Architecture Evaluation Methods:

Introduction to ATAM

UNIVERSIDAD
POLITECNICA
DE VALENCIA

Contents

- What is ATAM?
- What are the outputs of ATAM?
- Phases and Steps of ATAM
- ATAM Running Example

What is ATAM?

- **ATAM** (Architecture Trade-Off Analysis Method) is an architecture evaluation method developed in the Software Engineering Institute at the end of the 90s
  - The purpose of ATAM is to assess the **consequences** of **architectural design decision**s in the light of **quality attributes**.
  - ATAM helps in foreseeing how an **attribute** of **interest** can be **affected** by an **architectural design decision**.
  - The Quality Attributes of interest are clarified by **analyzing** the stakeholder's **scenarios** in terms of stimuli and responses.
  - Once the scenarios had been defined, the next step is to define which ***architectural approaches***[1] can **affect** to those **quality attributes**.

[1] In ATAM, the term ***architectural approaches*** are used to refer both to architectural styles or patterns

What are the outputs of ATAM?

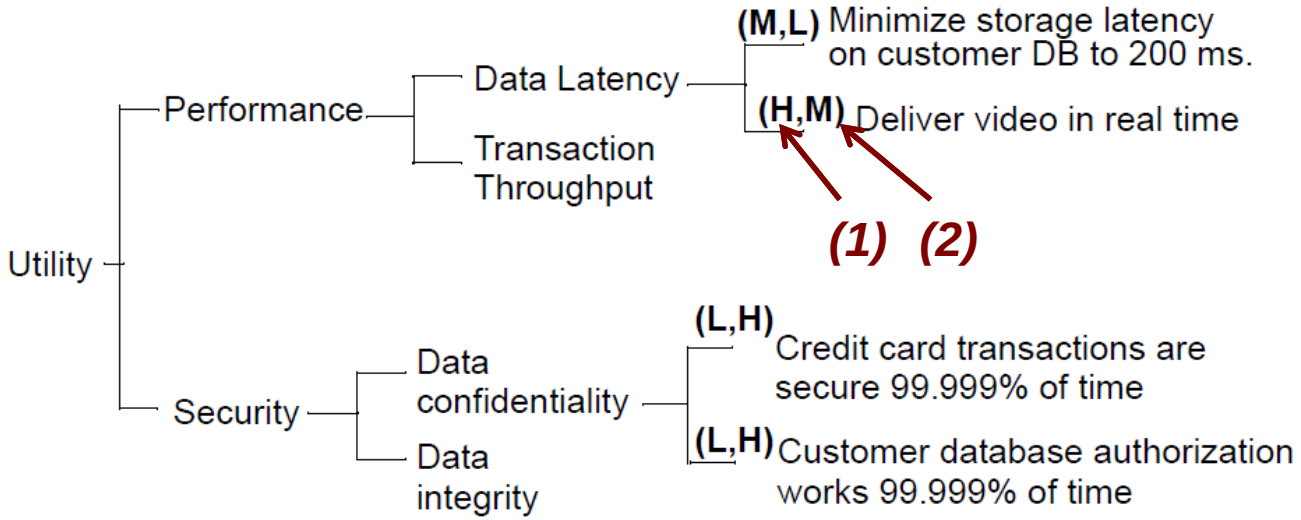
- The ATAM output are:
  - A set of ***architectural approaches*** identified and or applied: Sometimes we can identify architectural approaches that cannot be applied on our architecture
  - A ***Utility Tree***: a top-down mechanism for directly and efficiently translating the business drivers of a system into concrete quality attribute scenarios.
  - A ***set of scenarios*** identified and the subset that had been effectively mapped in the architecture.
  - A ***set of questions about the quality attributes*** in the architecture and the ***answers to these questions***. In our case our questions are a set of metrics and the answers are the values measured.
  - The ***risks identified***: risks that the architecture is able to mitigate and the risks that threaten the system and the business goals.

ATAM Scenarios

- In ATAM both the scenarios that had been identified and those that had been mapped into the architecture are documented in terms of ***stimuli and response***.
  - In some cases, we can identify scenarios that describe requirements that cannot be effectively mapped into the architecture.
  - We document both the ones that had been identified and those that had been mapped into the architecture.
- ***Stimuli*** are the events that cause the architecture to respond or change.
- To analyze an architecture for adherence to quality requirements, those requirements need to ***be expressed in terms that are concrete and measurable or observable***.

Utility Tree

- The utility tree contains attribute goals concrete enough for prioritization.
- Those scenarios will make these architectural response goals even more specific. The prioritization can be done by using ***Low, Medium or High*** literals associated to each leaf node.
- Usually this prioritization is done using two dimensions: the ***(1) importance of each node in the success of the system*** and the ***(2) degree of perceived risk associated to the achievement of that goal***.



