

Quality starts by defining your goals !

Cediti SA

Copyright © 2003



www.cediti.be



www.objectiver.com

- ▶ 1. Why do we need Requirements Engineering ?
- 2. The Objectiver approach
- 3. Deeper inside the Objectiver approach
- 4. References
- 5. Demo

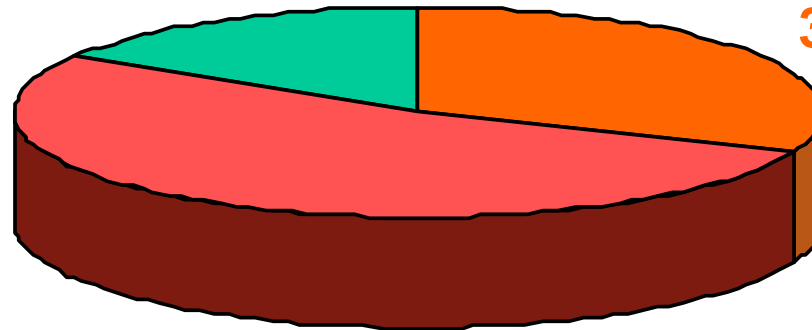
Requirements Engineering – Why ?

Project completed on time, on budget, and with all functions originally specified

success
16%

Project cancelled before completion or never implemented

failure
31%



challenged
53%

Project completed *but...* over budget, late, and with fewer functions than initially specified

Top 10 Reasons for Project Failures

- Lack of User Input**
- Incomplete Reqs & Specs**
- Changing Reqs & Specs**
- Lack of Executive Support 8 %
- Technology Incompetence 7 %
- Lack of Resources 6 %
- Unrealistic Expectations**
- Unclear Objectives**
- Unrealistic Time Frames 4 %
- New Technology 3 %

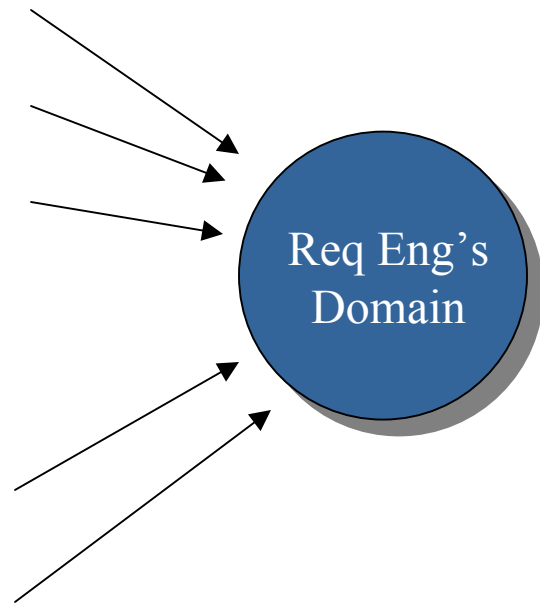
13 %

12 %

11 %

6 %

5 %



- 1. Why do we need Requirements Engineering ?
- ▶ 2. The Objectiver approach
- 3. Deeper inside the Objectiver approach
- 4. References
- 5. Demo

Objectiver goal-driven approach to requirements engineering

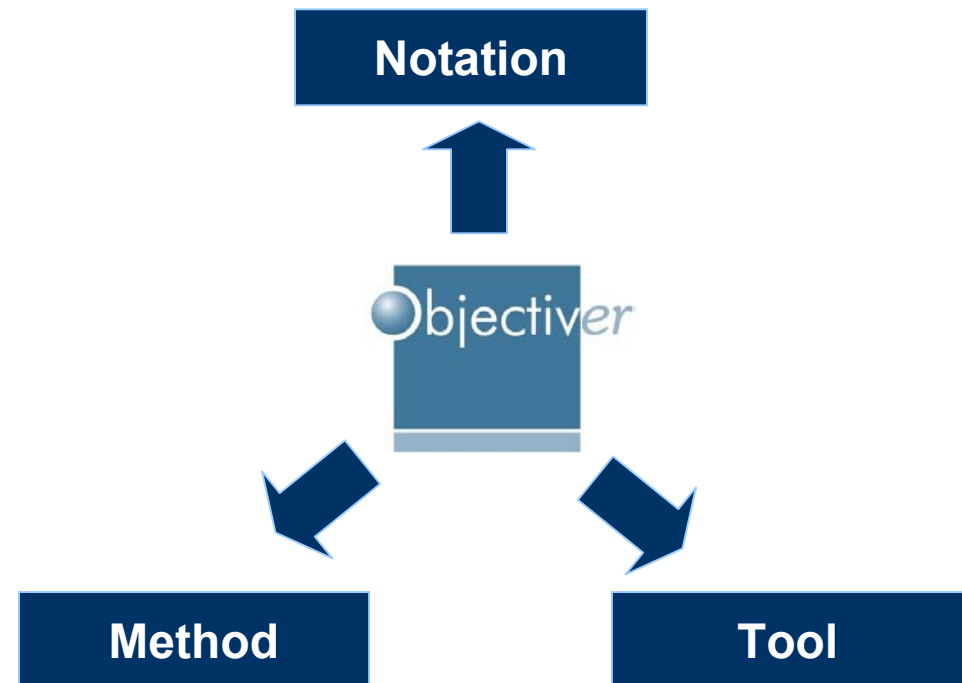
Objectiver

Start by studying the **problem** and its **environment** rather than by specifying the solution straight away

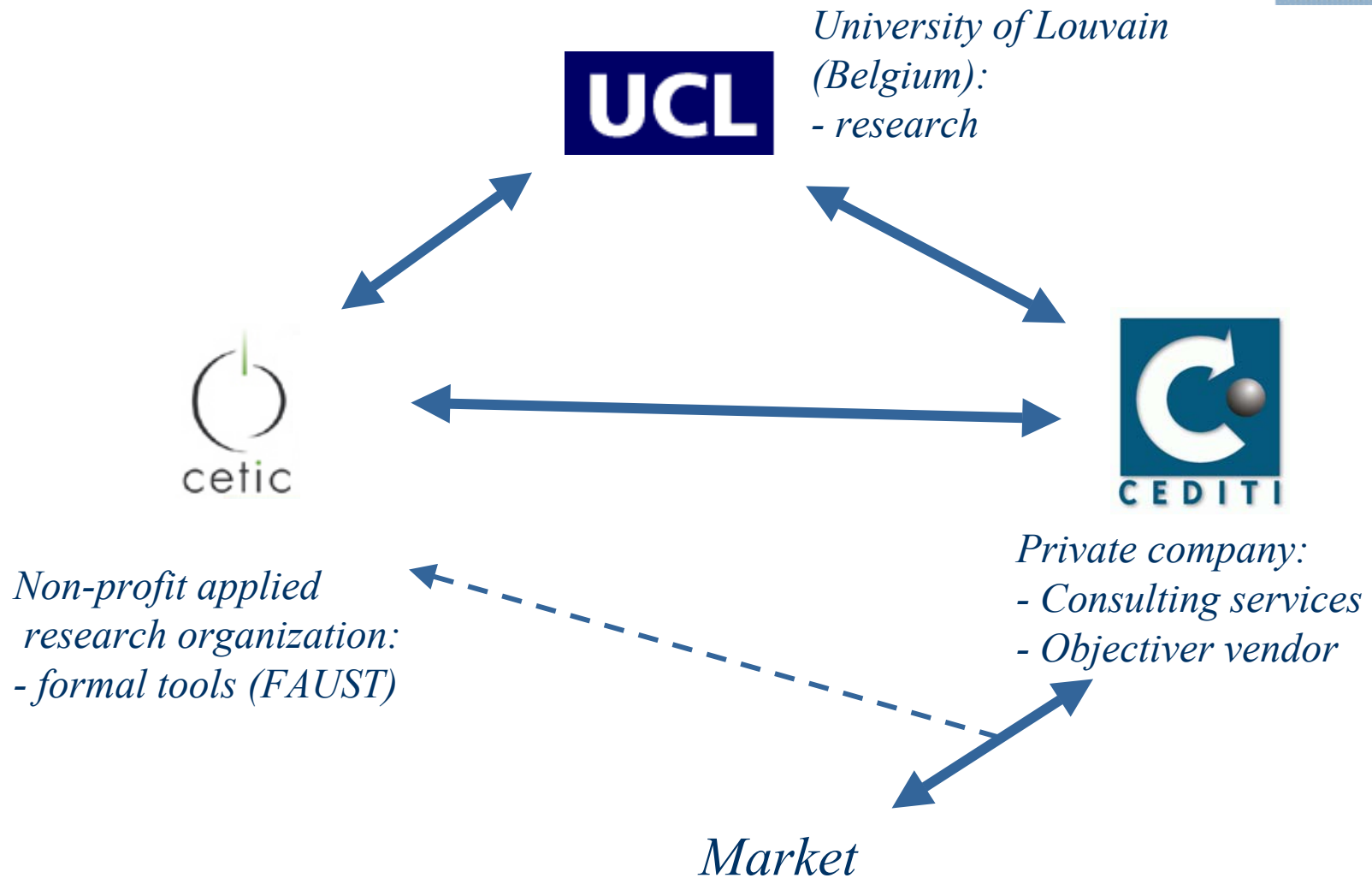
Link the project requirements to the **business strategy**

