

Chapter 9

- **Architectural Design**

Why Architecture?

The architecture is not the operational software. Rather, it is a representation that enables a software engineer to:

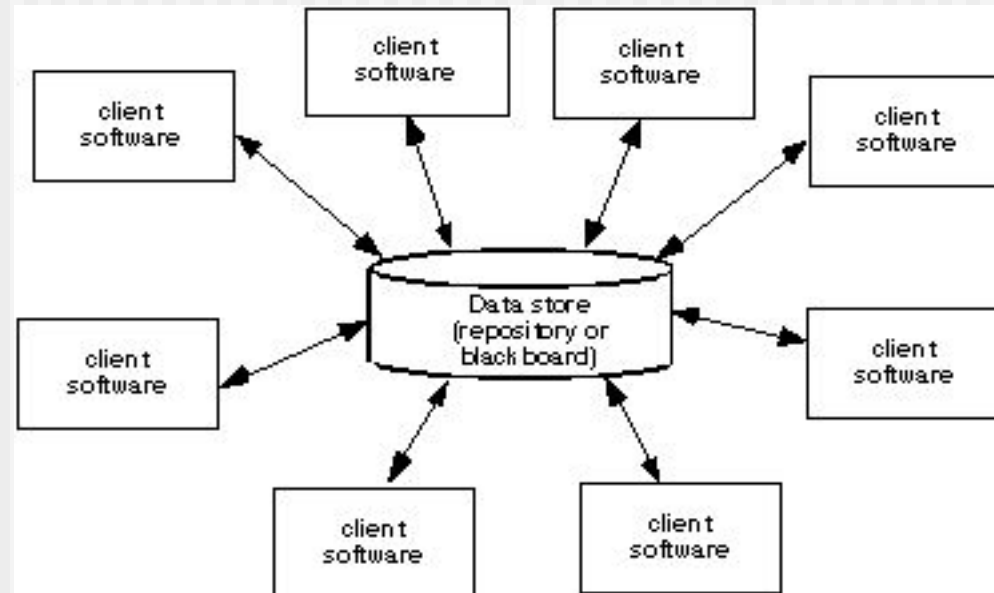
- (1) **analyze the effectiveness of the design** in meeting its stated requirements,
- (2) **consider architectural alternatives** at a stage when making design changes is still relatively easy, and
- (3) **reduce the risks** associated with the construction of the software.

Architectural Styles

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

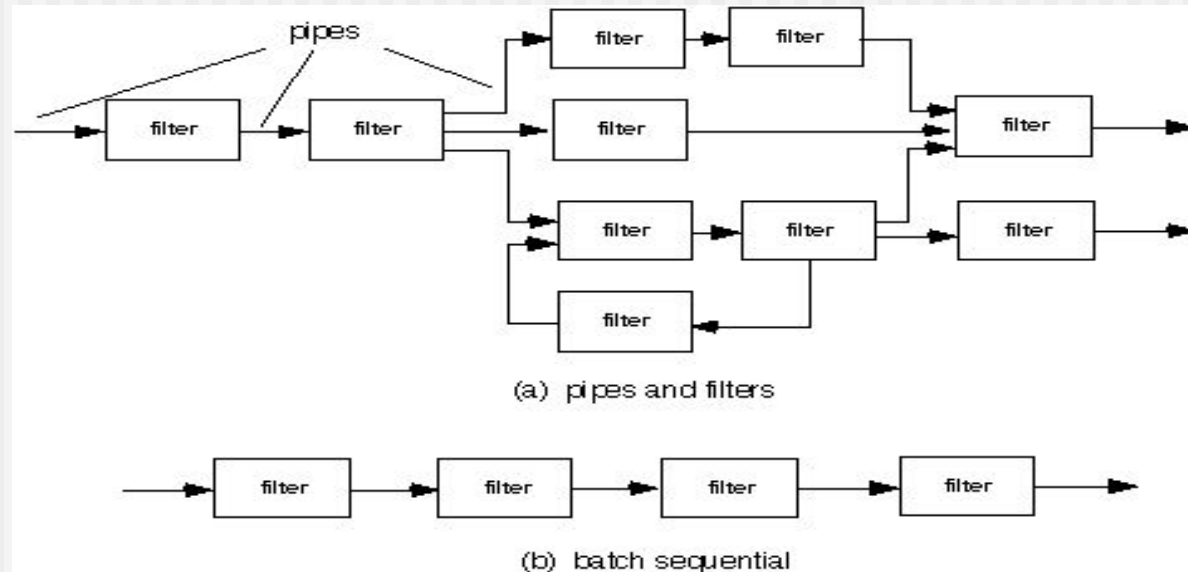
- Data-centered architectures
- Data flow architectures
- Call and return architectures
- Object-oriented architectures
- Layered architectures

