



# **Security Requirements Document**

## **Deliverable3\_Rashidfarokhi**

Mohammadali Rashidfarokhi

Feb 12, 2021

This document has been generated by STS-Tool  
<http://www.sts-tool.eu>



## Table of Contents:

Introduction .....	1
Social and organizational models .....	2
Social View .....	3
<i>Social View Diagram</i> .....	3
<i>Stakeholders</i> .....	4
<i>Stakeholders' documents</i> .....	7
<i>Stakeholders' documents and goals</i> .....	8
<i>Goal Refinement</i> .....	9
<i>Goal Contributions</i> .....	11
<i>Stakeholders Interactions</i> .....	11
Goal Delegations .....	11
Document Transmission .....	13
<i>Organisational Constraints</i> .....	13
<i>Events</i> .....	13
Information View .....	16
<i>Information View Diagram</i> .....	16
<i>Modelling Ownership</i> .....	17
<i>Representation of Information</i> .....	18
<i>Structure of Information and Documents</i> .....	18
Authorization View .....	19
<i>Authorization View Diagram</i> .....	19
<i>Authorization Flow</i> .....	20
Security Requirements .....	21
Well-formedness Analysis .....	29
Security Analysis .....	30
Appendix A .....	31
Appendix B .....	34
Appendix C .....	36



## Introduction

This document describes the security requirements for the "Deliverable3\_Rashidfarokhi"project. It provides a detailed description of: (I)social and organizational model, while capturing security requirements and automated analysis results;

## Social and organizational models

This section provides a detailed description of the socio-technical security requirements models from different views (*Social*, *Information*, *Authorization*) and then presents the list of *security requirements* derived from them.

The *Social view* represents stakeholders as intentional and social entities, representing their goals and important information in terms of documents, together with their interactions with other actors to achieve these goals and to exchange information. Stakeholders express constraints over their interactions in terms of *security needs*. The *Information view* represents the informational content of stakeholders' documents, showing how information and documents are interconnected, as well as how they are composed respectively. The *Authorization view* represents which stakeholders own what information, and captures the flow of permissions or prohibitions from one stakeholder to another. The modelling of authorizations expresses other *security needs* related to the way information is to be manipulated.

The section ends with the list of *security requirements* for the system to be expressed in terms of *social commitments*, namely promises with contractual validity stakeholders make to one another. The security requirements are derived automatically once the modelling is done and the designer has captured the security needs expressed by stakeholders. Whenever a security need is expressed over an interaction from one stakeholder to the other, a commitment on the opposite direction is expected from the second stakeholder to satisfy the security need.

## Social View

The social view shows the involved stakeholders, which are represented as *roles* and *agents*. Agents refer to actual participants (stakeholders) known when modelling the Deliverable3\_Rashidfarokhi project, whereas roles are a generalisation (abstraction) of agents. To capture the connection between roles and agents, the *play* relation is used to express the fact that certain agents play certain roles.

Stakeholders have goals to achieve and they make use of different information to achieve these goals. They interact with one another mainly by *delegating goals* and *exchanging information*. Information is represented by means of documents, which actors manipulate to achieve their goals.

## Social View Diagram

Figure 1 presents the graphical representation of the social view (a larger picture is shown in appendix A).

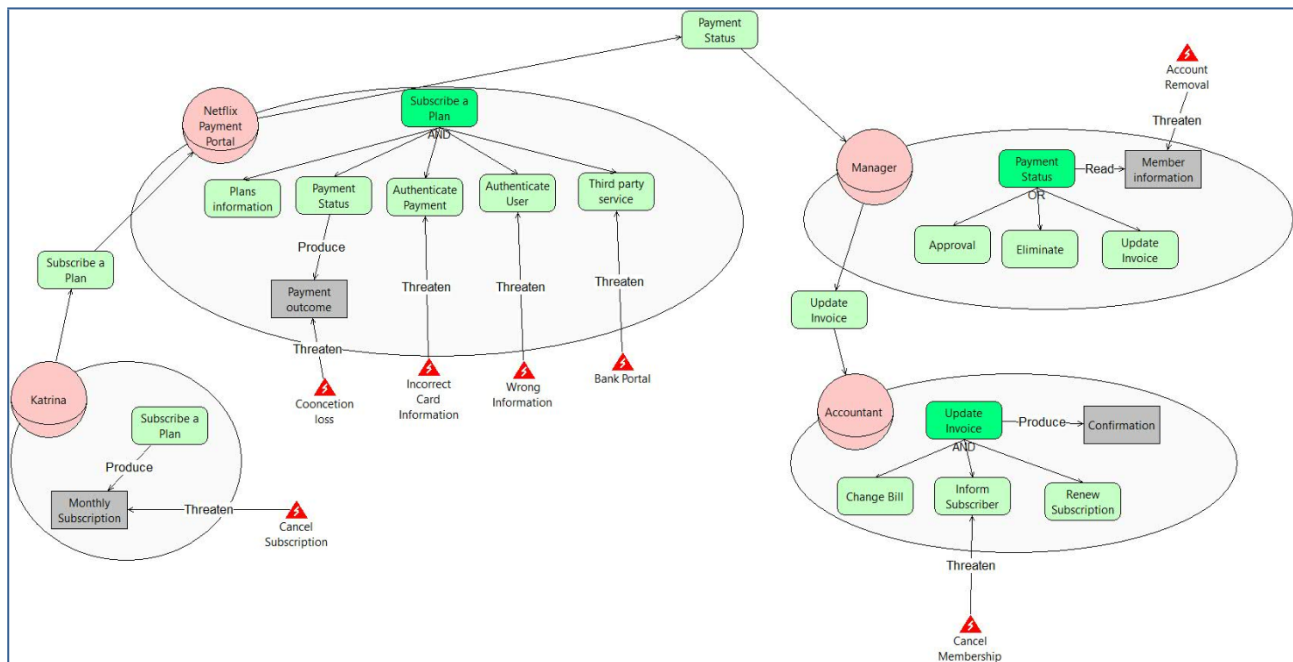


Figure 1 - Social View for the Deliverable3\_Rashidfarokhi project

## Stakeholders

This section describes the stakeholders identified in the Deliverable3\_Rashidfarokhi project. Stakeholders are represented as roles or agents.

