Socio-Technical Security Modelling Language

Elda Paja

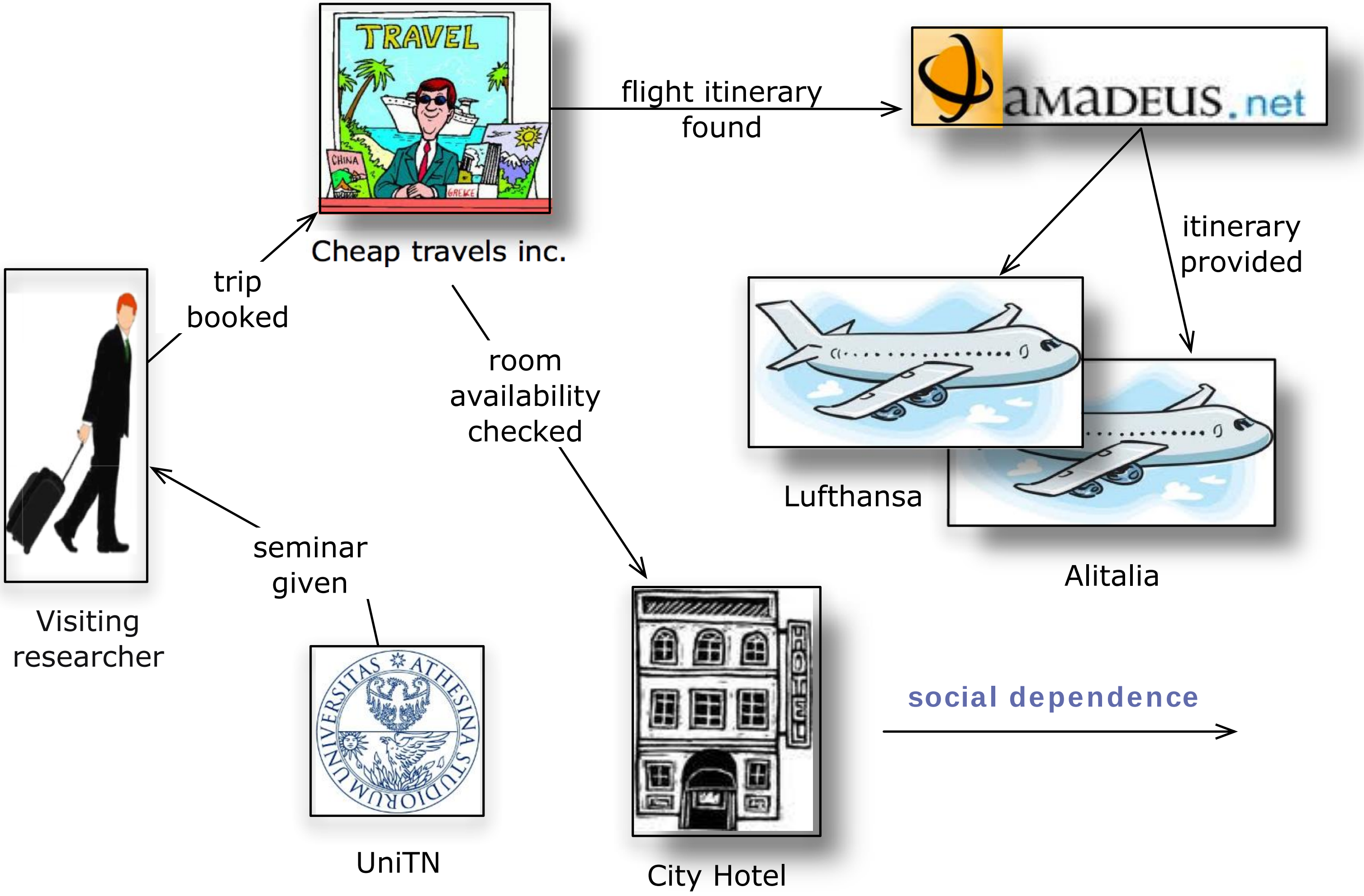
November 2014

Socio-Technical Systems (STS)

- An interplay of different subsystems
  - Not only technical, but also humans and organisations
  - Each subsystem is autonomous
  - Defined in terms of interaction among subsystems
    - Each subsystem needs to socially rely on others to fulfill its objectives

- Examples include
  smart homes, e-commerce sites, eHealth systems, etc.

An example of STS



Cheap travels inc.

flight itinerary found

trip booked

itinerary provided

Lufthansa

Alitalia

room availability checked

Visiting researcher

seminar given

UniTN

City Hotel

social dependence

The Security Problem in STS

- Interaction is everywhere!
  - Technical Systems – Technical Systems
  - Technical Systems – Social Actors
  - Social Actors – Social Actors
- Social aspects are a main concern
  - Decentralized setting: no controlling authority
  - Autonomy: security cannot be enforced
- Key idea: social contracts to constraint interaction
  - Social dependence
  - Information exchange

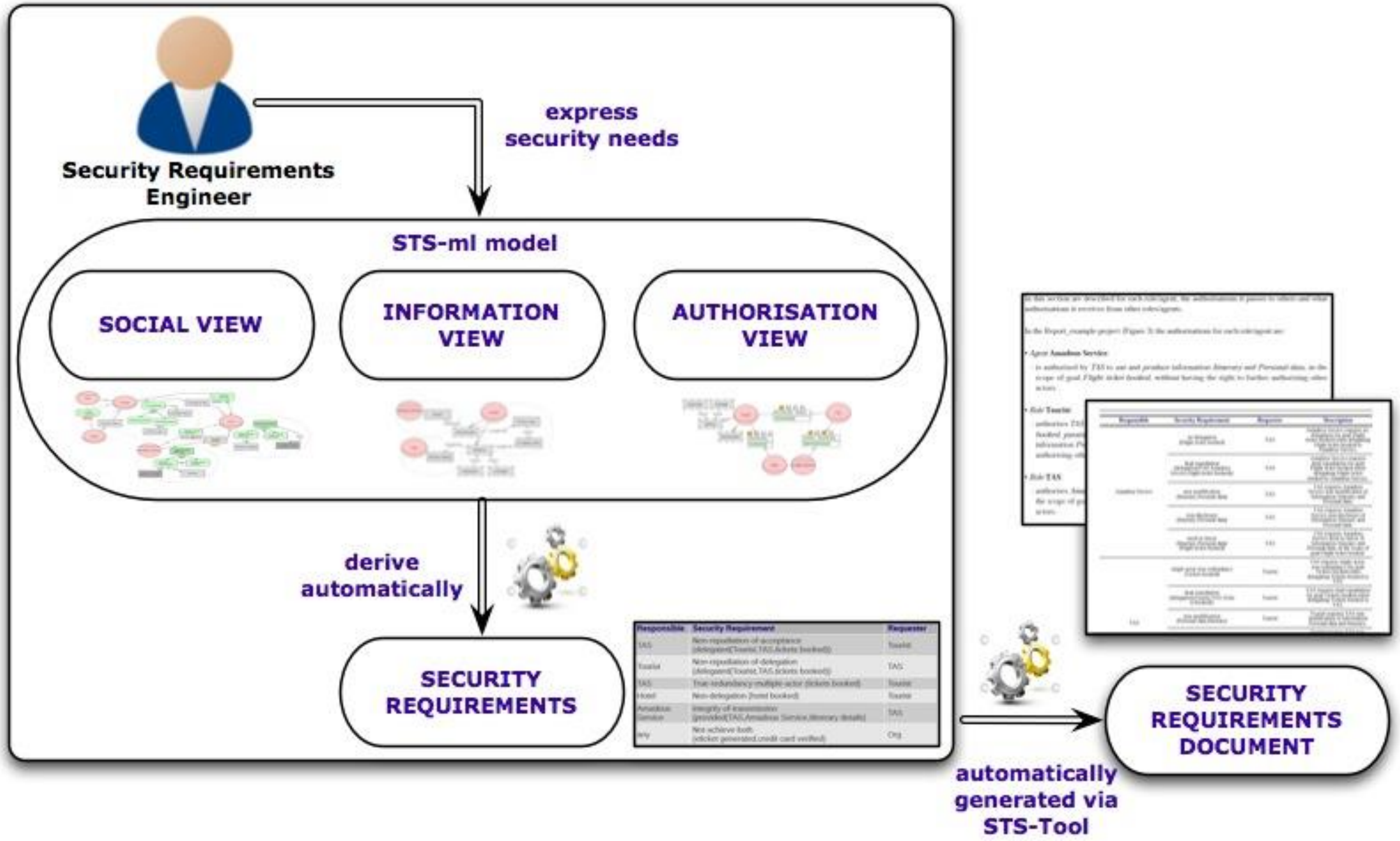
Socio-Technical Security Modeling Language (STS-ml)

- Actor – and goal – oriented requirements modeling language

- Models are built diagrammatically
  - Graphical concepts and relations are used to create the models
  - Multiple views, each focusing on a specific perspective

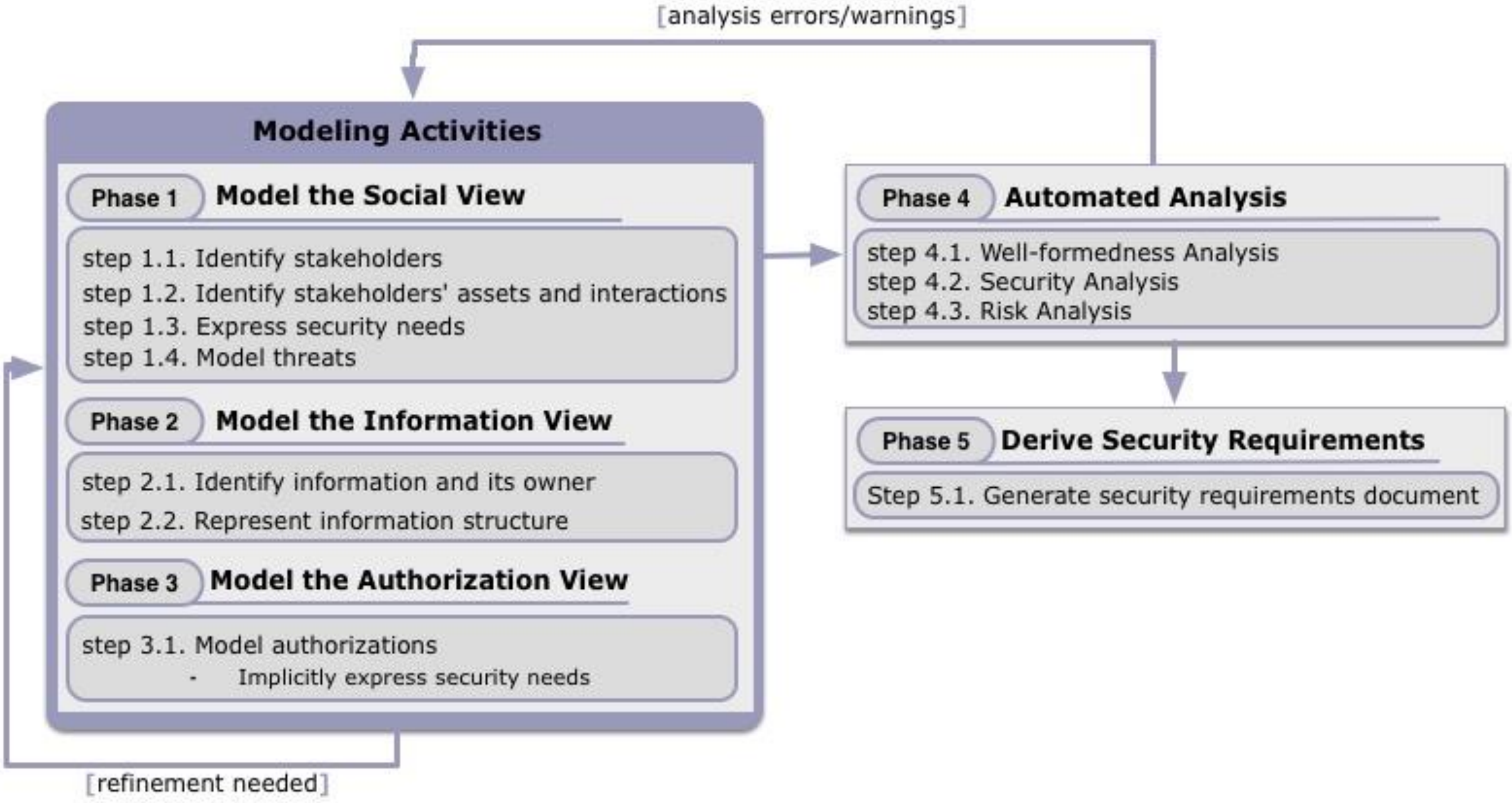
- Allow stakeholders to express constraints (security needs) over interactions
  - Social dependence (goal delegation)
    - E.g.: visiting researcher depends on the cheap travel inc. to book the hotel and flight tickets and he requires it not to deny having accepted the delegation
  - Documents exchange
    - E.g.: visiting researcher wants the cheap travel inc. to use his personal data information strictly to book the hotel and flight tickets, but not for any other purposes

STS-ml: outline



Responsible	Security Requirement	Requester
TAS	Non-repudiation of acceptance (delegated(Tourist,TAS,tickets booked))	Tourist
Tourist	Non-repudiation of delegation (delegated(Tourist,TAS,tickets booked))	TAS
TAS	True redundancy multiple actor (tickets booked)	Tourist
Hotel	Non-delegation (hotel booked)	Tourist
Amadeus Service	Integrity of transmission (provided(TAS,Amadeus Service,itinerary details)	TAS
any	Not achieve both (eticket generated,credit card verified)	Org

The STS method



[analysis errors/warnings]

Modeling Activities

Phase 1 Model the Social View

step 1.1. Identify stakeholders
step 1.2. Identify stakeholders' assets and interactions
step 1.3. Express security needs
step 1.4. Model threats

Phase 2 Model the Information View

step 2.1. Identify information and its owner
step 2.2. Represent information structure