ATAM Phases and Steps I

- ***Phase 1: Presentation***
  1. **Present ATAM:** The method is described to the evaluators and stakeholders.
  2. **Present business drivers:** business goals which motivate the development effort and hence that will be the primary architectural drivers are presented.
  3. **Present the architecture:** describe the proposed architecture, focusing on how it addresses the business drivers.
- ***Phase 2: Investigation and Analysis***
  4. **Identify architectural approaches (patterns):** Architectural approaches supporting the system and business goals are identified by the architect, but not analyzed
  5. **Generate quality attribute utility tree:** Quality factors that comprise system "utility" (performance, availability, security, etc.) are elicited, specified down to scenarios level, annotated with stimuli and responses, and prioritized.
  6. **Analyze architectural approaches:** The stakeholders and the architect analyze how the architectural approaches affect to the quality factors identified in Step 5.

ATAM Phases and Steps II

- ***Phase 3: Testing***

  7. **Brainstorm and prioritize scenarios:** If required, more scenarios are generated and prioritized together with the scenarios on the utility tree.

  8. **Analyze architectural approaches:** This step reiterates step 6, but here the highly ranked scenarios from Step 7 are considered and we should decide which architectural patterns are applied and how the architecture will be modified.

- ***Phase 4: Reporting***

  9. **Present results:** Based upon the information collected in the ATAM (patterns, scenarios, attribute-specific questions, the utility tree, risks, sensitivity points, tradeoffs) the ATAM team presents the findings to the assembled stakeholders and potentially writes a report detailing this information.

ATAM Running Example

UNIVERSIDAD
POLITECNICA
DE VALENCIA

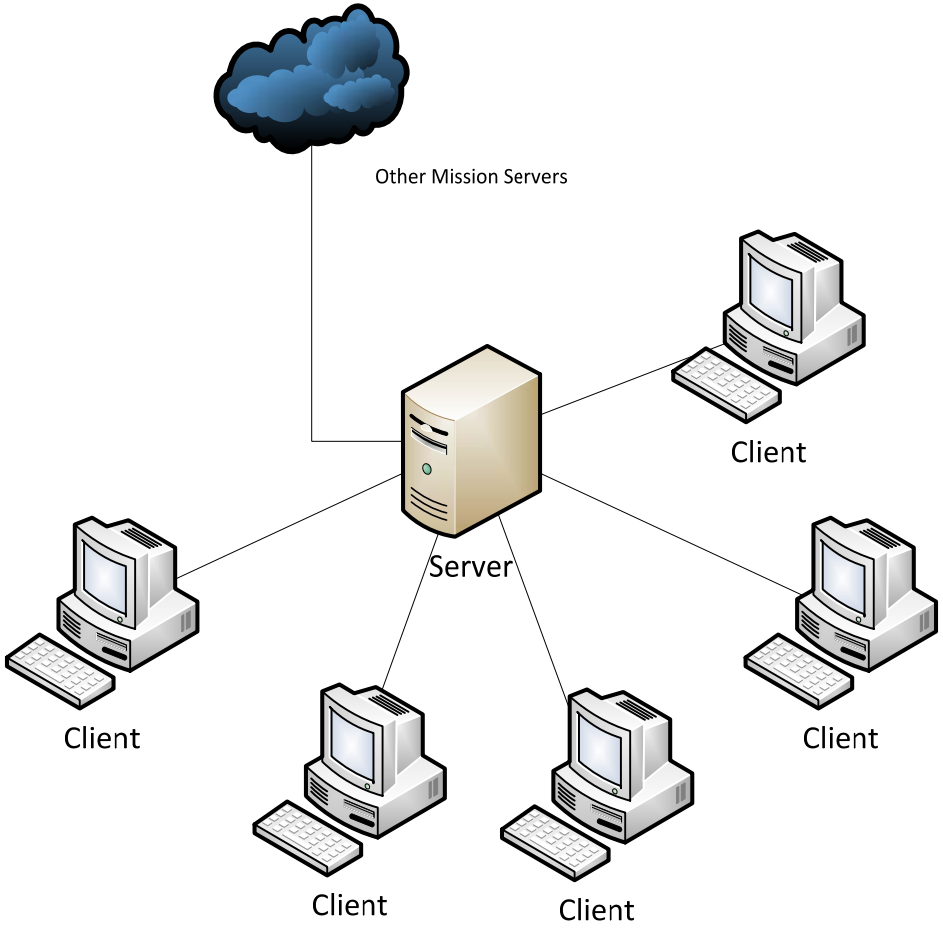
ATAM in a Mision Control System

FASE 1:

1. **ATAM Presentation**
2. **Business Goals Presentation:** In our example we are going to deal with a **mission control system**, and the business goals, are among others, the high **reliability** and **high performance.**
3. **Architecture Presentation:**
   - The system of our example is the ***Battlefield Control System***. This system is used to control the **movements** of **troops** in the battlefield. There are **two types of nodes** in its architecture:
     - **Server Node (commander):** is the node that supports those in head of taking decisions. It handles a database with the soldiers and battlefield status and also transmits the orders to the client nodes. In addition is capable of communicating with other Server nodes to send and receive orders.
     - **Client Nodes (Soldiers):** make requests to the server and update the server's database
   - The internode communication between the clients and the server is only though encrypted messages sent via a shared radio communication channel
     - 9600bds limited bandwidth
     - Only one node can be broadcasting at any moment.

