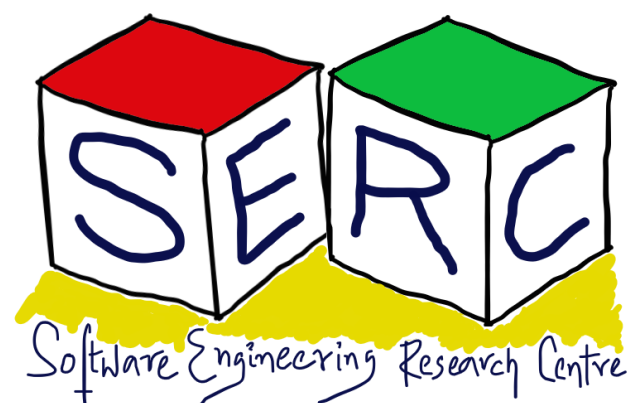


Introduction to Software Architecture

CS6.401: Software Engineering

Karthik Vaidhyanathan

<https://karthikvaidhyanathan.com>

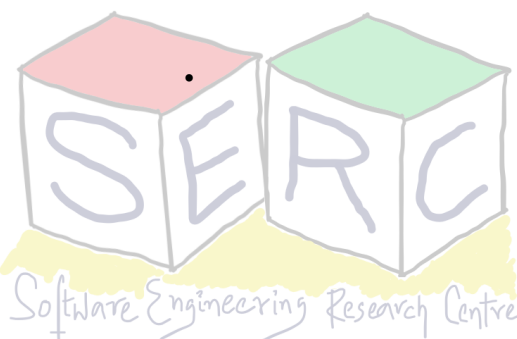


Acknowledgements

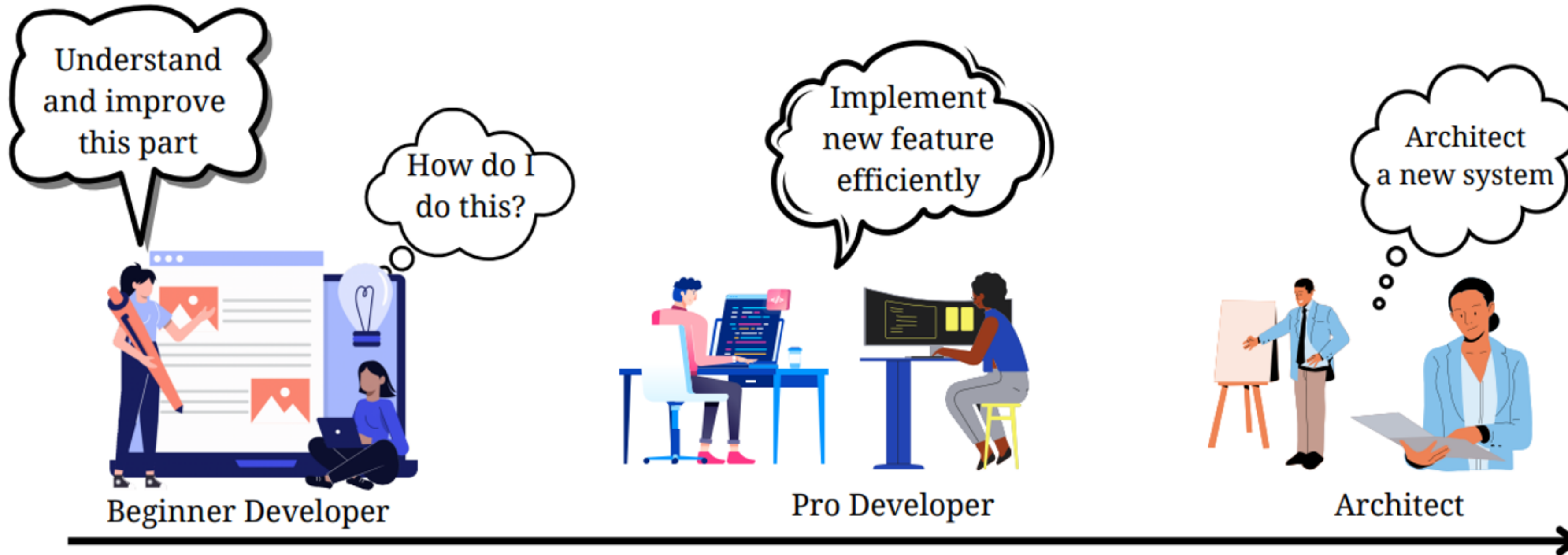
The materials used in this presentation have been gathered/adapted/generate from various sources as well as based on my own experiences and knowledge -- Karthik Vaidhyanathan

Sources:

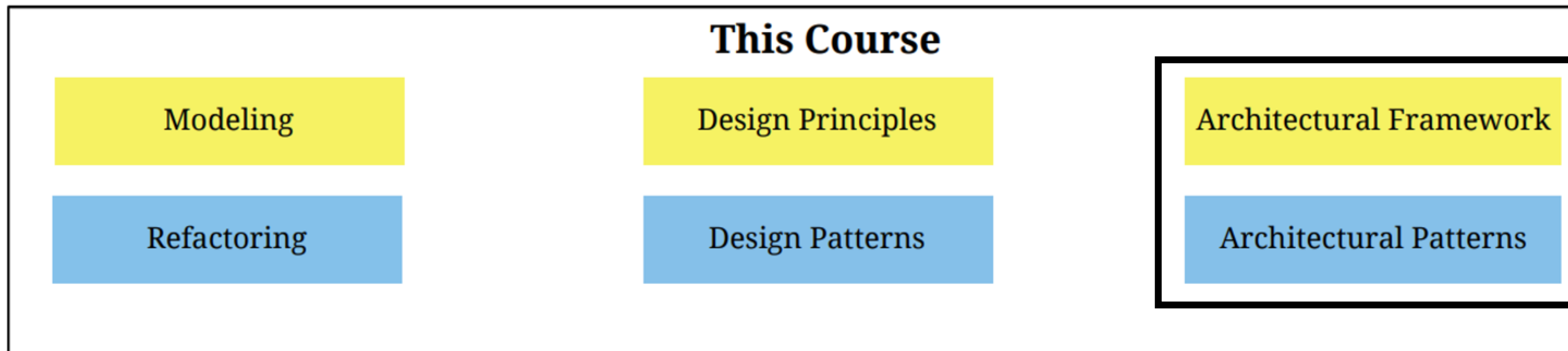
1. Introduction to Software Architecture, Henry Muccini, University of L'Aquila
2. Software Architecture in Practice, Len Bass, 3rd edition
3. Software Architecture (SE Course), Alessio Gambi, Saarland University, Germany
4. Software Architecture Design Reasoning Workshop, Antony Tang, ISAPS 2018



The Journey So Far

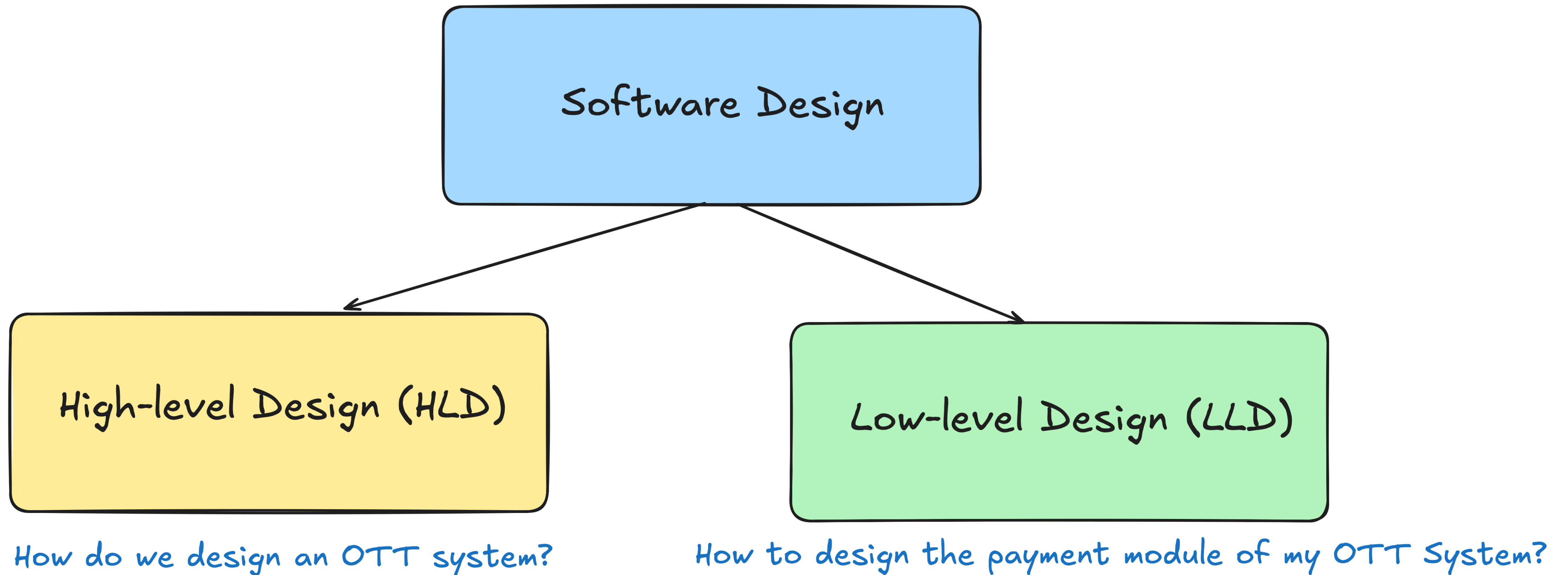


Career Timeline



Software Design

The function of good software is to make the complex appear to be simple. - Grady Booch



Lets start with a question

- **What, according to you, is “Software Architecture”?**
 - There is “Software”
 - There is also “Architecture”
 - We know about software and we know also about architecture
 - How do we put them together and what does it mean?

What will you do if you want to build a house?

- Family of 6
- Independent house
- Land of 40 cents
- Accomodate 6 + 2
-



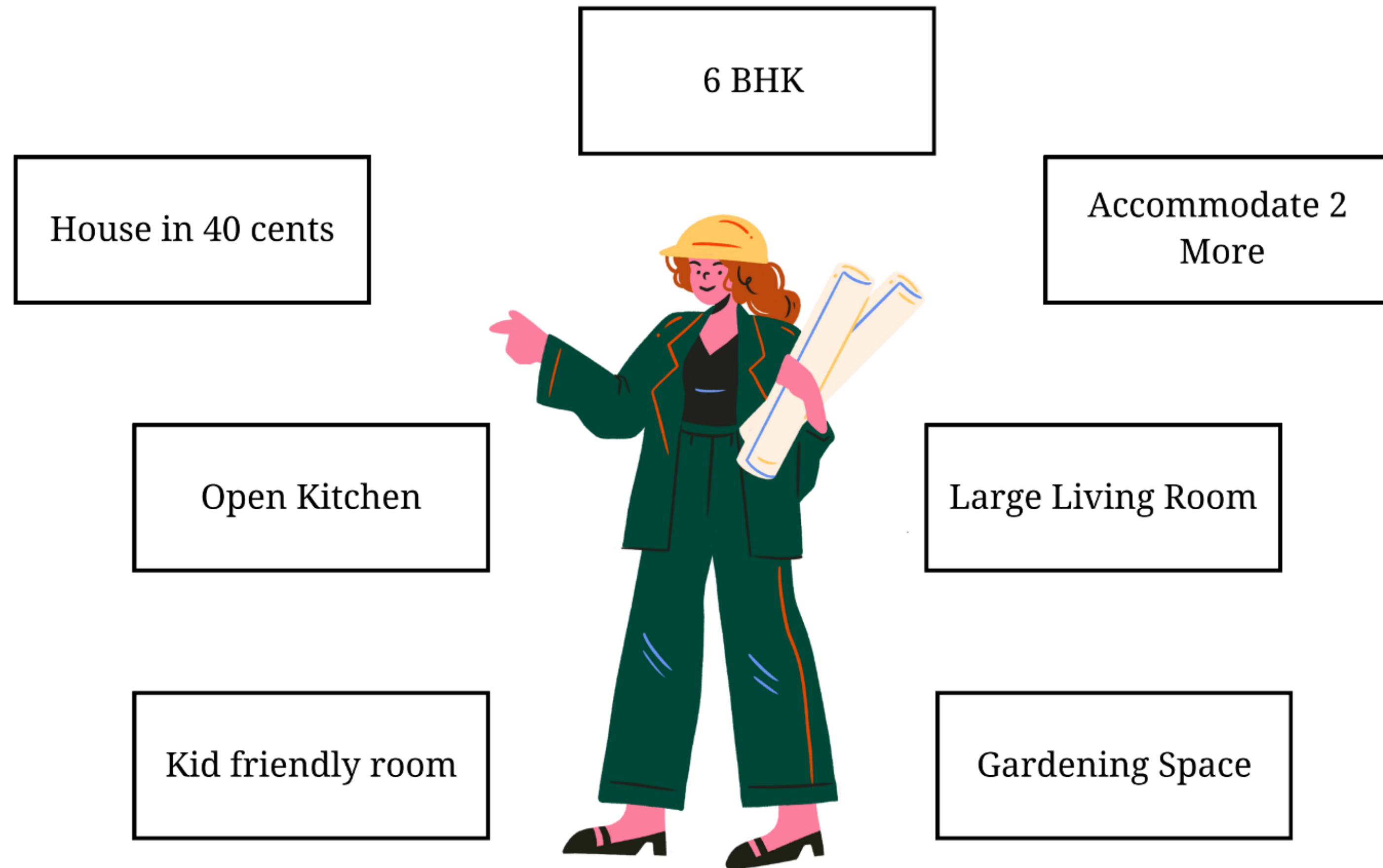
Give the requirements



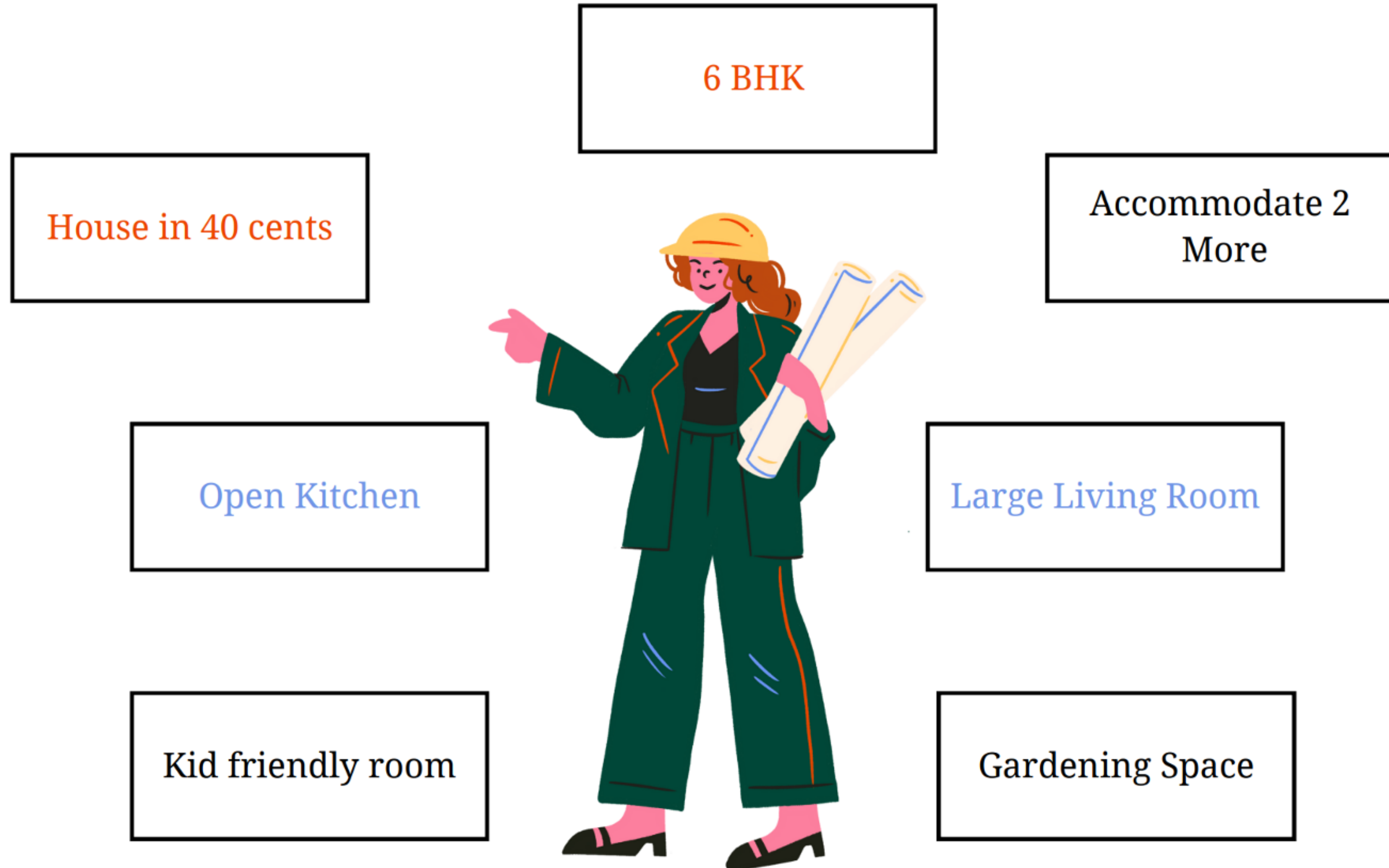
There are always multiple Stakeholders



Let me list all requirements



Let me See Constraints and Significant Requirements



Time to make some decisions

- What material to be used?
 - Budget
 - Weather condition...
- How to balance size?
 - Living room size
 - Kitchen size

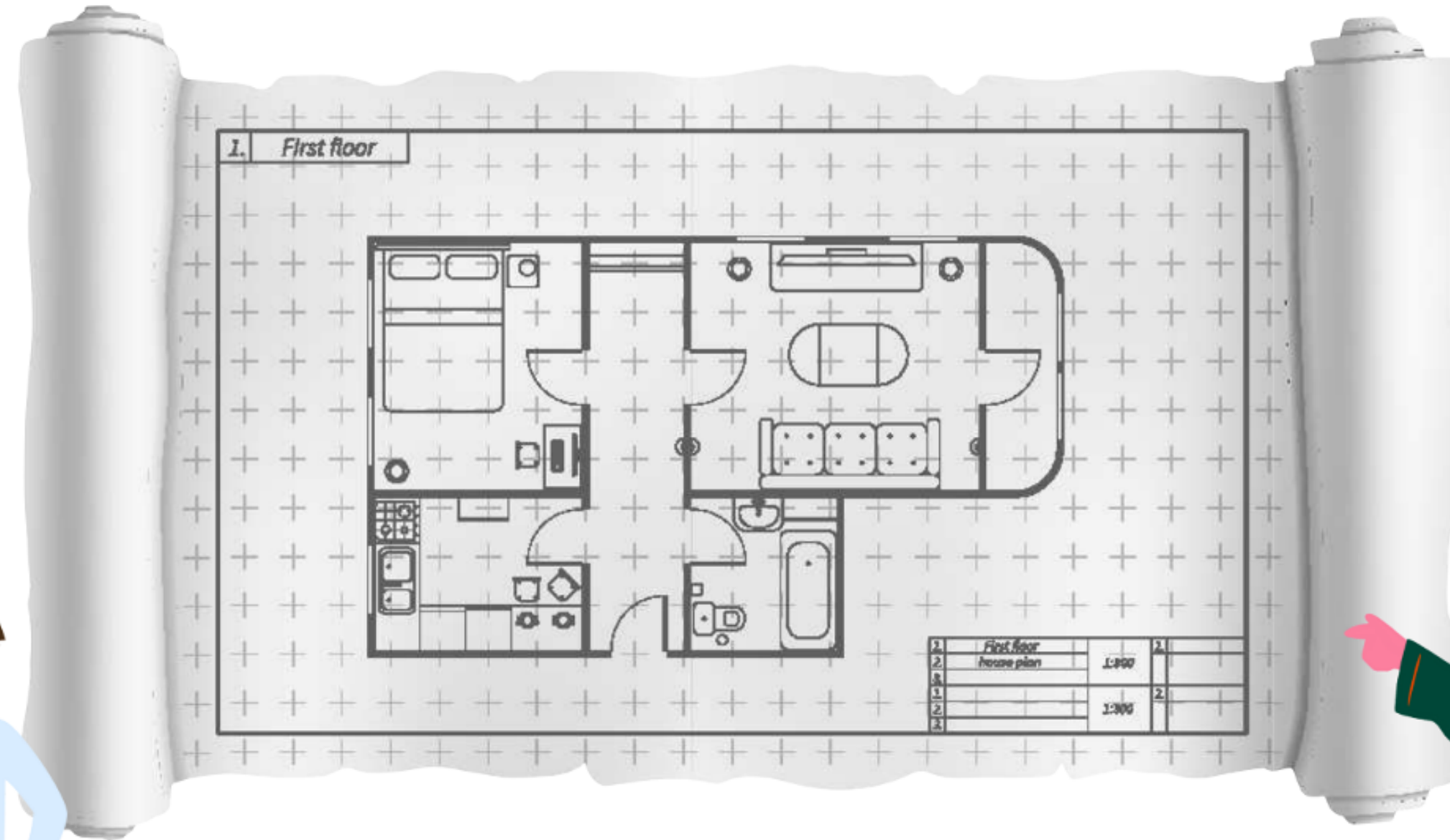
Design decisions!!



