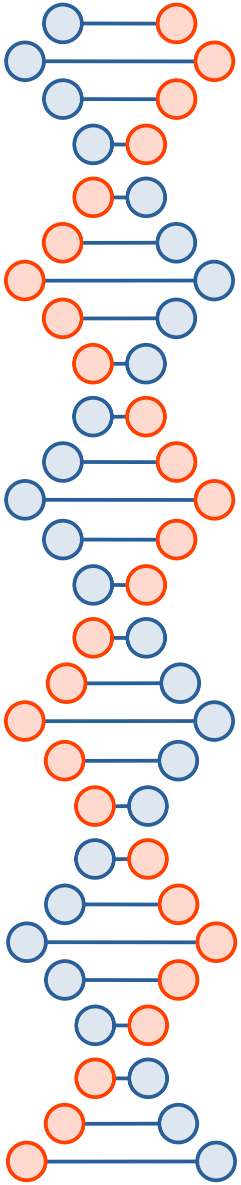
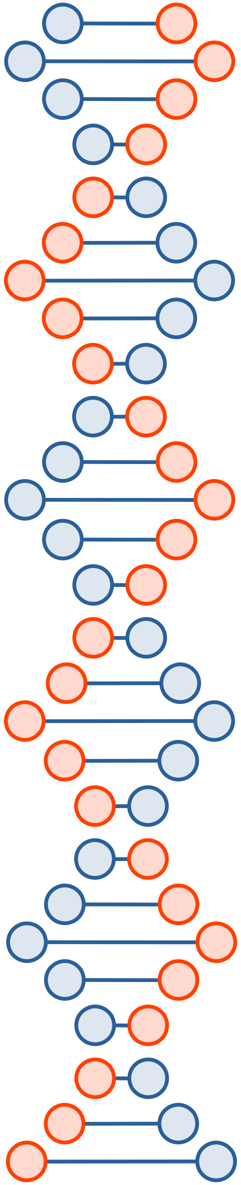


Software Architecture

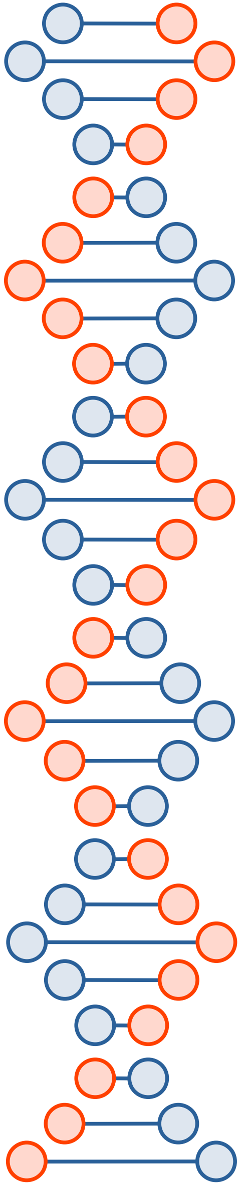
Outline

- What is Software Architecture
- Why Software Architecture in software engineering
- Software Architect
- Software Architecture View
- Software Architecture Patterns



Definition

Software architecture is, simply, the organization of a system. This organization includes all components, how they interact with each other, the environment in which they operate, and the principles used to design the software



Definition

Software architecture is a pictorial representation of the IT system that communicates the following:

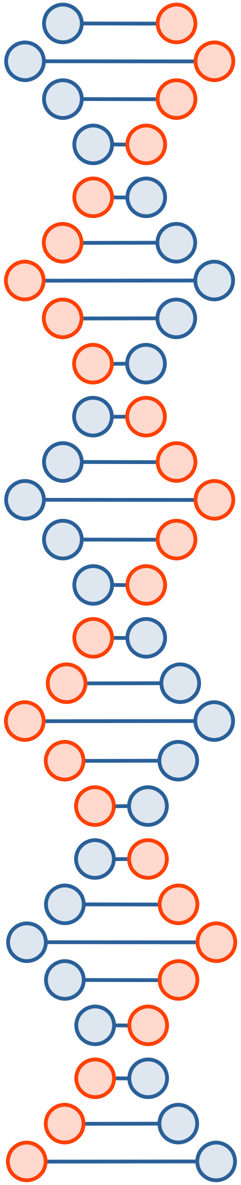
The organization of the system;

Different components vis-à-vis their functions;

How the components interact with each other.

