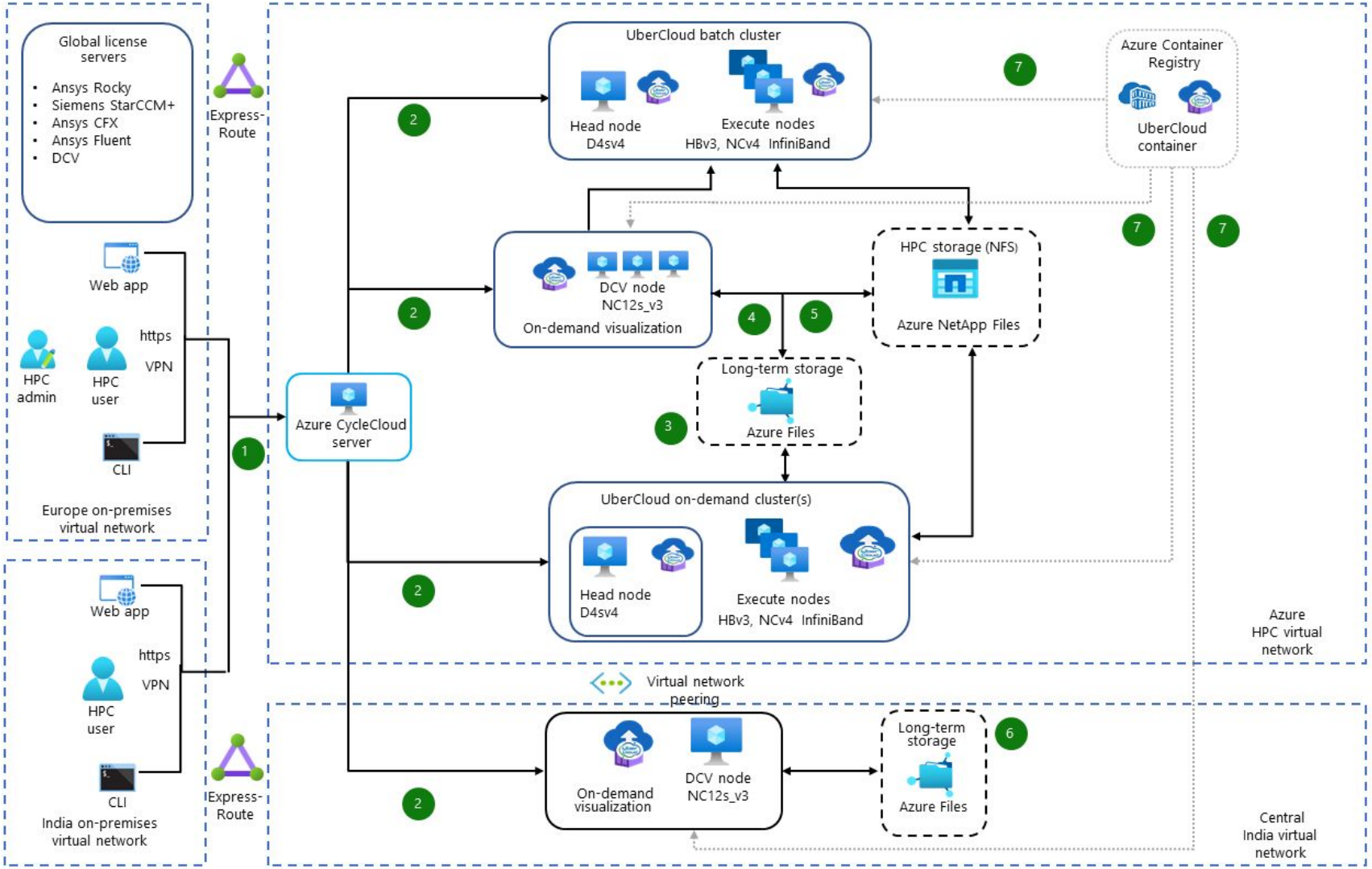
CAPOL207: Basics of Software Engineering

W10L1: Architectural Design Concepts

Dr. A. Joy Christy
School of Computing
SASTRA Deemed to be University

Outline

- Software Architecture
- Architectural Genres
- Architectural Styles
- Architectural Design
- Assessing Architectural Design
- Architectural Mapping Using Data Flow


Global license servers
Ansys Rocky
Siemens StarCCM+
Ansys CFX
Ansys Fluent
DCV
Express-Route
Web app
https
VPN
HPC admin
HPC user
CLI
Europe on-premises virtual network
Azure CycleCloud server
UberCloud batch cluster
Head node D4sv4
Execute nodes HBv3, NCv4 InfiniBand
Azure Container Registry
UberCloud container
DCV node NC12s_v3
On-demand visualization
HPC storage (NFS)
Azure NetApp Files
Long-term storage
Azure Files
UberCloud on-demand cluster(s)
Head node D4sv4
Execute nodes HBv3, NCv4 InfiniBand
Azure HPC virtual network
Virtual network peering
Web app
https
VPN
HPC user
CLI
India on-premises virtual network
Express-Route
On-demand visualization
DCV node NC12s_v3
Long-term storage
Azure Files
Central India virtual network
1
2
3
4
5
6
7

Architecture

- The architecture of the building has many attributes
  - Overall shape
  - Physical structure
  - Integration of various components of the building
  - Fitting to the environment
  - Meshing with other buildings
  - Satisfying the owner
  - Visual impact
- Architecture is an art

- Architecture
  - Thousands of decisions, both big and small
  - is the structure or structures of the system
    - Comprises
      - Software components
      - Externally visible properties of the components
      - Relationship among the components
  - Is not an operational software, but enables to
    - Analyze the effectiveness of design with respect to software requirements
    - Make changes in design
    - Reduce construction risks

- design is an instance of an architecture
- For instance
    - Design of a client-server architecture may be different from
        - Java Platform or Microsoft Platform
- Many designs may be created based on one architecture

Why Is Architecture Important?