- What style of building?
 - Ancient style
 - Contemporary
 -
- What about stair elevation?
 - Roof length
 - Type of people
 -

High level Plan Ready

This is good
but can you provide
3D view for
others?



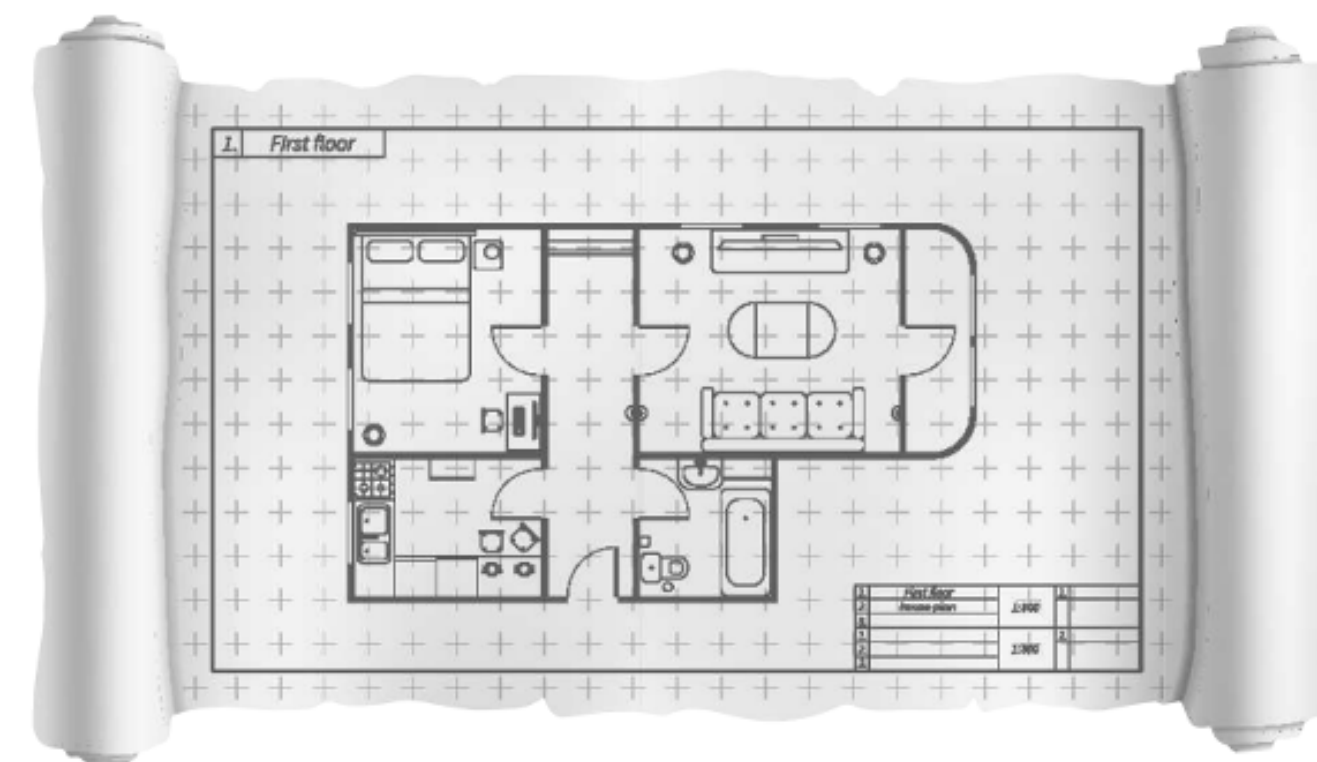
How's this?



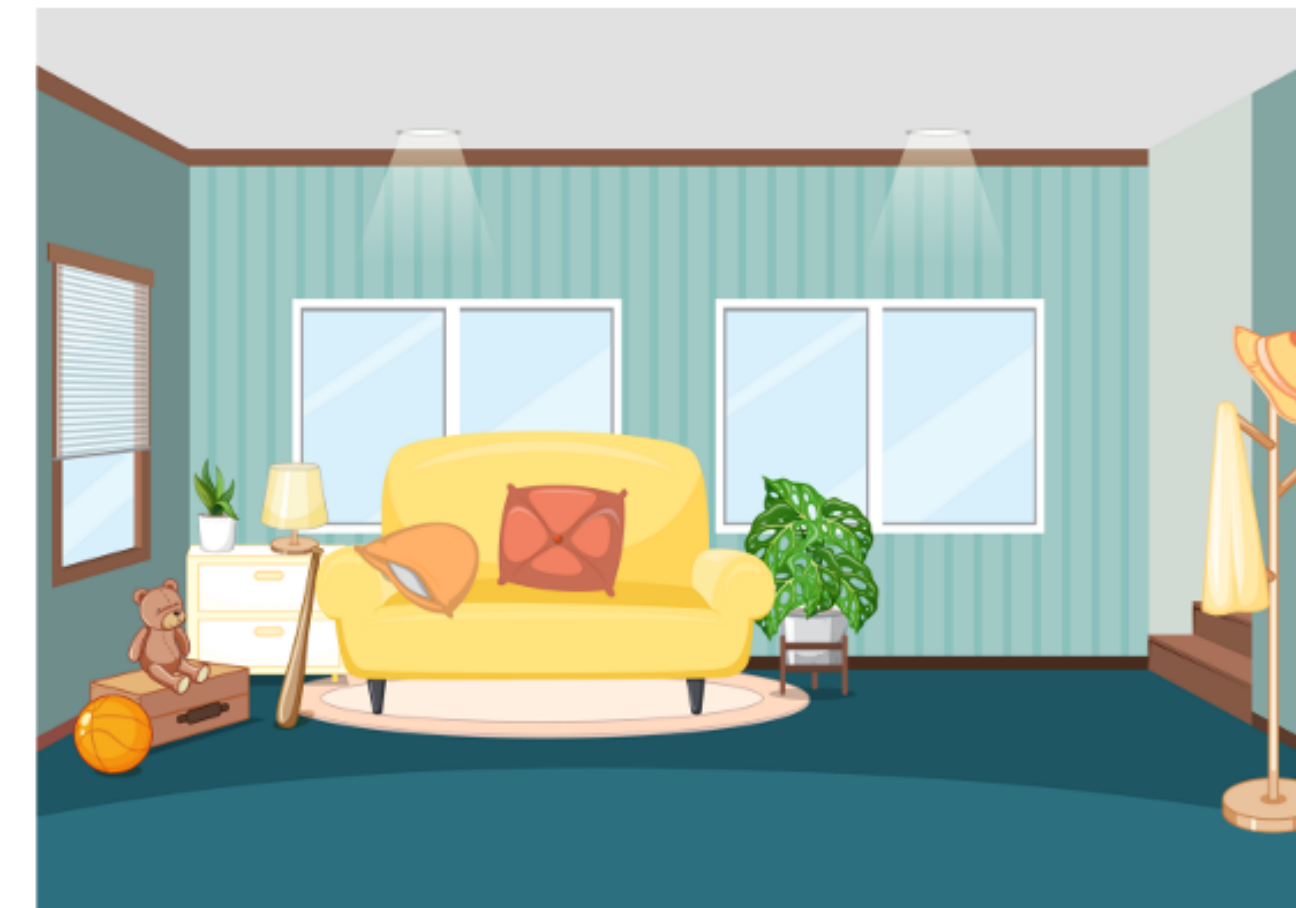
Multiple Views for Different Set of People



3D View



Detailed 2D plan



Interior View

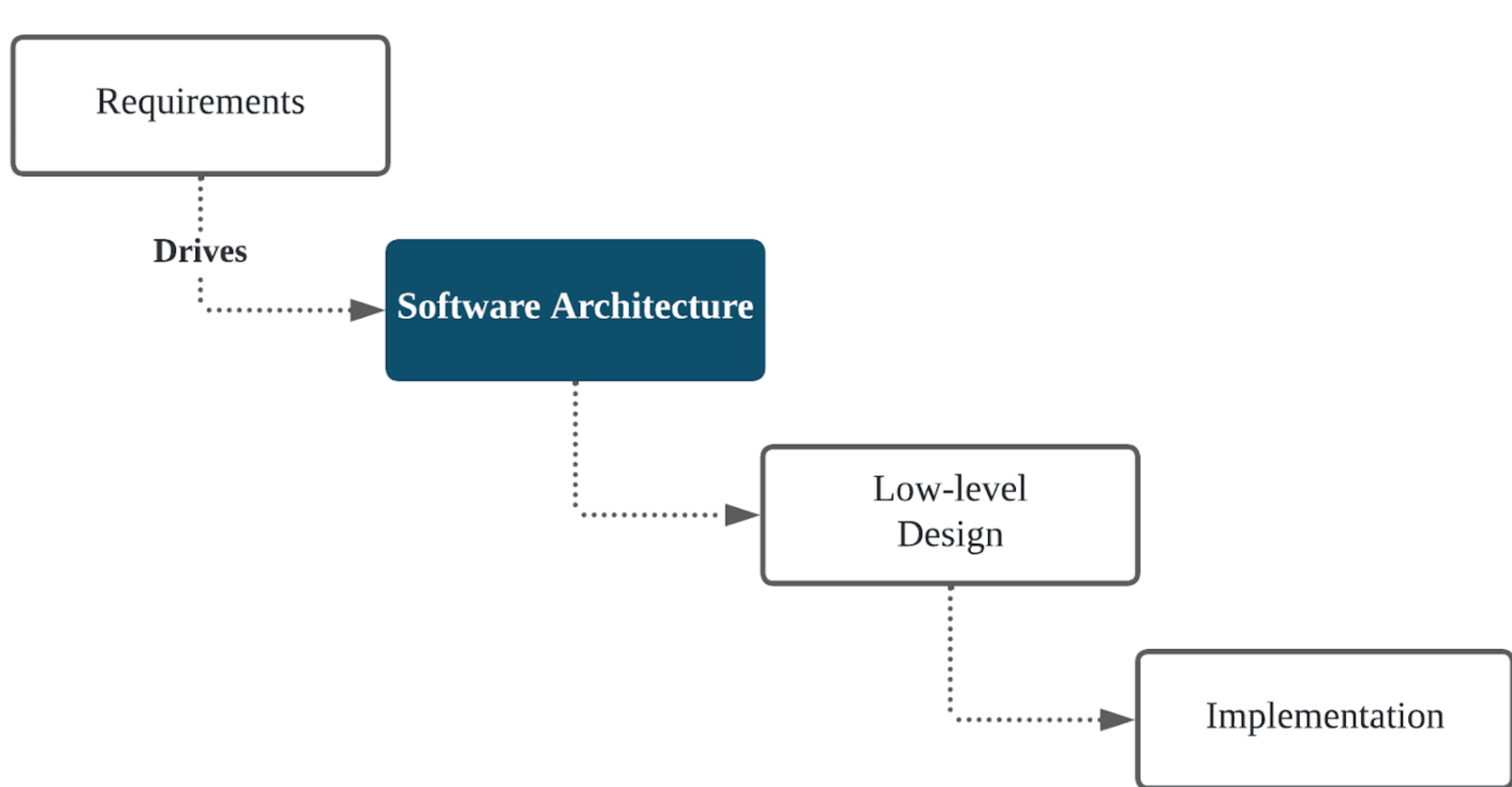
Views and Viewpoints

Construction happens and House is Ready!



Why don't we apply same to Software Design?

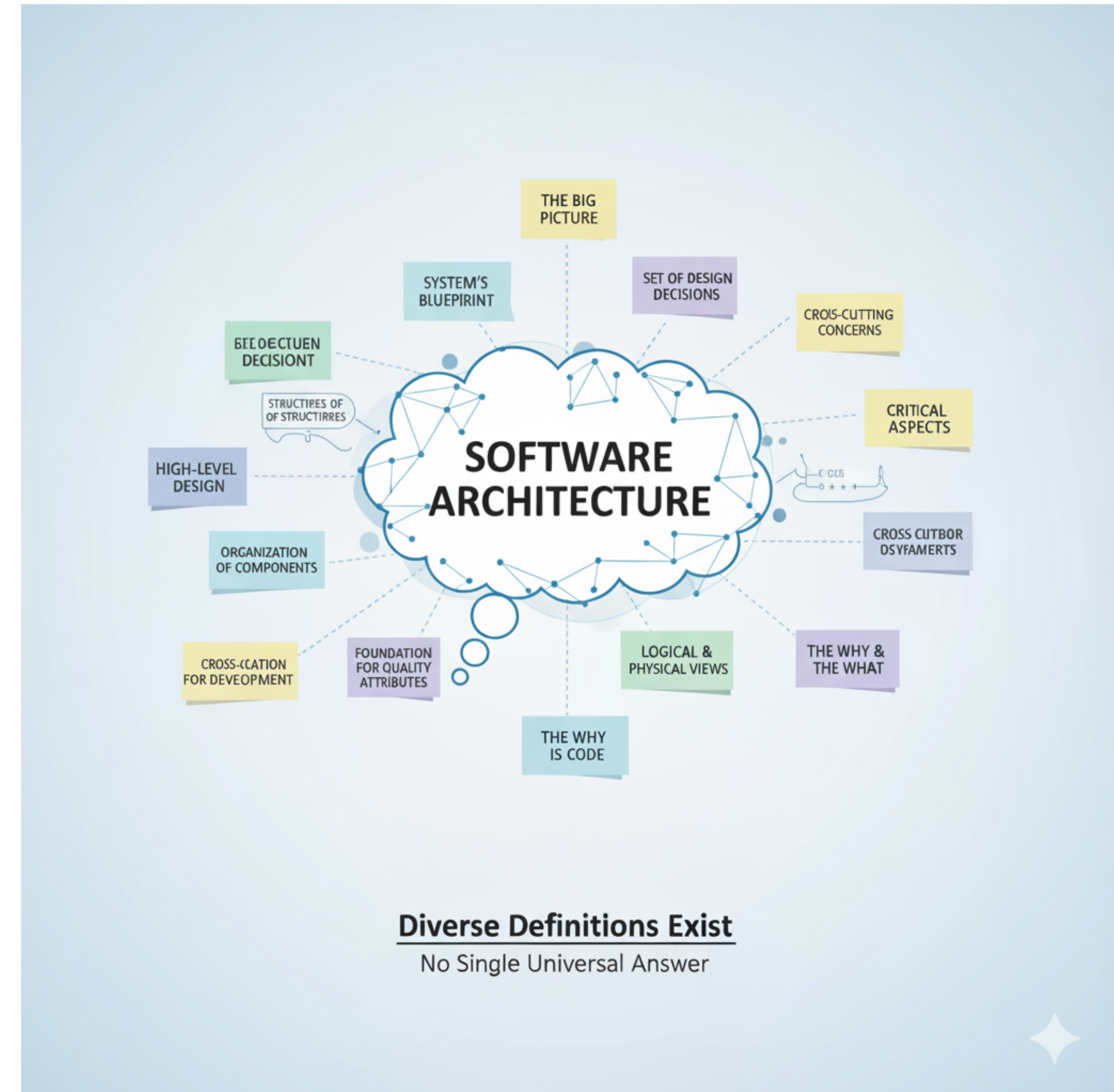
Where does Software Architecture come into the Picture?



What is Software Architecture?

Many different definitions exist for software architecture!