In particular, identified roles are: *Katrina*, *Netflix Payment Portal*, *Manager* and *Accountant* (Figure 1). Table 1 summarise the stakeholders.

Role	Description	Mission	Purpose
Katrina	Katrina is planning to become Netflix's member. Therefore, she has to choose a plan for her subscription.		Subscribe to one of the Netflix plans.
Netflix Payment Portal	As soon as Katrina has chosen her ideal plan, she has to log in and complete the payment authentication step.		Allow the new member to complete the payment before providing Netflix services.
Manager	The manager can manage the member's account by approving, removing, and updating the details of the payment.		Review the member's subscription status.
Accountant	An accountant has the responsibility of analyzing the subscribes billing information. For example, update the billing based on the subscribed plan.		Investigate the financial information of the subscriber.

Table 1 - Roles in the Deliverable3\_Rashidfarokhi project.

In the Deliverable3\_Rashidfarokhi project there are no plays relationships taking place for the given agents/roles.

## Stakeholders' documents

Stakeholders have documents they possess or exchange with others to achieve their goals. Documents are represented within the rationale of the role/agent (Figure 1).

In the Deliverable3\_Rashidfarokhi project (Figure 1) we have:

- **Katrina** has document *Monthly Subscription*.
- **Netflix Payment Portal** has document *Payment outcome*.
- **Manager** has document *Member information*.
- **Accountant** has document *Confirmation*.

Table 2 summarises stakeholders' documents for the Deliverable3\_Rashidfarokhi project.

Agent/Role	Document	Description
------------	----------	-------------

Katrina	Monthly Subscription
Netflix Payment Portal	Payment outcome
Manager	Member information
Accountant	Confirmation

Table 2 - Stakeholders' documents in the Deliverable3\_Rashidfarokhi project

## Stakeholders' documents and goals

Stakeholders' documents are linked to their goals: they read (make) documents to achieve their goals, they modify documents while achieving their goals, and they may produce documents from achieving their goals.

In the Deliverable3\_Rashidfarokhi project (Figure 1) stakeholders' documents and goals are related as follows:

- **Katrina** produces document *Monthly Subscription* to achieve goal *Subscribe a Plan*.
- **Netflix Payment Portal** produces document *Payment outcome* to achieve goal *Payment Status*.
- **Manager** reads document *Member information* to achieve goal *Payment Status*.
- **Accountant** produces document *Confirmation* to achieve goal *Update Invoice*.

Table 3 summarises goal-document relations for all stakeholders in the Deliverable3\_Rashidfarokhi project.

Agent/Role	Goal	Document	Relation
Katrina	Subscribe a Plan	Monthly Subscription	Produce
Netflix Payment Portal	Payment Status	Payment outcome	Produce
Manager	Payment Status	Member information	Read
Accountant	Update Invoice	Confirmation	Produce

Table 3 - Relation of stakeholders' documents to their goals

## Goal Refinement

Stakeholders have goals to achieve. Goals are represented within the rationale (round compartment attached to the role/agent, see Figure 1) of the role/agent representing the stakeholder. They achieve their goals by further refining them into finer-grained goals (subgoals) by means of AND/OR-decompositions. AND-decompositions structurally refine a goal into multiple subgoals (all AND subgoals need to be achieved for the goal to be achieved), while OR-decompositions represent alternative ways for achieving a goal (at least one of the subgoals in the OR-decomposition needs to be achieved for the goal to be achieved).

In the Deliverable3\_Rashidfarokhi project (Figure 1) we have:

- **Katrina** has to achieve goal *Subscribe a Plan*.

- **Netflix Payment Portal** has to achieve goal *Subscribe a Plan*. To achieve *Subscribe a Plan*, Netflix Payment Portal should achieve goal *Plans information*, goal *Authenticate Payment*, goal *Authenticate User*, goal *Third party service provider* and goal *Payment Status*
- **Manager** has to achieve goal *Payment Status*. To achieve *Payment Status*, Manager should achieve either goal *Approval*, goal *Eliminate* or goal *Update Invoice*
- **Accountant** has to achieve goal *Update Invoice*. To achieve *Update Invoice*, Accountant should achieve goal *Change Bill*, goal *Inform Subscriber* and goal *Renew Subscription*

Table 4 summarises the goals of each agent/role in the Deliverable3\_Rashidfarokhi project and how they are decomposed, when applicable.

Agent/Role	Goal	Dec. Type	Subgoals
Katrina	Subscribe a Plan	-	
Netflix Payment Portal	Subscribe a Plan	AND	Plans information
			Authenticate Payment
			Authenticate User
			Third party service provider
			Payment Status
Manager	Payment Status	OR	Approval
			Eliminate
			Update Invoice
Accountant	Update Invoice	AND	Change Bill
			Inform Subscriber
			Renew Subscription

Table 4 - Goal Decompositions

## Goal Contributions

Goals can contribute one to another. A contribution identifies the impact the fulfilment of one goal has on the fulfilment of another goal. This impact can be either positive or negative, and is represented with "++" and "--" respectively. Positive contribution means that the achievement of a goal also achieves the other goal. Negative contribution means that the achievement of a goal inhibits the achievement of another goal.

In the Deliverable3\_Rashidfarokhi project there are no contribution relations taking place for the given agents/roles.

## Stakeholders Interactions

This section describes stakeholders' interactions, providing insights on whom they interact with to fulfil their desired objectives, as well as which are the stakeholders that rely on them to fulfil their respective goals. This kind of interaction is carried out by means of *goal delegations*.



To achieve their goals stakeholders might need specific information. If they do not possess this information, they may ask other stakeholders to provide them documents. *Document transmission* is used to capture this interaction.

### Goal Delegations

Stakeholders interact with others to achieve some of their goals by means of goal delegations. Goal delegations are graphically represented as a relation that starts from a delegator actor to a delegatee actor (following the direction of the arrow), having a rounded corner rectangle representing the goal being delegated. Security needs are graphically specified as labels that appear below the delegated goal (Figure 1).

The following description enlists all the delegations from one role/agent to the others. When applicable, security needs expressed over the delegations are enumerated.

In the Deliverable3\_Rashidfarokhi project (Figure 1), we have the following goal delegations:

- **Katrina** delegates goal *Subscribe a Plan* to **Netflix Payment Portal**.
- **Netflix Payment Portal** delegates goal *Payment Status* to **Manager**.
- **Manager** delegates goal *Update Invoice* to **Accountant**.

