Architecture Drivers

Lotfi ben Othmane

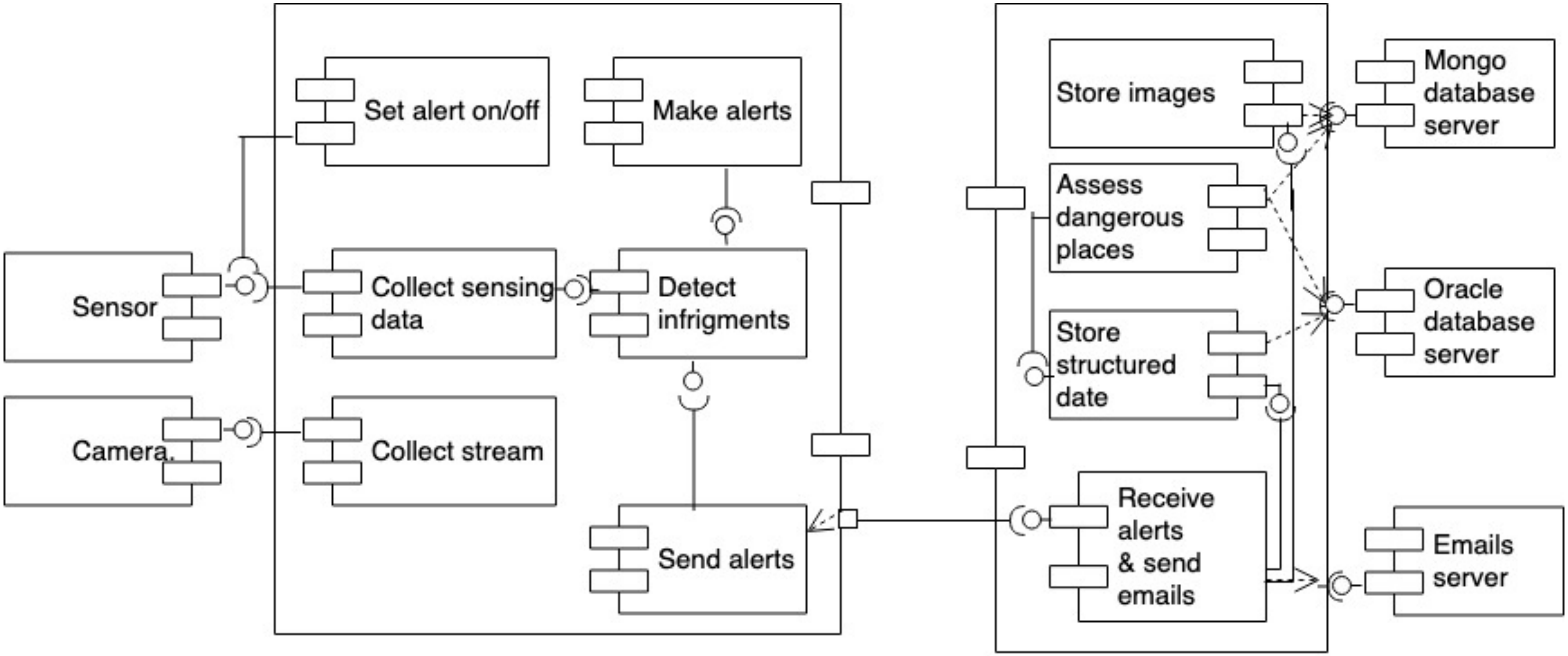
Functional Requirements

Develop a use case diagram for the home security system.

The main functionalities are:

1. Set a system on and off
2. Collect data using sensors and camera
3. Analyze the data to detect intruders
4. Make alerts
5. Send alerts to the central system
6. Keep data to assess dangerous neighborhoods
7. Store structured data into an Oracle database
8. Store images in a MongoDB database