Establish **traceability** from business processes to business goals

Find out how changes to business goals shall impact your processes and project requirements

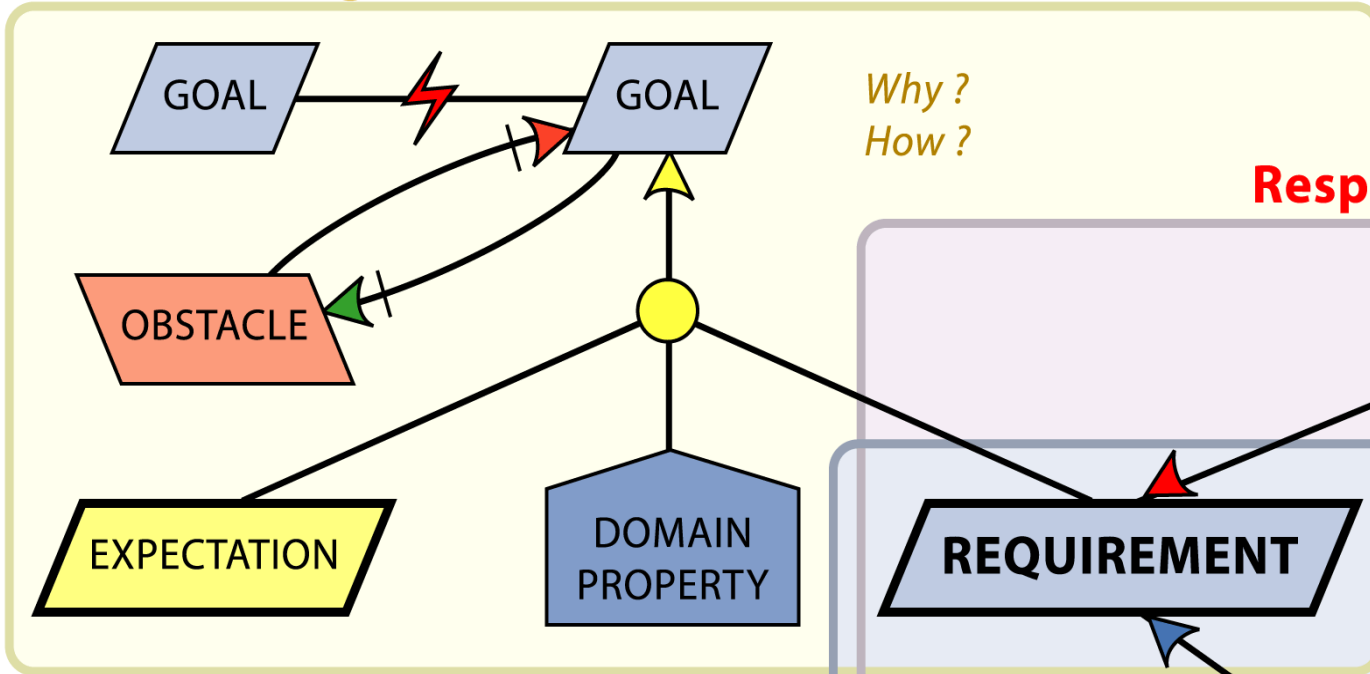


A strong scientific background

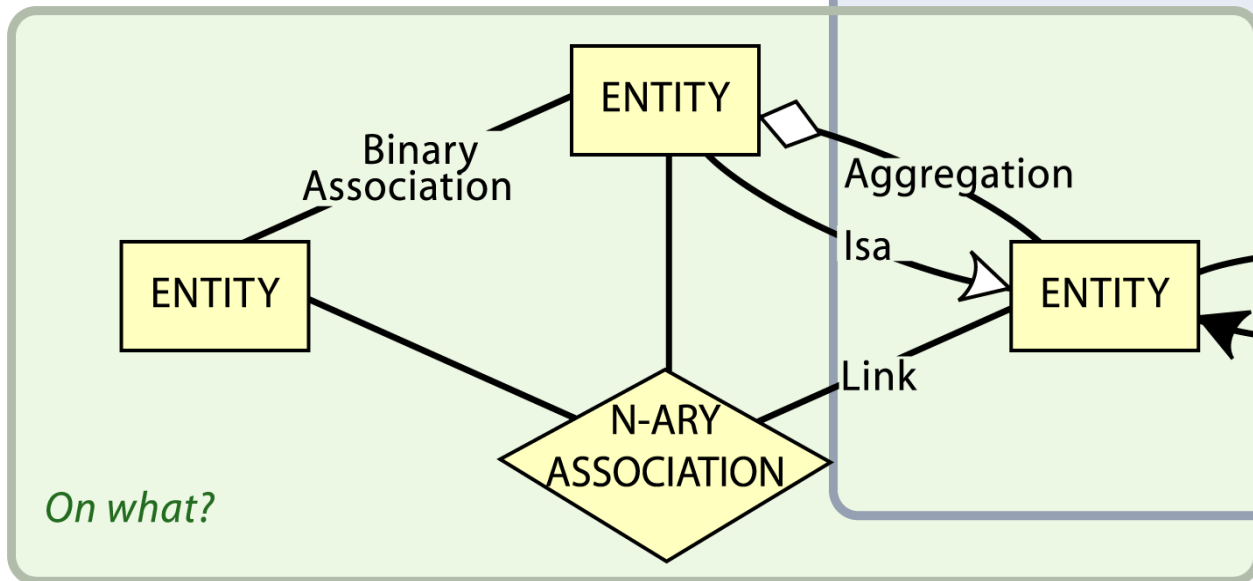
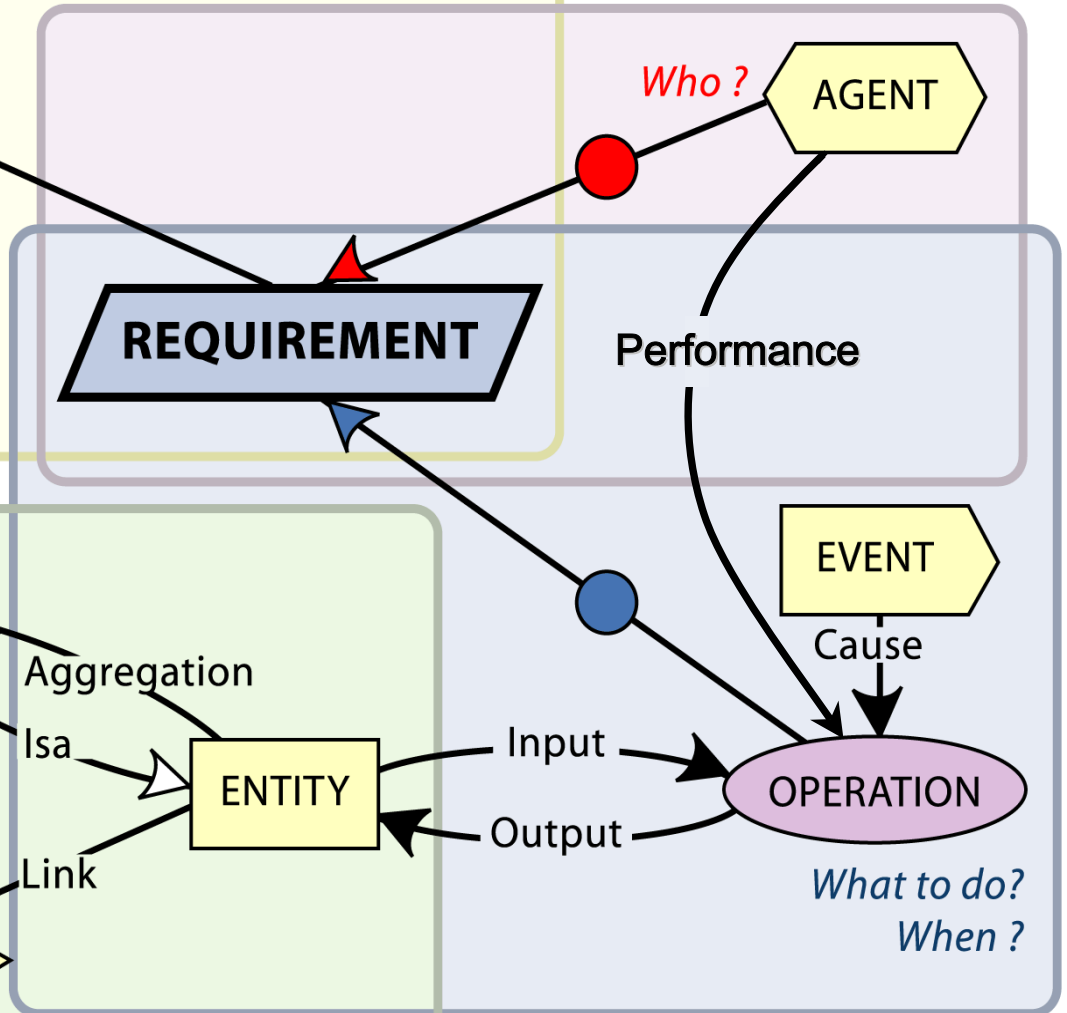


