Security Requirements Document

Security Requirements

Xingrong Zong

Mar 7, 2021

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

Table of Contents:

Introduction

This document describes the security requirements for the "Security Requirements"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 Security Requirements 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).

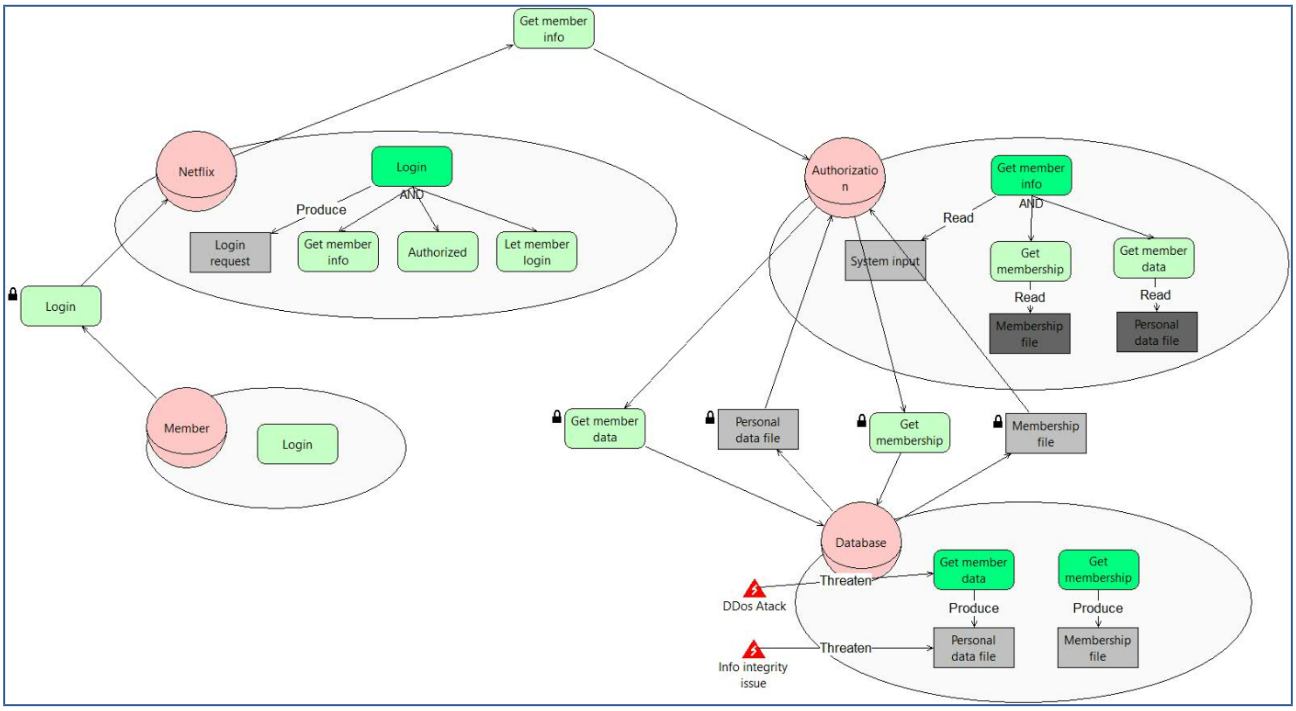

Figure 1 - Social View for the Security Requirements project

Stakeholders

This section describes the stakeholders identified in the Security Requirements project. Stakeholders are represented as roles or agents.

In particular, identified roles are: *Member*, *Netflix*, *Authorization* and *Database* (Figure 1). Table 1 summarise the stakeholders.

Role	Description	Mission	Purpose
Member			
Netflix			
Authorization			
Database			

Table 1 - Roles in the Security Requirements project.

In the Security Requirements 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 Security Requirements project (Figure 1) we have:

- **Netflix** has document *Login request*.
- **Authorization** has document *System input*. Moreover it has document *Personal data file* provided by *Database* and document *Membership file* provided by *Database*.
- **Database** has documents *Personal data file* and *Membership file*.