3. Architecture Presentation

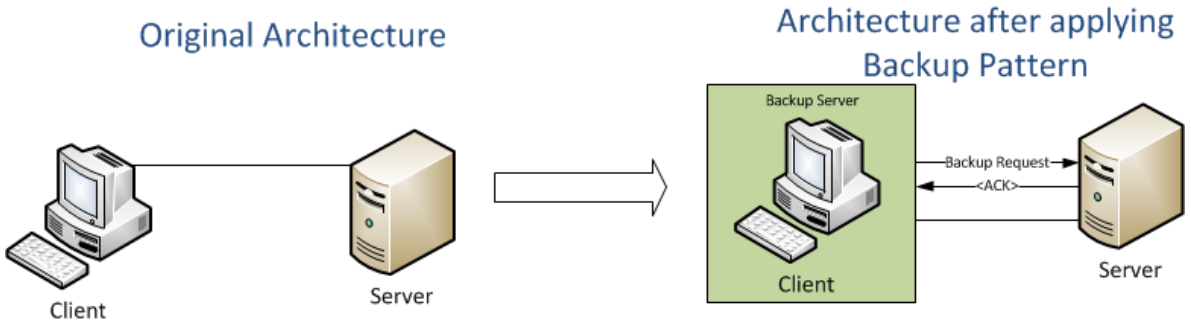
- **Original Architecture to be evaluated:**



Architectural Limitations

- This system architecture has a reliability problem, the system depends totally on the server
  - If the server fails the system becomes totally useless.
  - We should consider which architectural transformations can be applied to this architecture to solve this problem.

4. Architectural Patterns Identification: Backup Server



Problem addressed:

- In some systems a failure on the server means that the system goes down and the service is interrupted.

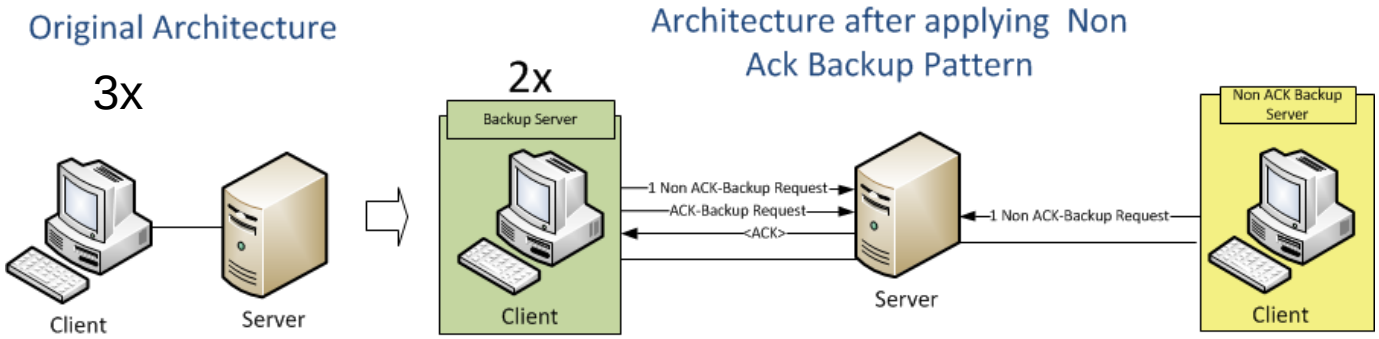
Structure:

- A client node can promote to backup server by sending a request and by receiving the corresponding ACK from the server.
- The backup server mirrors the server database each 10 minutes and monitors the communications between the server and the client nodes (stores every message sent). If the server fails automatically promotes to server.

