Class #10

03. Designing Software architectures

Software Architectures

Master in Informatics Engineering

Cláudio Teixeira (claudio@ua.pt)

Agenda

- Designing Software Architectures
  - Systematic approaches:
    - Attribute driven design (ADD)

Designing Software Architectures:

Systematic approaches:

Attribute driven design

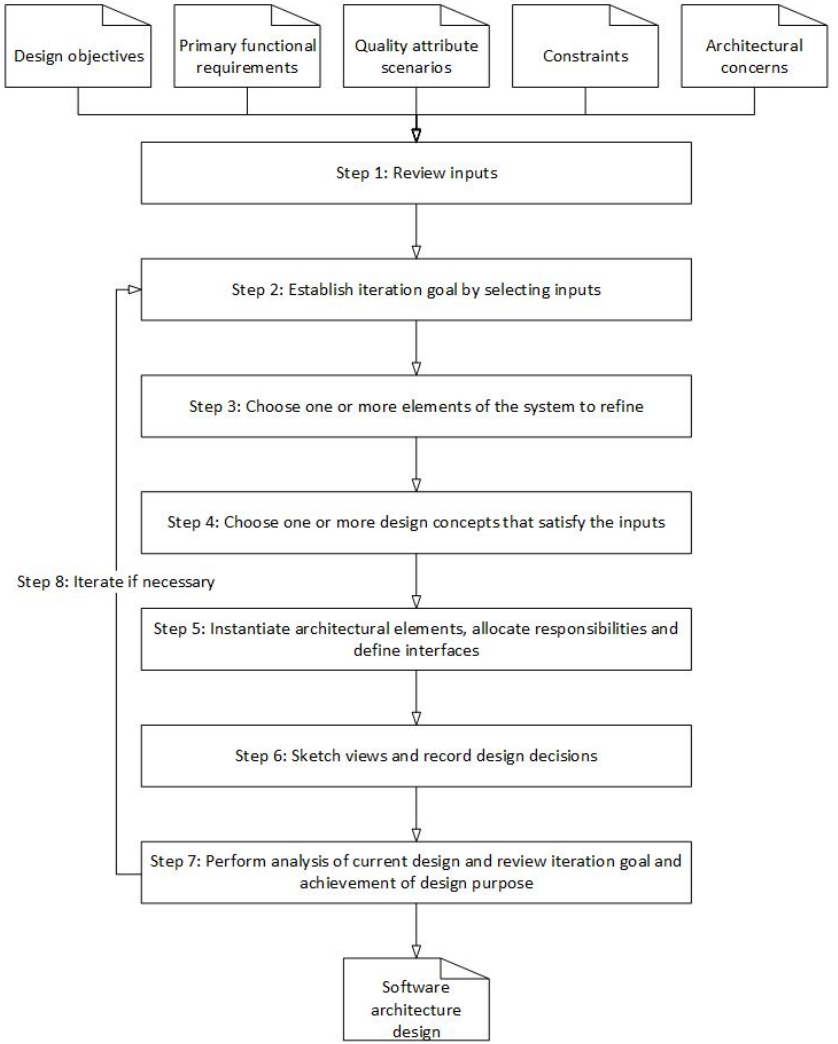
Attribute-driven design (ADD)

It is an iterative, organized, step-by-step method that can be followed during architectural design iterations.

This method pays particular attention to software quality attributes during the design process.

The ADD process is specifically focused on architecture design and, as such, doesn't cover the entire architectural life cycle. It may be combines with other methods to fill in these gaps.

- not included: gathering of architectural drivers, documenting the architecture, or evaluating the architecture once it is designed



EcoDrive App

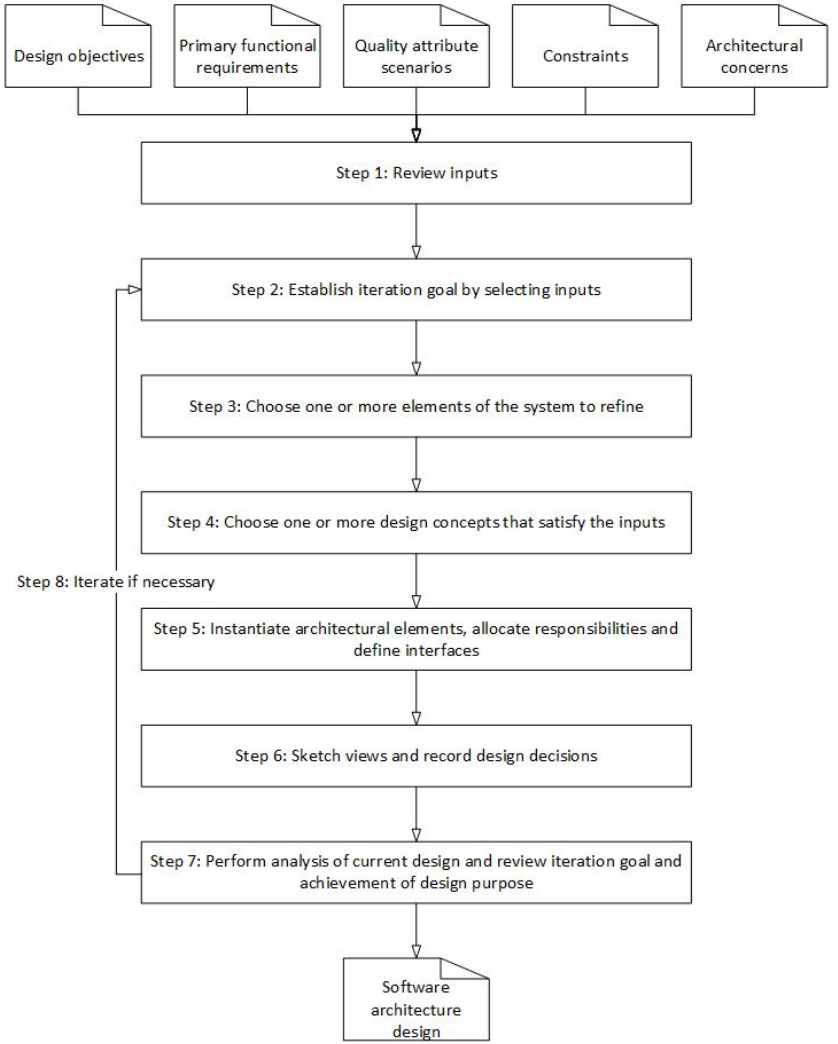
EcoDrive is a mobile application designed to encourage eco-friendly driving habits.