Table 5 summarises *goal delegations*, together with the eventual *security needs* when applicable, and eventual description respectively.

Delegator	Goal	Delegatee	Security Needs	Delegation Description
Katrina	Subscribe a Plan	Netflix Payment Portal		
Netflix Payment Portal	Payment Status	Manager		
Manager	Update Invoice	Accountant		

Table 5 - Goal Delegations and Security Needs

### Document Transmission

Stakeholders exchange information by means of documents with other stakeholders. The following description enlists all the transmission from one role/agent representing the stakeholder, to other roles/agents. *Document transmission* is represented as an arrow from the transmitter to the receiver, with a rectangle representing the document. The security needs expressed over the transmission are described, if applicable. Security needs are specified with the help of labels that appear below the document being transmitted.

In the Deliverable3\_Rashidfarokhi project there are no document provisions taking place for the given agents/roles.

### Organisational Constraints

Apart from the security needs actors specify over their interactions, there are others, which are dictated either by the organisation, business rules and regulations, or law. In this section we enlist these constraints, together with the security requirements derived from them. Currently, the language supports these organisational constraints: *Separation of Duties (SoD)* and *Binding of Duties (BoD)*.

Graphically we represent these constraints using a similar notation to that used in workflows, as a circle with the *unequal* sign within and as a circle with the *equals* sign within, respectively. The relations are symmetric, and as such they do not have any arrows pointed to the concepts they relate (being these roles or goals).

In the Deliverable3\_Rashidfarokhi project there are no organisational constraints specified.

## Events

Table 6 represents all the events modeled in the project Deliverable3\_Rashidfarokhi together with the set of elements each event threatens. Additionally, for each reported event a textual description is provided.

Event name	Threatened elements	Description
Bank Portal	Goal: Third party service provider	The bank portal will not respond during payment authentication.
Account Removal	Document: Member information	If the member delete the Netflix account, after 10 month all the profile information will be deleted.
Incorrect Card Information	Goal: Authenticate Payment	One of the information of the credit card has been entered wrong.
Cancel Subscription	Document: Monthly Subscription	
Cancel Membership	Goal: Inform Subscriber	
Cooncetion loss	Document: Payment outcome	
Wrong Information	Goal: Authenticate User	Either email or password has been entered wrong.

Table 6 - Events

## Information View

The information view gives a structured representation of the information and documents in the Deliverable3\_Rashidfarokhi project. It shows what is the informational content of the documents represented in the social view. Information is represented by one or more documents (*tangible by*), and the same document can make tangible multiple information entities. Moreover, the information view considers composite documents (information) capturing these by means of *part of* relations.

### Information View Diagram

Figure 2 presents the graphical representation of the information view (a larger picture is shown in appendix A).

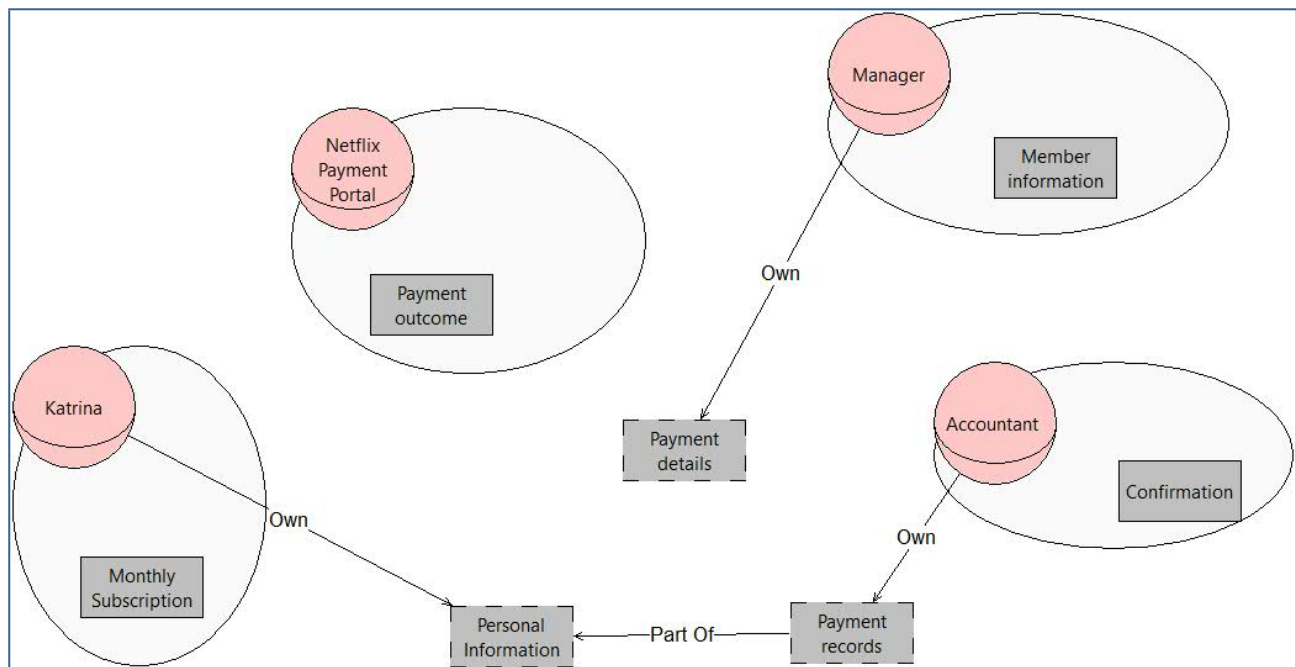


Figure 2 - Information View for the Deliverable3\_Rashidfarokhi project

### Modelling Ownership

The information view represents also who are the *owners* of the information that is being manipulated through the documents that represent them in the social view.

The owners for the different information in the Deliverable3\_Rashidfarokhi project are summarised in Table 7.

Agent/Role	Information	Description
Katrina	Personal Information	The member has some personal information.
Manager	Payment details	The manager has access to the member banking information such as credit card number.
Accountant	Payment records	It includes all the transactions that happened by the member.

Table 7 - Information owners

### Representation of Information

Information is represented (*made tangible by*) by documents, which stakeholders have and exchange.

