



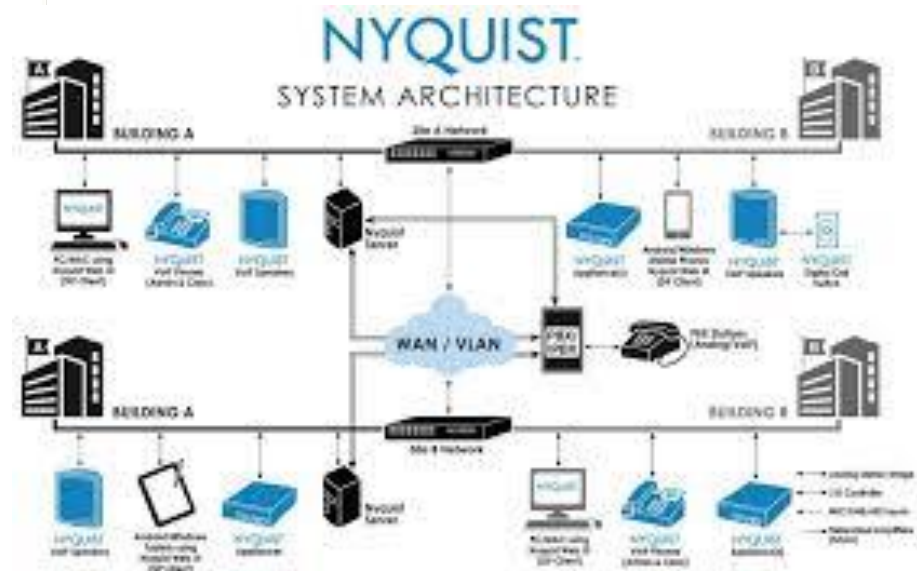
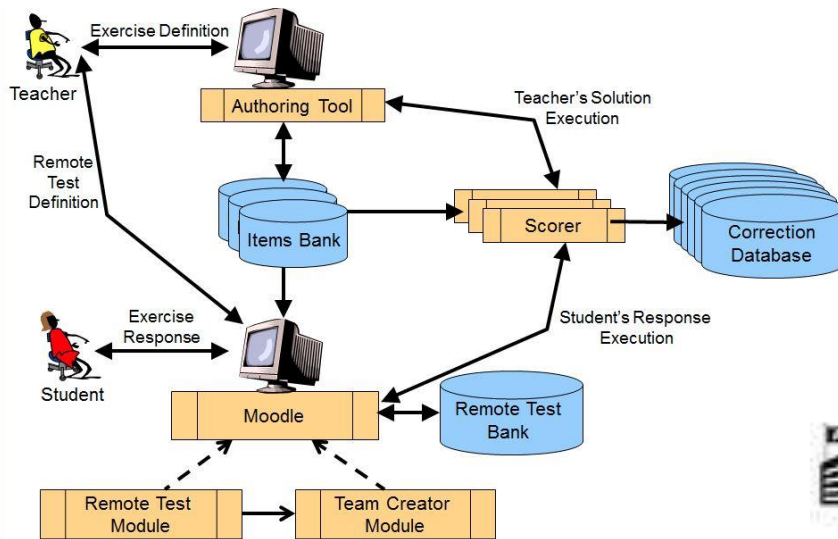
Chapter 6 – Architectural Design