Consequences:

- Improves the availability of the system
- Increases the communications through the channel, since the backup server has to receive a copy of the server's database (in our case 55Kb)

4.Architectural Patterns Identification: Non ACK Backup (1/2)



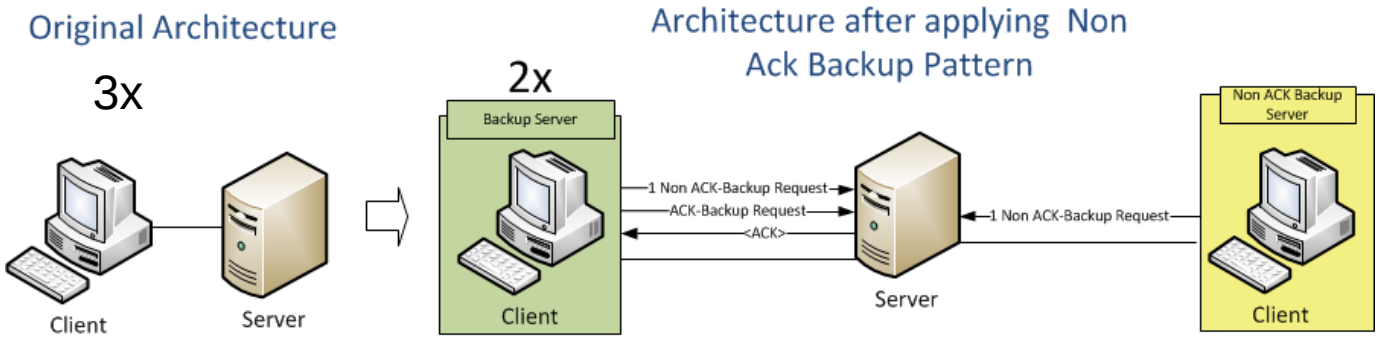
Problem:

- In some systems a failure on the server means that the system goes down and the service is interrupted.

Structure:

- A client node can promote to Non-ACK backup server by sending a request to the server.
- A Non-ACK Backup Server node can promote to backup server by sending a request and by receiving the corresponding ACK from the server.
- The Non-ACK Backup Server only monitors the communications, storing a copy of the messages sent through the communication channel. If a Non-ACK Backup Server needs to promote directly to Server, it will ask the other clients to resend information about their state.
- The backup server mirrors the server database each 10 minutes and monitors the communications between the server and the client nodes (stores every message sent). If the server fails automatically promotes to server.

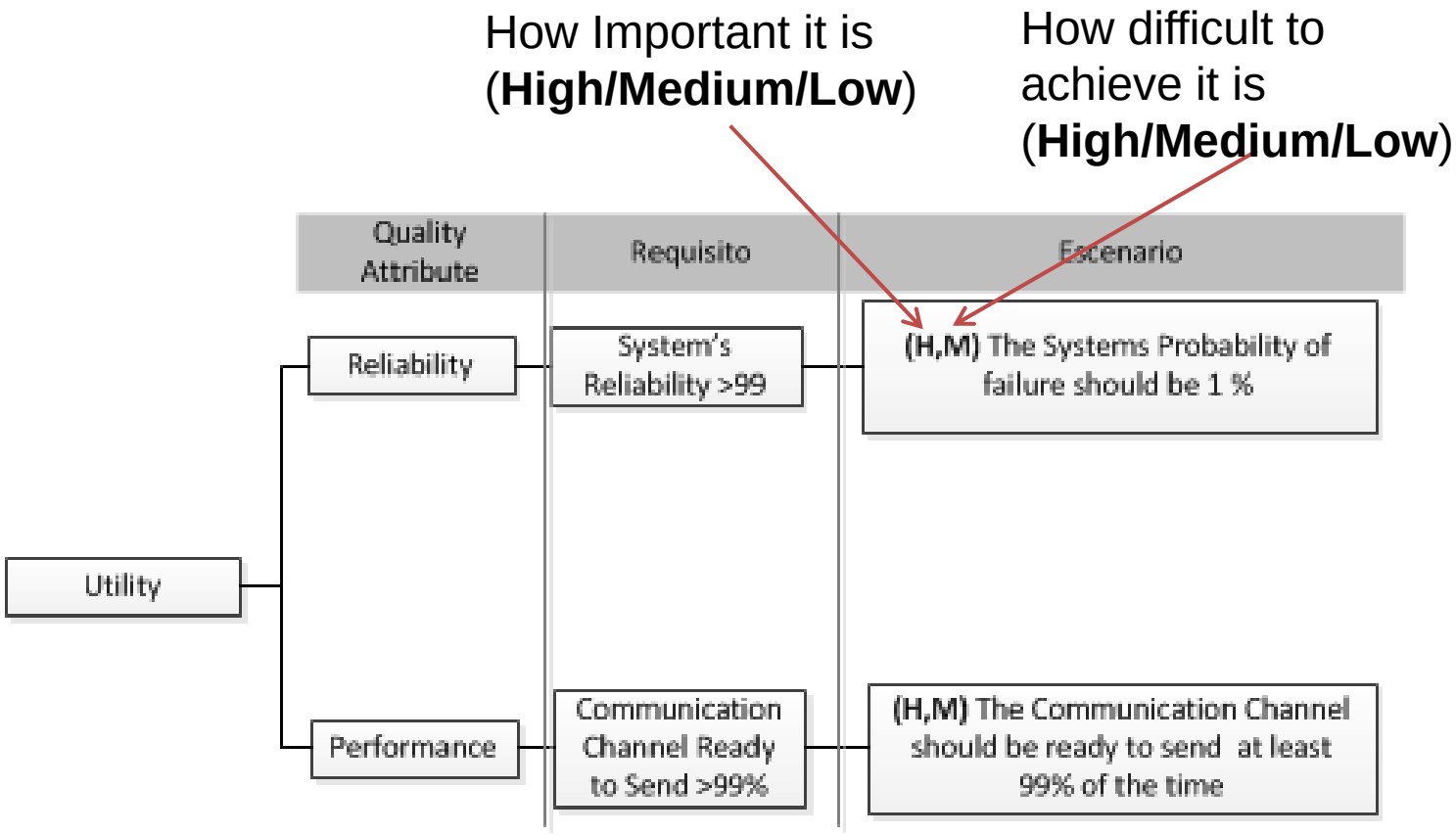
4. Architectural Patterns Identification: Non ACK Backup (2/2)



