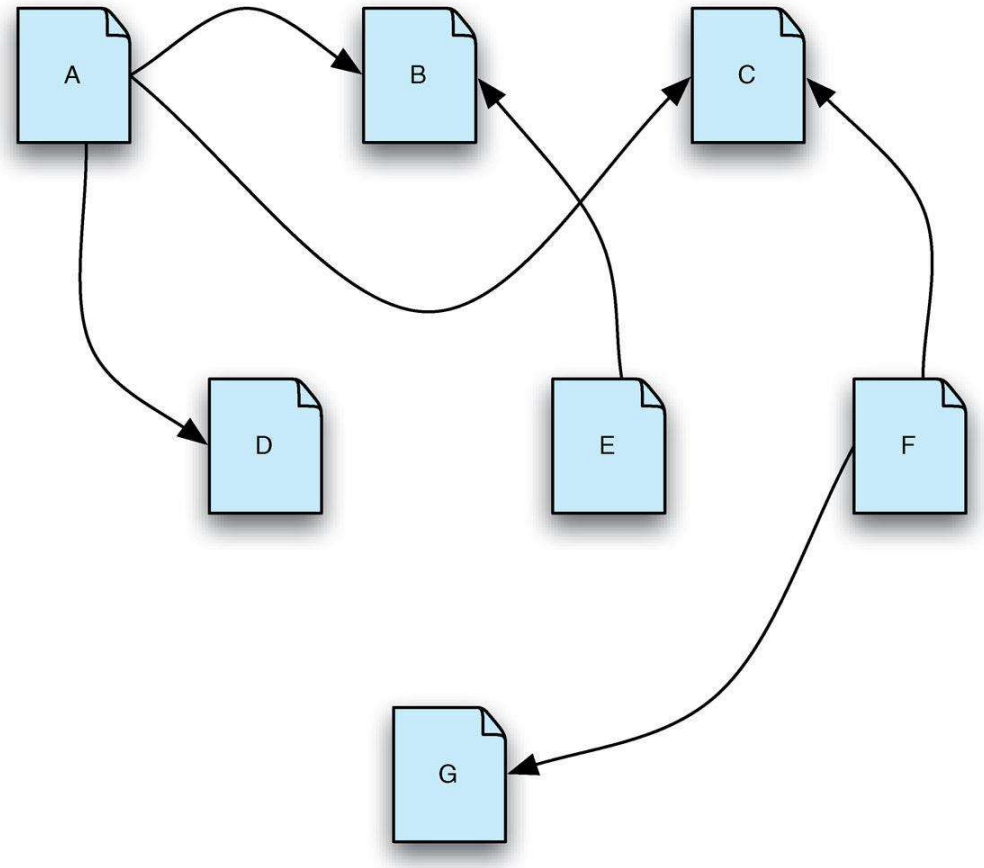
One View of the Web



Another View of the Web



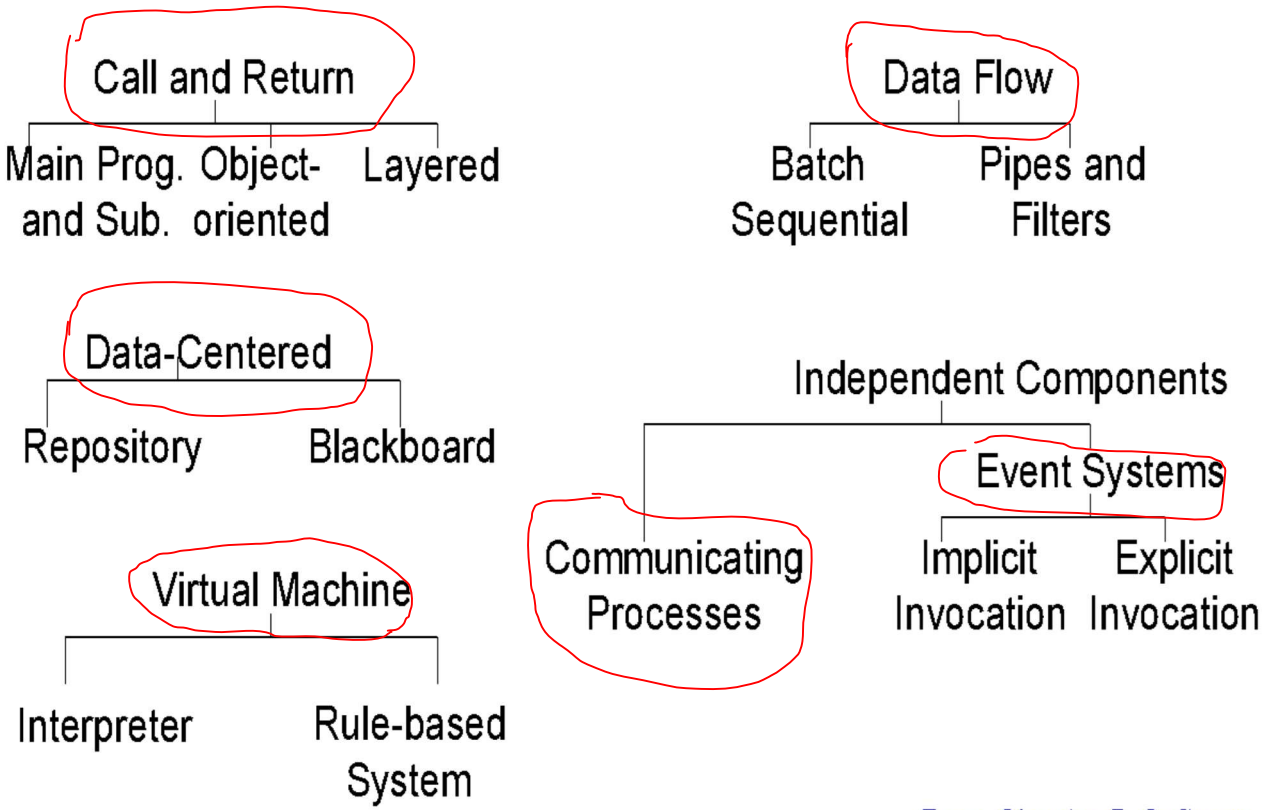
Yet Another View of the Web



Another Wiew of the Web!

DEI DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Architectural styles



From Chapter 5, Software Architecture in Practice, p. 95

Basic Concepts in Software Architecture

- Terminology
  - Architecture
  - Reference Architecture
  - Descriptive vs. Prescriptive Architecture
  - Component
  - Connector
  - Architectural Style
  - Architectural Pattern
- Models
- Processes
- Stakeholders

Architecture Definitions (from books)

- “The structure of the components of a program/system, their interrelationships, and principles and guidelines governing their design and evolution over time.” Perry & Wolf (probably the best definition)

- “A software system’s architecture is the set of principal design decisions made about the system.” (from the textbook)
- “Software architecture is the set of design decisions which, if made incorrectly, may cause your project to be cancelled.” – Eoin Woods

Architecture as a set of Design Decisions?

- One popular notion
- It's general – more than the structure of the system
- Design decisions include those related to:
  - System structure
  - Functional behavior
  - Interaction
  - Non-functional properties
  - Implementation

- Just about everything is a decision
- I don't particularly like this notion of architecture
  - It doesn't give me a good system-wide view
  - But it is still a useful concept