Topics covered

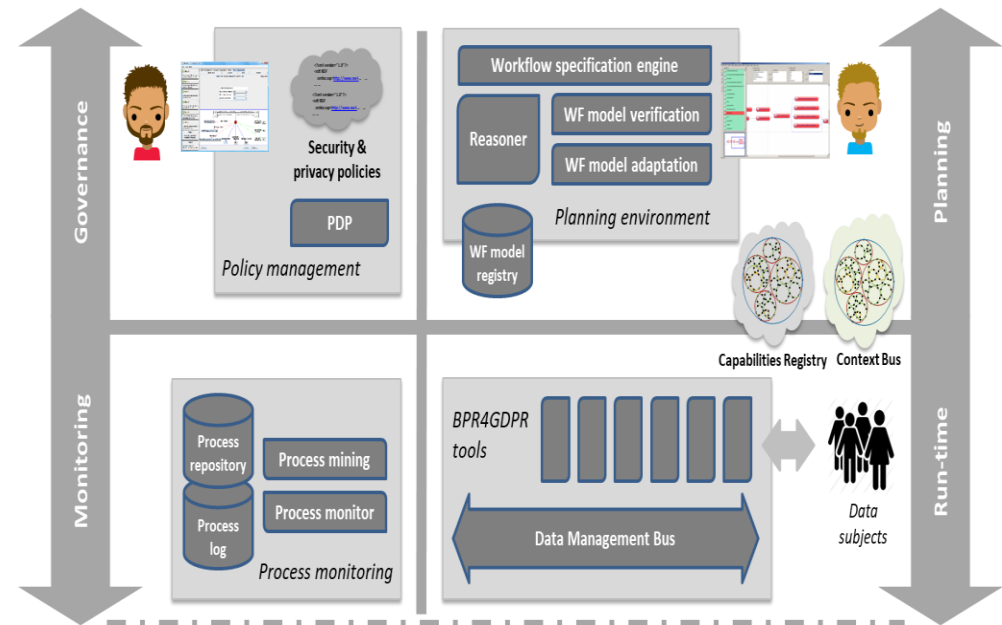
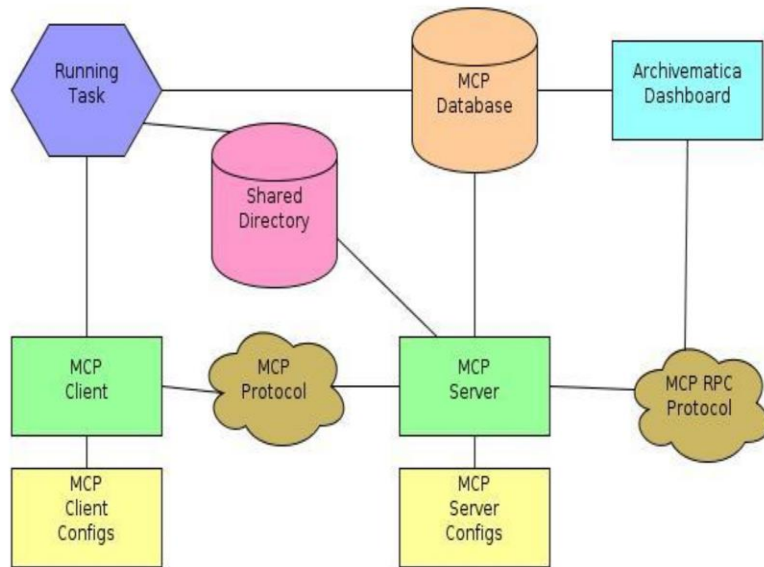


- ✧ Architectural design decisions
- ✧ Architectural views
- ✧ Architectural patterns
- ✧ Application architectures

System Architectures



System Architectures



Software Architectures



What is Software Architecture?



- The architecture is not the operational software, it is a representation that enables a software engineer to:
 1. Analyze the effectiveness of the design in meeting its stated requirements,
 2. Consider architectural alternatives at a stage when making design changes is still relatively easy, and
 3. Reduce the risks associated with the construction of the software.

Architectural Descriptions



- The IEEE Computer Society has proposed IEEE Std-42010-2011, *Systems and Software Engineering – Architecture Description*.

Describes the use of architecture viewpoints, architecture frameworks, and architecture description languages as a means of codifying the conventions and common practices for architectural description.

The IEEE Standard defines an *architectural description* (A D) as a “a collection of products to document an architecture.”

An architecture description shall identify the system stakeholders having concerns considered fundamental to the architecture of the system-of-interest.

These concerns shall be considered when applicable and identified in the architecture description: system purpose, suitability of the architecture, feasibility of constructing and deploying the system, risks and impacts of the system, and the maintainability and evolvability of the system.

Architectural abstraction



- ✧ **Architecture in the small** is concerned with the architecture of individual programs. At this level, we are concerned with the way that an individual program is decomposed into components.
- ✧ **Architecture in the large** is concerned with the architecture of complex enterprise systems that include other systems, programs, and program components. These enterprise systems are distributed over different computers, which may be owned and managed by different companies.

Architectural design



- ✧ Architectural design is concerned with understanding how a software system should be **organized and designing the overall structure of that system.**
- ✧ Architectural design is the **critical link between design and requirements engineering**, as it identifies the main structural components in a system and the relationships between them.
- ✧ The **output of the architectural design process** is an **architectural model** that describes how the system is organized as a set of communicating components.

Why is Software Architecture Important?



- Software architecture provides a representation that facilitates communication among all stakeholders interested in the development of a computer-based system.
- Architecture highlights early design decisions that will have a profound impact on all software engineering work that follows.
- Architecture constitutes a relatively small, intellectually graspable mode of how the system is structured and how its components work together.