1. Why do we need Requirements Engineering ?
2. The Objectiver approach
- ▶ 3. Deeper inside the Objectiver approach
4. References
5. Demo

Goal modeling



Responsibility modeling



Object modeling

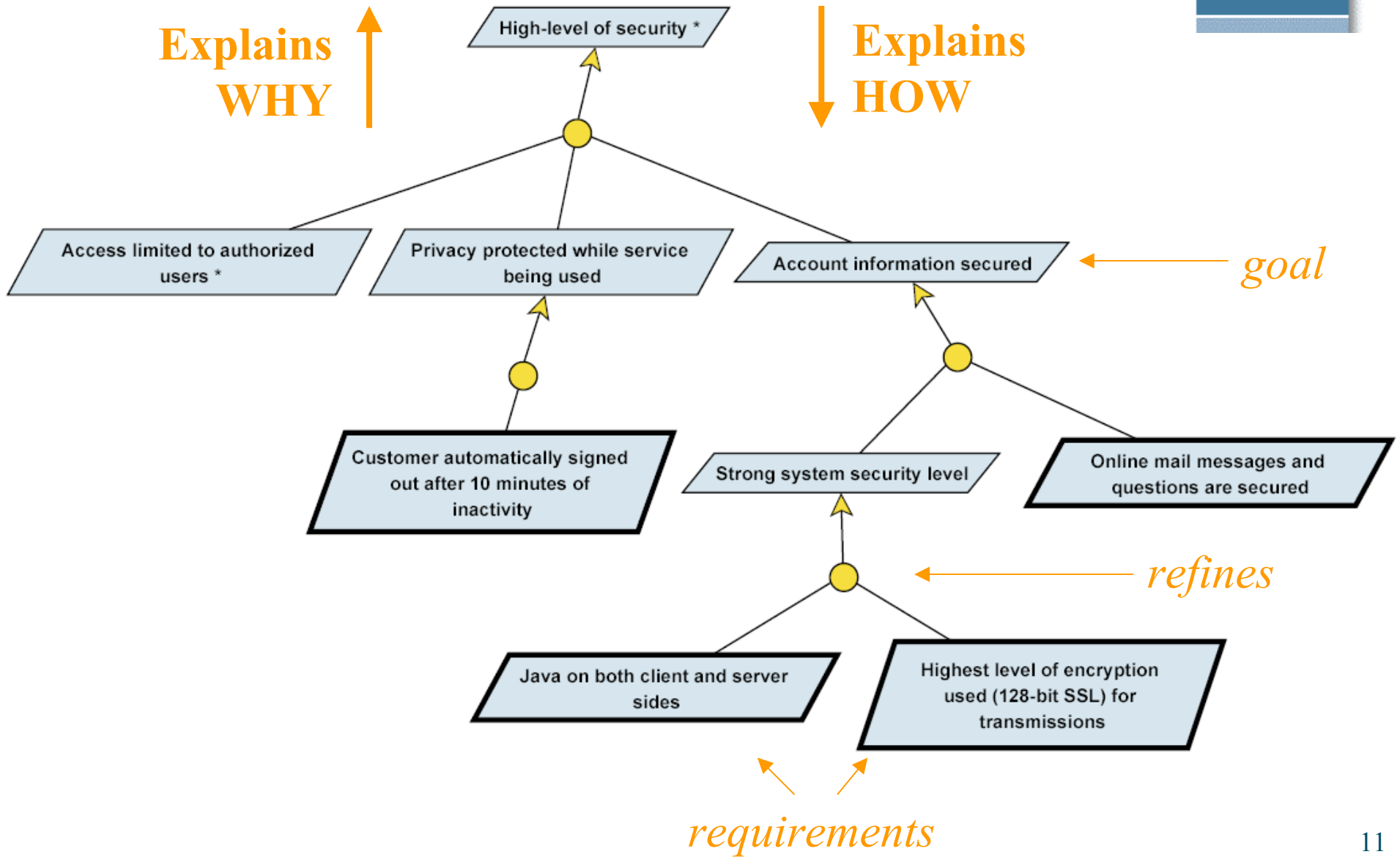
Operation modeling

1. Identify all the goals pursued by the project stakeholders and all involved people
2. Link them to higher-level strategic goals
3. Define how goals can be achieved

The use of a highly-graphic notation facilitates:

