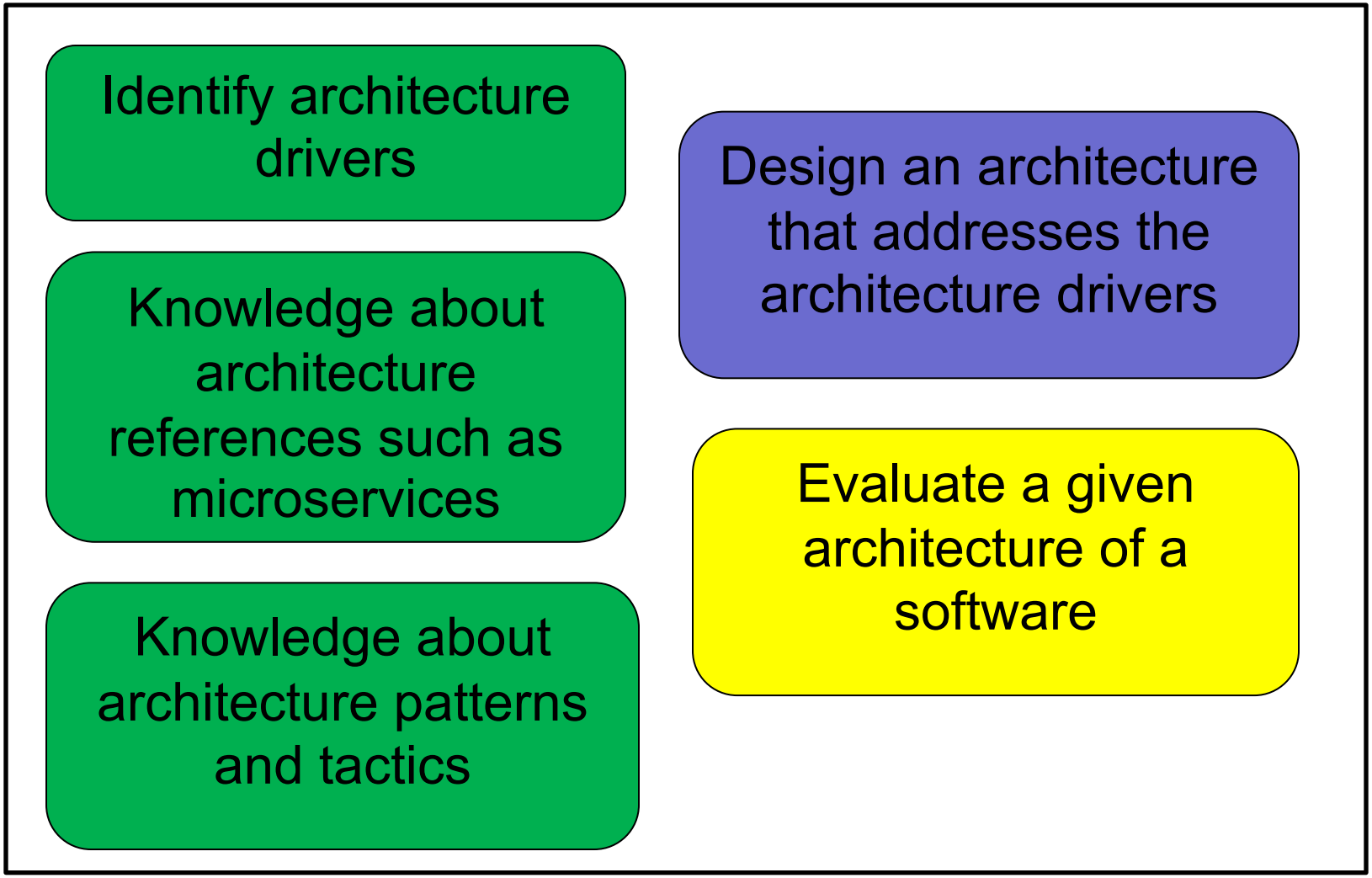
Architecture Evaluation

Lotfi ben Othmane

What Do We Know So Far?