Table 2 summarises stakeholders' *documents* for the Security Requirements project.

Agent/Role	Document	Description
Netflix	Login request	
Authorization	System input	
	Personal data file	
	Membership file	
Database	Personal data file	
	Membership file	

Table 2 - Stakeholders' documents in the Security Requirements 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 Security Requirements project (Figure 1) stakeholders' documents and goals are related as follows:

- **Netflix** *produces* document *Login request* to achieve goal *Login*.
- **Authorization** *reads* document *Membership file* to achieve goal *Get membership status*, *reads* document *Personal data file* to achieve goal *Get member data* and *reads* document *System input* to achieve goal *Get member info*.
- **Database** *produces* document *Membership file* to achieve goal *Get membership status* and *produces* document *Personal data file* to achieve goal *Get member data*.

Table 3 summarises goal-document relations for all stakeholders in the Security Requirements project.

Agent/Role	Goal	Document	Relation
Netflix	Login	Login request	Produce
Authorization	Get membership status	Membership file	Read
	Get member data	Personal data file	Read
	Get member info	System input	Read
Database	Get membership status	Membership file	Produce
	Get member data	Personal data file	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 Security Requirements project (Figure 1) we have:

- **Member** has to achieve goal *Login*.
- **Netflix** has to achieve goal *Login*. To achieve *Login*, Netflix should achieve goal *Get member info*, goal *Authorized* and goal *Let member login*
- **Authorization** has to achieve goal *Get member info*. To achieve *Get member info*, Authorization should achieve goal *Get membership status* and goal *Get member data*
- **Database** has to achieve goal *Get member data* and goal *Get membership status*.

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

Agent/Role	Goal	Dec. Type	Subgoals
Member	Login	-	

Netflix	Login	AND	Get member info
			Authorized
			Let member login
Authorization	Get member info	AND	Get membership status
			Get member data
Database	Get member data	-	
	Get membership status	-	

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 Security Requirements 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 Security Requirements project (Figure 1), we have the following goal delegations:

- **Member** delegates goal *Login* to **Netflix**.

  The following security needs apply to this delegation:

  Non Repudiation: acceptance.

- **Netflix** delegates goal *Get member info* to **Authorization**.
- **Authorization** delegates goal *Get member data* to **Database**.

The following security needs apply to this delegation:

Trustworthiness and Availability: 100.0.

- **Authorization** delegates goal *Get membership status* to **Database**.

  The following security needs apply to this delegation:

  Trustworthiness and Availability: 100.0.

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

Delegator	Goal	Delegatee	Security Needs	Delegation Description
Member	Login	Netflix	**Non Repudiation**: *acceptance*	
Netflix	Get member info	Authorization		
Authorization	Get member data	Database	**Trustworthiness Availability**: *100.0*	
	Get membership status	Database	**Trustworthiness Availability**: *100.0*	

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 Security Requirements project (Figure 1), we have the following *document transmissions*:

- **Database** transmit document *Personal data file* to **Authorization**.

  The following security needs apply to this transmission:

  Confidentiality: receiver.

- **Database** transmit document *Membership file* to **Authorization**.

  The following security needs apply to this transmission:

  Confidentiality: receiver.

Table 6 summarises the *document transmissions* for the Security Requirements project.

Transmitter	Document	Receiver	Security Needs	Transmission Descr.
Database	Personal data file	Authorization	**Confidentiality**: *receiver*	
	Membership file	Authorization	**Confidentiality**: *receiver*	

Table 6 - Document Transmissions and Security Needs

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 Security Requirements project there are no organisational constraints specified.

Events

Table 7 represents all the events modeled in the project Security Requirements together with the set of elements each event threatens. Additionally, for each reported event a textual description is provided.

Event name	Threatened elements	Description
Info integrity issue	Document: Personal data file	
DDos Atack	GoalReference: Get member data	

Table 7 - Events

Information View

The information view gives a structured representation of the information and documents in the Security Requirements 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.