Reference Architecture

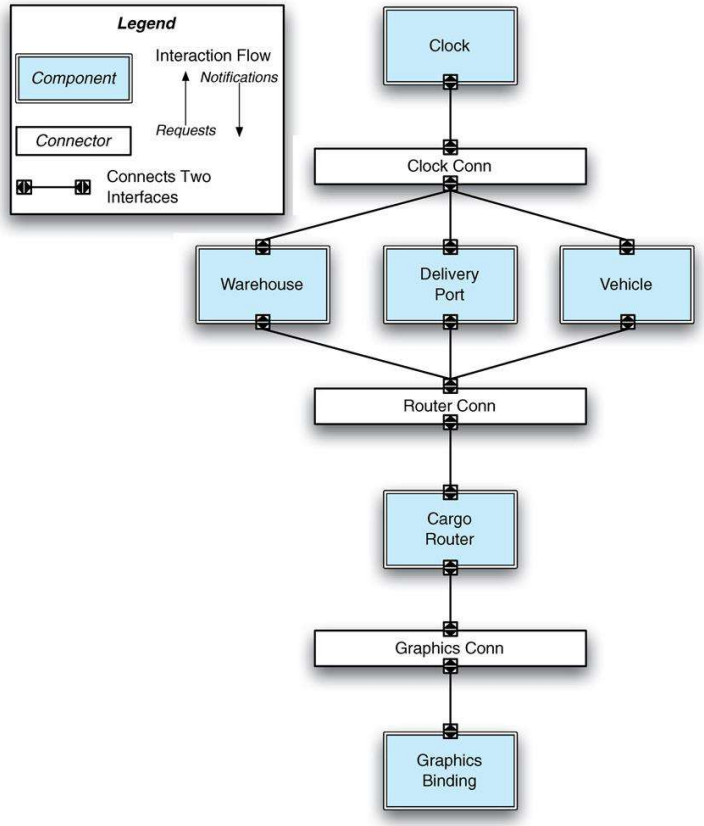
- A single software architecture for a family of related software systems
- “A reference architecture is the set of principal design decisions that are simultaneously applicable to multiple related systems, typically within an application domain, with explicitly defined points of variation.”

Descriptive vs. Prescriptive Architecture

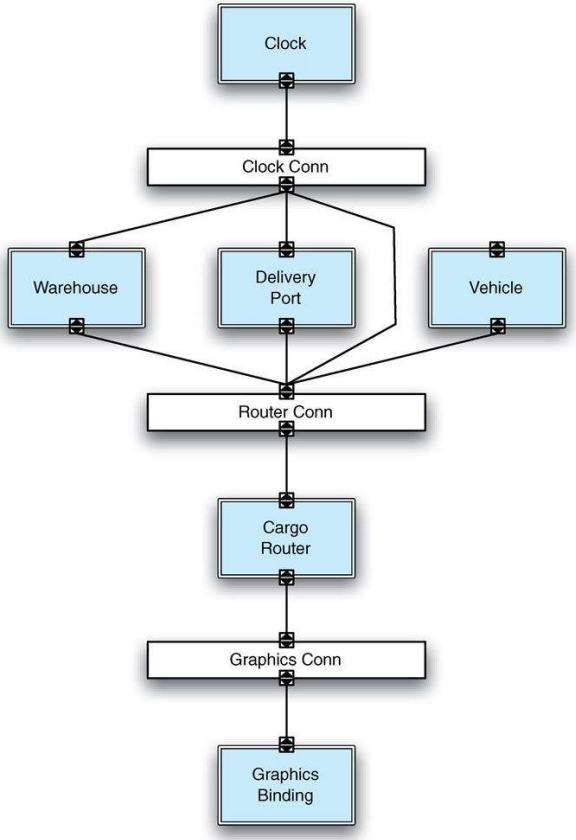
- Descriptive:
  - Describes what is there
  - Purposes:
    - Convey understanding of how a system is organized
    - Convey rationale for how the system is organized
    - Convey the theory of the program (includes the previous two)
    - Aid in maintenance and enhancement of the system
  - Prescriptive
    - Describes the structure of the system to be created
    - Establish the theory of the program
    - Specifies some technical/development issues