In the Deliverable3\_Rashidfarokhi project there are no "Tangible By" relations specified for the documents of the given agents/roles.

### Structure of Information and Documents

Documents (information) are composed of other documents (information). Composition of documents (information) is captured through *part of* relations. This gives us an idea of how information and/or documents in the Deliverable3\_Rashidfarokhi project are structured.

Table 8 summarises the information in the Deliverable3\_Rashidfarokhi project (Figure 2), showing how they are composed and describing the composition.

Information	Composition	Description
Personal Information	Payment records	

Table 8 - Information composition

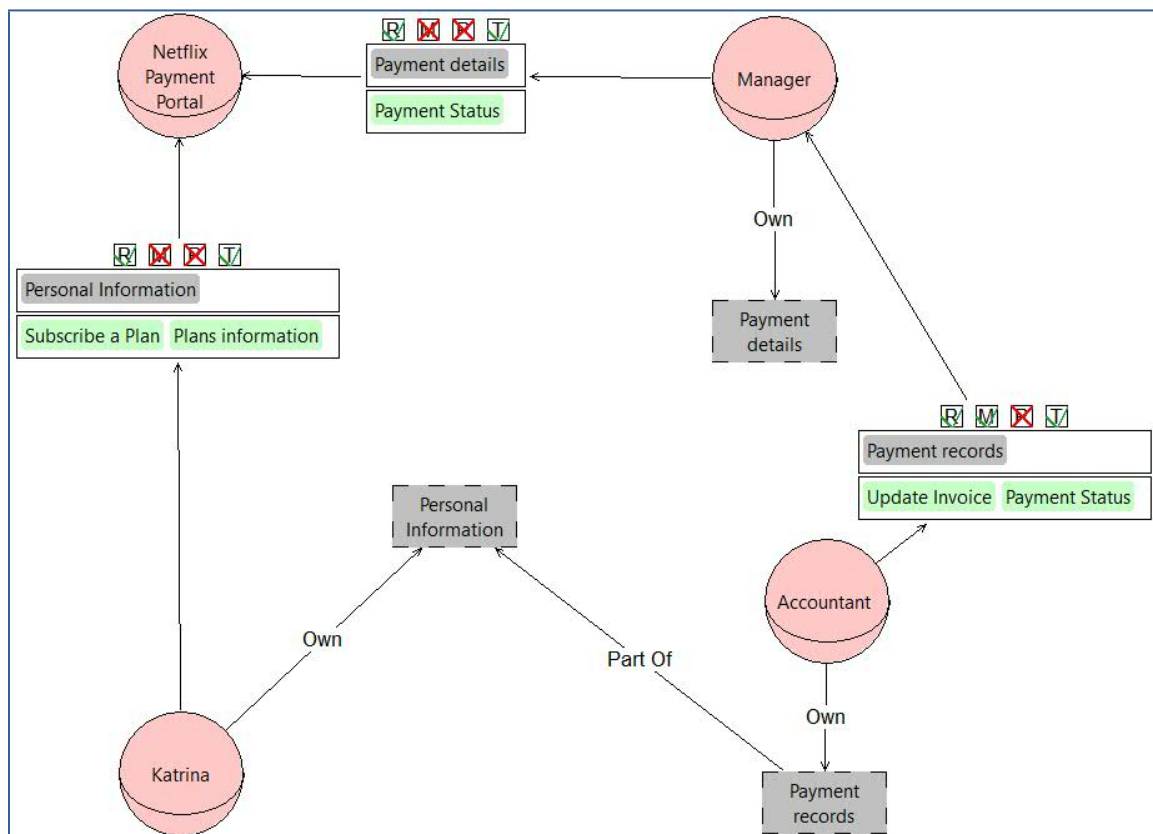
## Authorization View

The authorization view shows the permissions or prohibitions flow from a stakeholder to another, that is, the authorizations stakeholders grant or deny to others about information, specifying the operations the others can and must perform over the information. Apart from granting authority on performing operations, a higher authority can be granted, that of further authorising other actors (i.e. authorization transferability)

Authorizations start from the information owner. Therefore, in the authorization view, ownership is preserved and inherited from the information view.

### Authorization View Diagram

Figure 3 presents the graphical representation of the Authorization view (a larger picture is represented in appendix A).



## Authorization Flow

In this section are described for each role/agent, the authorizations it passes to others and what authorizations it receives from other roles/agents. In the Deliverable3\_Rashidfarokhi project (Figure 3) the authorizations for each role/agent are:

- **Role Katrina:**
  - **Katrina** authorises *Netflix Payment Portal* to *read* and *transmit* and prohibits to *modify* and *produce* information *Personal Information*, in the scope of goals *Subscribe a Plan* and *Plans information*, passing the right to further authorising other actors.
- **Role Netflix Payment Portal:**
  - **Netflix Payment Portal** s.
  - **Netflix Payment Portal** is authorised by *Netflix Payment Portal* to *read* and *transmit* and prohibited to *modify* and *produce* information *Personal Information*, in the scope of goal *Subscribe a Plan* and *Plans information*, having the right to further authorising other actors, and is authorised by *Netflix Payment Portal* to *read* and *transmit* and prohibited to *modify* and *produce* information *Payment details*, in the scope of goal *Payment Status*, having the right to further authorising other actors.
- **Role Manager:**
  - **Manager** authorises *Netflix Payment Portal* to *read* and *transmit* and prohibits to *modify* and *produce* information *Payment details*, in the scope of goal *Payment Status*, passing the right to further authorising other actors.
  - **Manager** is authorised by *Manager* to *read*, *modify* and *transmit* and prohibited to *produce* information *Payment records*, in the scope of goal *Update Invoice* and *Payment Status*, having the right to further authorising other actors.
- **Role Accountant :**
  - **Accountant** authorises *Manager* to *read*, *modify* and *transmit* and prohibits to *produce* information *Payment records*, in the scope of goals *Update Invoice* and *Payment Status*, passing the right to further authorising other actors.

## Security Requirements

This section provides the list of security requirements derived for the Deliverable3\_Rashidfarokhi project.

The list of security requirements shows the roles/agents that are *responsible* to satisfy them, so that stakeholders know what they have to bring about in order to satisfy the corresponding security needs. Security requirements also include the authorizations granted by stakeholders to other stakeholders.