Phase 3 Model the Authorization View

step 3.1. Model authorizations
- Implicitly express security needs

Phase 4 Automated Analysis

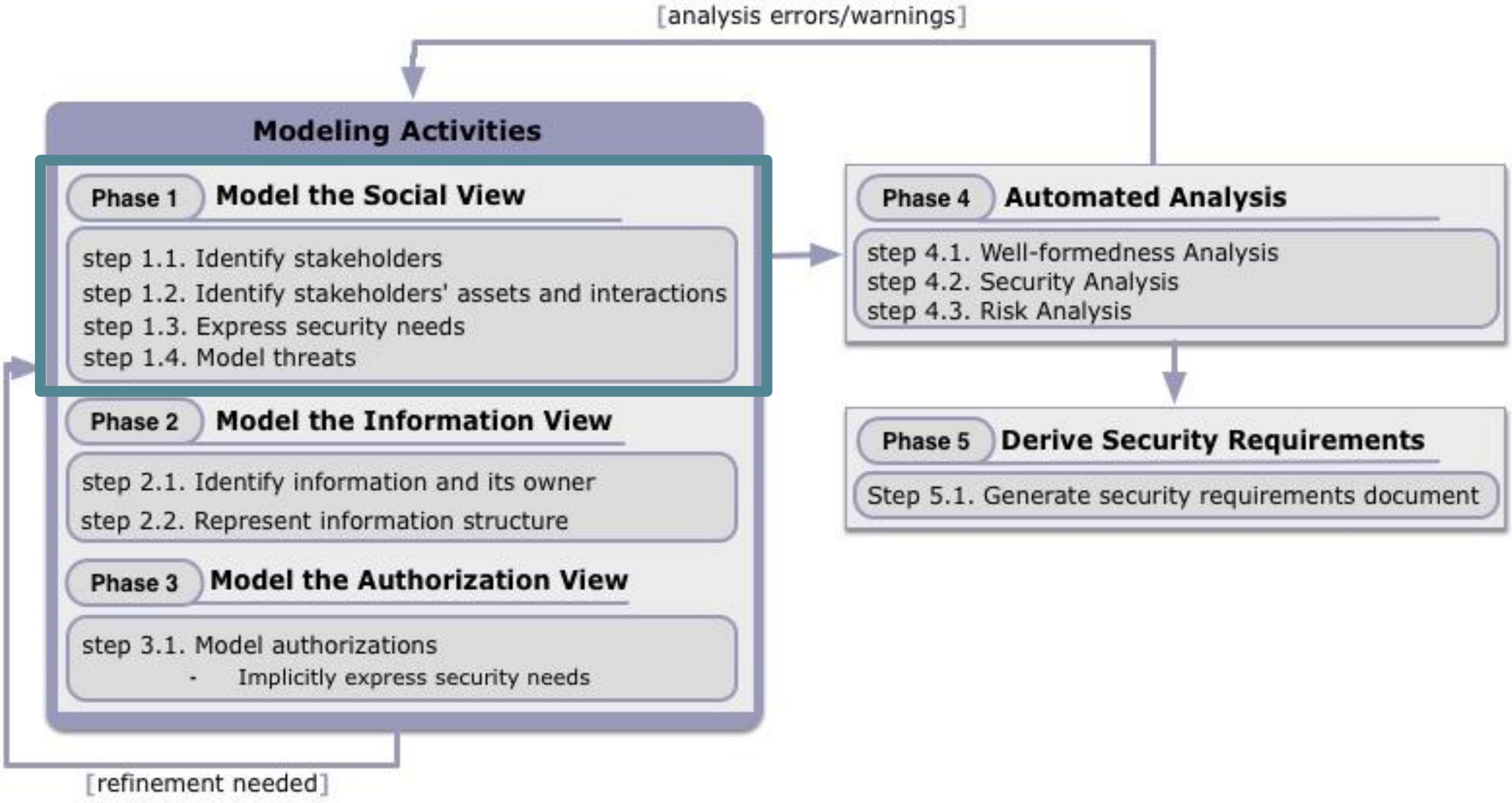
step 4.1. Well-formedness Analysis
step 4.2. Security Analysis
step 4.3. Risk Analysis

Phase 5 Derive Security Requirements

Step 5.1. Generate security requirements document

[refinement needed]

The STS method



[analysis errors/warnings]

Modeling Activities

Phase 1 Model the Social View

step 1.1. Identify stakeholders
step 1.2. Identify stakeholders' assets and interactions
step 1.3. Express security needs
step 1.4. Model threats

Phase 2 Model the Information View

step 2.1. Identify information and its owner
step 2.2. Represent information structure

Phase 3 Model the Authorization View

step 3.1. Model authorizations
- Implicitly express security needs

Phase 4 Automated Analysis

step 4.1. Well-formedness Analysis
step 4.2. Security Analysis
step 4.3. Risk Analysis

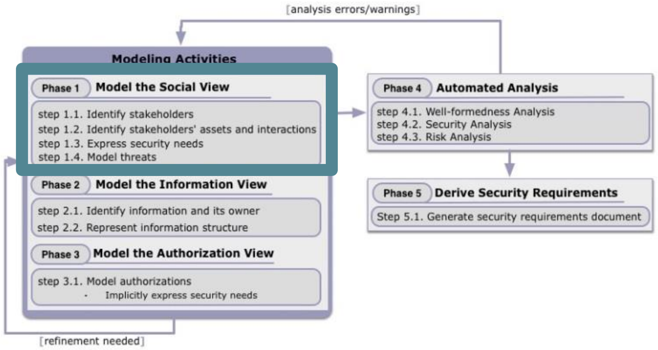
Phase 5 Derive Security Requirements

Step 5.1. Generate security requirements document

[refinement needed]

Phase 1. Modeling the Social View

- Step 1.1 Identify stakeholders
  - Agents and roles
- Step 1.2 Identify assets and interactions
  - Assets: goals, documents
  - Interactions: goal delegations and document provisions
- Step 1.3 Express security needs
  - Express expectations concerning security over interactions
    - Elicited from the stakeholders
- Step 1.4 Model threats
  - Represents events threatening ass



[analysis errors/warnings]

Modeling Activities

Phase 1 Model the Social View

step 1.1. Identify stakeholders
step 1.2. Identify stakeholders' assets and interactions
step 1.3. Express security needs
step 1.4. Model threats

Phase 2 Model the Information View

step 2.1. Identify information and its owner
step 2.2. Represent information structure

Phase 3 Model the Authorization View

step 3.1. Model authorizations
- Implicitly express security needs

[refinement needed]

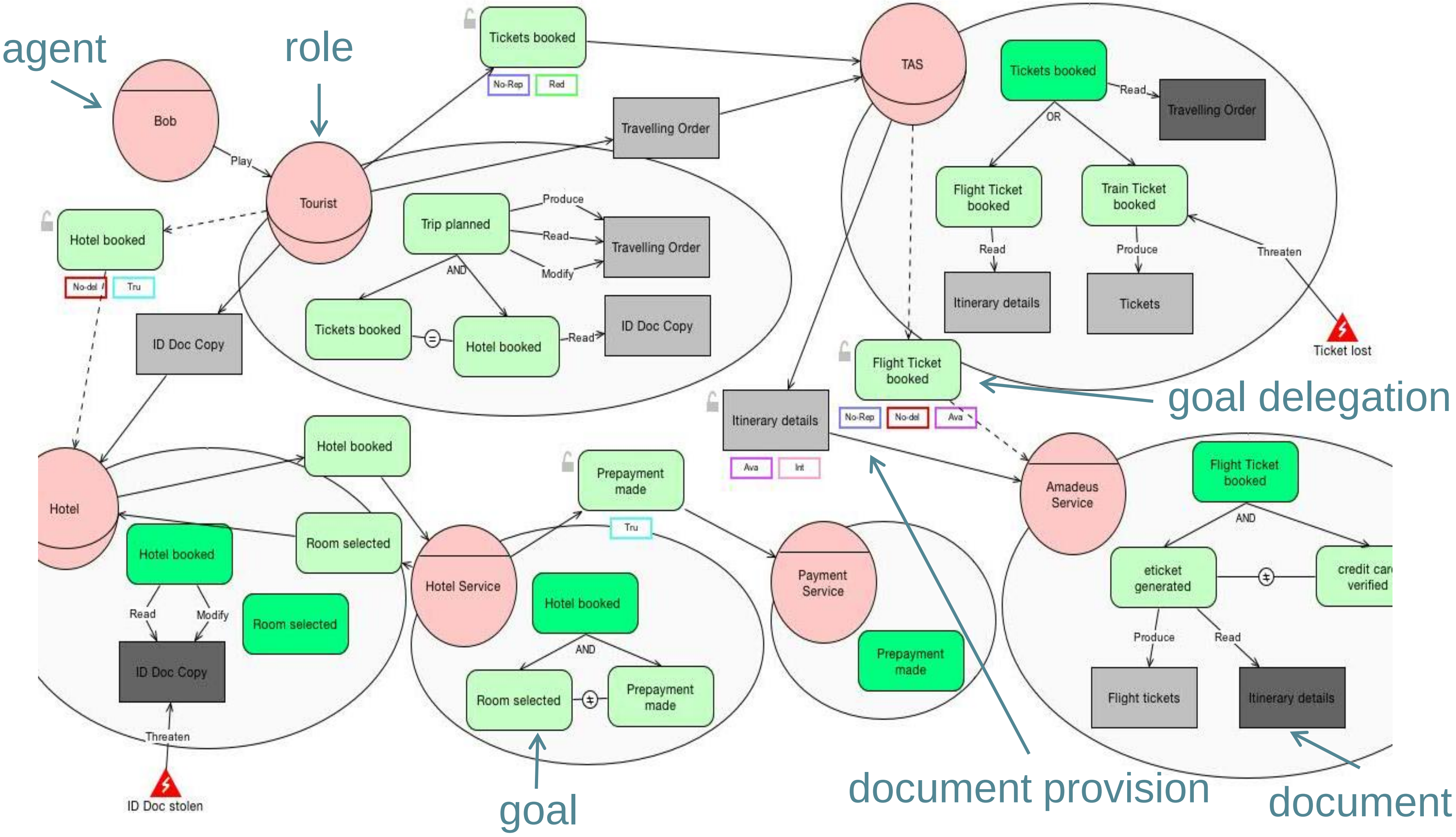
Phase 4 Automated Analysis

step 4.1. Well-formedness Analysis
step 4.2. Security Analysis
step 4.3. Risk Analysis

Phase 5 Derive Security Requirements

Step 5.1. Generate security requirements document

Social view: an example



agent

role

goal delegation

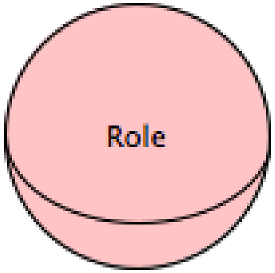
goal

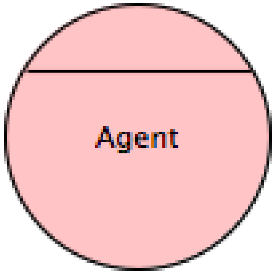
document provision

document

Step 1.1. Identify Stakeholders

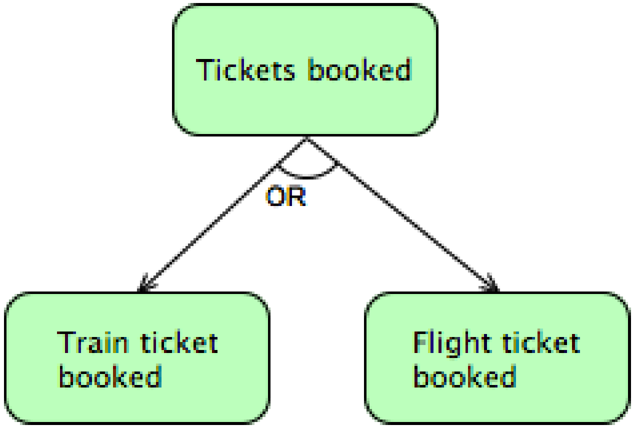
- Elicit roles and agents
  - Role is an abstract characterization of the behavior of an active entity within some context
    - Most participants are unknown at design time
    - e.g., Tourist, Travel Agency Service, Hotel, …
- Agents play (adopt) roles at runtime, and they can change the roles they play
  - e.g., Bob, Fabiano, CheapTravels Inc.
  - Some agents are known, e.g., Amadeus Flight Service



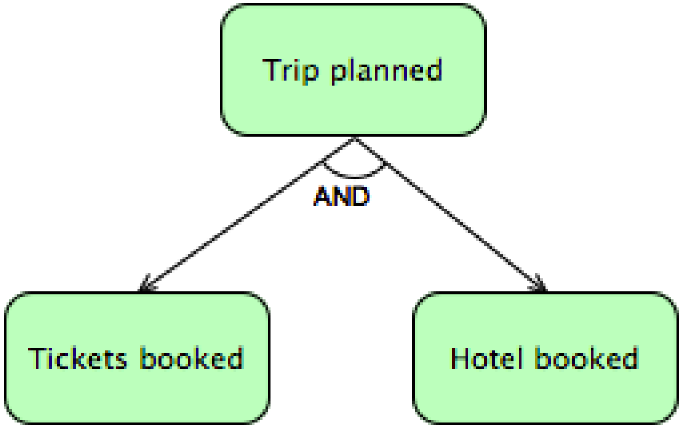


Step 1.2. Identify assets and interactions

- A goal is a state of affairs that an actor intends to achieve
  - e.g., trip planned, flight tickets booked
  - Used to capture motivations and responsibilities of actors
- Goal can be decomposed (refined)