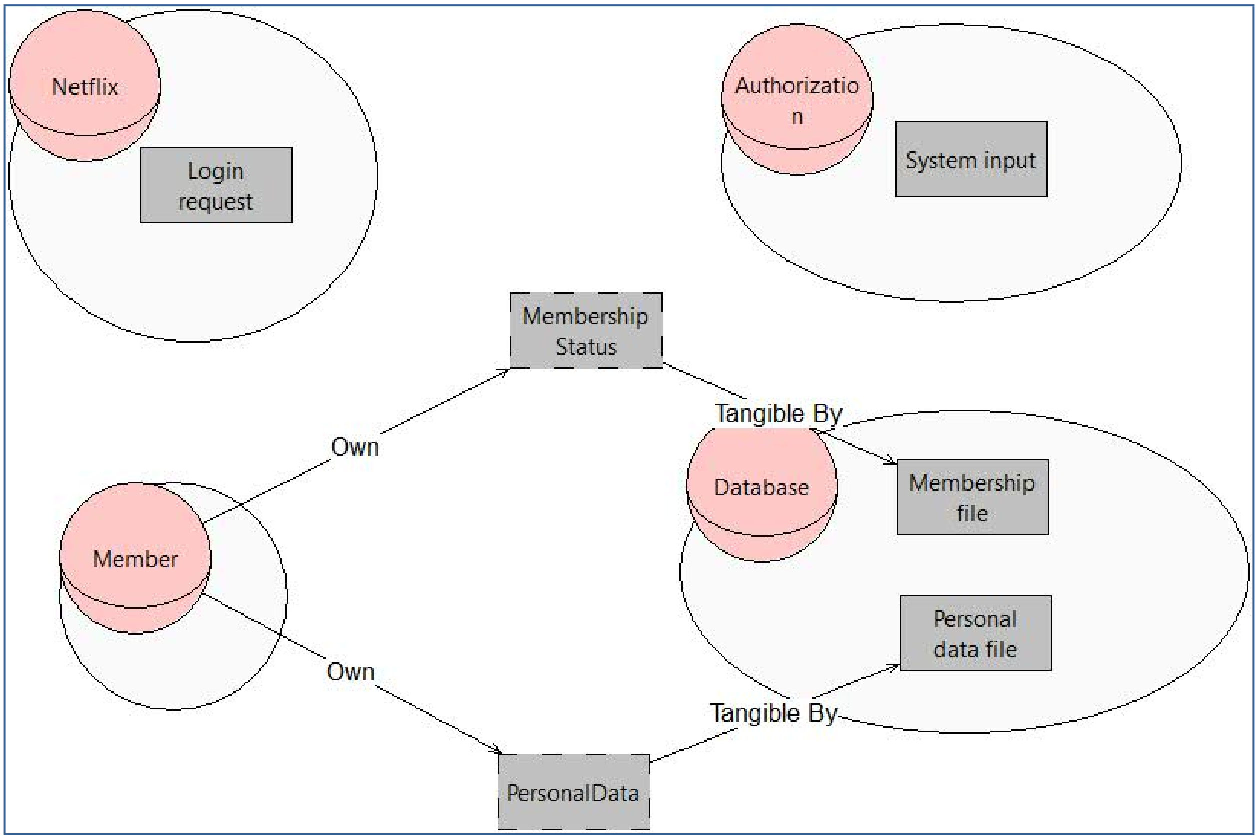

Figure 2 - Information View for the Security Requirements project

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 Security Requirements project are summarised in Table 8.

Agent/Role	Information	Description
Member	PersonalData	
	MembershipStatus	

Table 8 - Information owners

Representation of Information

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

The documents stakeholders in the Security Requirements project (Figure 2) have and exchange with one another contain the information as summarised in Table 9:

Information	Document	Description
MembershipStatus	Membership file	
PersonalData	Personal data file	

Table 9 - Representation of Information through Documents

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 Security Requirements project are structured.

