Software Architecture

Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo

Software Architecture

- Software architecture is the high level structure of a software system
  - Architecture design is a creative activity
- It links requirements (problem) to design (solution)
- Architecture is the first step to design the software solution

Architectural Design

- Architectural design is often performed together with requirements specification

- It involves
  - Identification of the main software components
  - Definition of the communication interfaces between components

Advantages

- Software architecture facilitates communication between stakeholders
- It captures early decisions about the high-level design
  - Such as, with respect to non-functional requirements
- It allows reuse of design components between projects

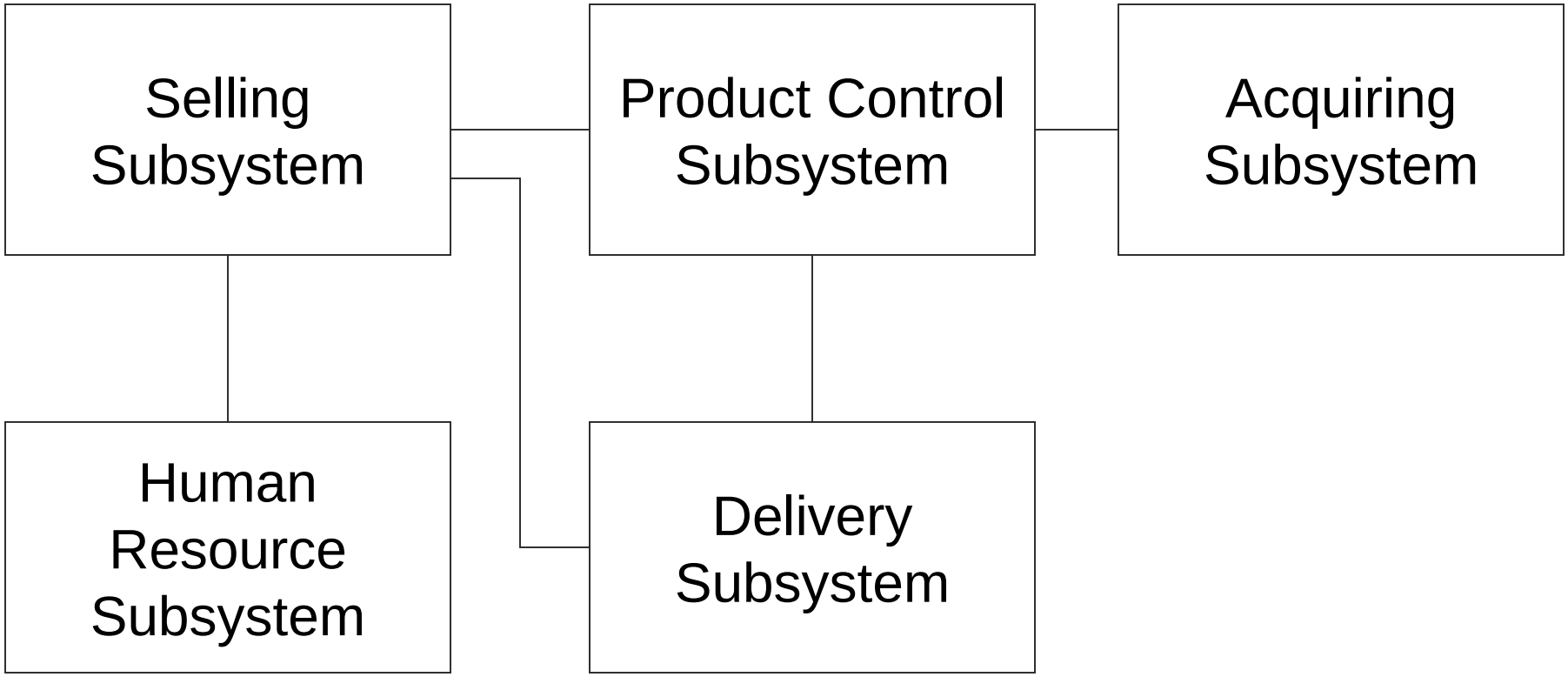
Bad Architecture Impacts…

- System performance
  - Bottlenecks in the architectural design
- System distribution
  - Enterprise systems may execute in many machines
- Software maintenance
  - Components should to be pluggable