Check: <https://sei.cmu.edu/architecture/definitions.html>



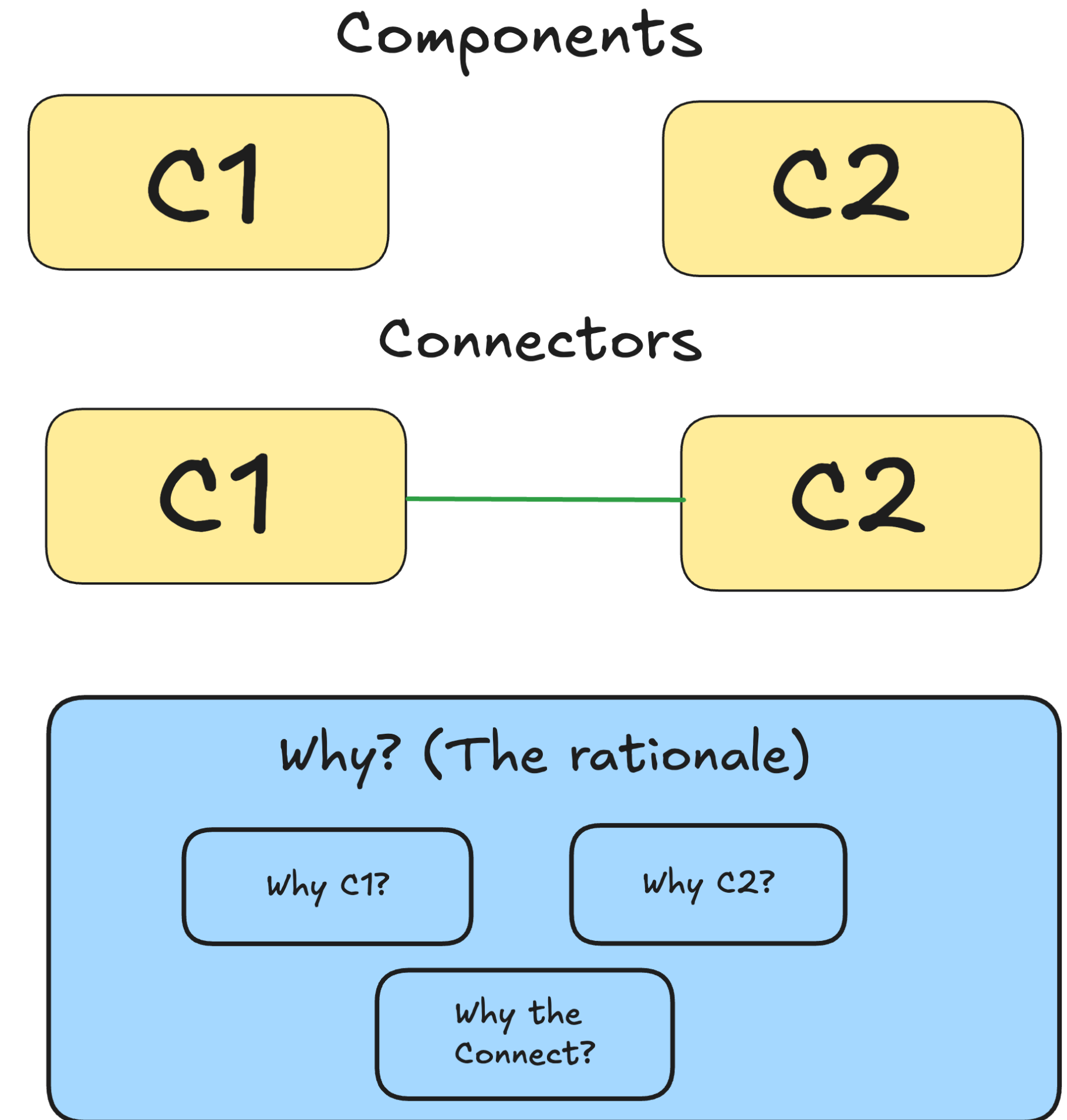
Software Architecture

Garlan and Shaw, '93:

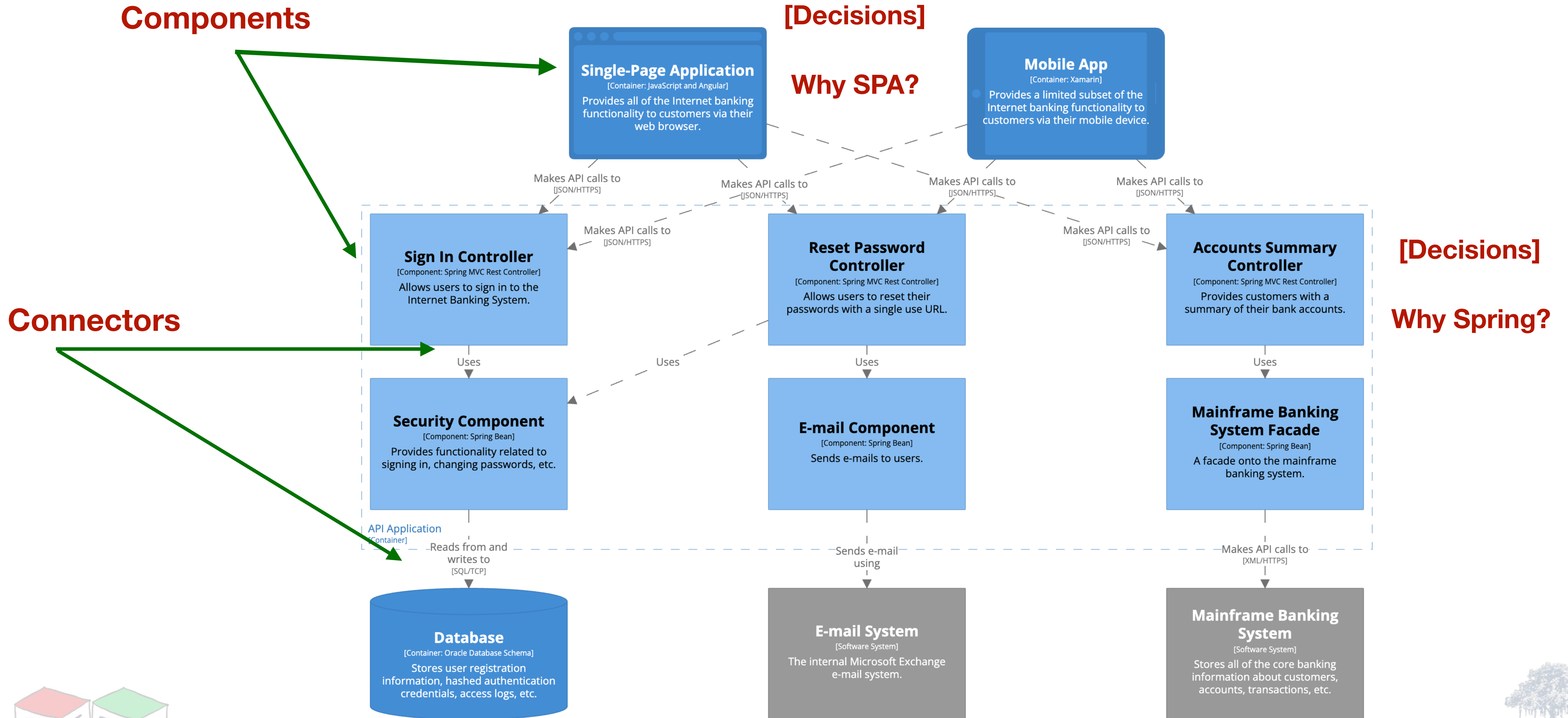
Architecture for a specific system may be captured as “**a collection of computational components** - or simply components - together with a description of the interactions between these components - the **connectors**”

Bass et al.:

"The software architecture of a program or computing system is the **structure or structures** of the system, which comprise **software elements**, the **externally visible properties** of those elements, and the **relationships** between them."



What is Software Architecture?



[Component] Internet Banking System - API Application

The component diagram for the API Application - diagram created with Structurizr.

Saturday, 11 November 2023 at 09:04 Greenwich Mean Time

Software Architecture

ACM SIGSOFT

SOFTWARE ENGINEERING NOTES vol 17 no 4

Oct 1992 Page 40

Foundations for the Study of Software Architecture

Dewayne E. Perry

Alexander L. Wolf

AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, New Jersey 07974
dep@research.att.com

Department of Computer Science
University of Colorado
Boulder, Colorado 80309
alw@cs.colorado.edu

© 1989,1991,1992 Dewayne E. Perry and Alexander L. Wolf

Software Architecture = {Elements, Form, Rationale}



Software Architecture

The Software Architecture is the **earliest model** of the **whole software system** created throughout the software lifecycle

A set of **components and connectors** communicating through an interface

A set of architectural **design decisions**

Focus on a set of **views and viewpoints**

Developed according to **architectural styles**

Interesting Rather Relevant Decisions

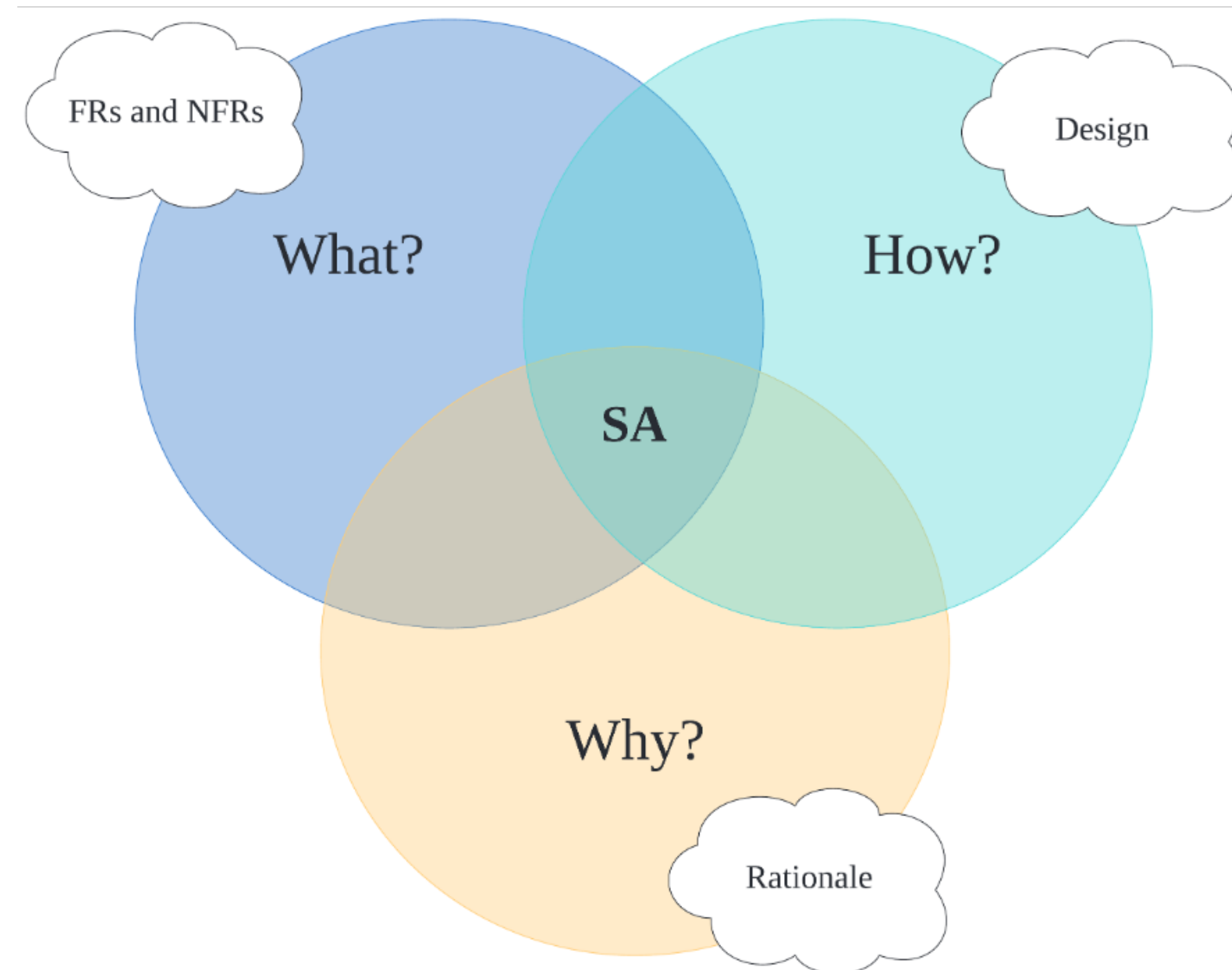
“The shared understanding that the expert developers have of the system design”

”The decisions you wish you could get right early in a project”

“Software Architecture is about **all the important stuff**, whatever that is”



Why Software Architecture? Abstraction



Manage complexity in the design (Transferable abstraction, promotes reuse)

Why Software Architecture? Communication

Software Architect



Software Architecture



Business



IT Team

Document, remember and communicate among stakeholders

Manifest early design decisions

Quality Analysis

Sustainability

Availability

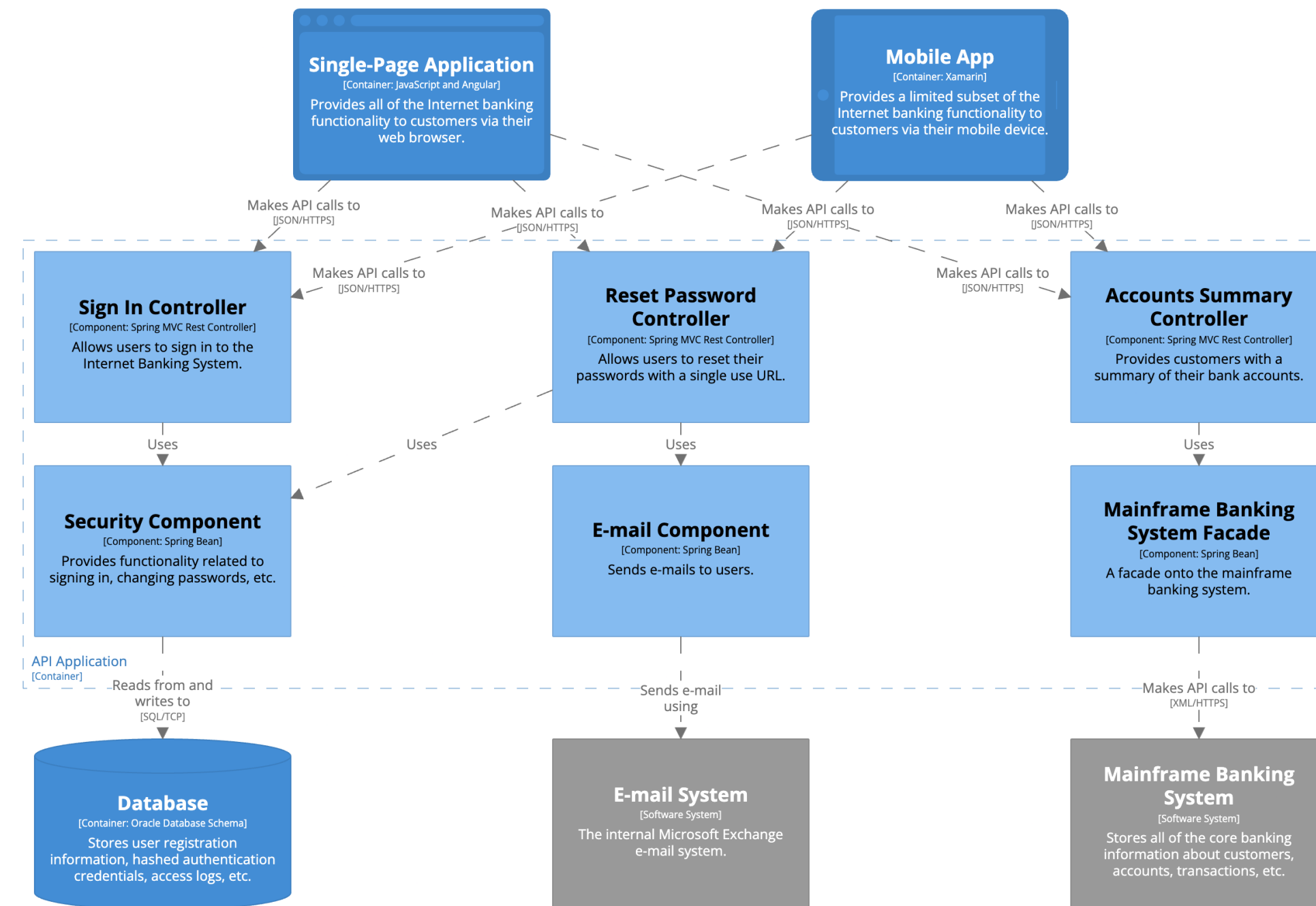
Portability

Reliability

Performance

Usability

Testability



[Component] Internet Banking System - API Application
 The component diagram for the API Application - diagram created with Structurizr.
 Saturday, 11 November 2023 at 09:04 Greenwich Mean Time

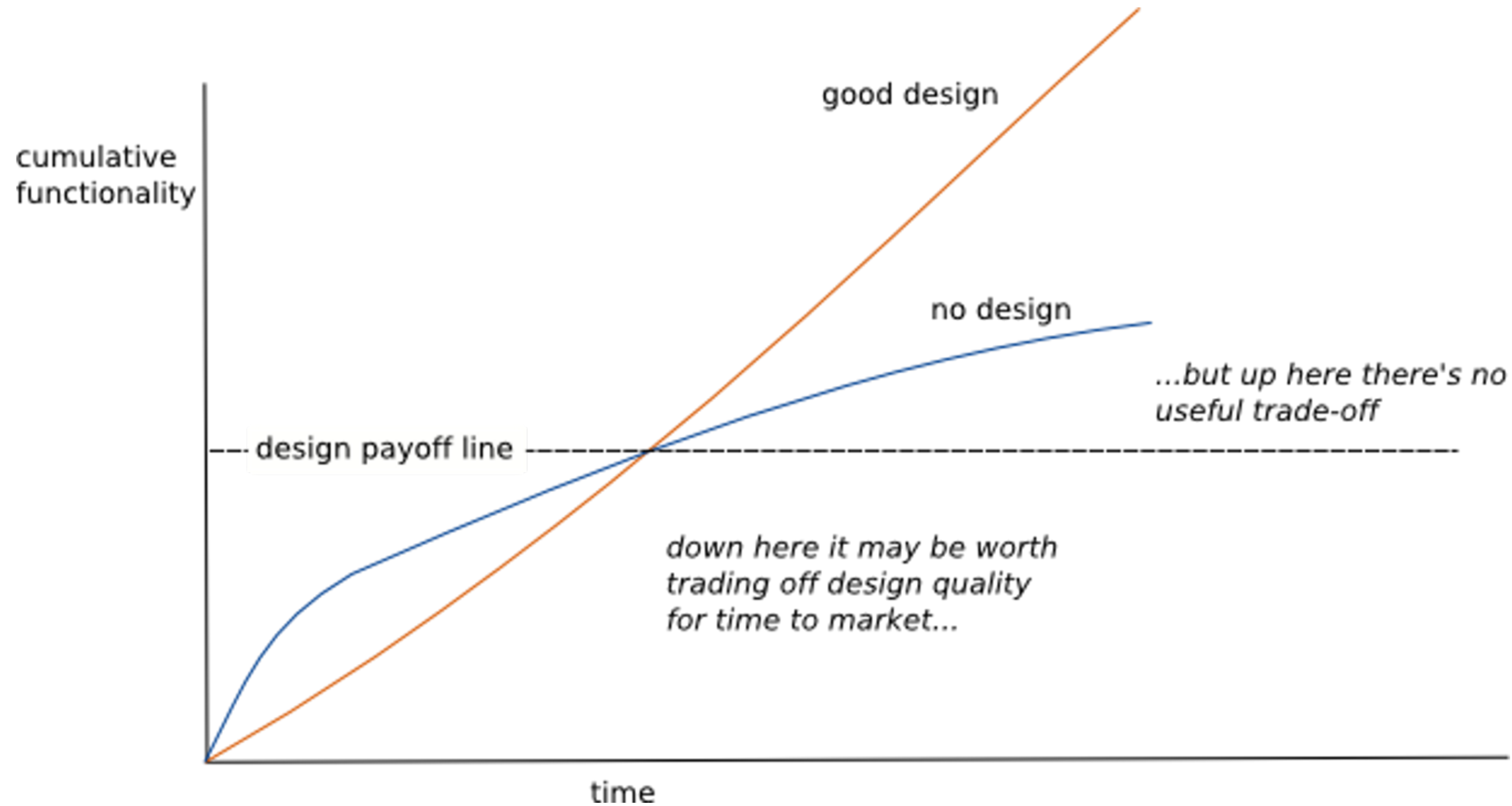
Security

Maintainability

Understand, Predict and Control

Why to Care about Software Architecture?

Design Stamina Hypothesis



Why to Care about Software Architecture?