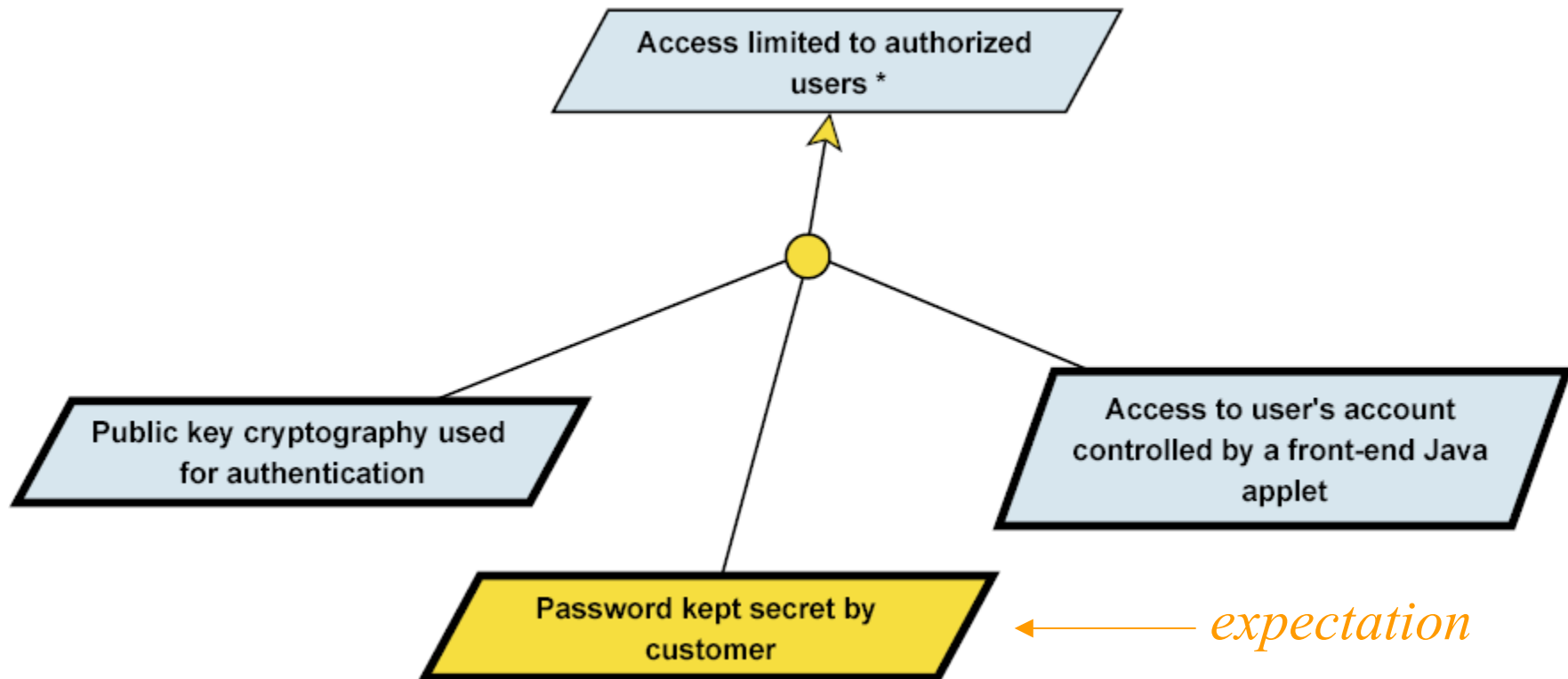
- Talking about project requirements and goals
- Quick identification of concept interrelationships
- Validation of the model

Goal Model

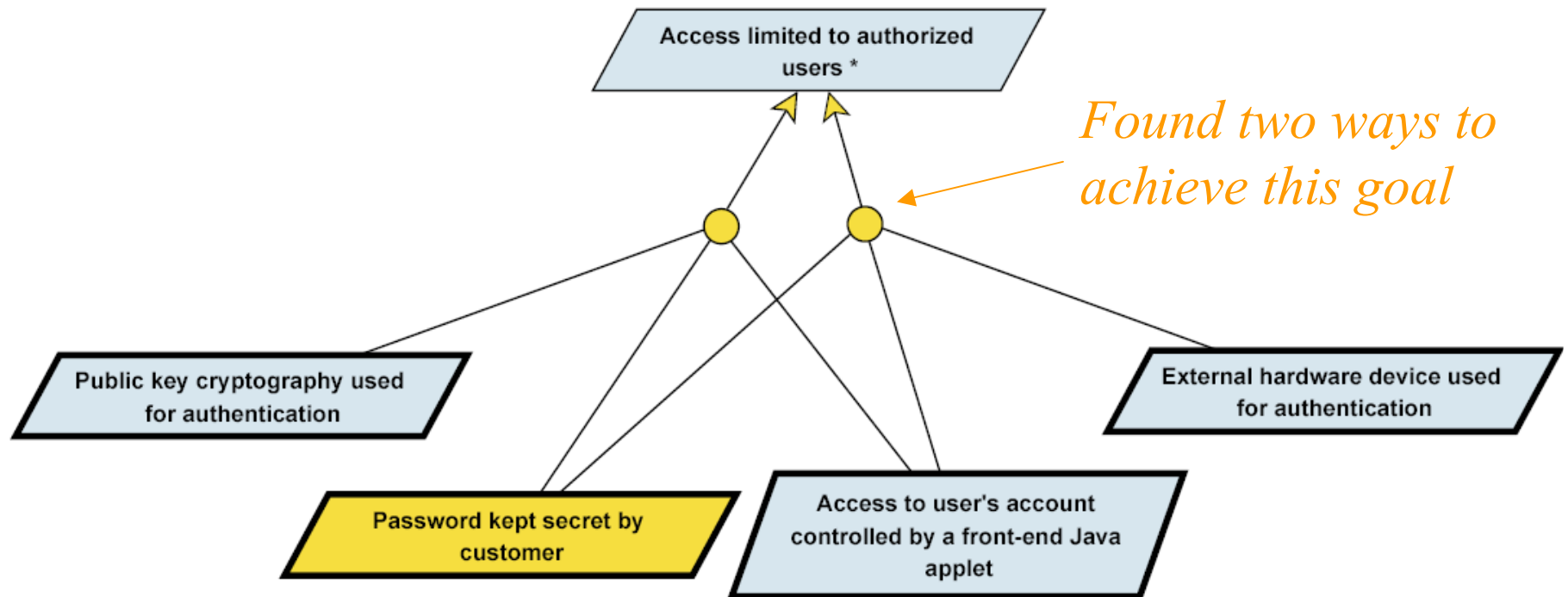


Expectations

Include in your model what is to be expected from the environment (system context)

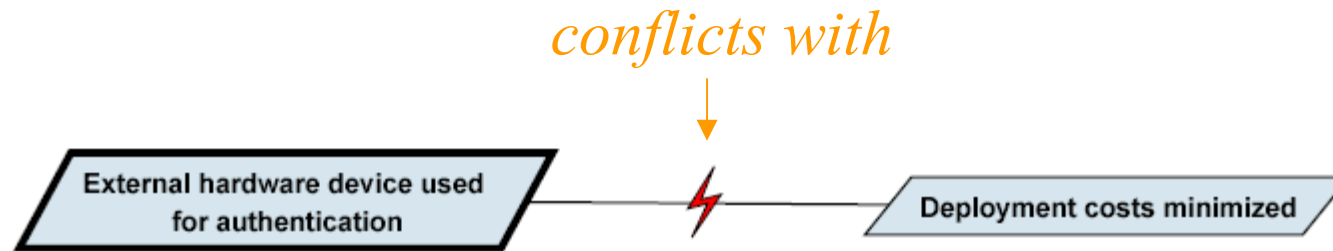


Alternatives



When alternatives are identified, further interviews and analysis shall be conducted to decide which solution shall be preferred over the other one.

Some goals may conflict with each other under certain circumstances.