In the Security Requirements project there are no composite documents or information.

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.

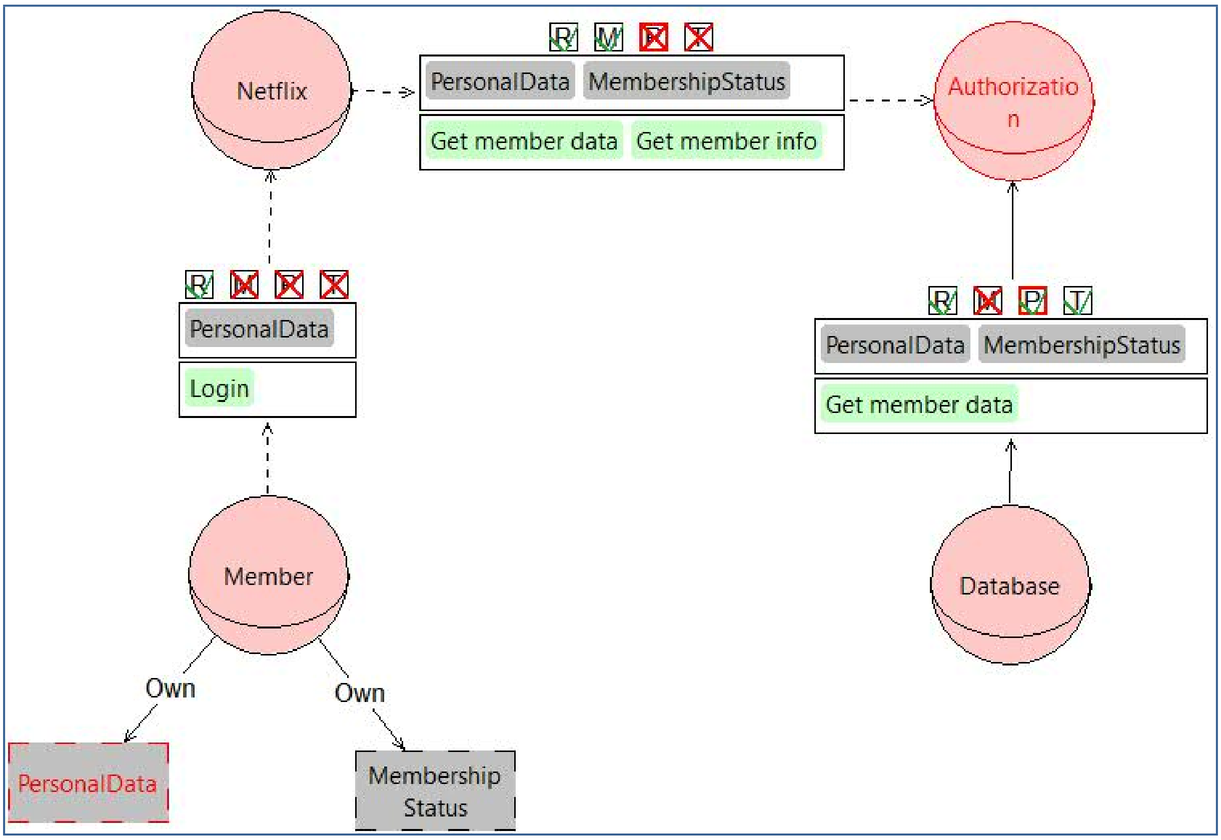

Figure 3 - Authorization View for the Security Requirements project

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 Security Requirements project (Figure 3) the authorizations for each role/agent are:

