# Modelling Method Conceptualisation within OMiLab: The Secure Tropos approach

University of Brighton

Secure and Dependable Software Systems

SenSe

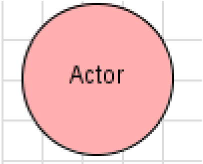
Research Cluster

# Secure Tropos

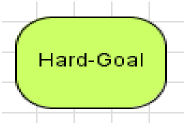
- A Secure Software Engineering Methodology
- Strongly Requirements Driven
  - *Adopts i*,* which offers actors, goals, and actor dependencies as primitive concepts for modelling during early requirements analysis
  - Enhanced with concepts from security engineering
- It describes both the organisational environment of the system and the system itself

# Secure Tropos Concepts

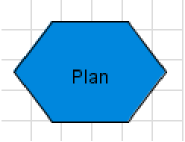
- **Actor**
  - an entity that has strategic goals and intentions.
  - An actor can be human, a system, or an organisation.



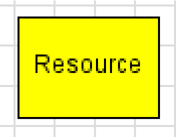
- **Malicious Actor**
  - A malicious actor's intention is to introduce threats to the system, which exploit vulnerabilities.
- **Goal**
  - Represents an actors' strategic interests.
  - Higher level strategic goals may be decomposed in simpler operational goals forming AND/OR goals hierarchy.
  - Our meta-model differentiates between organizational and security goals.



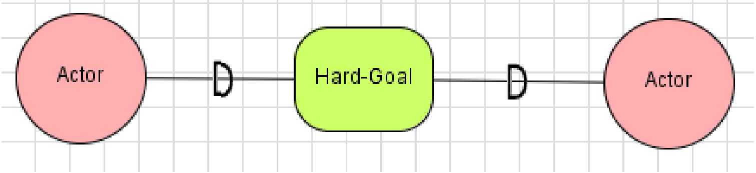
- **Plan**
  - A *plan* defines a specific way of operationalising a goal or a measure, i.e
  - the details and conditions under which a specific goal/measure is operationalised.



- **Resource**
  - A Physical or an informational entity



- **Dependency**
  - Indicates that an actor depends on another in order to achieve some goal/plan or to obtain a resource



# Security requirements in Secure Tropos

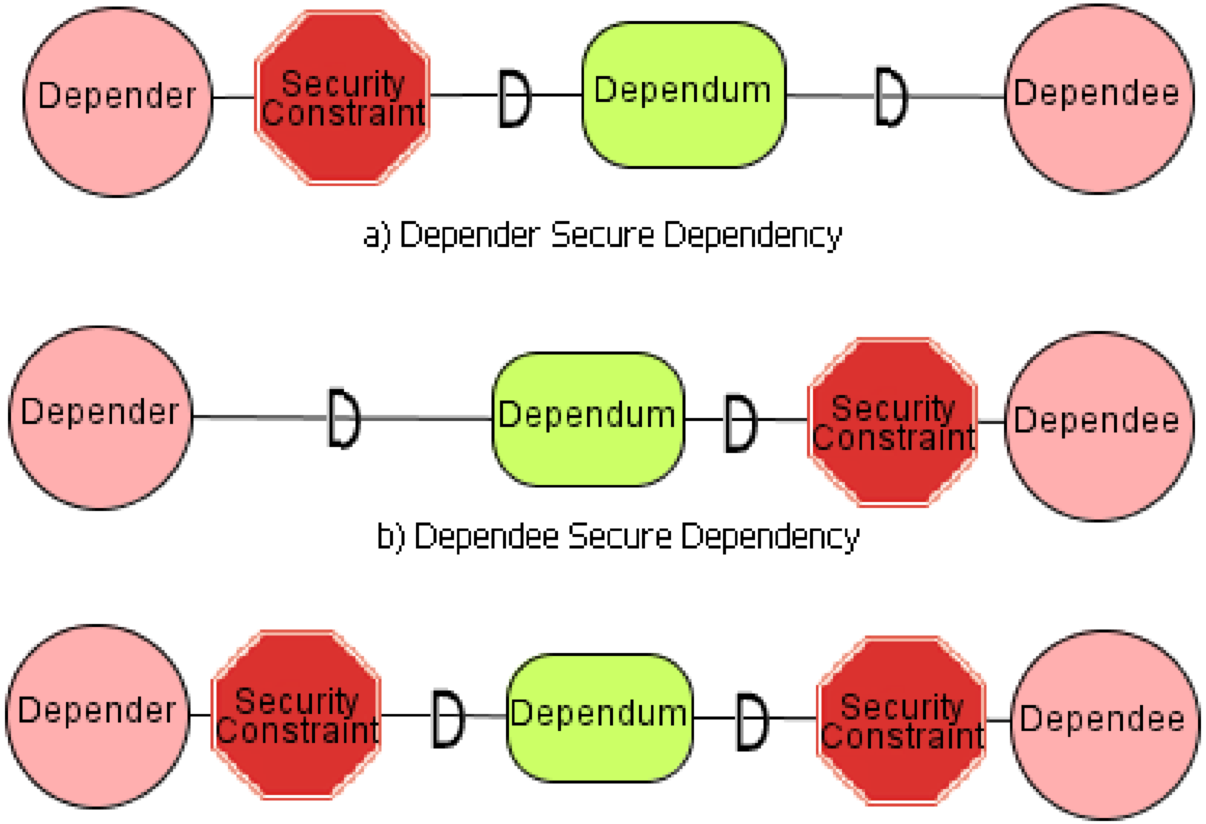
- How can we define and model security requirements?
  - As constraints!
  - Security requirements are most usefully defined as requirements for constraints on system functions
- In Secure Tropos security constraints define the system's security requirements
- A *Security Constraint* is used to represent a set of restrictions that do not permit specific actions to be taken or restrict the way that actions can be taken
- When a constraint is introduced, further analysis is required to establish if and how that constraint can be satisfied.