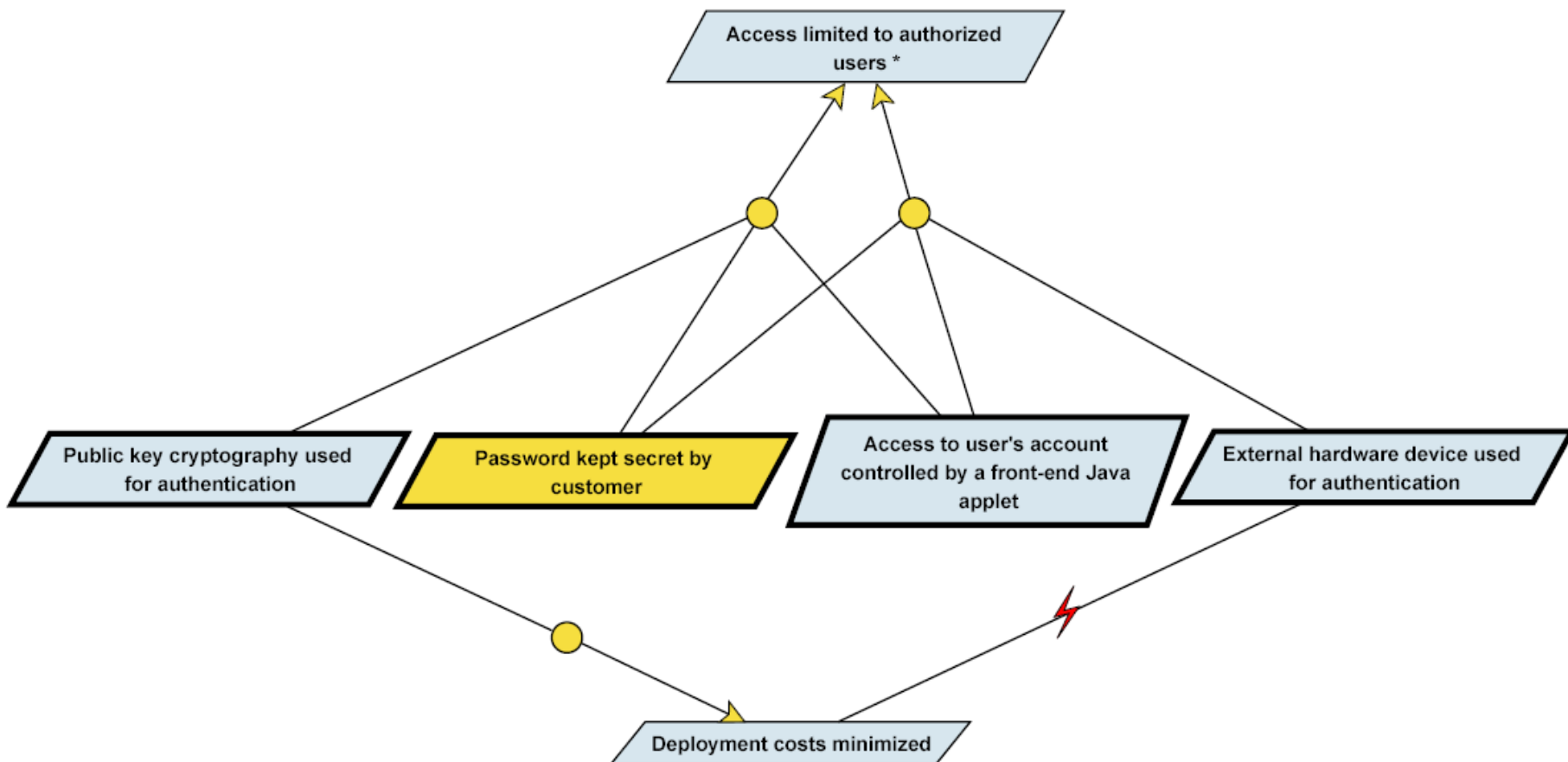


Solutions:

1. Introduce new goals/requirements that shall prevent the conditions leading to the conflict from occurring
2. Opt for an alternate solution (if we can't live with this conflict in the system)