- enabler for communication between all parties
- highlights early design decisions
- constitutes a relatively small, intellectually graspable model of
    - How the system is structured?
    - How its components work together?

Architectural Descriptions

- architecture
  - means different things to different people
- stakeholders see architecture
  - from different viewpoints driven from sets of concerns
- architectural description
  - is set of work products that reflects different views of the system

Architectural Descriptions

- Stakeholders' of a major office building: owner and structural steel fabricator
  - Owner: aesthetic, sufficient space and infrastructure
    - Architect: develop a description by addressing the owner's concerns
      - 3D drawing of the building (aesthetic view)
      - 2D floor plans (space infrastructure)

- Structural steel fabricator: needs information about structural steel that support building
  - Architect: develop description with specialized drawings for structural steel skeleton
    - I-beams
    - I-beams dimensions
    - Connectivity
    - materials

- Tyree and Akerman write
  - Architectural description of software-based system:
    - Developers want
      - clear, decisive guidance to proceed to design
    - Customers want
      - to understand the environment
      - Assurances that the architecture meet business needs
    - Other architects want
      - To understand the salient key aspects of the architecture
    - The ‘wants’ reflects in different view – each view is a different viewpoint

Recommended Practice for Architectural Description of Software-Intensive Systems - IEEE

1) to establish a conceptual framework and vocabulary for use during the design of software architecture

2) to provide detailed guidelines for representing an architectural description

3) to encourage sound architectural design practices

- The IEEE
  - architectural description (AD)
    - collection of products to document an architecture
    - Representing multiple views
    - Each view is a representation of whole system from stakeholder's perspective

Architectural Genres

- Implies specific category within the overall software domain
- each category, may encounter a number of subcategories
- Buildings encounter following general styles

  Houses Apartments Office buildings

  Industrial buildings warehouses

- Each style is described using set of predictable patterns

- architectural genres for software-based systems

Artificial Intelligence	Entertainment & Sports	Communications
Financial	Games	Government
Industrial	Legal	Medical
Military	Scientific	Transportation

- each genre represents a unique challenge

- Artificial Intelligence
  - Simulate or augment
    - Human cognition
    - Locomotion
    - Organic processes
- Commercial and nonprofit
  - Fundamental to operation of a business enterprise
- Games
  - Provide an entertainment experience for individuals or groups

- Communications
  - Provide infrastructure for
    - Transferring data
    - Managing data
    - Connecting users
    - Presenting data at the edge of infrastructure
- Content Authoring
  - Create or manipulate textual or multimedia artifacts
- Devices
  - Interact with physical world to provide service to an individual

- Entertainment and Sports
  - Manage public events
  - Provide group entertainment experience
- Financial
  - Provides infrastructure for
    - transferring and managing money
    - Security
- Industrial
  - Simulate /control physical processes

- Government
  - Support the conduct and operations of
    - Local
    - State
    - Federal
    - Global
    - Political entity
- Legal
  - Support the legal industry

- Medical
  - Diagnose
  - heal
  - Contribute to medical research
- Military
  - Consultation
  - Communications
  - Command
  - Control
  - Intelligence defense

- Operating systems
  - Sit above hardware to provide basic software services
- Platforms
  - Sit above operating systems to provide advanced services
- Scientific
  - Used for scientific research and applications
- Tools
  - Used to develop other systems

- Transportation
  - Control water
  - Ground
  - Air
  - Space vehicles
- Utilities
  - Interact with other software to provide service

- Game systems, sometimes called immersive interactive applications
  - require the computation of intensive algorithms
  - sophisticated computer graphics
  - streaming multimedia data sources
  - real-time interactivity via conventional and unconventional inputs
  - variety of other specialized concerns
- SAI (Software Architecture for Immersipresence) is a new model designed to address gaming concern

Architectural Styles

Cape Cod

Raised Ranch

A- type

Centre hall colonial

Source: https://www.architecturaldesigns.com/house-plans/center-hall-colonial-house-plan-44045td

