Design of software architecture

Project 2
OP4

ATAM method - Part 1

Euro Team

Alauzet Pierre, Ahvenniemi Mikko,

Colin Julien, Starck Benoit

KAIST

TABLE OF CONTENTS

ATAM ASSESSMENT STEPS

1. Present the ATAM
2. Present **FDIR** business drivers
3. Present **FDIR** architecture made through **ACME**
4. Identify **FDIR** architectural approaches made through **ACME**
5. Generate **FDIR** quality attribute utility tree
6. Analyze architectural **FDIR** approaches made through **ACME**
7. Brainstorm and prioritize scenarios of **FDIR requirement**
8. Analyze **FDIR** architectural approaches made through **ACME**
9. Present **FDIR** ATAM assessment results

1. ATAM PRESENTATION

- ***Architecture Tradeoff Analysis Method***

- Risk identification method to assess the consequences of architectural decisions in light of quality attribute requirements.
- The ATAM can be done early in the software development life cycle.
- It can be done relatively inexpensively and quickly (because it is assessing architectural design artifacts).
- The ATAM will produce analyses commensurate with the level of detail of the architectural specification.

1. ATAM PRESENTATION

- Three of the major goals of ATAM are to:
  - elicit and refine a precise statement of the architecture's driving quality attribute requirements
  - elicit and refine a precise statement of the architectural design decisions
  - evaluate the architectural design decisions to determine if they satisfactorily address the quality requirements

1. ATAM PRESENTATION

- The output of an ATAM is an out-brief presentation and/or a written report that includes the major findings of the evaluation.

- These are typically
  - the architectural styles identified
  - a "utility tree" - a hierarchic model of the driving architectural requirements
  - the tradeoff points
  - the sensitivity points
  - a set of identified risks
  - a set of identified non-risks

2. BUSINESS DRIVERS - CONTEXT

- *Client* : NASA
- *Users* : Spaceship crew and flight control can manually control the system

- *The problem* : Fault detection
  - Detected when monitored values are out-of-tolerance

- *The solution* : Fault protection system (FDIR)
  - Act when the spacecraft is going through an error or a fault
  - FDIR is a layered system. If a lower layer cannot resolve an issue it's forwarded to an upper layer. If the Issue cannot be resolved by the system. It's escalated to manual control.
  - Automatic system