It collects data on driving patterns, provides feedback on fuel efficiency, and suggests routes and driving behaviors that reduce fuel consumption and emissions.

1 - Reviewing inputs

The inputs are the architectural drivers:

- Design objectives
- Primary functional requirements
- Quality attribute scenarios
- Constraints
- Architectural concerns
- Existing architecture (if any)



EcoDrive
Step 1

Inputs: User stories, functional requirements (track driving behavior, suggest eco-friendly routes), quality attribute requirements (high performance for real-time feedback, scalability to handle growing user base, usability for driver interaction).

Approach: Analyze requirements to identify key system functionalities and attribute priorities.

Results: Identification of core functionalities and prioritization of quality attributes:

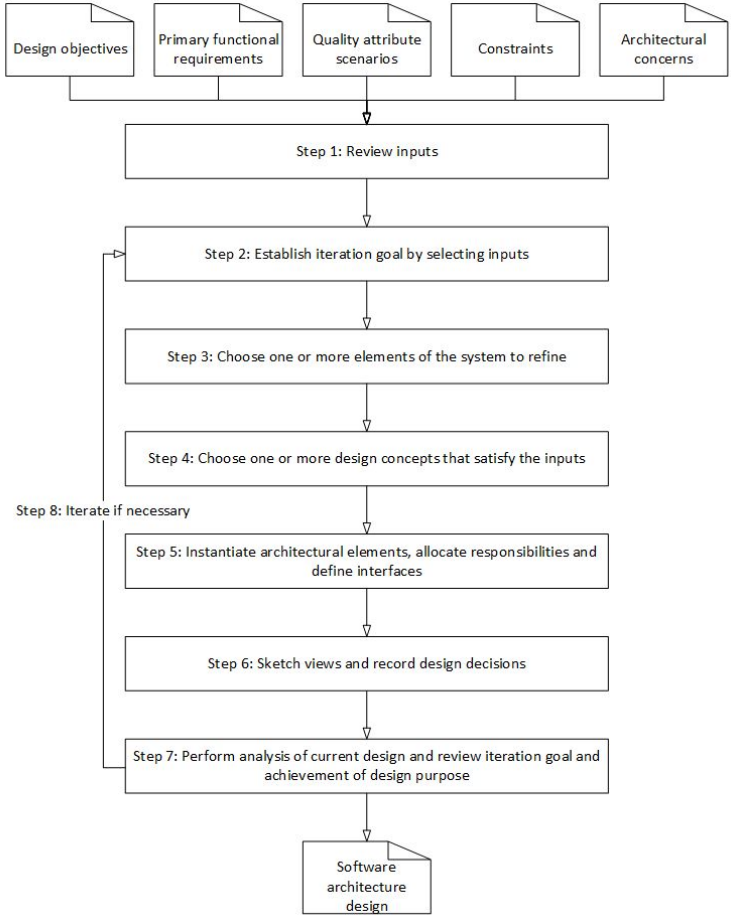
1. Performance
2. Scalability
3. Usability.

2 - Iteration goal & Input Selection

What design issue are we trying to solve in the iteration?

Each goal will be associated with one or more inputs.

The inputs, or architectural drivers, that are relevant to the goal are identified and will be the focus of the iteration.

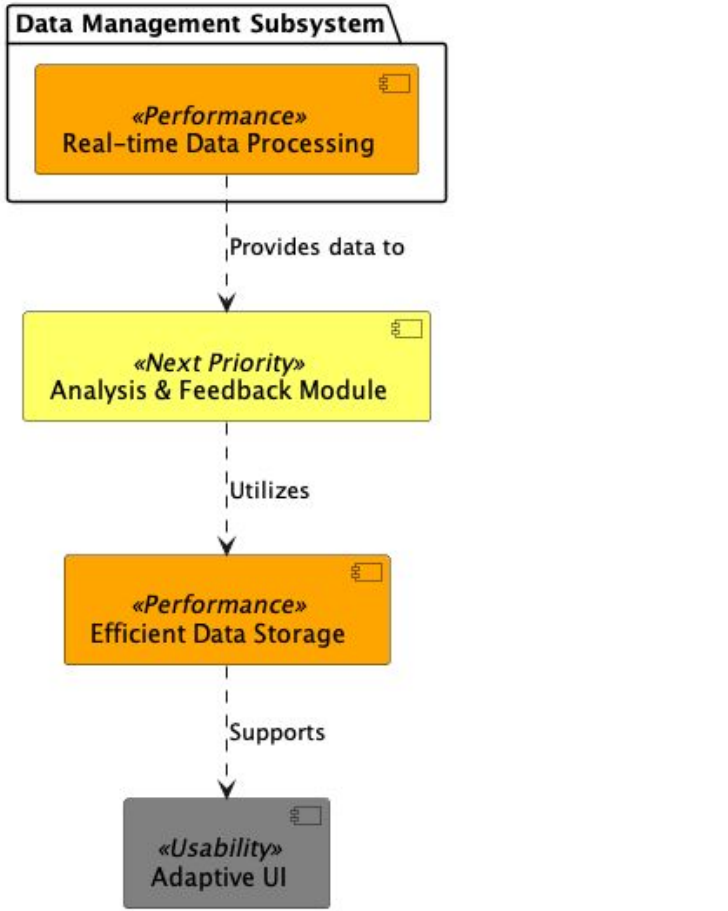


EcoDrive
Step 2: Iteration goal & Input Selection

Goal: Design the system's architecture to meet both functional and quality attribute requirements.

