1 Lecture 5

1.1 Exercise 1: Integrating Containerization using Docker

Objective:

This exercise focuses on integrating containerization, specifically using Docker, into your software architecture. By containerizing different components of your system, you aim to enhance the portability, scalability, and maintainability of your software, allowing for more seamless deployment, scaling, and development processes.

Tasks:

- *Identify Components for Containerization:* From the systems and subsystems identified in previous exercises, determine which components would benefit most from containerization.
  - Consider components that require isolation, have specific environment requirements, or need to be scalable.
- *Design Dockerfiles and Compose Files:* For the identified components. You can use the presented example system for inspiration.
  - Create Dockerfiles to define the environment, dependencies, and configurations needed.
  - If multiple containers need to work together, design Docker Compose files to manage the interactions between them.
- *Justify Your Choices:* Clearly explain the reasoning behind containerizing the selected components and the configurations chosen in the Dockerfiles and Compose files. Relate your decisions back to how they improve the quality attributes of your system.
- *Assess Impact on Architecture:* Reflect on the integration of Docker and its impact on the software architecture's quality attributes, such as deployability, modifiability, and scalability.

Example: Financial Analysis System

- **Components for Containerization:** Data Processing Subsystem, User Interface Subsystem.

Justification:

- **Data Processing Subsystem:** Requires a specific set of dependencies and environment configurations to process financial data, making it an ideal candidate for containerization to ensure consistent behavior across different environments.

- **User Interface Subsystem:** Containerizing allows for easier scaling to handle varying loads and isolates the user interface to prevent conflicts with other subsystems.
- *Quality Attributes:* The use of Docker enhances the deployability and scalability of the system while ensuring consistent behavior across different deployment environments.

Deliverables:

- A document detailing the components selected for containerization, the designed Dockerfiles and Docker Compose files, and their justifications, along with insights on the impact on the architecture's quality attributes.
- A minimum of 2 components should be containerized and communicate using a containerized message bus. The images for these components should be built and pushed as a part of your CI/CD pipeline.
- Optionally, you can also add a containerized database.

Assessment Criteria:

The evaluation will focus on the appropriateness of the chosen components for containerization, the effectiveness of the designed Dockerfiles and Compose files in addressing the components' needs, the depth and clarity of the justifications provided, and the understanding of the impact of integrating Docker on the architecture's quality attributes.