- *Role* **Member**:
    - **Member** authorises *Netflix* to *read* and prohibits to *modify*, *produce* and *transmit* information *PersonalData*, in the scope of goal *Login*, *passing* the right to further authorising other actors.
- *Role* **Netflix**:
    - **Netflix** authorises *Authorization* to *read* and *modify* and prohibits to *produce* and *transmit* information *PersonalData* and *MembershipStatus*, in the scope of goals *Get member data* and *Get member info*, *passing* the right to further authorising other actors.
    - **Netflix** is authorised by *Netflix* to *read* and prohibited to *modify*, *produce* and *transmit* information *PersonalData*, in the scope of goal *Login*, *having* the right to further authorising other actors.
- *Role* **Authorization**:
    - **Authorization** s.
    - **Authorization** is authorised by *Authorization* to *read* and *modify* and prohibited to *produce* and *transmit* information *PersonalData* and *MembershipStatus*, in the scope of goal *Get member data* and *Get member info*, *having* the right to further authorising other actors, and is authorised by *Authorization* to *read*, *produce* and *transmit* and prohibited to *modify* information *PersonalData* and *MembershipStatus*, in the scope of goal *Get member data*, *having* the right to further authorising other actors.
- *Role* **Database**:
    - **Database** authorises *Authorization* to *read*, *produce* and *transmit* and prohibits to *modify* information *PersonalData* and *MembershipStatus*, in the scope of goal *Get member data*, *passing* the right to further authorising other actors.

Security Requirements

This section provides the list of security requirements derived for the Security Requirements 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 Security Requirements project (Table 10) are:

- **Member** requires *Netflix non-repudiation-of-acceptance* of the delegation of goal *Login*, when delegating *Login* to *Netflix*.
- **Member** requires *Netflix* the *non-modification*, *non-production* and *non-disclosure* of information *PersonalData*, and *need-to-know* of these pieces of information for the goal *Login*, when authorising *Netflix* to *read PersonalData* in the scope of goal *Login*not-reauthorised is required since the authorization is non-transferable.
- **Netflix** requires *Authorization* the *non-production* and *non-disclosure* of information *PersonalData* and *MembershipStatus*, and *need-to-know* of these pieces of informations for the goals *Get member data* and *Get member info*, when authorising *Authorization* to *read* and *modify PersonalData* and *MembershipStatus* in the scope of goals *Get member data* and *Get member info*not-reauthorised is required since the authorization is non-transferable.
- **Authorization** requires *Database* an *availability* level of 100.0% and *trustworthiness*, when delegating *Get member data* to *Database*; while it requires *Database* an *availability* level of 100.0% and *trustworthiness*, when delegating *Get membership status* to *Database*.
- **Database** requires *Authorization* a *receiver-confidentiality* , when transmitting *Personal data file* to *Authorization*requires *Authorization* a *receiver-confidentiality* , when transmitting *Membership file* to *Authorization*.
- **Database** requires *Authorization* the *non-modification* of information *PersonalData* and *MembershipStatus*, and *need-to-know* of these pieces of informations for the goal *Get member data*, when authorising *Authorization* to *read*, *produce* and *distribute PersonalData* and *MembershipStatus* in the scope of goal *Get member data*.

Responsible	Security Requirement	Requester	Description
Netflix	non-repudiation-of-acceptance (delegated(Member,Netflix,Login))	Member	Member require non-repudiation-of-acceptance for goal Login,when delegating Login to Netflix.