Security
Constraint

# Secure Dependency

- Secure dependency
  - Introduces security constraint (s) that must be fulfilled for the dependency to be satisfied
- Three different types depending on which actor needs to satisfy the security constraint(s)
  - The depender must satisfy the security constraint(s)
  - The dependee must satisfy the security constraint(s)
  - Both must satisfy the security constraints

# Types of Secure Dependencies



a) Depender Secure Dependency

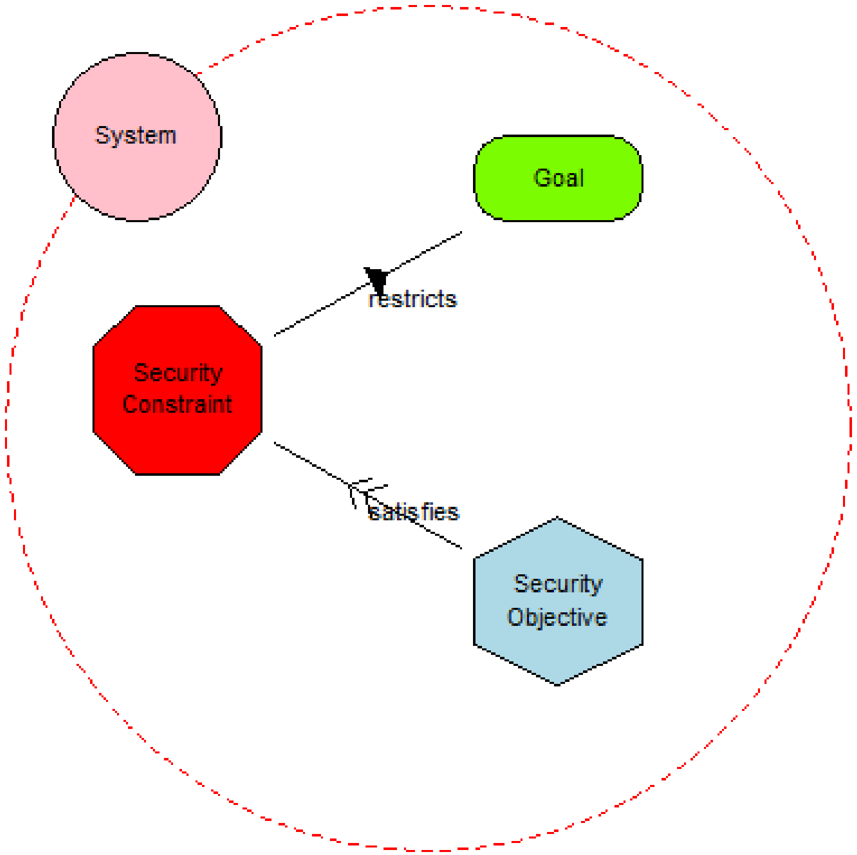
b) Dependee Secure Dependency

c) Double Secure Dependency

# Security Objective

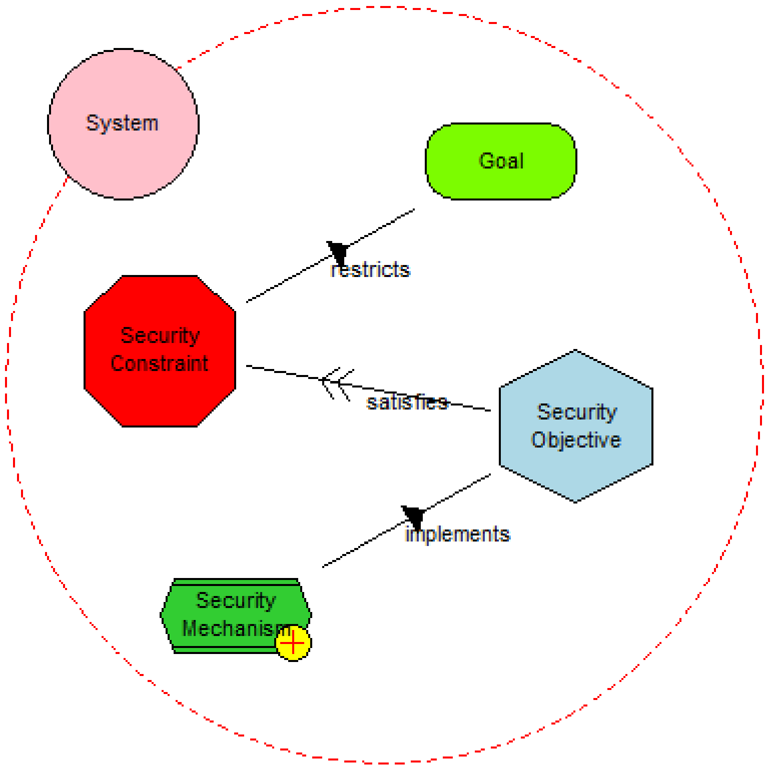
- Represents strategic interests of an actor with respect to security
- Security objectives are mainly introduced in order to contribute towards the achievement of an actor's or system's security constraints.
- The satisfaction of one or more security constraints by a security objective is defined through a Satisfies relationship.
- A security objective does not particularly define how the security constraints can be achieved, since alternatives can be considered.