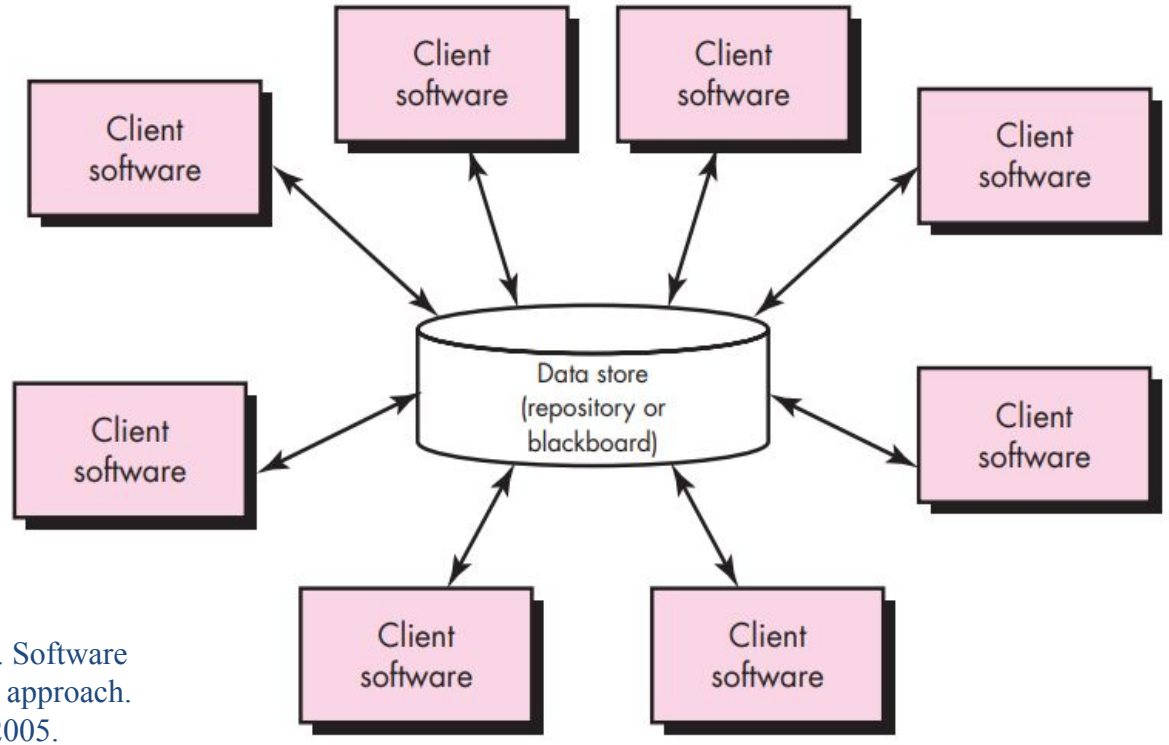
Architectural Styles

- Architectural style
  - Is a
    - template for construction
    - Design of an entire system
  - Establishes a structure for all components of the system
- Software exhibits one of many architectural styles

- Each style encompasses:
  - Set of components:
    - functions required by the system (database, computational modules)
  - Set of connectors:
    - enables communication, coordination and cooperation among components
  - Constraints:
    - Define how components can be integrated to form the system
  - Semantic models:
    - enable understanding overall properties of a system

Data-centered architectures

- A data store resides at the center and is accessed frequently by other components to
  - Update
  - Add
  - Delete
  - Modify data



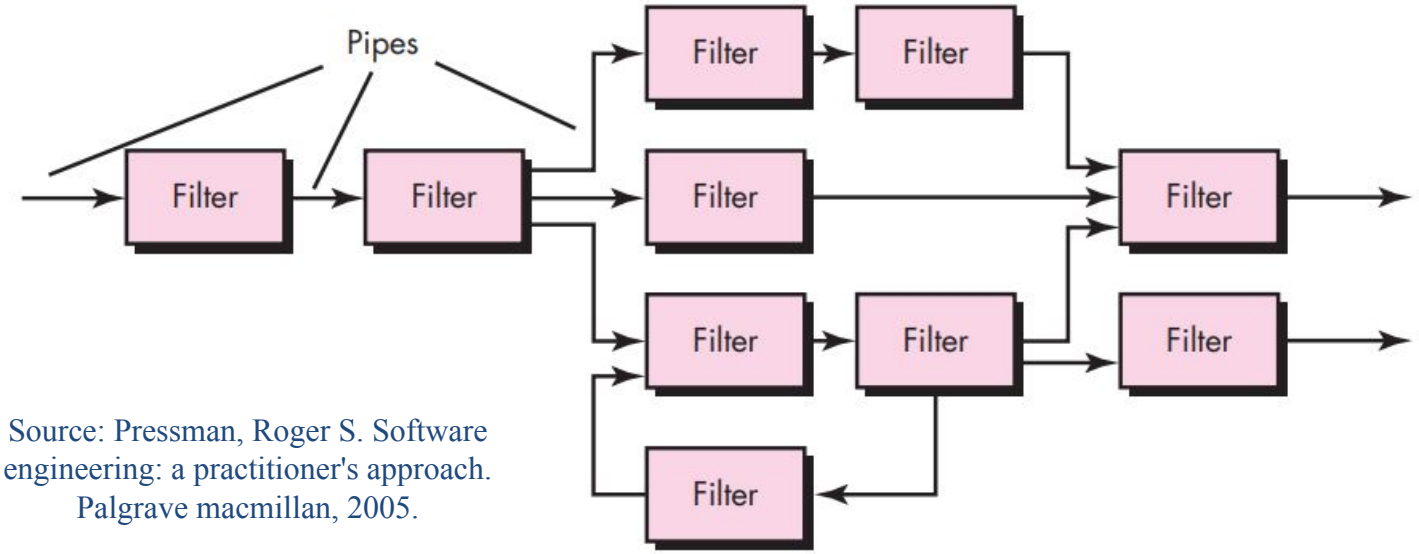
Source: Pressman, Roger S. Software engineering: a practitioner's approach. Palgrave macmillan, 2005.