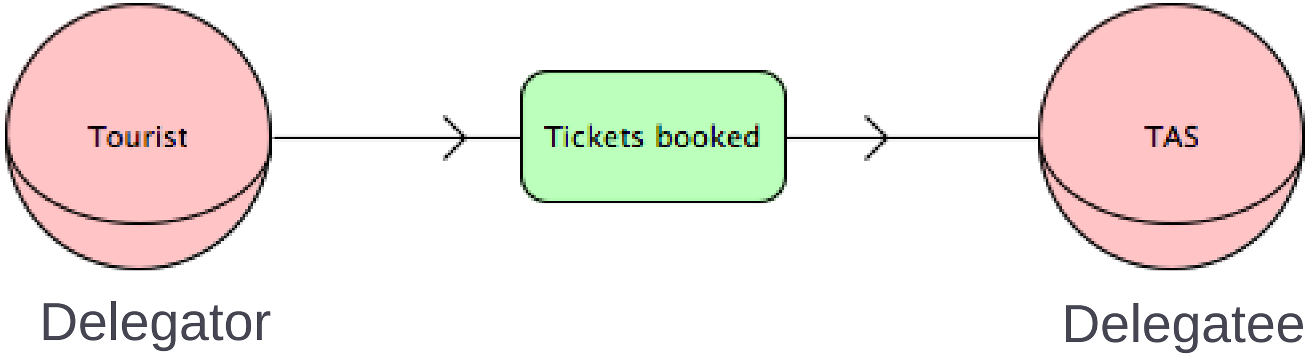
Or-decomposition



And-decomposition

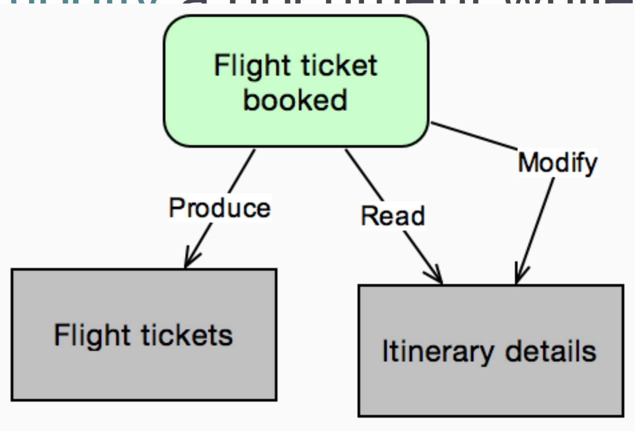
Step 1.2. Identify assets and interactions

- Goal delegation
  - A Delegator actor delegates the fulfillment of a goal (delegatum) to a different actor (delegatee)
    - Lack of capability or transfer of responsibility
  - e.g., Tourist is not capable of booking the tickets on his own, he depends on a Travel Agency Service to achieve this goal
  - In STS-ml, only leaf goals can be delegated



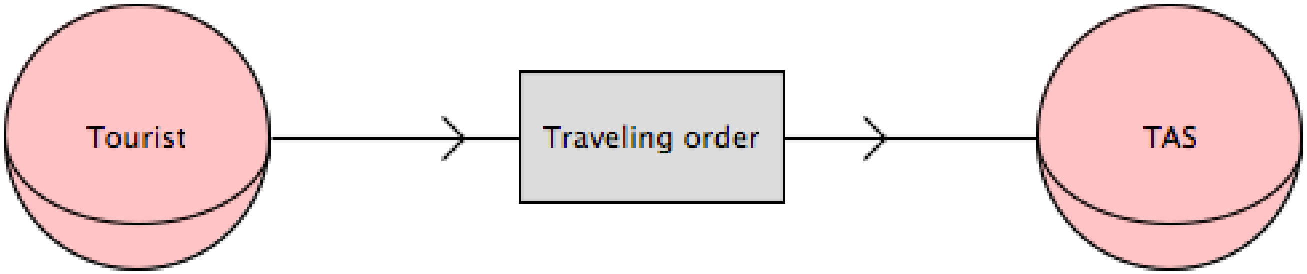
Step 1.2. Identify assets and interactions

- A document represents an exchangeable entity which may contain some information
  - Actors possess or manipulate documents to achieve their goals
- Goal-document relationships
  - An actor may read one or more documents to fulfill a goal
  - An actor may produce documents while fulfilling a goal
  - An actor may modify a document while fulfilling a goal



Step 1.2. Identify assets and interactions

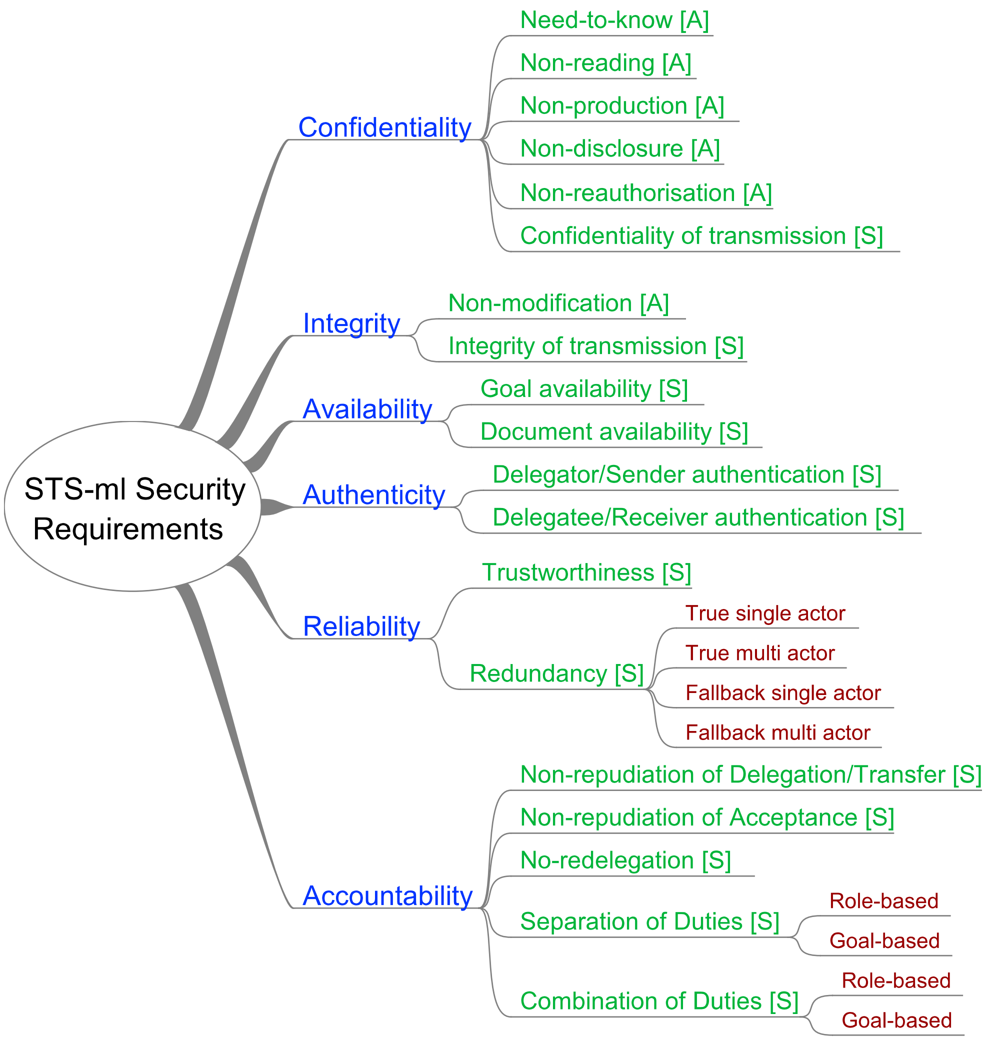
- Document exchange: *document transmission*
  - Captures exchange of documents between a sender actor and a receiver actor
  - Sender: an actor that possesses the document
  - Receiver: an actor that might need the transmitted document(s) to achieve its goals



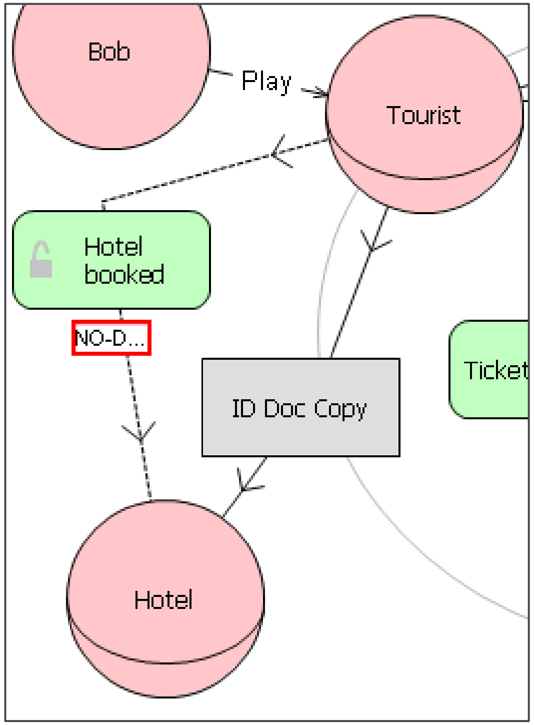
Sender

Receiver

Step 1.3. Express security needs

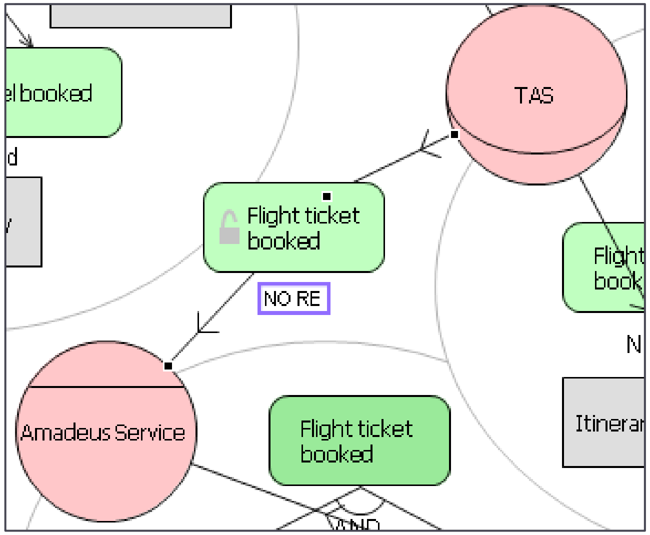


Step 1.3. Express security needs



No-delegation

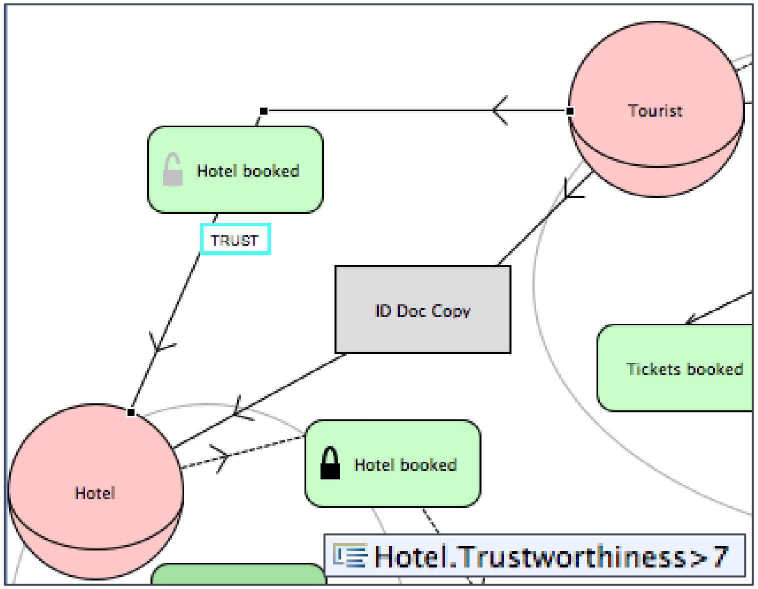
The re-delegation of the fulfilment of a goal is forbidden



Non-repudiation

- The delegator cannot repudiate he delegated
- The delegatee cannot repudiate he accepted the delegation

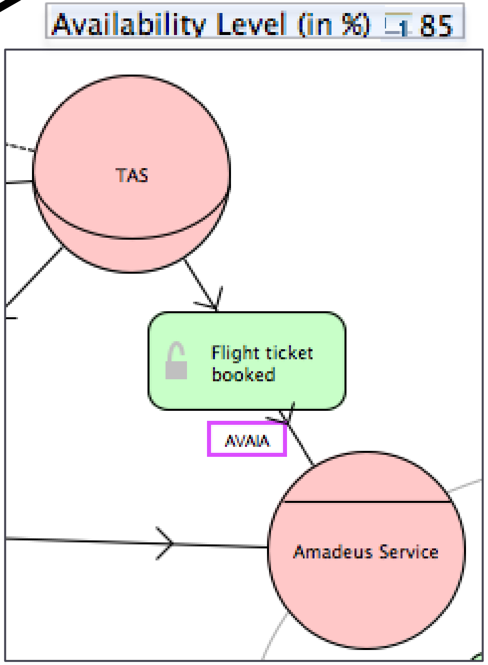
Step 1.3. Express security needs



Hotel.Trustworthiness>7

Min trustworthiness level

The delegation of the goal will take place only if the delegatee has a min required trustworthiness level



Availability Level (in %) 85

Availability

The delegatee should ensure a
min availability level for the delegated goal

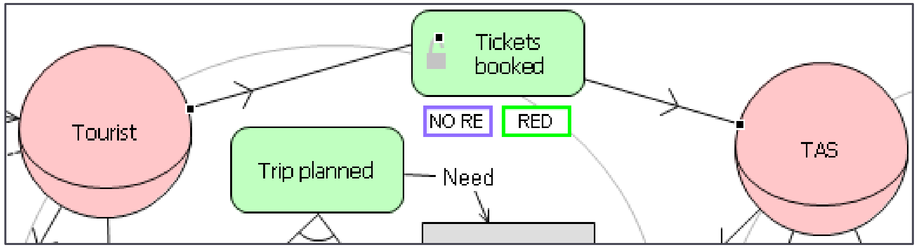
Step 1.3. Expressing security needs

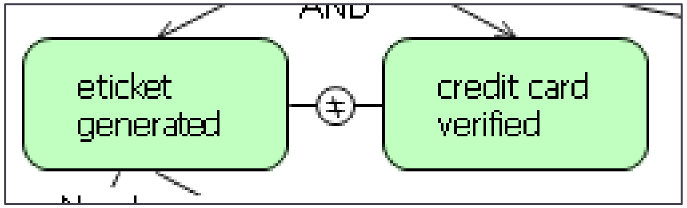
Redundancy

- Alternative ways of achieving a goal
- Different redundancy types
  - True and Fallback
  - Single and Multi Actor