# Security Objective example



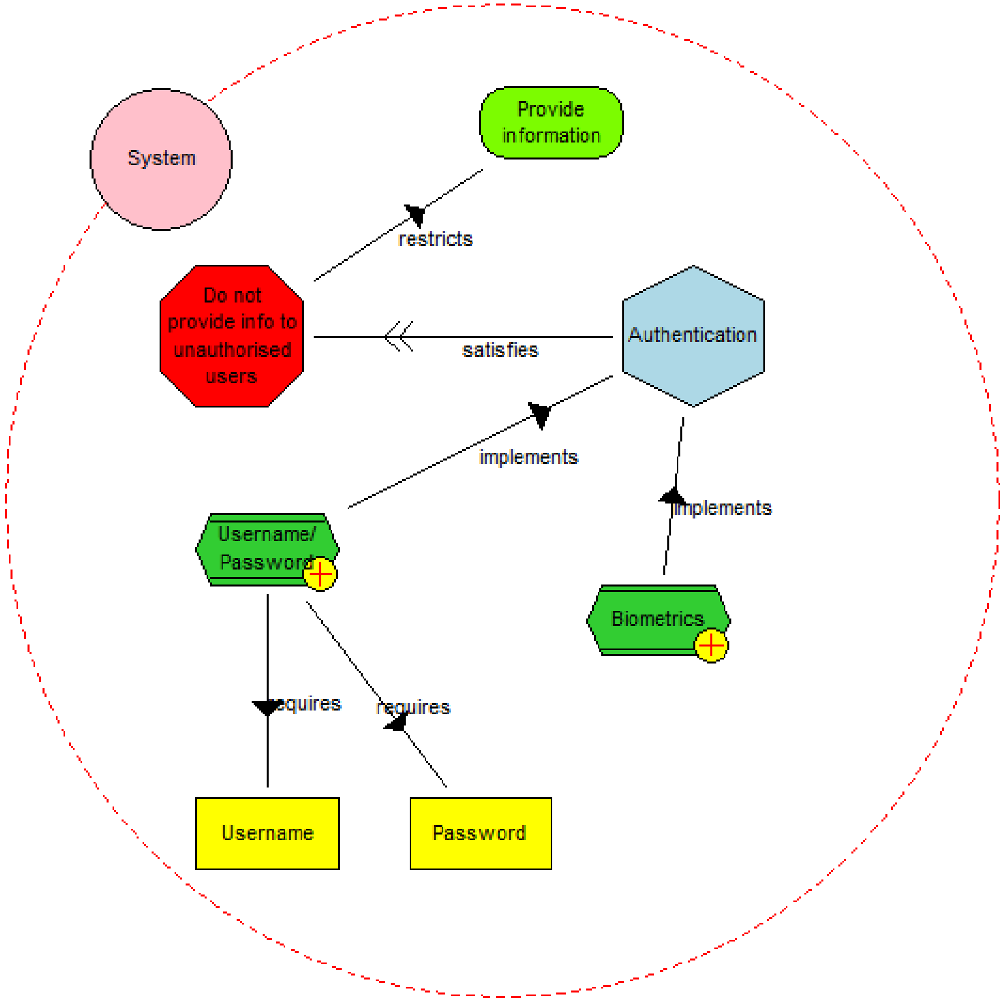
# Security Mechanism

- A security mechanism represents a particular way for implementing a security objective
- In the context of Secure Tropos, this means a specific and defined action that an actor executes to operationalise a security objective.



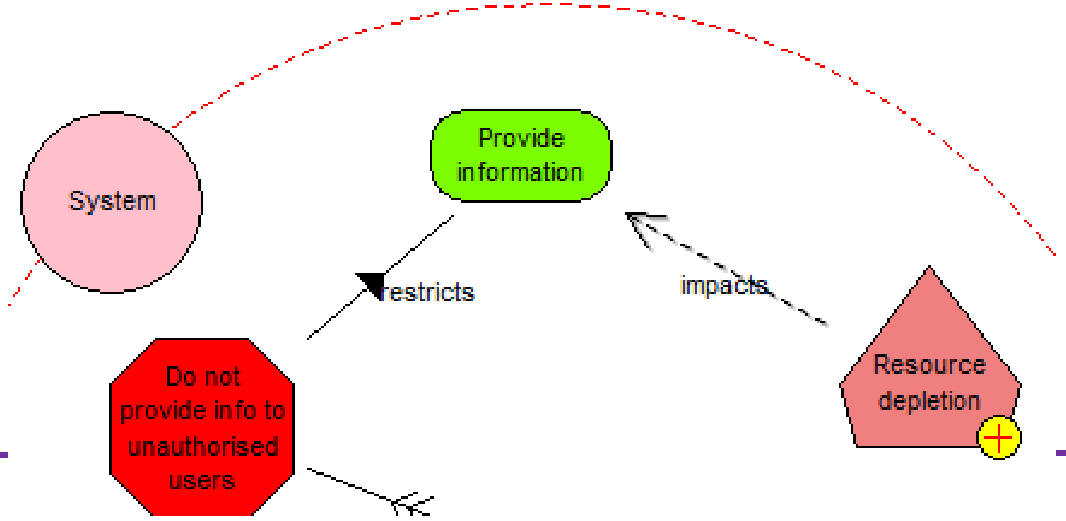
# Secure Resource

- An informational or physical entity that is needed for the achievement of a security objective or the fulfilment of a security mechanism.