Home Security System – Component Diagram

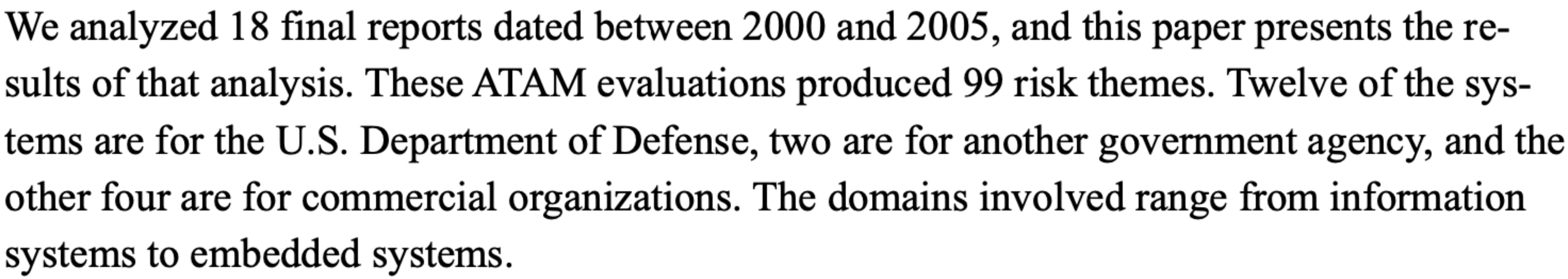


But …

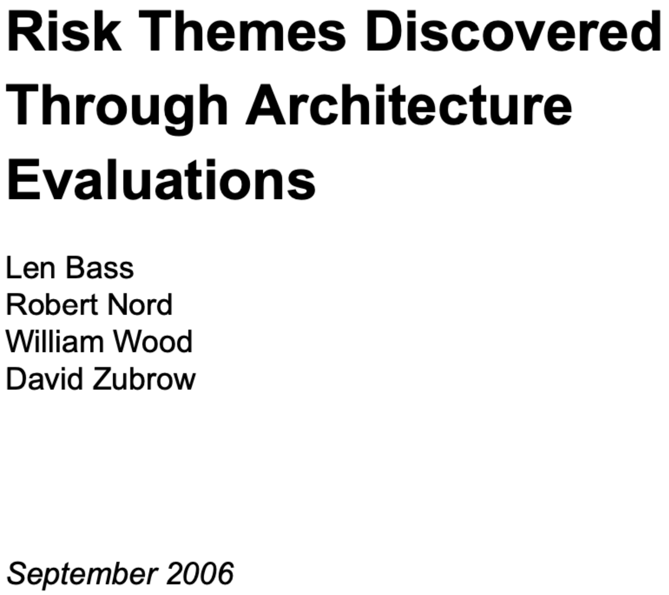
We did not check if any of the following requirements apply

1. Support diverse devices
2. Support an important number of devices
3. Support intermittent communication
4. Support frequent and small messages
5. Support limited processing capabilities of the devices
6. Support important size of data
7. Enforce time-critical Actions

Well…



We analyzed 18 final reports dated between 2000 and 2005, and this paper presents the results of that analysis. These ATAM evaluations produced 99 risk themes. Twelve of the systems are for the U.S. Department of Defense, two are for another government agency, and the other four are for commercial organizations. The domains involved range from information systems to embedded systems.