The case of Therac 25



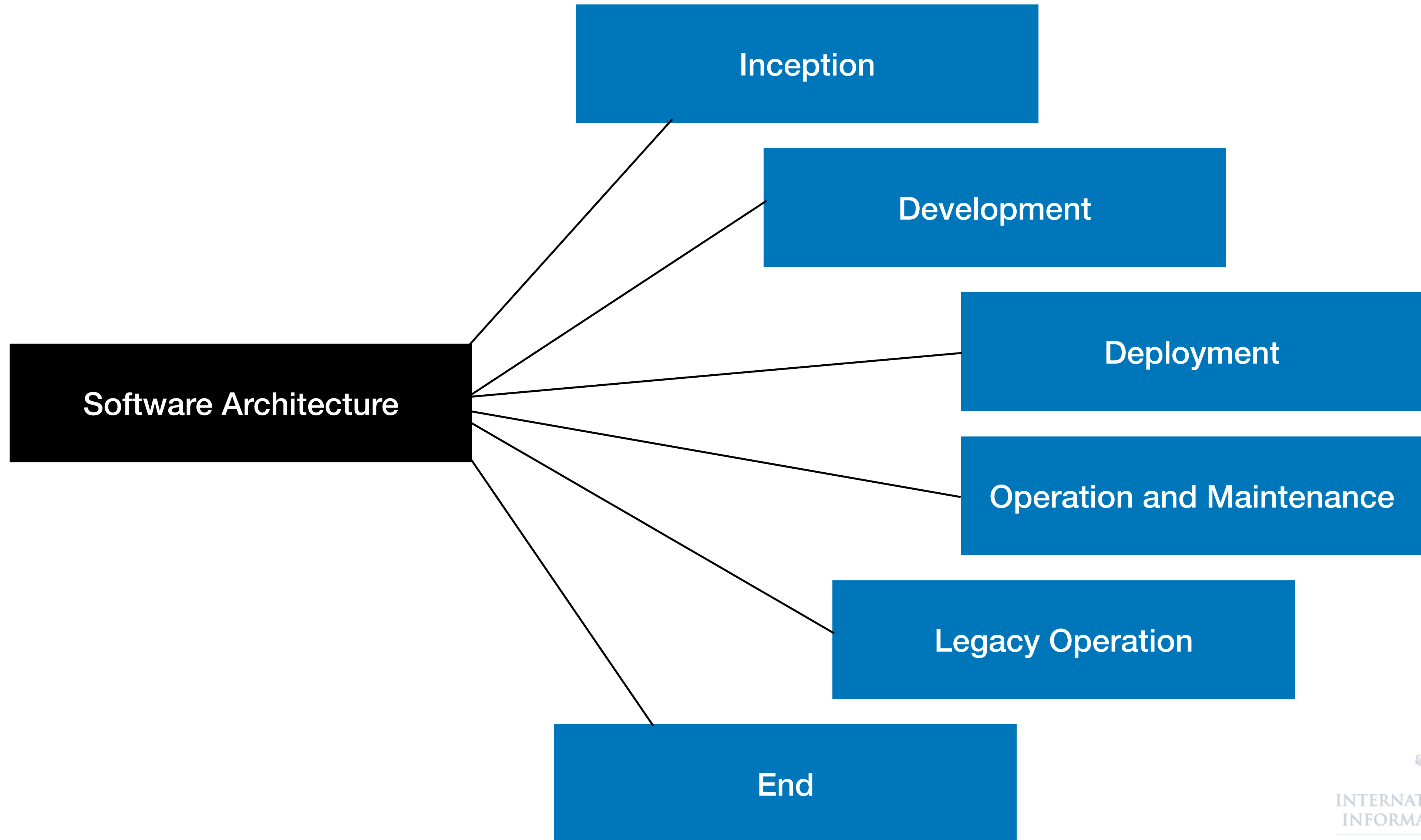
- Approx. 100 times the intended dose of radiation
- About 3 people died, and 3 got severely injured
- Overconfidence in software!
- Inadequate software engineering practices
- Lack of defensive design

Image generated by GPT-5

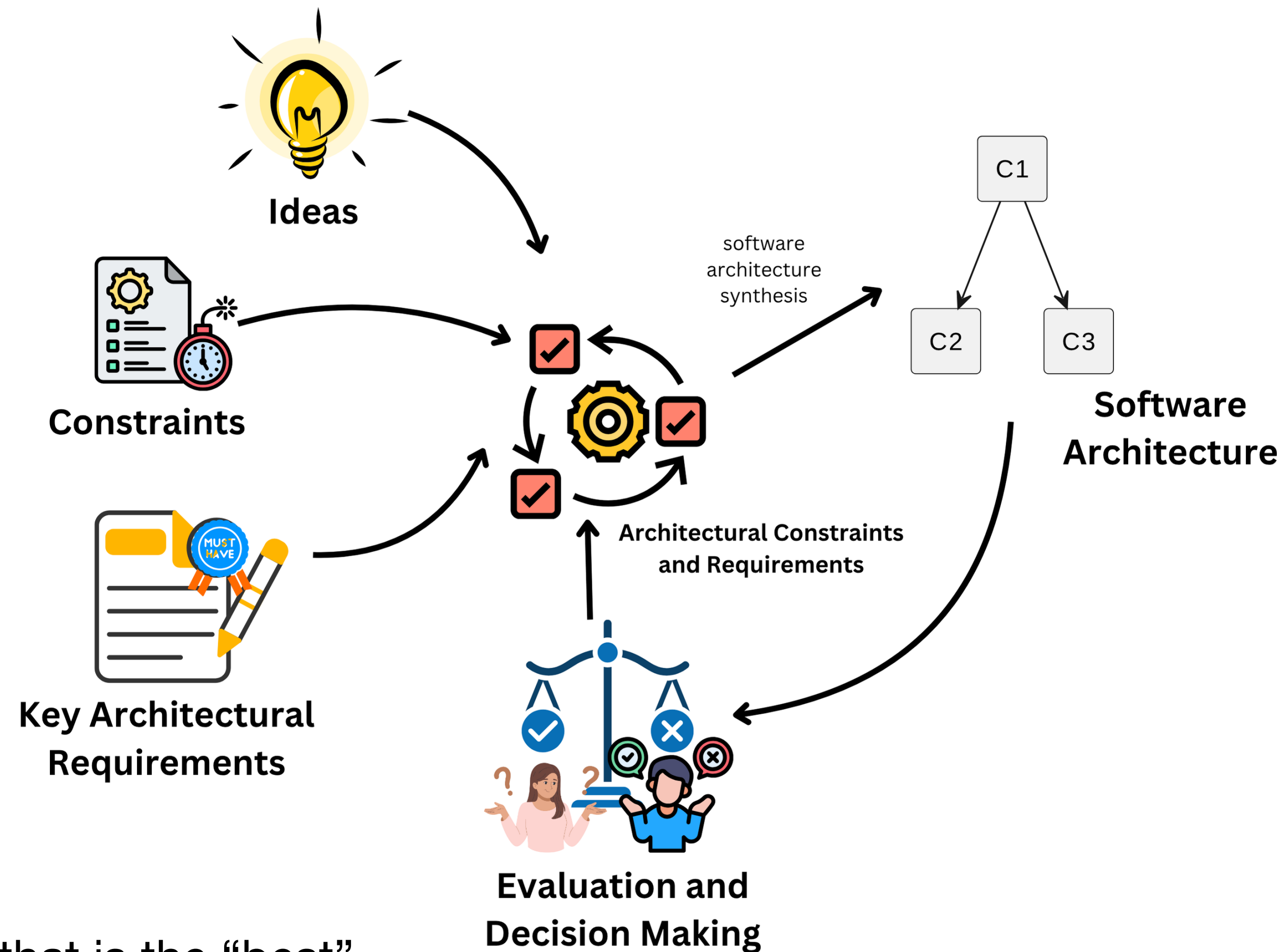
Why to Care about Software Architecture?

- All software systems have an architecture
- All the **critical/complex** systems must have it carefully and explicitly specified
- Architecture-level decisions impact the **scalability, performance, testability,** and **functioning** of the produced system
- Even if the code is perfectly written, a wrong architecture produces a wrong system

When does Software Architecture End?



The Overall Architecting Process



- There is no architecture that is the “best”
- Look for the one that satisfies, at best, the requirements and constraints
- The “less” risky one

Architecting a System

Case of a Museum Crowd Management System

- Historical sites in any part of the world get lot of visitors
- One of the most visited in any country
- Booking systems are not dynamic
- Crowding happens sometimes, even resulting in stampedes
- **Goal:** Build a crowd management system for historical sites



Requirements for the System

Functional Requirements:

FR1: User Registration

FR2: Check Availability

FR3: Entry booking

FR4: Recommendations

...

Non-Functional or Extra Functional Requirements

EFR1: Performance – Latency/request < 0.1 sec

EFR2: Security – Prevent unauthorized access

EFR3: Availability – 99.999%

EFR4: Scalability – 1000 users/second

**Let key requirements drive
high-level design of the system!**

Software Architecture

The Software Architecture is the **earliest model** of the **whole software system** created throughout the software lifecycle

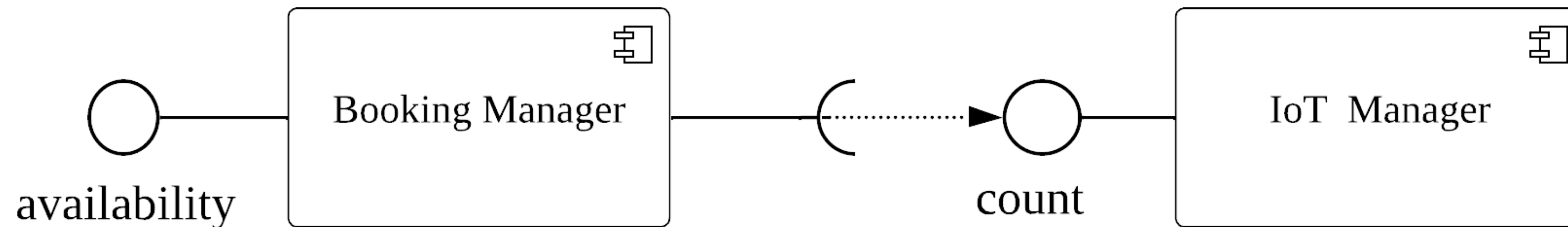
A set of **components and connectors** communicating through an interface

A set of architectural **design decisions**

Focus on a set of **views and viewpoints**

Developed according to **architectural styles**

Components and Connectors



network calls

Components:

- Data or processing element
- Has a **provided** and **required** interface

Eg: database, client, server, etc.

Connectors:

- Enables interaction among components
- Can be implicit or explicit

Eg: HTTP events, procedure calls, etc.

Design Decisions

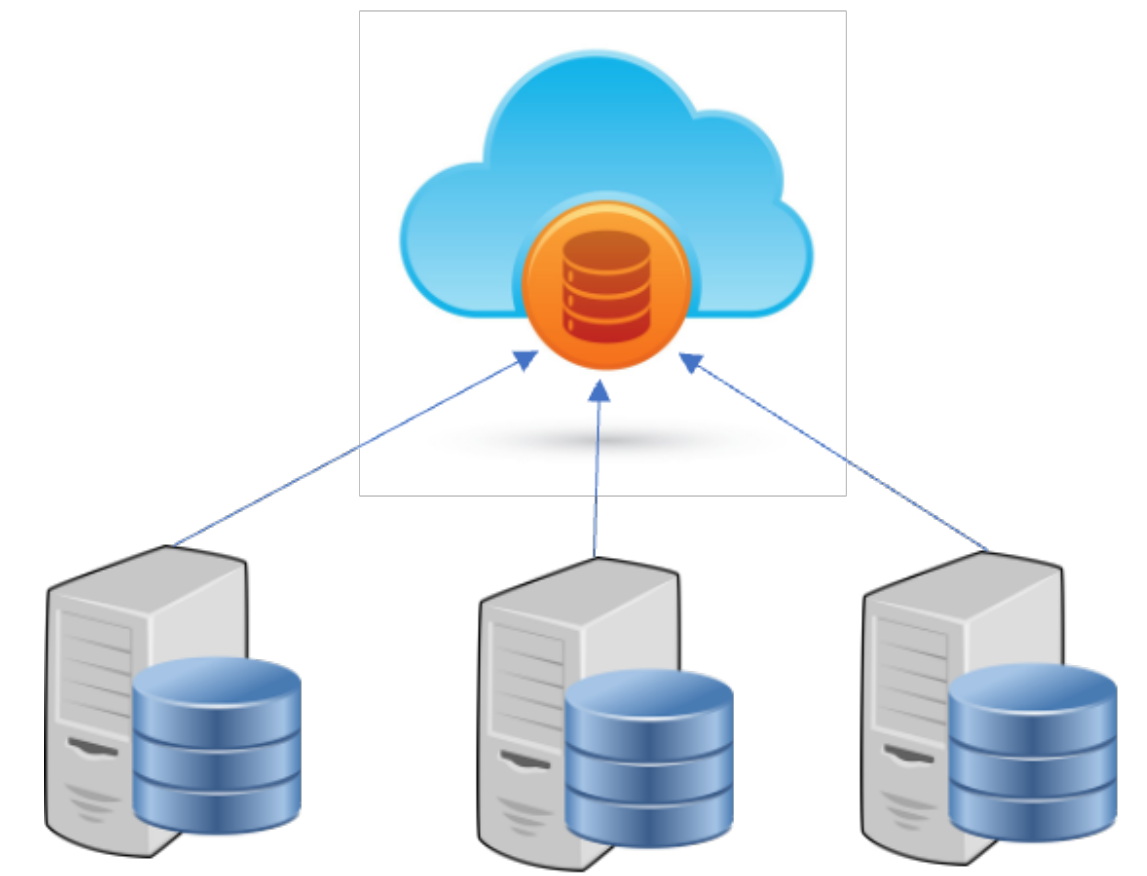
Revisit our case: What to Choose?



Store data in cloud



Store data in local Fog



Each site has its data

- Each decision has implications on performance, security, privacy, etc.

Reasoning with Simple Logic May not work

- Oracle is more scalable than MySQL
- MySQL is more scalable than Informix

Therefore, Oracle is more scalable than Informix

Q: I need a scalable RDBMS, Shall I go with Oracle?

A: It depends!!!



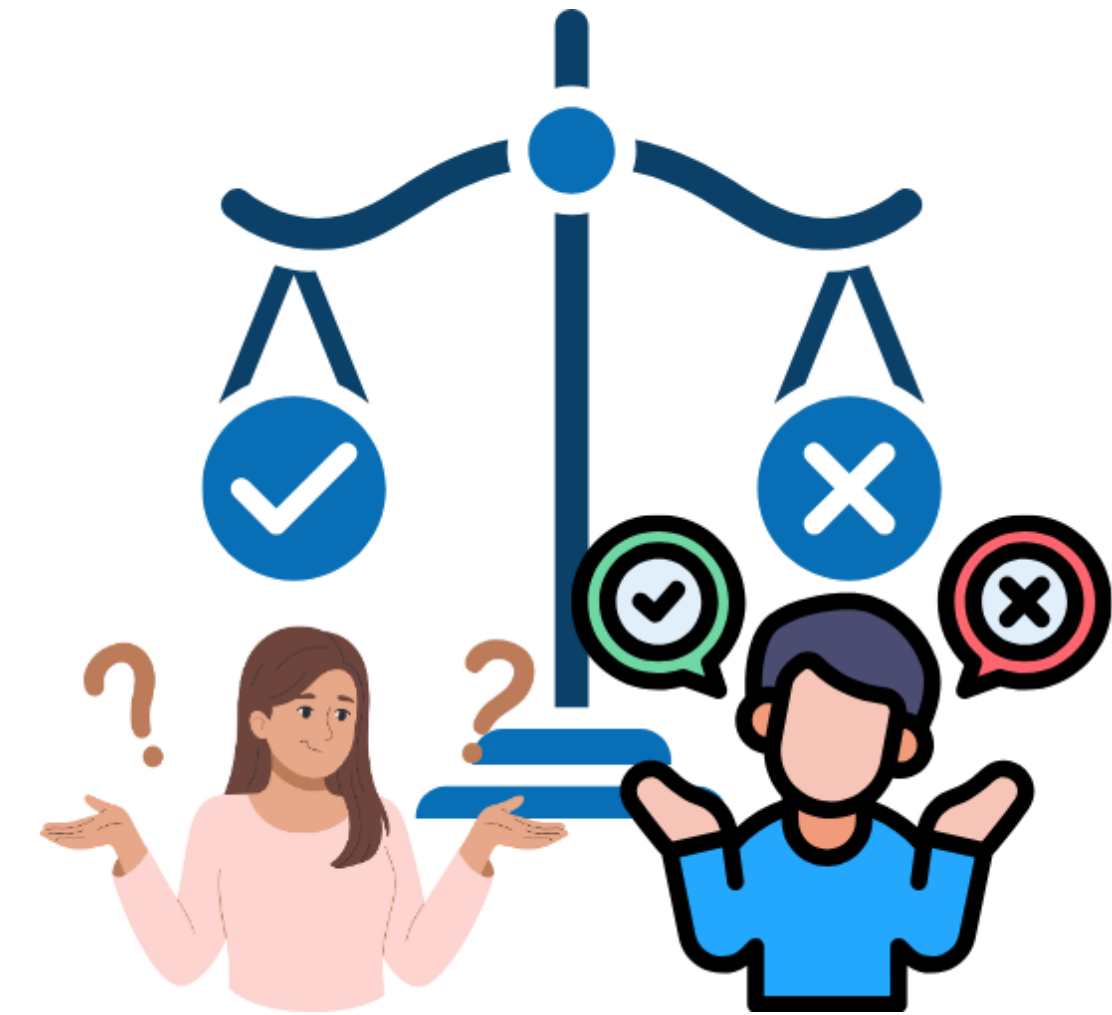
Architectural Design Decisions

- Decisions about

Selected components/interfaces/connectors
Distribution/Configuration of components/connectors
Expected behavior
SA Styles, Patterns and Tactics
HW/SW/Deployment and other views
Components' Nesting and sub-systems
NF attributes

Consequences of Design Decisions

- Defines constraints on implementation
- Dictates organizational structure
- Inhibits or enables system's quality attributes
- System qualities may be predicted
- Easier to manage change
- Helps in evolutionary prototyping
- Enables cost and schedule estimates



Software Architecture

The Software Architecture is the **earliest model** of the **whole software system** created throughout the software lifecycle