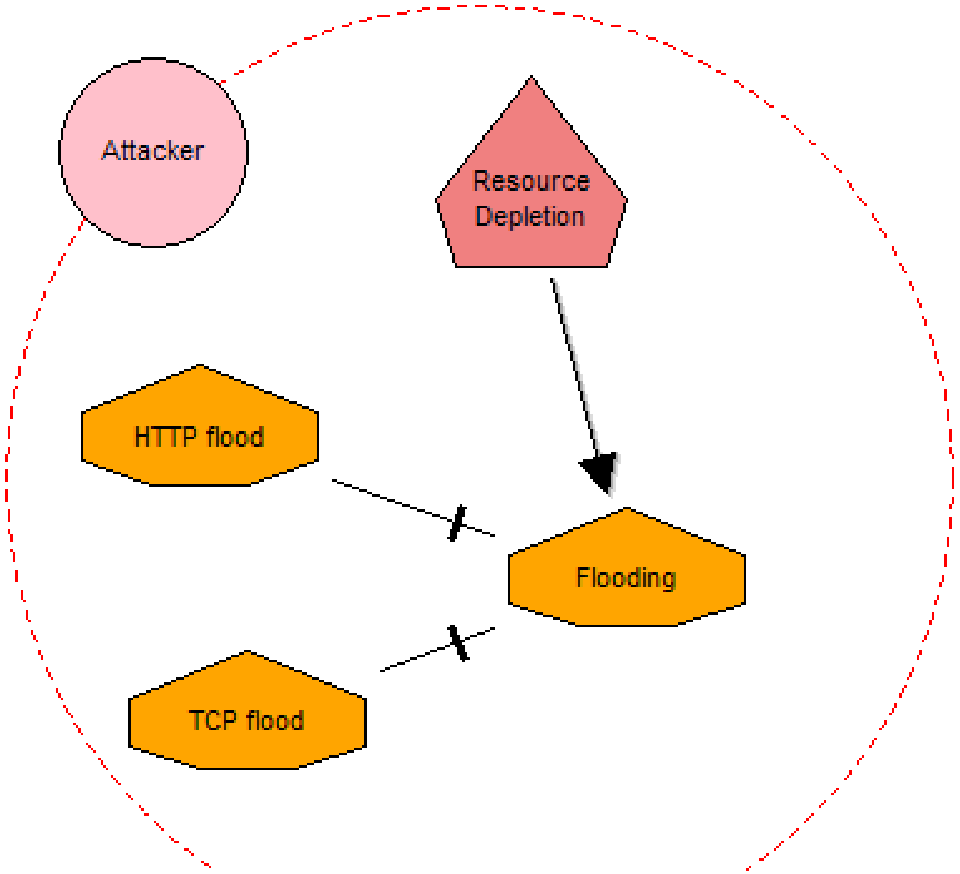
# Threat

- Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service



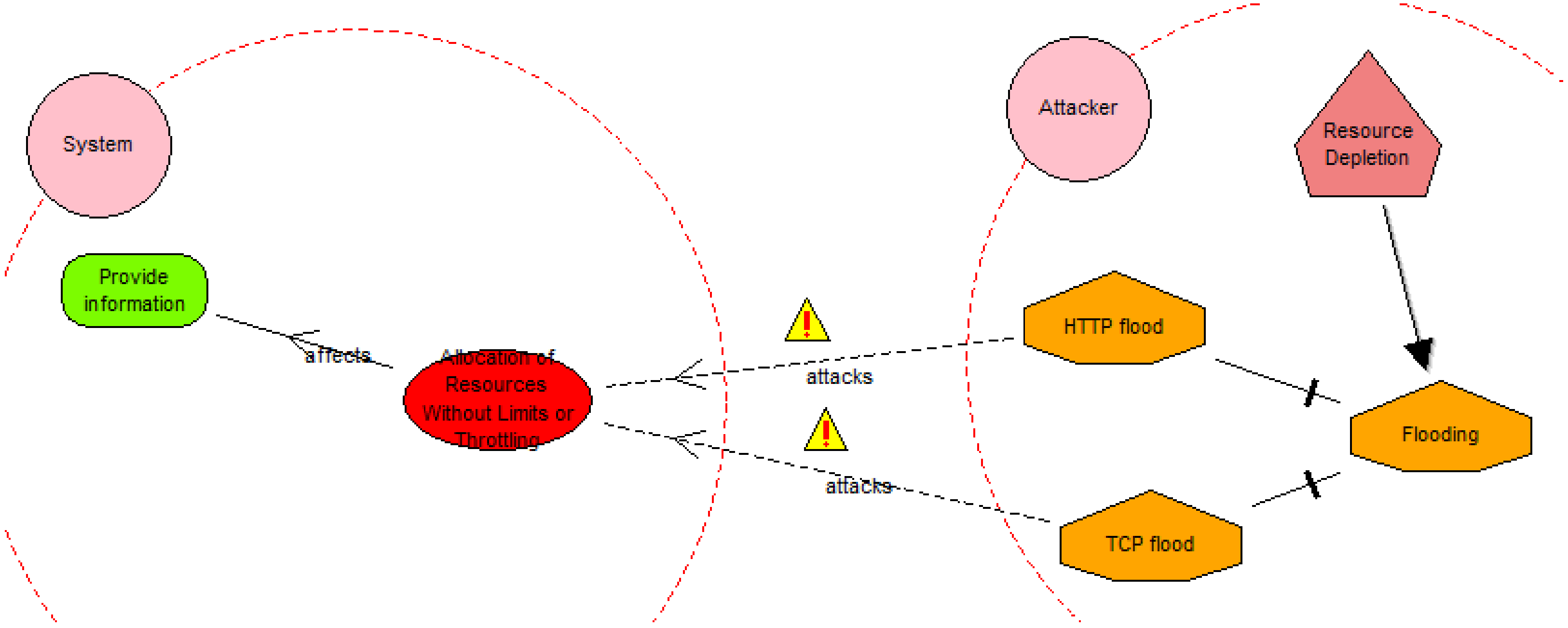
# Attack Method

- An activity that attempts to gain unauthorized access to system services, resources, or information, or an attempt to compromise system integrity

