SE 329 – Software Project Management

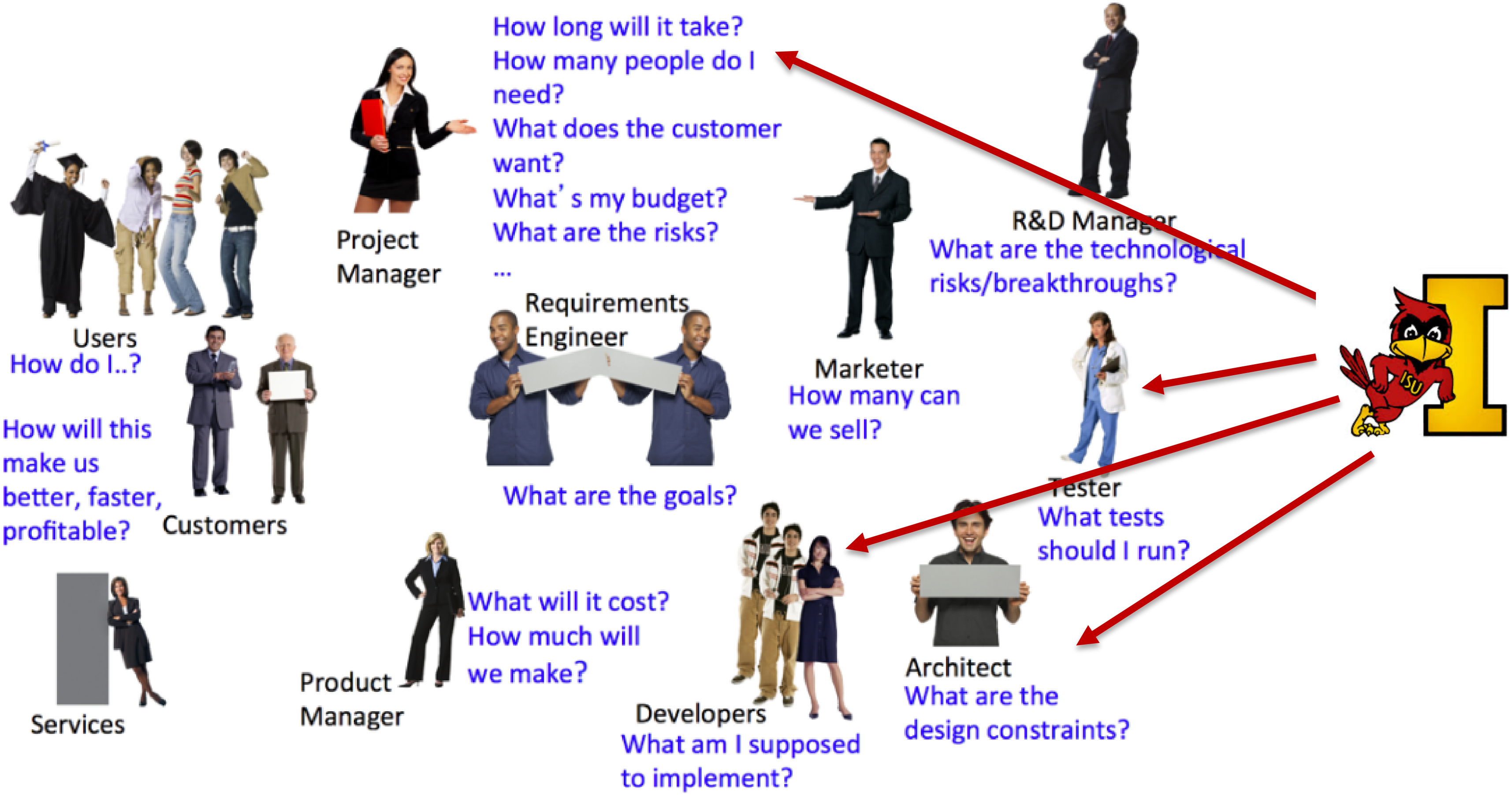
Software architecture and project management

Lotfi ben Othmane

Why Should We Care?

- Can software architecture be useful for managing projects?
- Are there conflicts between software architecture and project management?