Combine/ Incompatible BoD/SoD

- Two goals shall be achieved by different (the same) actors
- Two roles are incompatible, i.e., cannot be played by the same agent

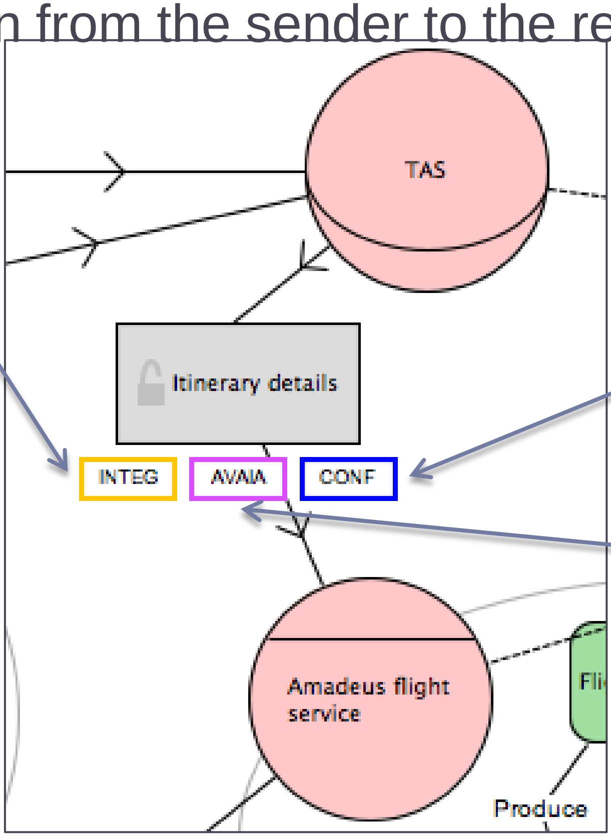




Step 1.3. Expressing security needs

Integrity of transmission

The sender should ensure that the document shall not be altered during the transmission from the sender to the receiver



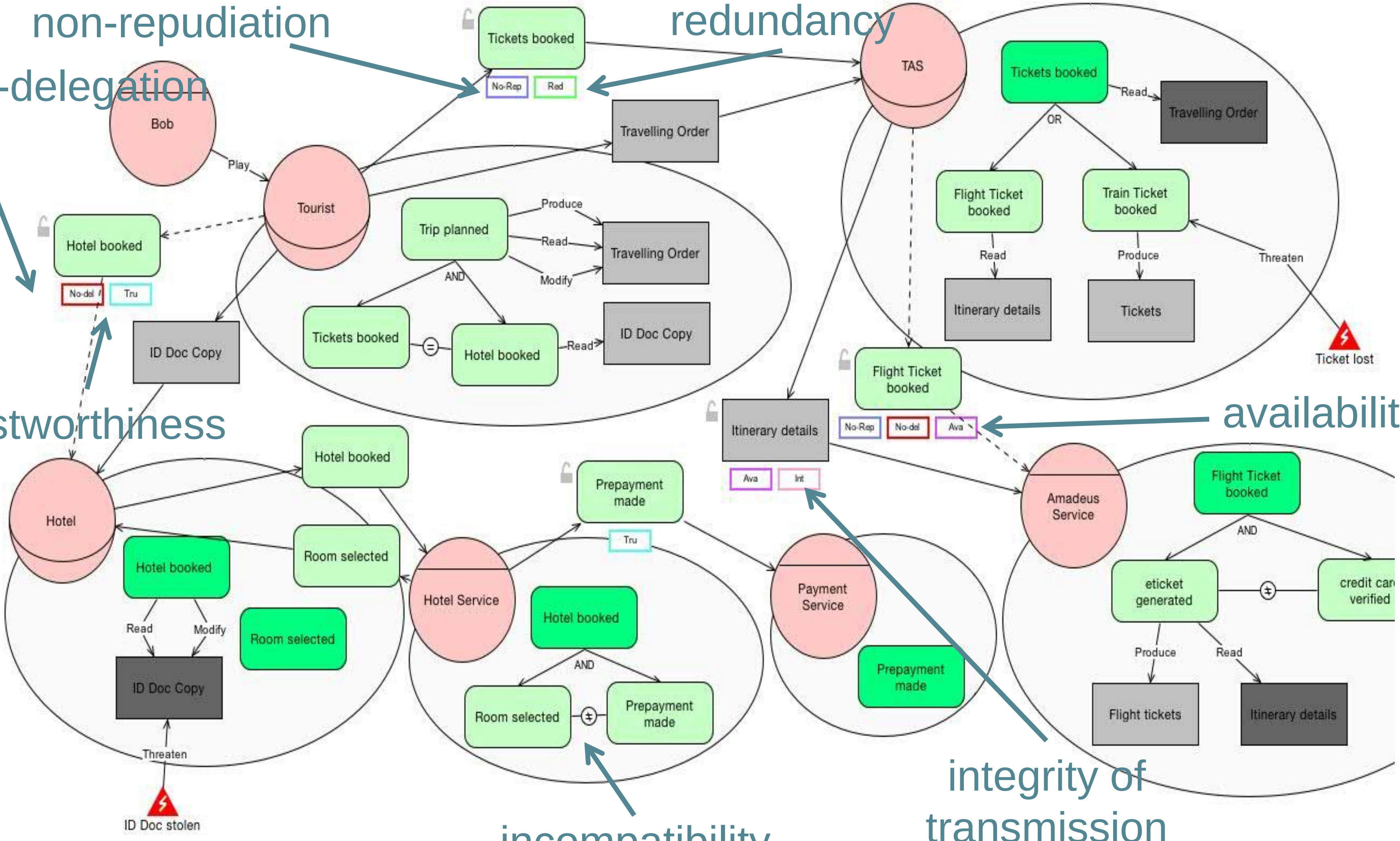
Confidentiality of transmission

The sender should ensure the confidentiality of transmission for the transmitted document

Availability

The sender should ensure a min availability level (in %) for the transmitted document

Social view: expressing security needs



non-repudiation

redundancy

non-delegation

trustworthiness

availability

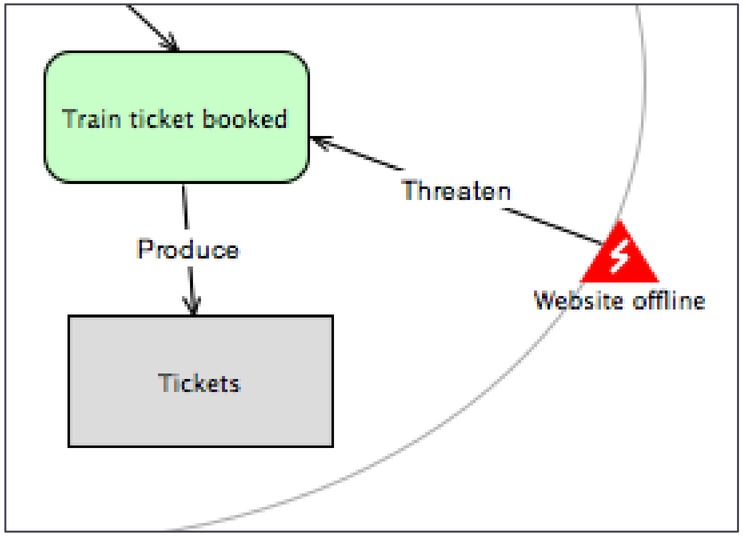
incompatibility (SoD:Separation of Duties)

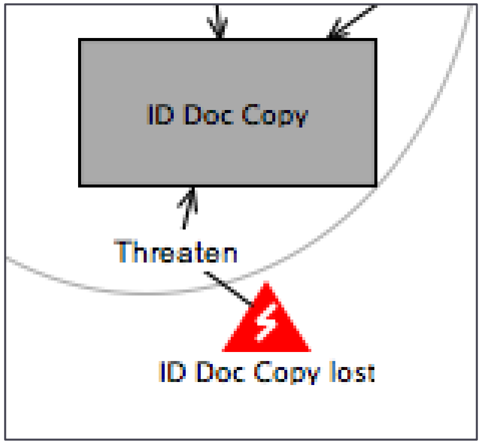
integrity of transmission

Step 1.4. Modeling risks

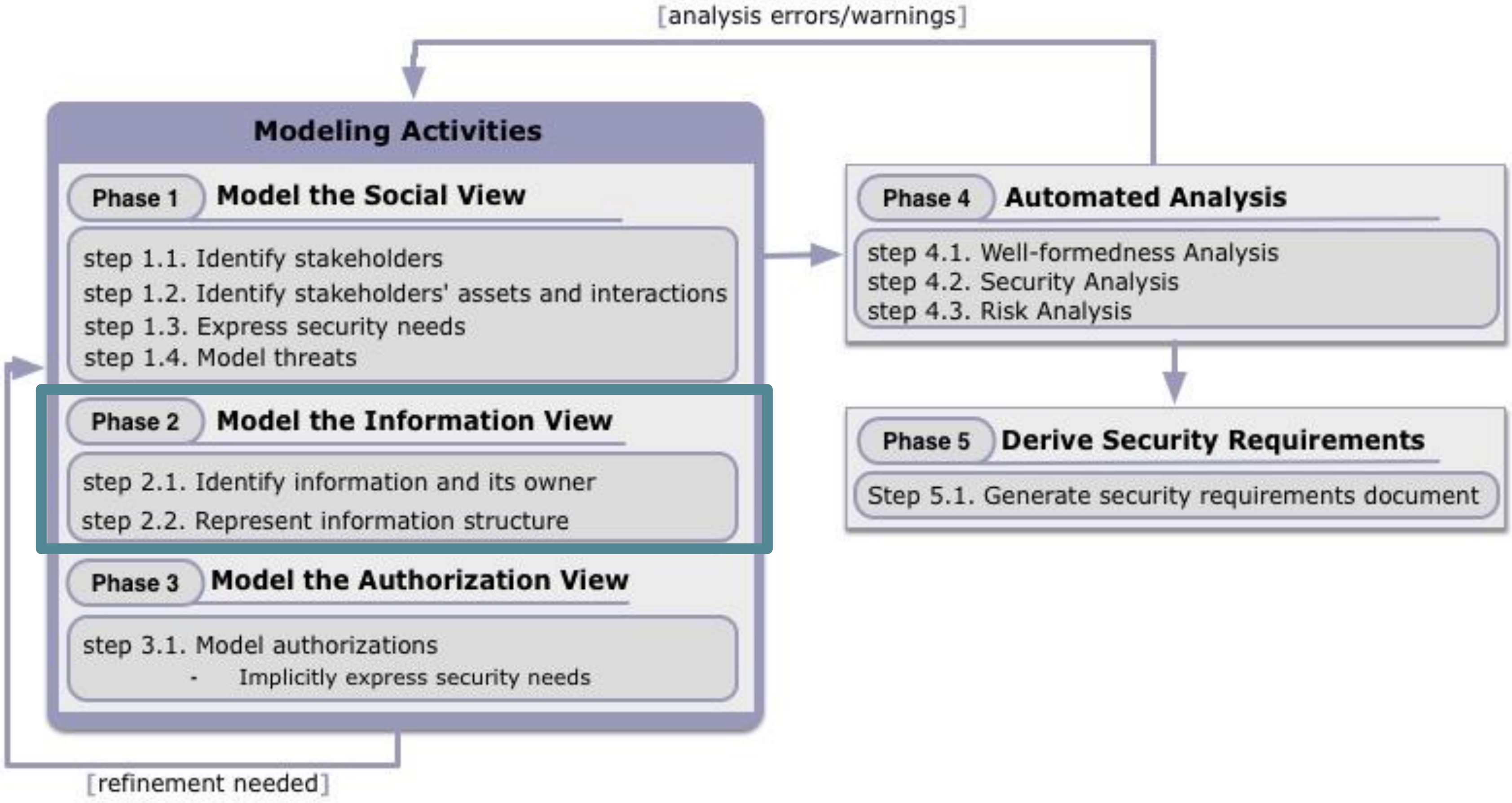
Represent events threatening assets

- Over goals
  - Goal cannot be reached
- Over documents
  - Document becomes unavailable





The STS method



[analysis errors/warnings]

Modeling Activities

Phase 1 Model the Social View

step 1.1. Identify stakeholders
step 1.2. Identify stakeholders' assets and interactions
step 1.3. Express security needs
step 1.4. Model threats

Phase 2 Model the Information View

step 2.1. Identify information and its owner
step 2.2. Represent information structure

Phase 3 Model the Authorization View

step 3.1. Model authorizations
- Implicitly express security needs

Phase 4 Automated Analysis

step 4.1. Well-formedness Analysis
step 4.2. Security Analysis
step 4.3. Risk Analysis

Phase 5 Derive Security Requirements

Step 5.1. Generate security requirements document

[refinement needed]

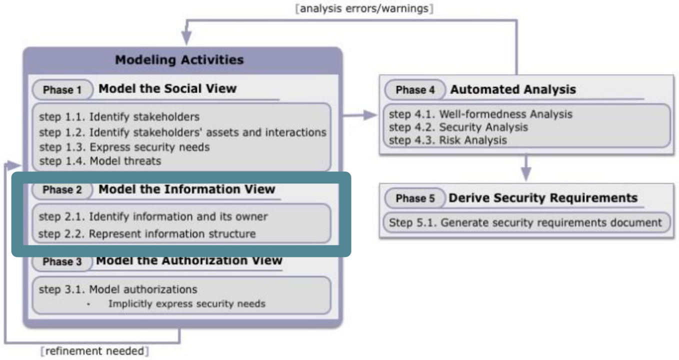
Phase 2. Modeling the Information View

- Confidentiality requirements are concerned with protecting the disclosure and usage of information
  - It is important to know who are information owners
  - It is important to know what is the informational content of the documents actors possess and/or manipulate while achieving their goals