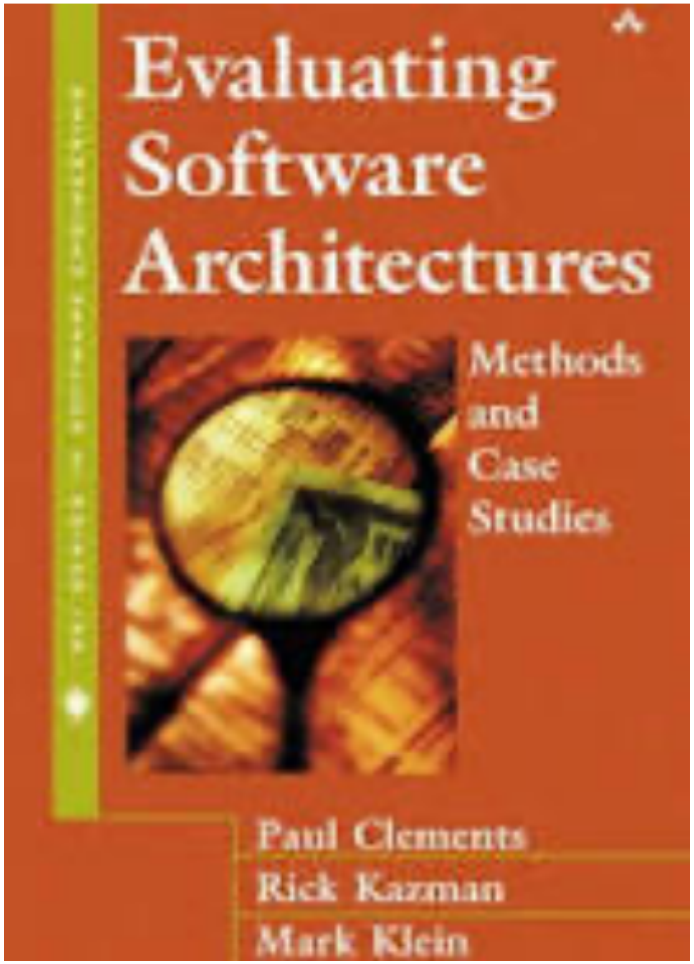
Course outcomes

1. Understand what a software architecture is and explain why it is important
2. Understand the relationship between software quality attributes and software architectures
3. Ability to elicit software architecture drivers
4. Ability to use architecture styles, patterns, and tactics
5. Ability to use the attribute-driven method to design software architecture
6. Ability to document software architecture
7. Ability to evaluate software architecture

What is ATAM

- ATAM stands for: Architecture trade-off Analysis Method.

- Reference: Chapter 3 of the book Evaluating Software Architecture



Architecture Risks

Risk Themes Discovered Through Architecture Evaluations

Runtime Qualities

Availability: These risk themes mention risks to availability or reliability goals. Issues that arose in this category included

- having a single point of failure
- not including availability mechanisms
- using infrastructure that does not support availability mechanisms

Performance: These risk themes mention problems with achieving performance goals. Issues that arose in this category included

- not knowing performance requirements
- not performing any performance modeling or prototyping
- unfamiliarity with infrastructure choices
- not using known performance mechanisms

Security: These risk themes mention problems with achieving security goals. Issues that arose in this category included

- unknown requirements
- not using known mechanisms to support security goals

Development Time Qualities

Modifiability: These risk themes mention problems with achieving modifiability goals. Issues that arose in this category included

When Do we Evaluate an Architecture?

- A project manager may want to evaluate the architecture of their project
- An acquisition team may want to assess a company/system that they plan to buy

- Assessing the students' architectures in this course?

Case Study -Smartsync

The report of SMARTSYNC is offered by a team of students to use for learning.

It includes a design like the ones that you may do!!

What are the goals of ATAM?

1. Design a robust architecture
2. Identify weaknesses of a given architecture
3. Provide a principled way to evaluate architecture
4. Provide a structured way to understand the consequences of the decisions
5. Analyze how an architecture achieves its goals
6. Analyze the risks of a given architecture

ATAM stands for: Architecture trade-off Analysis Method.

Steps of ATAM

The ATAM evaluation consists of nine steps:

1. Present the ATAM
2. Present business drivers
3. Present the architecture
4. Identify the architectural approaches
5. Generate the quality attributes utility tree
6. Analyze the architectural approaches
7. Brainstorm and prioritize scenarios
8. Analyze the architectural approaches
9. Present results

Steps of ATAM

- Evaluation team
  - Present ATAM
  - Present results
- Architecture team
  - Present business drivers
  - Present the architecture
- Evaluation and architecture teams
  - Identify architecture approaches
  - Develop utility tree
  - Analyze architectural approaches.
- Stakeholders
  - Brainstorm and prioritize scenarios
  - Analyze architectural approaches

Step 2 -Present Business Drivers

- The project manager or representative presents to the committee
  1. The system most important functions
  2. Technical, managerial, economic, or political constraints
  3. Business goals and context
  4. Major stakeholders
  5. Major architecture drivers

Step 2 -Present Business Drivers- SmartSync

SmartSyncis a home management system for:

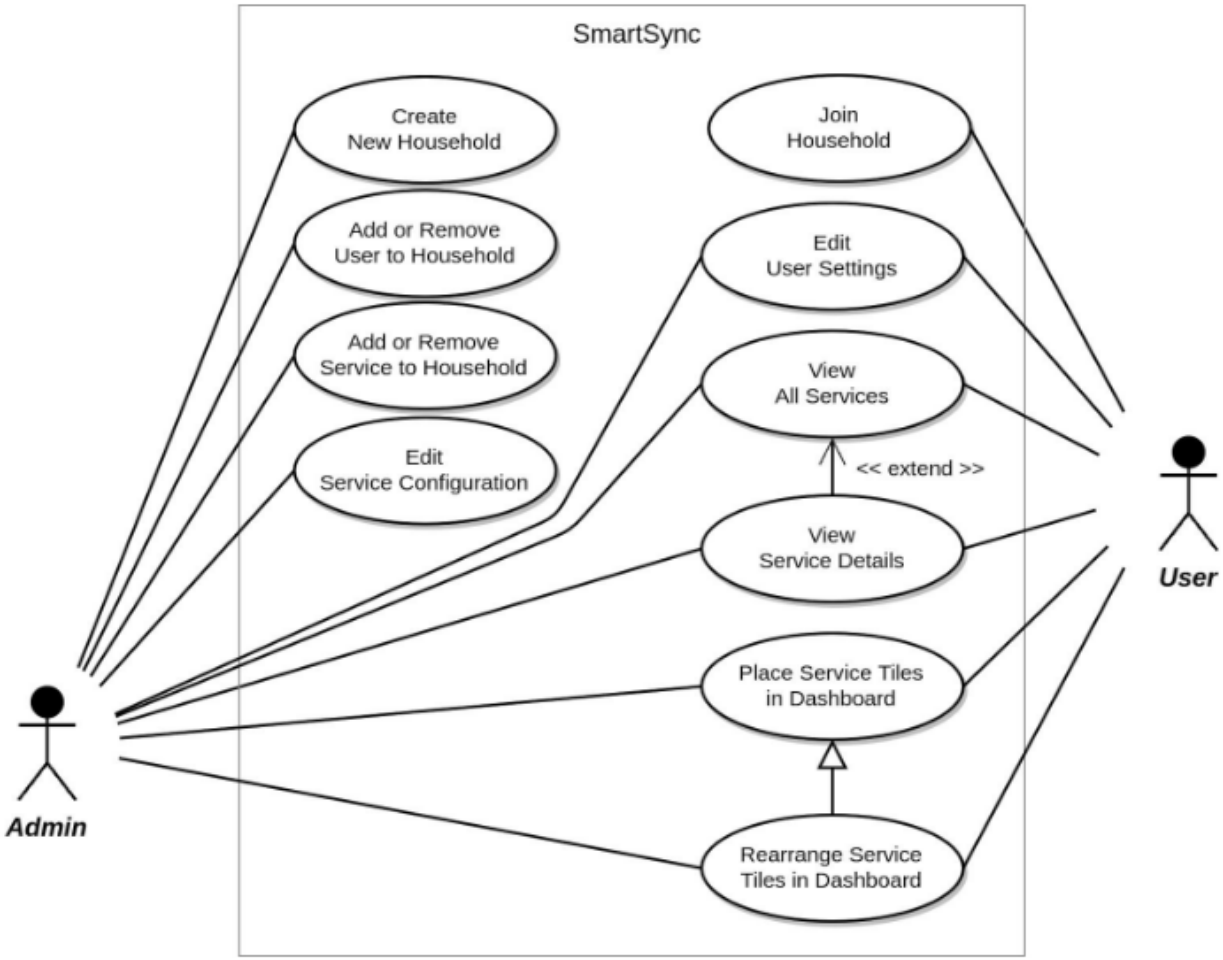
1. Synchronizing and managing Internet of Things devices and services
2. Monitoring the status of multiple devices/services
3. Administering the system
4. Securing the use of the system

Step 3 -Present Architecture

- The lead architect presents the software architecture of the given system. This includes
  1. Technical constraints such as operating systems, middleware, etc.
  2. Systems that interact with the evaluated system
  3. Architectural approaches (e.g., styles, patterns) used to meet the quality attributes

- The team may use informal views
- The evaluation team may ask questions to understand the architecture