Agility and Architecture



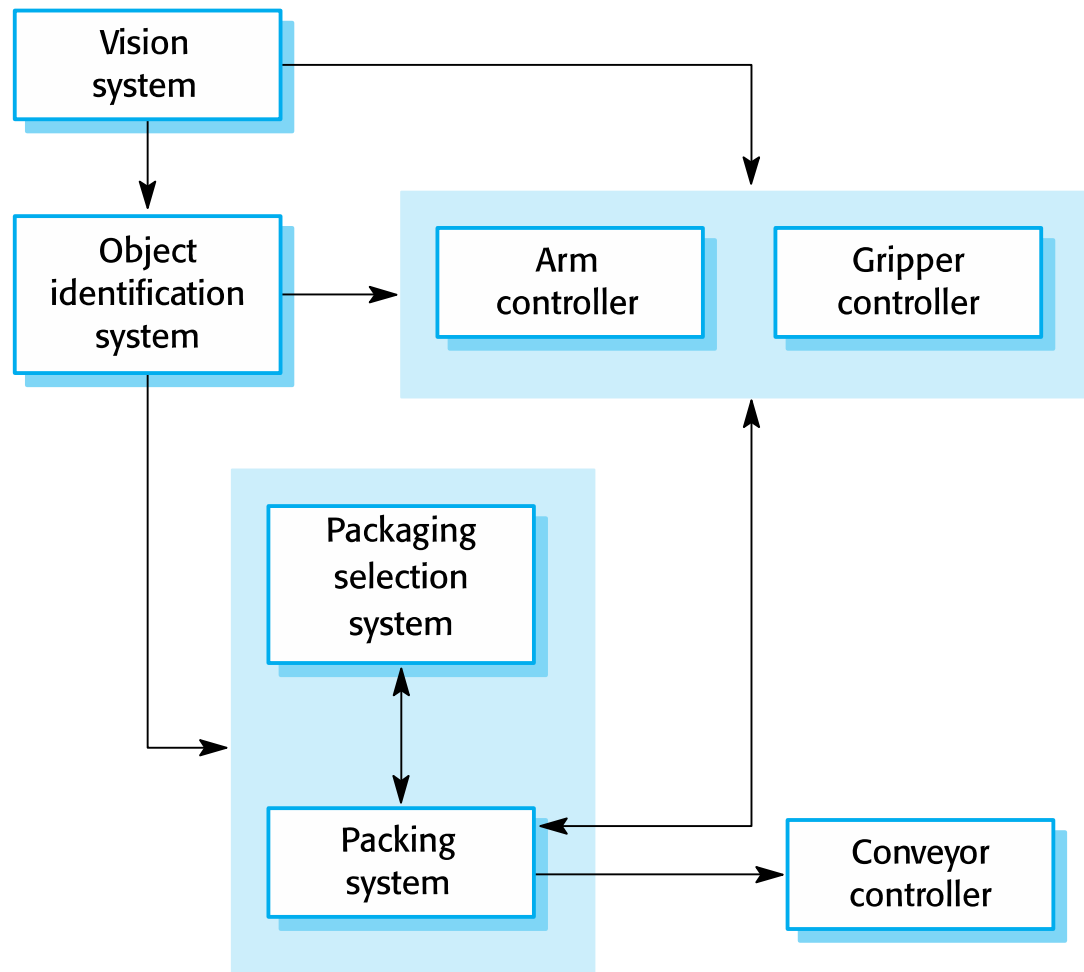
- ✧ It is generally accepted that an **early stage of agile processes** is to design an **overall systems architecture**.
- ✧ **Refactoring** the system architecture is usually **expensive** because it affects so many components in the system

Agility and Architecture



- To avoid rework, user stories are used to create and evolve an architectural model (walking skeleton) before beginning any coding.
- Use models which allow software architects to add user stories to the evolving storyboard and works with the product owner to prioritize the architectural stories as “sprints” (work units) are planned.
- Well run agile projects include delivery of architectural documentation during each sprint.
- After the sprint is completed, the architect reviews the working prototype for quality before the team presents it to the stakeholders in a formal sprint review.