Stakeholder Questions



Project Manager

How long will it take?
How many people do I need?
What does the customer want?
What’s my budget?
What are the risks?
…

R&D Manager
What are the technological risks/breakthroughs?

Users
How do I..?

How will this make us better, faster, profitable?

Customers

Requirements Engineer

What are the goals?

Marketer
How many can we sell?

Tester
What tests should I run?

Services

Product Manager

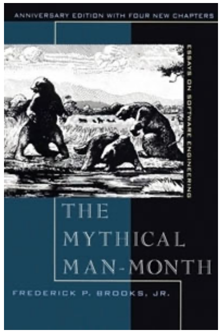
What will it cost?
How much will we make?

Developers
What am I supposed to implement?

Architect
What are the design constraints?

"It is a very humbling experience to make a multimillion-dollar mistake, but it is also very memorable…."

(Fred Brooks - "Mythical Man-Month" p.47)


ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS
ESSAYS ON SOFTWARE ENGINEERING
THE MYTHICAL MAN-MONTH
FREDERICK P. BROOKS, JR.

What is Software Architecture?

Software architecture is the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution. (IEEE 1471-2000)

What is Software Architecture?

- The architecture is an abstract model of the software
  - Architecture defines major components
  - Architecture defines component relationships (structures) and interactions
- Architecture is the first design artifact where a system's quality attributes are addressed

What is Software Architecture?

Includes:

- Architecture represents the set of earliest design decisions
- It defines the rationale behind the components and the structure

Architecture Requirements

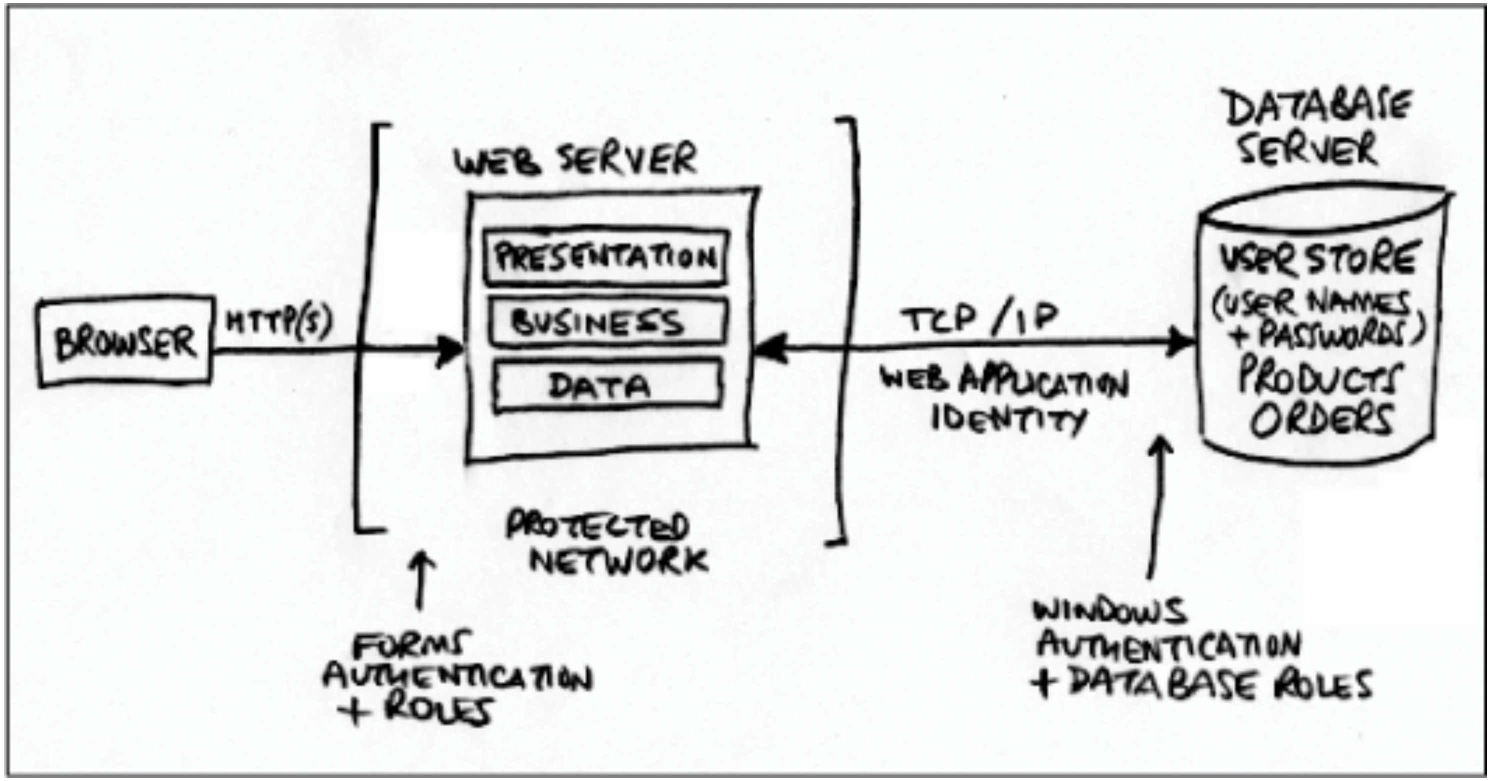
- Architectures exist to build systems that satisfy requirements
- An architecturally significant requirement (ASR) is a requirement that drive the architecture
- ASR imply a decision on how the components of system must or should behave