Component Diagram

- The UML Component Diagram is often used to represent software architecture
  - It depicts how components are wired together to form larger software systems
- Simplified notation
  - Boxes to represent components
  - Lines to represent communication channels between two components
  - Inner boxes to represent composed components

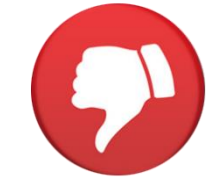
Example of Architecture (1)



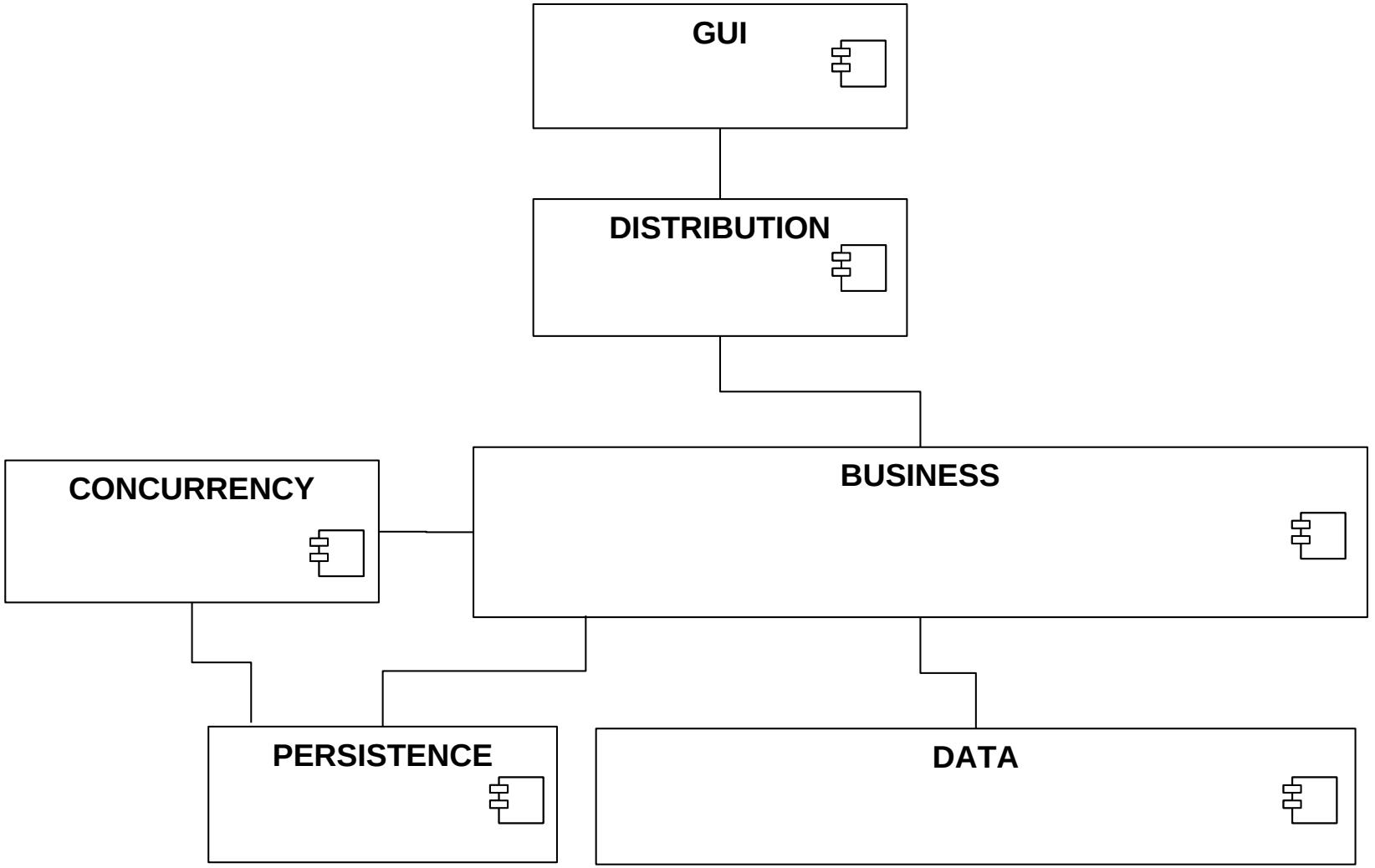
Advantages and Drawbacks

- Advantages
  - It is easy to understand
  - It facilitates the design planning
  - It supports parallel development