- Client software accesses a central repository
- In some cases the data repository is passive
    - Client access data independent of any changes of other client software
- variation on this approach transforms the repository into a “blackboard”
    - Sends notifications on changes
- promote integrability
    - Components can be changed/added without concern about other clients

Data-flow architectures

- Transforms i/p data through series of computational/manipulative components into o/p data
- Pipe-and-filter pattern
  - Set of components (filters)
  - Pipes(connects filters to transmit data from one component to the next)
  - filter
    - works independently of those components upstream and downstream
    - does not require knowledge of the workings of its neighboring filter
  - Batch sequential
    - Data flow degenerates into single line of transformation

Data-flow architectures

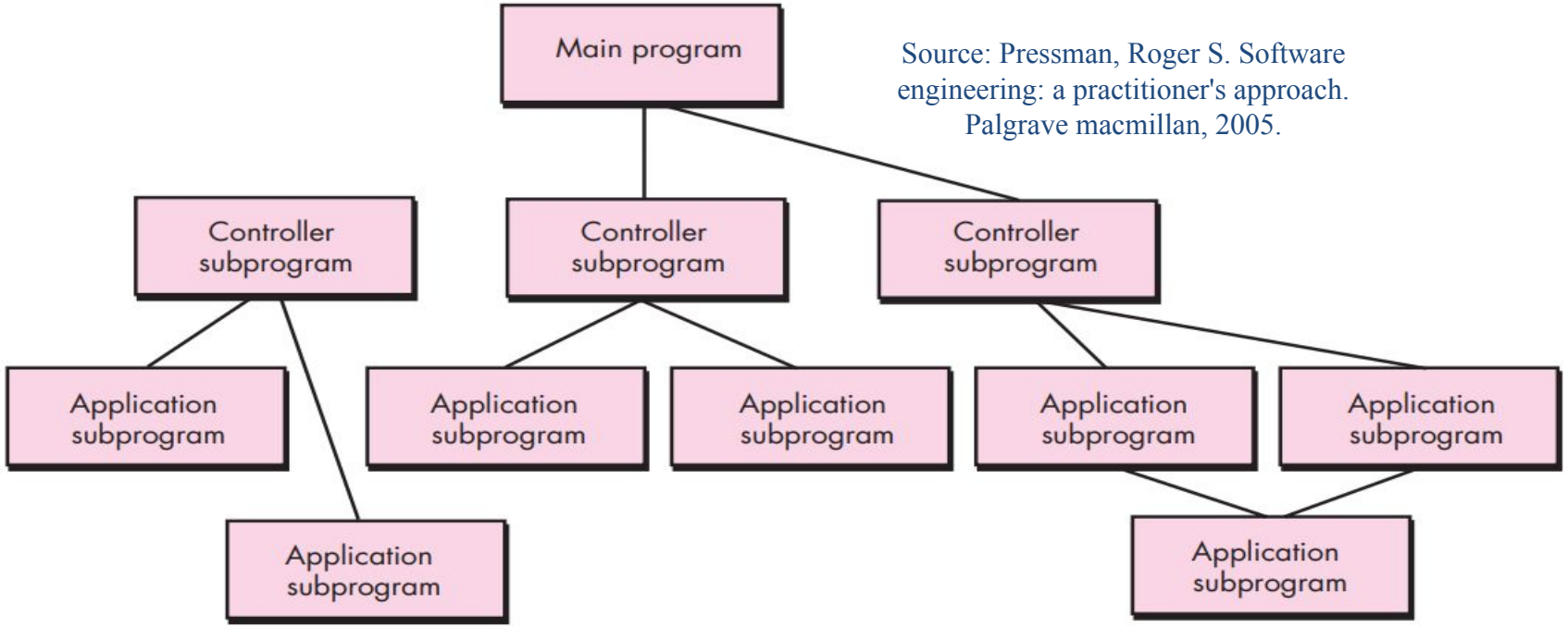


Source: Pressman, Roger S. Software engineering: a practitioner's approach. Palgrave macmillan, 2005.

Pipes and filters

Call and return architectures

- Main program/subprogram architectures
- Remote procedure call architectures
  - Main program/subprogram architecture are distributed across multiple computers on a network