Sources of Architecture Significant Requirements

- Requirements documents
- Interviewing stakeholders
- Understanding business goals

Interviewing stakeholders

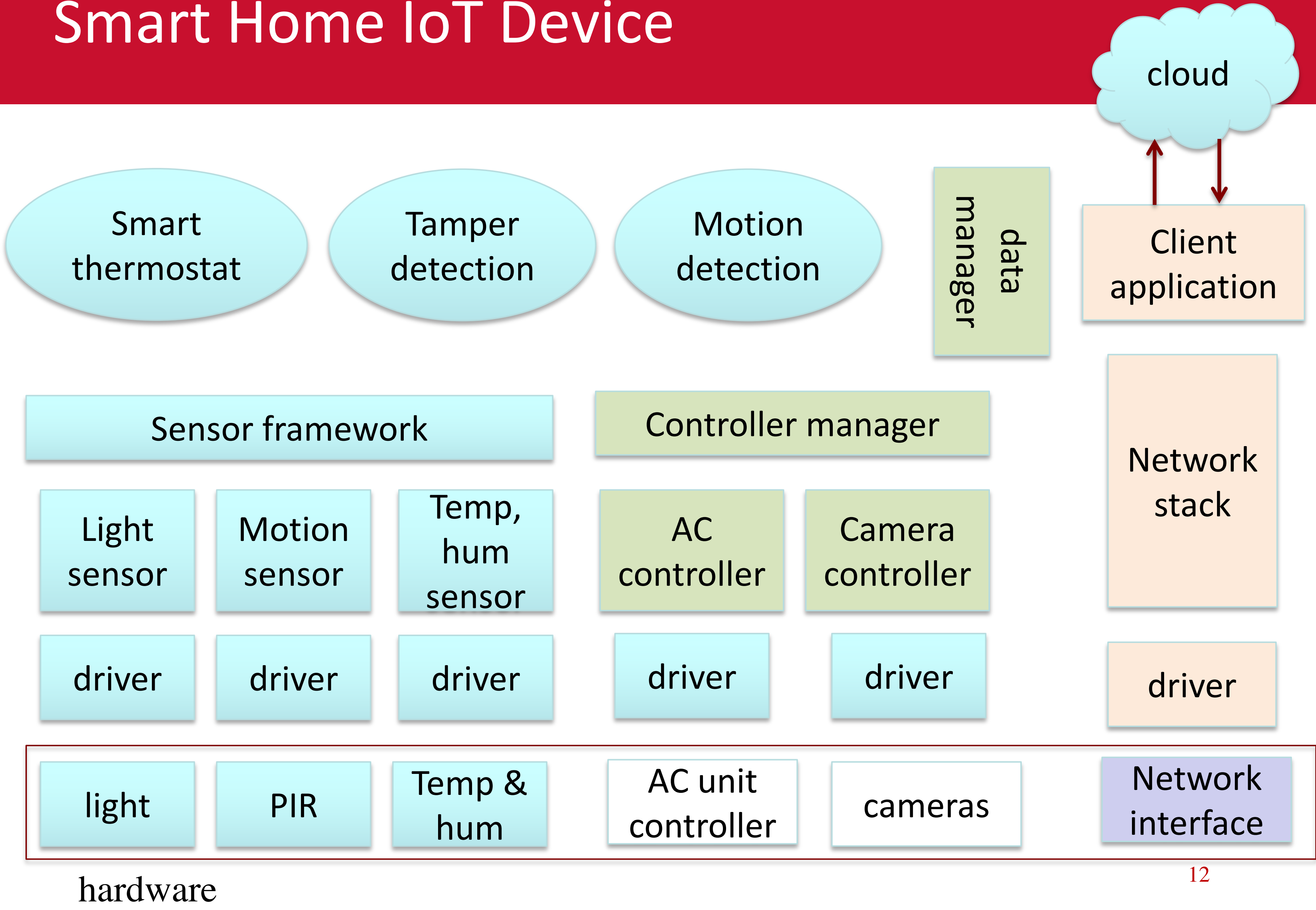
- Whiteboard Your Architecture - Show the major constraints and decisions in order to frame and start conversations