A set of **components and connectors** communicating through an interface

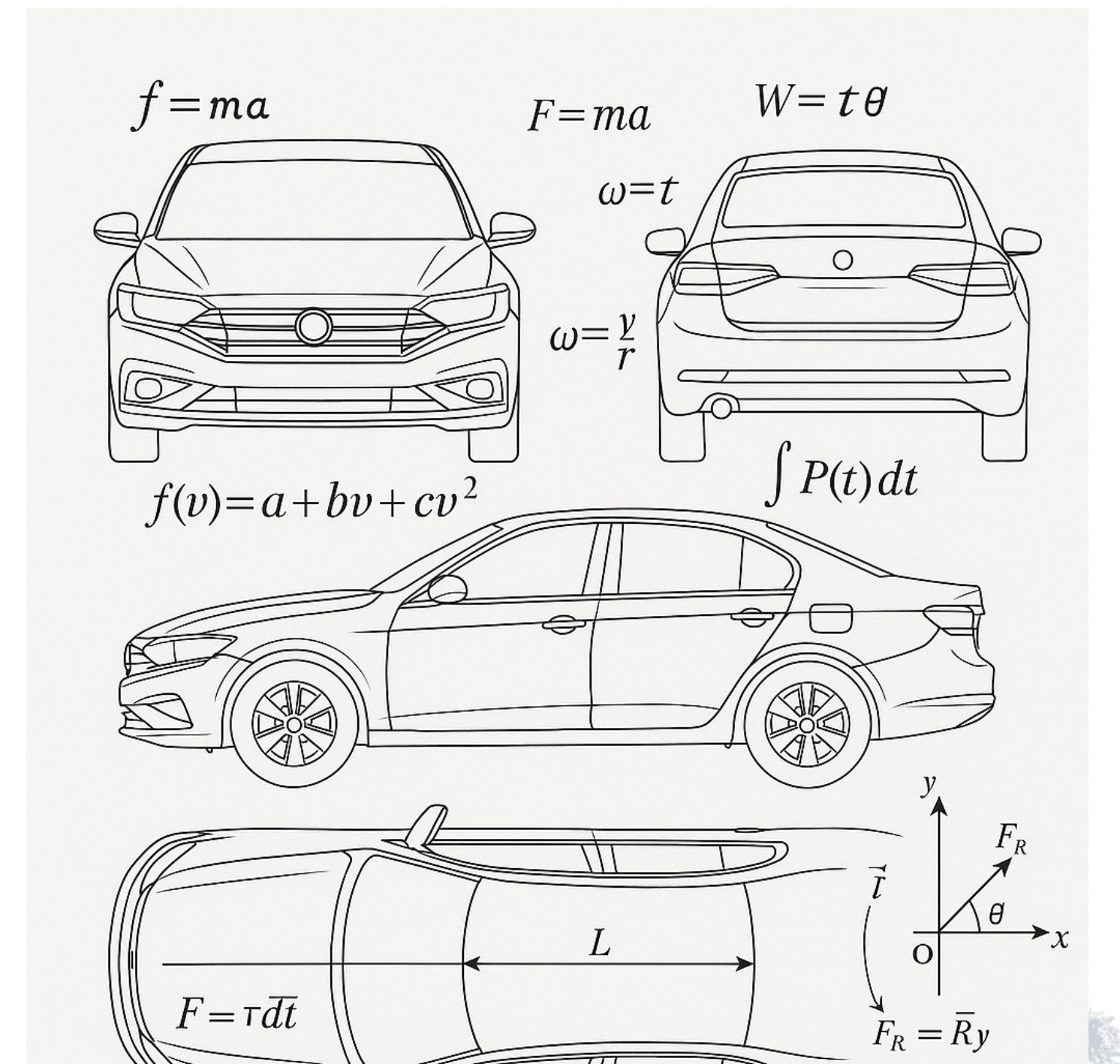
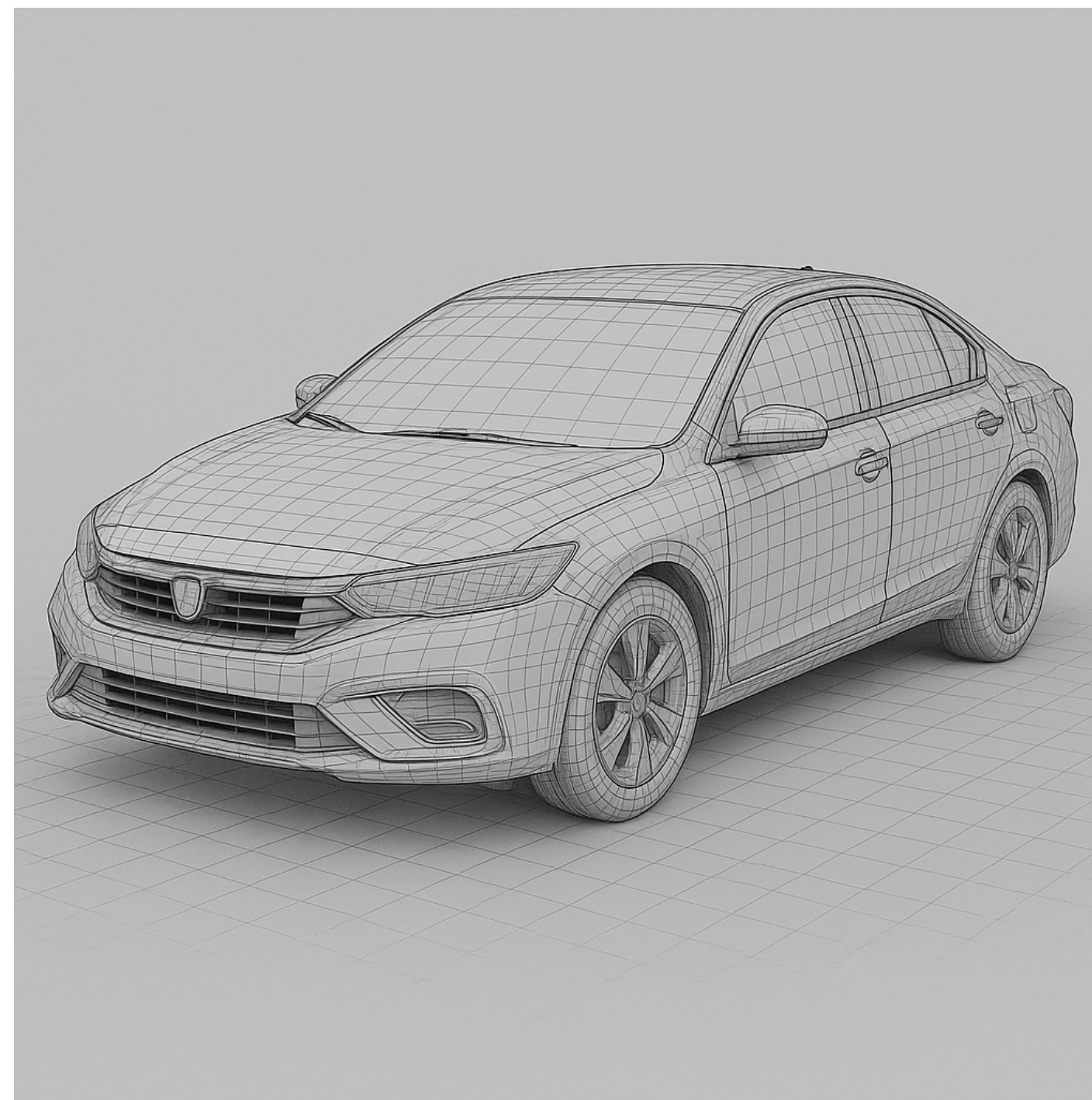
A set of architectural **design decisions**

Focus on a set of **views and viewpoints**

Developed according to **architectural styles**

Architecture View and Viewpoints

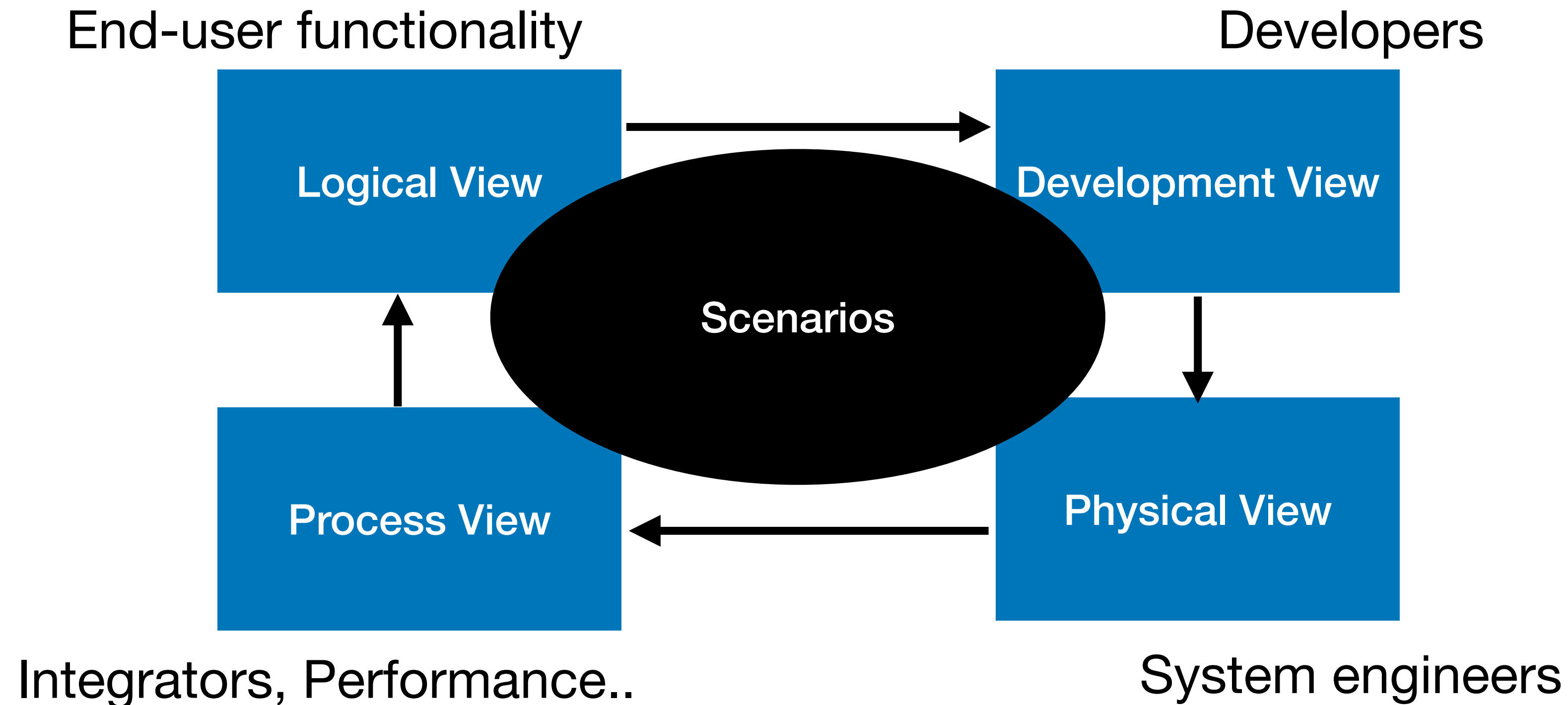
- Viewpoint is where you see from
- View is what you see - Viewpoint governs the view!!



Architectural Views: How Many

- A view represents a collection of architectural elements and relations among them
- Two fundamental views – Structural and Behavioural
 - **Structural:** The components in the system and their arrangement
 - **Behavioural:** Interaction between the components
- Many models have been proposed – eg, 4+1 view model

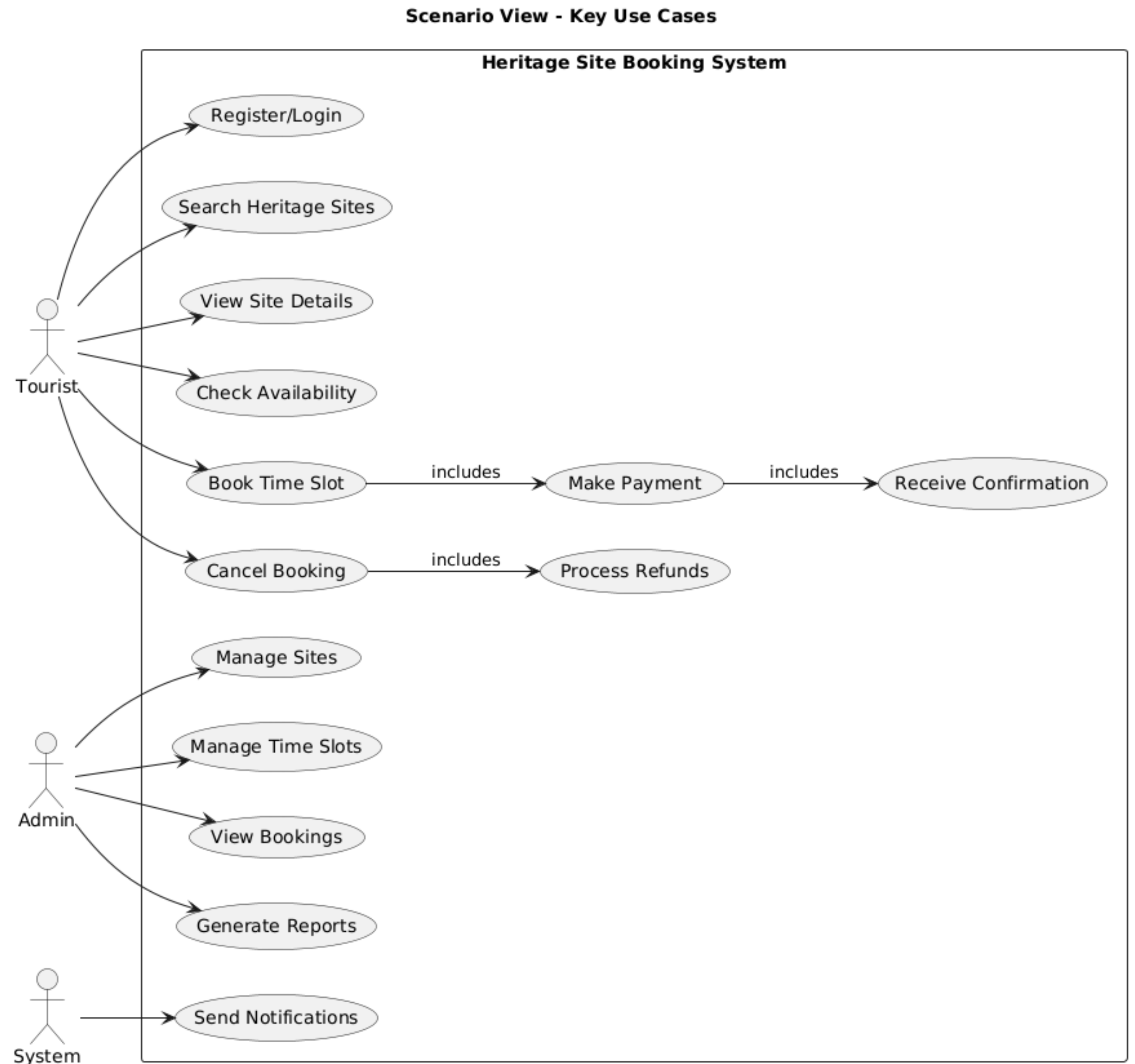
4+1 View Model for Software Architecture



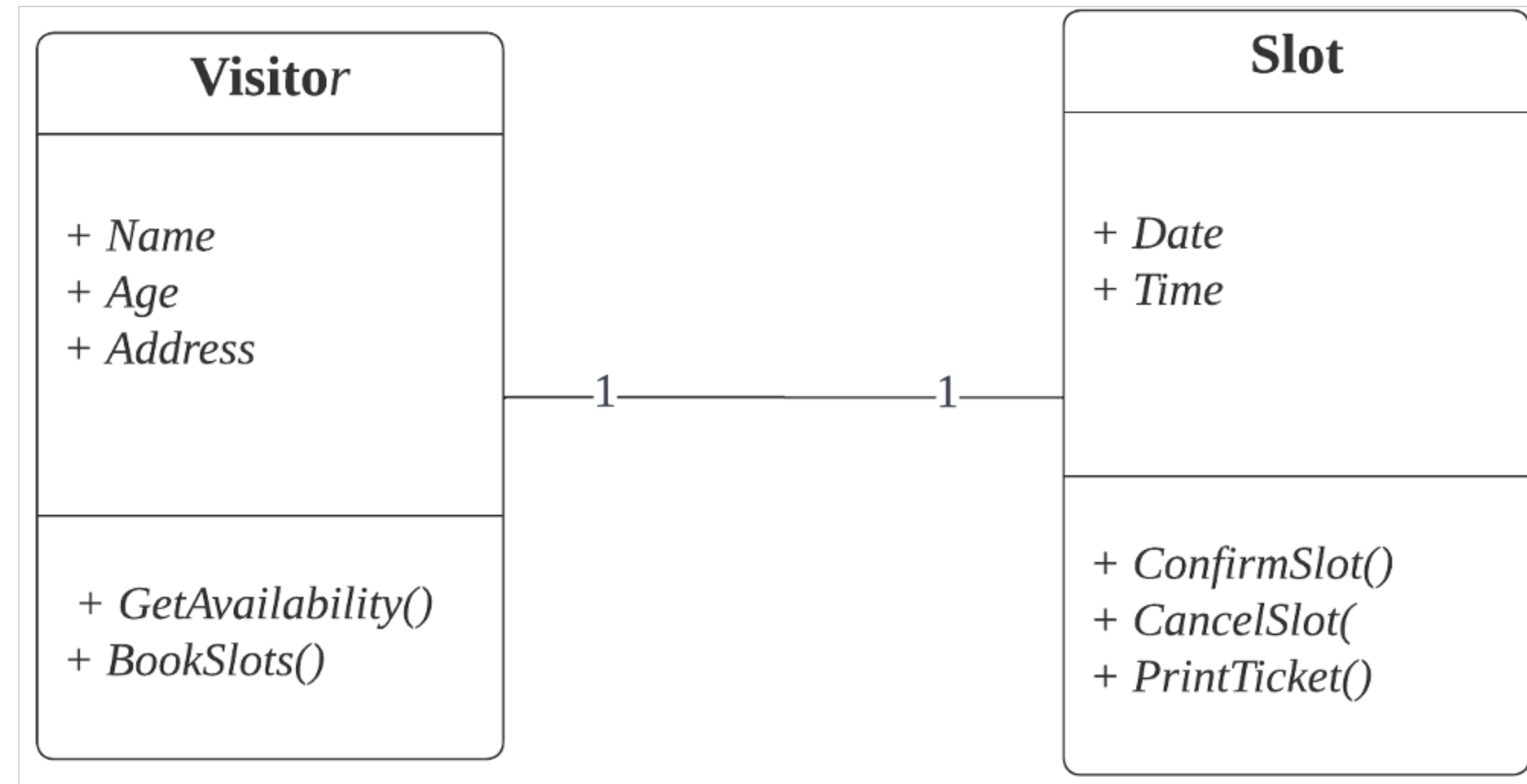
The “4+1” view model is rather “generic”: other notations and tools can be used, other design methods can be used, especially for the logical and process decompositions, but we have indicated the ones we have used with success. - Philippe Kruchten, Architectural Blueprints, The “4+1” View Model of Software Architecture

Scenarios

- Represent the different use cases
- **Stakeholder:** End-user, developer
- **Concerns:** Understandability
- **Diagram:** Use case diagrams



Logical View



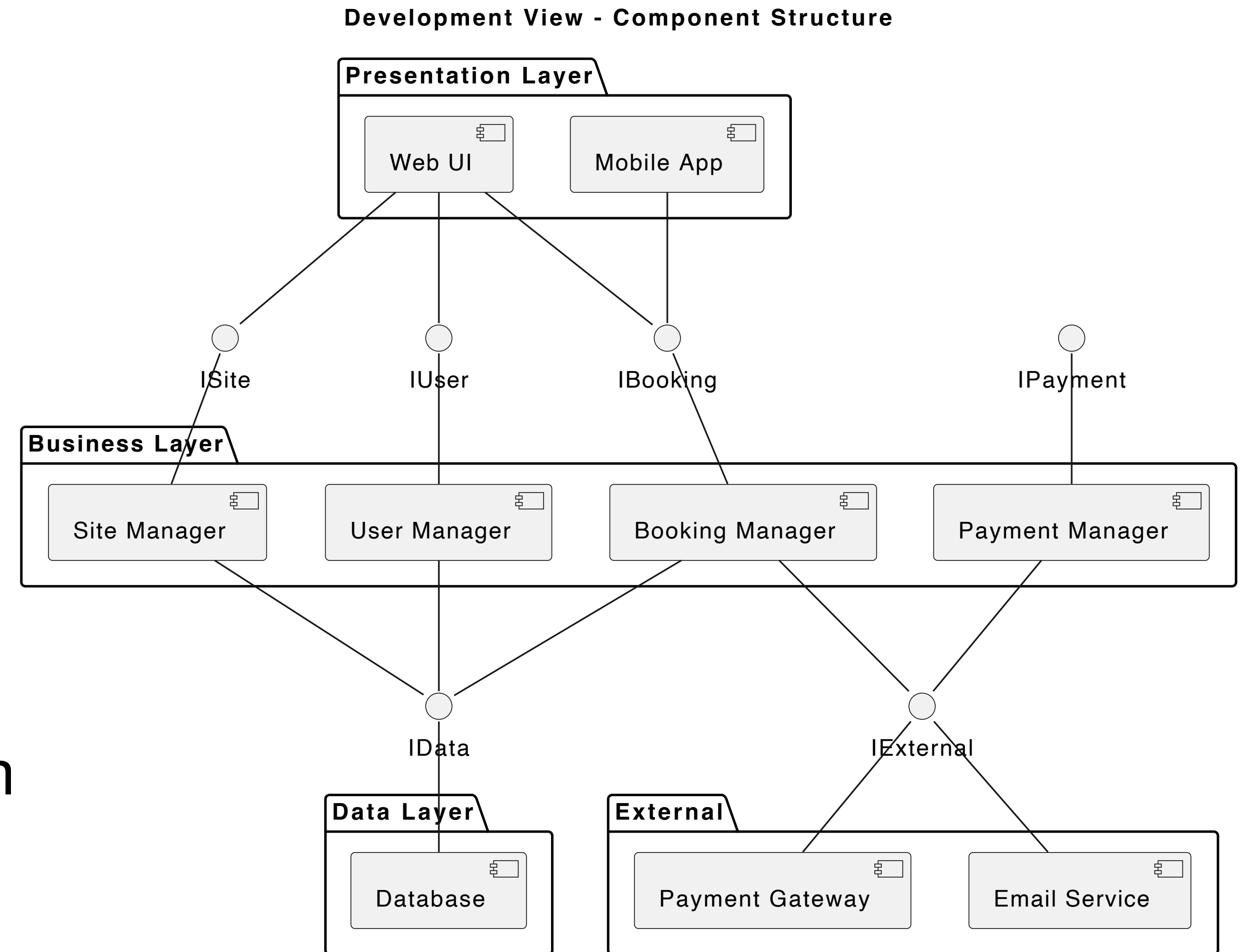
System decomposed into a set of abstractions (objects or object classes)