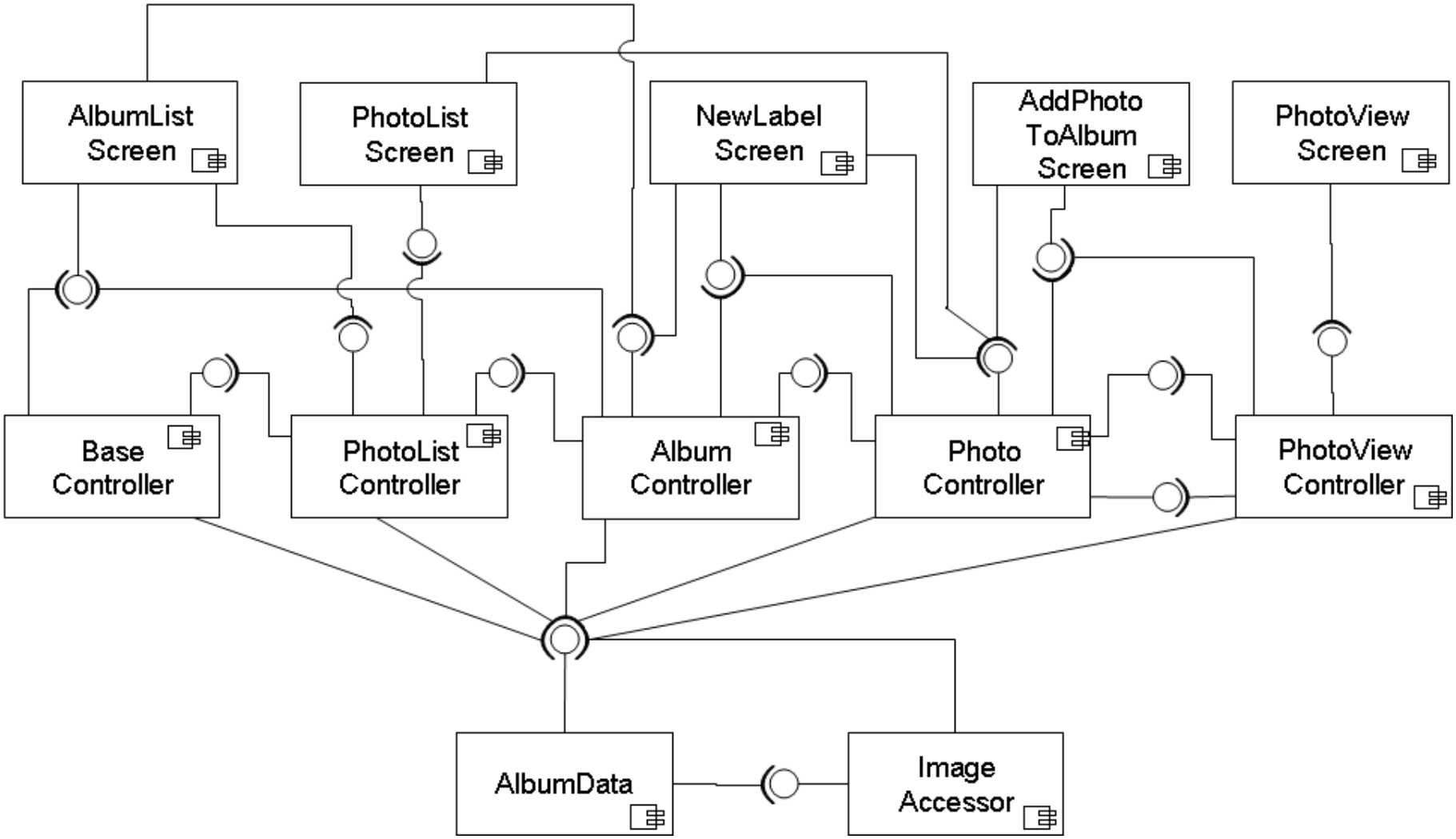
- Drawbacks
  - It does not detail the nature of relationships
  - It does not show internal properties of components