What is Software Architecture

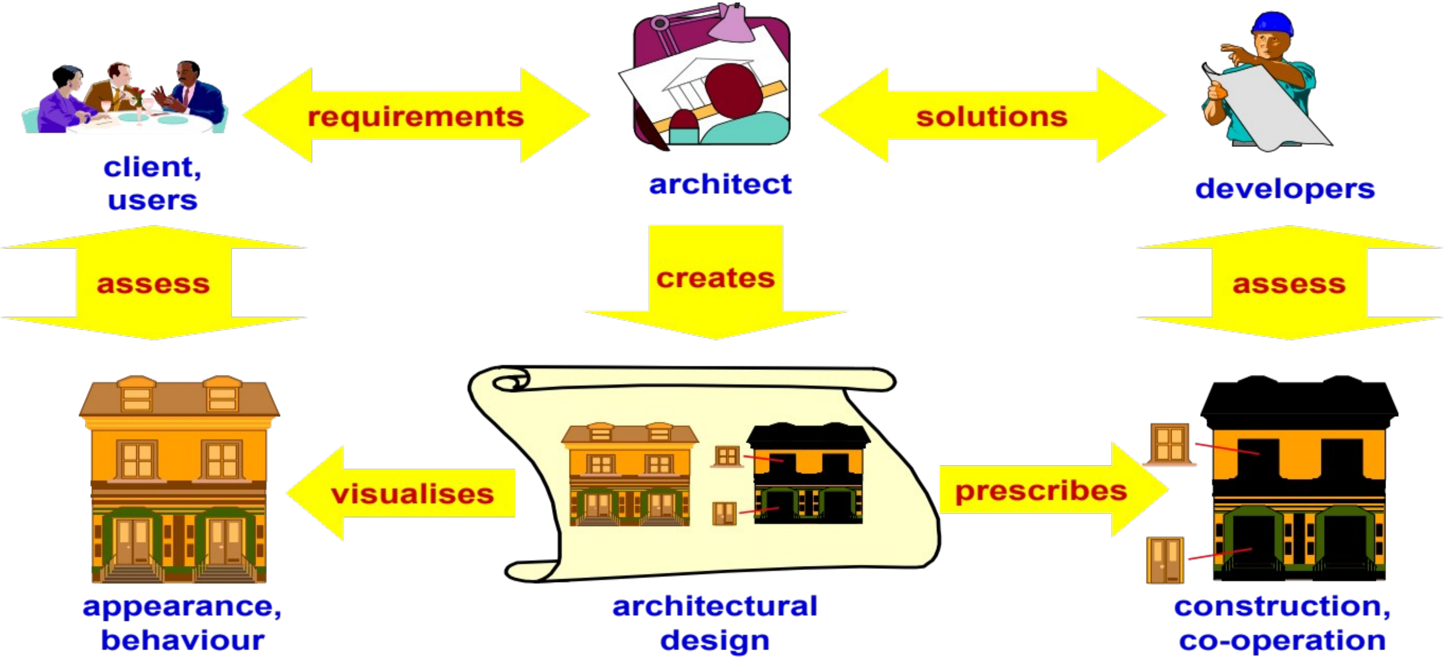
- The foundation or the fundamental design
- Has the software elements, their function and their relations
- Shows how it addresses the key business objectives such as performance, reliablility and security



Who is a Software Architect?

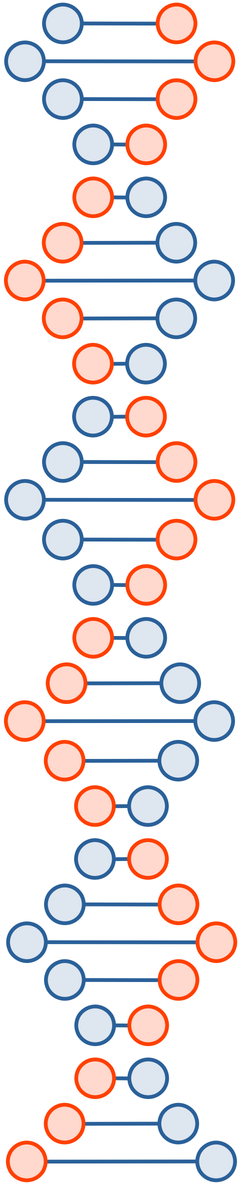
- A role that create the fundamental structures and documents them
- Analyzes the business requirements
- Identifies the key risks
- Makes fundamental choices about the software system

The Role of the Architect



Why Software Architecture in software engineering

- A software system without any architecture when built may be difficult to maintain or change
- Helps software developers understand the fundamental structure
- Possible to address the key risk
- Higher code maintainability, productivity and adaptability
- Documentation facilitates design and code reuse across projects