The architecture of a packing robot control system



Architectural Styles/Patterns



Architectural Styles

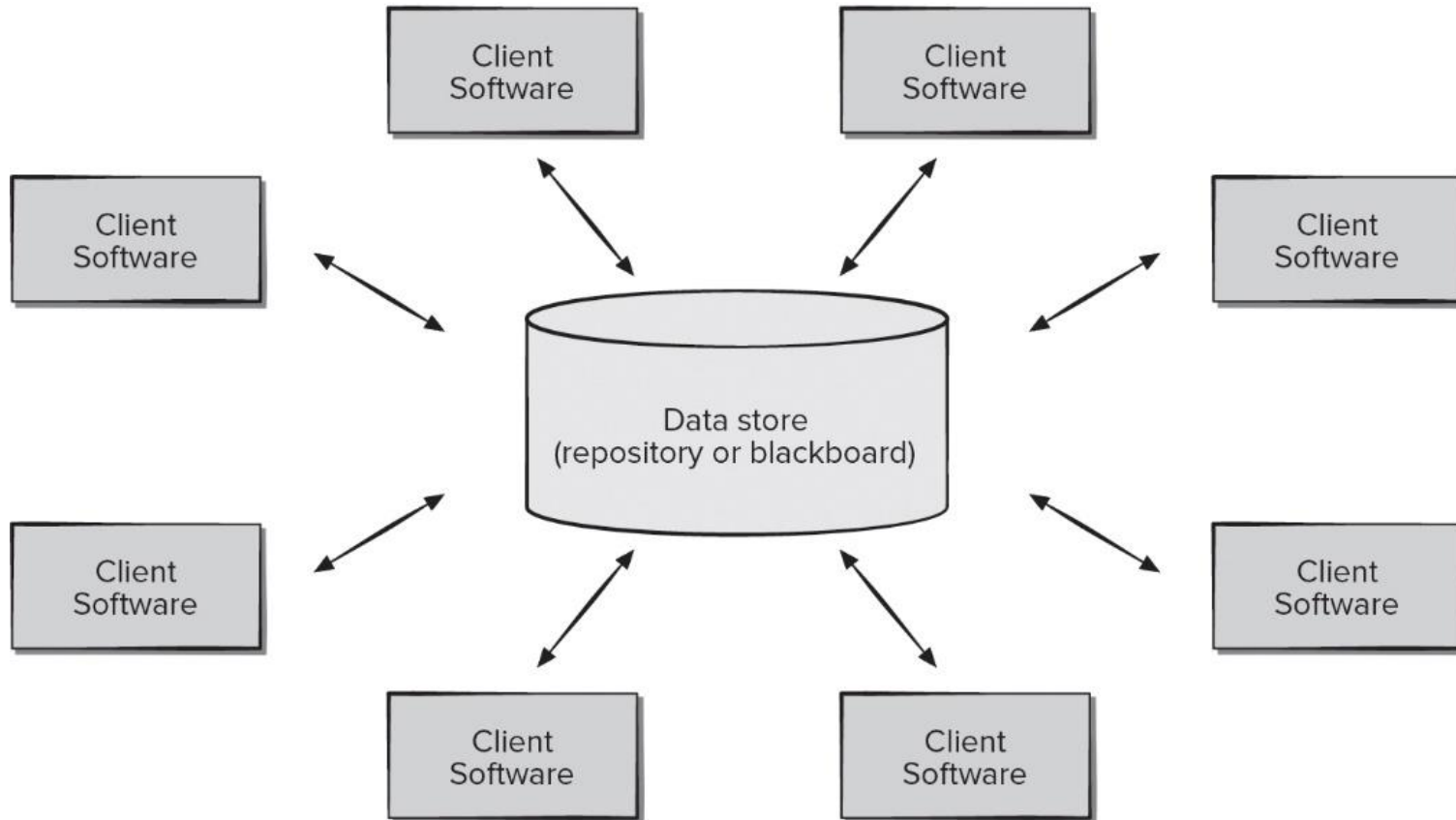


- Each style describes a system category that encompasses:
 1. **set of components** (for example: a database, computational modules) that perform a function required by a system.
 2. **set of connectors** that enable “communication, coordination and cooperation” among components.
 3. **constraints** that define how components can be integrated to form the system.
 4. **semantic models** that enable a designer to understand the overall properties of a system by analyzing the known properties of its constituent parts.

Data Centered Architecture



Copyright © McGraw-Hill Education. All rights reserved. No reproduction or distribution without the prior written consent of McGraw-Hill Education.