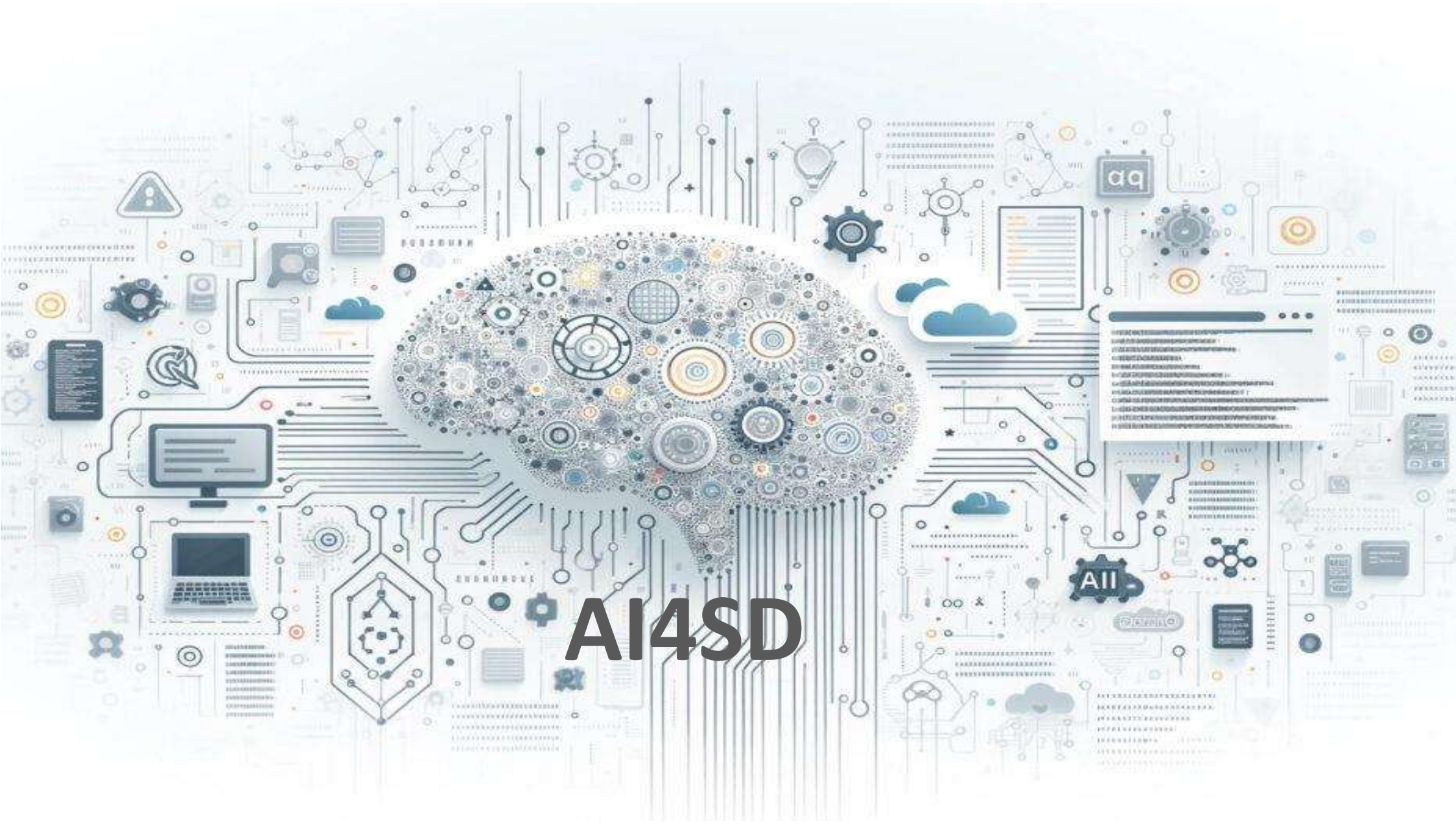
Prescriptive: as imagined

A cargo routing application

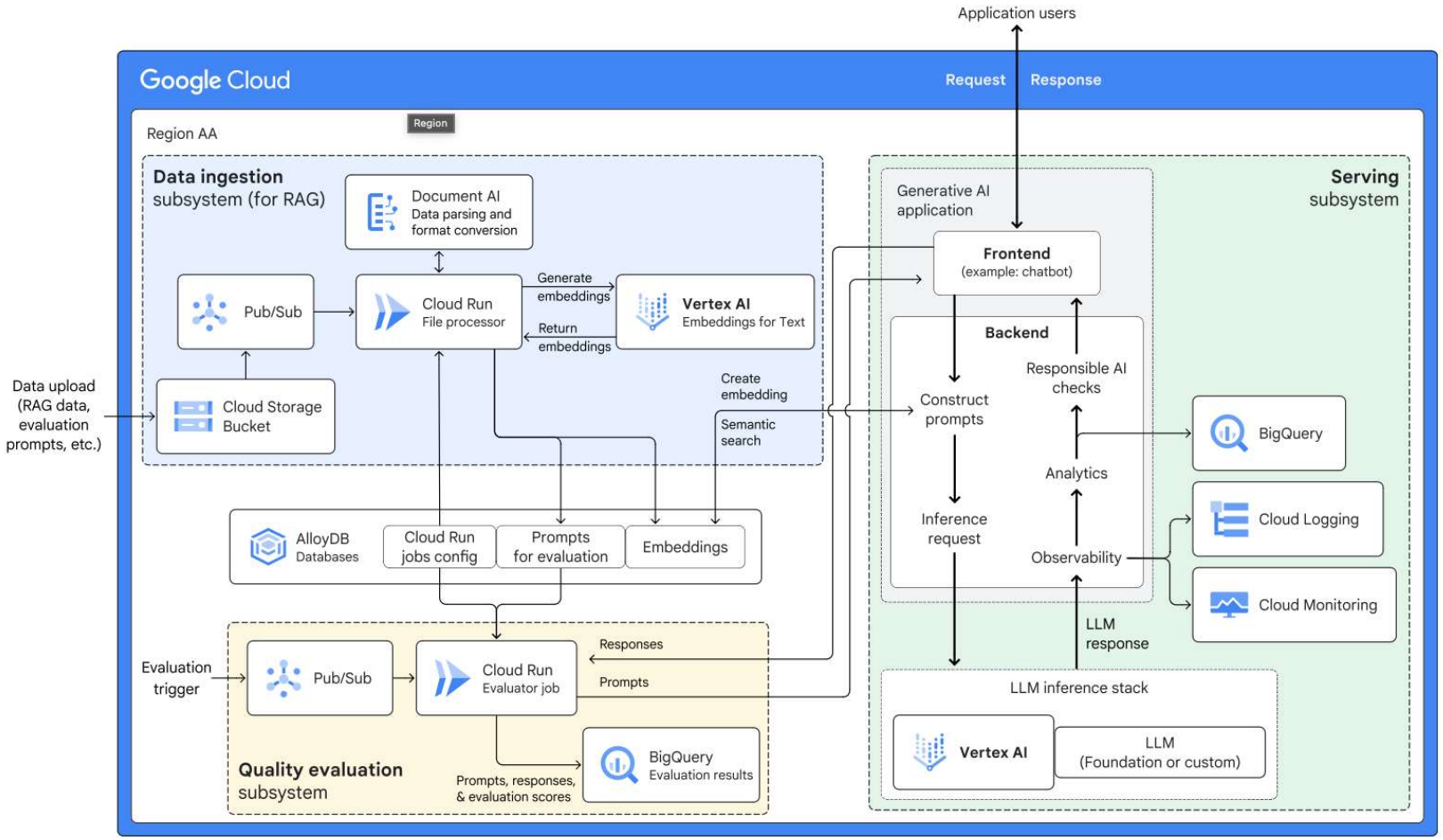


Descriptive: what was implemented




AI4SD

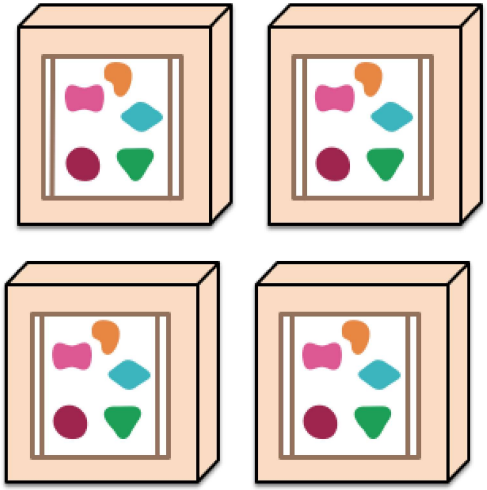
RAG at Google Cloud Platform…



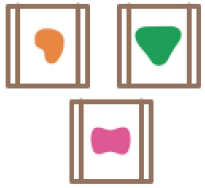
Microservices

A monolithic application puts all its functionality into a single process...