*Security needs* are expressed mainly over goal delegations, document provisions and authorizations. Therefore, the list of security requirements is derived from every type of security need. Moreover, the organisational constraints specify further *needs* over roles and goal, leading to the generation of other security requirements.

Finally, the *requester* actors are represented to capture the actors requiring certain security needs to be brought about.

The security requirements for the Deliverable3\_Rashidfarokhi project (Table 9) are:

- **Katrina** requires *Netflix Payment Portal* the *non-modification* and *non-production* of information *Personal Information*, and *need-to-know* of these pieces of information for the goals *Subscribe a Plan* and *Plans information*, when authorising *Netflix Payment Portal* to *read* and *distribute* *Personal Information* in the scope of goals *Subscribe a Plan* and *Plans information*.
- **Manager** requires *Netflix Payment Portal* the *non-modification* and *non-production* of information *Payment details*, and *need-to-know* of these pieces of information for the goal *Payment Status*, when authorising *Netflix Payment Portal* to *read* and *distribute* *Payment details* in the scope of goal *Payment Status*.
- **Accountant** requires *Manager* the *non-production* of information *Payment records*, and *need-to-know* of these pieces of information for the goals *Update Invoice* and *Payment Status*, when authorising *Manager* to *read*, *modify* and *distribute* *Payment records* in the scope of goals *Update Invoice* and *Payment Status*.

Responsible	Security Requirement	Requester	Description
Netflix Payment Portal	non-modification (Personal Information)	Katrina	Katrina requires Netflix Payment Portal non-modification of Information Personal Information.
	non-production (Personal Information)	Katrina	Katrina requires Netflix Payment Portal non-production of Information Personal Information.
	need-to-know (Personal Information) (Subscribe a Plan,Plans information)	Katrina	Katrina requires Netflix Payment Portal need-to-know of Information Personal Information, in the scope of goal Subscribe a Plan and Plans information.
	non-modification (Payment details)	Manager	Manager requires Netflix Payment Portal non-modification of Information

Manager	non-production (Payment details)	Manager	Payment details. Manager requires Netflix Payment Portal non-production of Information Payment details.
			Manager requires Netflix Payment Portal need-to-know of Information Payment details, in the scope of goal Payment Status.
	need-to-know (Payment details) (Payment Status)	Manager	Accountant requires Manager non-production of Information Payment records.
			Accountant requires Manager need-to-know of Information Payment records, in the scope of goal Update Invoice and Payment Status.

*Table 9 - Security Requirements for the Deliverable3\_Rashidfarokhi Project*

Table 10 summarises the authorizations actors in the Deliverable3\_Rashidfarokhi project grant to one another.

Authorisor Information		Goal	Allowed Operations	Denied Operations	Authorisee	Description
Katrina	Personal Information	Subscribe a Plan Plans information	R, T	M, P	Netflix Payment Portal	Transferable authority
Manager	Payment details	Payment Status	R, T	M, P	Netflix Payment Portal	Transferable authority
Accountant	Payment records	Update Invoice Payment Status	R, M, T	P	Manager	Transferable authority

*Table 10 - Authorizations in the Deliverable3\_Rashidfarokhi project*



### *Well-formedness Analysis*

The purpose of well-formedness analysis is to verify whether the diagram for the project Deliverable3\_Rashidfarokhi is consistent and valid. A diagram is considered to be consistent if its constituent elements (concepts and relationships) are drawn and interconnected following the semantics of the modelling language (STS-ml in our case). Thus, well-formedness analysis performs post checks to verify compliance with STS-ml semantics for all checks that cannot be performed live over the models.

More details about the performed checks and their purpose can be found in Appendix B.

*The Well-formedness Analysis analysis for Deliverable3\_Rashidfarokhi project didn't find any errors.*

## *Security Analysis*

The purpose of security analysis is to verify whether the diagram for the project Deliverable3\_Rashidfarokhi allows the satisfaction of the specified security needs or not. As a result, for all security needs expressed by stakeholders, it checks in the model whether there is any possibility for the security need to be violated. This analysis takes into account the semantics of STS-ml, defining the behaviour of the different elements represented in the models. The elements' behaviour is defined by propagation rules that consider what concepts and what relationships the specification of a given security need affects. Datalog is used to define the semantics of STS-ml to express facts (things always hold) and rules.

You can find more details about the performed checks in Appendix C.

*The Security Analysis analysis for Deliverable3\_Rashidfarokhi project didn't find any errors.*