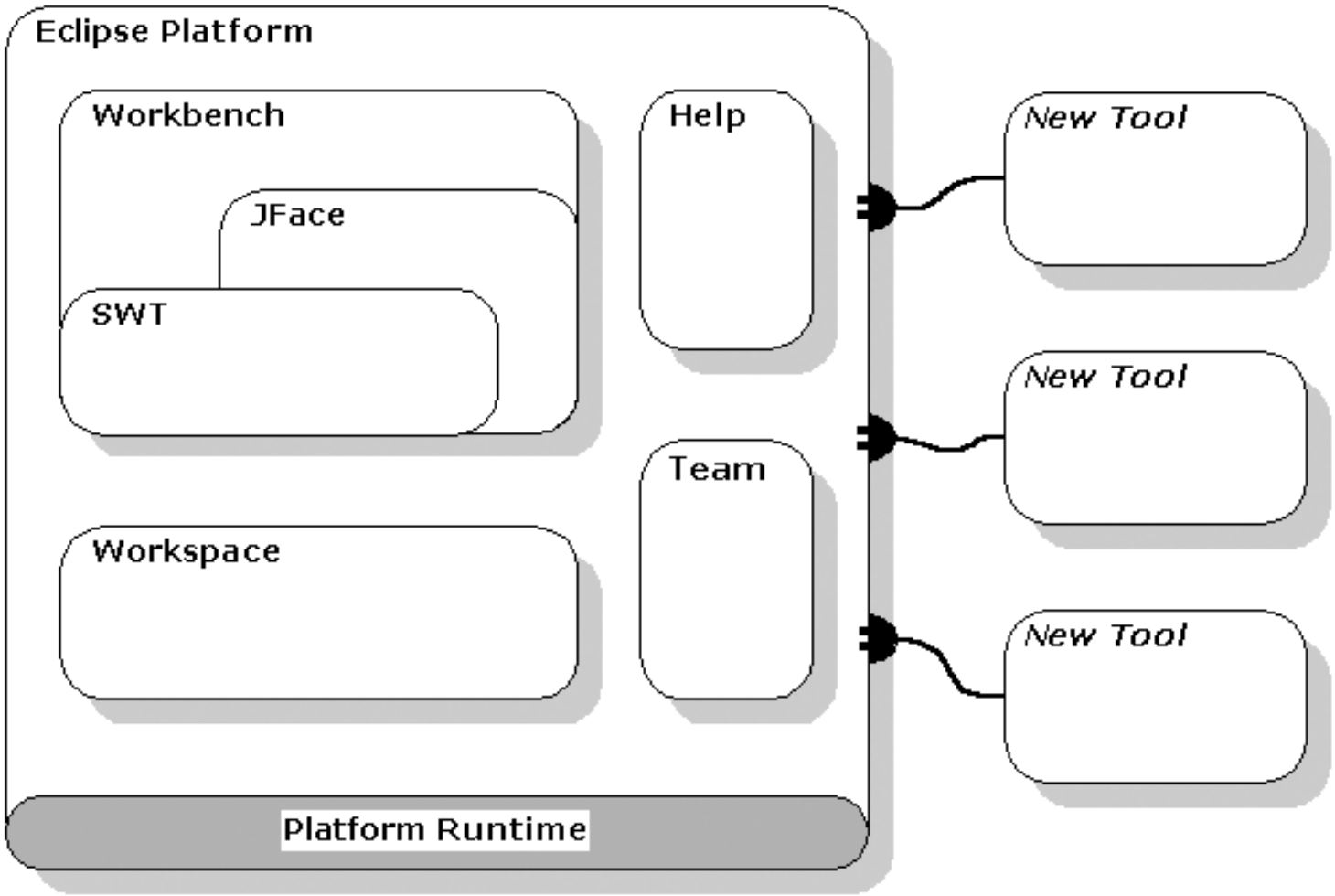
Example of Architecture (2)



Example of Architecture (3)



Example of Architecture (4)



Architecture Design Decisions

- Software architects must be aware of issues, such as
  - How is the system distributed among machines?
  - How are functionalities decomposed among components?
  - How to evaluate an architecture according to non-functional requirements?
  - Is there a general architecture to be used?

Non-Functional Requirements

- NFR must be considered in a software architecture
  - **Performance**: high communication among distributed components must be avoided
  - **Security**: private features must be hidden, for instance, in inner layers
  - **Availability**: critical components need to be fault tolerant or redundant

Bibliography

- Ian Sommerville. **Software Engineering**, 9th Edition. Pearson Education, 2011.
  - Chapter 6 Architectural Design