	non-modification (PersonalData)	Member	Member requires Netflix non-modification of Information PersonalData.
	non-production (PersonalData)	Member	Member requires Netflix non-production of Information PersonalData.
	non-disclosure (PersonalData)	Member	Member requires Netflix non-disclosure of Information PersonalData.
	need-to-know (PersonalData) (Login)	Member	Member requires Netflix need-to-know of Information PersonalData, in the scope of goal Login.
	not-reauthorized ({PersonalData},{Login},{ R})	Member	Member wants Netflix not to redistribute permissions on information {PersonalData} to other actors.
Authorization	trustworthiness (Database, delegated(Authorization, Database,Get member data))	Authorization	Database shall provide proof of trustworthiness for Authorization to delegate him goal Get member data.
	trustworthiness (Database, delegated(Authorization, Database,Get membership status))	Authorization	Database shall provide proof of trustworthiness for Authorization to delegate him goal Get membership status.
	recivier-confidentiality (transmitted(Database,A uthorization,Personal data file))	Database	Authorization shall ensure the confidentiality of transmission of the document Personal data file being transmitted.
	recivier-confidentiality (transmitted(Database,A uthorization,Membership file))	Database	Authorization shall ensure the confidentiality of transmission of the document Membership file being transmitted.
	non-production (PersonalData,Membersh ipStatus)	Netflix	Netflix requires Authorization non-production of Information PersonalData and MembershipStatus.
	non-disclosure (PersonalData,Membersh ipStatus)	Netflix	Netflix requires Authorization non-disclosure of Information PersonalData and MembershipStatus.
	need-to-know (PersonalData,Membersh ipStatus) (Get member data,Get member info)	Netflix	Netflix requires Authorization need-to-know of Information PersonalData and MembershipStatus, in the scope of goal Get member data and Get member info.
	not-reauthorized ({PersonalData,Members	Netflix	Netflix wants Authorization not to redistribute

	hipStatus},{Get member data,Get member info},{R})		permissions on information {PersonalData,Membership Status} to other actors.
	not-reauthorized ({PersonalData,MembershipStatus},{Get member data,Get member info},{M})	Netflix	Netflix wants Authorization not to redistribute permissions on information {PersonalData,Membership Status} to other actors.
	non-modification (PersonalData,MembershipStatus)	Database	Database requires Authorization non-modification of Information PersonalData and MembershipStatus.
	need-to-know (PersonalData,MembershipStatus) (Get member data)	Database	Database requires Authorization need-to-know of Information PersonalData and MembershipStatus, in the scope of goal Get member data.
Database	availability (Get member data,100.0%)	Authorization	Authorization require Database to assure an availability level of 100.0% for goal Get member data.
	availability (Get membership status,100.0%)	Authorization	Authorization require Database to assure an availability level of 100.0% for goal Get membership status.

Table 10 - Security Requirements for the Security Requirements Project

Table 11 summarises the authorizations actors in the Security Requirements project grant to one another.

Authorisor	Information	Goal	Allowed Operations	Denied Operations	Authorisee	Description
Member	PersonalData	Login	R	M, P, T	Netflix	Non-transferable authority
Netflix	PersonalData Membership Status	Get member data Get member info	R, M	P, T	Authorization	Non-transferable authority
Database	PersonalData Membership Status	Get member data	R, P, T	M	Authorization	Transferable authority

Table 11 - Authorizations in the Security Requirements project

Well-formedness Analysis

The purpose of well-formedness analysis is to verify whether the diagram for the project Security Requirements 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 Security Requirements project didn't find any errors.

Security Analysis

The purpose of security analysis is to verify whether the diagram for the project Security Requirements 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 the Security Requirements has identified the problems summarised in Table 12.

Type	Category	Text	Description
ERROR	Authorization Conflict check	There is a conflict of authorizations related to the transmission of information PersonalData for actor Authorization	There is a conflict of authorizations related to the transmission of information PersonalData for actor Authorization
ERROR	Authorization Conflict check	There is a conflict of authorizations related to the production of information PersonalData for actor Authorization	There is a conflict of authorizations on production of information PersonalData for Authorization, since there are two incoming authorizations to Authorization, one from Database allowing Authorization and the other one from Netflix requiring non-production of information PersonalData.
ERROR	Authorization Conflict check	There is a conflict of authorizations related to the production of information MembershipStatus for actor Authorization	There is a conflict of authorizations on production of information MembershipStatus for Authorization, since there are two incoming authorizations to Authorization, one from Database allowing Authorization and the other one from Netflix requiring non-production of information MembershipStatus.
ERROR	Authorization Conflict check	There is a conflict of authorizations related to the modification of information MembershipStatus for actor Authorization	There is a conflict of authorizations on modification of information MembershipStatus for Authorization, since there are two incoming authorizations to Authorization, one from