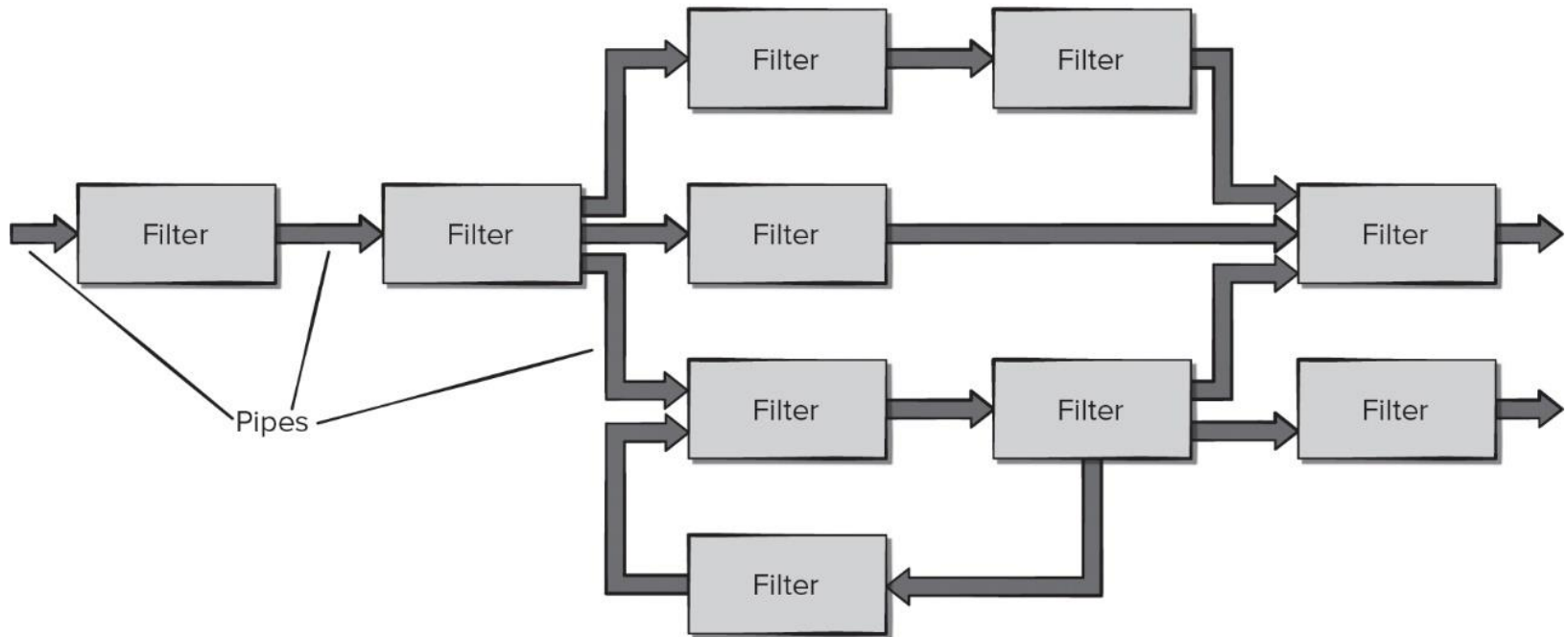


- [Access the text alternative for slide images.](#)

Data Flow Architecture



Copyright © McGraw-Hill Education. All rights reserved. No reproduction or distribution without the prior written consent of McGraw-Hill Education.

