SINTEF REPORT

SINTEF ICT

Address: NO-7465 Trondheim, NORWAY
Location: Forskningsveien 1
Telephone: +47 22 06 73 00
Fax: +47 22 06 73 50

Enterprise No.: NO 948 007 029 MVA

TITLE

A Feasibility Study in Model Based Prediction of Impact of Changes on System Quality

AUTHOR(S)

Aida Omerovic, Anette Andresen, Håvard Grindheim, Per Myrseth, Atle Refsdal, Ketil Stølen and Jon Ølnes

CLIENT(S)

Research Council of Norway

REPORT NO.	CLASSIFICATION	CLIENTS REF.	
SINTEF A13339	Open	180052/S10	
CLASS. THIS PAGE	ISBN	PROJECT NO.	NO. OF PAGES/APPENDICES
Open	978-82-14-04455-3	90B245	75/9
ELECTRONIC FILE CODE		PROJECT MANAGER (NAME, SIGN.)	CHECKED BY (NAME, SIGN.)
		Ketil Stølen	Olav Skjelkvåle Ligaarden
FILE CODE	DATE	APPROVED BY (NAME, POSITION, SIGN.)	
	2010-01-26	Bjørn Skjellaug	

ABSTRACT

We propose a method, called PREDIQT, for model based prediction of impact of architecture design changes on system quality attributes. PREDIQT supports simultaneous analysis of several quality attributes and their trade-offs. This paper argues for the feasibility of the PREDIQT method based on a comprehensive industrial case study targeting a system for managing validation of electronic certificates and signatures worldwide. We first give an overview of the PREDIQT method, and then present an evaluation of the method in terms of a feasibility study and a thought experiment. The evaluation focuses on security and its trade-offs with the overall quality attributes of the target system.

Key words: Quality, Prediction, System architecture, Modeling, Simulation, Impact analysis, Design

KEYWORDS	ENGLISH	NORWEGIAN
GROUP 1	Quality, Prediction, System architecture	Kvalitet, Prediksjon, Systemarkitektur
GROUP 2	Modeling, Simulation, Impact analysis Design	Modellering, Simulering, Konsekvensanalyse, Design
SELECTED BY AUTHOR	Quality, Prediction, System architecture	Kvalitet, Prediksjon, Systemarkitektur

TABLE OF CONTENTS

A Feasibility Study in Model Based Prediction of Impact of Changes on System Quality*

Aida Omerovic[1,2], Anette Andresen[3], Håvard Grindheim[3], Per Myrseth[3], Atle Refsdal[1], Ketil Stølen[1,2] and Jon Ølnes[3]

1: SINTEF ICT, Oslo, Norway;
2: Department of Informatics, University of Oslo, Norway;
3: DNV, Høvik, Norway

Abstract. We propose a method, called PREDIQT, for model based prediction of impact of architecture design changes on system quality. PREDIQT supports simultaneous analysis of several quality attributes and their trade-offs. This paper argues for the feasibility of the PREDIQT method based on a comprehensive industrial case study targeting a system for managing validation of electronic certificates and signatures worldwide. We first give an overview of the PREDIQT method, and then present an evaluation of the method in terms of a feasibility study and a thought experiment. The evaluation focuses on security and its trade-offs with the overall quality attributes of the target system.

Key words: Quality prediction, system architecture design, change impact analysis, modeling, simulation

1 Introduction

Quality is defined as "The totality of features and characteristics of a software product that bear on its ability to satisfy stated and implied needs" [3]. Examples of quality attributes include availability, scalability, security and reliability. When adapting a system to new usage patterns or technologies, it is necessary to foresee what such adaptions of architectural design imply in terms of system quality. Predictability with respect to non-functional requirements is one of the necessary conditions for the trustworthiness of a system. Examination of quality outcomes through implementation of the different architecture design alternatives is often unfeasible. A model based approach is then the only alternative.