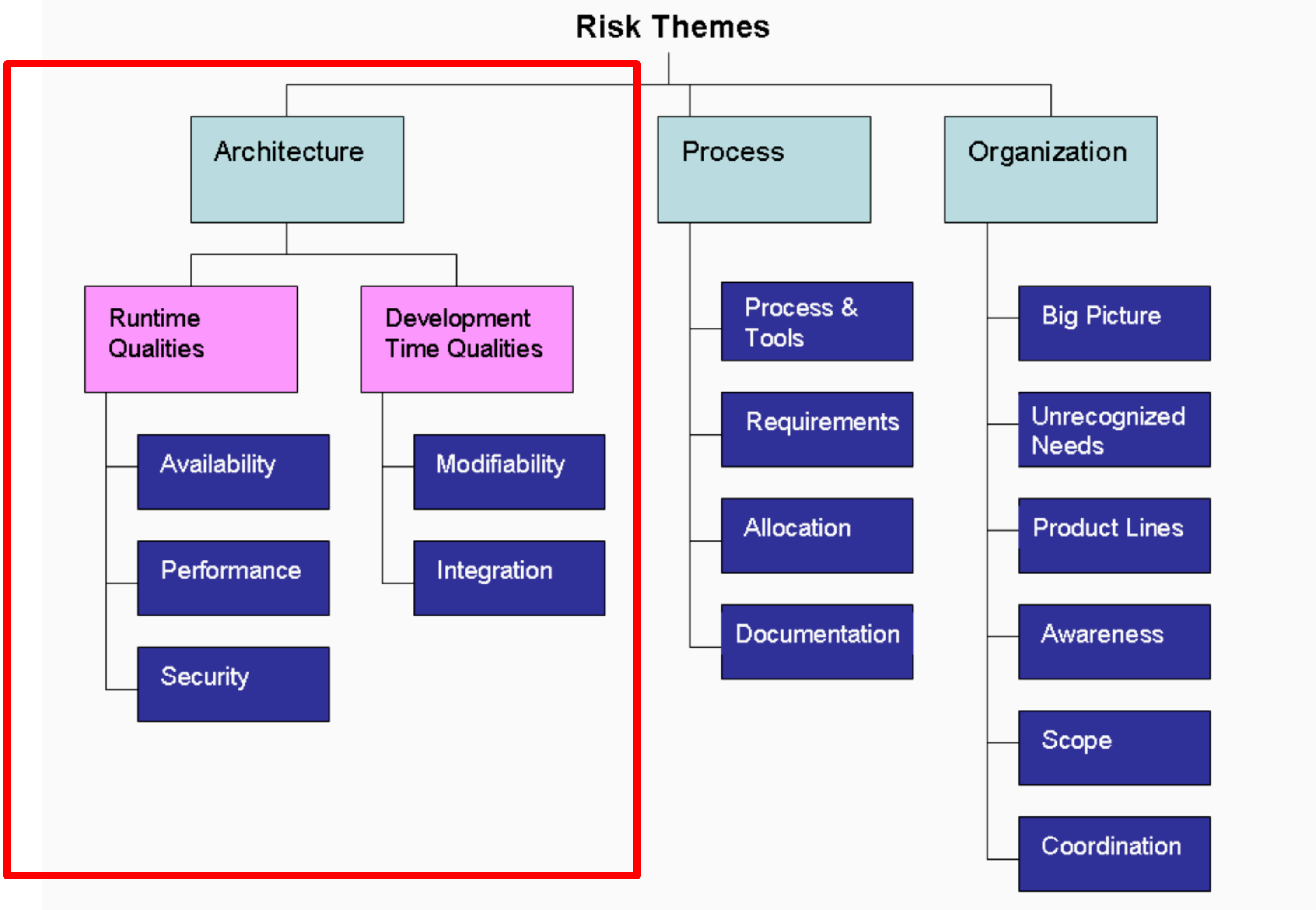


Risk Themes Discovered Through Architecture Evaluations

Len Bass
Robert Nord
William Wood
David Zubrow

September 2006

Architecture Risks Themes



Risk Themes - Runtime Qualities

- Availability or reliability
  1. having a single point of failure
  2. not including availability mechanisms
  3. using infrastructure that does not support availability mechanisms
- Performance:
  1. not knowing performance requirements
  2. not performing any performance modeling or prototyping
  3. unfamiliarity with infrastructure choices
  4. not using known performance mechanisms

Risk Themes - Runtime Qualities

- Security
  1. unknown requirements
  2. not using known mechanisms to support security goals

Risk Themes - Development Time Qualities

- Modifiability
  1. allocating functionality in a way that jeopardizes portability
  2. supporting the addition and deletion of different devices
  3. lack of attention to potential growth paths
  4. unknown requirements
- Integration
  1. problems with migrating legacy systems
  2. not using known integration mechanisms
  3. lack of uniformity in key areas

Architecture requirements are not **only** functional requirements

Architecture Drivers

The architecture drivers define the what and why about the architecture

- They include:
  1. Primary functionality
  2. Design purpose
  3. Quality attributes
  4. Architectural concerns
  5. Architectural constraints