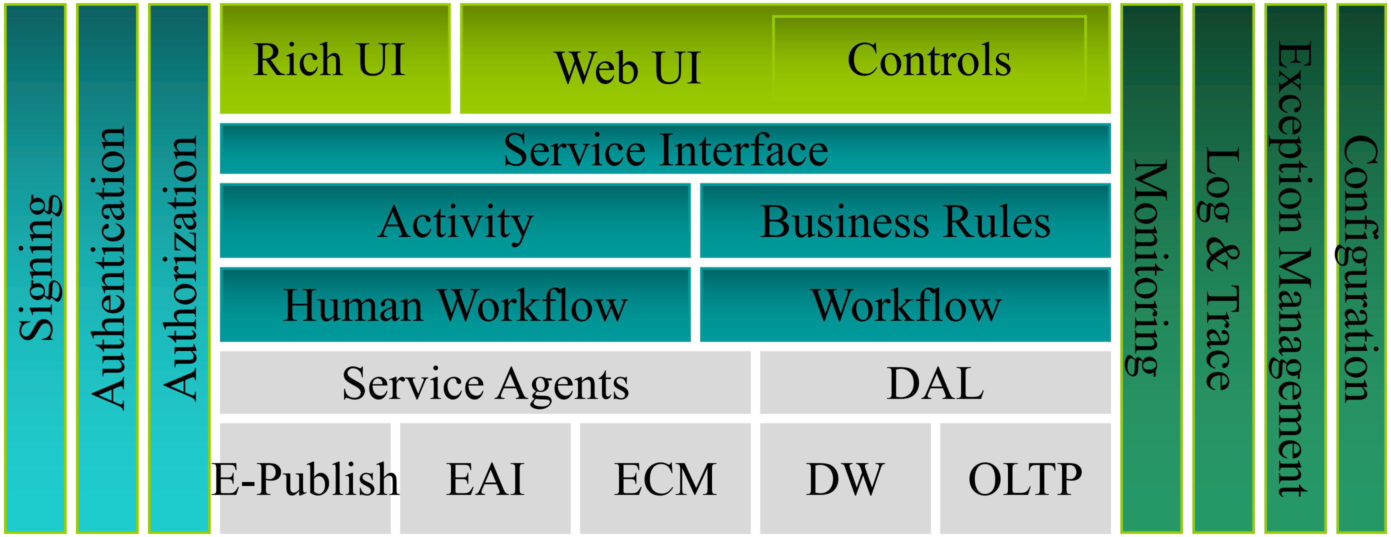
What is a Software Component

A software component is a piece of software offering a predefined service and is able to communicate with other components.

Smart Home IoT Device



“Non Standard” - Block Diagrams



What is Software Architecture?