We have developed the PREDIQT method with the aim to facilitate prediction of impacts of architecture design changes on system quality. The PREDIQT method produces and applies a multi-layer model structure, called prediction models, which represent system design, system quality and the interrelationship between the two. Our overall hypothesis is that the PREDIQT method

* Originally published in: F. Massacci, D. Wallach, and N. Zannone (Eds.): ESSoS 2010, LNCS 5965, pp. 231-240, 2010.

can, within practical settings and with needed accuracy, be used to predict the effect of specified architecture design changes on the quality of a system. Quality is decomposed through a set of quality attributes, relevant for the target system. PREDIQT supports simultaneous analysis of all the identified quality attributes and their trade-offs. The PREDIQT approach of merging quality concepts and design models into multiple, quality attribute oriented "Dependency Views" (DVs), and thereafter simulating impacts of architecture design changes on quality, is novel.

The prediction models represent system relevant quality concepts (through "Quality Models") and architecture design (through "Design Models"). In addition, the prediction models comprise "Dependency Views" (DVs), which represent the dependencies between architecture design and quality in form of multiple weighted and directed trees. The aim has been to develop a method that provides the necessary instructions and means for creating and maintaining prediction models of sufficient quality. That is, prediction models which provide the necessary means for an analyst to simulate the impacts of architecture design-related changes on system quality at both quality attributes and overall quality level. Architectural adaptions are the main objective of PREDIQT, but the aim is also to support prediction of effects of low-level design changes. Our intention is that PREDIQT, combined with a cost-benefit analysis of system changes, should ease the task of prioritizing between a set of architecture designs, according to their cost-effectiveness.

This paper reports on experiences from using the PREDIQT method in a major industrial case study focusing on a so-called "Validation Authority" (VA) system [19] for evaluation of electronic identifiers (certificates and signatures) worldwide. We first give an overview of the PREDIQT method, and then present an evaluation of the method in terms of a feasibility study and a thought experiment (which complemented the feasibility study). Security is one of the three major quality attributes that the VA quality is decomposed into. The evaluation focuses on security and its trade-offs with the overall quality attributes identified and defined with respect to the VA system, namely scalability and availability. The results indicate that PREDIQT is feasible in practice, in the sense that it can be carried out on a realistic industrial case and with limited effort and resources. Moreover, results of the evaluation of PREDIQT based on the thought experiment, are promising.

The paper is organized as follows: An overview of the PREDIQT method is provided in Section 2. Section 3 presents the feasibility study. Section 4 presents the setup and results from the thought-experiment. Results from the evaluation based on the feasibility study and the thought-experiment are discussed in Section 5. A discussion of the threats to validity and reliability is undertaken in Section 6. Section 7 provides an overview of related research. Concluding remarks and future work are summarized in Section 8.

The appendices appear in the following order: Appendix 1 provides an overview of the workshops performed during the trial, with the related activities and the deliverables. Appendix 2 includes a selection of the design models of the VA,

with their brief explanations. Appendix 3 contains the quality models of the VA, with their full structure and definitions. Appendix 4 includes the conceptual model of the VA. Appendix 5 provides the structure of an attribute specific DV, and the structure of the total quality DV of the VA system (the parameters are omitted due to confidentiality). Schema for documentation of results of the change simulation (the values are omitted due to confidentiality) is provided in Appendix 6. Appendix 7 includes the measurement plan for the VA. Schema for documentation of results of the thought experiment is included in Appendix 8. The process diagram for evaluation of the PREDIQT method is included in Appendix 9.

2 Overview of the PREDIQT Method

This section outlines the PREDIQT method, including its structure, process and outcomes. The process of the PREDIQT method consists of the three overall phases illustrated in Fig. 1. Each of these phases is decomposed into sub-phases. Sections 2.1 and 2.2 present the "Target modeling" and "Verification of prediction models" phases, respectively. The "Application of prediction models" phase consists of the three sub-phases presented in Section 2.3.

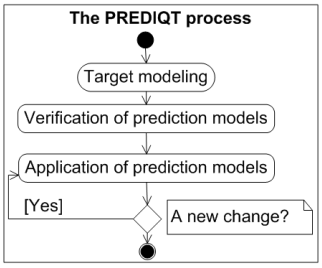

Fig. 1. The overall PREDIQT process

2.1 Target modeling

The sub-phases within the "Target modeling" phase are depicted in Fig. 2. The requirement specifications and system design models are assumed to be made available to the analysis team, along with the intended usage, operational environment constraints (if available) and expected nature and rate of changes. An overview of the potential changes helps to:

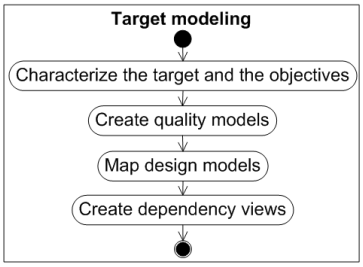

Fig. 2. Target modeling phase

- limit the scope of the prediction models to the needs of the system quality prediction, and
- focus on the relevant aspects of system quality and design during the target modeling and verification of the prediction models.