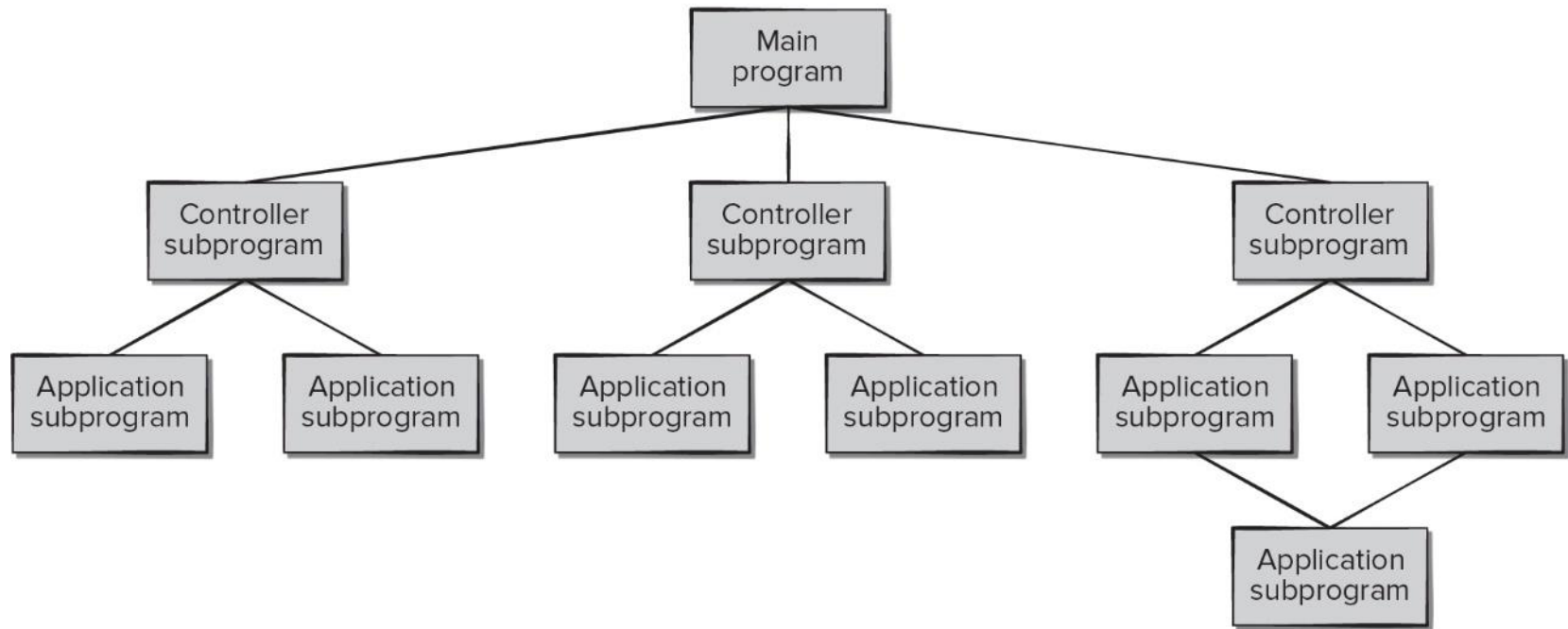


- [Access the text alternative for slide images.](#)

Stack(Call Return) Architecture



Copyright © McGraw-Hill Education. All rights reserved. No reproduction or distribution without the prior written consent of McGraw-Hill Education.

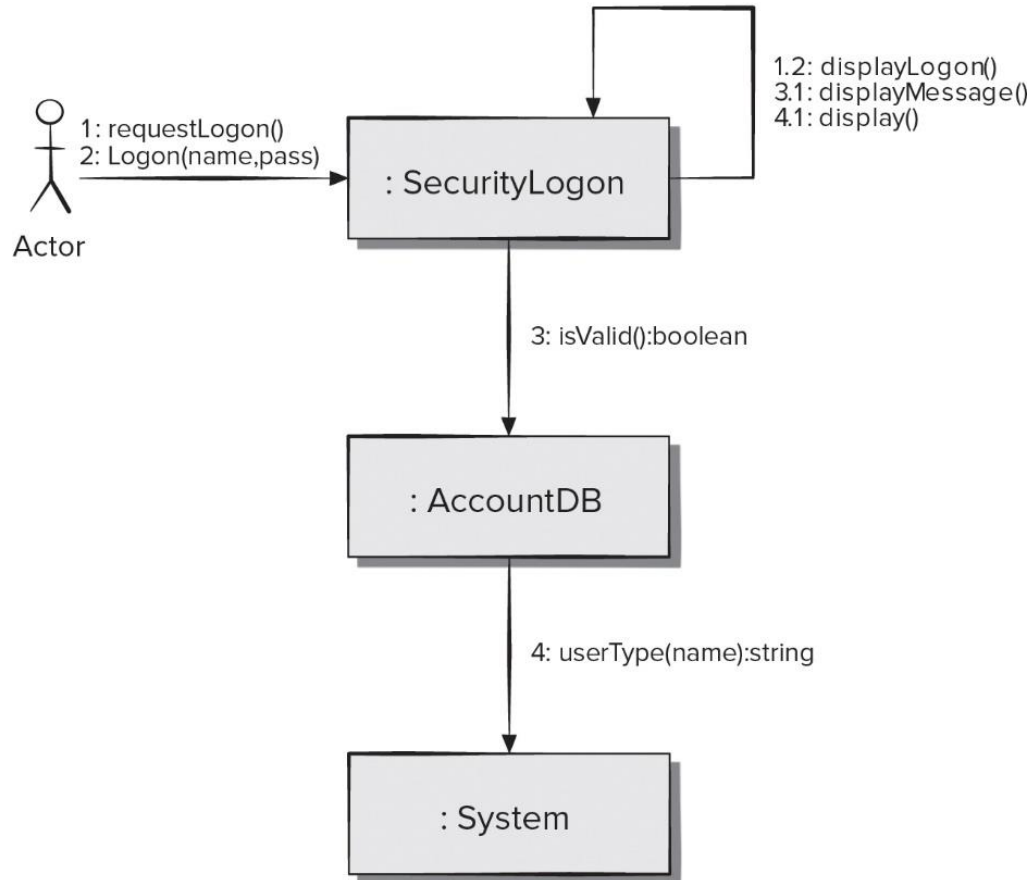


- [Access the text alternative for slide images.](#)