Approach: Select the most critical quality attribute (**performance**) to address in the first iteration.

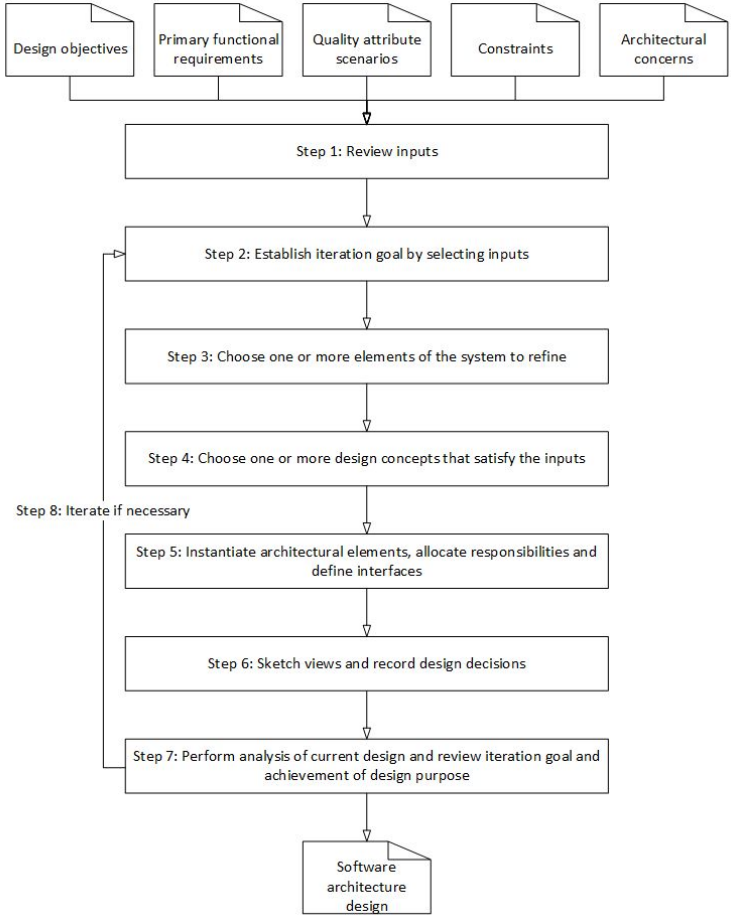
Results: Focus on ensuring real-time feedback on driving patterns to users.