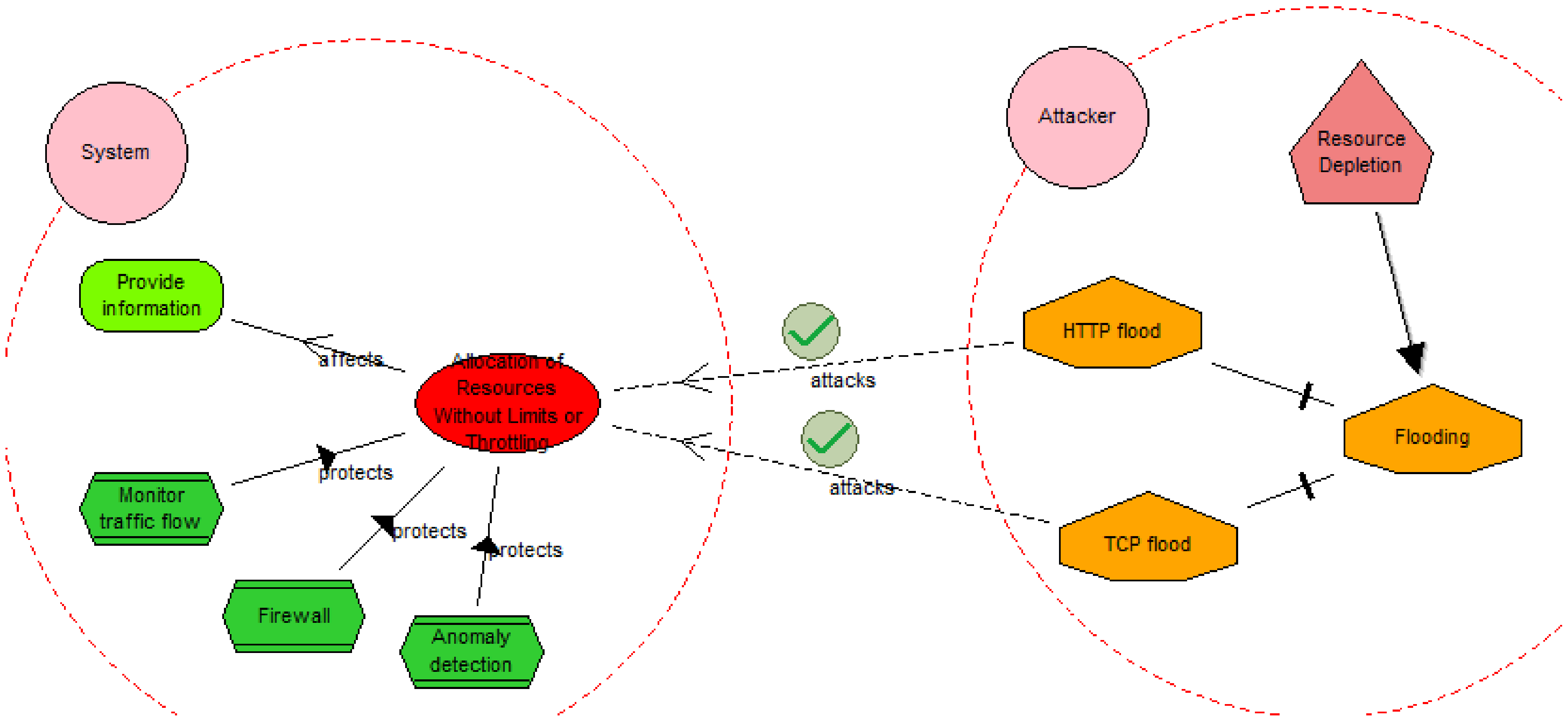


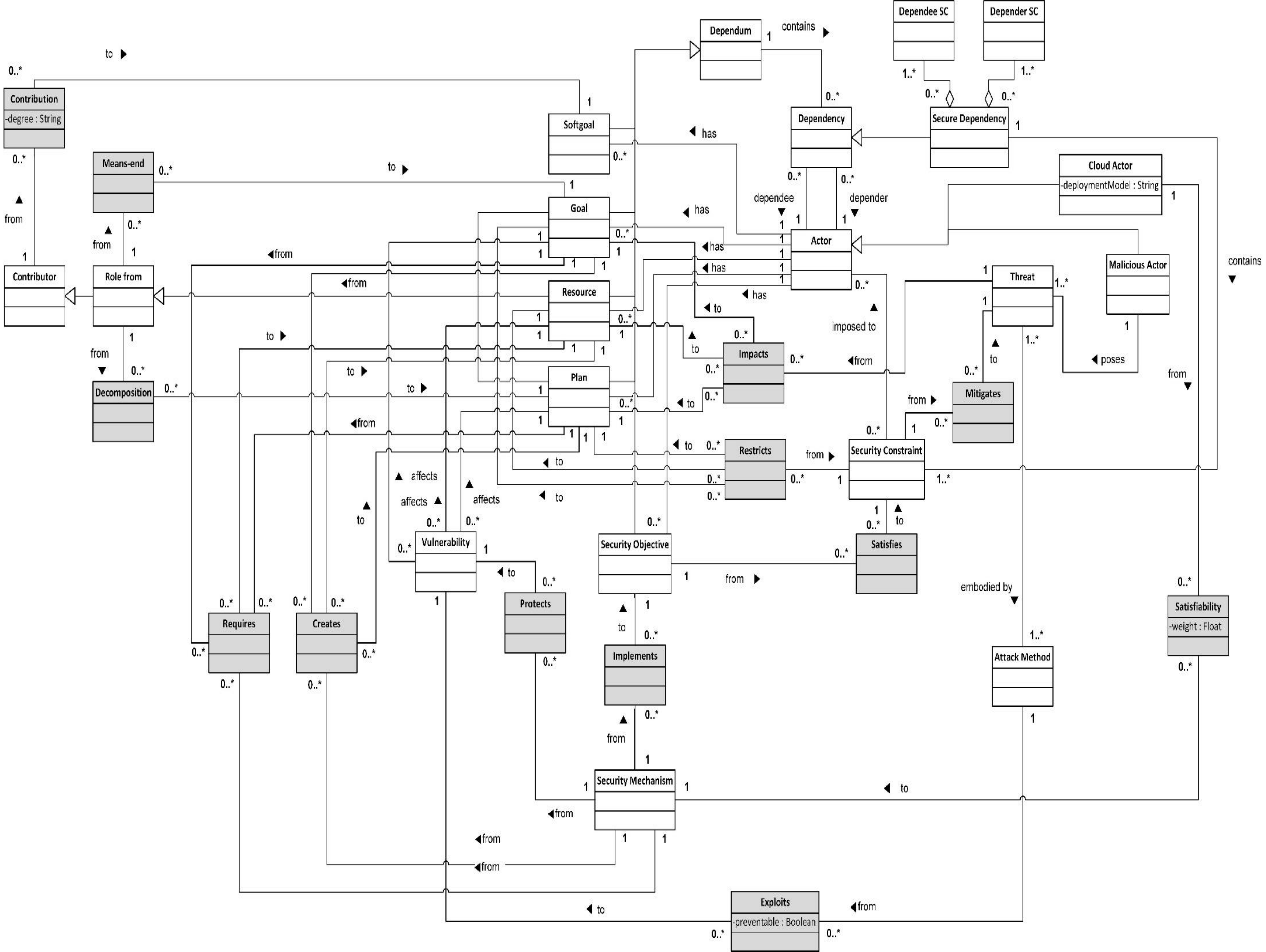
# Vulnerability

- Weakness in an information system that could be exploited or triggered by a an attack method



# Security Mechanism





Secure and Dependable

SenSe

Research Cluster

Software Systems

Universi

# University Virtual Learning Environment

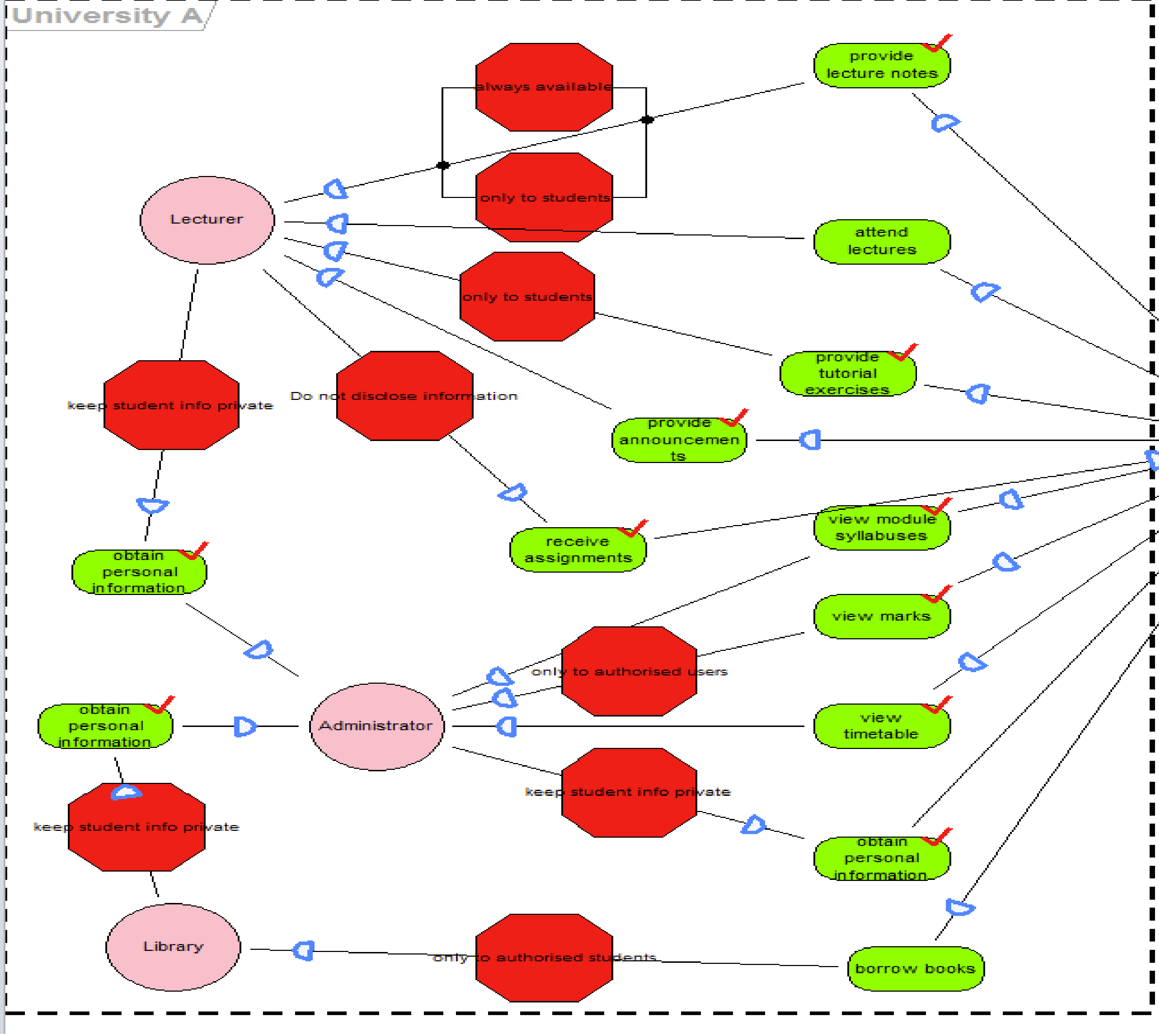
- A Virtual Learning Environment (VLE) is a system for delivering learning materials to students via the web. These systems include assessment, student tracking, collaboration, and communication tools. They can support students' learning outside the lecture hall 24 hours a day, seven days a week
- Access any academic material related to their course.
- Access and modify the personal information of their student record.
- Access their academic student record (grades, attendance, registered courses, etc.).