Object-Oriented Architecture



Copyright © McGraw-Hill Education. All rights reserved. No reproduction or distribution without the prior written consent of McGraw-Hill Education.

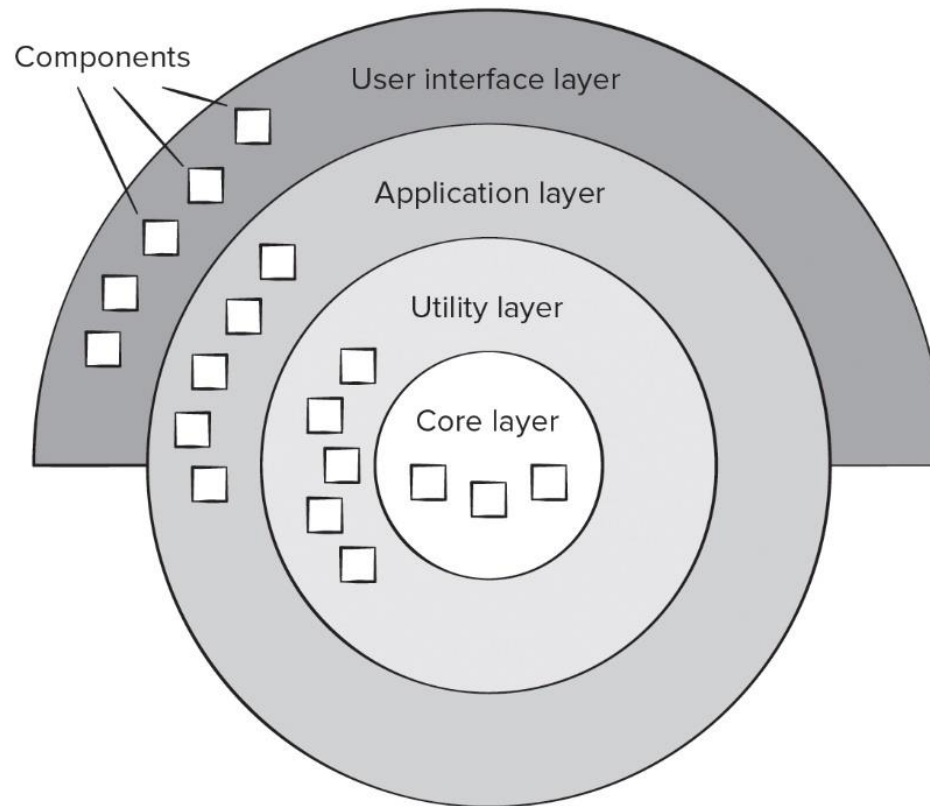


- [Access the text alternative for slide images.](#)

Layered Architecture



Copyright © McGraw-Hill Education. All rights reserved. No reproduction or distribution without the prior written consent of McGraw-Hill Education.