## Appendix A

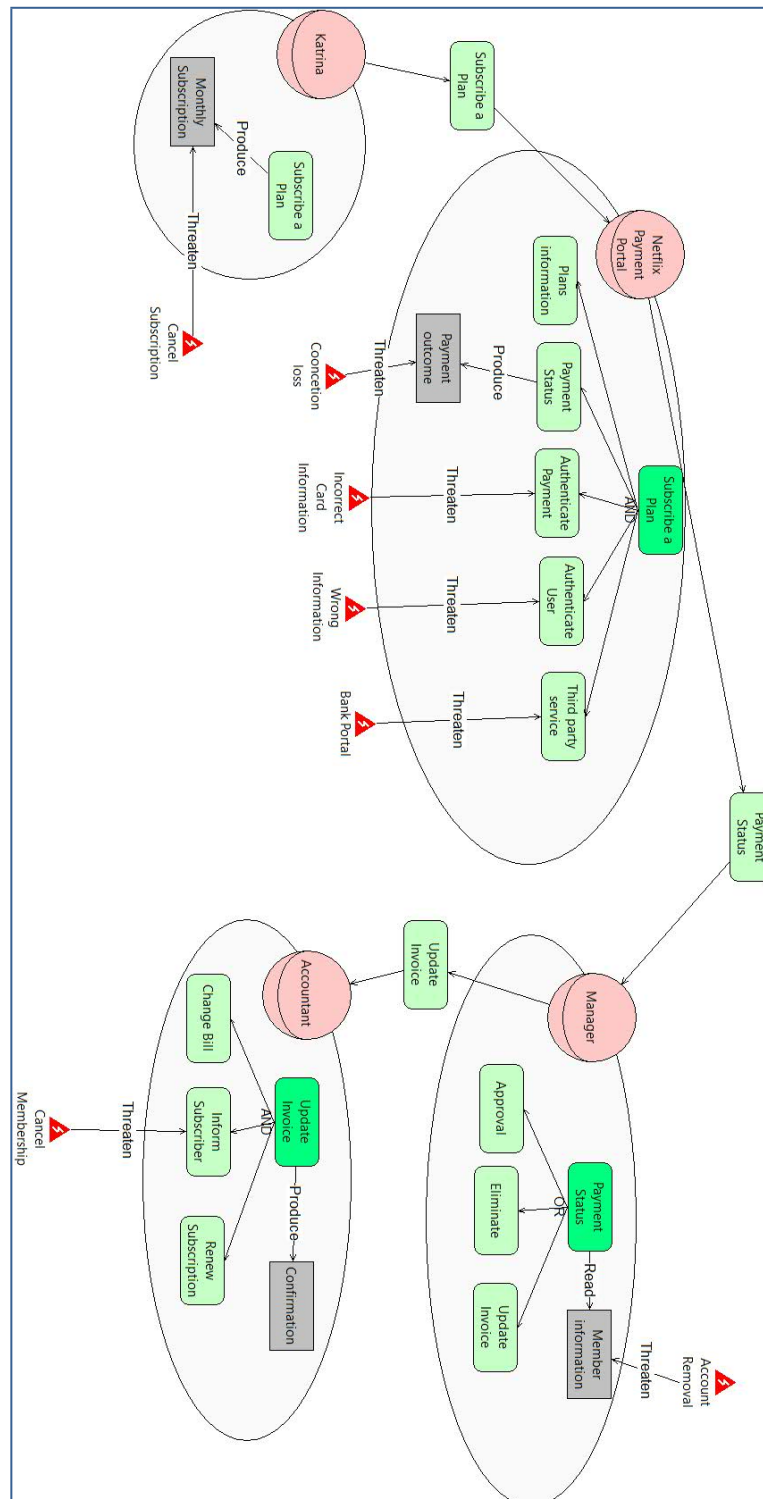
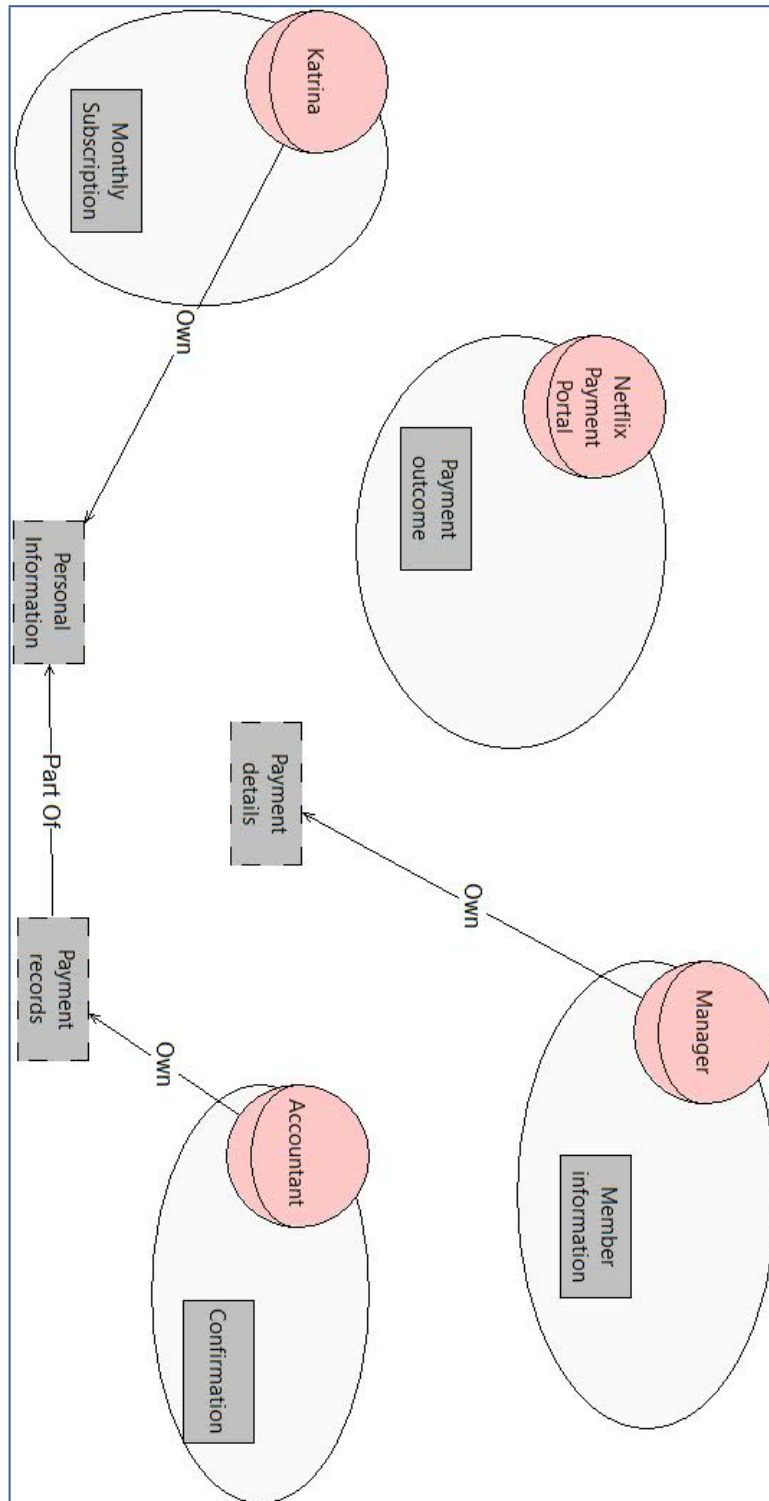