Step 3 -Present Architecture -SmartSync

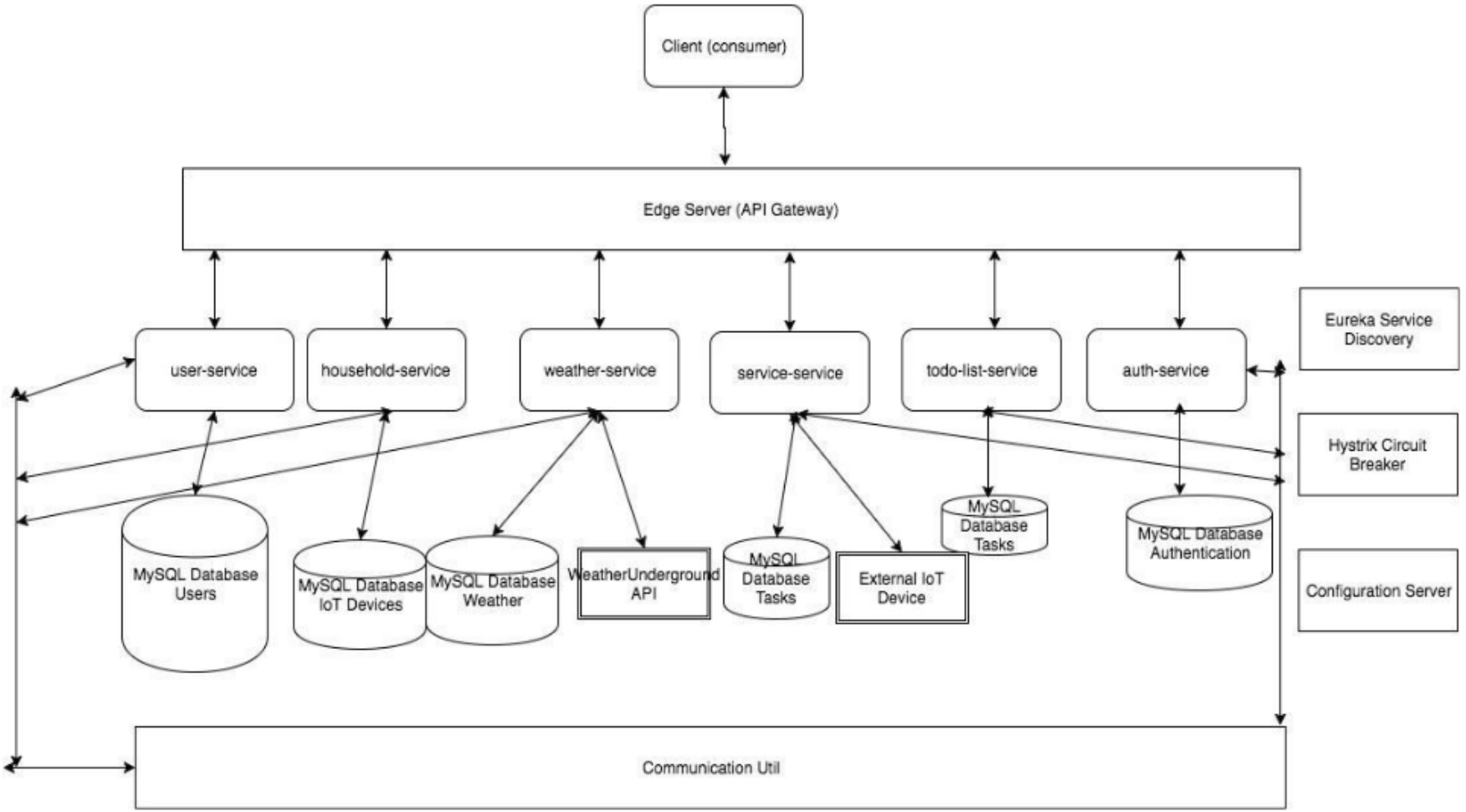


Step 3 -Present Architecture -SmartSync

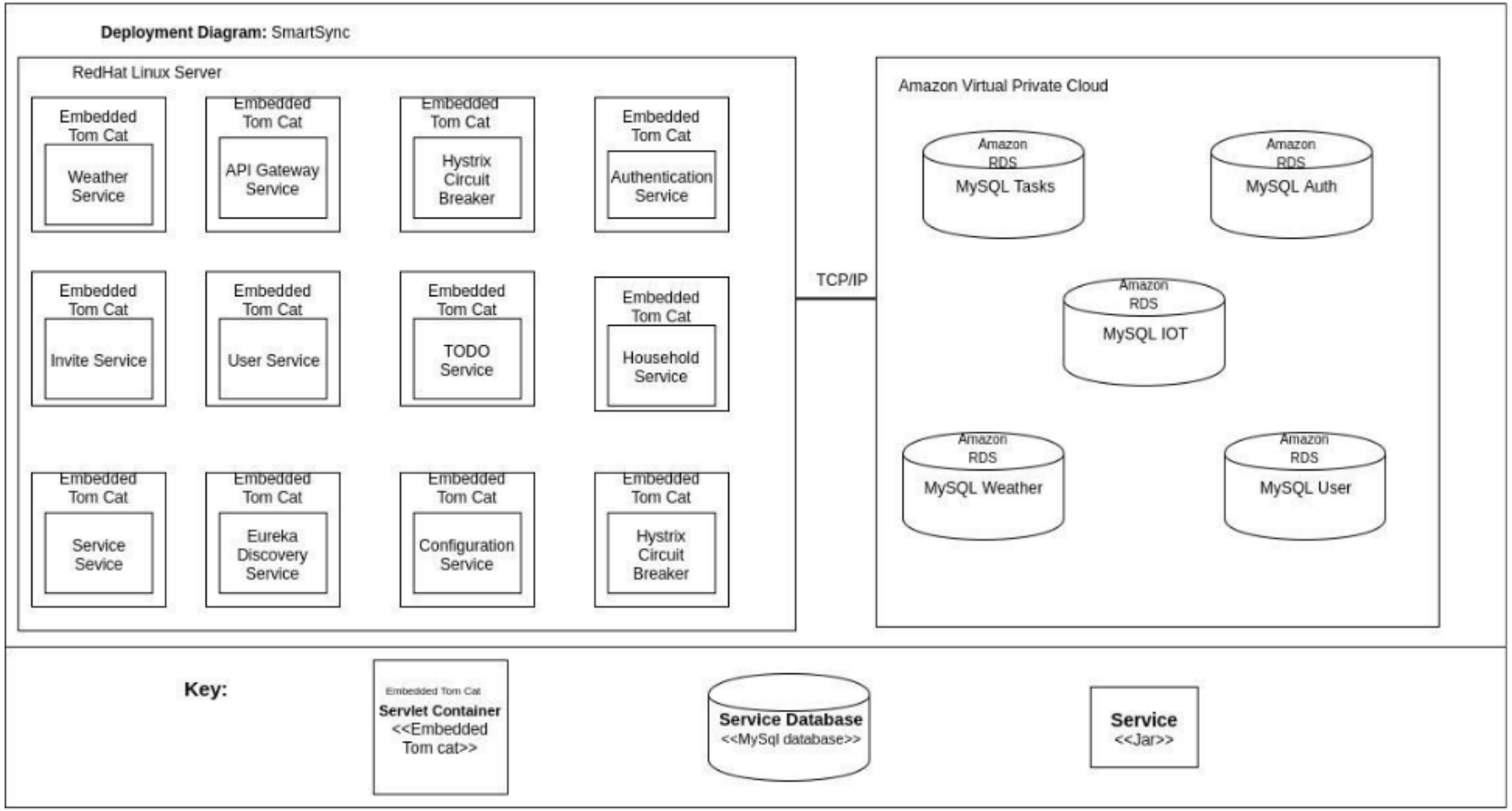
Quality attributes:

- Users must be authenticated before accessing the system
- All requests must be answered within 30 seconds
- Users access the features in conformance with the access policies

Step 3 -Present Architecture -SmartSync