2. BUSINESS DRIVERS - REQUIREMENTS

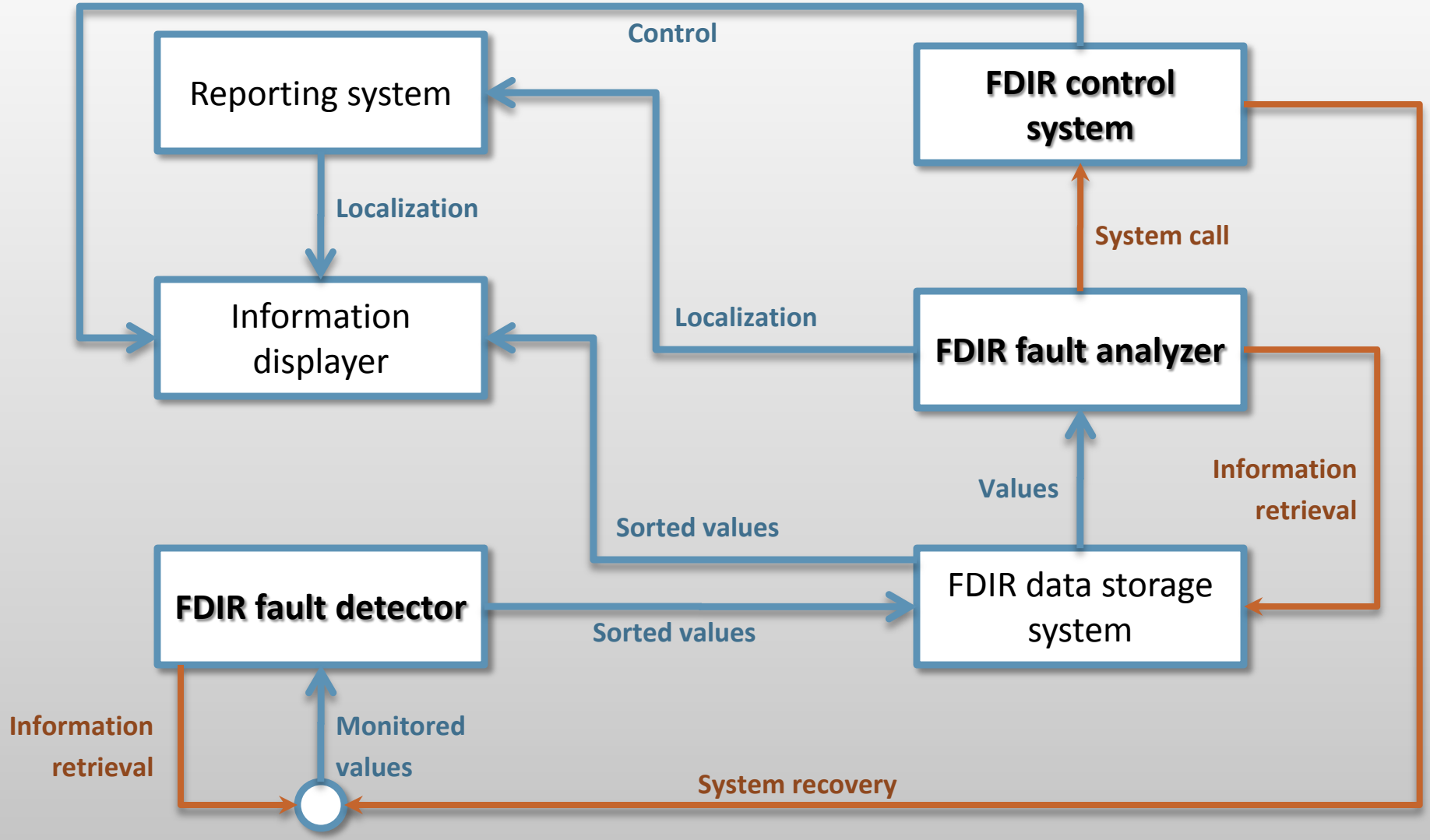
- Clients global needs :
  - Guarantee the completion of any time critical activities of the spaceship
  - Keep the control of the spacecraft with safety, observability & commandability

- Main qualities attributes :
  - ***Reliability*** : The system must be reliable in all operating conditions.
  - ***Availability*** : The system must not lock or stall when processing data. It must work asynchronously and must be available all the time.
  - ***Adaptability*** : FDIR has to be adaptable for manned and unmanned spacecraft.