Consequences:

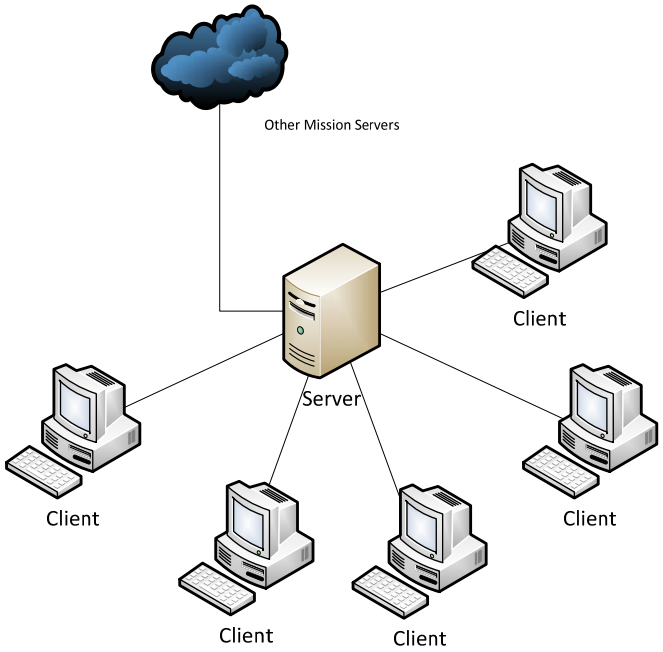
- Improves the system's availability. The system survives to an eventual failure both on the server and on the backup servers before a backup server has promoted to server.
- Maintain two backup servers mirroring the server's database increases the network load, since the two backup servers have to receive a copy of the server's database (in our case 55Kb)
- If there is a simultaneous failure on the server and on each single backup server require the Non-ACK backup to ask the other client nodes to resend the state, increasing the recovery time.

5. Utility Tree Generation



Architecture Evaluation (Additional Step)

- In this step, we are going to apply the metrics that evaluate the fulfillment of each NFR.



- **Reliability Analysis:**
  - The probability of failure of a node (and of the server) is 5%, so the probability of failure of the system ≃ P(Server) = 5%
- **Performance Analysis:**
  - The communication channel is only used for transition between server and clients and then it is ready to send a 100% of the time.

Architecture Evaluation (Additional Step)

- We are going to obtain the values for the metrics by means of an Excel spreadshe

Architecture Evaluation (Additional Step)

- We are going to obtain the values for the metrics by means of an Excel spreadsheet that automates the calculations.
  - **Reliability: Probability of Failure**
    - Value Obtained: 5%
    - Does it fulfill the minimal threshold defined in NFR_001? NO

  - **Performance: % of Time Ready To Send**
    - Value Obtained: 100%
    - Does it fulfill the minimal threshold defined in NFR_001? YES

6. Architectural Approaches Analysis

- **Architectural Approaches Analysis**
  - How will the approaches identified in Step 4 improve the quality factors presented in the utility tree?
  - The application of the Backup Pattern will increase the reliability of the system, but on the contrary, it makes use of the communication channel to mirror the server's database.
  - The application of the Non-ACK Backup will improve much more the reliability, but on the other hand the communication channel will probably be busy too much time.