Figure 1 - Social View for the Deliverable3\_Rashidfarokhi project



*Figure 2 - Information View for the Deliverable3\_Rashidfarokhi project*

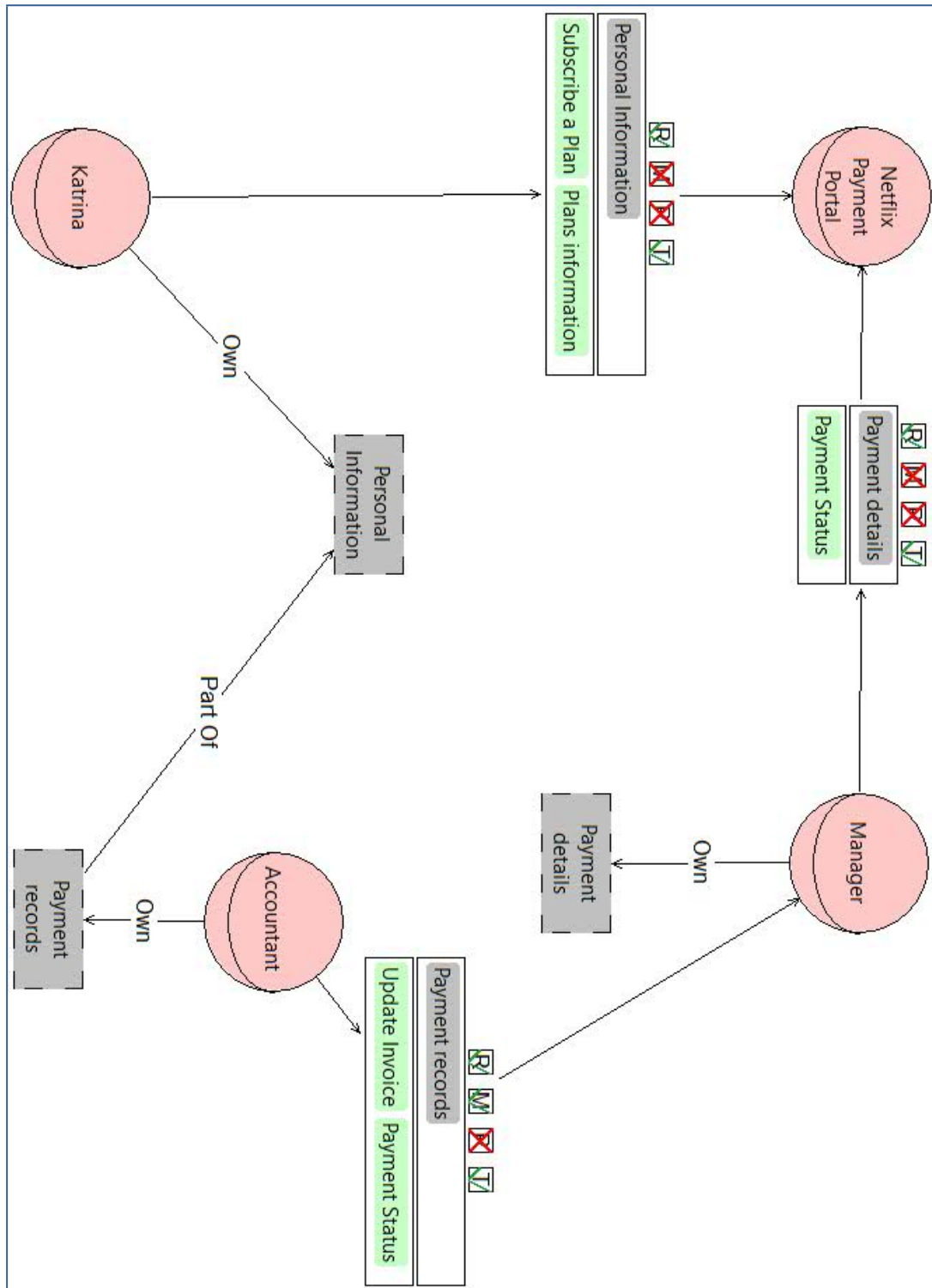


Figure 3 - Authorization View for the Deliverable3\_Rashidfarokhi project

## Appendix B

Details of Well-formedness analysis:

- **Empty Diagram**

This check verifies whether the given diagram is empty or not. If that is the case, then no other well-formedness checks are performed. If the diagram is not empty, the well-formedness analysis returns: "No errors found" and continues performing the rest of the well-formedness checks.

- **Goal Single Decomposition**

This check verifies the consistency of goal decompositions. Following the semantics of STS-ml a given goal is decomposed in two or more subgoals. As a result, the decomposition should specify at least two subgoals. Therefore, goal single decomposition verifies whether there are cases of decompositions to a single subgoal.

- **Delegation Child Cycle**

This check verifies the consistency of goal delegations, so that no cycles or loops are identified as a result of the delegatee decomposing the delegatum (delegated goal) and re-delegating back one of the subgoals. Delegation child cycle verifies exactly this and gives a warning in case of inconsistency.

- **Delegated Goal Part Of a Decomposition**

This check verifies that all goals (in the delegatee's scope) that have been delegated are not child (subgoals) in the decomposition.

- **Inconsistent Contribution Cycle**

This check verifies whether there are loops of positive or negative contribution relationships, and whether this loop contains contradictory relationships. If such a loop is identified, the well-formedness analysis returns a warning.

- **Negative Contributions Between AND Subgoals**

This check verifies that there are no negative contribution relationships between and-subgoals of a given goal (within an actor's scope). It returns a warning if such a case is identified.

- **Documents PartOf Cycle**

This check verifies whether there is a loop or cycle of Part Of relationships starting from and ending to a given document. If a case like this is verified, a warning is returned enumerating the documents that form the cycle.

- **Informations PartOf Cycle**

This check verifies whether there is a loop or cycle of Part Of relationships starting from and ending to a given document. If a case like this is verified, a warning is returned enumerating the documents that form the cycle.

- **Information No Ownership**

This check verifies that all information have an owner. If there are cases of information without any ownership relationships from any actor in the diagram, the well-formedness analysis returns a warning.

- **Authorizations Validity**

This check verifies that all authorization relationship between two given actors are valid. An authorization relationship specifies authorizations or permissions an actor grants to another on some information, to perform some allowed operations. The authorizations could be limited to a goal scope and they can be re-delegated or not. However, the first two attributes should be specified for an authorization relationship to be valid. If there are no information specified, the well-formedness analysis returns an error. The same applies to the cases, in which no allowed operations are specified.

- **Duplicate Authorizations**

This check verifies that there are no duplicate authorization relationships, that could be merged. There are several cases that are addressed by this check: (i) we encounter two identical authorization, i.e., between the same roles, in the same direction, for the same set of information, allowed operations and goals, and having the same value of transferability; (ii) identify authorization relationships between the same roles, in the same direction, in which one grants permissions that are subset of the other authorization's relationship.