Data-Centered Architecture



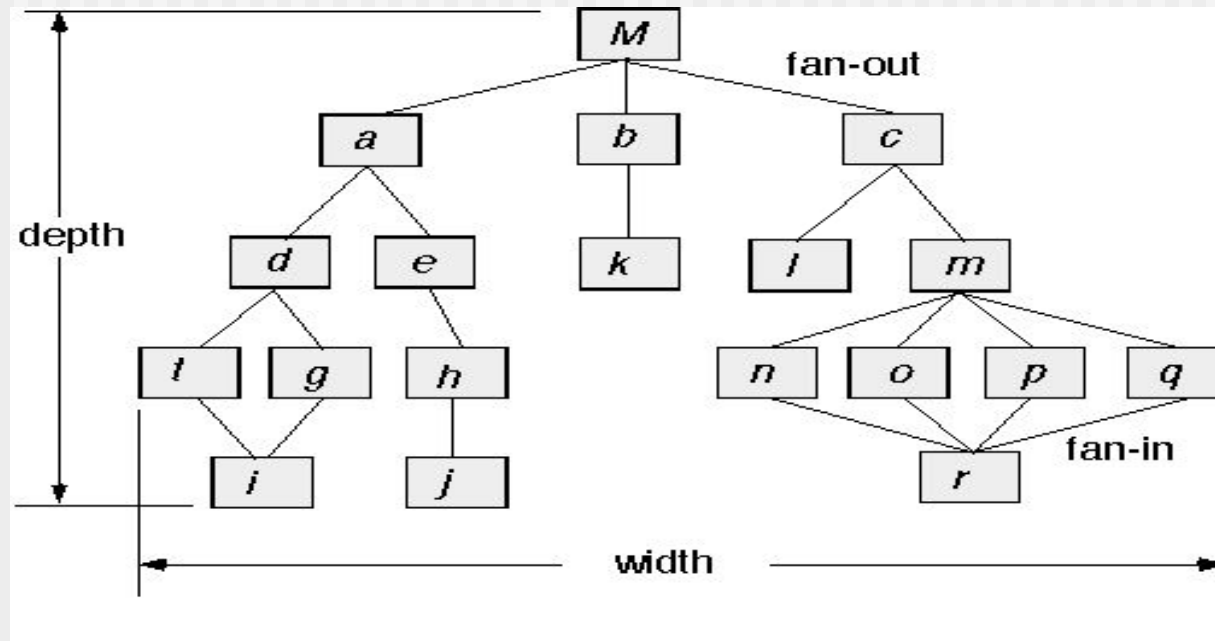
Data resides at the center and accessed by other components for add , update , delete or modify