The possible architecture design changes (or their causes) include: new technology, modified usage, different dataflow, structural design changes, new components, different usage profile, changed load, changed quality characteristics of system parts, etc.

Characterize the target and the objectives: Based on the initial input, the stakeholders involved deduce a high level characterization of the target system, its scope and the objectives of the prediction analysis, by formulating the system boundaries, system context (including the operational profile), system life time and the extent (nature and rate) of design changes expected.

Create quality models: Quality models are created in the form of a tree, by decomposing total quality into the system specific quality attributes. A qualitative interpretation and a formal rating of each quality attribute are provided. Each quality attribute relates to a number of indicators, called sub-characteristics. Each sub-characteristic is assigned a qualitative interpretation and a formal measure. The purpose of having both formal and informal parts of the definitions is two-fold:

- putting the attribute and the sub-characteristic definitions into the context of the system under consideration, and
- providing a system-customized and unambiguous basis for incorporating both subjective and objective input (during the later sub-phases).

The quality models are customized for the system under analysis, but the individual definitions and the notation may be reused from established standards (such as [3] and [5]). For completeness purpose, each set of attributes and sub-characteristics having a common parent is supplemented by a node "Other". All nodes having a common parent are decomposed and defined so that they are orthogonal. The quality models represent a taxonomy with interpretations and formal definitions of system quality notions, but no value assignments to the notions are defined. The output of this sub-phase are "quality models".

Map design models: The initially obtained design models are customized so that:

1. only their relevant parts are selected for use in further analysis, and
2. a mapping within and across high-level design and low-level design models (if available), is made.

The mapped models result in a class diagram which includes the relevant elements and their relations only. The output of this sub-phase are "design models".

Create dependency views: In order to support traceability to (and between) the underlying quality models and the mapped design model, a conceptual model

(a tree formed class diagram) where classes represent elements from the underlying design and quality models, relations show the ownership, and the class attributes indicate the dependencies, interactions and the properties, is created. The conceptual model is then transformed into a generic DV – a directed tree (without any parameter values assigned) representing relationships among quality aspects and the design of the system. For each quality attribute defined in the quality model, a quality attribute specific DV is created, by a new instantiation of the generic DV. Each set of nodes having a common parent is supplemented with an additional node called "Other", for completeness purpose. In addition, a total quality DV is deduced from the quality models. The parametrization of the quality attribute specific DVs involves assigning a "degree of Quality attribute or Characteristic Fulfillment" (QCF) value to each leaf node, followed by assigning "Estimated degree of Impact" (EI) values to arcs. A QCF expresses the value of the rating of the attribute in question (the degree of its fulfillment), within the domain of the node. Due to the rating definitions, the values of QCFs are constrained between 0 (minimum) and 1 (maximum). The EI value on an arc expresses the degree of impact of a child node (which the arc is directed to) on the parent node, or to what degree the parent node depends on a child node. An EI value is assigned with respect to the sub-characteristics of the quality attribute under analysis (defined in the quality models) and their respective impact on the relation in question. Once the EIs are evaluated for all arcs directed to nodes having a common parent, their values are normalized so that they sum up to 1, due to model completeness.

The total quality DV is assigned weights on the arcs, which, based on the attribute definitions in the quality models, express the impact of each attribute (in terms of the chosen stakeholder's gain or business criticality) on the total system quality. The weights are system general objectives. The weights are normalized and sum up to 1, since also this DV is complete. The leaf node QCFs of the total quality DV correspond to the root node QCFs of the respective quality attribute specific DVs. The output of this sub-phase is one DV per quality attribute defined in the quality models and a total quality DV, with assigned estimates. Evaluation of the DV parameters may be based on automated data acquisition or judgments of domain experts.