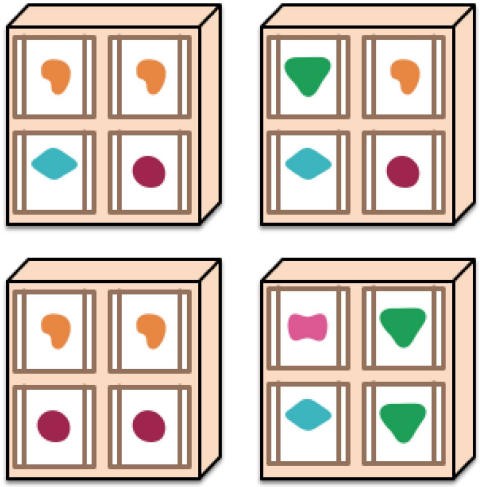
... and scales by replicating the monolith on multiple servers



A microservices architecture puts each element of functionality into a separate service...



... and scales by distributing these services across servers, replicating as needed.



https://martinfowler.com/articles/microservices.html

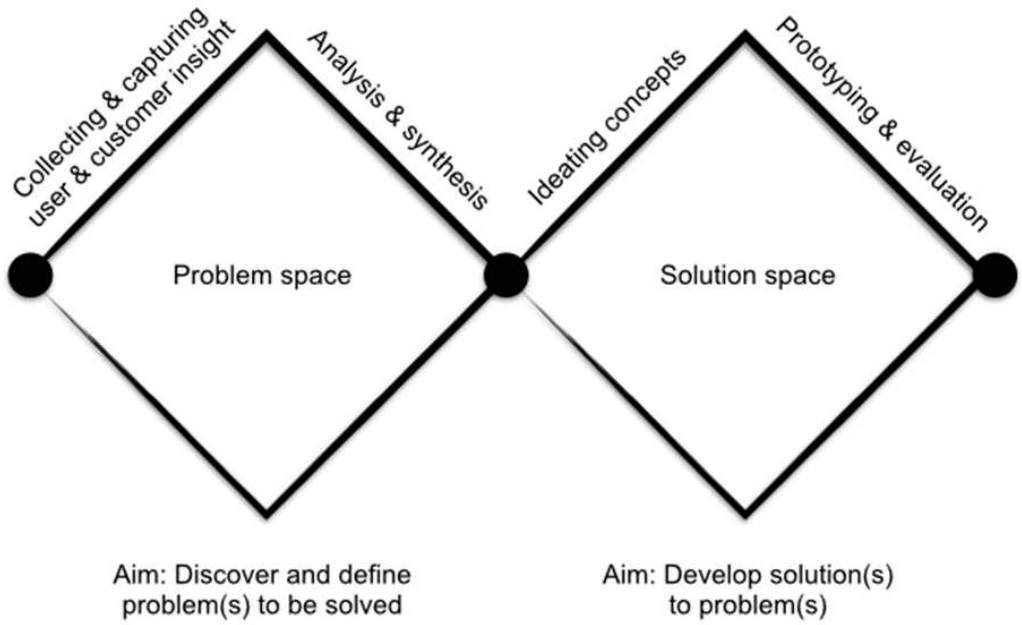
Do you Have any Questions?

- During Analysis and Architecture
  - We have many questions
  - Many things are unknown – by definition!

Why we Care about Questions?

- During architectural design, many questions arise, and many are answered.
- The nature of architectural design is that many things are unknown
  - Requirements may not be clear – or even known – yet.
  - Design is (by definition) not known yet.
  - How the project is managed may not be clear.