Data Flow Architecture



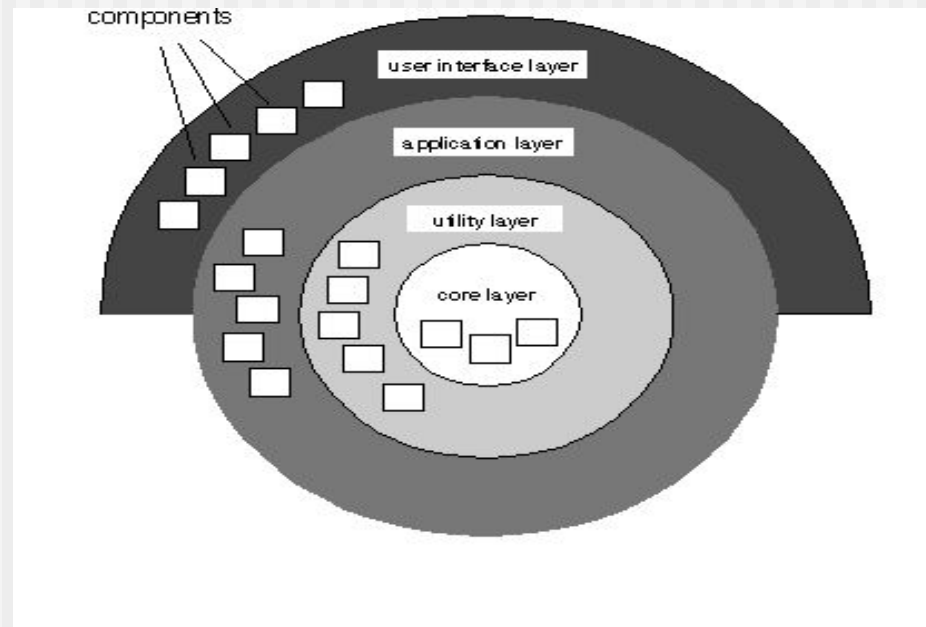
Input data are to be transformed through a series of computational or manipulative components into output data

Call and Return Architecture



Main program subprogram architecture , program structure decomposes function into a control hierarchy