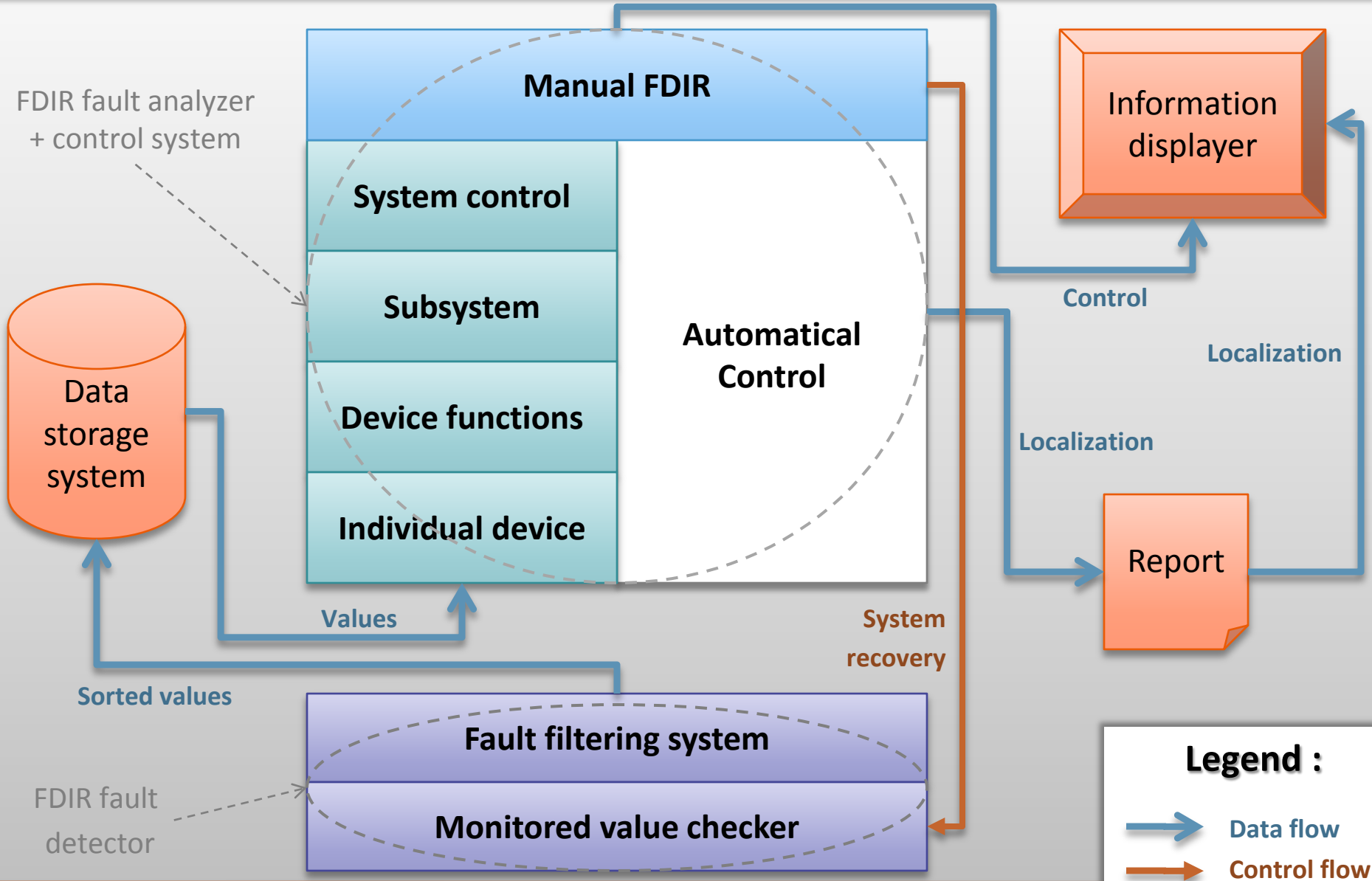
3. ARCHITECTURE SYSTEMS

- Fault detector
  - Monitored value checker
  - Fault filtering system
- Data storage system (logs, monitored values, reports, etc.)
- Fault analyzer
  - Layers system
    - individual device
    - device functions
    - Subsystem
    - system control
    - Manual FDIR
  - Automatical control
- Control system
- Reporting system
- Information displayer