- The architecture is an abstract model of the software
  - Architecture defines major components
  - Architecture defines component relationships (structures) and interactions

- Architecture is the first design artifact where a system's quality attributes are addressed

How Do We Get the Components?

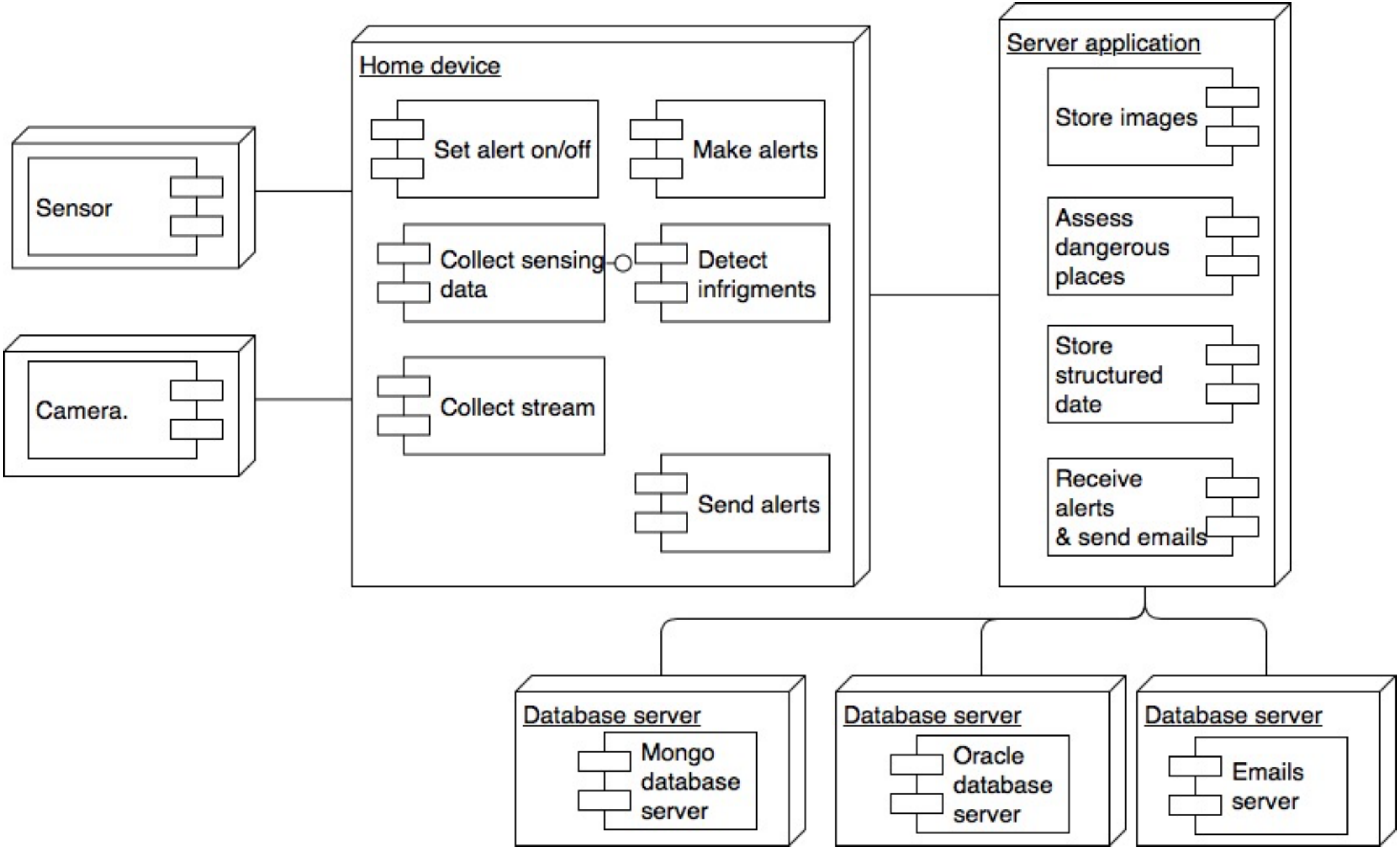
- Allocation of responsibilities, which involves
  - Identification of the main functions of the software
  - Deciding on how these functions are allocated at non-runtime and runtime
- Strategies
  - Functional decomposition
  - Modeling real-word objects
  - Group functionalities based on specific logic