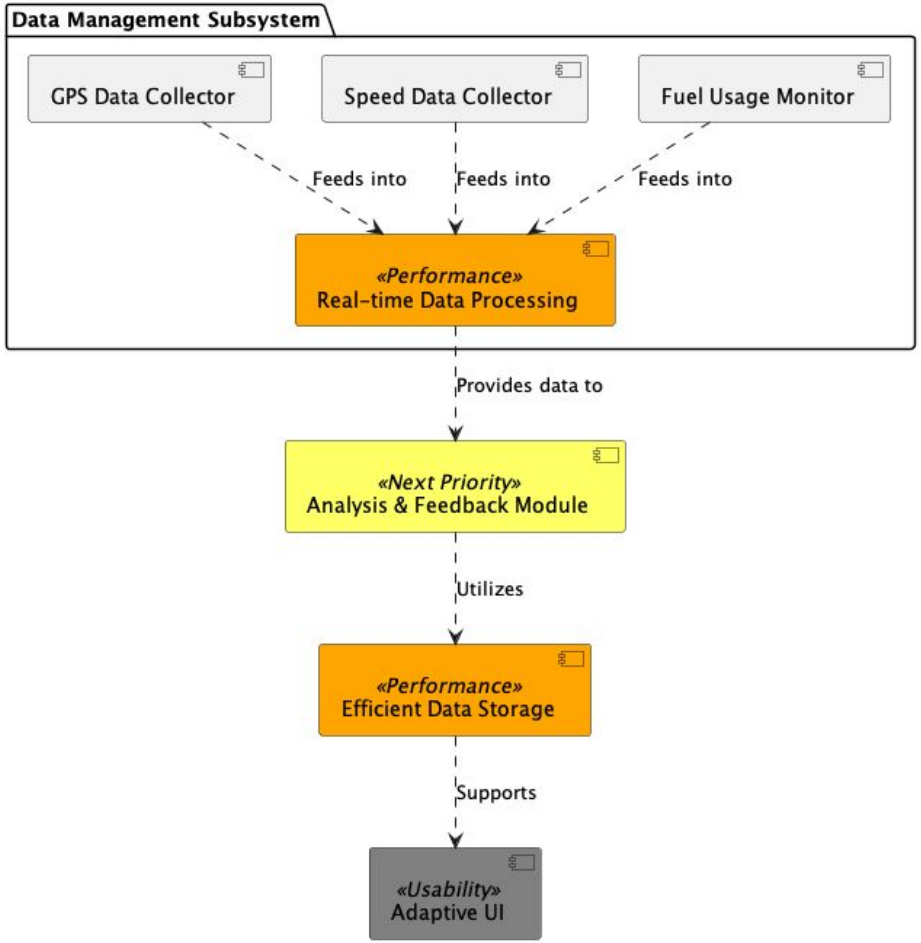


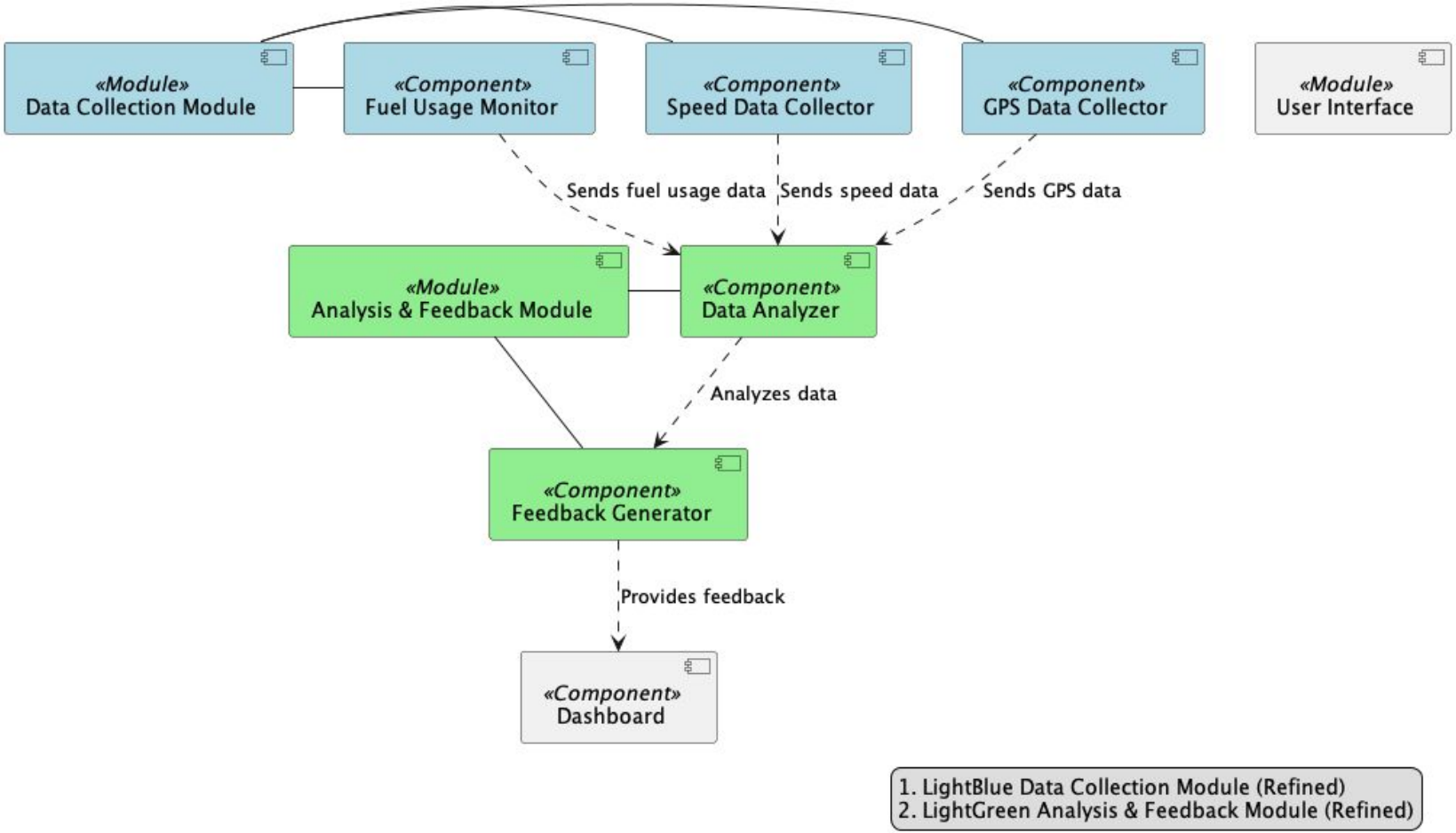
Performance-related features are highlighted in orange (Primary Focus).
Scalability and Usability features are shown in gray (Not prioritized in this iteration).
Next Priority Features are indicated in yellow.

3 - Choose one or more elements of the system to refine

Select the various elements that we want to decompose and where to focus on this iteration