3. OVERALL ARCHITECTURE



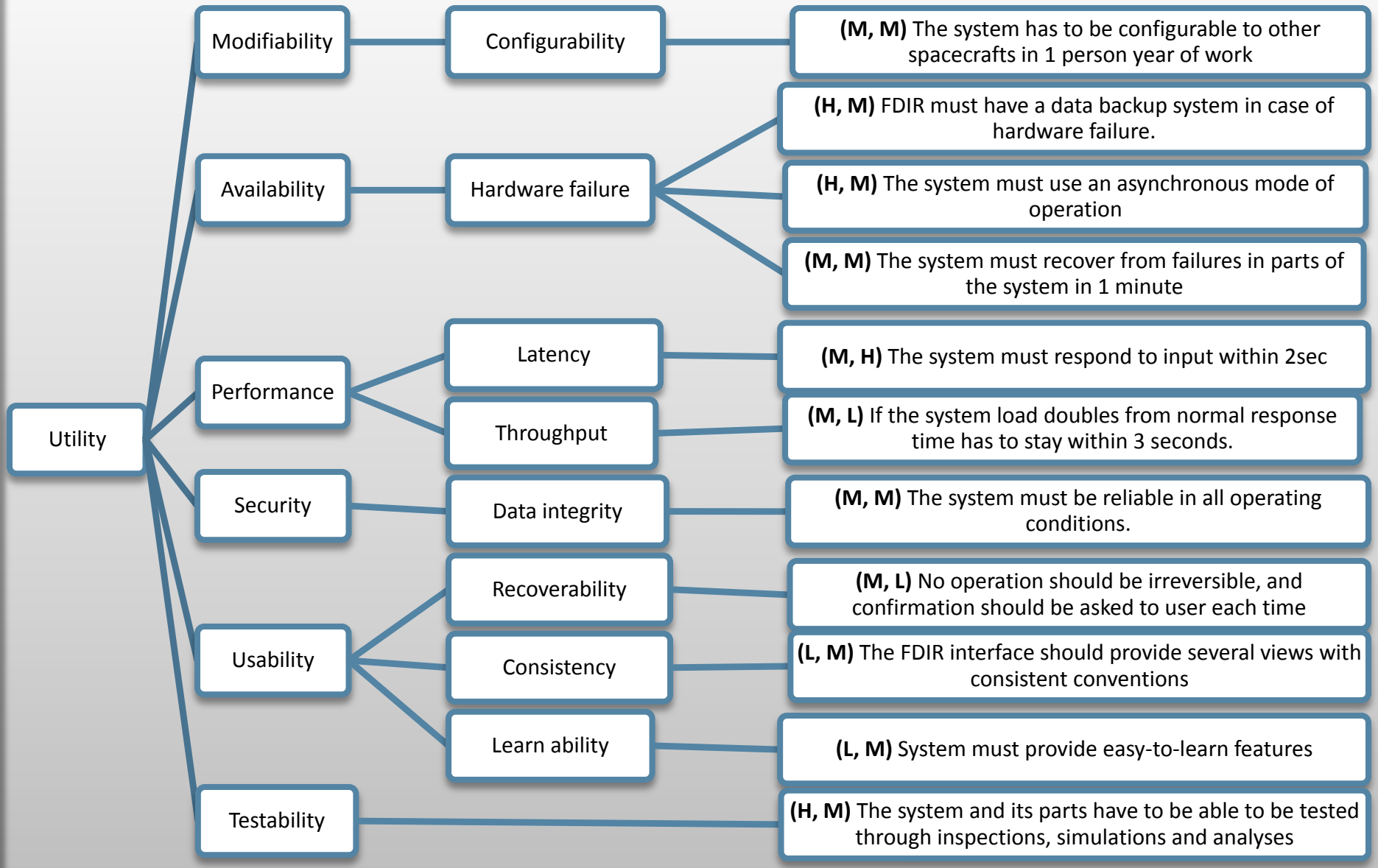
3. OVERALL ARCHITECTURE (CONT.)



4. ARCHITECTURAL APPROACH

- Event driven architecture
  - Devices subscribe to sub-systems which in turn listen to events broadcasted by the devices.
  - Such events can be for example “announce value” event
  - Choice based on identified quality attributes
    - Enables asynchronous processing
    - High potential for resilience in case of failure
    - Load can be balanced efficiently between systems
  - Architectural approach follows logically from system architecture

5. UTILITY TREE



5. UTILITY SCENARIOS

- Use case scenarios
  - No operation should be irreversible, and confirmation should be asked to user each time
  - The FDIR interface should provide several views with consistent conventions
- Growth scenarios
  - A new sub-system must be able to be installed in to the FDIR in 1 person day of work
- Exploratory scenarios
  - If the system load doubles from normal response time has to stay within 3 seconds.
  - The system has to be configurable to other spacecrafts in 1 person year of work

Thank you for attention !

Design of software architecture

Project 2
OP4

ATAM method - Part 1

Euro Team

Alauzet Pierre, Ahvenniemi Mikko,

Colin Julien, Starck Benoit

KAIST