Example

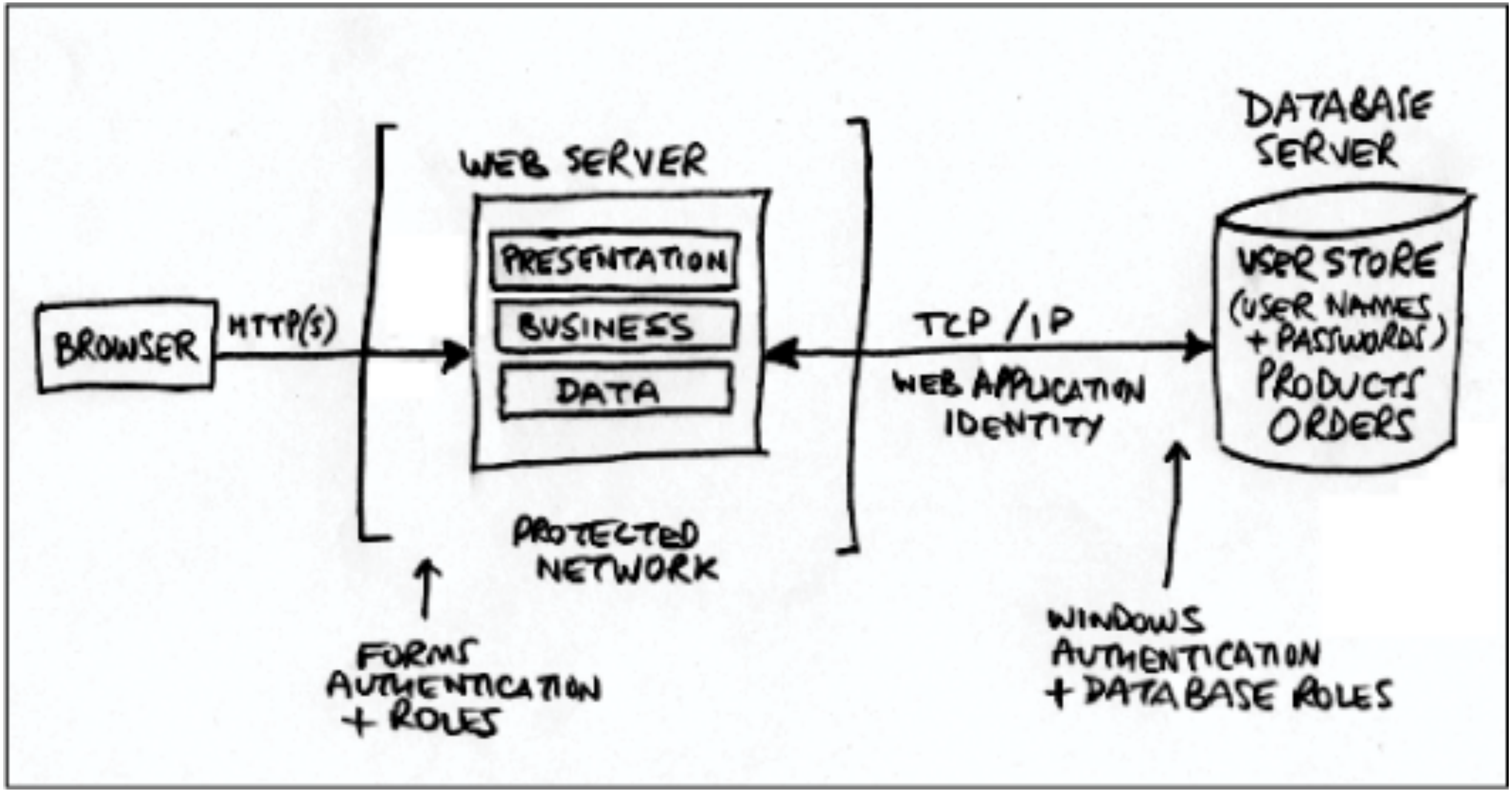
Assume that you are asked to design a home security system that uses cameras and motion sensors to detect movements and distress button to send alters to a central system.