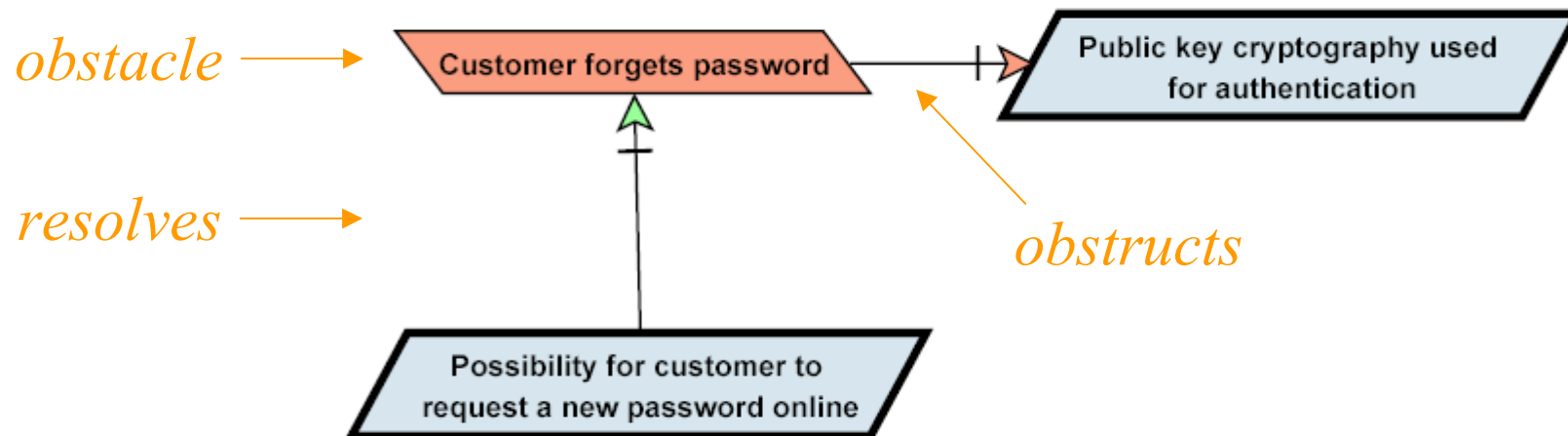
Back to our alternative

The conflict is used to guide our choices between existing alternatives



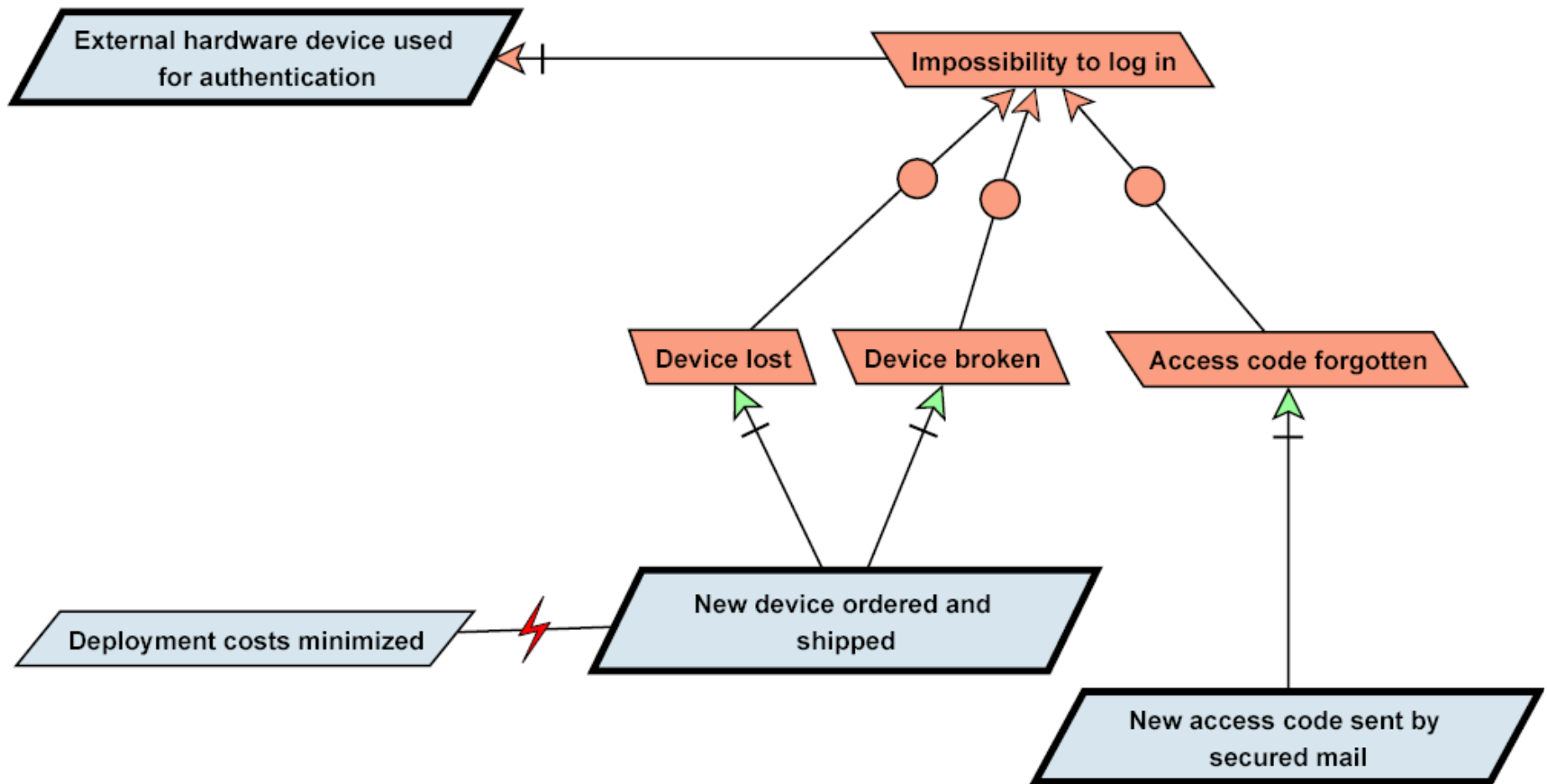
Obstacles

1. Identify the obstacles that could prevent reaching any goal
2. Identify new goals/requirements that shall resolve or alleviate these obstacles



Obstacle Analysis

1. Refine generic obstacles as more specific obstacles
2. Resolve the specific obstacles individually



1. Identify agents
2. Assign them responsibility for the expectations and requirements identified in the goal model

Agent : a human, device or system component

system agent : part of the system being modeled

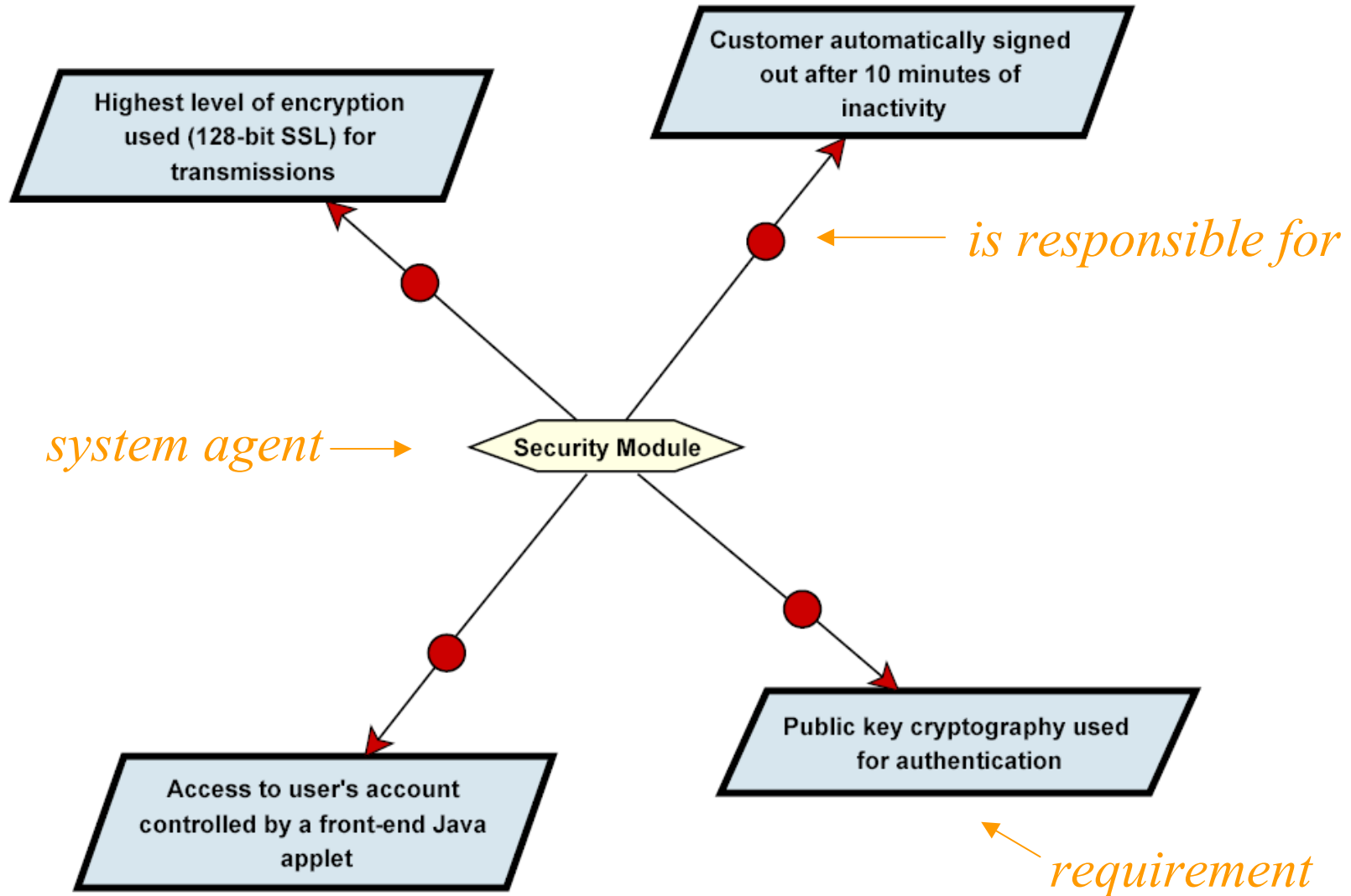
environmental agent : part of the system environment

Requirement : a low-level goal placed under the responsibility of a unique system agent

Expectation : assigned to an environmental agent. As such expectations cannot be enforced by the system.