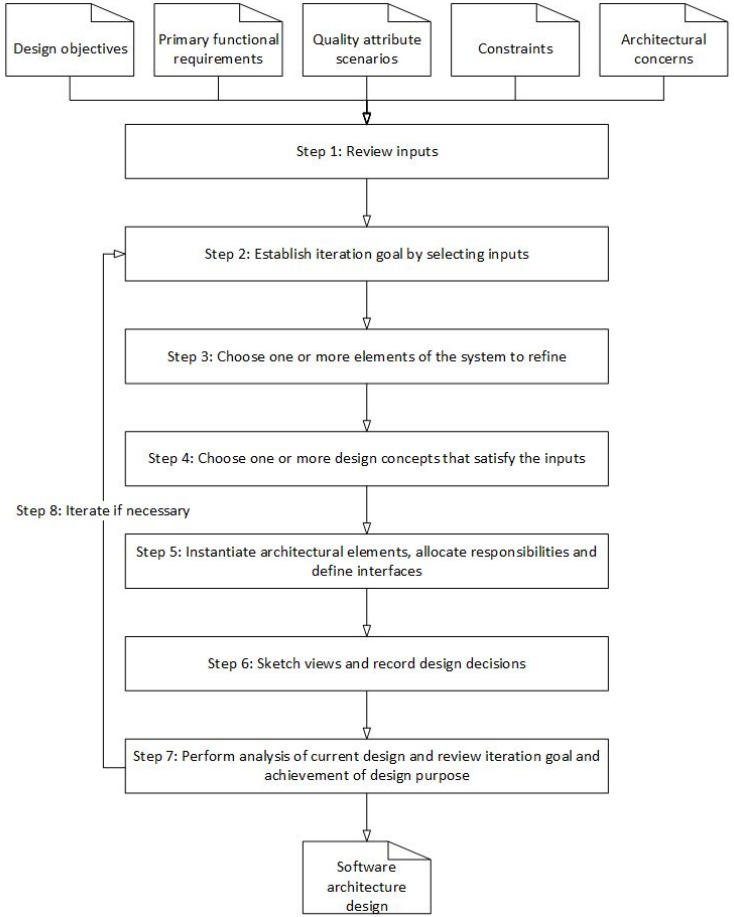

Data Management Subsystem
GPS Data Collector
Speed Data Collector
Fuel Usage Monitor
Feeds into
Feeds into
Feeds into
«Performance»
Real-time Data Processing
Provides data to
«Next Priority»
Analysis & Feedback Module
Utilizes
«Performance»
Efficient Data Storage
Supports
«Usability»
Adaptive UI
Performance-related features are highlighted in orange (Primary Focus).
Scalability and Usability features are shown in gray (Not prioritized in this iteration).
Next Priority Features are indicated in yellow.


«Module»
Data Collection Module
«Component»
Fuel Usage Monitor
«Component»
Speed Data Collector
«Component»
GPS Data Collector
«Module»
User Interface
Sends fuel usage data
Sends speed data
Sends GPS data
«Module»
Analysis & Feedback Module
«Component»
Data Analyzer
Analyzes data
«Component»
Feedback Generator
Provides feedback
«Component»
Dashboard
1. LightBlue Data Collection Module (Refined)
2. LightGreen Analysis & Feedback Module (Refined)

4 - Choosing one or more design concepts

Once elements have been selected for decomposition, we need to select one or more design concepts that can be used to meet the iteration goal and satisfy the inputs (architectural drivers).

Design concepts refer to design principles and solutions such as architecture patterns, reference architectures, tactics, and externally developed software.



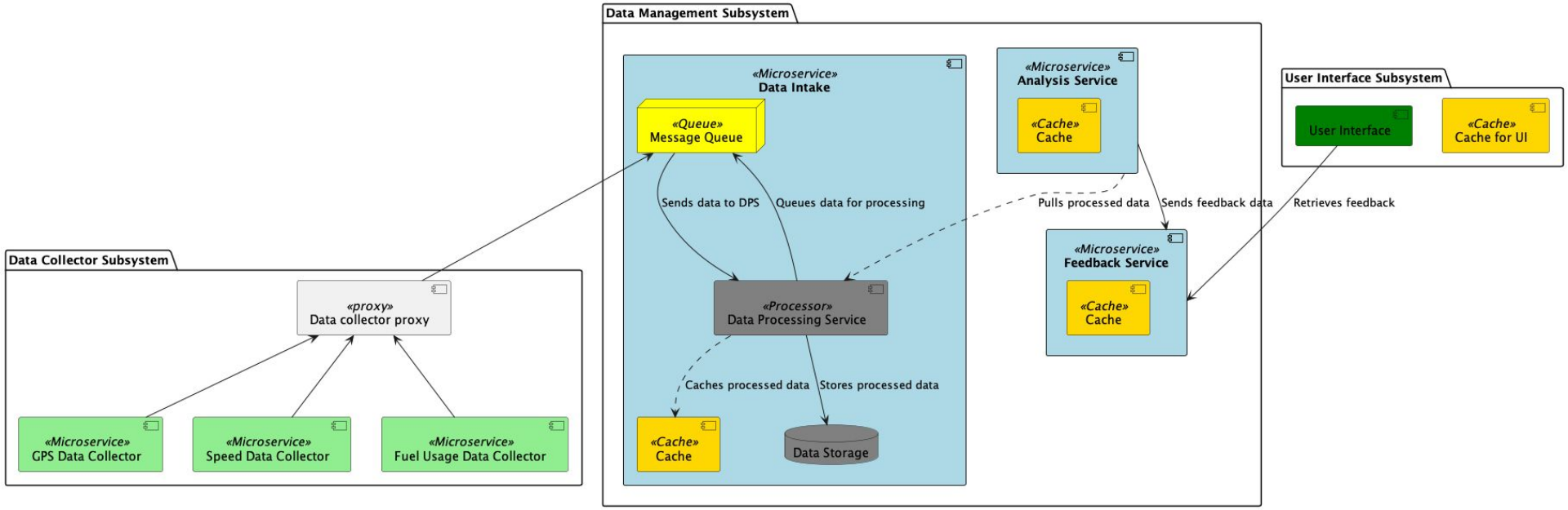
EcoDrive
Step 4: Choosing Design Concepts

Iteration Goal: performance
Focus: real-time feedback on driving patterns

Design Concepts: Microservices architecture for scalability, caching for performance, and intuitive UI design for usability.

Approach: Evaluate design concepts against quality attributes to select the most suitable ones.

Results: Adoption of microservices for flexibility and scalability, implementation of caching to enhance performance.