Design an architecture sketch for a home security system.

Architecture Components



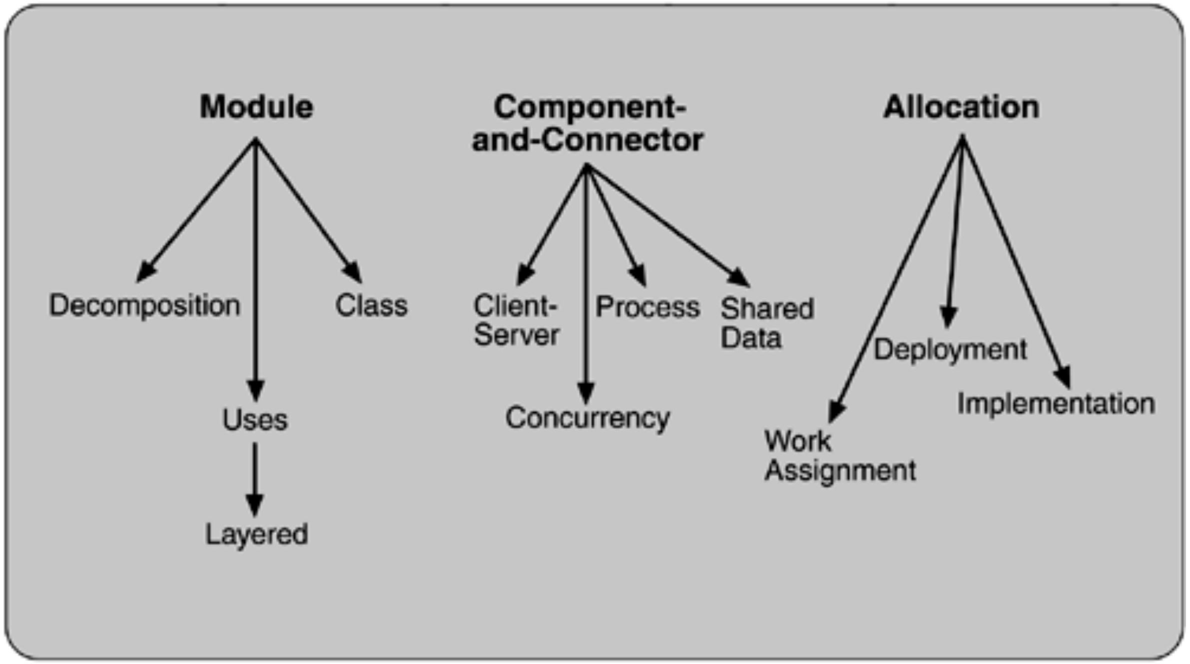
Architecture Components



Architecture Structures?

Is it enough to know the components?

- Elements
- Relationships between these elements



What is Software Architecture?

- The architecture is an abstract model of the software
- Architecture defines major components
- Architecture defines component relationships (structures) and interactions
- Architecture is the first design artifact where a system’s quality attributes are addressed

What Is Quality Attributes?

- Quality attributes (QAs) are the factors that affect the run-time behavior, the system design, and the user experience
- QAs represent areas of concern that have the potential for application wide impact across layers and tiers
- The common ones are:
  1. Availability
  2. Interoperability
  3. Modifiability
  4. Performance
  5. Security
  6. Testability
  7. usability

How to Formulate QAs?

Performance scenario for submitting applications

1. Stimulus: Regular application (event)
2. Stimulus source: User
3. Response: Save the application, process the payment, return a confirmation to the user
4. Response measure: < 5 sec
5. Environment: normal and overload conditions
6. Artifact: submission, payment, save file, save database

How to Address QAs?

- Performance -- The goal is to generate a response to an event arriving at the system within given time.

