Lecture_9

Architectural Patterns

Architectural Patterns - Client Server

- This pattern is used when data in a shared database has to be accessed from a range of locations. Because servers can be replicated they may be used when the load on the system is variable
- Disadvantages
  - Each service is a single point of failure so it is susceptible to DDOS attacks
  - Performance may be unpredictable because it depends on the networks
  - This may lead to management problems

Architectural Patterns - Pipe and Filter

- The processing of the data in a system is organized so that each processing component is discrete and carries out it process
- Violates:
  - Dont repeat yourself - Repeat Functionality
  - Kiss - Keep it simple stupid
- Some tasks might be compute-intensive tasks that could benefit from running on powerful hardware
- Able to reuse subsystems in the big modules
- Replicate a system in order to handle heavier tasks

Task	Type
Incoming Message	
Message Queue	
Virus Protection	Subsystem Filter
Spam	Subsystem Filter

- Disadvantages:
  - The format for data transfer has to be agreed upon between communicating transformations. Each transformation must parse its input and unparse its output to the agreed form. This increases system overhead meaning it may be impossible to reuse functional transformation with incompatible data.
  - If data flows between filters in a pipeline is lost, the entire system fails

Architectural Patterns - Message Bus

- The pattern is commonly used to facilitate communication and decouple different components of a software system
- If one client retires from the system it does not impact others
- Advantages
  - Efficiency and Scalability
  - Reliability and Resilience
  - Flexibility and Interoperability
- Disadvantages
  - Performance and Latency
  - Ensuring messages are delivered in a timely manner. This can be especially challenging in high-volume systems, where message processing times can become a bottleneck
  - Complexity and Maintenance
  - Complex troubleshooting especially in large complex systems

Architectural Patterns - Popular

- Layered pattern
- Client-server pattern
- Master-slave pattern
- Pipe-filter pattern
- Broker pattern
- Peer-to-peer pattern
- Event-bus pattern
- Model-view-controller pattern
- Blackboard pattern
- Interpreter pattern

Architectural Evaluation

- Identify risks and mitigations
  - Aims to identify risks early, in an effort to minimize impact on the project.
- Compliance and standards
  - Make sure the project aligns with the regulations and rules set by governing bodies
- Stakeholder Alignment
  - Make sure the project aligns with the stakeholder's goals
- Cost and Resource Optimization
  - How to optimize the project cost and resource efficacy

Architectural Patterns - Decisions

- Architectural design is a creative process so the process differs depending on the type of system being developed
- A decision step is a step in the process
    - Record rationale
    - Tradeoffs
    - Backtracking