# Process

- Iterative process
  - It is based on the development of a set of models that are incrementally refined to include further details;
  - It provides a structured way of eliciting and analysing security requirements;
  - It supports the identification of security mechanisms that fulfill relevant security requirements;
  - It supports identification and analysis of threats and vulnerabilities;
  - It provides a framework to model and analyse security attacks (in terms of scenarios).

# Organisational View

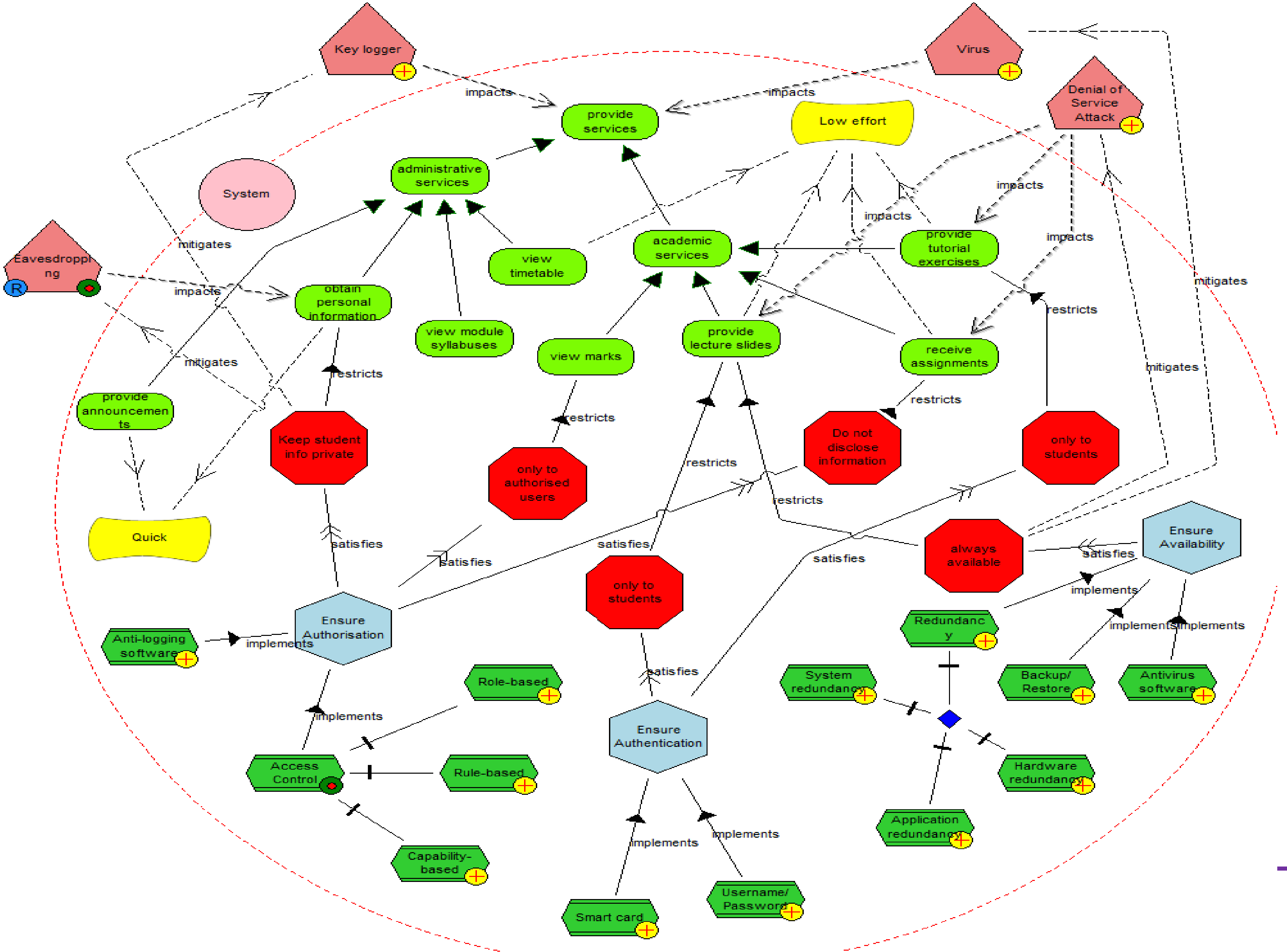
- This view represents the organisational architecture allowing developer to understand the requirements of the organisation and any interactions between the organisation and external actors (or systems).
- It should display an organisational boundary, where organisational actors reside; any external actors are modelled outside of this boundary.