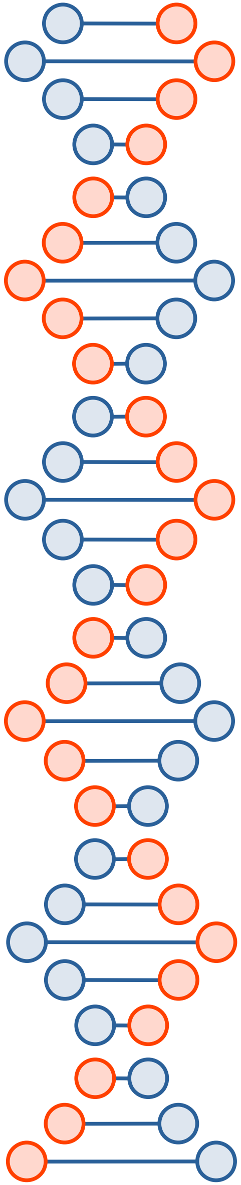
How do You Know if Your Software Architecture is Good?

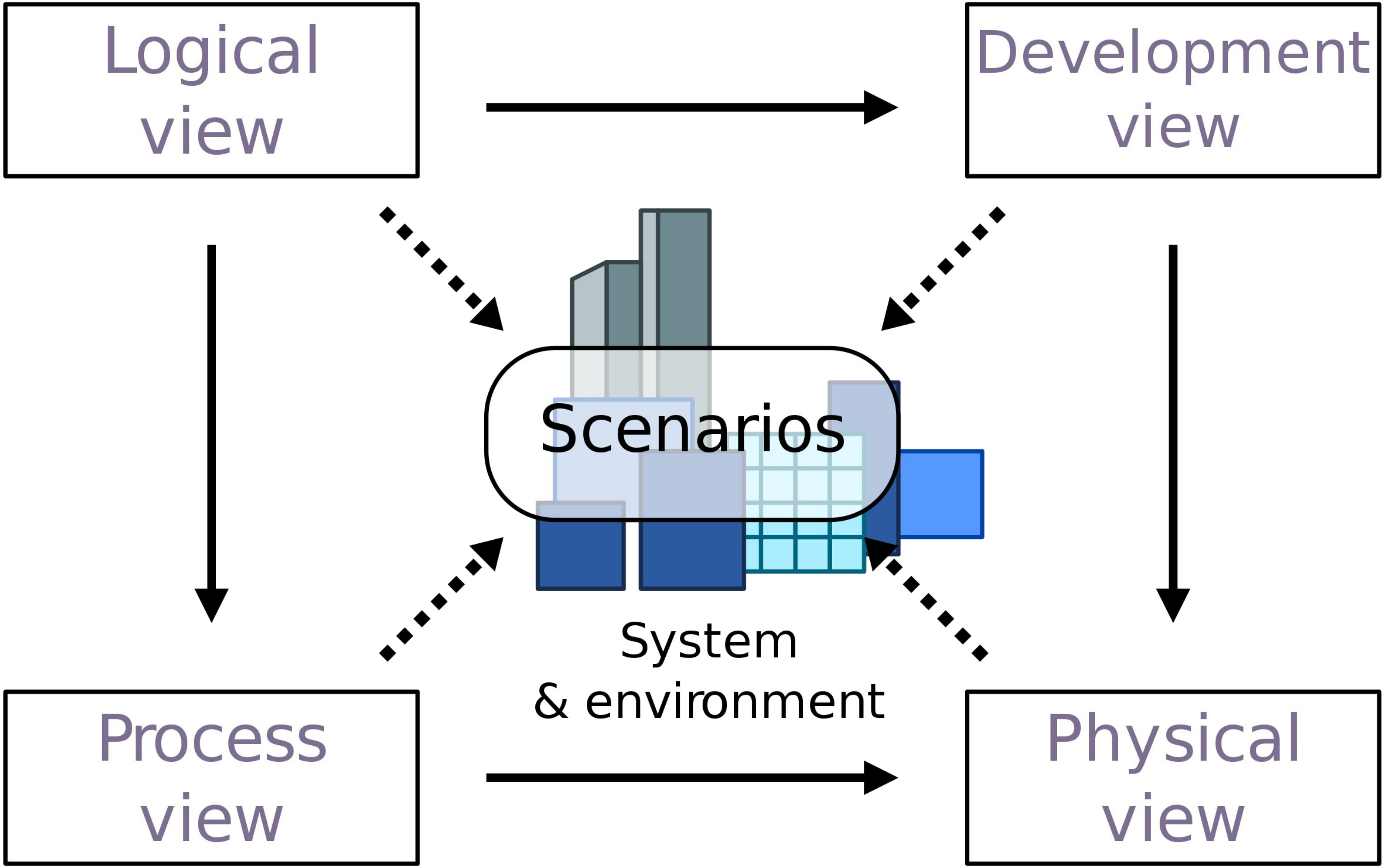
The following are indications that you have a good software architecture:

- The software is easy to maintain;
- Business stakeholders can understand it easily;
- Good software architectures are usable over the long-term;
- Such architecture patterns are flexible, adaptable, and extensible;
- It should facilitate scalability;
- The team can easily add features, moreover, the system performance doesn’t diminish due to this;
- There is no repetition of the code;
- The system can be refactored easily.

4+1 architectural view model

- 4+1 is a view model used for "**describing the architecture of software-intensive systems, based on the use of multiple, concurrent views**". The views are used to describe the system from the viewpoint of different **stakeholders**, such as end-**users**, **developers**, **system engineers**, and **project managers**. The four views of the model are **logical**, **development**, **process** and **physical view**. In addition, selected use cases or scenarios are used to illustrate the architecture serving as the 'plus one' view. Hence, the model contains 4+1 view





Logical view

Development view

Scenarios

System
& environment

Process view

Physical view

4+1 architectural view model

- **Logical view**: The logical view is concerned with the functionality that the system provides to end-users.
- UML diagrams are used to represent the logical view, and include **class diagrams,** and **state diagrams**.

4+1 architectural view model

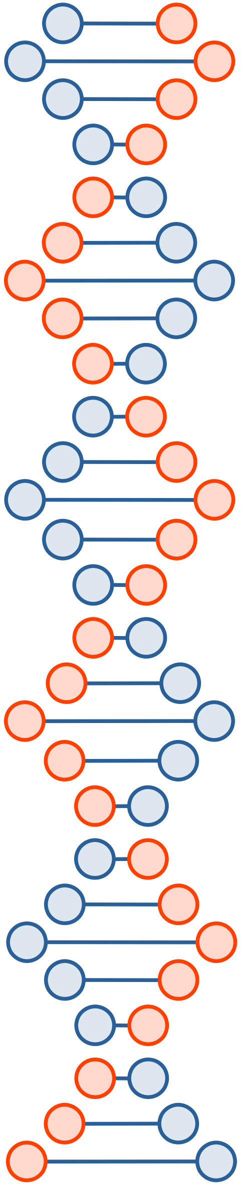
- **Process view**: The process view deals with the **dynamic aspects** of the system, explains the system processes and how they communicate, and focuses on the run time behavior of the system. The process view addresses concurrency, distribution, integrator, performance, and scalability, etc.
- UML diagrams to represent process view include the **sequence diagram**, communication diagram, activity diagram.

4+1 architectural view model

- **Development view**: The development view illustrates a system from a programmer's perspective and is concerned with software management. This view is also known as the implementation view.
- **UML Diagrams** used to represent the development view include the Package diagram and the Component diagram.

4+1 architectural view model

- **Physical view**: The physical view (aka the deployment view) depicts the system from a system engineer's point of view. It is concerned with the topology of software components on the physical layer as well as the physical connections between these components.
- **UML diagrams** used to represent the physical view include the deployment diagram.



Software Architecture Patterns

Common Architectural Design

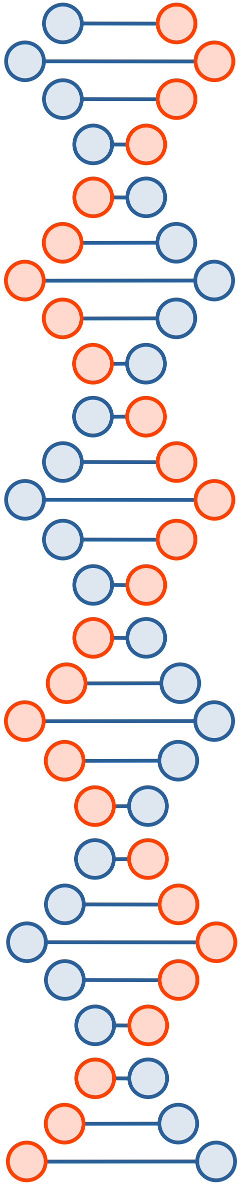
Category	Architectural Design	Description
Communication	Message bus	Prescribes use of a software system that can receive and send messages using one or more communication channels.
	Service–Oriented Architecture (SOA)	Defines the applications that expose and consume functionality as a service using contracts and messages.