Scenario Prioritization and Approach Analysis

Phase 3:

7. Scenario Prioritization

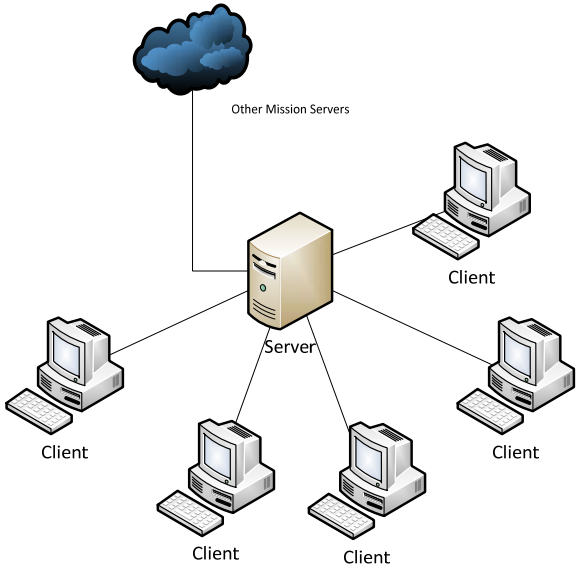
- In our case since we did not aggregate new scenarios and both scenarios on the utility tree are high priority scenarios.

8. Architectural Approaches Analysis

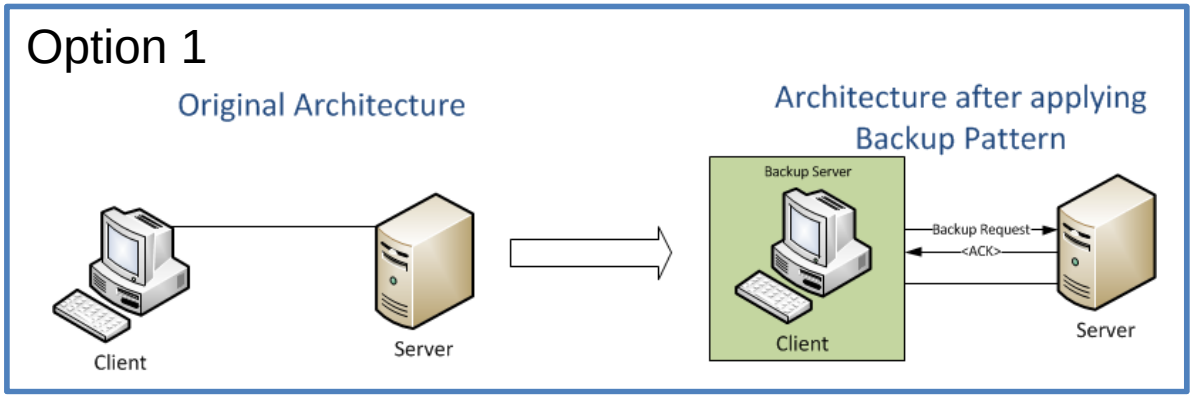
- Considering the architecture, the architectural approaches and the high priority scenarios we should decide which approach is more appropriated, and detect the changes on the architecture.

