



CARGO: Context-Aware Reasoning using Goal-Oriented

Mira Vrbaski, Gunter Mussbacher, Dorina Petriu



Context-Aware Systems

- Context aware systems are:
 - highly adaptable** systems;
 - system operation adapts to the current context at runtime without explicit user intervention;
 - goal: increase runtime usability and effectiveness.
- Examples of context aware systems:
 - mobile systems;
 - hospitals and homes for the elderly;
 - smart homes.
- Characteristics:
 - Significant use of wireless appliances, smart phones, PDAs, sensors.

What is Context?

- Context** is a very broad concept.
- Definition: Context is any **information that can be used to characterize the situation of an entity**. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves.
- Context includes the following environmental aspects:
 - computing environment:** available processors, devices accessible for user input and display, network capacity, connectivity, and costs of computing;
 - user environment:** location, collection of nearby people, and social situation;
 - physical environment:** lighting and noise level.

Context Reasoning

- Context reasoning** is an important component of context awareness. Based on previously defined **rules**, which are most of the time **logic-based**, the system will perform different actions at runtime according to the current context values, without direct human intervention.

Rule Base and Rule Engine

- In knowledge-based systems, human knowledge and reasoning is captured and stored in the form of complex rules. Typically, rules are described using declarative languages.
- Rules are stored in a **Rule Base** and processed by a Rule (Inference) Engine. At any point in time, the **Rule Engine** evaluates the rule conditions based on the current context values and determines which rules are eligible to be fired (executed).
- While rules excel in filtering out unsuitable solutions based on clear criteria, it is difficult to rank suitable solutions based on vague, qualitative criteria with a rule-based approach.