Layered Architecture

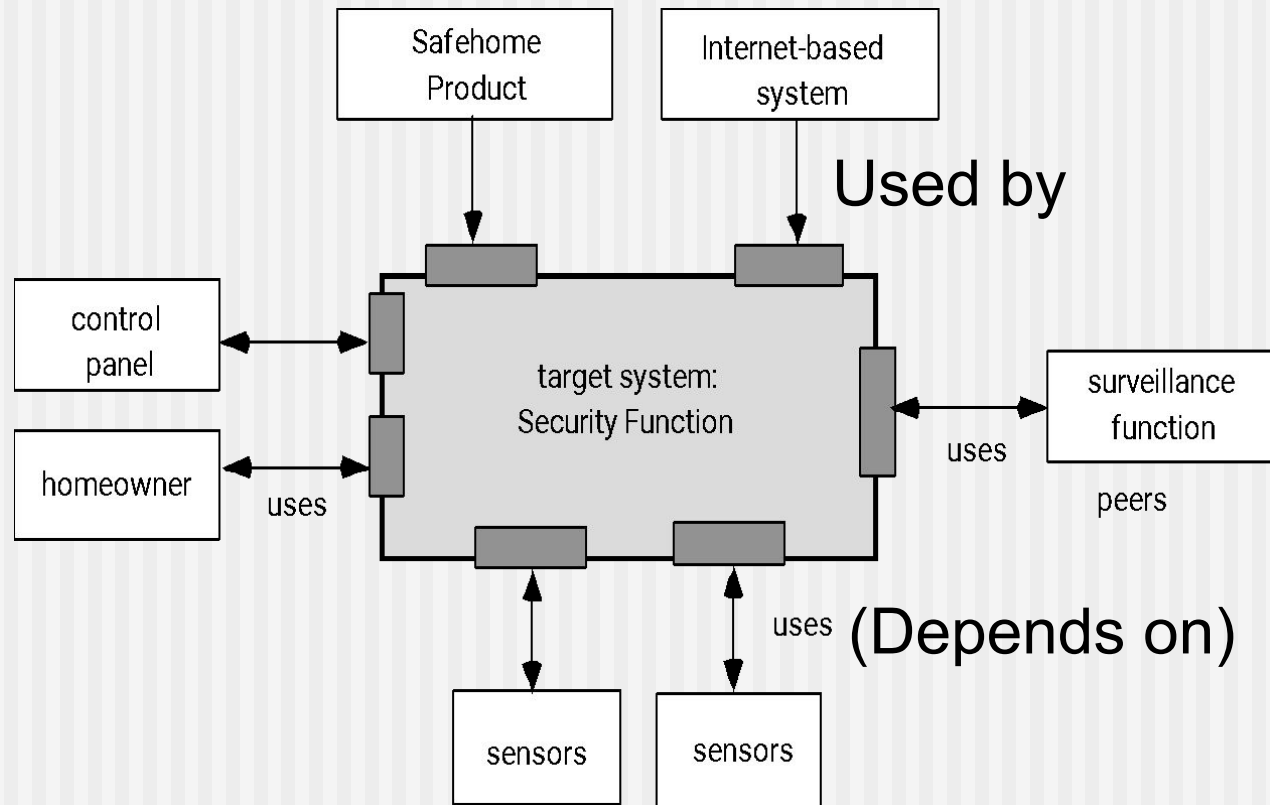


Number of different layers are defined , each accomplishing operations that progressively become closer to the machine instruction set

Architectural Design

- The software must be placed into context
 - the design should define the external entities (other systems, devices, people) that the software interacts with and the nature of the interaction
- A set of architectural archetypes should be identified
 - An *archetype* is an abstraction (similar to a class) that represents one element of system behavior
- The designer specifies the structure of the system by defining and refining software components that implement each archetype