8. Approach Analysis II

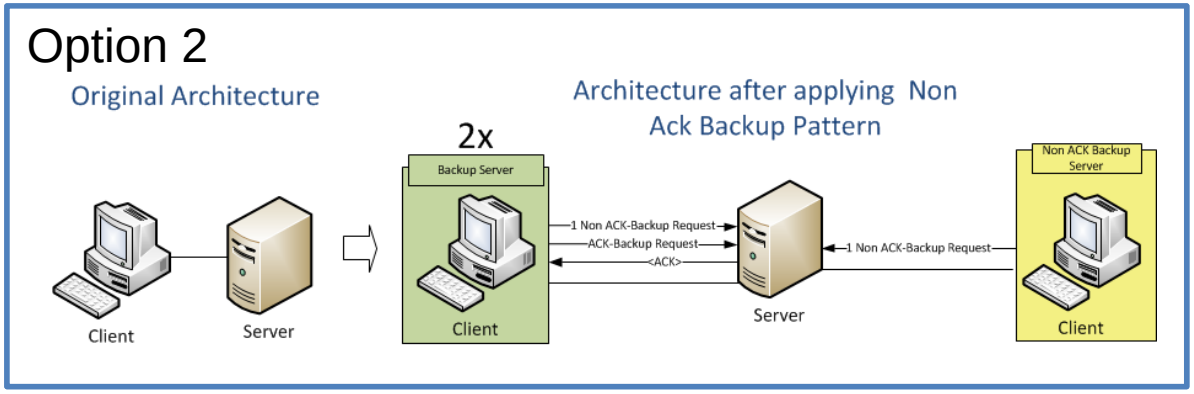
Original Architecture



Option 1



Option 2



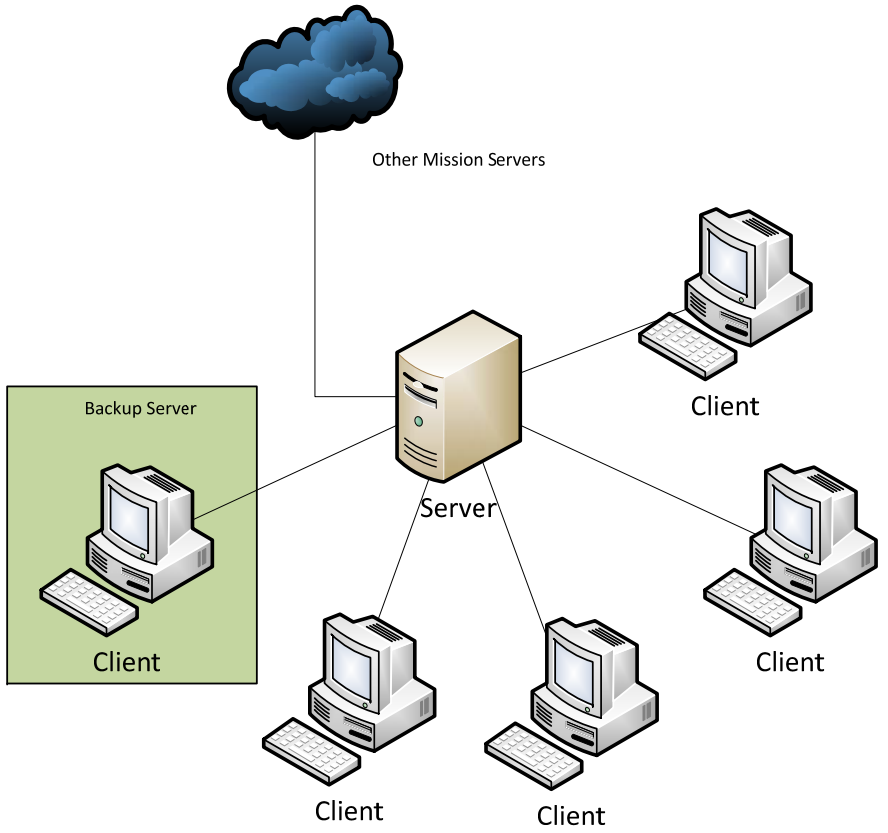
Scenarios

(H,M) The Systems Probability of failure should be 1 %

(H,M) The Communication Channel should be ready to send at least 99% of the time

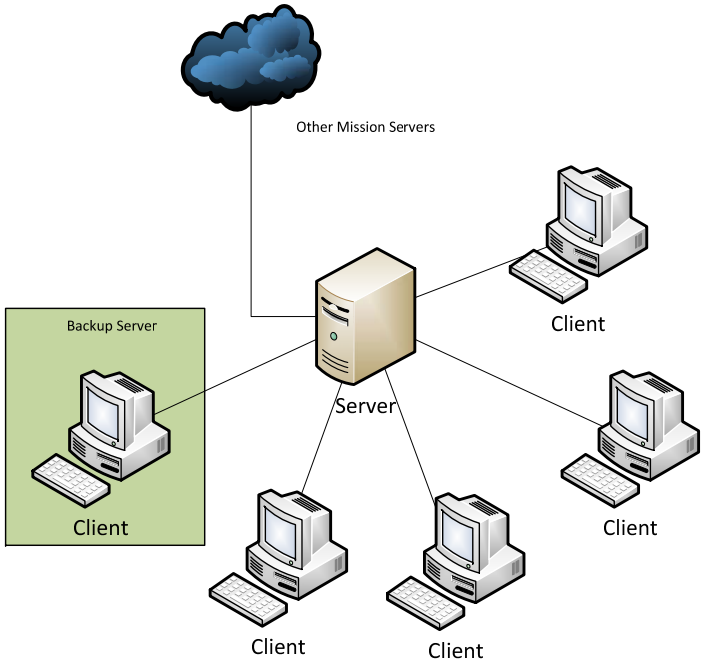
8. Approach Analysis III: Backup Server Alternative

- After considering the original architecture, the availability problem, all the scenarios in the utility tree and the patterns available we can conclude that the best option is to apply the **Backup Pattern**.



Resulting Architecture Evaluation (Additional Step)

- In this step we are going to apply the metrics that measure each NFR in order to evaluate the NFR fulfillment in the resulting architecture after applying the backup pattern.