Phase 2. Modeling the Information View

- Step 2.1 Identify information and its owner
  - Documents represent information
  - Represent the owners of different information

- Step 2.2 Represent information structure
  - Tangible By: information → document
  - Part Of: Info (doc) → Info (doc)



[analysis errors/warnings]

Modeling Activities

Phase 1 Model the Social View

step 1.1. Identify stakeholders
step 1.2. Identify stakeholders' assets and interactions
step 1.3. Express security needs
step 1.4. Model threats

Phase 2 Model the Information View

step 2.1. Identify information and its owner
step 2.2. Represent information structure

Phase 3 Model the Authorization View

step 3.1. Model authorizations
- Implicitly express security needs

[refinement needed]

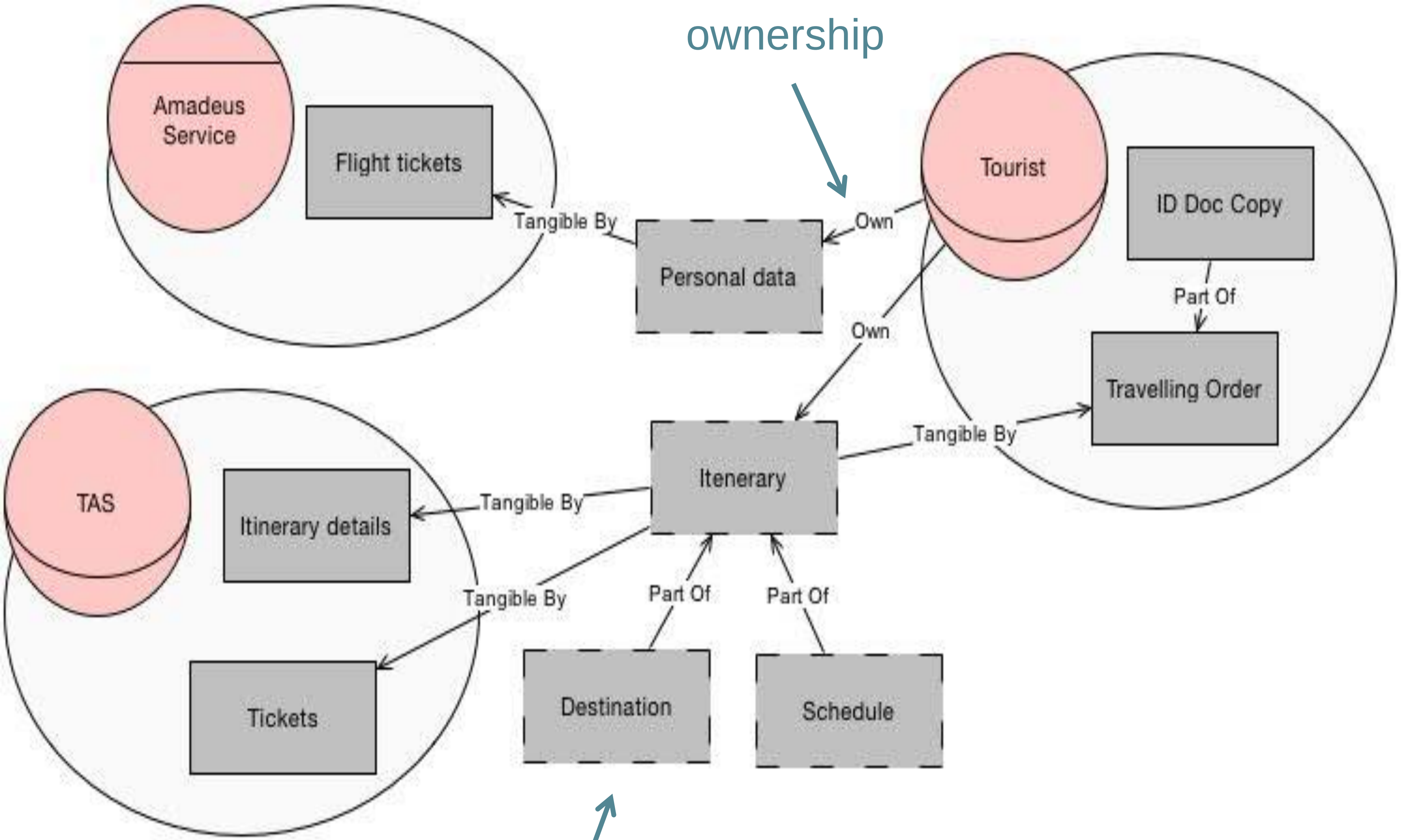
Phase 4 Automated Analysis

step 4.1. Well-formedness Analysis
step 4.2. Security Analysis
step 4.3. Risk Analysis

Phase 5 Derive Security Requirements

Step 5.1. Generate security requirements document

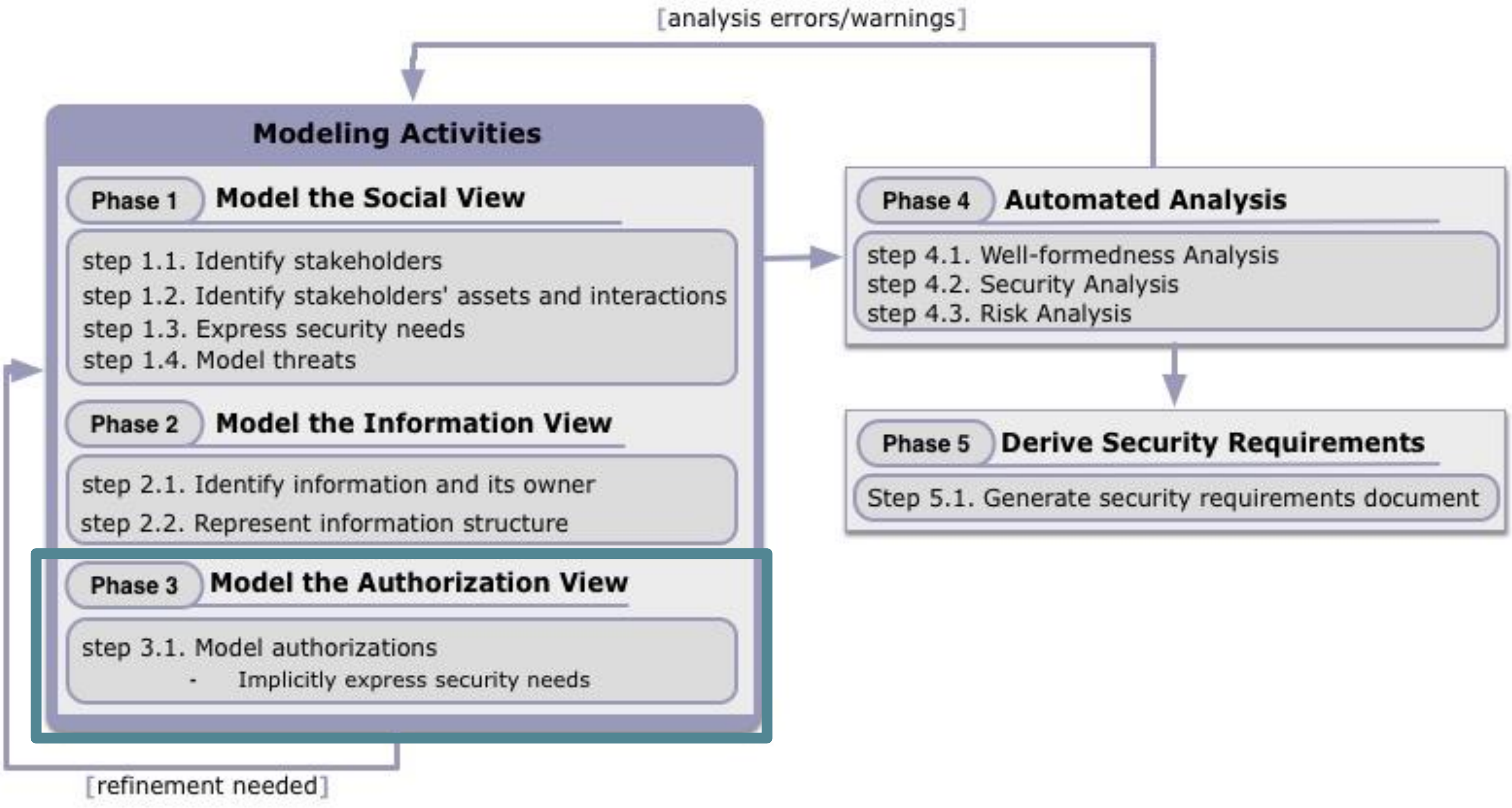
Information view: an example



ownership

information

The STS method



[analysis errors/warnings]

Modeling Activities

Phase 1 **Model the Social View**

step 1.1. Identify stakeholders
step 1.2. Identify stakeholders' assets and interactions
step 1.3. Express security needs
step 1.4. Model threats

Phase 2 **Model the Information View**

step 2.1. Identify information and its owner
step 2.2. Represent information structure

Phase 3 **Model the Authorization View**

step 3.1. Model authorizations

- Implicitly express security needs

Phase 4 **Automated Analysis**

step 4.1. Well-formedness Analysis
step 4.2. Security Analysis
step 4.3. Risk Analysis

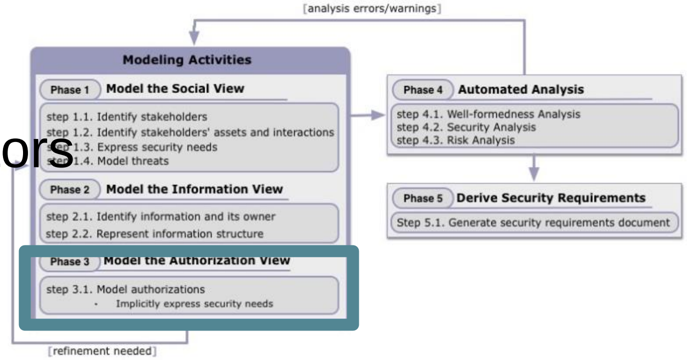
Phase 5 **Derive Security Requirements**

Step 5.1. Generate security requirements document

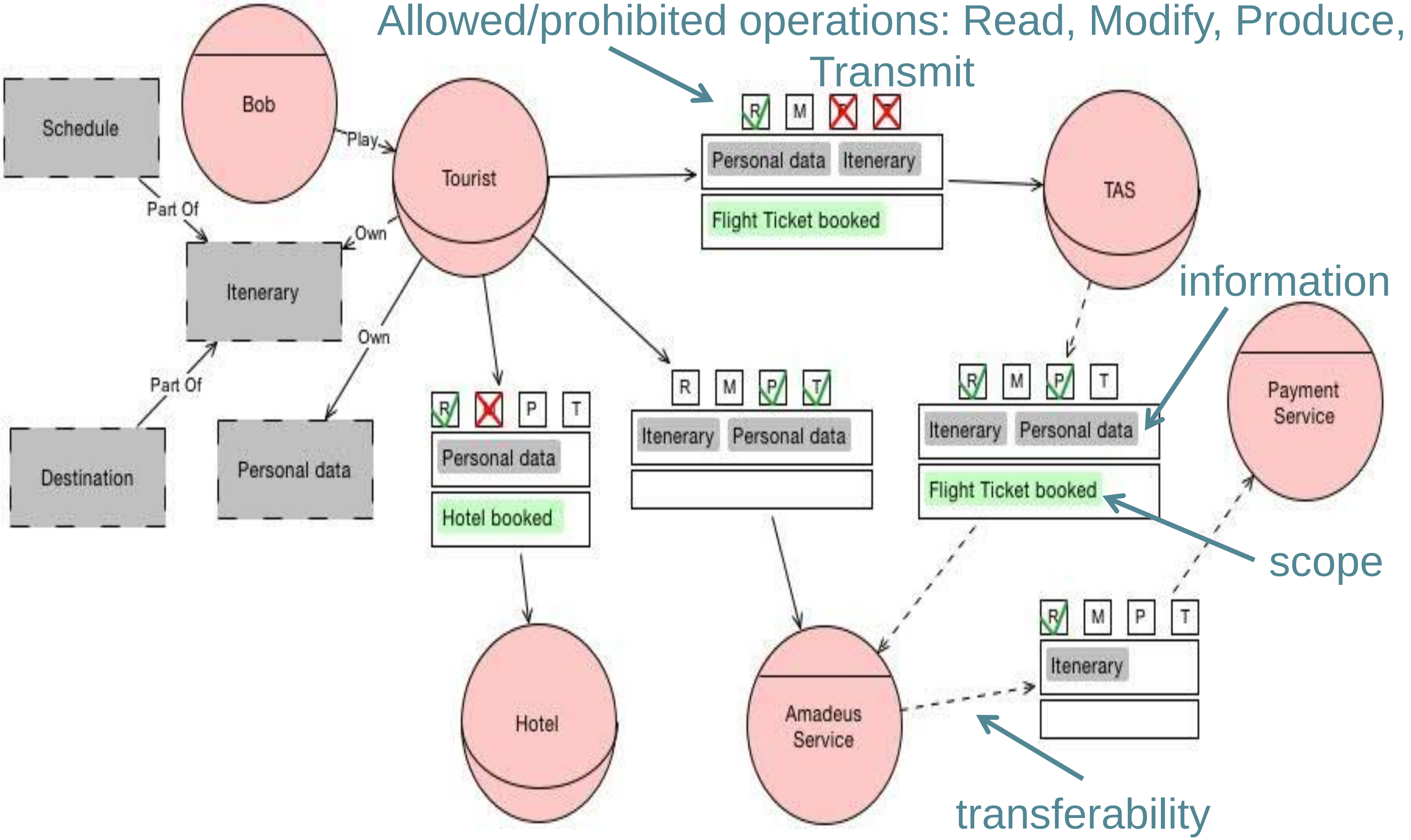
[refinement needed]