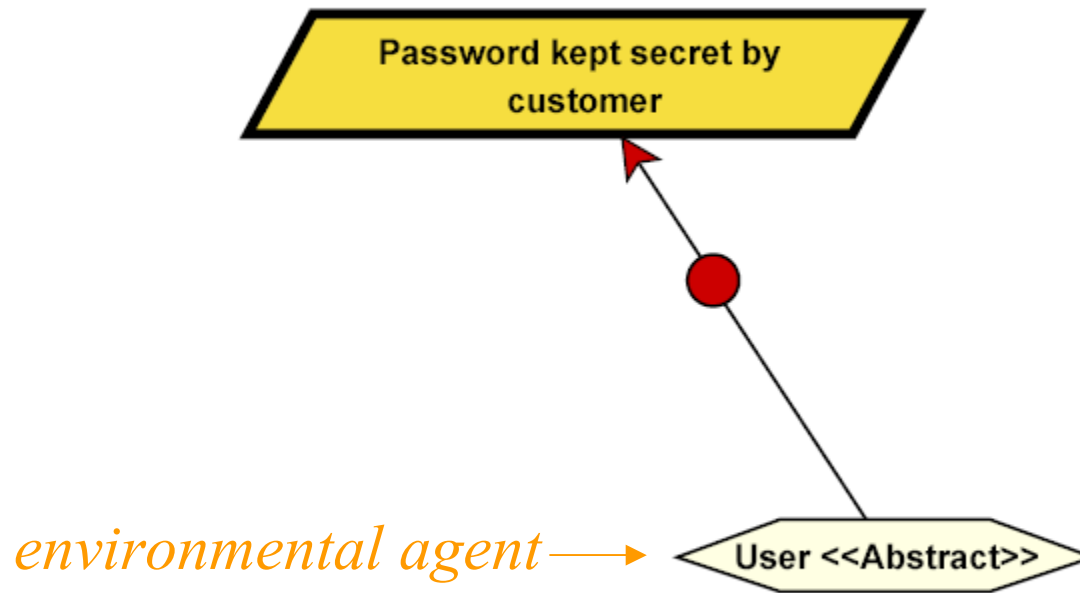
Responsibilities for *requirements*

Assign requirements to system agents

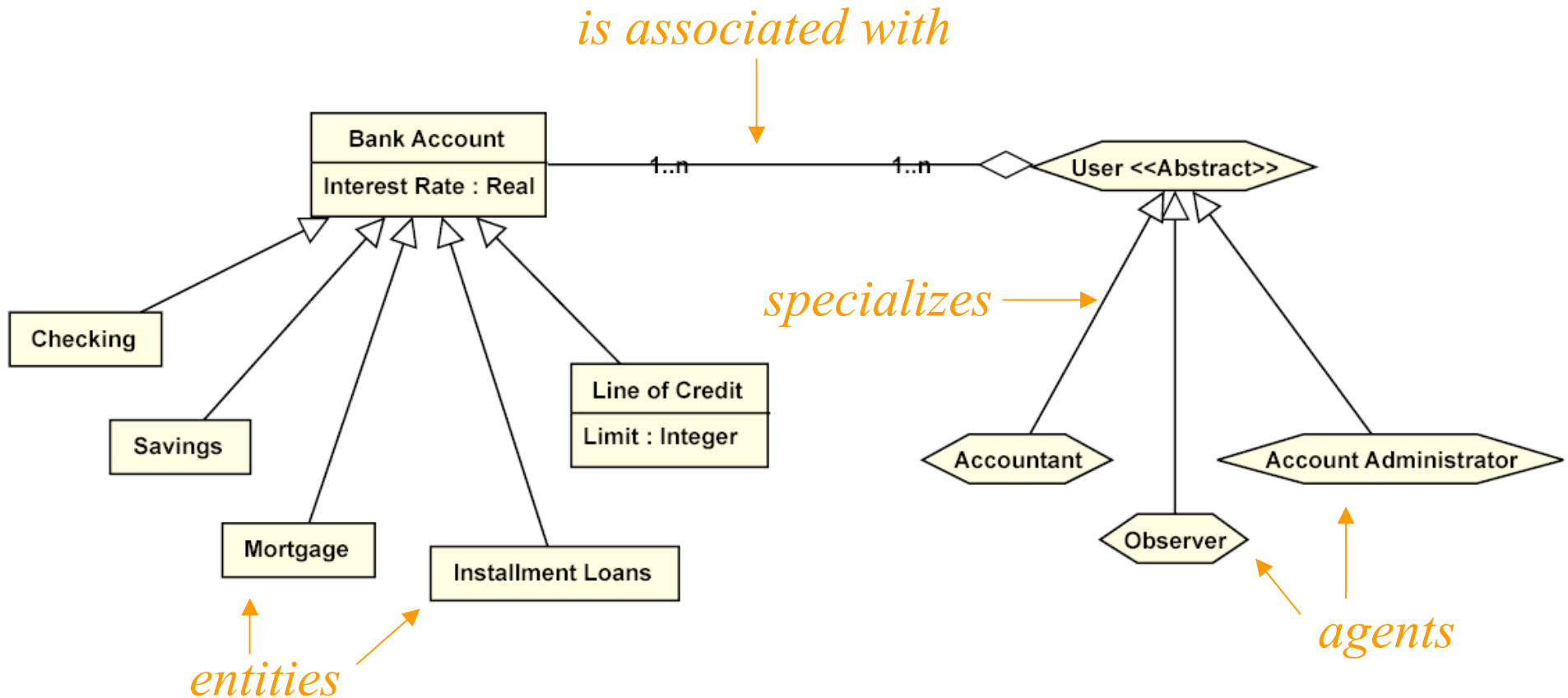


Responsibilities for *expectations*

Assign expectations to environmental agents



Object model



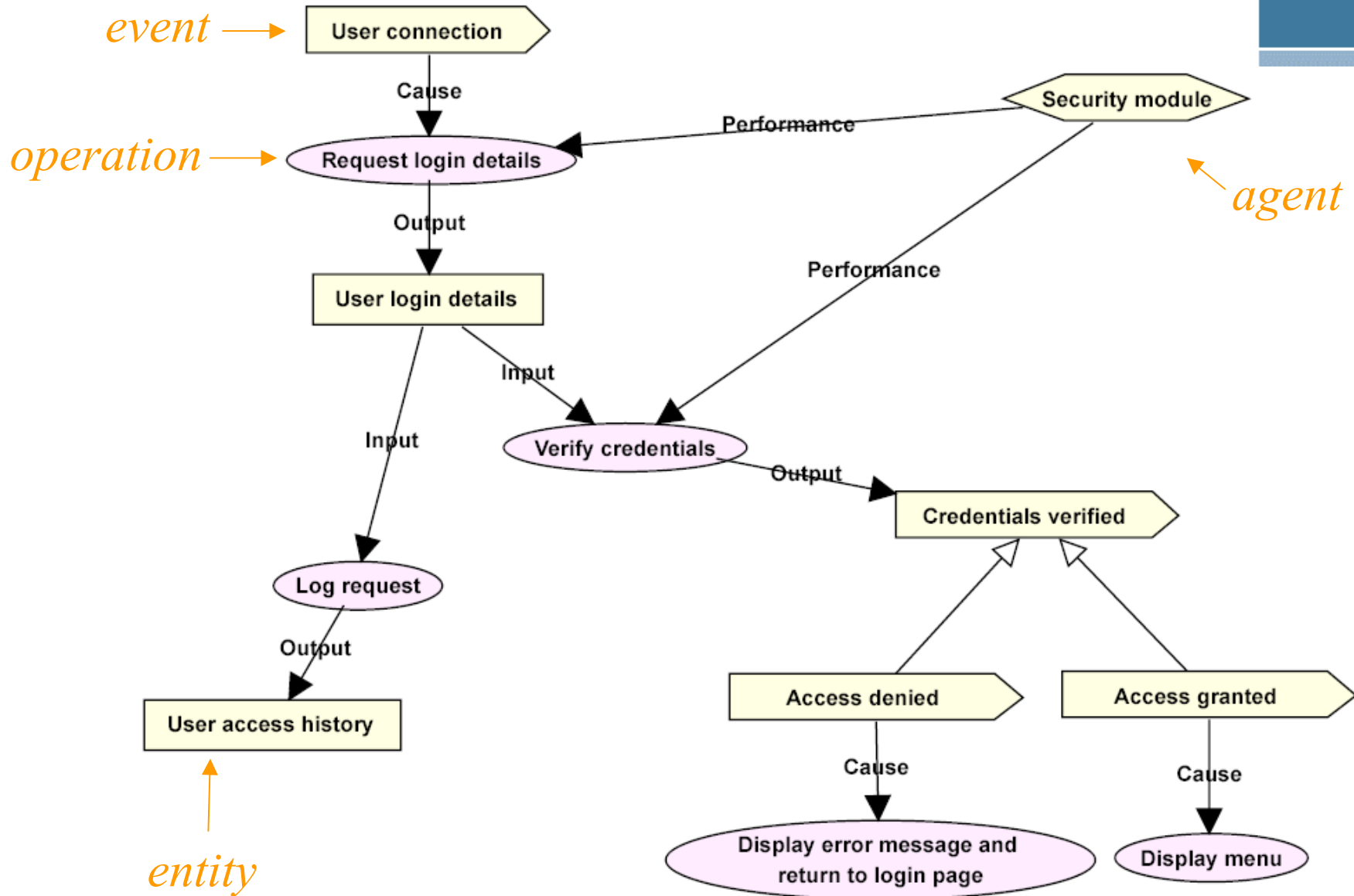
Model the domain objects, including entities, agents