			Netflix allowing Authorization and the other one from Database requiring non-modification of information MembershipStatus.
ERROR	Authorization Conflict check	There is a conflict of authorizations related to the transmission of information MembershipStatus for actor Authorization	There is a conflict of authorizations related to the transmission of information MembershipStatus for actor Authorization
ERROR	Authorization Conflict check	There is a conflict of authorizations related to the modification of information PersonalData for actor Authorization	There is a conflict of authorizations on modification of information PersonalData for Authorization, since there are two incoming authorizations to Authorization, one from Netflix allowing Authorization and the other one from Database requiring non-modification of information PersonalData.
ERROR	Non_Production Violation	"Database" makes an unauthorised production of information "MembershipStatus"	There is no authorization relationship towards "Database" for information "MembershipStatus", but "Database" can produce "MembershipStatus" since there is a produce relationship from its goal "Get membership status" towards document "Membership file" representing "MembershipStatus"
ERROR	Non_Production Violation	"Database" makes an unauthorised production of information "PersonalData"	There is no authorization relationship towards "Database" for information "PersonalData", but "Database" can produce "PersonalData" since there is a produce relationship from its goal "Get member data" towards document "Personal data file" representing "PersonalData"
ERROR	Non_Disclosure Violation	"Database" makes an unauthorised distribution of information "MembershipStatus"	There is no authorization relationship towards "Database", but "Database" is distributing "MembershipStatus" to "Authorization" by providing document "Membership file" to "Authorization"
ERROR	Non_Disclosure Violation	"Database" makes an	There is no authorization

		unauthorised distribution of information "PersonalData"	relationship towards "Database", but "Database" is distributing "PersonalData" to "Authorization" by providing document "Personal data file" to "Authorization"
ERROR	Explicit non-reauthorization	"Netflix" violates its authority passing permissions without having the authority to transfer rights	"Netflix" has no authority to transfer authority to other actors, but it still authorises "PersonalData"
ERROR	Non-reauthorization Violation: read	"Netflix" violates its authority passing permission to read, in an unauthorised way	"Netflix" has no authority to read information "MembershipStatus", but still authorises "Authorization" to read "MembershipStatus"
ERROR	Non-reauthorization Violation: read	"Database" violates its authority passing permission to read, in an unauthorised way	"Database" has no authority to read information "MembershipStatus", but still authorises "Authorization" to read "MembershipStatus"
ERROR	Non-reauthorization Violation: read	"Database" violates its authority passing permission to read, in an unauthorised way	"Database" has no authority to read information "PersonalData", but still authorises "Authorization" to read "PersonalData"
ERROR	Non-reauthorization Violation: modify	"Netflix" violates its authority passing permission to modify, in an unauthorised way	"Netflix" has no authority to modify information "MembershipStatus", but still authorises "Authorization" to modify "MembershipStatus"
ERROR	Non-reauthorization Violation: produce	"Database" violates its authority passing permission to produce, in an unauthorised way	"Database" has no authority to produce information "PersonalData", but still authorises "Authorization" to produce "PersonalData"
ERROR	Non-reauthorization Violation: produce	"Database" violates its authority passing permission to produce, in an unauthorised way	"Database" has no authority to produce information "MembershipStatus", but still authorises "Authorization" to produce "MembershipStatus"
ERROR	Non-reauthorization Violation: transmit	"Database" violates its authority passing permission to distribute, in an unauthorised way	"Database" has no authority to distribute information "PersonalData", but still authorises "Authorization" to distribute "PersonalData"
ERROR	Non-reauthorization Violation: transmit	"Database" violates its authority passing permission to distribute, in an unauthorised way	"Database" has no authority to distribute information "MembershipStatus", but still authorises "Authorization" to distribute "MembershipStatus"