Phase 3. Modeling the Authorization View

- Step 3.1 Model authorizations
  - Transfer of rights/permissions and/or prohibitions between actors
- Authorizations about information, specifying
  - Scope of usage (a set of goals)
    - The customer permits the travel agency to read her personal data only to book the tickets
  - Allowed/prohibited operations: read, modify, produce, transmit
  - Transferability
    - Further propagate rights to other actors


[analysis errors/warnings]
Modeling Activities
Phase 1 Model the Social View
step 1.1. Identify stakeholders
step 1.2. Identify stakeholders' assets and interactions
step 1.3. Express security needs
step 1.4. Model threats
Phase 2 Model the Information View
step 2.1. Identify information and its owner
step 2.2. Represent information structure
Phase 3 Model the Authorization View
step 3.1. Model authorizations
· Implicitly express security needs
Phase 4 Automated Analysis
step 4.1. Well-formedness Analysis
step 4.2. Security Analysis
step 4.3. Risk Analysis
Phase 5 Derive Security Requirements
Step 5.1. Generate security requirements document
[refinement needed]

Authorization view: an example



Allowed/prohibited operations: Read, Modify, Produce, Transmit

information

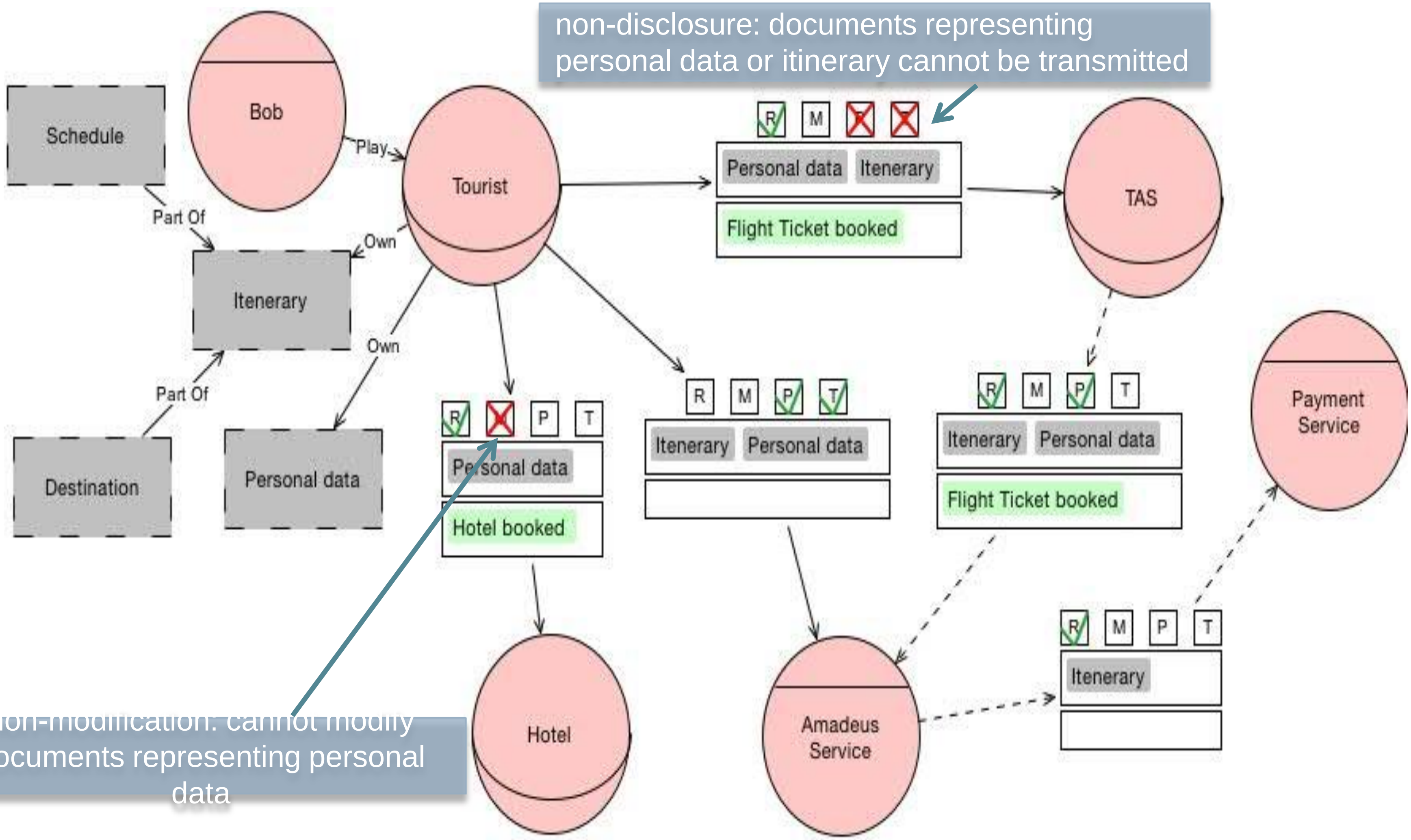
scope

transferability

Expressing security needs via authorizations

- Security needs via authorizations are expressed by prohibiting certain operations and limiting the scope
  - Need-to-know ← limiting the scope
  - Non-reading ← not allowing usage
  - Non-modification ← not allowing modification
  - Non-production ← not allowing production
  - Non-disclosure ← not allowing distribution
  - Non-reauthorization ← authorization transferability is set to false

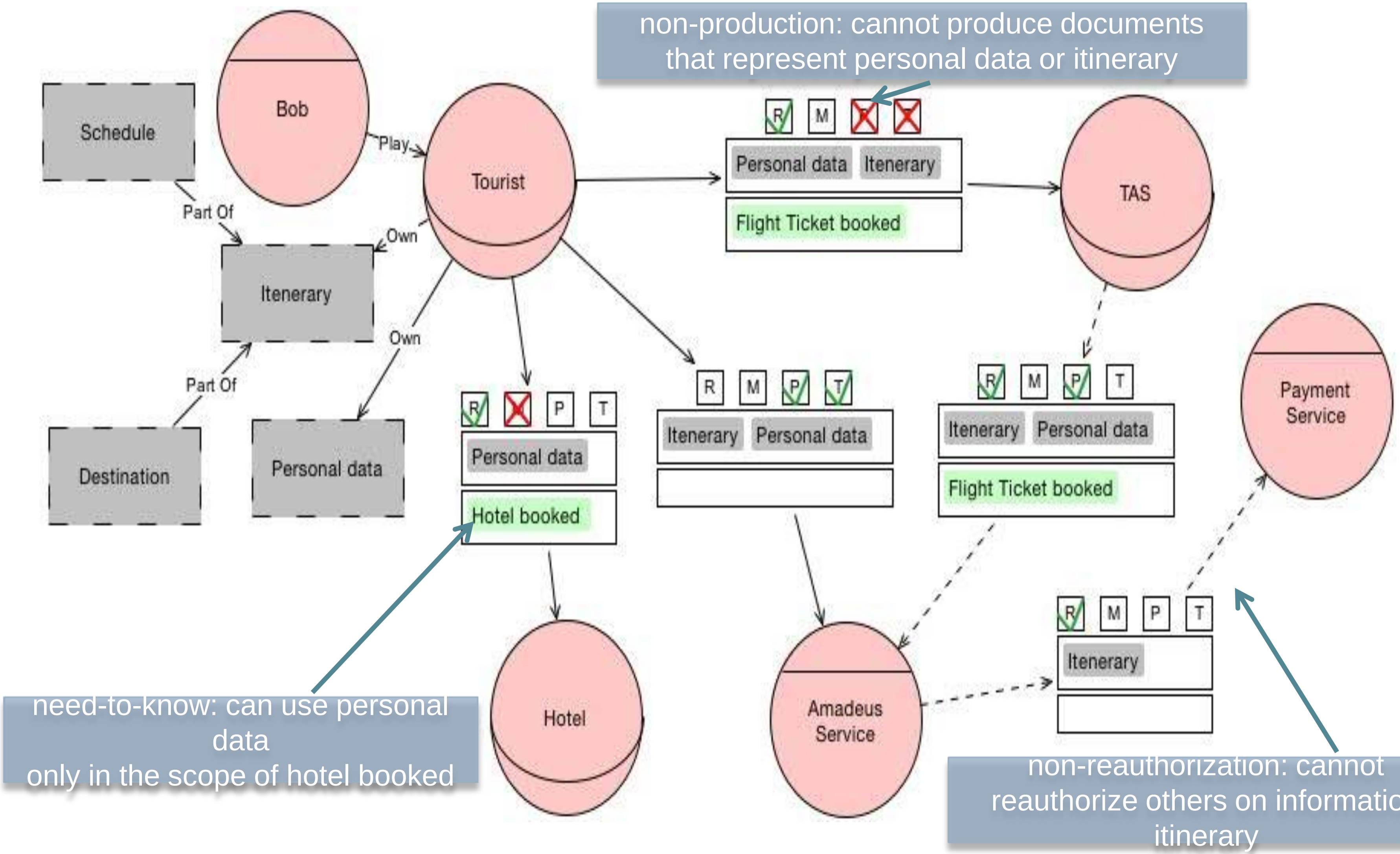
Security needs via authorizations



non-disclosure: documents representing personal data or itinerary cannot be transmitted

non-modification: cannot modify documents representing personal data

Security needs via authorisations

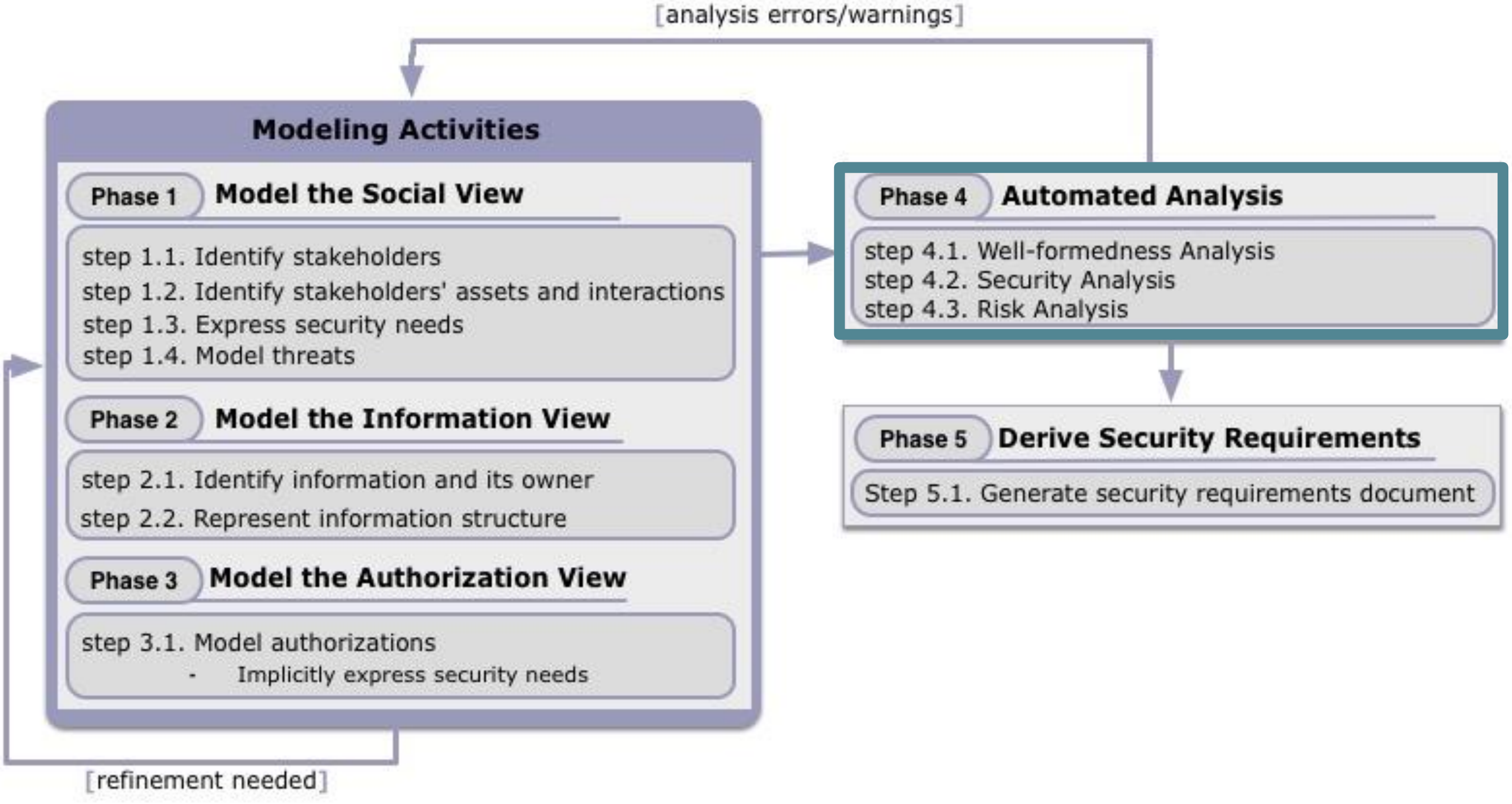


non-production: cannot produce documents that represent personal data or itinerary

need-to-know: can use personal data
only in the scope of hotel booked

non-reauthorization: cannot reauthorize others on information itinerary

The STS method



[analysis errors/warnings]

Modeling Activities

Phase 1 **Model the Social View**

step 1.1. Identify stakeholders
step 1.2. Identify stakeholders' assets and interactions
step 1.3. Express security needs
step 1.4. Model threats

Phase 2 **Model the Information View**

step 2.1. Identify information and its owner
step 2.2. Represent information structure

Phase 3 **Model the Authorization View**

step 3.1. Model authorizations
- Implicitly express security needs

Phase 4 **Automated Analysis**

step 4.1. Well-formedness Analysis
step 4.2. Security Analysis
step 4.3. Risk Analysis