Architectural Context



Archetypes

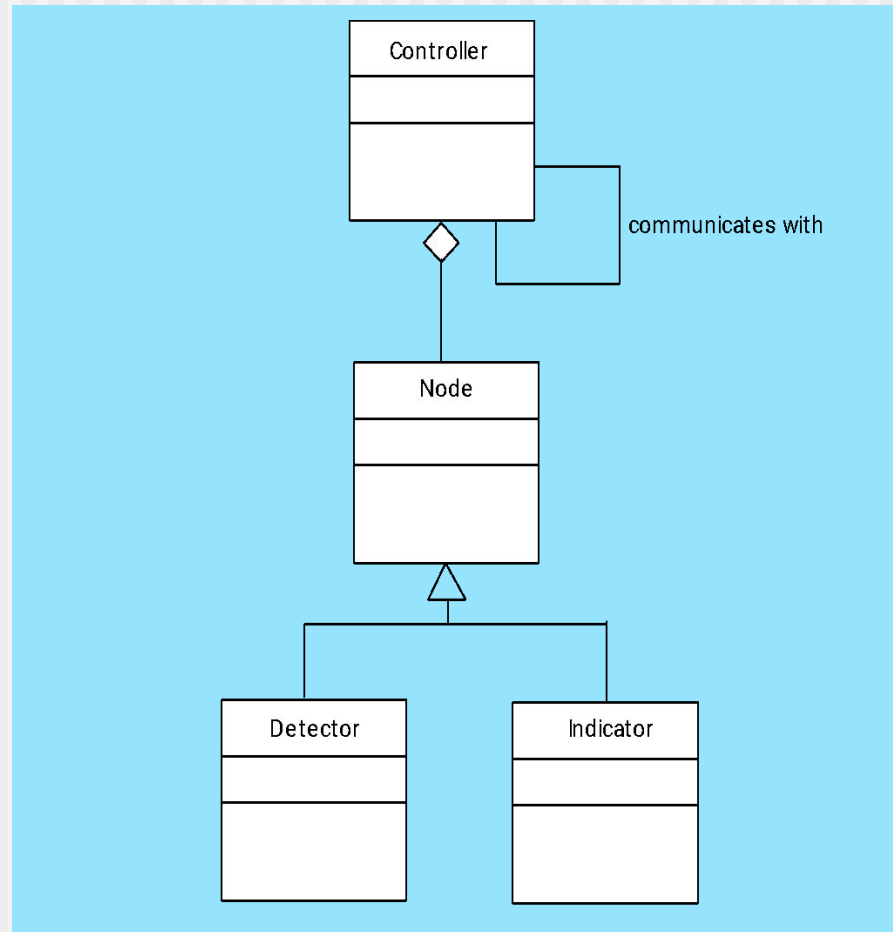
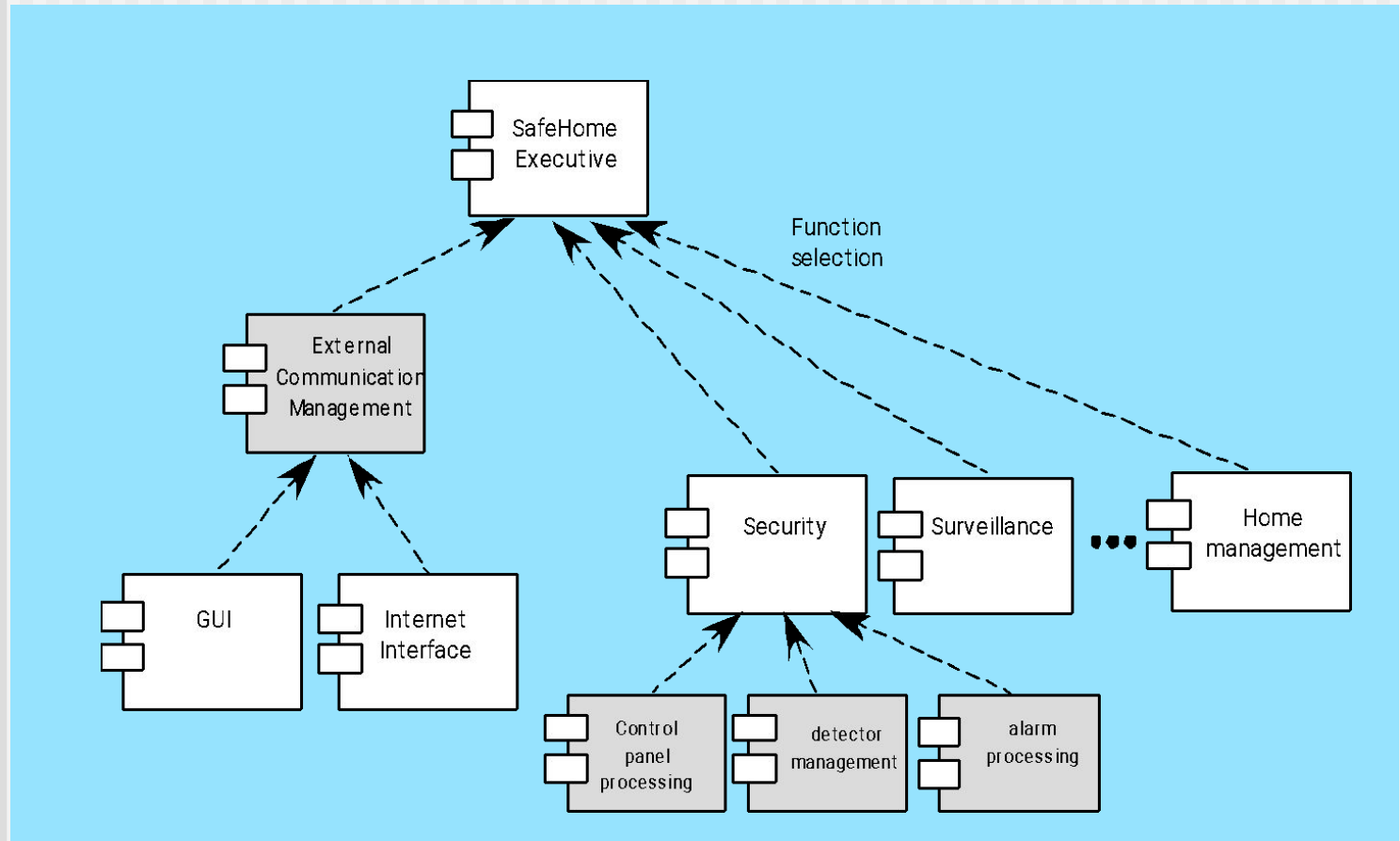
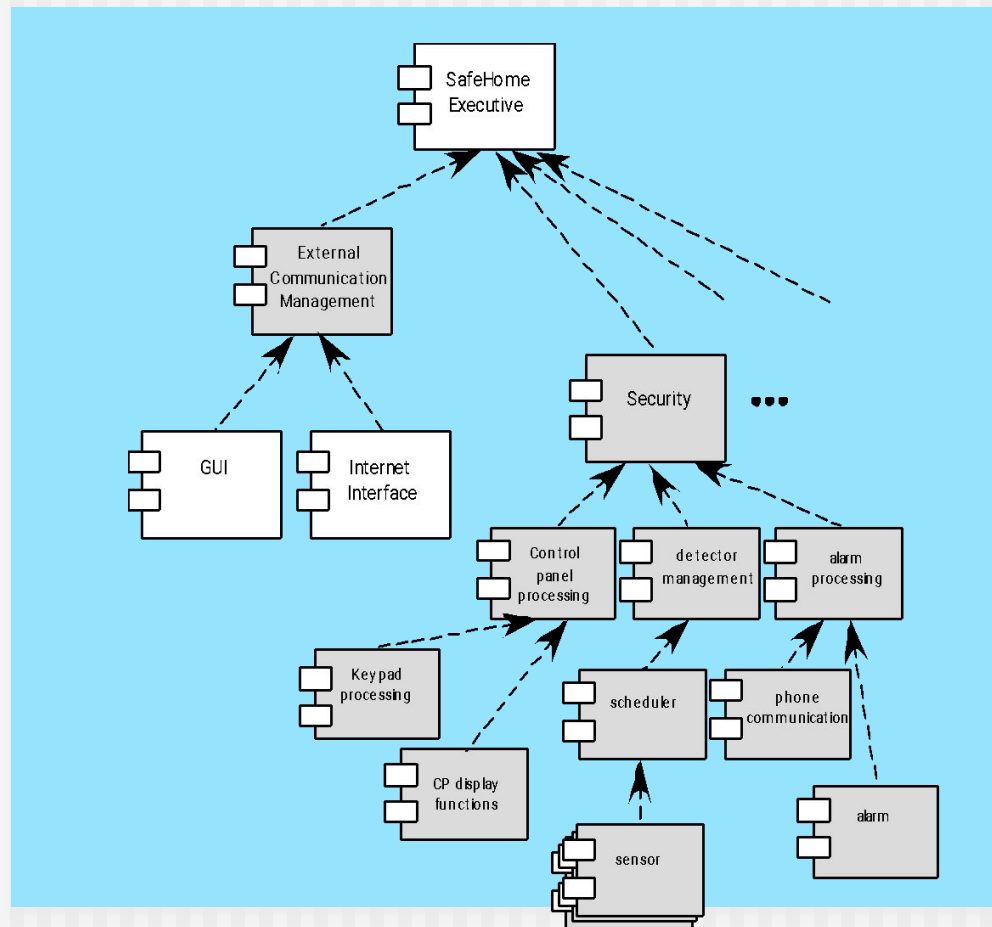


Figure 10.7 UML relationships for SafeHome security function archetypes (adapted from [BOS00])

Component Structure



Refined Component Structure



These slides are designed to accompany *Software Engineering: A Practitioner's Approach*, 7/e (McGraw-Hill, 2009). Slides copyright 2009 by Roger Pressman.

Exercise

- What is architecture ? Explain any two architectural styles with figure
- Explain with figure Architectural context diagram