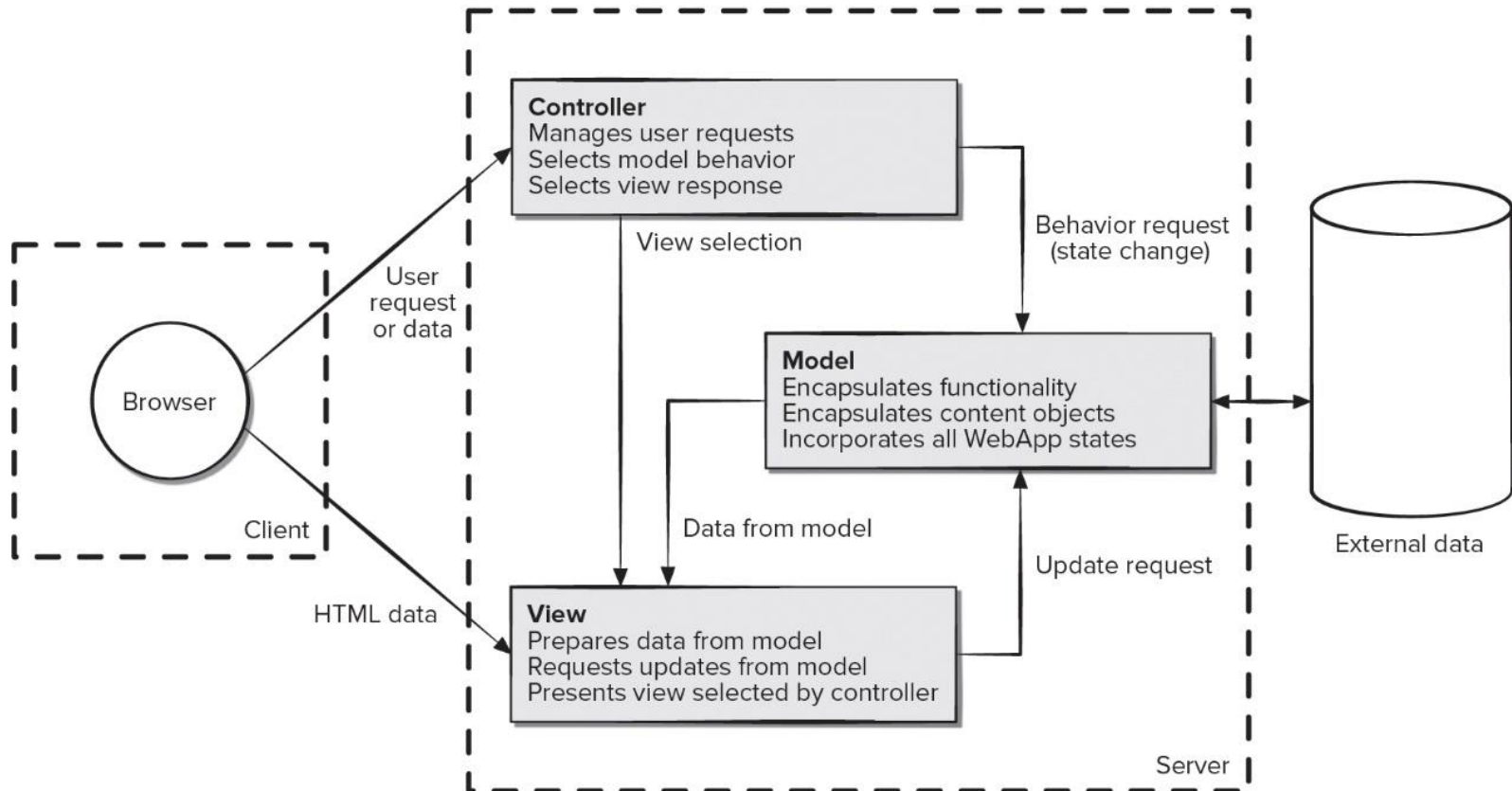


- [Access the text alternative for slide images.](#)

Model View Controller Architecture



Copyright © McGraw-Hill Education. All rights reserved. No reproduction or distribution without the prior written consent of McGraw-Hill Education.



Source: Adapted from Jacyntho, Mark Douglas, Schwabe, Daniel and Rossi, Gustavo, "An Architecture for Structuring Complex Web Applications," 2002, available at <http://www-di.inf.puc-rio.br/schwabe/papers/OOHDMJava2%20Report.pdf>

- [Access the text alternative for slide images.](#)

Architectural Organization and Refinement ₁



- **Control.**
- How is control managed within the architecture?
- Does a distinct control hierarchy exist, and if so, what is the role of components within this control hierarchy?
- How do components transfer control within the system?
- How is control shared among components?
- What is the control topology (that is, the geometric form that the control takes)?
- Is control synchronized, or do components operate asynchronously?

Architectural Organization and Refinement ²



- **Data.**
- How are data communicated between components?
- Is the flow of data continuous, or are data objects passed to the system sporadically?
- What is the mode of data transfer?
- Do data components exist, and if so, what is their role?
- How do functional components interact with data components?
- Are data components passive or active?
- How do data and control interact within the system?

Advantages of explicit architecture



✧ Stakeholder communication

- Architecture may be used as a focus of discussion by system stakeholders.

✧ System analysis

- Means that analysis of whether the system can meet its non-functional requirements is possible.

✧ Large-scale reuse

- The architecture may be reusable across a range of systems
- Product-line architectures may be developed.