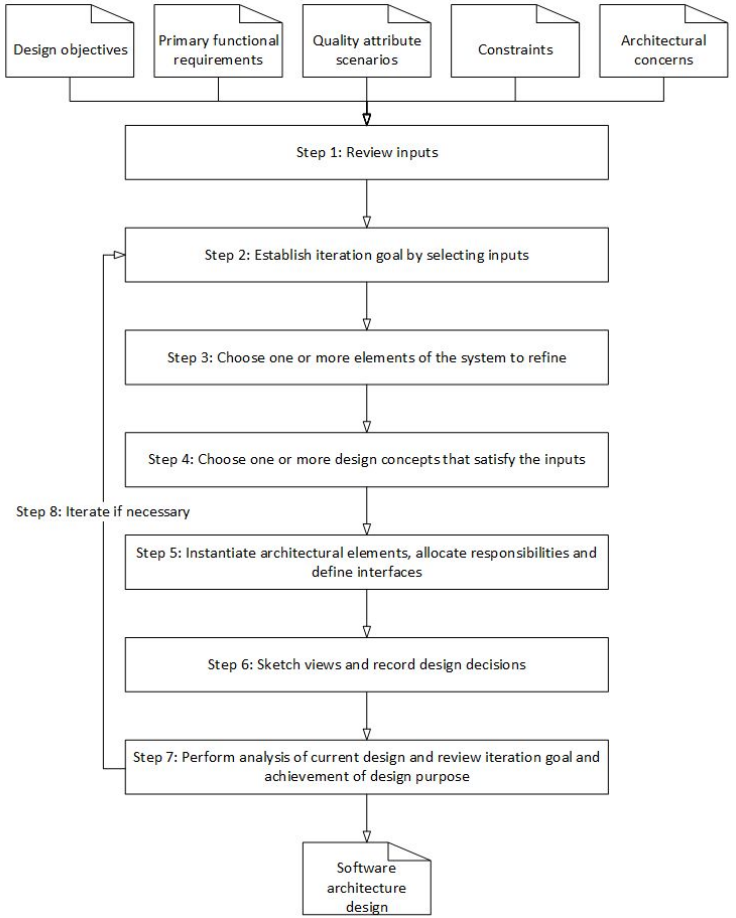

Data Management Subsystem
«Microservice»
Data Intake
«Queue»
Message Queue
Sends data to DPS
Queues data for processing
«Processor»
Data Processing Service
Caches processed data
Stores processed data
«Cache»
Cache
Data Storage
«Microservice»
Analysis Service
«Cache»
Cache
Pulls processed data
Sends feedback data
«Microservice»
Feedback Service
«Cache»
Cache
User Interface Subsystem
User Interface
«Cache»
Cache for UI
Retrieves feedback
Data Collector Subsystem
«proxy»
Data collector proxy
«Microservice»
GPS Data Collector
«Microservice»
Speed Data Collector
«Microservice»
Fuel Usage Data Collector

5 - Instantiate architectural elements, allocating responsibilities, and defining interfaces

Based on the design concepts that can be leveraged for this iteration, analysis is performed so that details can be provided regarding the responsibilities for the elements being decomposed, along with the public interfaces of those elements that will be exposed.

Each element being decomposed (parent element) may yield one or more child elements.

- By considering the responsibilities of the parent element, we can assign responsibilities to the various child elements.
- All of the responsibilities of the parent element are considered, whether or not they are architecturally significant.



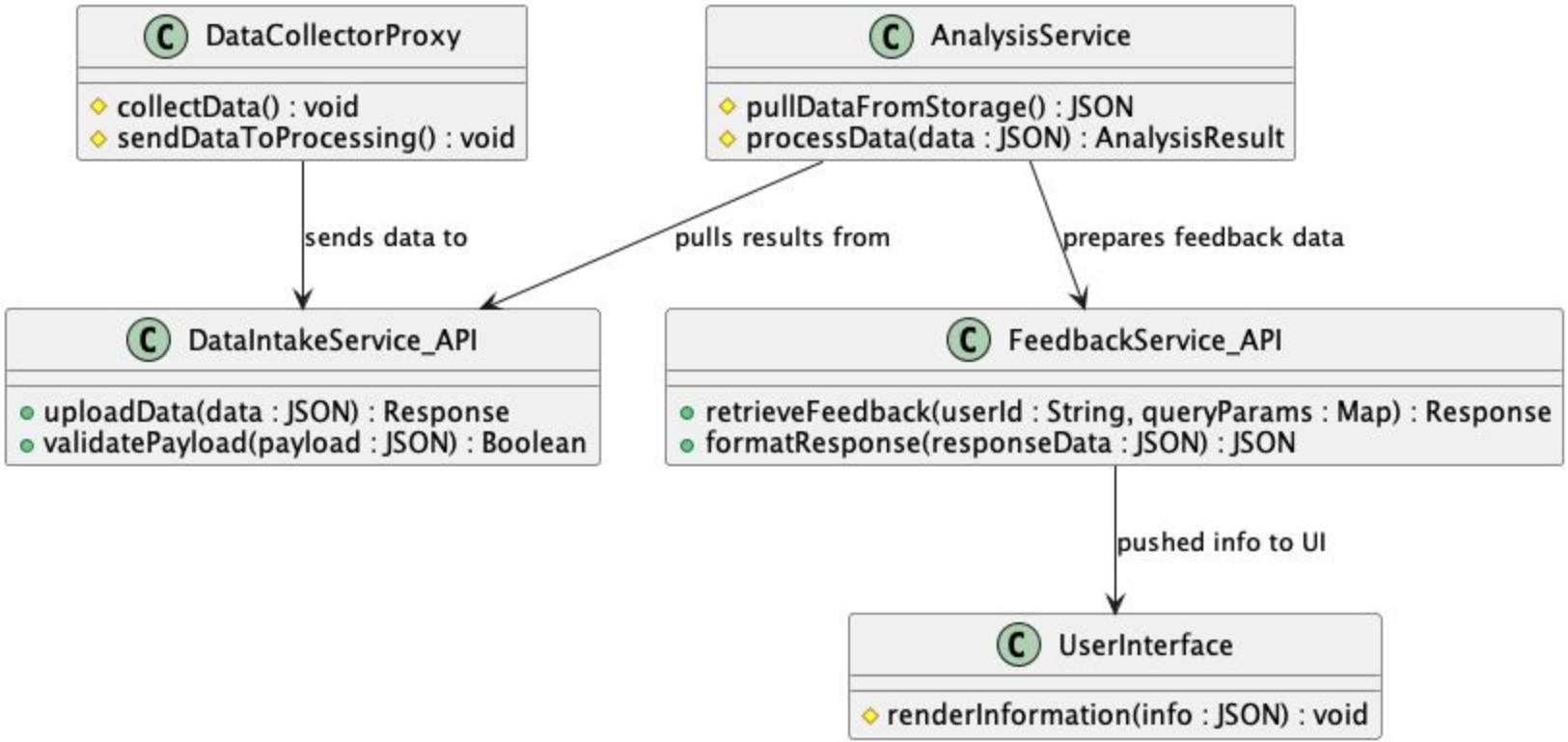
EcoDrive
Step 5: Instantiate Architectural Elements