- Example of tactics for performance requirements

1. Increase resources
2. Maintain multiple copies of computations
3. Maintain multiple copies of data

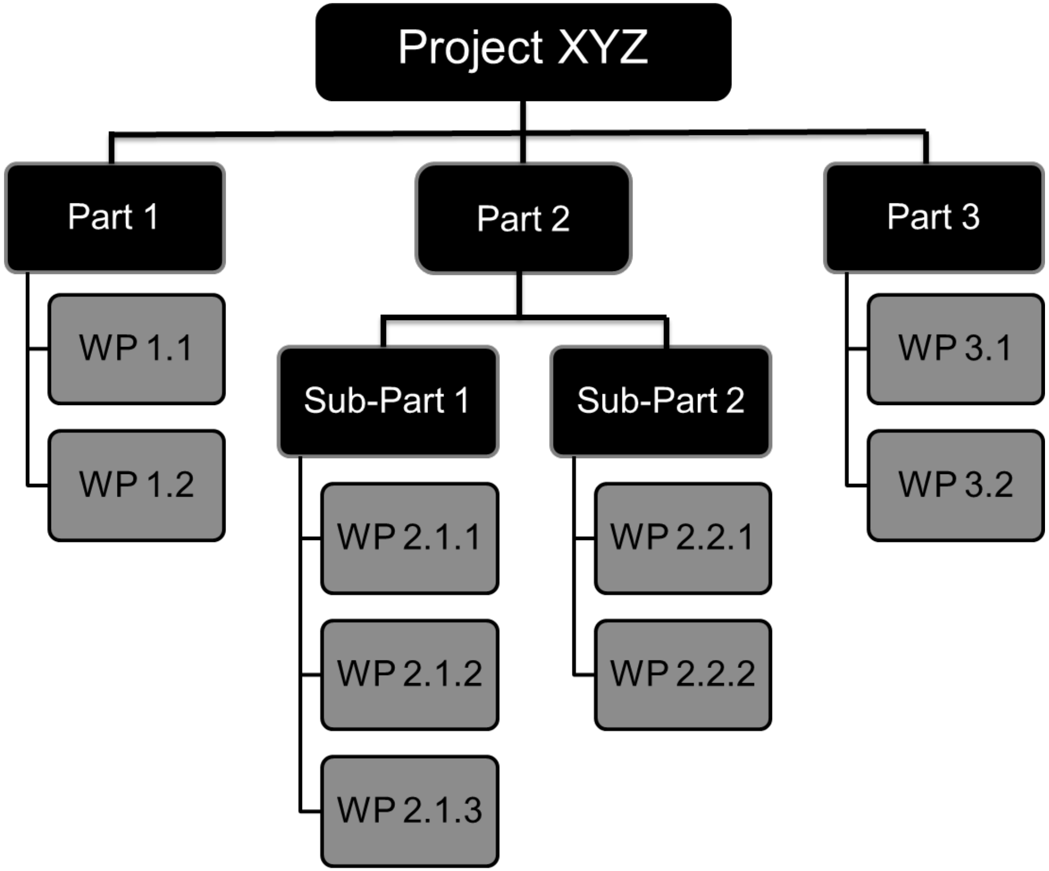
Use of Architecture for Proposal

- Each software shall solve a problem - The solution lays in the architecture of the software
- Stakeholders need to agree on the vision supported by the architecture
- It helps to believe that the solution can be achieved with a predictable cost and schedule

Use of Architecture for Estimation

- The modules could be used for the breakdown structure

- The breakdown structure is used for the project estimation



Architecture and Project Risks

- Recall that quality attributes are the factors that affect the run-time behavior, system design, and user experience
- The major technical risks are often related to the architecture
  - Risks related to choice of technologies
  - Risks related to the satisfaction of quality attributes
    - Think about performance, security, etc.
- Project success is often dependent upon the architectural approach

Are There Conflicts Between PM and Architect?

- PM and architect have different concerns, so developers…
- PM wants
  - Balance cost, time, and quality
  - “efficient” architecture
- Architect wants
  - Solution that addresses the functional and non functional requirements
  - Could be extended easily
- Example of conflicts
  - PMs are often reserved to new technologies and ideas (source of RISK)
  - Architects often embrace new technologies and ideas (opportunities)

Self Check

- What is software architecture?
- How can software architecture be useful for managing projects?
- Are there conflicts between software architecture and project management?

Any Question?

Next lecture topic: Time and cost planning