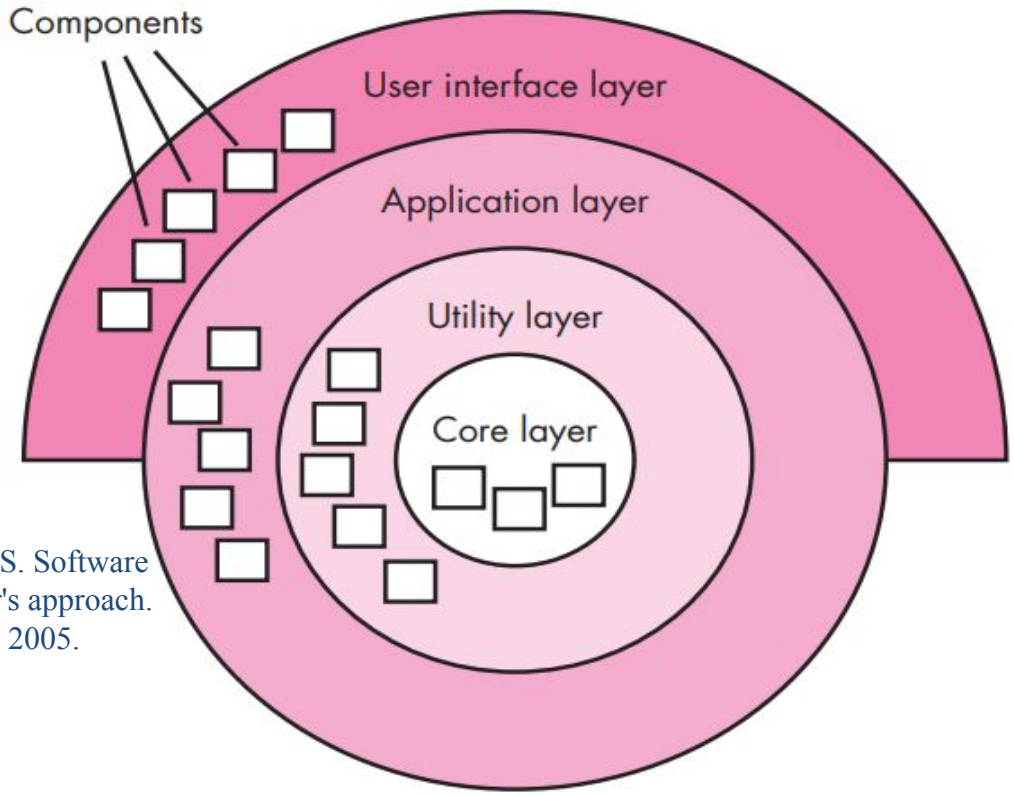


Source: Pressman, Roger S. Software engineering: a practitioner's approach. Palgrave macmillan, 2005.

Object-oriented architectures

- Ensure
  - encapsulate data and the operations that manipulate the data
  - communication and coordination between components

Layered architectures



Source: Pressman, Roger S. Software engineering: a practitioner's approach. Palgrave macmillan, 2005.

- Layers define operations progressively become closer to machine instruction set
- Outer layer:
  - Components service user interface operations
- Inner layer:
  - Components perform operating system interfacing
- Intermediate layers
  - Components provide
    - utility services
    - Application software functions

- Data-centered
- Data Flow
- Call-Return
- Object oriented
- Layered architecture
    - Are only a small subset of those available
- Combination of styles may be designed and evaluated
- Layered architecture is appropriate for most systems
- Layered style can be combined with data-centered architecture in many DB applications

Architectural Patterns

- Addresses an application-specific problem
  - Within a specific context
  - Under a set of limitation and constraints
- Proposes solution that serves as the basis for architectural design

Organization and Refinement

- Control Transfer
  - How is control managed within the architecture?
  - Does a distinct control hierarchy exist?
  - How do components transfer control within the system?
  - How is control shared among components?
  - Is control synchronized?

- Data Transfer
  - How data are communicated between components?
  - Is the flow of data continuous?
  - What is the mode of data transfer?
  - Do data components exist? (ex: blackboard or repository)
  - How do functional components interact with data components?
  - Are data components passive or active?
  - How do data and control interact within the system?

Architectural Design

- the software should be put into context
  - the design should
    - define the external entities the s/w interact
      - the nature of interaction
    - identify set of architectural archetypes
      - archetype
        - is an abstraction (similar to a class)
        - represents one element of system behavior