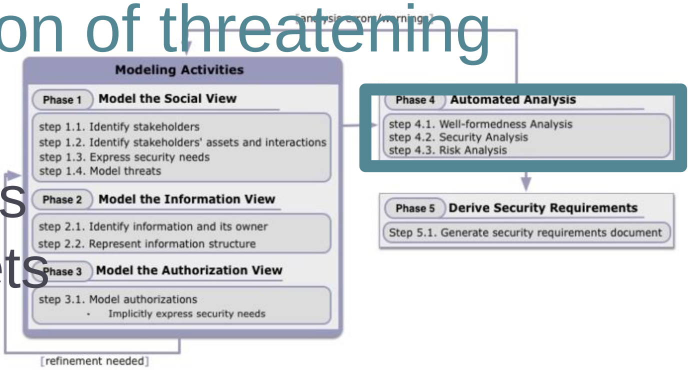
Phase 5 **Derive Security Requirements**

Step 5.1. Generate security requirements document

[refinement needed]

Phase 4. Automated analysis

- Step 4.1 Well-formedness Analysis
  - Is the STS-ml model syntactically well-formed?
  - E.g.: part-of cycles, contribution cycles
- Step 4.2 Security Analysis: security properties verification
  - Security requirements cannot be fulfilled in the modeled socio-technical system
  - E.g.: violation of no-delegation, non-usage, non-disclosure, separation of duty, …
- Step 4.3 Risk Analysis: propagation of threatening events
  - How does the specification of events threatening assets affect other assets



[analysis errors/warnings]

Modeling Activities

Phase 1 Model the Social View

step 1.1. Identify stakeholders
step 1.2. Identify stakeholders' assets and interactions
step 1.3. Express security needs
step 1.4. Model threats

Phase 2 Model the Information View

step 2.1. Identify information and its owner
step 2.2. Represent information structure

Phase 3 Model the Authorization View

step 3.1. Model authorizations
- Implicitly express security needs

[refinement needed]

Phase 4 Automated Analysis

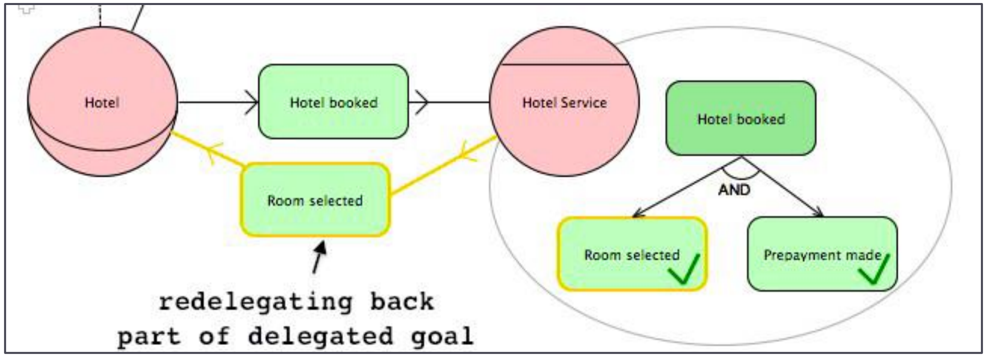
step 4.1. Well-formedness Analysis
step 4.2. Security Analysis
step 4.3. Risk Analysis

Phase 5 Derive Security Requirements

Step 5.1. Generate security requirements document

Step 4.1. Well-formedness analysis

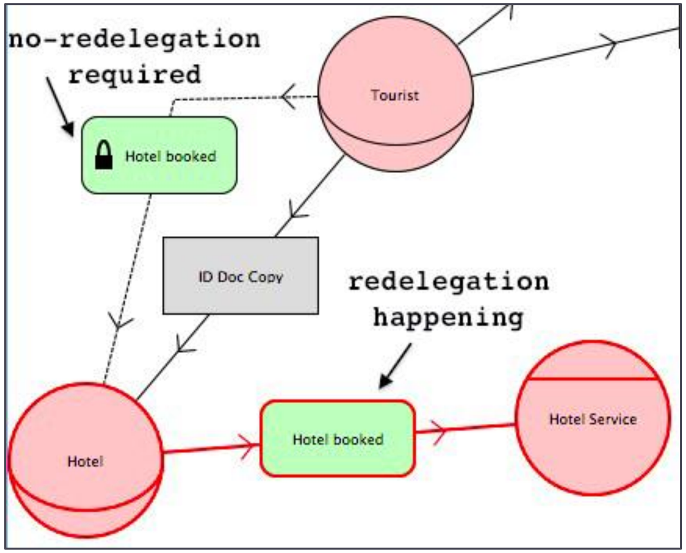
- Post-modelling well-formedness checks
  - Give warnings or errors and visualize to designer
- Current checks
  - Single goal decompositions
  - Leaf goal delegation
  - Delegation cycles
  - Part-of cycles
  - Ownership
    - Information without owner
  - Authorisations
    - Not empty, no duplicates



warning

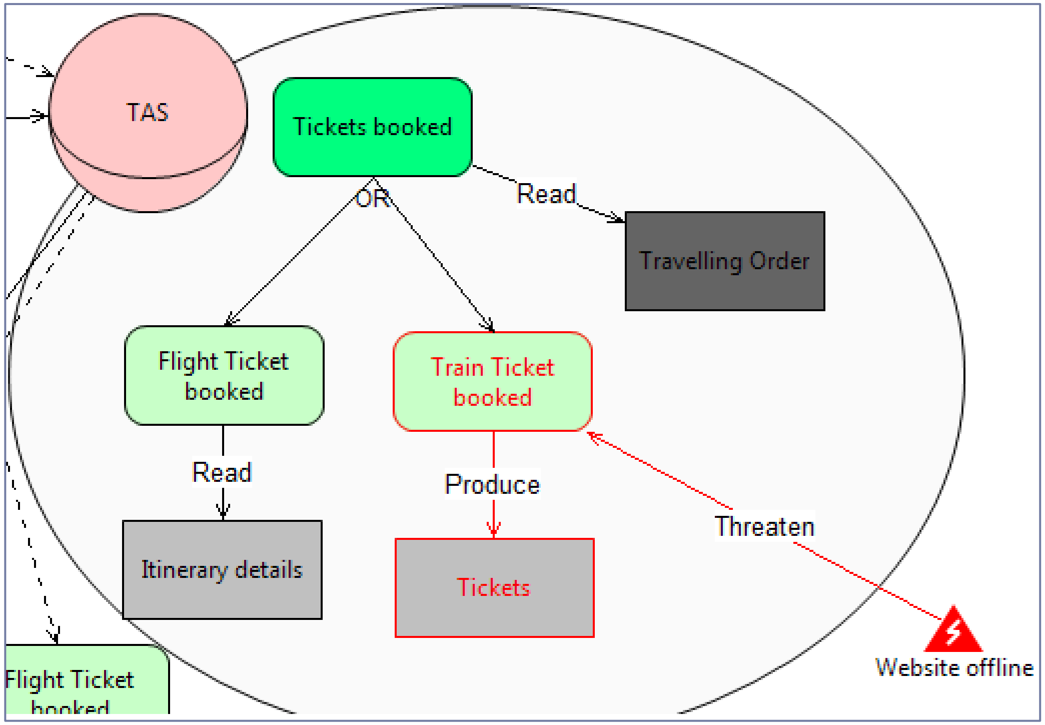
Step 4.2. Security analysis

- Is it possible in the model that a security requirement is violated?
  - Identify and visualize possible problems
  - The engineer fixes the problem



error

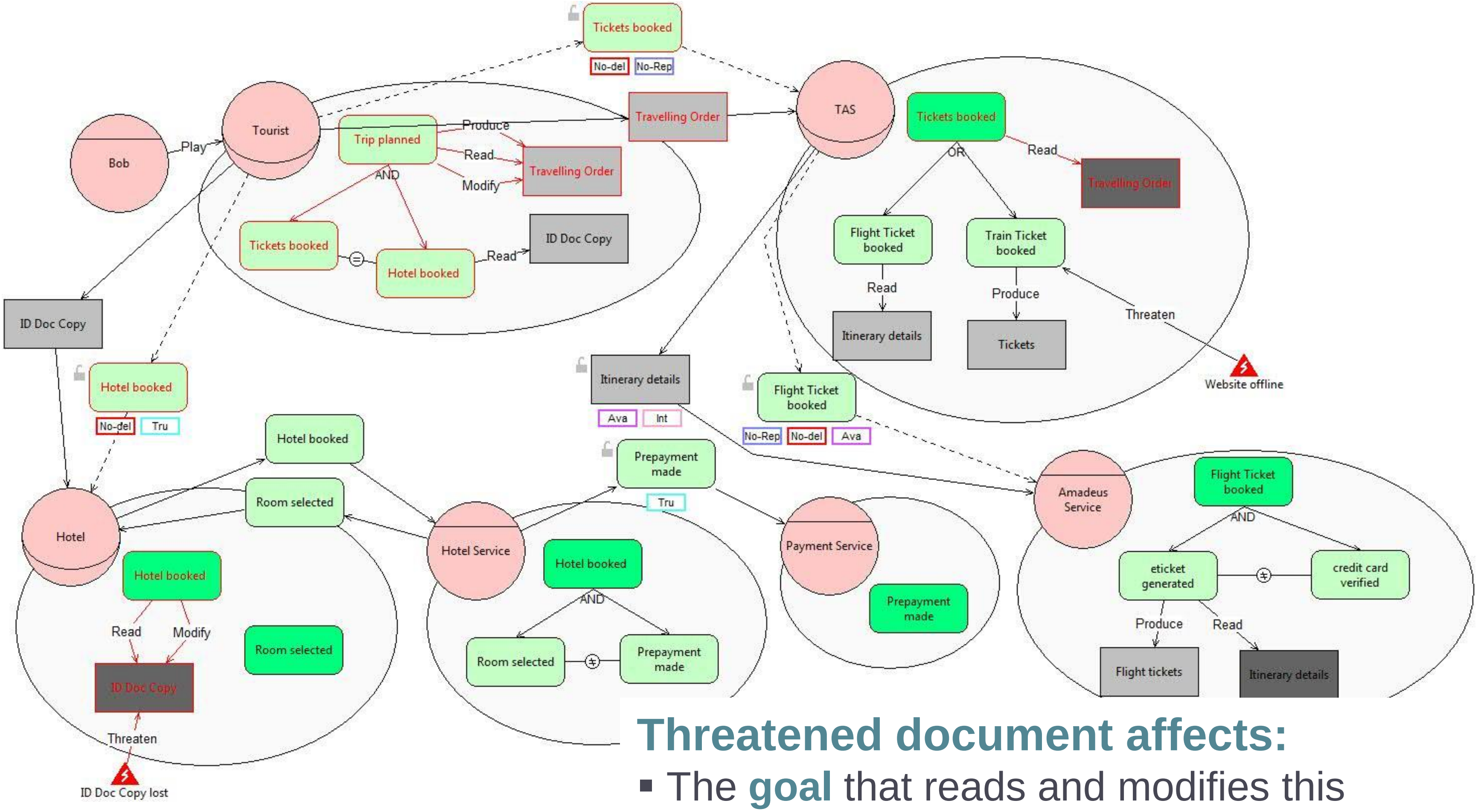
Step 4.3. Risk analysis



Threatened goal affects:

- The document it produces

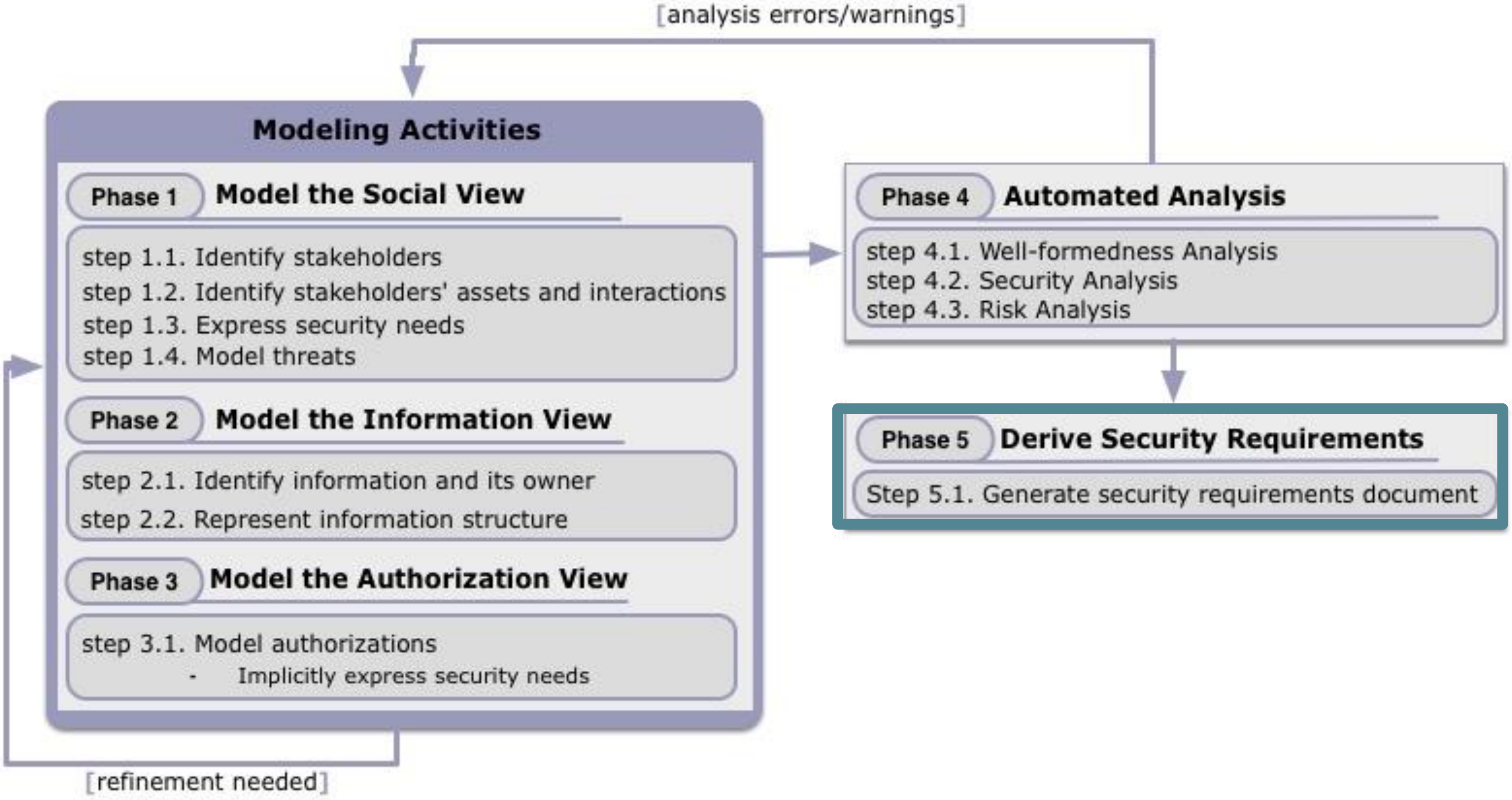
Step 4.3. Risk analysis



Threatened document affects:

- The **goal** that reads and modifies this document
- If the goal is delegated, the goal of the delegator

The STS method



[analysis errors/warnings]

Modeling Activities

Phase 1 **Model the Social View**

step 1.1. Identify stakeholders
step 1.2. Identify stakeholders' assets and interactions
step 1.3. Express security needs
step 1.4. Model threats

Phase 2 **Model the Information View**

step 2.1. Identify information and its owner
step 2.2. Represent information structure

Phase 3 **Model the Authorization View**

step 3.1. Model authorizations
- Implicitly express security needs

Phase 4 **Automated Analysis**

step 4.1. Well-formedness Analysis
step 4.2. Security Analysis
step 4.3. Risk Analysis

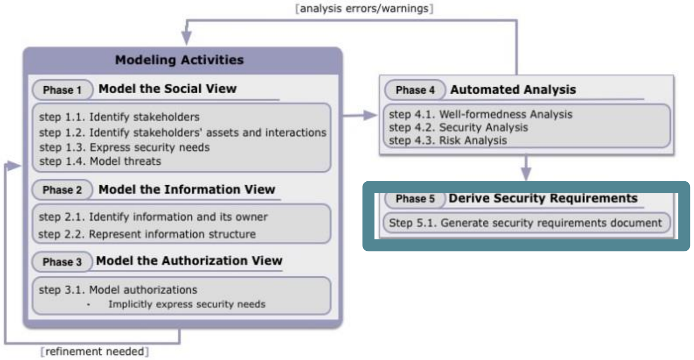
Phase 5 **Derive Security Requirements**

Step 5.1. Generate security requirements document

[refinement needed]

Phase 5. Derive security requirements

- Requirements models are useful for communication purposes with the stakeholders
- Requirements specifications tell designers what the system has to implement
  - In STS-ml, security requirements specifications are automatically derived from requirements models
  - Output: security requirements document



[analysis errors/warnings]

Modeling Activities

Phase 1 Model the Social View

step 1.1. Identify stakeholders
step 1.2. Identify stakeholders' assets and interactions
step 1.3. Express security needs
step 1.4. Model threats

Phase 2 Model the Information View

step 2.1. Identify information and its owner
step 2.2. Represent information structure

Phase 3 Model the Authorization View

step 3.1. Model authorizations
- Implicitly express security needs

[refinement needed]

Phase 4 Automated Analysis

step 4.1. Well-formedness Analysis
step 4.2. Security Analysis
step 4.3. Risk Analysis

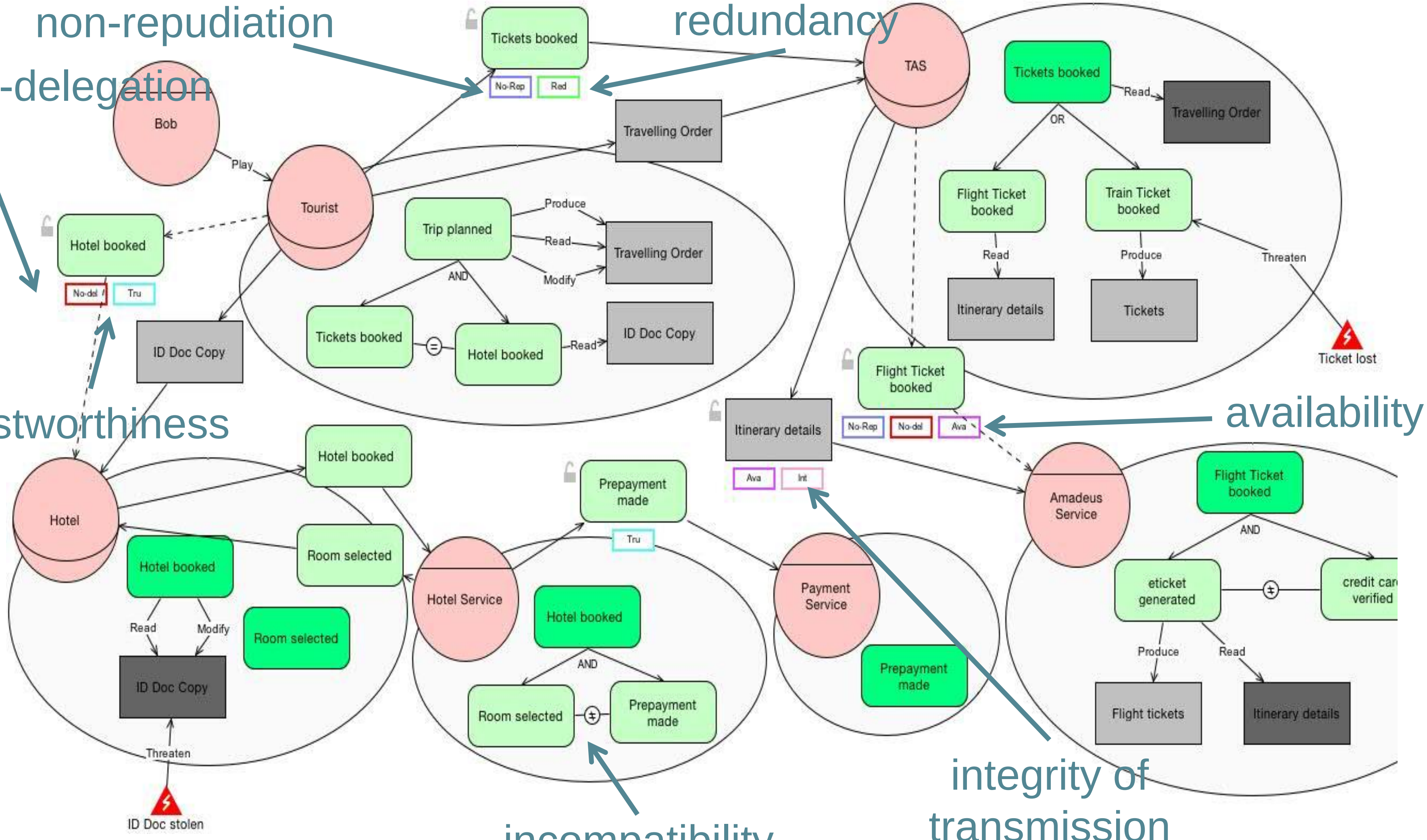
Phase 5 Derive Security Requirements

Step 5.1. Generate security requirements document

Step 5.1. Derive security requirements

- In STS-ml
  - Security requirements constrain interactions in contractual terms
  - These contracts are expressed for each required security need
    - For each security need expressed from one actor to the other, a requirement is generated on the opposite direction to express compliance with the required security need
  - For each requirement
    - Requestor, Requirement, Responsible

Social view: expressing security needs



non-repudiation

redundancy

non-delegation

trustworthiness

availability

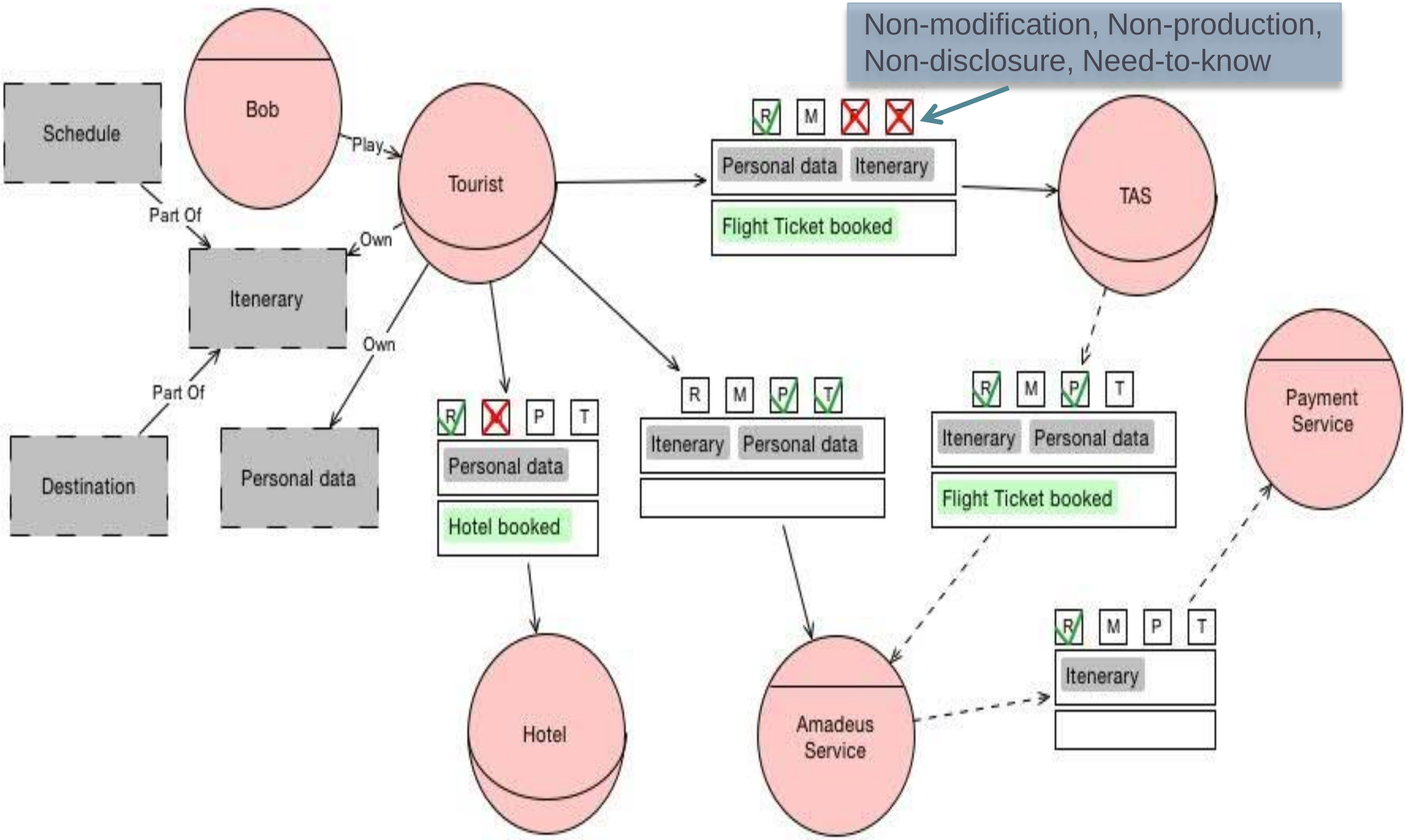
integrity of transmission

incompatibility (SoD:Separation of Duties)

Step 5.1. Derive security requirements

Responsible	Security Requirement	Requester
TAS	non-repudiation-of-acceptance (delegated(Tourist,TAS,tickets booked))	Tourist
Tourist	non-repudiation-of-delegation (delegated(Tourist,TAS,tickets booked))	TAS
TAS	true-redundancy-multiple-actor(tickets booked)	Tourist
Hotel	no-delegation(hotel booked)	Tourist
Amadeus FS	integrity-of-transmission (provided(TAS,Amadeus Service,Itinerary details)	TAS
Any	not-achieve-both (eticket generated,credit card verified)	Org
Amadeus FS	availability(flight ticket booked, 85%)	TAS
Tourist	delegatedTo(trustworthy(Hotel))	Tourist

Deriving security requirements: an example


Non-modification, Non-production, Non-disclosure, Need-to-know

Step 5.1. Derive security requirements

Responsible	Security Requirement	Requester
TAS	need-to-know(personal data ∧ itinerary, tickets booked)	Tourist
TAS	non-modification(personal data ∧ itinerary	Tourist
TAS	non-production(personal data ∧ itinerary)	Tourist
TAS	non-disclosure(personal data ∧ itinerary)	Tourist

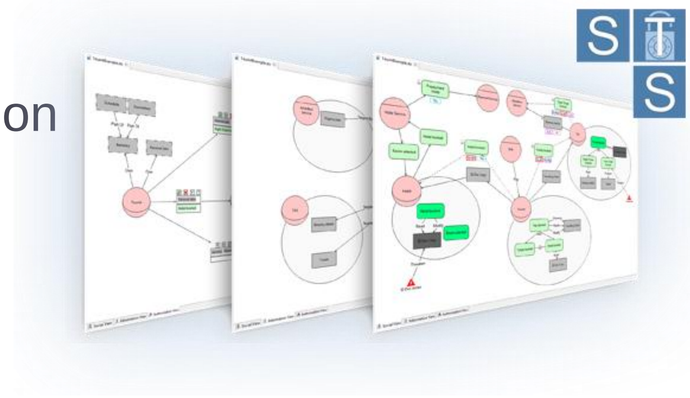
Tool Support: STS-Tool

- STS-Tool is the modeling and analysis support tool for STS-ml
  - Built on top of Eclipse
    - Standalone Eclipse RCP application

- Freely available for download: http://www.sts-tool.eu

- Derivation of security requirements
  - Automatic Requirements Document generation

- Multi-platform (Win, Linux, Mac)



The End

paja@disi.unitn.it

- **Thank you!**

November 2014