Secure and Dependable
SenSe
Research Cluster
Software Systems

# Security Requirements View

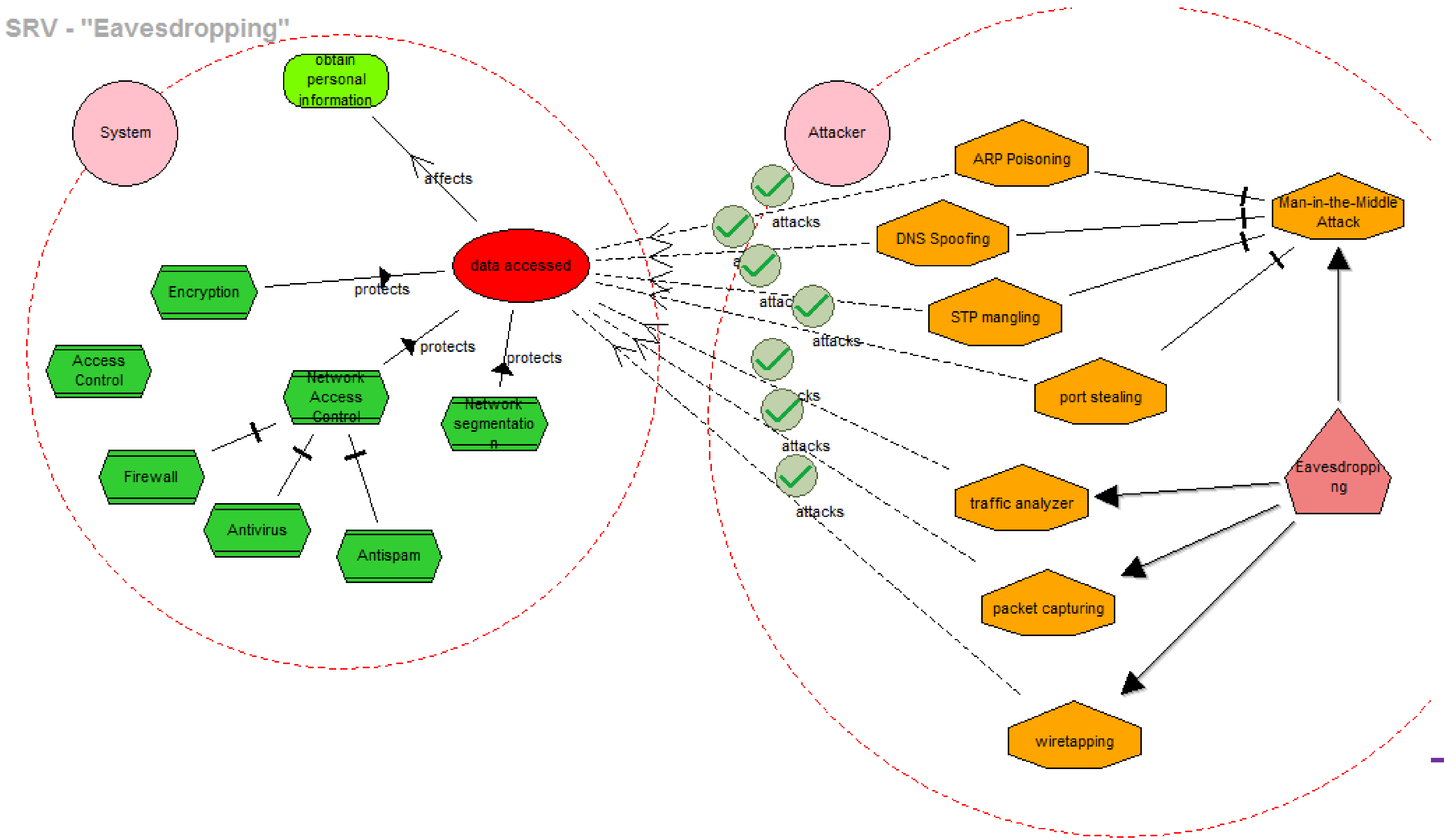
- *Deeper* representation of the organisational view.
- System actors and their goals are designed including the security analysis concepts.
- So essentially the modelling activity focuses on the responsibilities of the system and other actors as well as the interaction of actors with the system itself.


Key logger
Virus
impacts
impacts
provide services
Low effort
Denial of Service Attack
System
administrative services
impacts
impacts
impacts
Eavesdropping
R
mitigates
academic services
provide tutorial exercises
view timetable
mitigates
impacts
obtain personal information
restricts
view module syllabuses
view marks
provide lecture slides
receive assignments
mitigates
mitigates
provide announcements
restricts
restricts
Keep student info private
restricts
only to authorised users
Do not disclose information
only to students
restricts
restricts
Quick
satisfies
satisfies
satisfies
always available
Ensure Availability
satisfies
only to students
satisfies
implements
Ensure Authorisation
Anti-logging software
implements
Redundancy
implements
implements
Role-based
satisfies
System redundancy
Backup/ Restore
Antivirus software
implements
Ensure Authentication
Access Control
Rule-based
Hardware redundancy
Application redundancy
implements
implements
Capability-based
Smart card
Username/ Password

# Security Attacks View

- Allows the evaluation of the system security against various attacks.
- The security attacks modelling takes place by checking if threats introduced in the Security Requirements View are mitigated by the security mechanisms available within the system.
- If any of the attacks are likely to succeed the developer has an opportunity to go back to the previous views and adjust the design accordingly.

# Security Attack View



# SecTro v2

- The framework is supported by a tool that has been developed based on the Open Models Initiative ADOxx Platform (www.openmodels.at).
- The tool provides an environment for developers to create a number of diagrams that support the described process.
- The tool automatically generates documentation (PDF, WORD) with the security requirements specification/analysis.