Context-Aware Goal Modeling

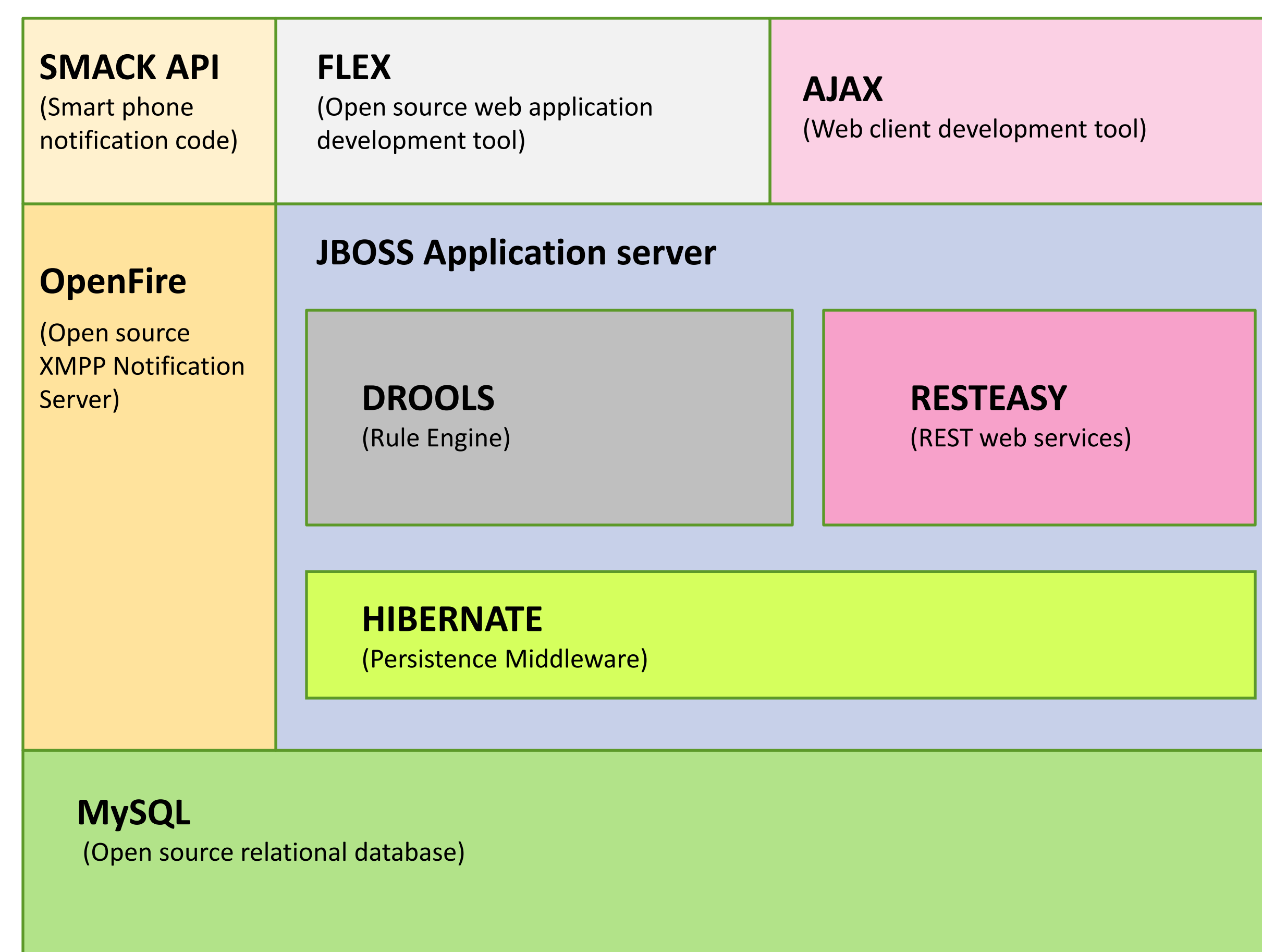
- Goal modeling is an early requirements technique that focuses on the modeling of:
 - stakeholders** and their high-level **goals**;
 - solutions** and their **impact** on achieving the goals;
 - key performance indicators**, i.e., real-world measures that characterize more precisely the proposed solutions.
- Goal models can be evaluated:
 - assessment of a proposed solution results in **satisfaction values** for stakeholders;
 - trade-off analysis** compares the proposed solutions taking the satisfaction values of all stakeholders into account;
 - the starting point of an evaluation are the key performance indicators.
- In context-aware systems:
 - a **Goal Engine** can **complement** a logic-based Rule Engine by allowing a more holistic assessment of the context while taking the goals of many stakeholders into account;
 - key performance indicators capture **context**-related information, making it available for reasoning at the goal level.
 - with support for scenario modeling and execution, a context-aware system can be described at a very high level of abstraction.