- Model business or system processes
- Get traceability from operations (processes) to requirements

Operation = behavior that a system agent has to adopt to meet a requirement (that he's responsible of)

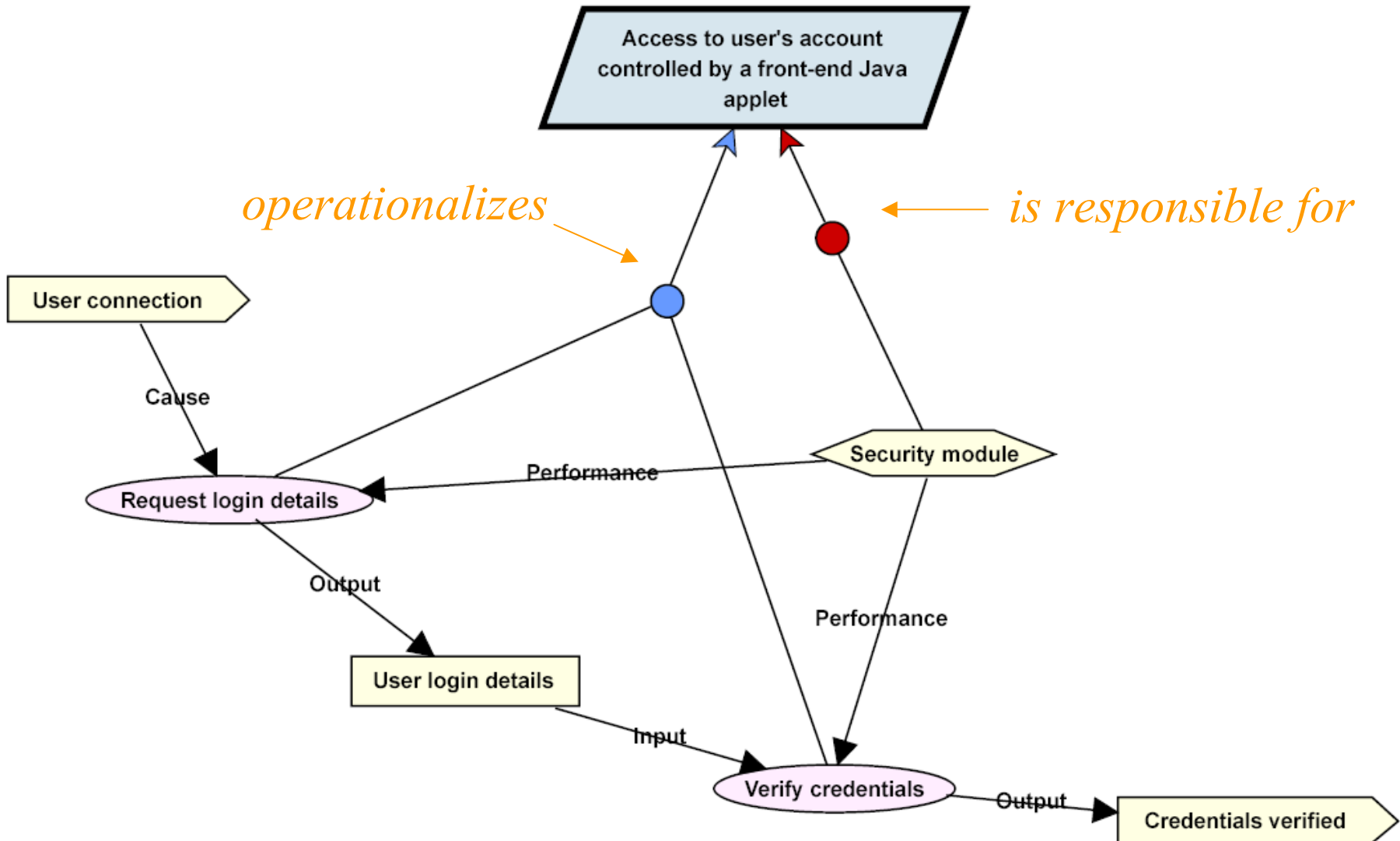
Events trigger operations

Operation Model



Traceability to requirements

Link operations to requirements for traceability



1. Why do we need Requirements Engineering ?
2. The Objectiver approach
3. Deeper inside the Objectiver approach
4. References
- ▶ 5. Demo

Objectiver, the Req Eng tool



The screenshot displays the Objectiver 1.5.1 interface. On the left is a Package View/Model View tree. The main area is split into two panes. The top pane, titled 'User access controlled Alt-F1', shows a diagram with nodes like 'User connection', 'Request login details', 'Security Module', 'Verify credentials', 'User Login Details', 'Access granted', and 'Access denied', connected by arrows labeled 'Cause', 'Performance', 'Output', and 'Input'. The bottom pane, titled 'User Access Controlled Alt-F3', shows a text editor with the text: 'Connection of a user to the system' followed by a paragraph: 'When the user connects to the system, he shall enter his User Login Details (username and password), the system shall verify his credentials (Verify credentials) before granting or denying access to the system.' Below the text are buttons for 'Create concept' and 'Reference'. A third pane at the bottom left shows a property editor with a table:

Name	Value
Name	Access to user's account c ...
Def	...

Diagram editor

explorers

Text editor

Property editor

Web Generated Documentation

The screenshot shows a Microsoft Internet Explorer browser window displaying a web page titled "User access controlled". The browser's address bar shows the file path "C:\tmp\online-banking\index.html". The page content is divided into several sections:

- Documents:** A tree view on the left showing a hierarchy of goals: "Goals" (expanded) containing "Attractiveness", "High security" (expanded) containing "High-level of security" and "User Access Controlled", and "Low cost".
- Concepts by type:** A list of concept types including "Agent", "Binary Association", "Cause", and "User connection 'Cause' Req".
- Concept specification:** A section for defining the concepts used in the model.
- User access controlled (Graphical View):** A central diagram showing the flow of a user login process. It starts with "User connection" (Cause) leading to "Request login details" (Performance). "Request login details" leads to "User Login Details" (Output). "User Login Details" leads to "Verify credentials" (Input). "Verify credentials" leads to "Access granted" and "Access denied" (Output). "Verify credentials" also leads to "Security Module" (Performance), which then leads to "Access to user's account controlled by a front-end Java applet" (Performance).
- User Access Controlled - description (Text View):** A section titled "Connection of a user to the system" with the following text: "Details (username and password), the system shall verify his credentials (Verify credentials) before granting or denying access to the system".

explorers

graphical view

property view

text view

More than a simple drawing tool



- Elicit and specify requirements in a **systematic** way
- Produce structured, motivated, easy to understand requirements documents
- Facilitate **communication**

And ...

Objectiver

- **Calls for tender** managed more easily
- Provide **traceability** from processes to business goals
- Highly **integrated views** on the model
- Powerful **querying** tool (for analysis, validation & verification)

A **power tool**
to **engineer** your **business**
and **technical requirements**

Objectiver

Get your free trial version now !

www.objectiver.com