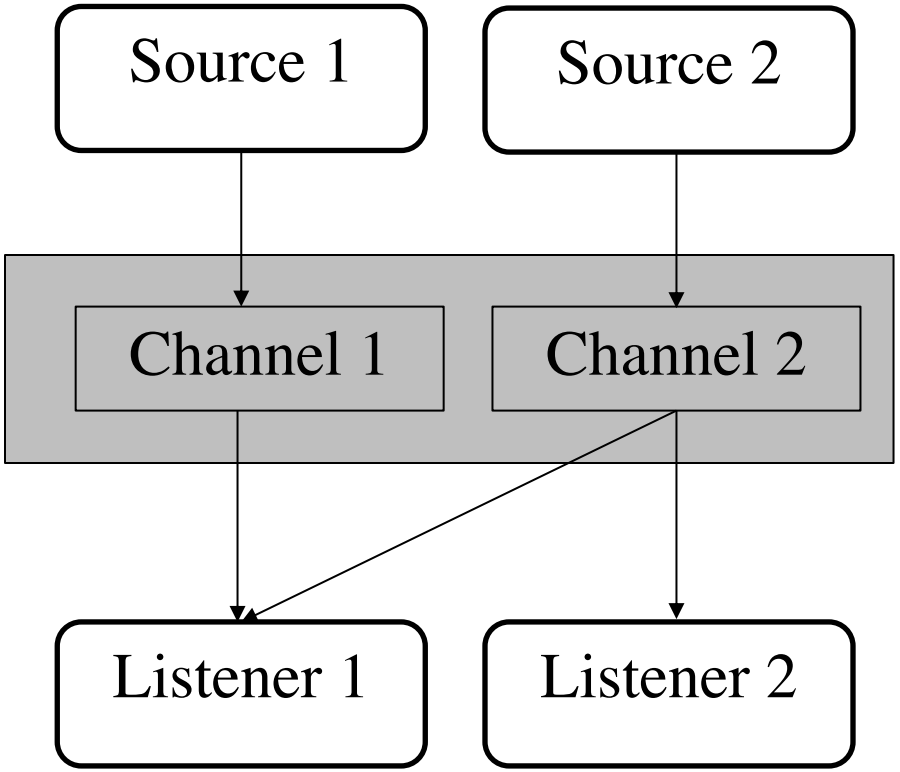


Step 3 -Present Architecture -SmartSync



Step 4 -Identify the Architectural Approaches

- The evaluation team asks the architecture team to name the architecture approaches that are used in the design
- The teams collect data but do not analyze the performance of the used approaches yet



Step 4 -Identify the Architectural Approaches-SmartSync

What architectural style did the architects use?