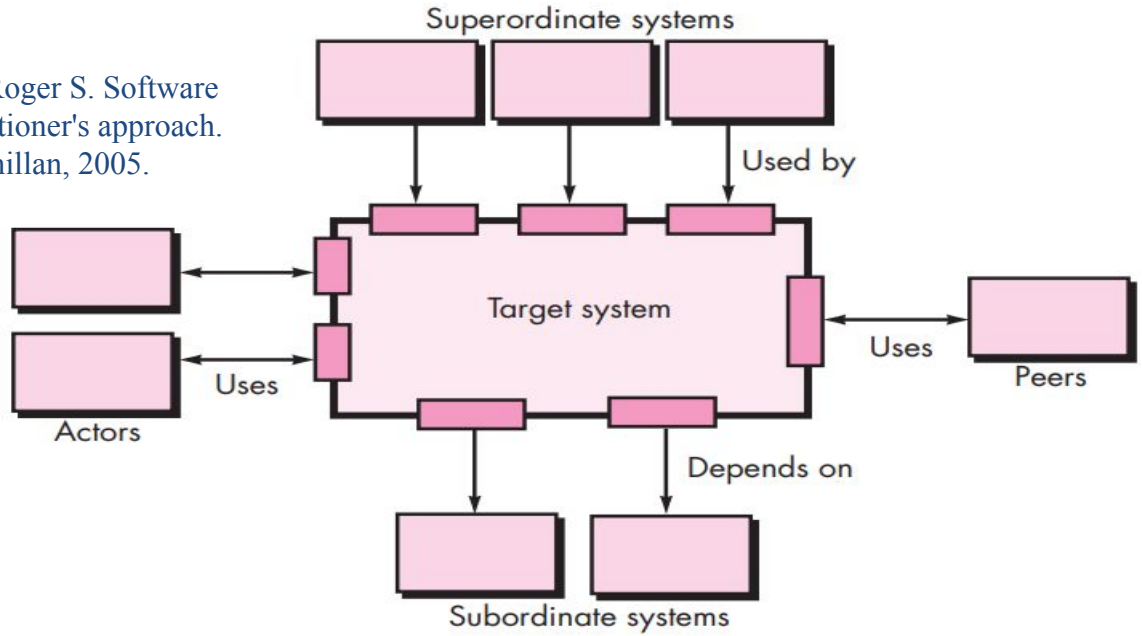
Representing the system in Context

- architect uses

  Architectural context diagram

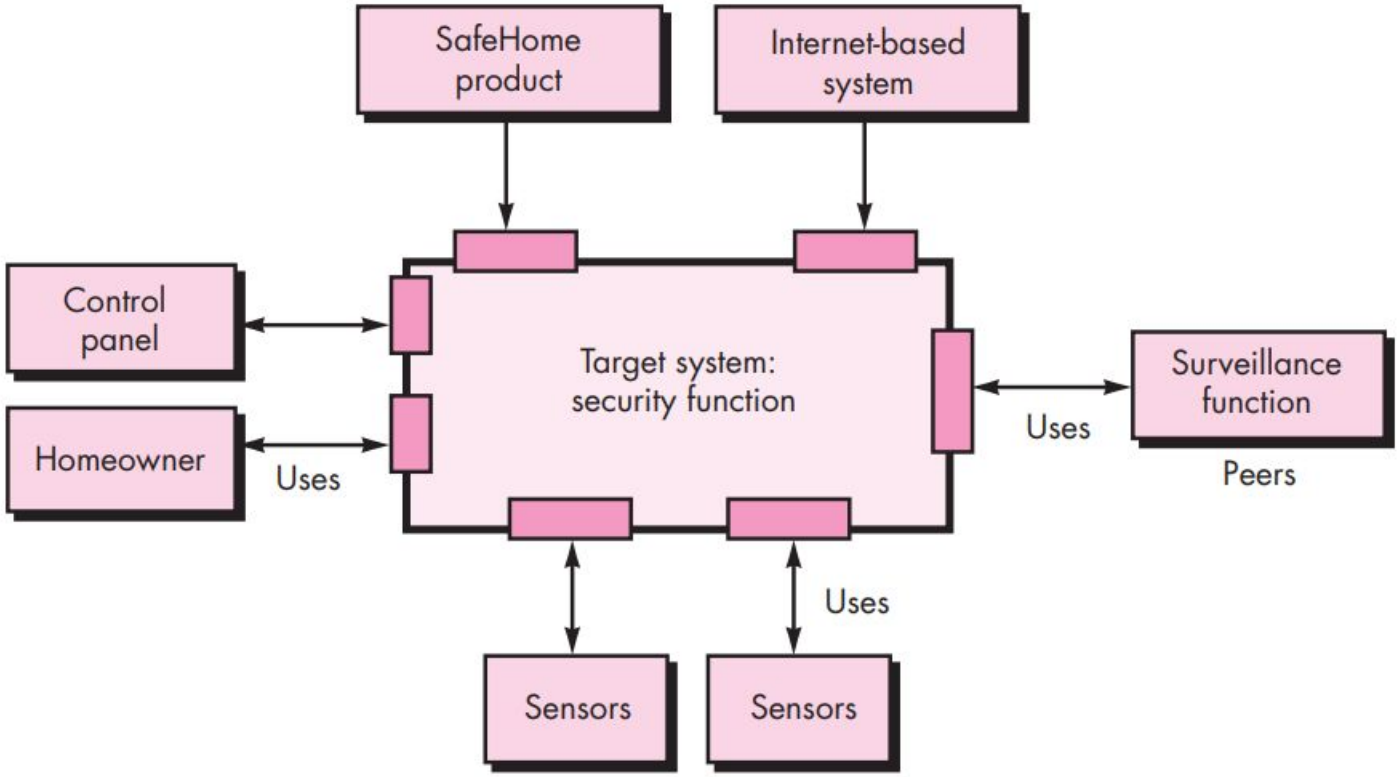
  - to model software interaction with entities external to its boundaries

Source: Pressman, Roger S. Software engineering: a practitioner's approach. Palgrave macmillan, 2005.



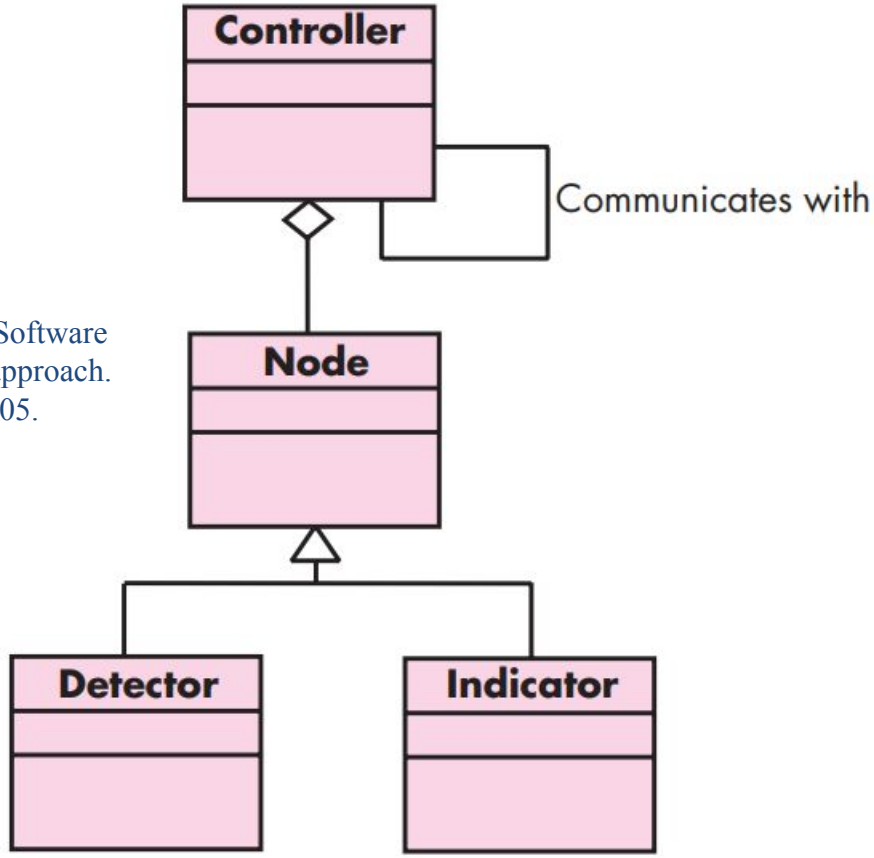
- target system
  - the system for which an architectural design is to be developed
- superordinate systems
  - systems that uses target system as part of higher-level processing scheme
- subordinate systems
  - systems used by the target system
  - provides necessary data or processing to complete target system functionality
- peer-level systems
  - systems that interact on a peer-to-peer basis
    - information produced and consumed by peers and the target system
- actors
  - entities that interact with target system (people, devices)
  - produce or consume information necessary for requisite processing

Source: Pressman, Roger S. Software engineering: a practitioner's approach. Palgrave macmillan, 2005.