Problem Space, Solution Space



Questions appear in both spaces

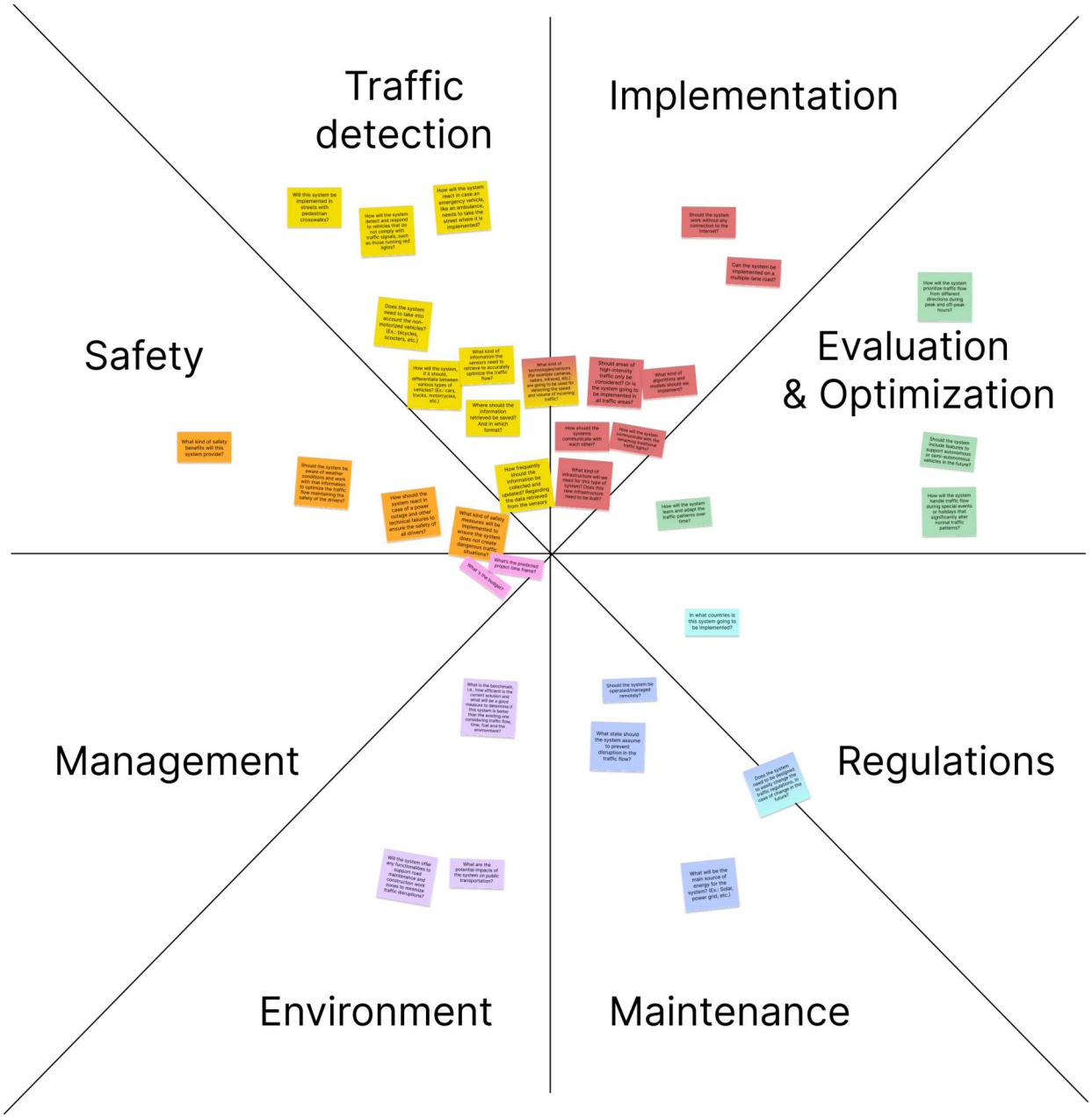
Intermingling of Architectural Questions

- Requirements Engineering: problem space
- Quality Attribute Analysis: mainly problem space
- Architectural Design: solution space

- Activities focus on problem space OR solution space, but questions come up both areas, intermingled

Potential problems of single focus

- What are the risks when a question about the design arises while you are focused on requirements engineering, for example?
  - Question might be ignored
  - Question must be conveyed to others, tracked somehow
  - If the question is blocking progress, do you make a “best guess”? (And what if the guess is wrong?)



FEUP.DEI

DEI - DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
dei.fe.up.pt | secdei@fe.up.pt | +351 225 082 134

FEUP - FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
www.fe.up.pt | feup@fe.up.pt | +351 225 081 400