1. Microservices
2. Mobile application
3. IOT-based style
4. Client/server
5. Another style

Step 4 -Identify the Architectural Approaches -SmartSync

- Reference architecture: Microservices
- Use Spring boot as framework
- AngularJS 2 for front end
- Use Google’s OAuth2 API for authentication

Step 4 -Identify the Architectural Approaches -SmartSync

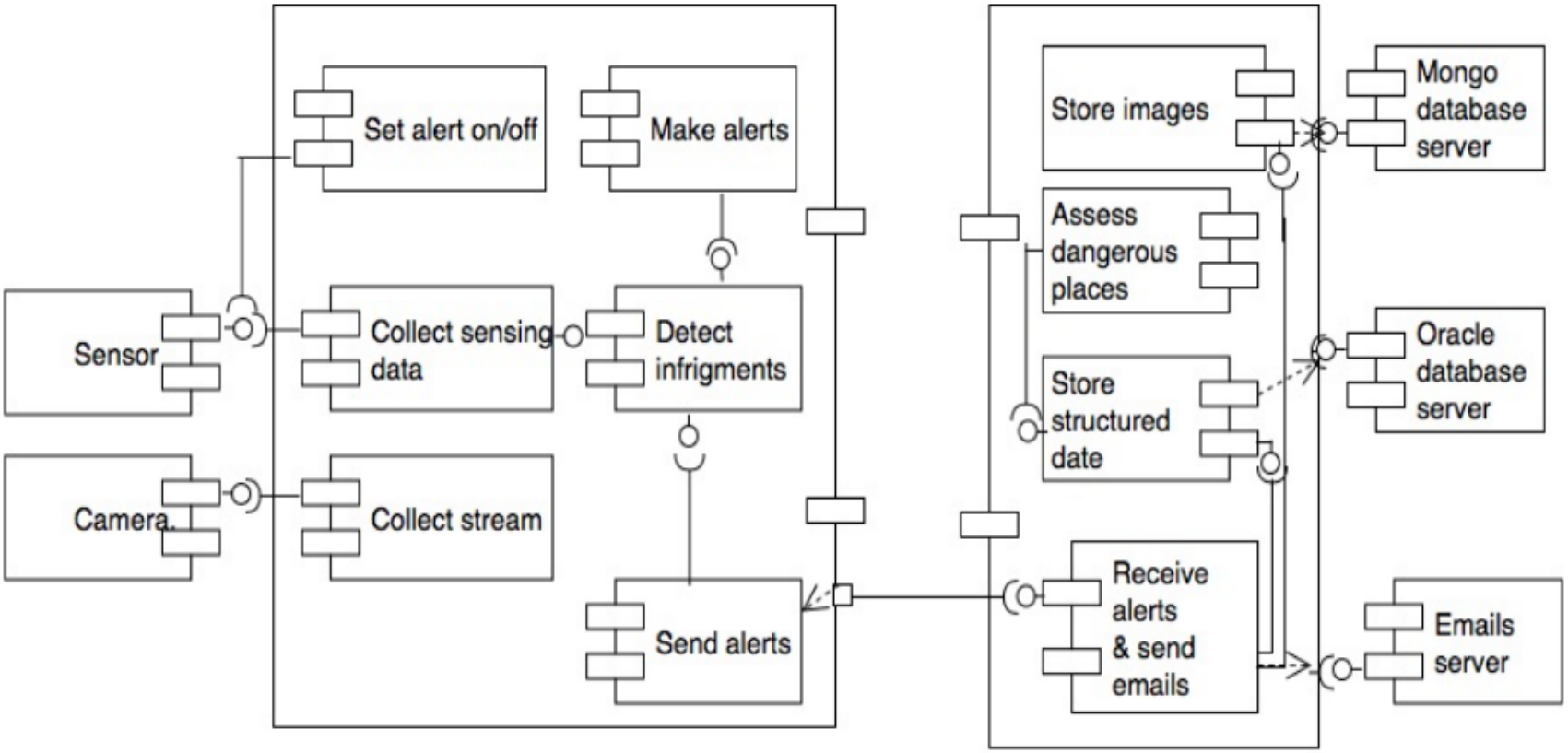
Used architectural approaches

1. Microservice
2. API Gateway
3. Layer pattern—e.g., MVC
4. Caching
5. Three-tiers pattern

Step 5 -Generate Quality Attributes Utility Tree

- The evaluation and architecture team work together to identify and prioritize the main quality attributes
- Utility tree is a tree of QA requirements of the system
  - The root is called utility
  - The quality attributes (performance, etc.) form the second level
  - The QA scenarios make the leaf nodes
- The teams should give ranks (H,M,L) to the QA scenarios based on importance and difficulty to implement

Step 5 -Generate Quality Attributes Utility Tree



Step 5 -Generate Quality Attributes Utility Tree -SmartSync

Utility

1. Performance
    1. Response time for the services
    2. Latency for the devices
    3. Response time for the authenticating users using the security cameras
2. Modifiability
    1. Adding and removing new services
    2. Adding and removing new views
3. Availability
    1. Up time for services should be 99.99%
    2. Fail of devices should be identified in 30 seconds
4. Security
    1. All users must be authenticated
    2. Only household can grant and deny permissions to their smart home services
    3. Only household can authorize users to join their smart home.