Defining Archetypes

- archetype is a class or pattern
  - represents core abstraction
  - helps in designing architecture for the target system
- the target system architecture is composed of archetypes
- represents stable elements of the architecture
- derived by examining analysis classes



Source: Pressman, Roger S. Software engineering: a practitioner's approach. Palgrave macmillan, 2005.

UML relationships for SafeHome security function archetypes

- Node:
  - Represents a cohesive collection of input and output elements
  - Comprises of
    - Various sensors (input)
    - Variety of alarm(output)

- Detector:
  - Abstraction of all sensing equipment that feeds information to target system

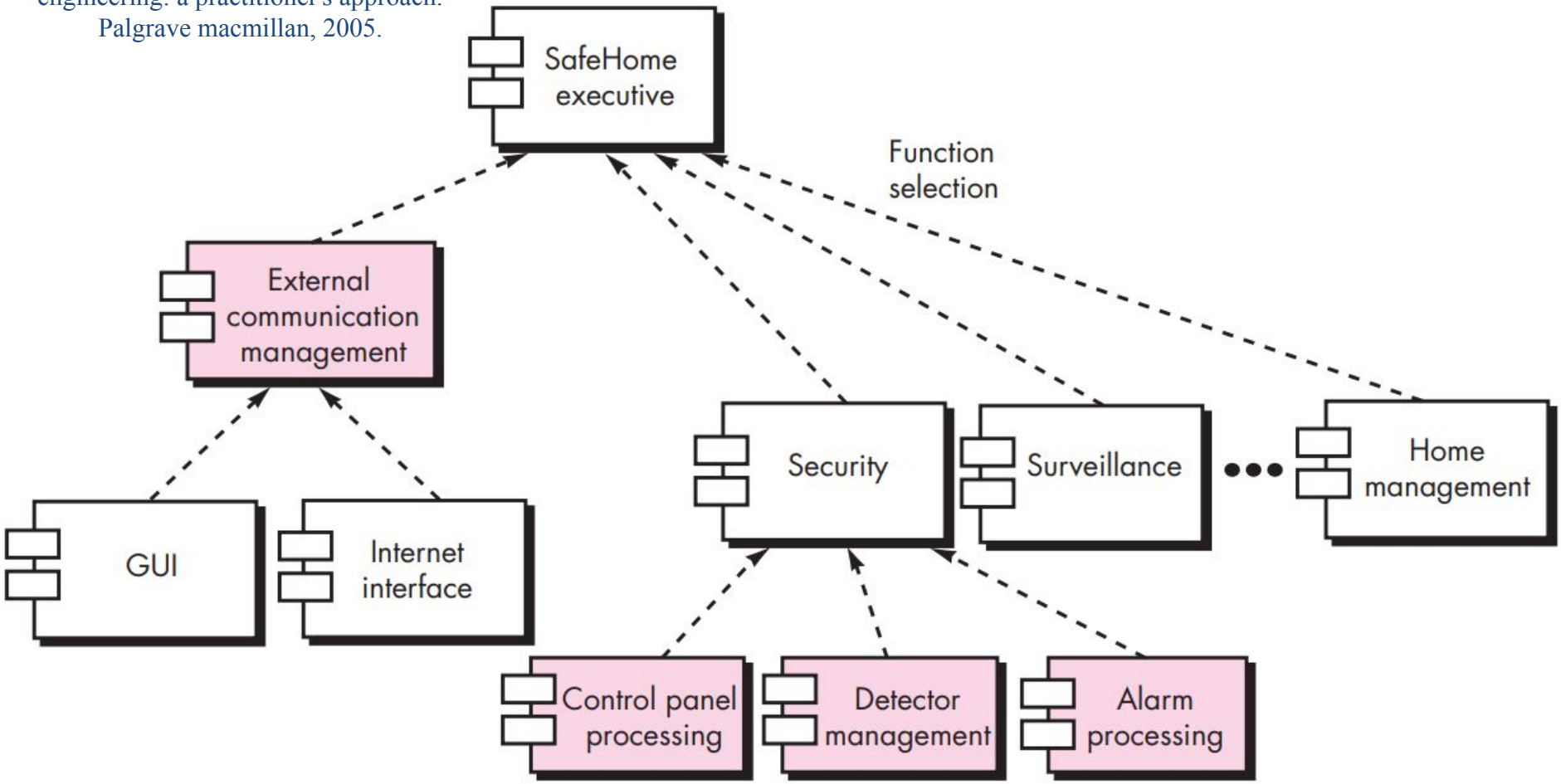
- Indicator:
  - Abstraction of all mechanisms for indicating an alarm condition
  - Eg: alarm siren
  - flashing lights
  - bell

- Controller:
  - Abstraction depicting the mechanism for arming and disarming a node