## Appendix C

Details of security analysis:

- **No\_Delegation Violation check**

This violation is verified whenever a delegatee actor further delegates a goal, over the delegation of which a no-delegation security need is specified from the delegator actor. No-delegation is specified over a goal delegation by the delegator, who requires the delegatee not to further delegate the delegated goal. Therefore, to check for any violations of no-delegation, the analysis searches for redelegations of the delegatum (delegated goal) or any of its subgoals.

- **Redundancy Violation check**

This check verifies if redundancy is satisfied by controlling that single actor redundancy or multi actor redundancy are not violated. At design time we cannot make the distinction between fallback and true redundancy, so they cannot be verified at this stage. Therefore, both fallback redundancy single and true redundancy single are mapped to single actor redundancy. Similarly for multi actor redundancy. The analysis verifies a redundancy violation if one of the following occurs: (1) actor does not decompose the delegated goal in any or-subgoals, for which both types of redundancy are violated (2) actor decomposes the goal into or-subgoals and delegates one to another actor when single actor redundancy has been specified, for which this type of redundancy is violated (3) actor decomposes the goal into or-subgoals, but does not delegate any of the subgoals to another actor when multi actor redundancy has been specified, for which this type of redundancy is violated.

- **Authorization Conflict check**

This task identifies a conflict of authorization whenever at least two authorization relationships for the same information are drawn towards the same actor from two illegible actors (being the owner of information or another authorised actor) such that: (1) one limits the authorization to a goal scope (requiring a need-to-know security need) and the other does not (authorising the actor without any limitations) (2) for the same goals or intersecting goal scopes, different permissions are granted in terms of operations or authority to transfer authorisation. That is, one passes the actor the authority to perform operations (use, modify, produce, distribute) on a given information, and the other does not (requiring non-usage, non-modification, non-production, non-disclosure); one passes the actor the authority to further transfer authorizations and the other requires no further authorizations take place.

- **Non\_Reading Violation**

This violation is detected whenever an actor discloses information without having the right to distribute it. Non-disclosure expresses the need of not disclosing or further distributing the given information to other actors, apart from the authoriser. Thus, authority to distribute the information is not passed. The way actors exchange information is through document provision. In order to disclose some information, an actor would have to provide to others the document(s) containing that information. Hence, to verify if there are any unauthorized disclosures of information, the analysis checks for provisions of documents representing the given information from any unauthorized actors towards other actors.

- **Non\_Modification Violation**



This violation is detected whenever an actor modifies information without having the right to modify it. Non-modification expresses the need that information should not be changed (modified), i.e. authority to modify the information is not granted. To verify if there could be any violations of non-modification, the analysis looks if the authorisee (or an actor that is not authorised by authorised party) modifies the given information. For this, it searches for modify relationships from any goal of this actor to any document representing the given information.

- **Non\_Production Violation**

This violation is detected whenever an actor produces information without having the right to produce it. Non-production expresses the need that information should not be produced in any form, i.e. authority to produce the information is not granted. To verify if there could be any violations of non-production, the analysis checks whether if the authorisee (or an actor that is not authorised by authorised party) produces the given information. For this, it searches for produce relationships from any goal of this actor to any document representing the given information.

- **Non\_Disclosure Violation**

This violation is detected whenever an actor discloses information without having the right to distribute it. Non-disclosure expresses the need of not disclosing or further distributing the given information to other actors, apart from the authoriser. Thus, authority to distribute the information is not passed. The way actors exchange information is through document provision. In order to disclose some information, an actor would have to provide to others the document(s) containing that information. Hence, to verify if there are any unauthorized disclosures of information, the analysis checks for provisions of documents representing the given information from any unauthorized actors towards other actors.

- **NTK Violation**

This violation is detected whenever an actor uses, modifies or produces information for other purposes (goal achievement) than the ones for which it is authorized. Need-to-know requires that the information is used, modified, or produced in the scope of the goals specified in the authorization. This security need concerns confidential information, which should not be utilised for any other purposes other than the intended ones. To verify if there could be any violations of need-to-know, security analysis checks if the authorisee (or an actor that is not authorised by any authorised party) uses, modifies or produces the given information while achieving some goal different from the one it is authorised for. In a nutshell, it searches for need, modify, or produce relationships starting from goals different from the specified ones towards documents representing the given information.

- **Explicit non-reauthorization**

Verifies whether a given actor transfer rights to others even when it does not have the authority to further delegate rights.

- **Non-reauthorization Violation: read**

Verifies whether a given actors transfer to other actors the right to use a given information, without having itself the right to do so.

- **Non-reauthorization Violation: modify**

Verifies whether a given actors transfer to other actors the right to modify a given information, without having itself the right to do so.

- **Non-reauthorization Violation: produce**

Verifies whether a given actors transfer to other actors the right to modify a given information, without having itself the right to do so.

- **Non-reauthorization Violation: transmit**

Verifies whether a given actors transfer to other actors the right to distribute a given information, without having itself the right to do so.

- **Sod Goal Violation**

This violation is detected whenever a single actor may perform both goals, between which an SoD constraint is expressed. Goal-based SoD requires that there is no actor performing both goals among which SoD is specified. To perform this verification, the analysis checks that the final performer of the given goals is not the same actor.

- **Bod Goal Violation**

This violation is detected whenever a single actor may perform both goals, between which an SoD constraint is expressed. Goal-based SoD requires that there is no actor performing both goals among which SoD is specified. To perform this verification, the analysis checks that the final performer of the given goals is not the same actor.

- **Agent Play Sod**

This check verifies the consistency of the Separation of Duty (SoD) constraint between roles. This constraint requires that two roles are not played by the same agent, therefore the check verifies whether there is one agent playing both roles. If that is the case an error is identified, otherwise the check finds no errors.

- **Agent Not Play Bod**

This check verifies the consistency of the Binding of Duty (BoD) constraint between roles. This constraint requires that two roles are played by the same agent, therefore the check verifies whether there is one agent playing both roles. If that is the case the check finds no errors, otherwise an error is identified.

- **Organizational Constraint Consistency**

This check verifies that no conflicting organisational constraints (SoD or BoD) between goals are specified.