Tool Support

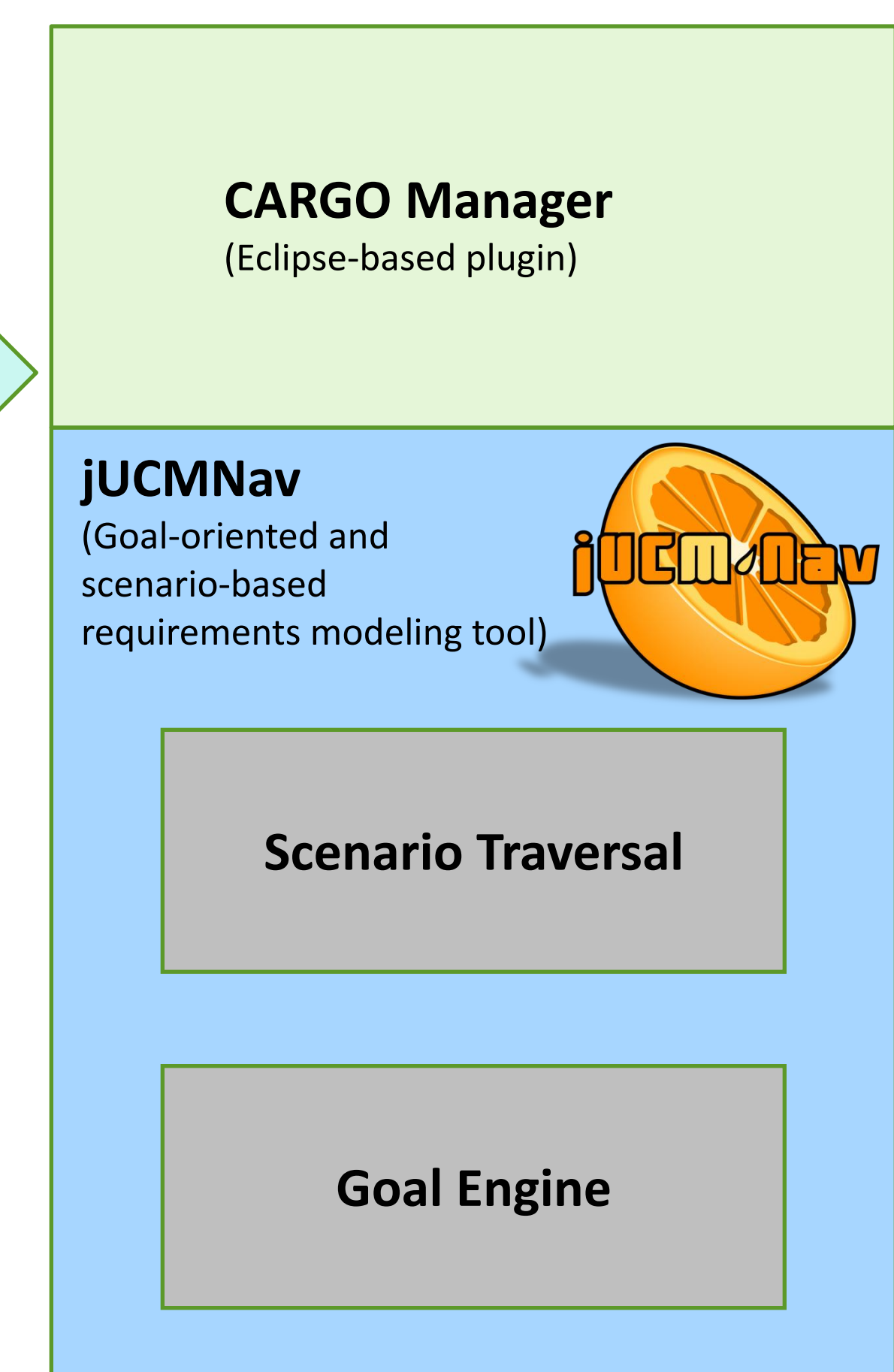
- jUCMNav** can be used for context-aware goal modeling:
 - supports the **User Requirements Notation**, an international standard for the elicitation, analysis, specification, and validation of requirements;
 - goal-oriented and scenario-based modeling;
 - scenario execution with traversal mechanism;
 - goal model evaluation algorithms;
 - open-source Eclipse plugin.

CARGO System Architecture

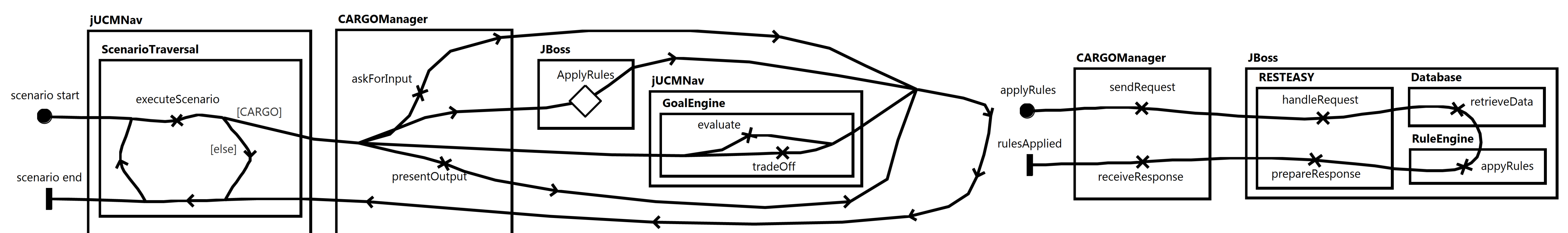
Context Awareness Framework with RULE ENGINE...



... and GOAL ENGINE



CARGO System Behavior



Example: Hospital Food Service