The set of the preliminary prediction models developed during the "Target modeling" phase, consists of design models, quality models and the DVs, depicted in Fig. 3. The conceptual model is derived from the design models and the quality models. The conceptual model is transformed into a generic DV model, which is then instantiated into as many quality attribute specific DVs as there are quality attributes defined in the quality model (the total number of the quality attributes, and therefore the quality attribute specific DVs, corresponds to "n" on Fig. 3). The total quality model is an instantiation of the top two levels of the quality model. The relationship between the quality attribute DVs and the total quality DV is due to the inheritance of the root node QCF value of each quality attribute DV into the respective leaf node of the total quality DV.

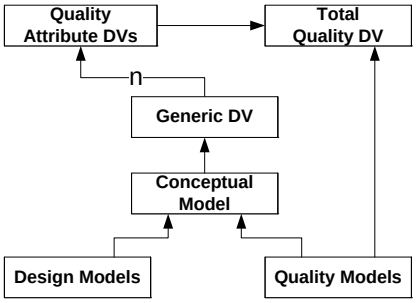

Fig. 3. Prediction models – an overview

2.2 Verification of prediction models

The "Verification of prediction models" phase (depicted in Fig. 4) aims to validate the prediction models (in terms of the structure and the individual parameters), before they are applied. The prediction models are revised and fitted until an acceptable threshold of deviation between estimates and measurements is reached. Empirical evaluations are made on the influential and uncertain parts of the prediction models. Such a threshold is expressed in terms of the tolerable degree of uncertainty of the predictions, and may be quality attribute specific, or general.

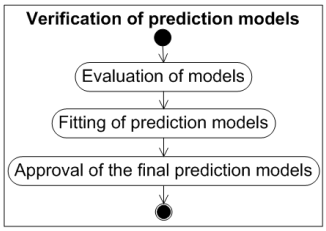

Fig. 4. Verification of models – phase

Evaluation of models: A measurement plan with the necessary statistical power is developed, describing what should be evaluated, when, how and by whom. The measurement plan is developed with respect to two major aspects:

1. parts of the DVs having dominating or uncertain values of $QCF \cdot EI$, and
2. feasibility of measurement.

Both internal and leaf nodes of DVs are evaluated, in order to test the correctness of the overall model (with respect to both structure and the inferred QCF values) and the individual estimates. The source of the input has to be more valid and reliable than the one used for estimation. Once the input is ready, it is compared with the estimates (obtained during development of the DVs) and the correspondence is evaluated. In case of uncertainty, additional or repeated measurements are undertaken. The candidates for evaluation are ranked starting from the children of the root node of each DV, and moving down the tree.

Fitting of prediction models: Model fitting is conducted in order to adjust the DV parameters to the evaluation results. The modification candidate parameters (and to some extent the degree of modification) are identified by relating the deviations uncovered during the evaluation to the relevant parts of the conceptual model. Based on this and with reference to the definitions from the quality models, the identified parameters (EIs and leaf node QCFs) are modified by the domain experts and the analyst.

Approval of the final prediction models: The objective of this sub-phase is to evaluate the prediction models as a whole and to validate that they are complete, correct and mutually consistent after the fitting. A deviation threshold expressing the acceptable deviation uncertainty at quality attribute level, is set by the stakeholders in charge. The overall deviation between estimates and measurements is compared to a pre-defined threshold and the structure is examined. If a revision of the design or quality models is needed, the target modeling phase above has to be re-initiated. In case only the DV parameters are above the deviation threshold while the model structure is approved, the verification phase is re-initialized with the fitted prediction models as input. Approval of the prediction models is a prerequisite for proceeding to the next phase. Once the prediction models are approved, they can, in the next phase, be exploited to simulate alternative ways of reaching desired quality levels by modifying system design, in a "what if" manner. Otherwise, the verification sub-phase is re-initialized with the fitted prediction models as input.

2.3 Application of prediction models

During the "Application of prediction models" phase (depicted in Fig. 5), a specified change is applied and simulated on the approved prediction models. The "Application of prediction models" phase is depicted in Fig. 5. During this phase, a proposed change is applied to the prediction models, which allow analysis of its propagation. The simulation reveals which design parts and aspects are affected by the change and the degree of impact (in terms of the quality notions defined by the quality models).

Specify a change: The change specification should clearly state all deployment relevant facts, necessary for applying the change on the prediction models. The specification should include the current and the new state and characteristics of the design elements/properties being changed, the rationale and the assumptions made.