Step 6 -Analyze Architectural Approaches

- The evaluation team needs to answer the question:
  - Are the instantiation of the architecture approaches appropriate for the architecture drivers?

- The analysis include:
  - Identify applicable known weaknesses
  - Assess the sensitivity point, e.g., response as transaction/sec.
  - Trade-offs of the used architecture drivers

Step 6 -Analyze Architectural Approaches -SmartSync

1. Microservice –for adding and removing new services
2. API gateway –for complexity of the microservices
3. Layer pattern -e.g., MVC
4. Caching –for response time for the services
5. Three-tiers pattern –for use cloud service

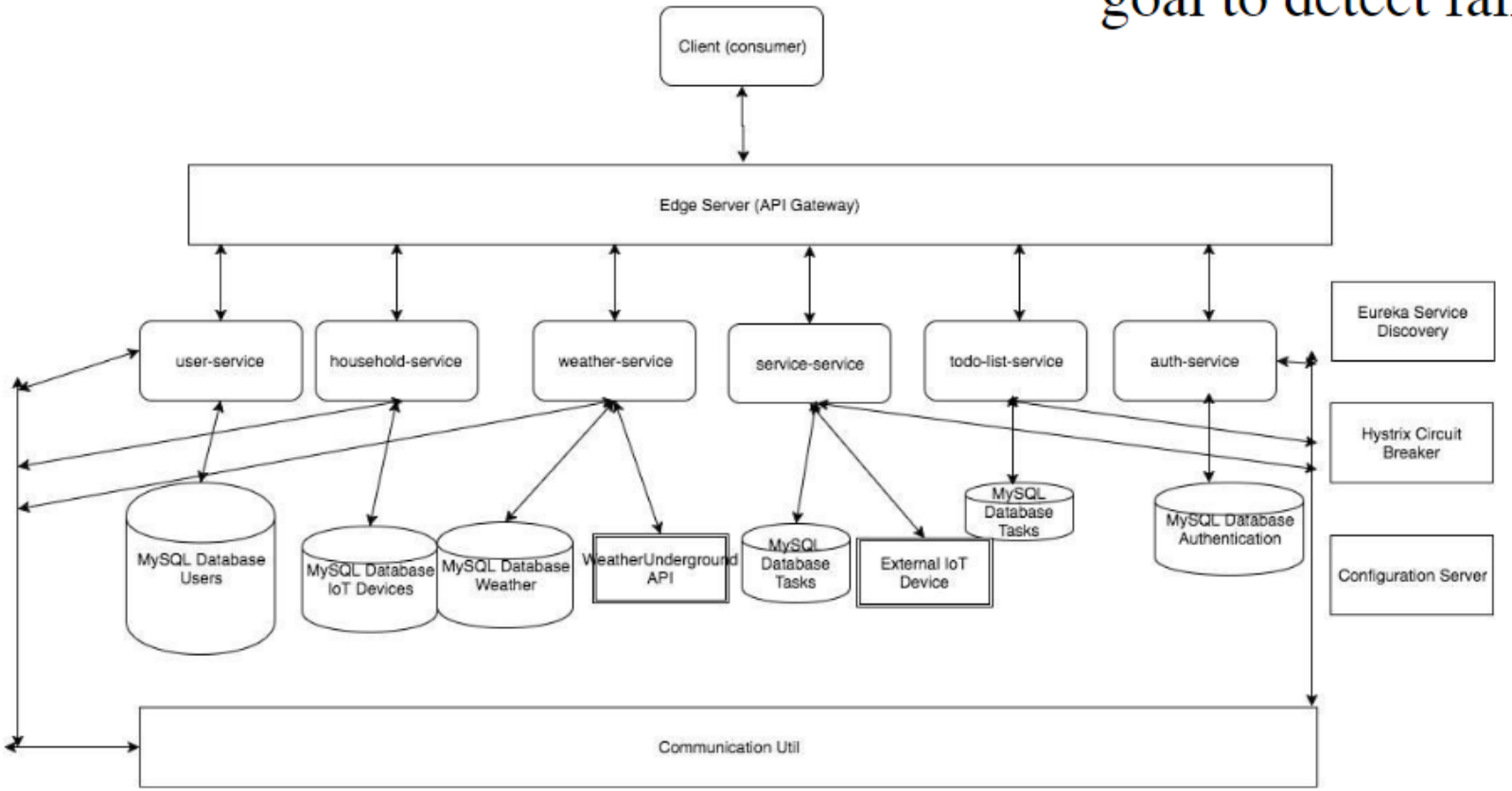
Step 6 -Analyze Architectural Approaches -SmartSync

- Does the architect use pattern, tactics or styles to address the following requirements?