- **Stakeholders:** Developer
- **Concerns:** Functionality

- **Diagrams:** UML Class diagrams, logical connection diagrams

Development View

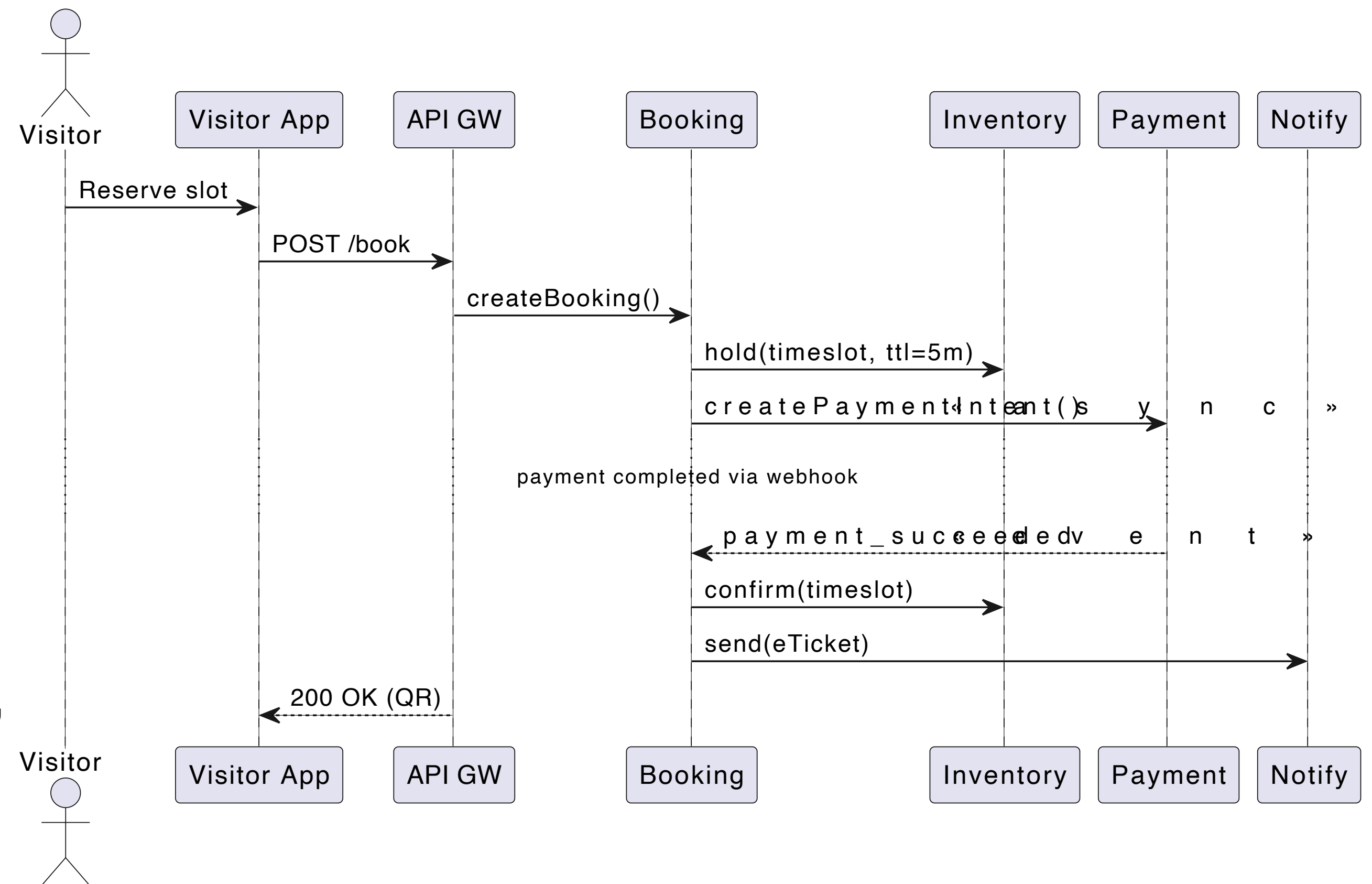
- Organization of the System into subsystems/modules
- **Stakeholders:** Developer, manager
- **Concerns:** Organization, reuse, portability
- **Diagram:** UML Component diagram



Process View

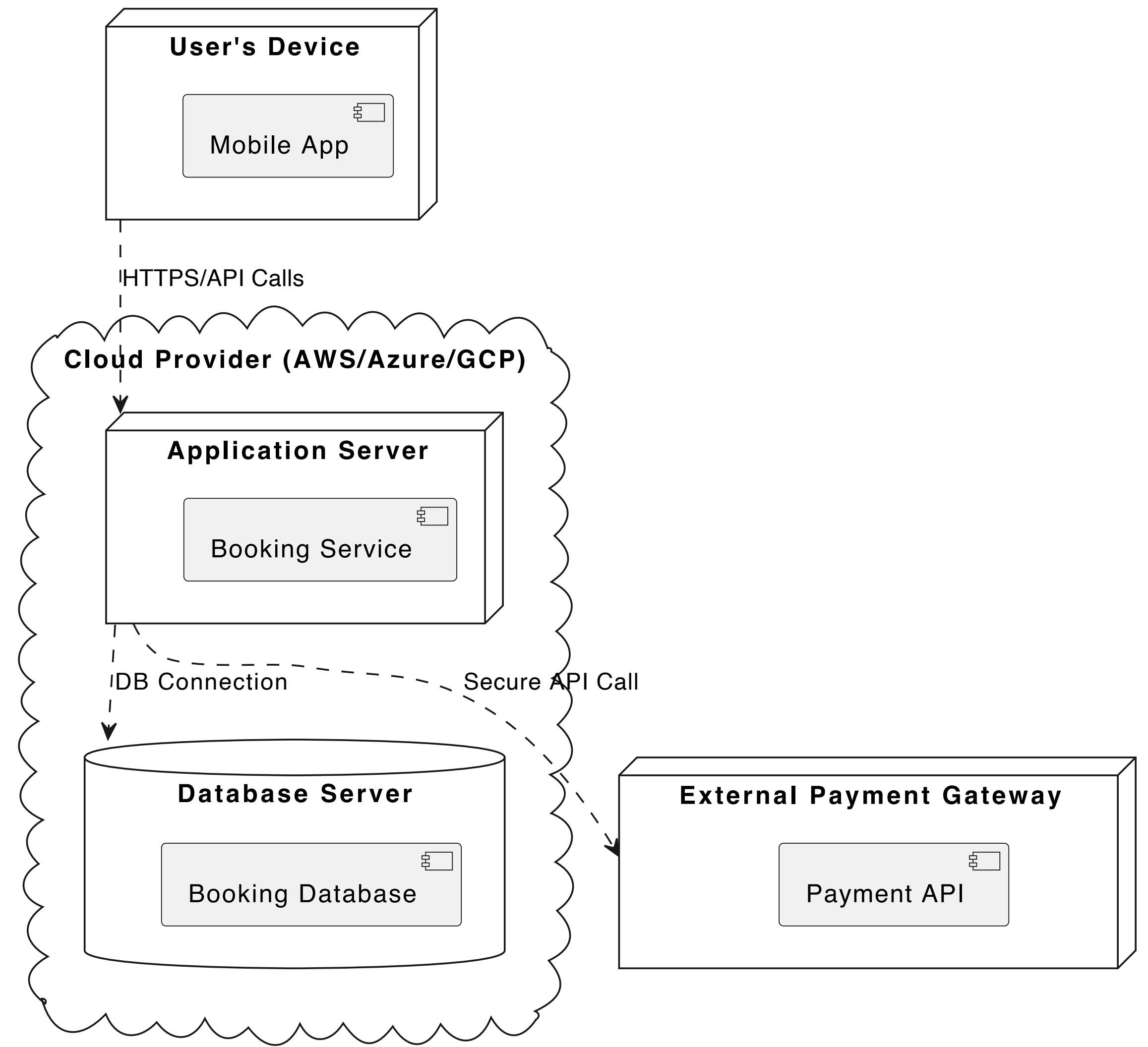
- Model dynamic aspects of the software
- **Stakeholders:** System designer, integrator
- **Concerns:** Performance, fault tolerance
- **Diagram:** UML Sequence diagram, Process diagram, Data flow

Process View (Runtime & Concurrency)



Physical View

- Mapping of SW elements into deployment nodes
- **Stakeholders:** System designer, Admin
- **Concerns:** Performance, Scalability, Availability
- **Diagram:** UML Deployment diagram, Network diagram, etc.



Before We Proceed!!

• **SCAN HERE**

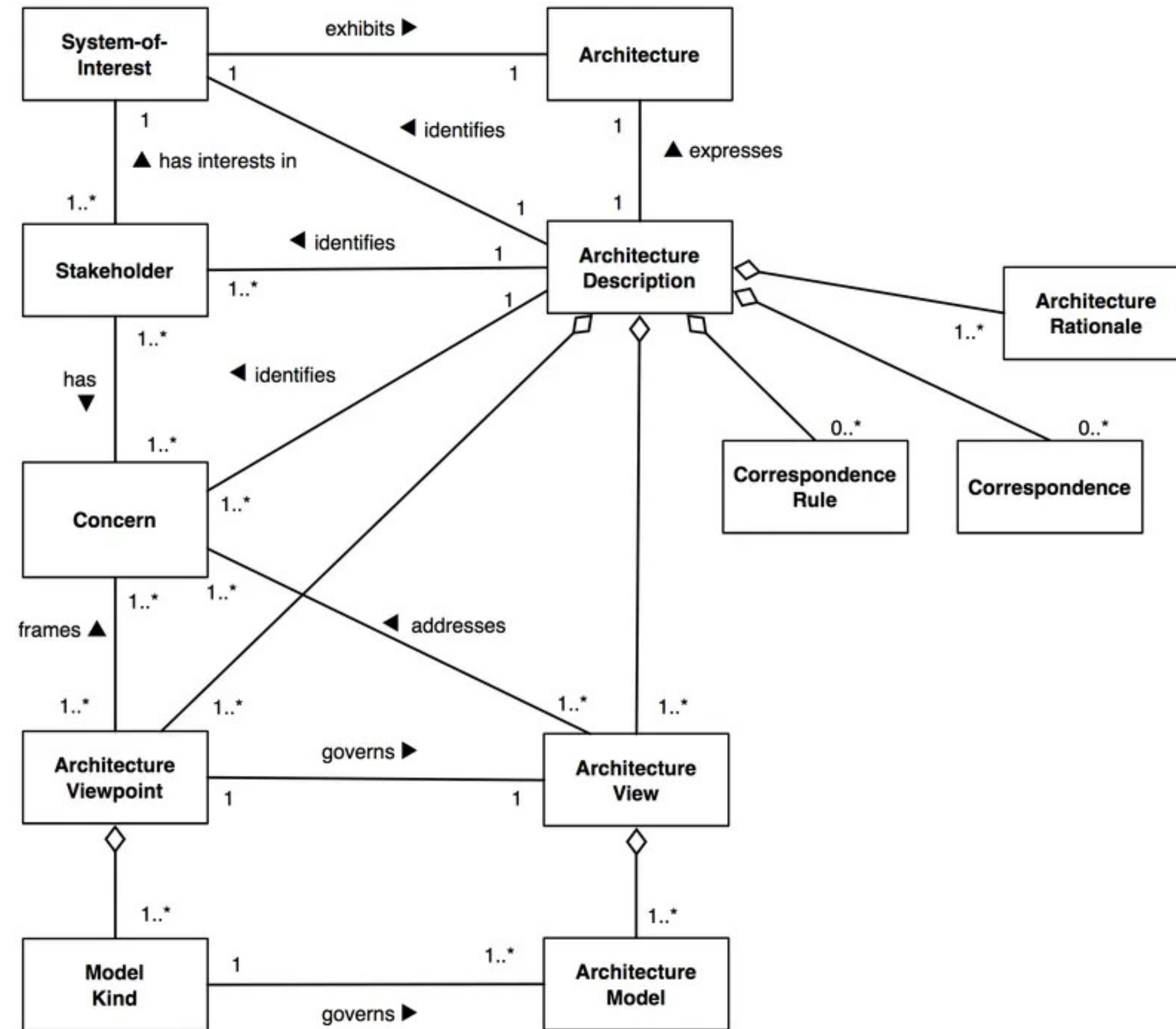


• **ANONYMOUS**

• **MID FEEDBACK** •

• **CS6.401 SE**

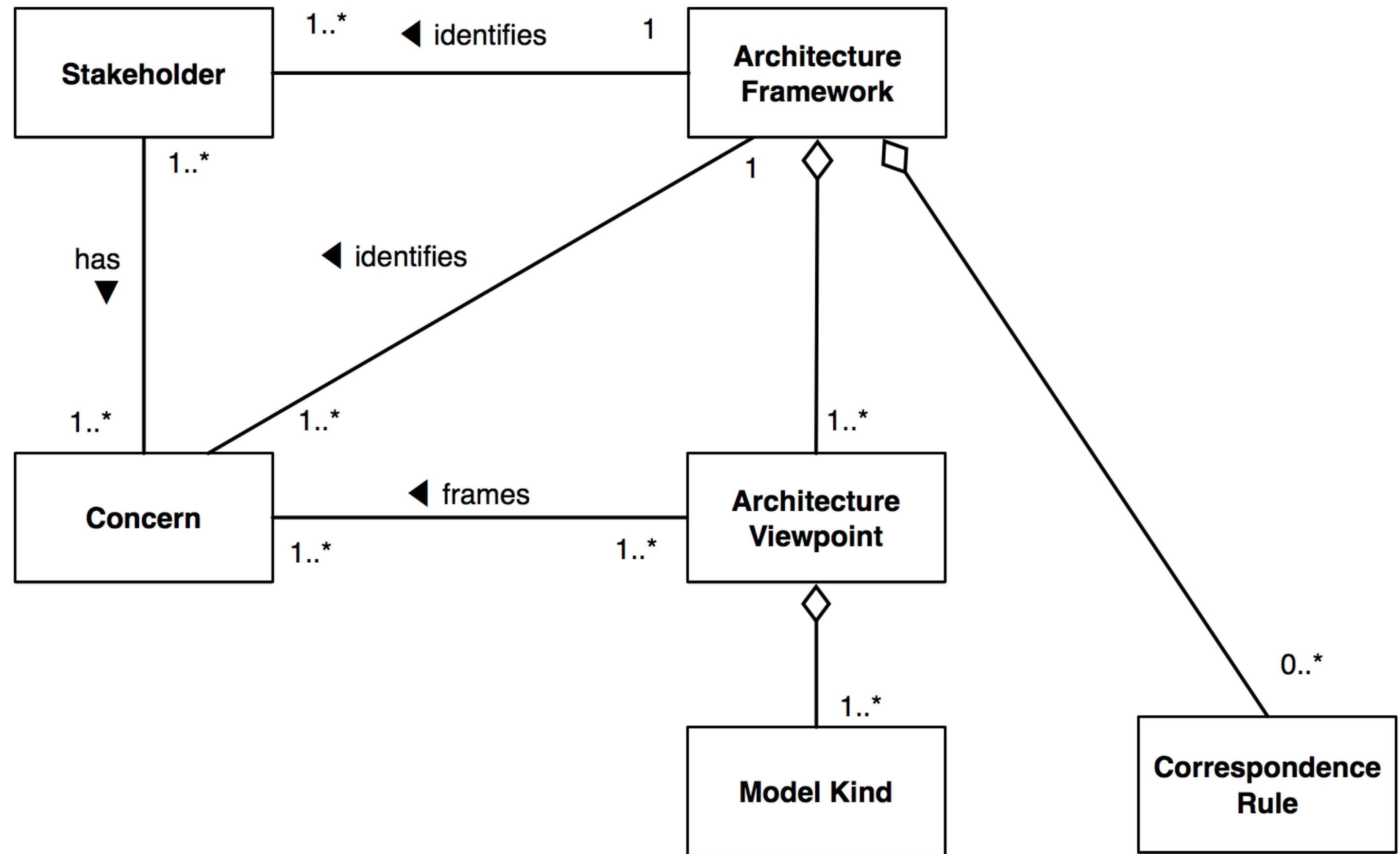
Architecture Description



Architectural Framework

An architecture framework is a coordinated set of viewpoints, conventions, principles, and practices for architecture description within a specific domain of application or community of stakeholders