Refining the Architecture into Components

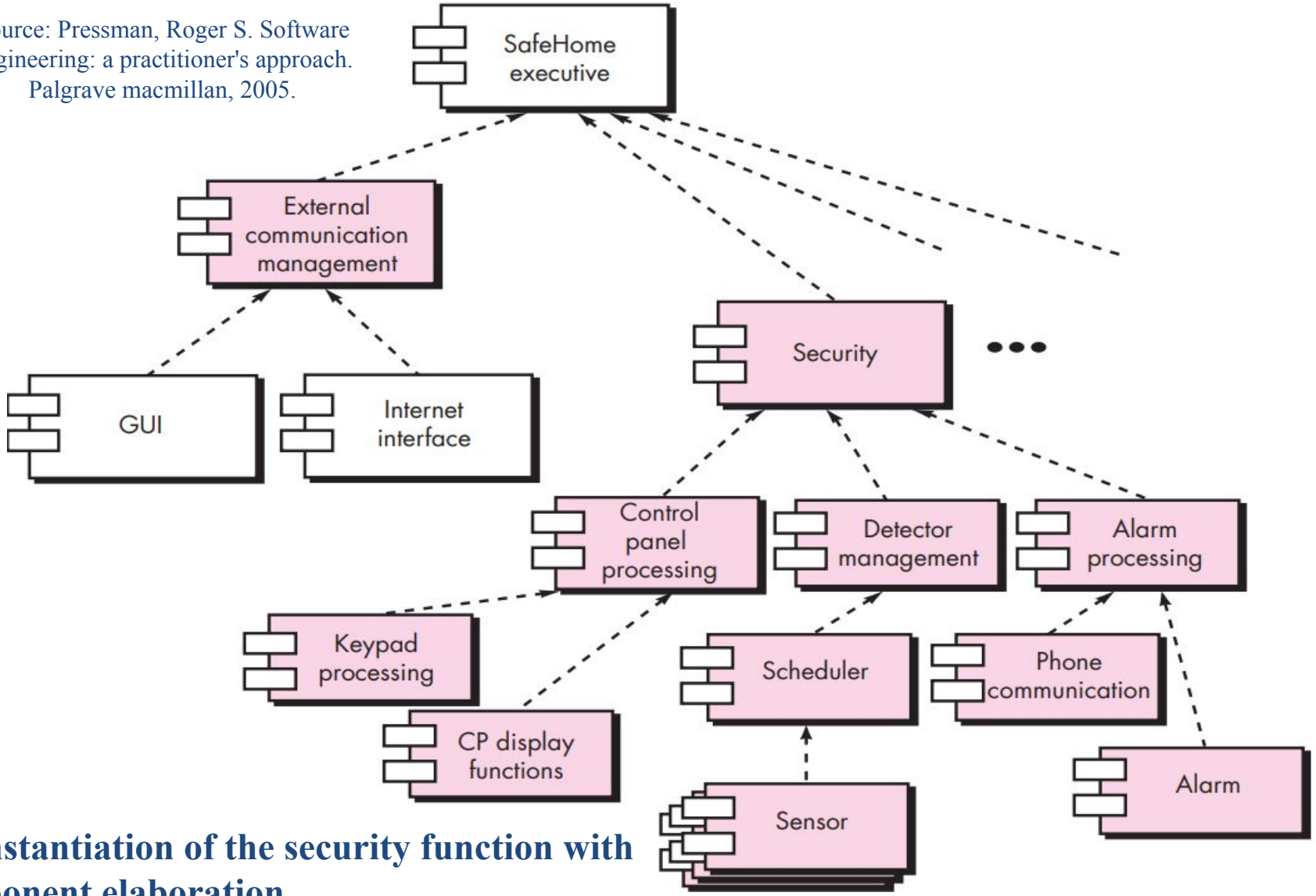
- represent entities within
  - application domain
  - infrastructure domain

Source: Pressman, Roger S. Software engineering: a practitioner's approach. Palgrave macmillan, 2005.



Overall architectural structure for SafeHome with top-level components

Source: Pressman, Roger S. Software engineering: a practitioner's approach. Palgrave macmillan, 2005.



An instantiation of the security function with component elaboration

Assessing Alternative Architectural Designs

- Architecture trade-off analysis method
  - is proposed by The Software Engineering Institute (SEI)
  - Establishes an iterative evaluation process for software architectures

- collect scenarios
  - Set of use cases is developed to represent the system from user's point of view
- Elicit requirements, constraints and environment description
  - Information collected during requirements engineering is used
- Describe the architectural styles/patterns to address the scenarios and requirements
  - Architectural style may be described using one of the following
    - Module view:
      - analysis of work assignment with components
      - Degree to which information hiding has been achieved
    - Process view:
      - Analysis of system performance
    - Data flow view:
      - Analysis of the degree to which architecture meets functional requirements

- Evaluate quality attributes by considering each attribute in isolation
  - Reliability
  - Performance
  - Security
  - Maintainability etc.,

- Identify the sensitivity of quality attributes to various architectural attributes
  - Accomplished by making small changes in the architecture and determine
    - How sensitive a quality attribute is to change (eg. Performance)
    - Any attributes significantly affected by variation in the architecture are termed as
      - Sensitivity points

- Critique candidate architectures using sensitivity analysis
  - Finding trade-off points for each sensitive points
  - Eg:
    - Performance of client-server architecture is highly sensitive to number of servers
    - Trade-off point- number of servers

THANK YOU

Reference:

Pressman, Roger S. Software engineering: a practitioner's approach. Palgrave macmillan, 2005.