Common Architectural Design

Category	Architectural Design	Description
Deployment	Client/server	Separate the system into two applications, where the client makes requests to the server.
	3-tier or N-tier	Separates the functionality into separate segments with each segment being a tier located on a physically separate computer.

Common Architectural Design

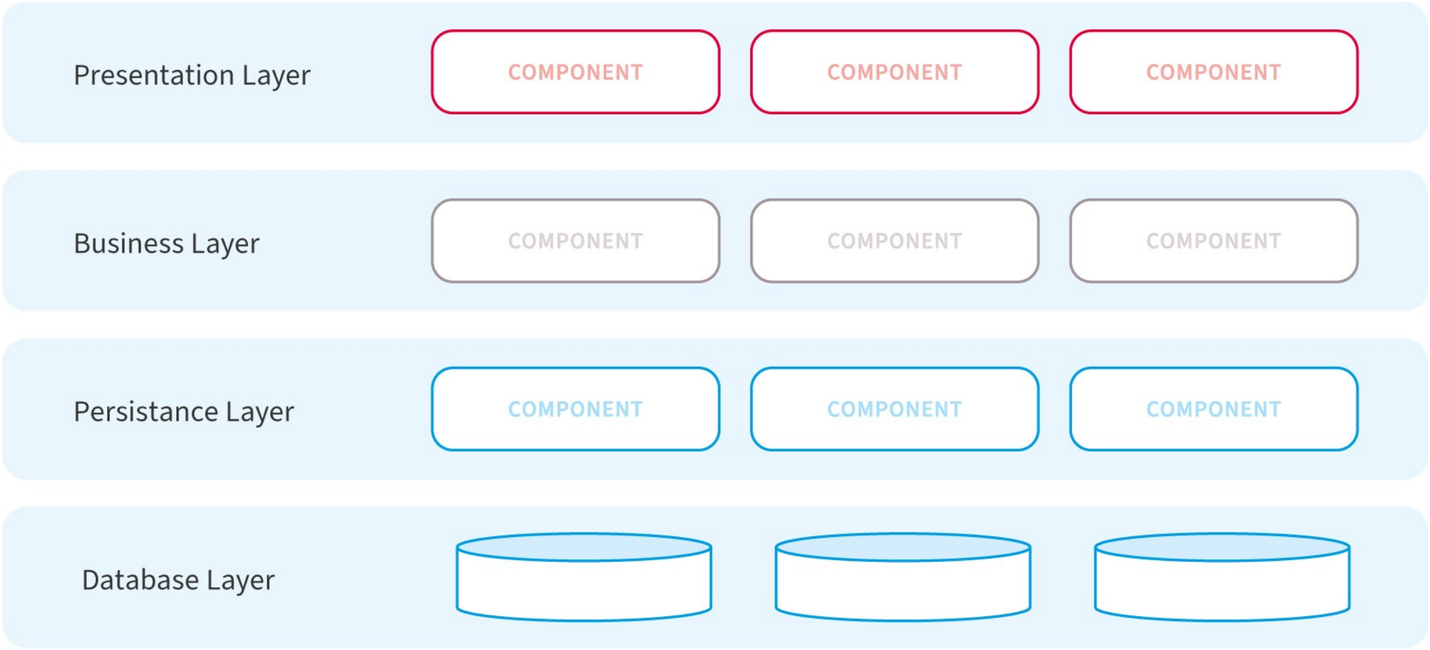
Category	Architectural Design	Description
Domain	Domain Driven Design	Focused on modeling a business domain and defining business objects based on entities within the business domain.



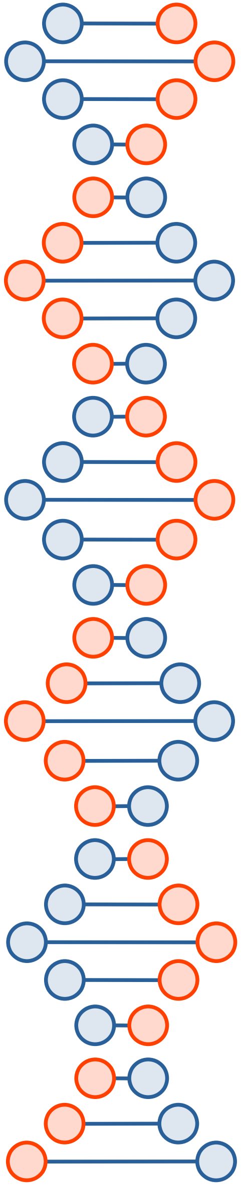
Common Architectural Design

Category	Architectural Design	Description
Structure	Component Based	Breakdown the application design into reusable functional or logical components that expose well-defined communication interfaces.
	Layered	Divide the concerns of the application into stacked groups (layers).
	Object oriented	Based on the division of responsibilities of an application or system into objects, each containing the data and the behavior relevant to the object.