- Stakeholders
- Concerns
- Viewpoints



Architecture Framework in Action

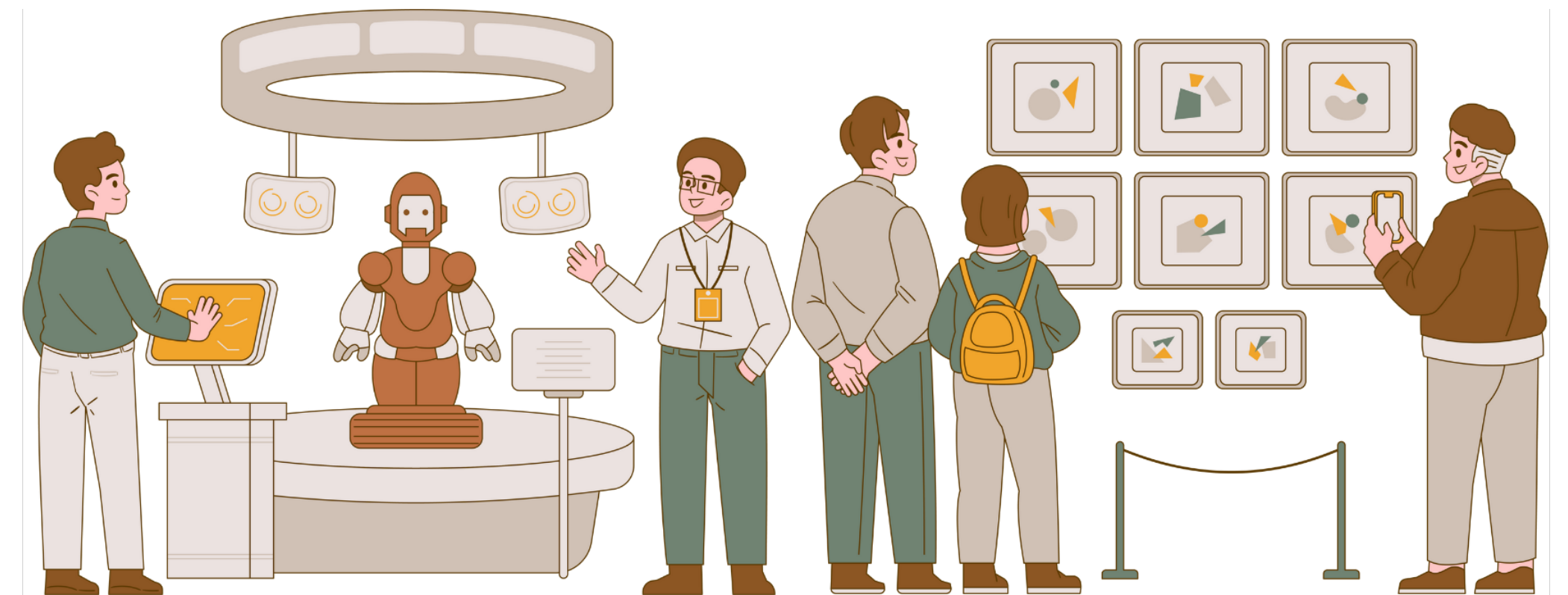
The Case Study of an Exhibition Night

Key Observations

Around 35K visitors per year

Late hours are more crowded than early hours

Weather affects user's preferences



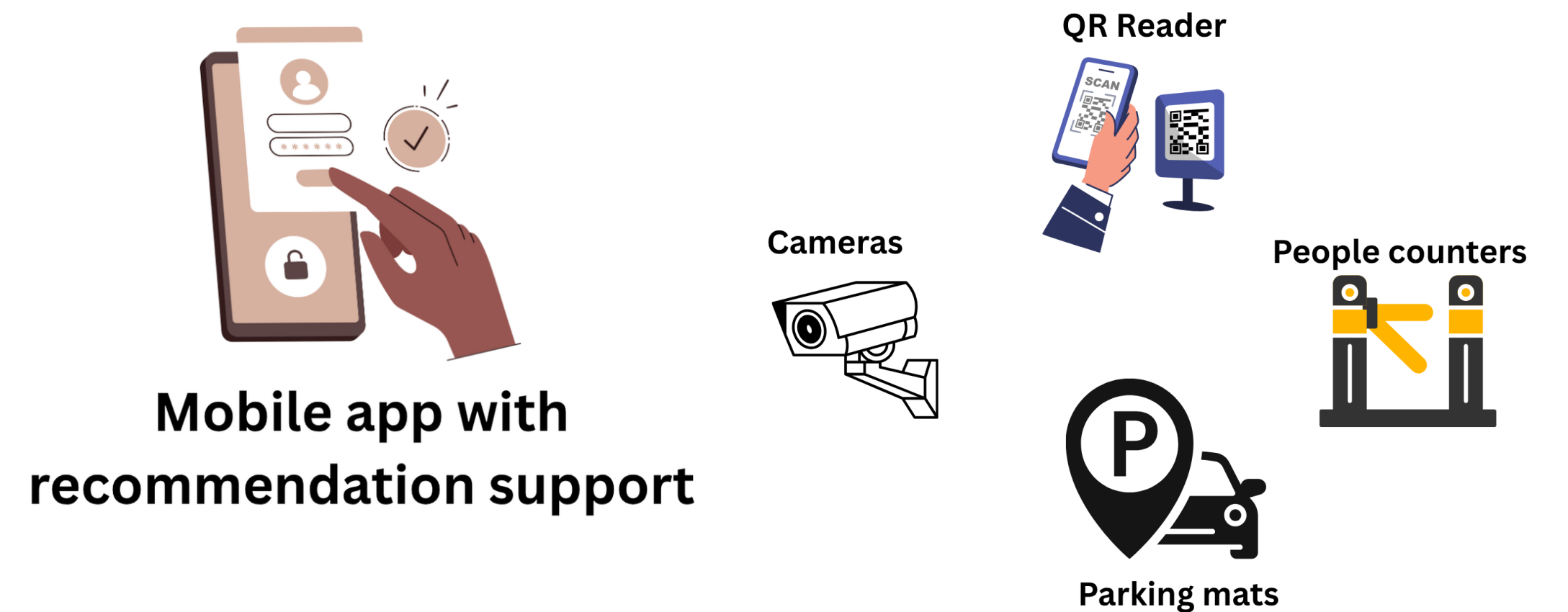
Goal

A solution for crowd management

Automated venue and parking lot management

Constraints

Limited power and service accuracy constraints



Functional and Non-Functional Requirements

Functional Requirements

FR1: The app shall allow users to register to the NdR event

FR2: The app shall allow users to provide their preferred events

....

Extra(Non)-functional Requirements

NFR1: The app must provide a response time of < 1 seconds

NFR2: The app should be able to support 1000 users/second while maintaining optimal performance

Stakeholders



Software Developers



IoT Engineers



Urban planners



Software architect(s)

Many more....

Concerns

Language choice, the
modules,
Interactions, ...



Software Developers

Memory,
Battery,
platform ...



IoT Engineers

Space constraints,
Location for sensors,



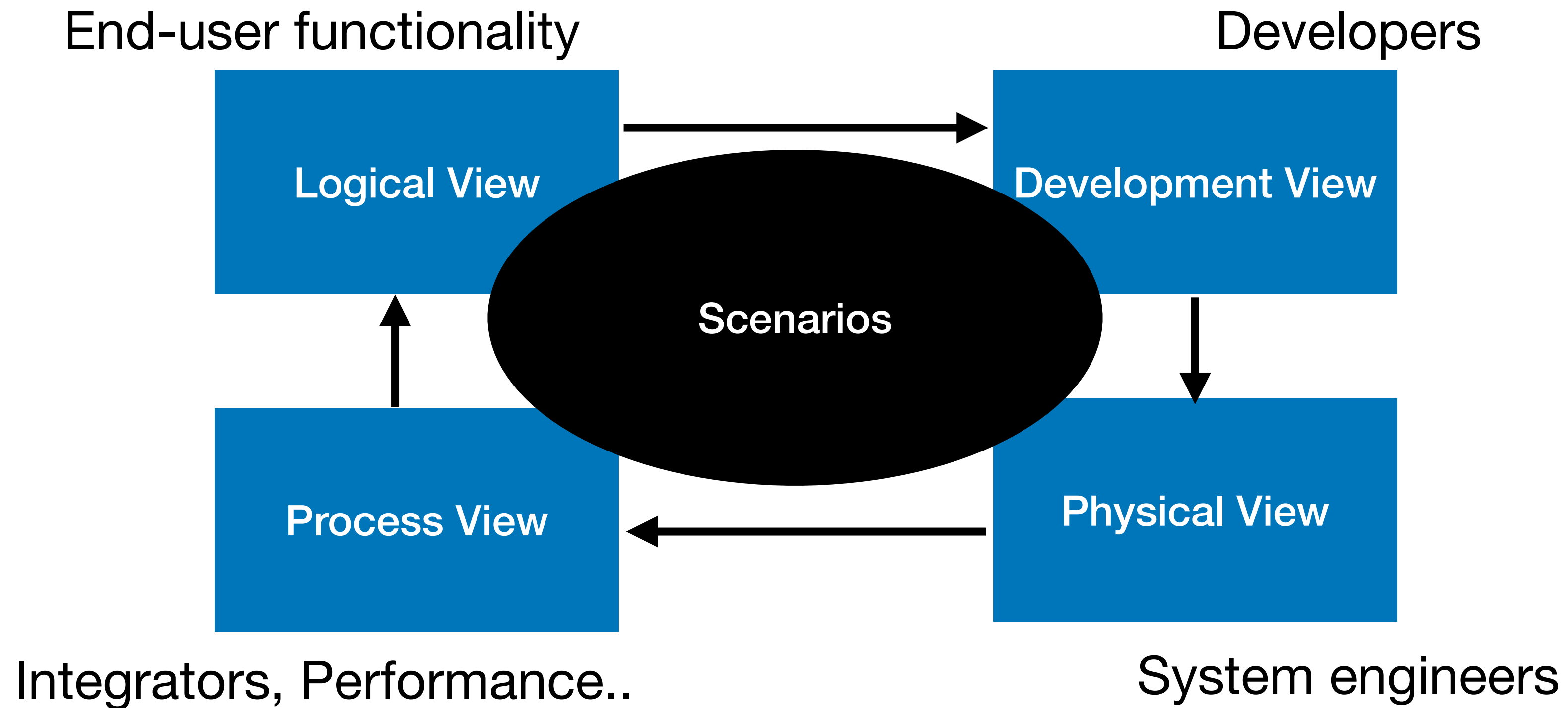
Urban planners

System performance,
Integration,
Management,...



Software architect(s)

Can we create some models for each view?



What could be some scenarios?

Book venue

Check
availability

Check weather

Get
recommendations

Get confirmation

Book parking lots

Breaking down into multiple subsystems

IoT System

Booking System

Weather System

Analytics System

Going Deeper

IoT System

Sensor
Controller

IoT Middleware

Database

Visualisation
Engine

Going Deeper

Booking System

Web Application

Mobile Application

Database

EM Backend

Breaking down further

EM Backend

Venue Booking
Manager

Parking Lot
Booking Manager

Recommendation
Generator

Payment
Manager

And many more...Can you name?

You can further break this down..think of classes!

How to Model?

- One can use UML to create the different models
 - Different UML diagrams: use case, component, sequence, deployment....
- You can also make use of **C4 model**
 - Created by Simon Brown, Independent Consultant
 - Overcome challenges of UML, more intuitive language
 - Visualize architectures in terms of context, containers, components and code
 - Developer friendly approach to architecture diagramming



Is Software Architecture Just some Box and Arrow?

*“Aside from providing clear and precise documentation, the primary purpose of specifications is to provide **automated analysis** of the document and to expose various kinds of problems that would otherwise go undetected” - Perry and Wolf, 1992*

*“An architectural system representation is often essential to the **analysis and description** of the high-level properties of a complex system” - Garlan and Shaw, 1994*

Architecture Description Language

UML is a general-purpose language

An Architecture Description Language (ADL) or an Architecture Definition Language is a:

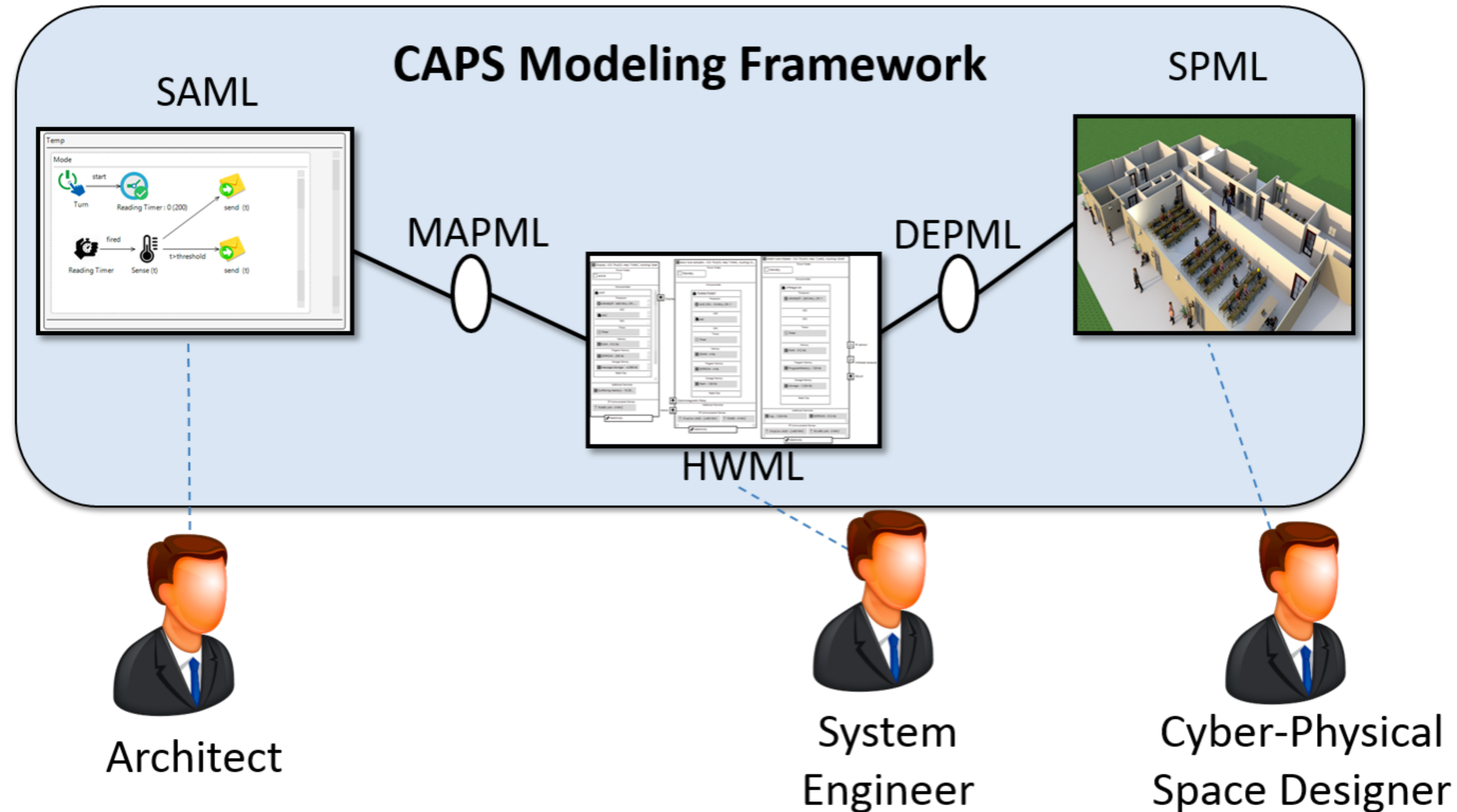
- Formal specification language
- For describing the structure and behavior of a software system

Some popular ADLs: Darwin, ACME, AADL,...

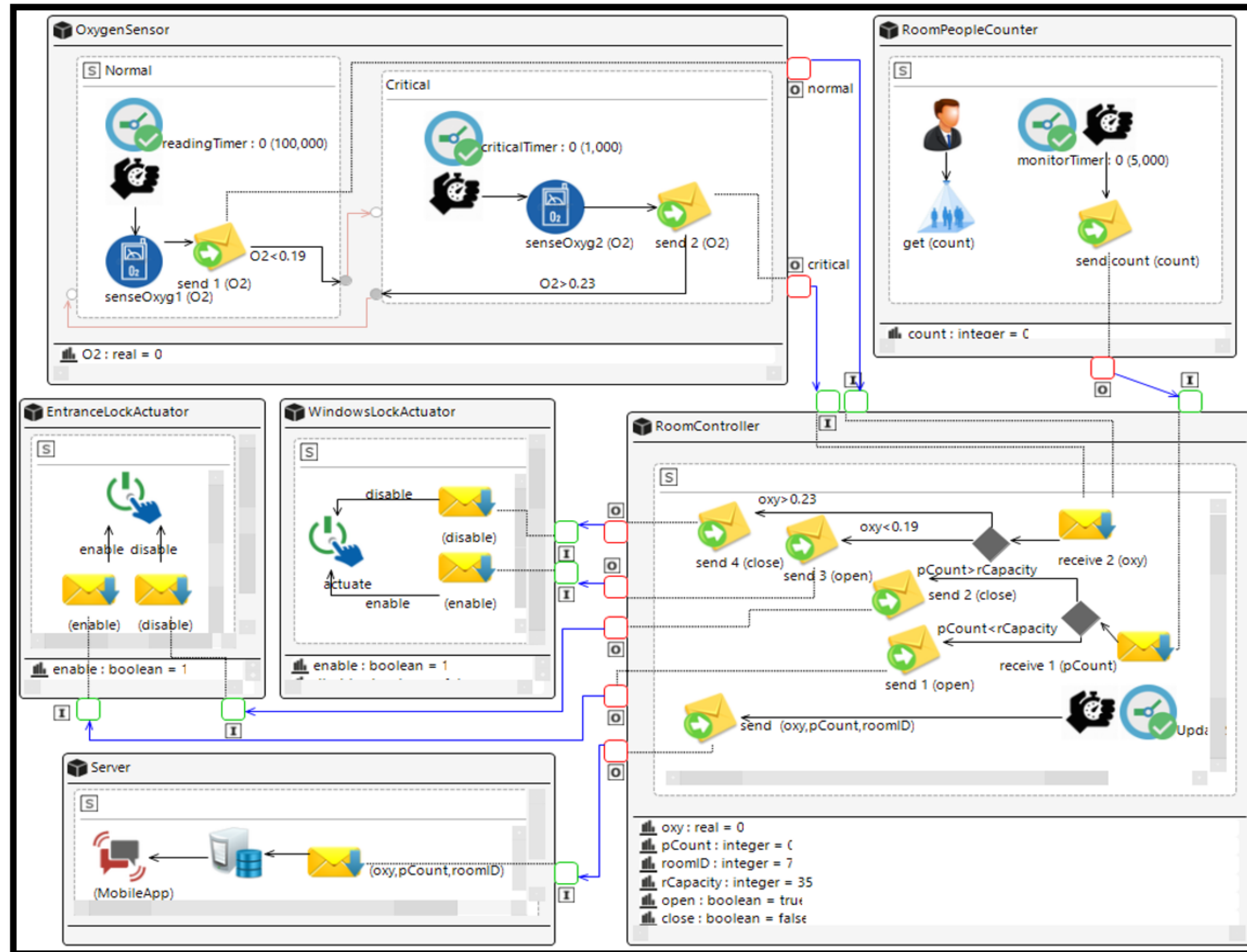
CAPS provides a way to perform an analysis of IoT Architecture before deployment