1. Up time for services should be 99.99%
2. Fail of devices should be identified in 30 seconds

Step 6 -Analyze Architectural Approaches -SmartSync

Used Erika with the goal to detect failure.



Step 6 -Analyze Architectural Approaches -SmartSync

Utility

1. Performance
   1. Response time for the services
   2. Latency for the devices
   3. Response time for the authenticating users using the security cameras
2. Modifiability
   1. Adding and removing new services
   2. Adding and removing new views
3. Availability
   1. Up time for services should be 99.99%
   2. Fail of devices should be identified in 30 seconds
4. Security
   1. All users must be authenticated
   2. Only household can grant and deny permissions to their smart home services
   3. Only household can authorize users to join their smart home.

Step 7 -Brainstorm and Prioritize Scenarios

- All the stakeholders meet to test the architecture
- They develop scenarios for the architecture
  - Use case scenarios –The stakeholders expect the system to do the use cases
  - Growth scenarios –Scenarios that the architecture should support in the moderate to near term
  - Exploratory scenarios –Includes dramatic changes such as new major availability or performance requirements
- The scenarios are ranked

Step 8 -Analyze Architectural Approaches

- The architecture team analyzes how the architecture addresses the scenarios identified in the brainstorming session
- The evaluation team and architecture team work together on mapping the architecture approaches to scenarios

Step 9 –Present Results

- The evaluation team presents the work to stakeholders
- The evaluation team prepares a report for the stakeholders
- The collected information could be used as a knowledge base. It could also be used to derive statistics, tendency assessment, etc.

Architecture Design

Architecture drivers

- Design purposes
- Quality attributes
- Primary functionalities
- Architectural concerns
- Constraints

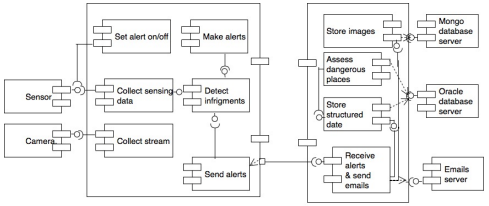
Design concepts

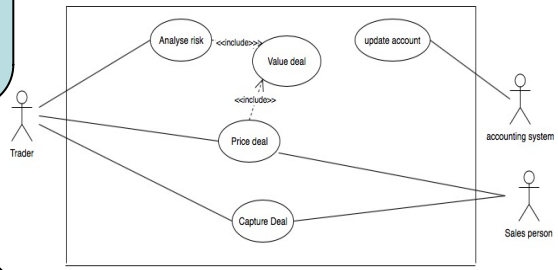
Selects and instantiates

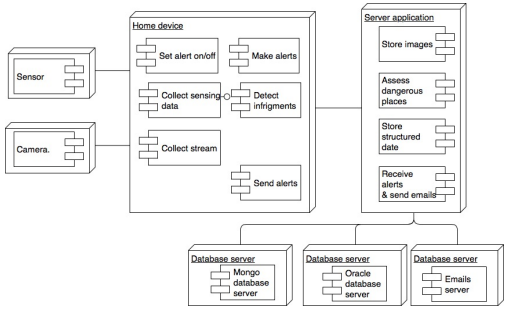
Candidate design decisions

Architect

Architecture structures







Class Discussion

Facebook Chat

By Eugene Letuchy on Tuesday, May 13, 2008 at 10:56 PM

One of the things I like most about working at Facebook is the ability to launch products that are (almost) immediately used by millions of people. Unlike a three-guys-in-a-garage startup, we don't have the luxury of scaling out infrastructure to keep pace with user growth; when your feature's userbase will go from 0 to 70 million practically overnight, scalability has to be baked in from the start. The project I'm currently working on, Facebook Chat, offered a nice set of software engineering challenges:

Real-time presence notification:

The most resource-intensive operation performed in a chat system is not sending messages. It is rather keeping each online user aware of the online-idle-offline states of their friends, so that conversations can begin.

The naive implementation of sending a notification to all friends whenever a user comes online or goes offline has a worst case cost of O(average friendlist size * peak users * churn rate) messages/second, where churn rate is the frequency with which users come online and go offline, in events/second. This is wildly inefficient to the point of being untenable, given that the average number of friends per user is measured in the hundreds, and the number of concurrent users during peak site usage is on the order of

Self-Check

- What is ATAM?
- Why companies evaluate architecture?
- What are the participants in architecture evaluation?
- Why the stakeholders are involved in the architecture evaluation?
- Why the analysis of architecture analysis is repeated?
- What scenario would you suggest for the SmartSycproject?

Thank you

Questions?