Table 12 - Security Analysis Analysis Results

Appendix A

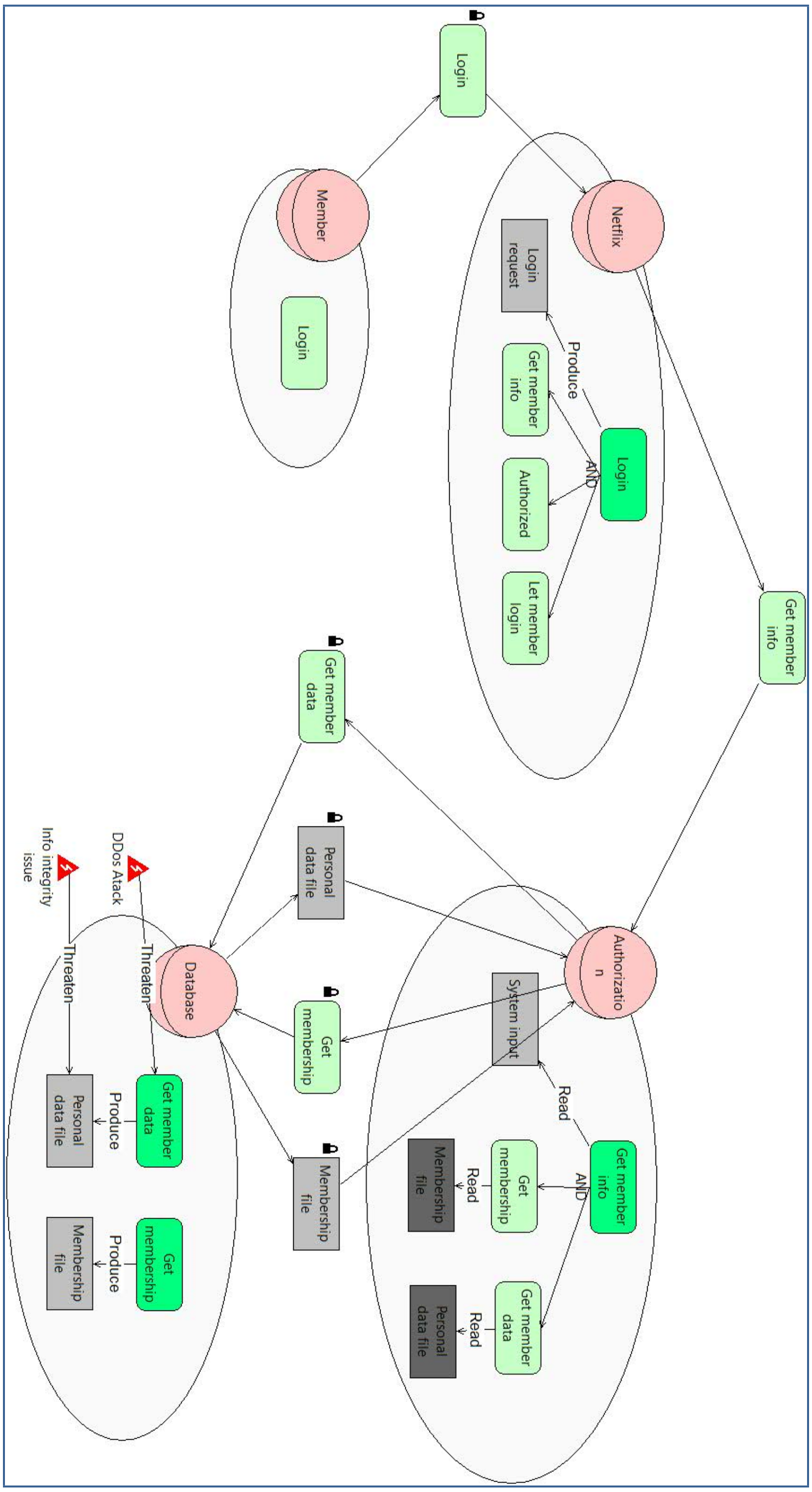

Figure 1 - Social View for the Security Requirements project

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 authoristaion. 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.