How to Model? - Glimpse Using CAPS

Multi-view Modeling Framework - IEEE 42010



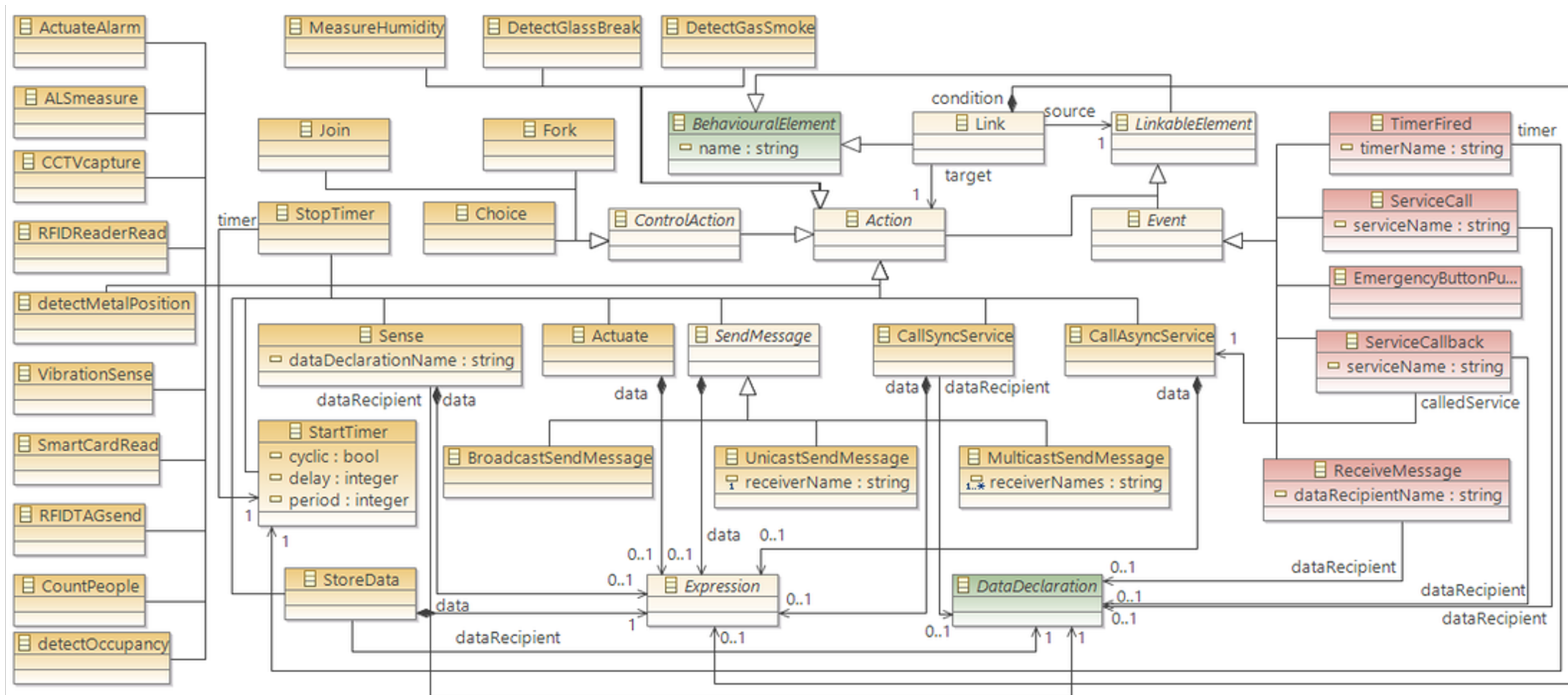
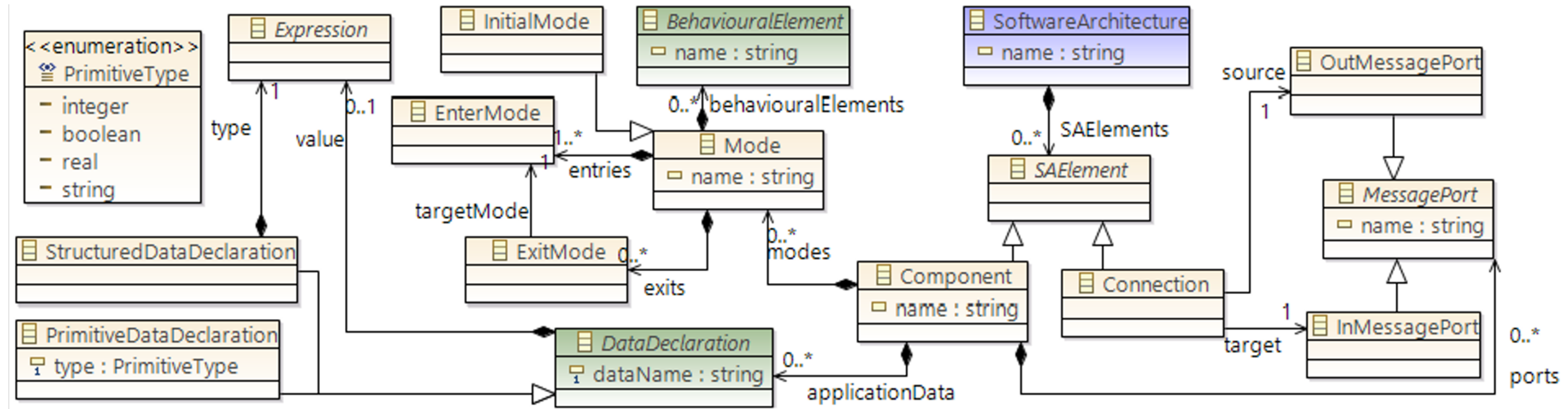
Software Architecture Modelling Language



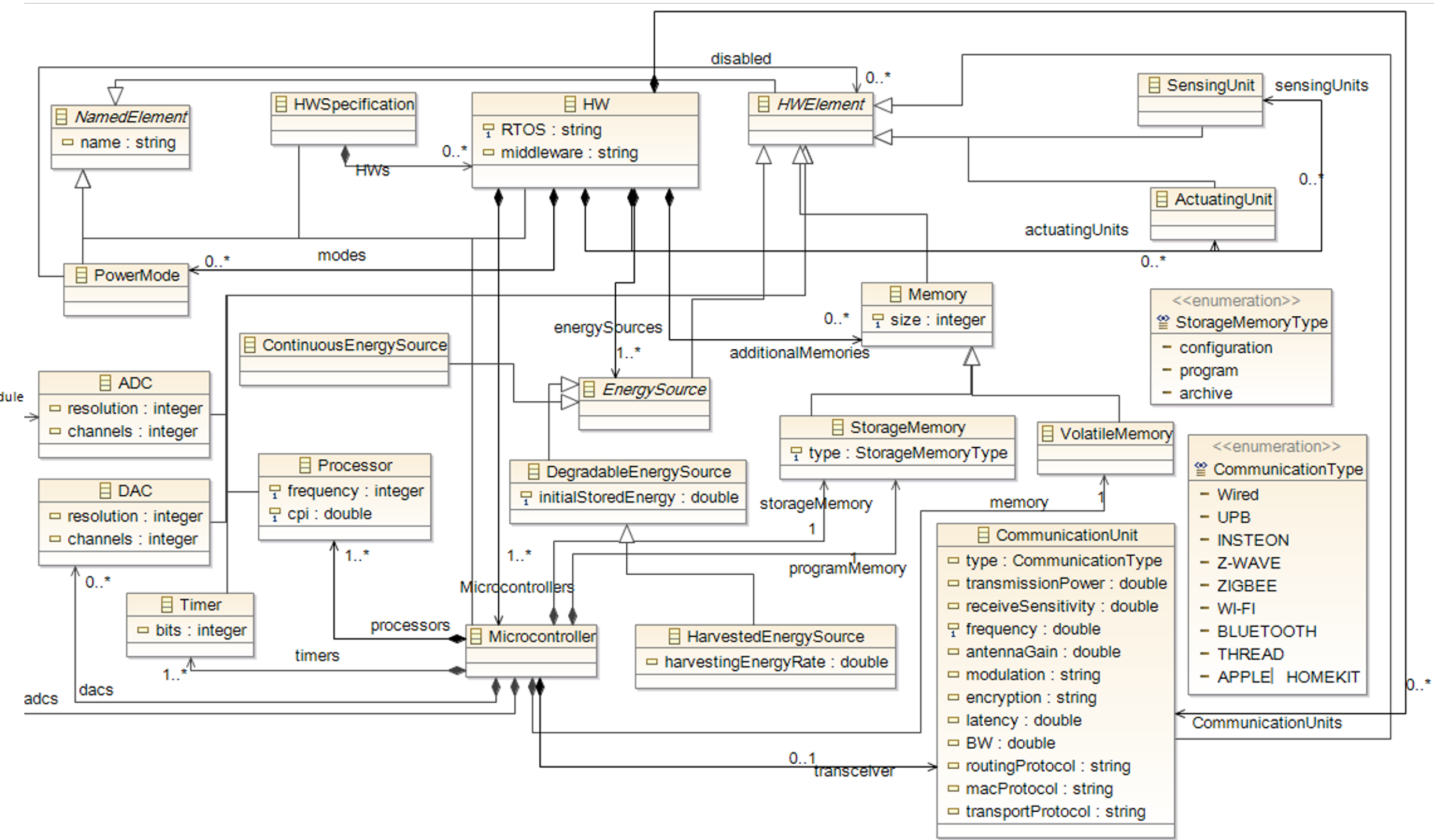
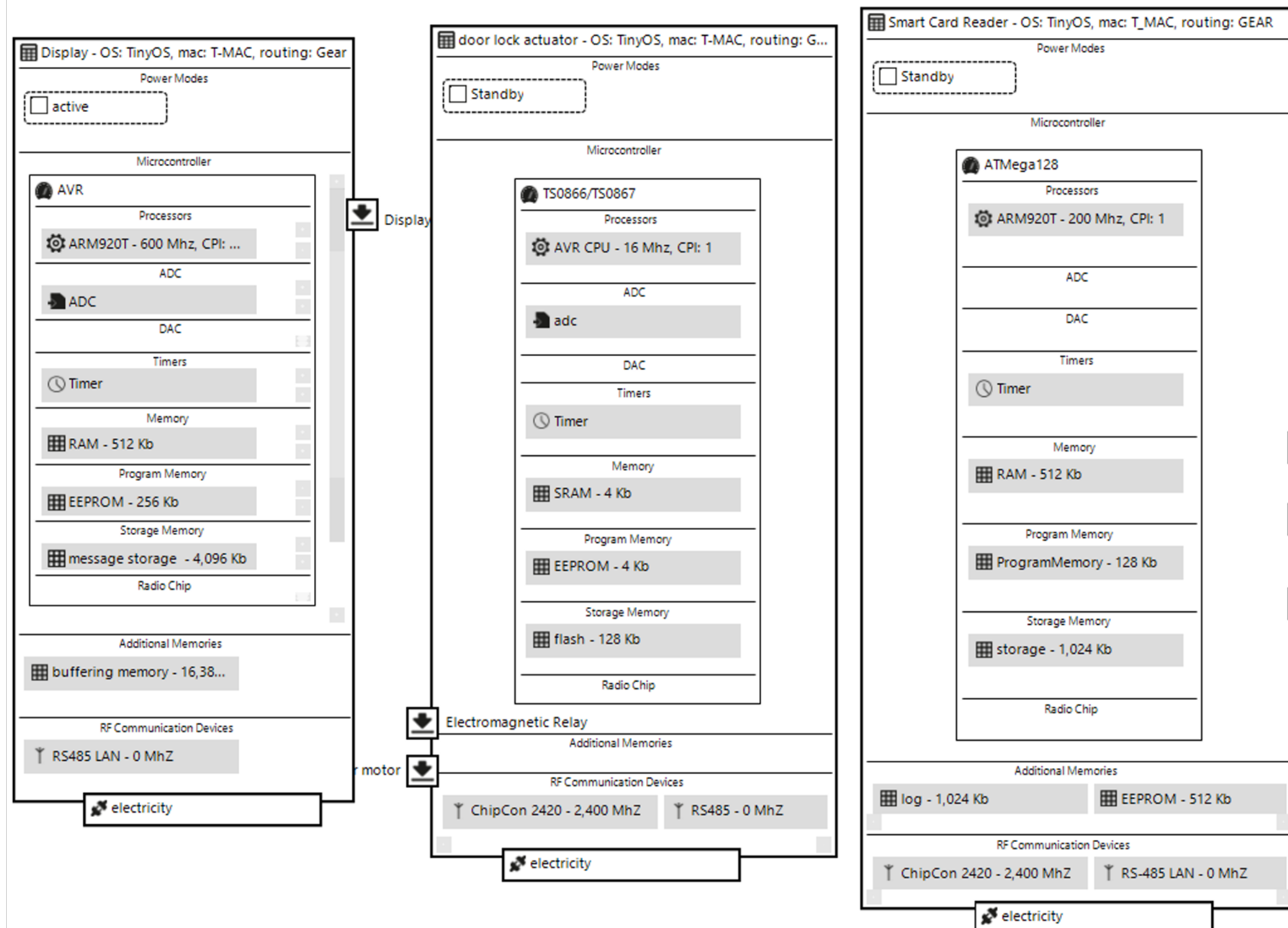
Three aspects covered

1. Structural
2. Functional
3. Behavioural

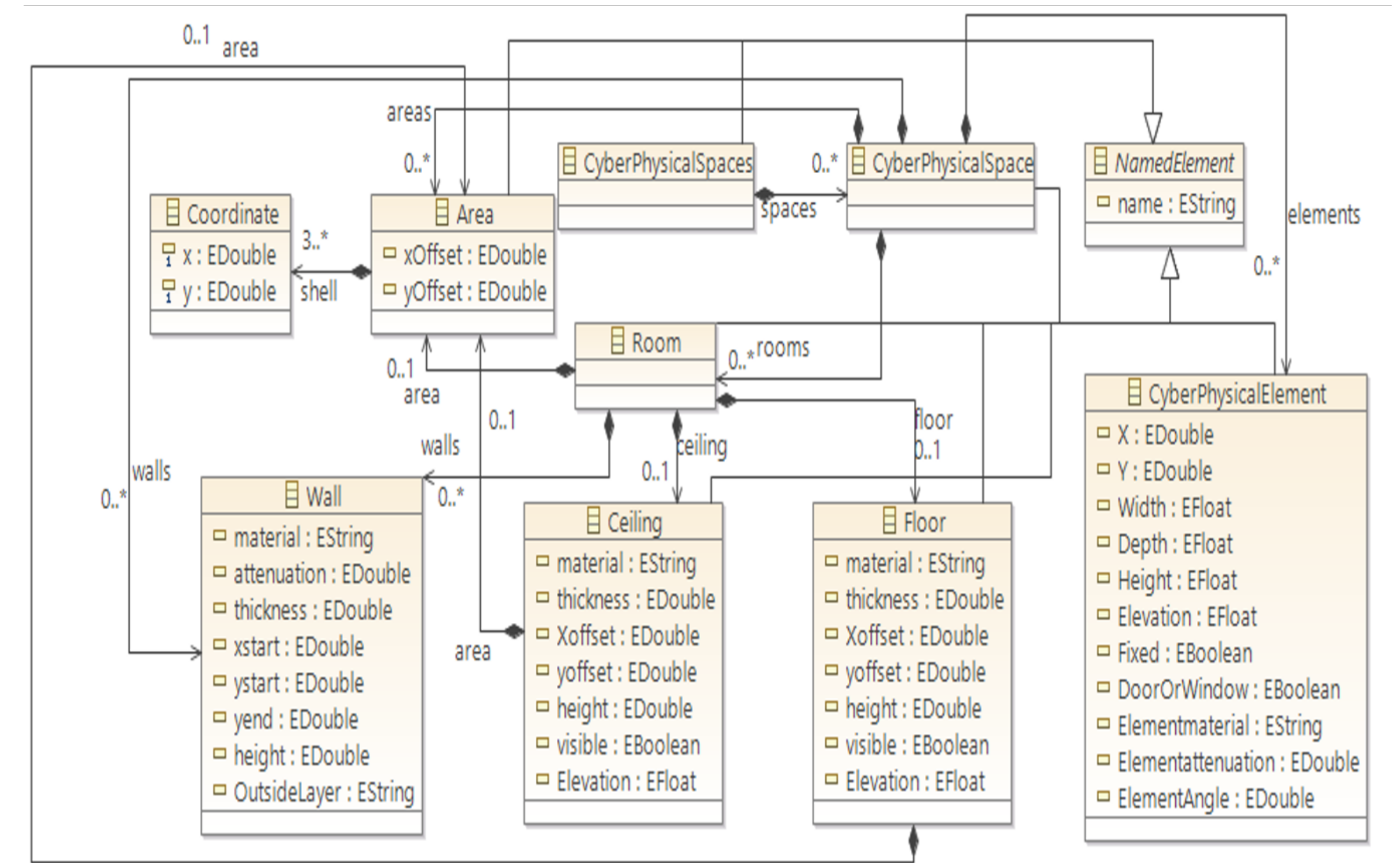
The MetaModel Behind the Scenes



The Hardware Modeling Language



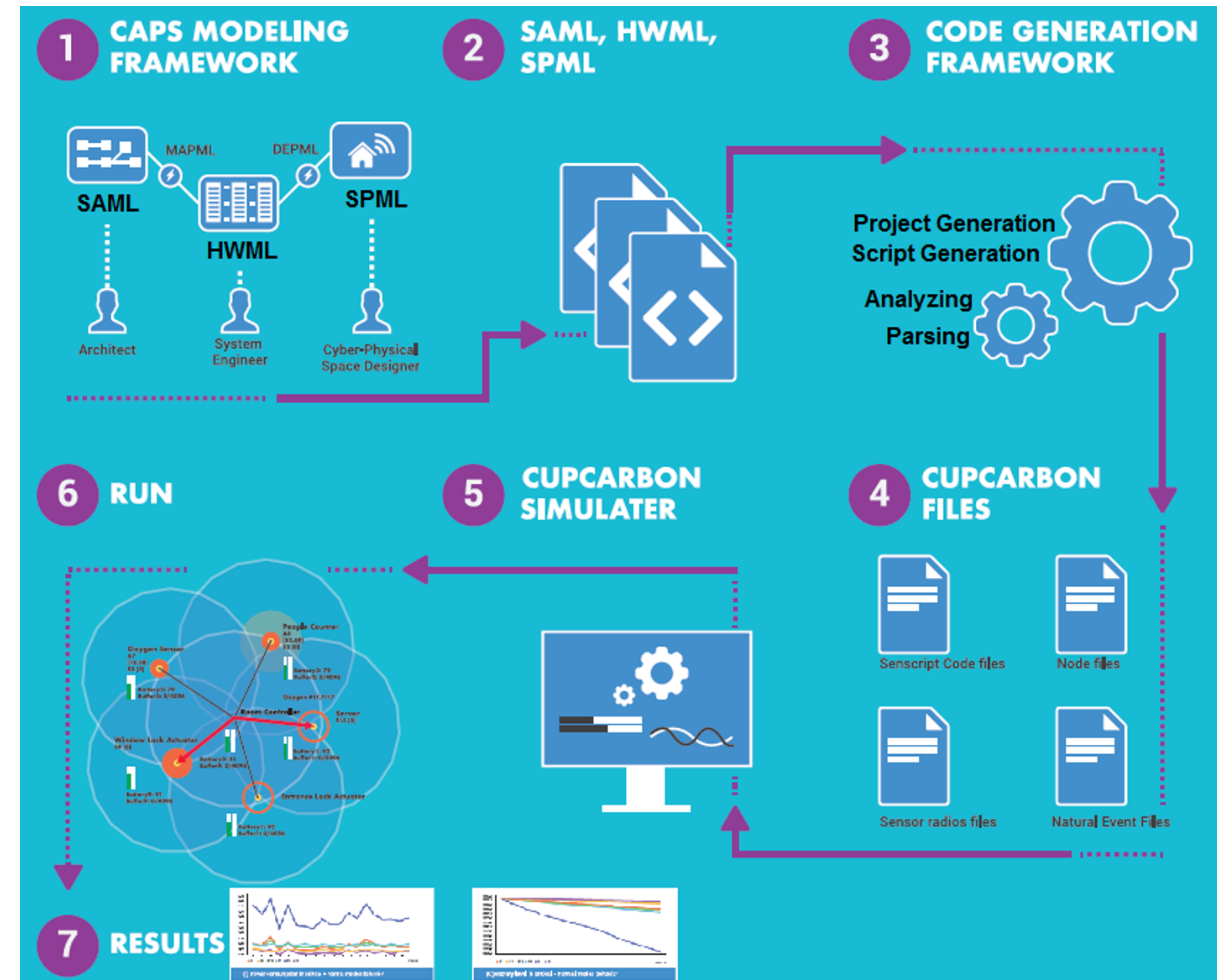
The Space Modeling Language



The 3D model done in SweetHome 3D will be converted to .xml model

Glimpse into Simulation from CAPS world

- Generated models are converted to simulation files
- Data for simulation can be from real-world
- CupCarbon is a simulator for smart city applications
- The model is also directly converted to Arduino code
- Moving from architecture to code!!



Software Architecture

The Software Architecture is the **earliest model** of the **whole software system** created throughout the software lifecycle

A set of **components and connectors** communicating through an interface

A set of architectural **design decisions**

Focus on a set of **views and viewpoints**

Developed according to **architectural styles**



Thank you

Email: karthik.vaidhyanathan@iiit.ac.in

Twitter: @karthi_ishere