Iteration Goal: performance
Focus: real-time feedback on driving patterns
How: microservices & cache

Architectural Elements: Data Collection Service, Analysis Service, Feedback Service, User Interface.

Approach: Allocate responsibilities to each service; define RESTful interfaces for inter-service communication.

Results: Clear separation of concerns, with specific responsibilities allocated, enabling independent scaling and enhancement.

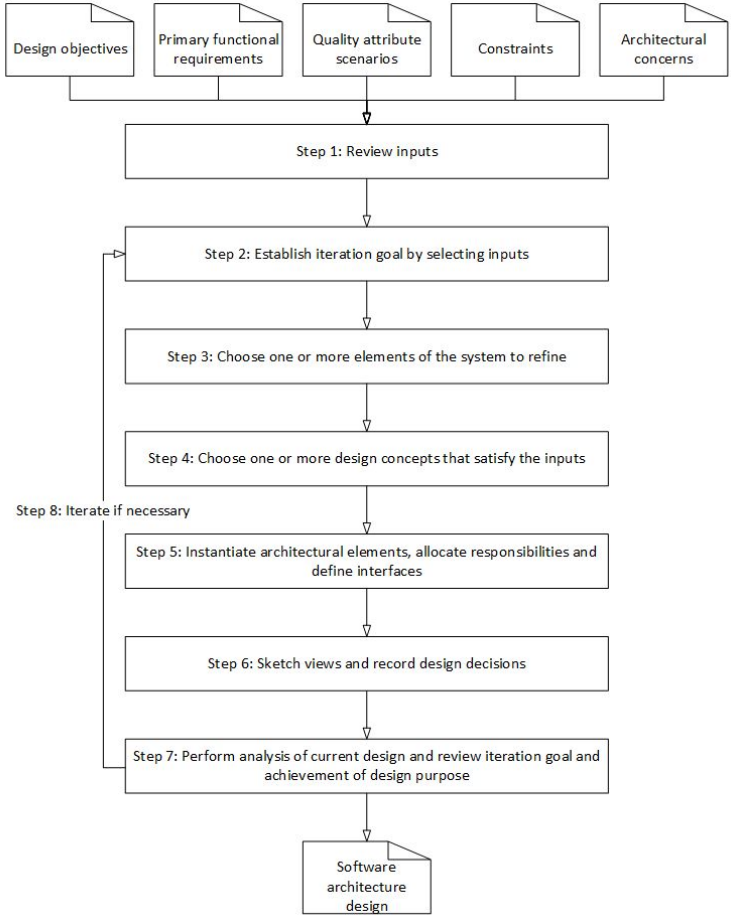

DataCollectorProxy
collectData() : void
sendDataToProcessing() : void
AnalysisService
pullDataFromStorage() : JSON
processData(data : JSON) : AnalysisResult
sends data to
pulls results from
prepares feedback data
DataIntakeService_API
uploadData(data : JSON) : Response
validatePayload(payload : JSON) : Boolean
FeedbackService_API
retrieveFeedback(userId : String, queryParams : Map) : Response
formatResponse(responseData : JSON) : JSON
pushed info to UI
UserInterface
renderInformation(info : JSON) : void

6 - Sketching views and recording design decisions

Views should be sketched recording the solution designed so that it can be communicated.

All of the design decisions are documented, along with design rationale.

The artifacts created in this step can simply be sketches and do not have to be the formal, detailed software architecture views.



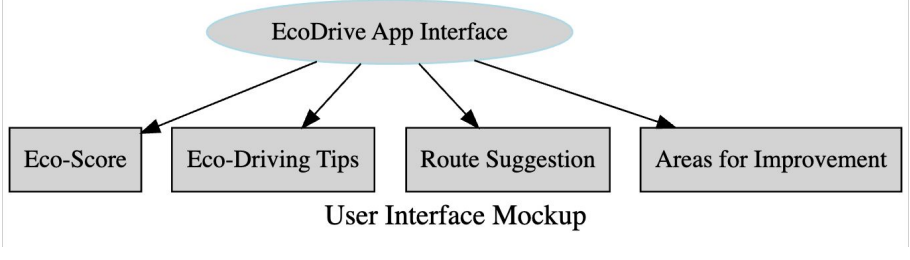
EcoDrive
Step 6: Sketching Views & Recording Decisions

Iteration Goal: performance
Focus: real-time feedback on driving patterns
How: microservices & cache