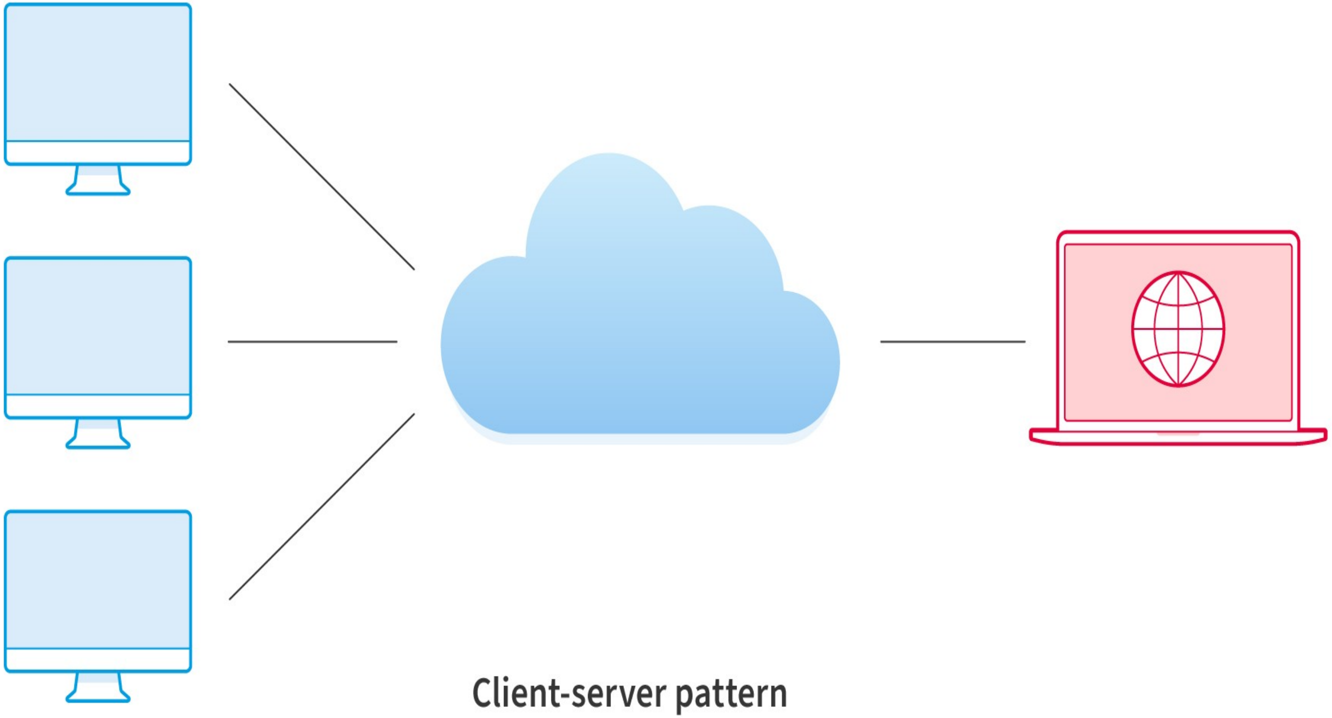
Layered architecture



Layered pattern

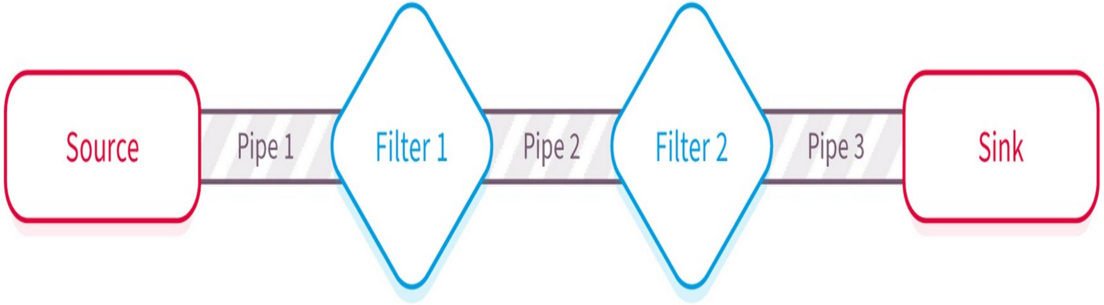


Client-server



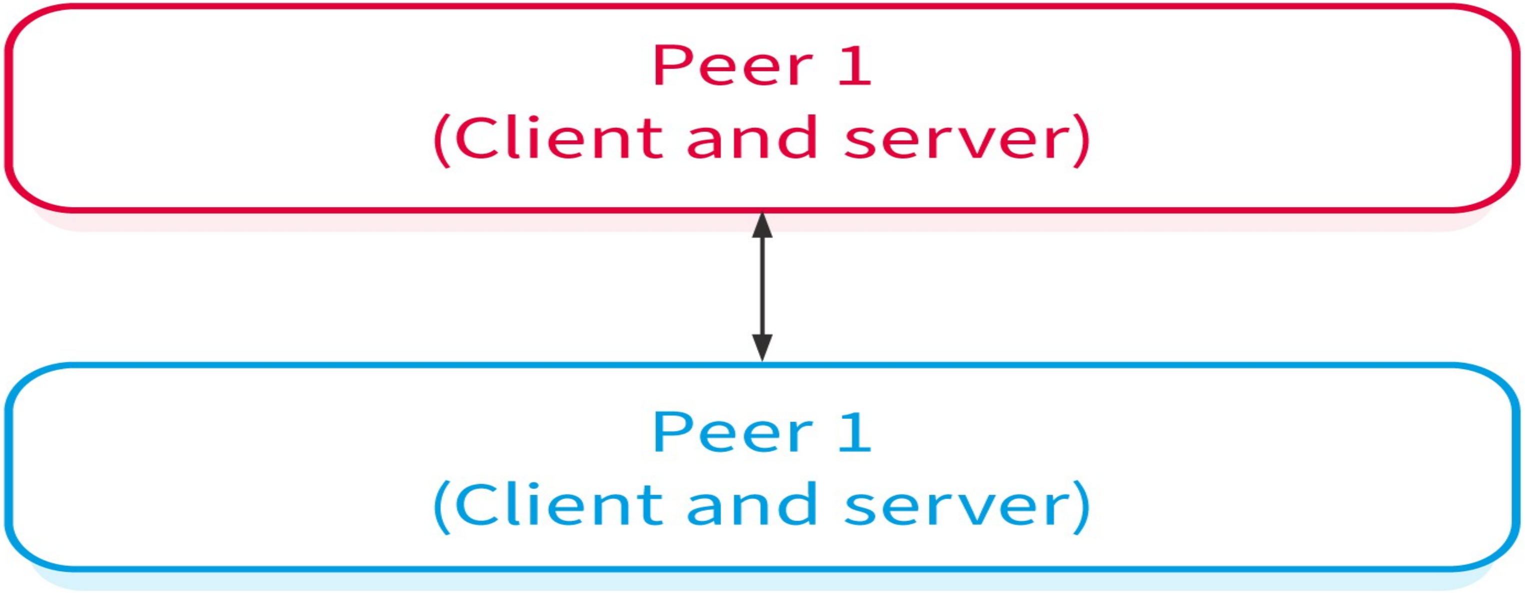
Client-server pattern

Layered architecture



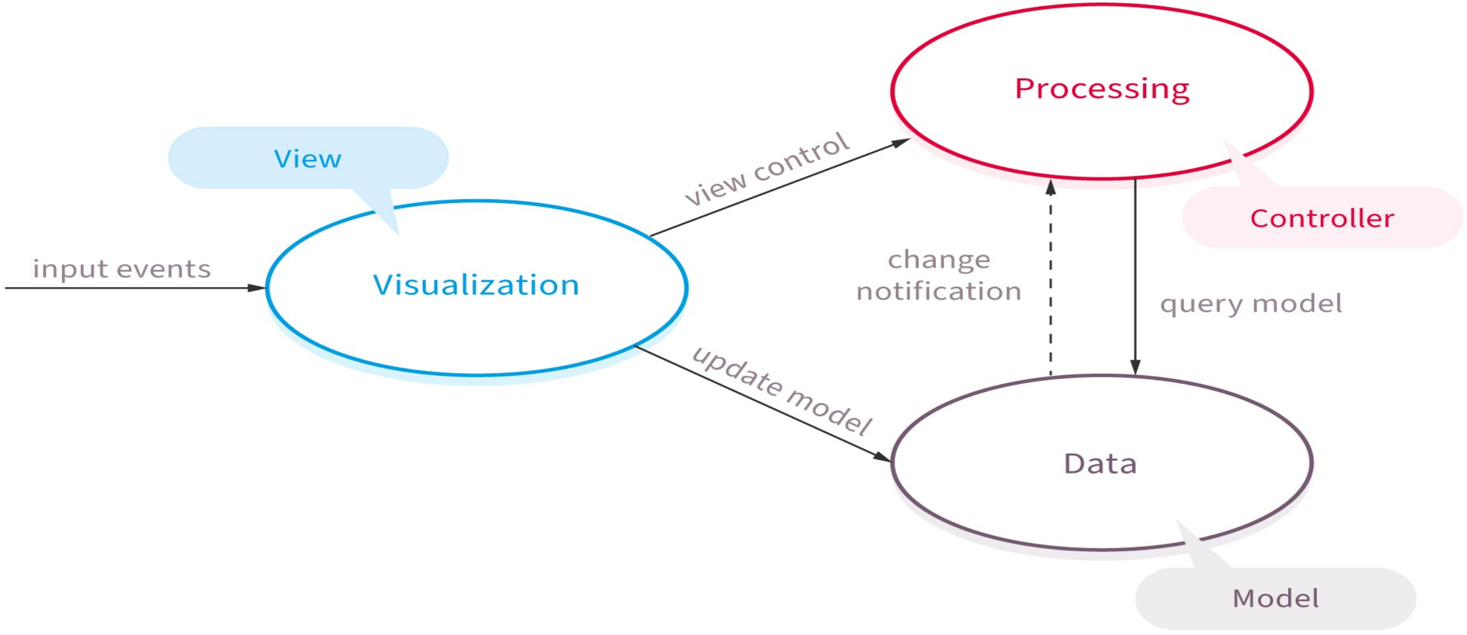
Pipe-filter pattern

Peer-to-peer (P2P)



Peer-to-peer pattern

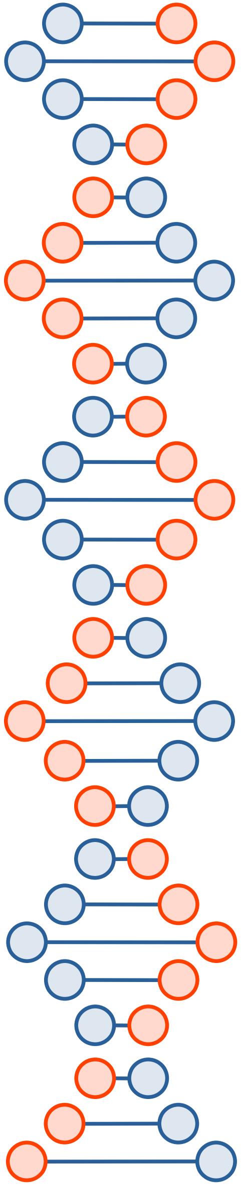
Model-View-Controller (MVC)



Model-view-controller pattern

Types of Architecture