- **Reliability Analysis:**
  - The probability of failure of the system is now the cumulative probability of a simultaneous failure both on the server and on the backup server. P(System) = P(Server) * P(Backup Server)

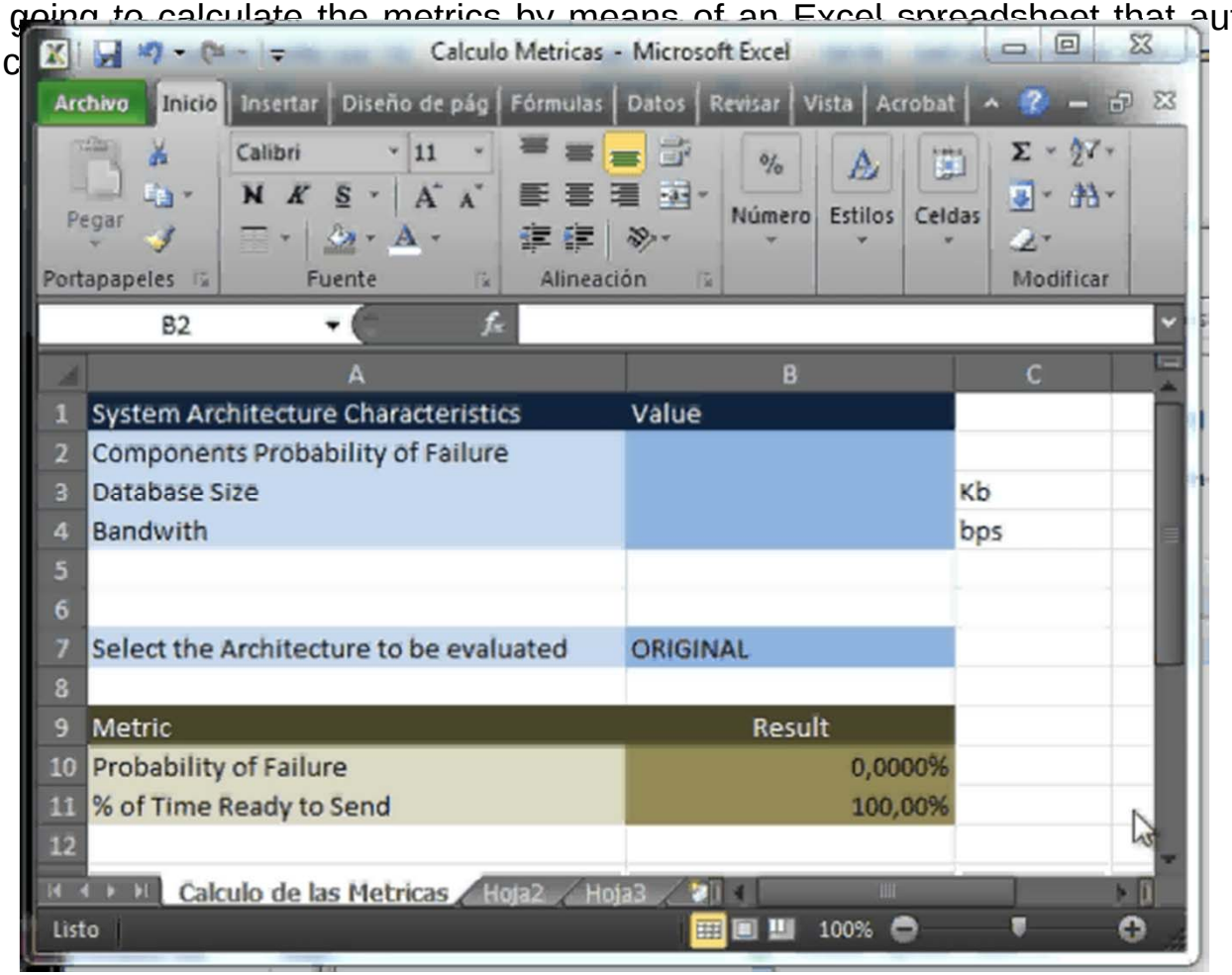
- **Performance Analysis:**
  - We should analyze the percentage of the time in which the communication channel is used with mirroring purposes.
    - The backup server mirrors once each 10 minutes

$$Mirroring\ Ussage = \frac{55Kb/9600}{10 * 60} * 100$$

$$\%\ Ready\ To\ Send = 100 - Mirroring\ Ussage$$

Resulting Architecture Evaluation (Additional Step)

- We are going to calculate the metrics by means of an Excel spreadsheet that automates the metrics c



Resulting Architecture Evaluation (Additional Step)

- We are going to calculate the metrics by means of an Excel spreadsheet that automates the metrics calculation.
  - **Reliability: Probability of Failure**

Value Obtained	0,25%
Does it fulfill the minimal threshold defined in NFR_001?	YES

  - **Performance: % of Time Ready To Send**

Value Obtained	99%
Does it fulfill the minimal threshold defined in NFR_001?	YES

9. Present Results

- The original architecture ***did not support the reliability scenario***
- Once the architecture has been modified (after the application of the **Backup Server Pattern)** it accomplishes both scenarios.

Reliability: Probability of failure	0.25%
Performance: Percentage of time that the communication channel is ready to send	99%

Architecture Evaluation Methods:

Introduction to ATAM

UNIVERSIDAD
POLITECNICA
DE VALENCIA