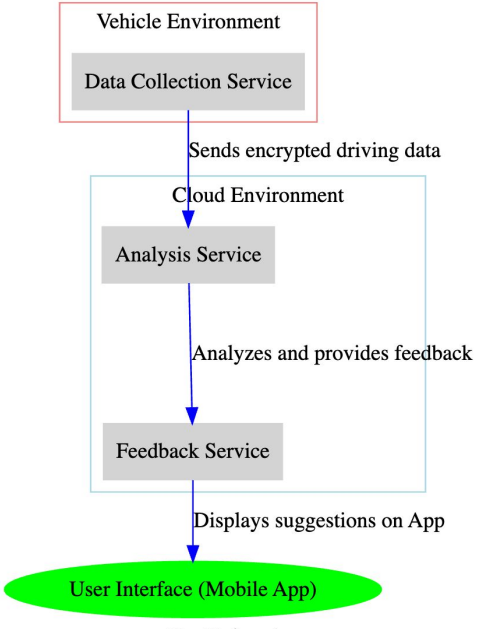
Views: Deployment view showing microservices distribution, sequence diagram for data flow, and UI mockups.

Approach: Use UML diagrams and mockups to visualize system structure and interactions.

Design Decisions: Use of cloud-based services for data collection and analysis, adoption of a responsive design for the UI.



User Interface Mockup

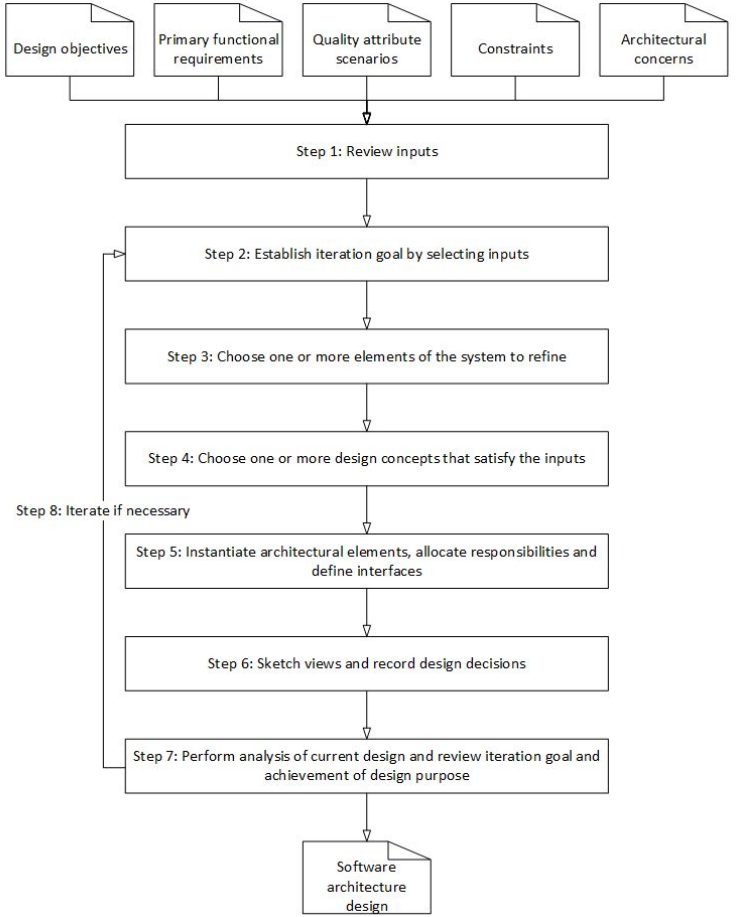


EcoDrive App

7 - Performing analysis of current design and reviewing the iteration goal and design objectives

The software architect and other team members should analyze the current design, ensuring that it is correct and satisfies the iteration goal and architectural drivers that were established for the iteration.

If needed, go to step 2.



EcoDrive
Step 7: Performing Analysis & Review

Iteration Goal: performance
Focus: real-time feedback on driving patterns
How: microservices & cache

Analysis: Assess the architecture's alignment with performance, scalability, and usability goals.

Approach: Perform load testing for performance, scalability testing for the microservices architecture, and usability testing with potential users.

Review Goals: Ensure real-time feedback is effectively provided, the system can scale, and the UI meets user expectations.

EcoDrive
Next Steps 2

- Quality Attributes Focus: Security and privacy
  - Integrate secure data transmission protocols, encrypt sensitive data, and ensure user data privacy through access controls and data anonymization
  - Potential Changes:
    - Refine the Data Collection Service to include encryption of data in transit.
    - Implement authentication and authorization mechanisms for accessing the Analysis and Feedback Services.
    - Conduct a security review of all components and services.
- Quality Attributes Focus: Usability and user engagement.
  - Enhance the User Interface with more interactive elements, gamify the feedback to encourage eco-friendly driving, and personalize suggestions.
  - Potential Changes:
    - Redesign the UI to be more intuitive and engaging, incorporating user feedback.
    - Develop algorithms for personalized driving tips based on user behavior analysis.
    - Integrate gamification elements like rewards or badges for achieving eco-driving goals.
- ...

Smart Home Energy Management System Design using ADD

Group assignment
90 min

https://docs.google.com/document/d/1UIoT4-p_3pcpsWL_8F08pI8-vucLCJ7AlZtHK0qtOgA/edit

Bibliography

- https://learning.oreilly.com/library/view/software-architects-handbook/9781788624060/9236a39f-8469-4632-b013-4ed9582aab77.xhtml
- https://learning.oreilly.com/library/view/software-architecture-in/9780136885979/part02.xhtml
- https://learning.oreilly.com/library/view/software-architects-handbook/9781788624060/0adb7344-e386-4f19-8493-55d6d02c2345.xhtml
- https://learning.oreilly.com/library/view/software-architecture-the/9781492086888/ch01.html