Architecture Design

Architecture drivers

Design purposes

Quality attributes

Primary functionalities

Architectural concerns

Constraints

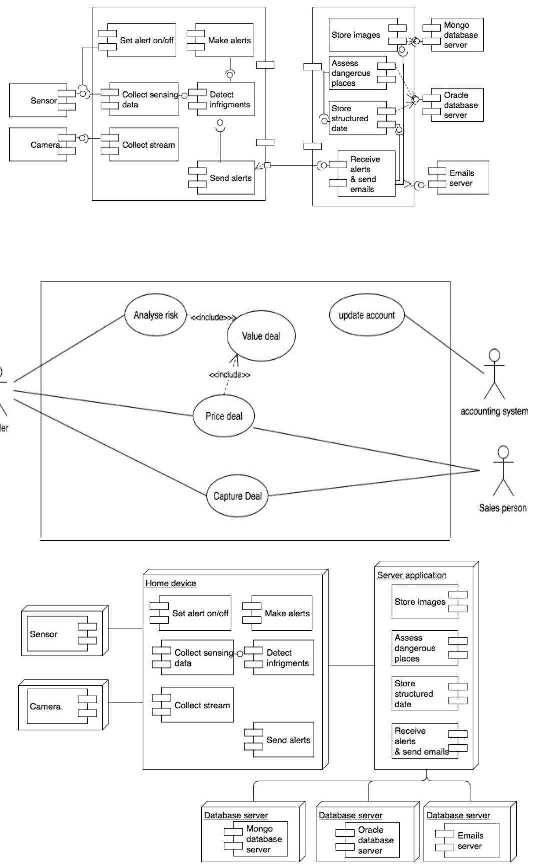
Design concepts

Selects and instantiates

Candidate design decisions

Architect

Architecture structures



FCAPS As An Example

Existing system

- Servers use Network Time Protocol (NTP) to synchronize times
- Servers clustered in regions
- Each region has a stratum 1, equipped with hardware for precise time
- The servers support Simple Network Management Protocol (SNMP)
- The system conforms to FCAPS

Goal:

- Extend the system to support VOIP
- Add servers that support other management protocols than SNMP

Design Purpose

- Purpose: Why are you doing the design?
  - This involves the business goals that the organizations wants to address.

- Examples:
  1. Design an architecture for a proposal
  2. Design for a prototype
  3. Design for development

Example of Design Purpose

Existing system

- Servers use Network Time Protocol (NTP) to synchronize times
- Servers clustered in regions
- Each region has a stratum 1, equipped with hardware for precise time
- The servers support Simple Network Management Protocol (SNMP)
- The system conforms to FCAPS

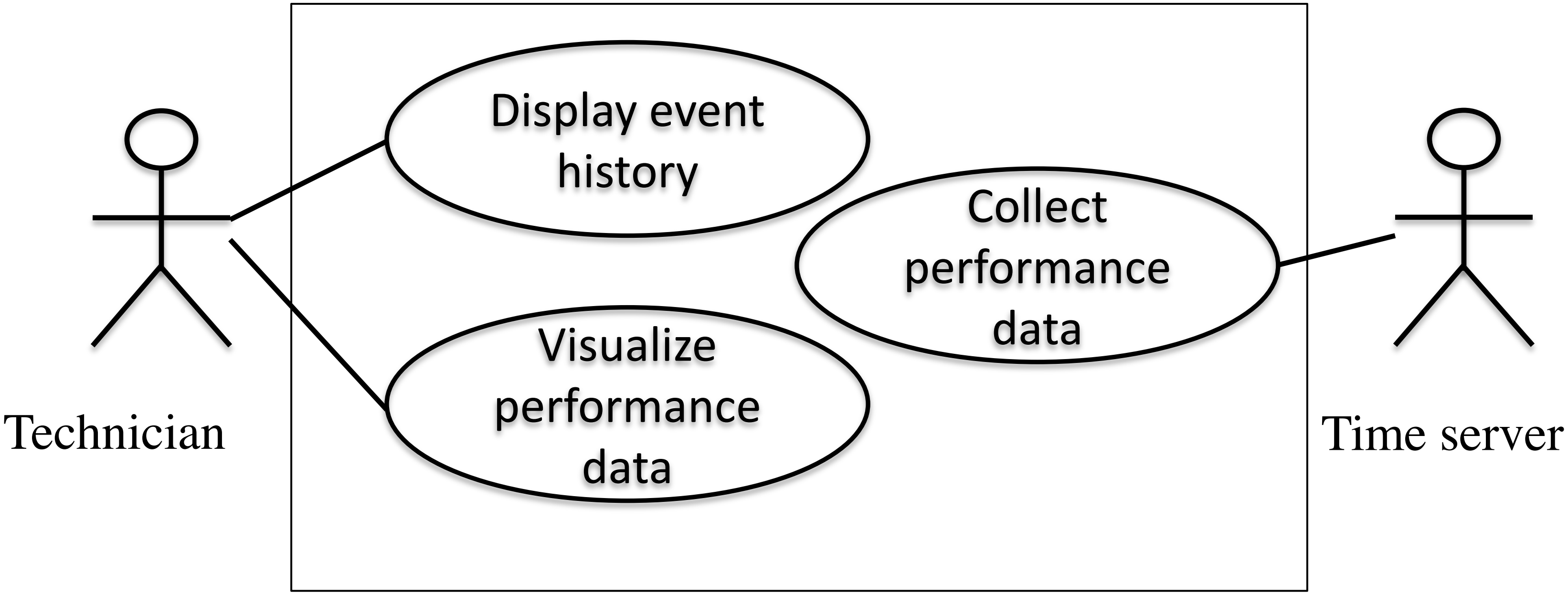
Goal:

- Extend the system to support VOIP
- Add servers that support other management protocols than SNMP

Purpose: Produce detailed design to support the construction of the system

Primary Functionality

- The system must fulfill the main expected functionalities
- We use use cases to represent the functionalities



Primary Functionality

- Primary functionality will be allocated to elements of the architecture to promote modifiability or reusability and assignments of work

- Decisions of the allocation of the functionality establish precedent for how the rest of functionality should be allocated.

Quality Attributes

- QAs indicates how well the system satisfies the needs of the stakeholders
  - Are measurable and testable properties of the system
  - Constraints on the functional requirements

- Important functional requirements should be associated with quality attributes, e.g.,
  - How fast should the function be?
  - How secure should the function be?
  - How modifiable should the function be?

Quality Attributes - Example

Existing system

- Servers use Network Time Protocol (NTP) to synchronize times
- Servers clustered in regions
- Each region has a stratum 1, equipped with hardware for precise time
- The servers support Simple Network Management Protocol (SNMP)
- The system conforms to FCAPS

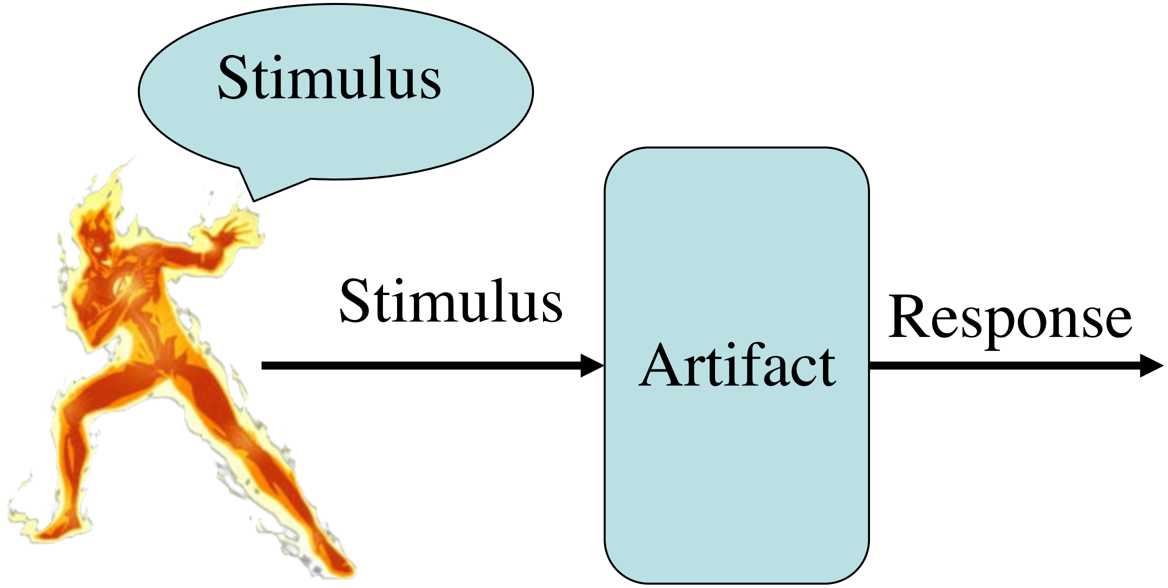
Goal:

- Extend the system to support VOIP
- Add servers that support other management protocols than SNMP

1. The management system collects all performance data within 5 min
2. A new time management protocol is added to the system without any changes to the core components of the system.
3. It is possible to know who performed a given operation and when.

Quality Attribute Scenario

Scenario: System response to some stimulus



Groups status

All AIX Servers	AG (105)	OK (10) UNK (1000)
Development/Testing	AG (0)	
External Hosts	AG (705)	OK (1137) WAR (1) CRI (281)
Hosting Servers	AG (8)	AL (1) OK (64) UNK (4)
HPUX Servers	AG (101)	UNK (1001)
Network	AG (5)	UNK (2)
SAP servers	AG (3)	OK (39)
Servers	AG (17)	OK (15) UNK (88)
Solaris Servers	AG (100)	UNK (1000)

Global health

Server health

Monitor health

Module sanity

Source of stimulus

Response measure

Specifying Quality Attribute

The parts of the scenario are

1. Stimulus – event that triggers the scenario
2. Stimulus source – who initiates the stimulus
3. Response – what the system do with the stimulus
4. Response measure – a metric that measures whether the response is satisfactory or not
5. Environment – conditions for the response
6. Artifact – parts of the system that contribute to the scenario

Example of Quality Attribute Scenarios

Scenario: Performance scenario for submitting applications

- Stimulus: Regular application
- Stimulus source: User
- Response: Save the application, process the payment, return a confirmation to the user
- Response measure: < 5 sec
- Environment: normal and overload conditions
- Artifact: submission, payment, save file, save database

Participation 9

You are assigned to a project to develop a system water-consumption management. Customers have a device that opens/closes the water pipe and measures the water consumption. The device communicates with the central office through cellular network. The main requirements of the system are:

1. Export anonymized data for data analytics; that is, replace ID, names, and ad- dresses of customers by codes.
2. Etc.

Architectural Concerns

- Architectural concerns are additional aspects that need to be considered in the architecture design.
- They include:
  - General concerns: e.g., establish the overall system or supporting delivery and updates
  - Specific concerns: e.g., exception management, dependency management, logging, caching
  - Internal concerns: facilitate development or operation
  - Issues: they result from analysis activities, e.g., risks

Architectural Concerns - Example

Existing system

- Servers use Network Time Protocol (NTP) to synchronize times
- Servers clustered in regions
- Each region has a stratum 1, equipped with hardware for precise time
- The servers support Simple Network Management Protocol (SNMP)
- The system conforms to FCAPS

Goal:

- Extend the system to support VOIP
- Add servers that support other management protocols than SNMP

1. Establish an overall system architecture
2. Leverage the team's knowledge about Java
3. Allocate work to members of the development team

Constraints

Architectural constraints are constraints on the development of the system.

- They include:
  - Mandated technologies
  - Other systems that the system must/should interact with
  - Laws and standards that must be complied with
  - Deadlines
  - Developers abilities

Examples of Architectural Constraints

Existing system

- Servers use Network Time Protocol (NTP) to synchronize times
- Servers clustered in regions
- Each region has a stratum 1, equipped with hardware for precise time
- The servers support Simple Network Management Protocol (SNMP)
- The system conforms to FCAPS

Goal:

- Extend the system to support VOIP
- Add servers that support other management protocols than SNMP

1. A minimum of 50 simultaneous users **must** be supported
2. The system is accessible using browsers
3. Must use an existing relational database
4. Use low-bandwidth internet connection

Architecture Drivers

Non functional requirements include:

1. Quality attributes
2. Architectural concerns
3. Architectural constraints

Check Your Knowledge

- What are the 5 types of architecture drives?
- What is the difference between architecture drivers and non-functional requirements?

Thank you

Questions?