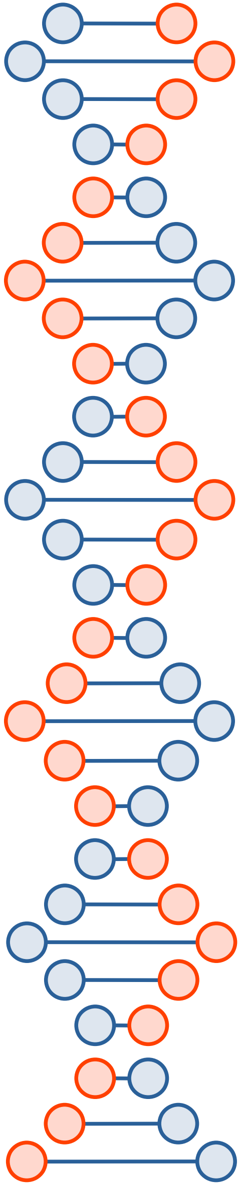
- There are four types of architecture from the viewpoint of an enterprise and collectively, these architectures are referred to as enterprise architecture.



Types of Architecture

- **Business architecture** − Defines the strategy of business, governance, organization, and key business processes within an enterprise and focuses on the analysis and design of business processes.
-
- **Application (software) architecture** − Serves as the blueprint for individual application systems, their interactions, and their relationships to the business processes of the organization.
-
- **Information architecture** − Defines the logical and physical data assets and data management resources.

- **Information technology (IT) architecture** − Defines the hardware and software building blocks that make up the overall information system of the organization.



- **Thank you**