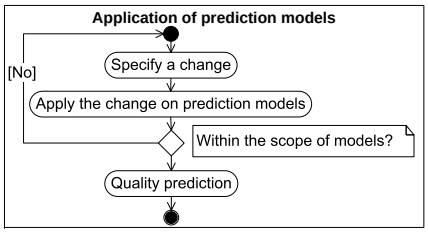

Fig. 5. Application of models – phase

Apply the change on prediction models: This phase involves applying the specified change on the prediction models. The change is applied on the design models first, reflected on the conceptual model and finally on the DVs. In case design model elements are replaced, added or moved, both structure and attributes of the conceptual model may be affected, thus causing the corresponding modifications in structure and parameters of the quality attribute DVs. If only properties or interactions of the design elements are modified, the attributes of the corresponding classes in the conceptual model are affected, and thus the corresponding parameters (EIs and leaf node QCFs) of the DVs are modified (which of the identified parameters are modified on which DVs depends on the relevant definitions provided in the quality model). If the specified change can be fully applied, it is within the scope of the prediction models, which is a prerequisite for proceeding to the next sub-phase. Otherwise, the modifications are canceled and the change deemed not predictable by the models as such. Upon applying a change on the prediction models, we can determine whether it is within the scope of the prediction models, which is a prerequisite for proceeding to the next sub-phase.

Quality prediction: The propagation of the change throughout the rest of each one of the modified DVs, as well as the total quality DV, is performed based on the general DV propagation model, according to which the QCF value of each parent node is recursively calculated by first multiplying the QCF and EI value for each closest child and then summing these products. Such a model is legitimate since each quality attribute DV is complete, the EIs are normalized and the nodes having a common parent are orthogonal due to the structure. The root node QCF values (which represent the system-level rating value of the quality attribute that the DV is dedicated to) on the attribute specific DVs are reflected to the corresponding node of the total quality DV and propagated throughout the total quality DV, according to the above mentioned general DV propagation model. The general DV propagation model can be applied since the total quality DV is complete too, the weights are normalized and the nodes are orthogonal due to their definitions in the quality model. We have developed a tool on the top of Microsoft Excel. It enables automated change propagation within and among DVs, according to the general DV propagation model. Deploying a change which is within the scope of the models does not require new target modeling or verification of the prediction models, but only adoption of the updates made during the simulation. Model erosion due to deployment of the changes within the scope is handled by re-initiating the verification phase. When it should be undertaken depends on the required prediction accuracy, which is beyond the scope of this paper.

Each DV contains two corresponding sensitivity charts, visualizing the degree of impact (on quality attribute represented by the DV) of the individual nodes (QCF) and arcs (EI), respectively. Actual deployment of a change on the system leads to adoption of the new version of the prediction models, which has resulted from the related simulation. However, if the deployed change is outside the scope

of the prediction models, the development of prediction models phase has to be re-initiated in order to re-calibrate the prediction models. Otherwise, impacts of different or additional changes may be simulated by once again initializing the "Application of prediction models" phase.

3 Application of PREDIQT in the Industrial Case

The PREDIQT method was tried out in a major industrial case study focusing on a Validation Authority (VA) [18] system. The VA offers an independent service for digital signatures and electronic ID validation, used worldwide [19]. The case study was held during six workshops, and included four domain experts and one analyst. We focus on illustrating the part of the analysis relevant to the security quality attribute and its trade-offs with the overall attributes due to its central role in the analysis, and importance for the system in question. The VA system was at the moment in a piloting phase.

3.1 Target modeling

Based on the initially obtained documentation (requirement specifications, operational environment specifications, etc.) and several interactions with the domain experts, UML based design models of the VA were developed. These are the design models that PREDIQT presupposes being in place prior to the analysis.

Characterize the target and the objectives: Our target was the VA, the purpose of which is the validation of certificates and signatures in the context of any kind of workflow involving authentication. An overview of the functional properties of the VA, the workflows it is involved in and the potential changes, was provided, in order to determine the needed level of detail and the scope of our prediction models. Among the assumed changes were increased number of requests, more simultaneous users and additional workflows that the VA will be a part of. The target stakeholder group is the system owner.

Create quality models: Requirements specifications, use cases and domain expert judgments were among the influential input in order to define and decompose the quality in a system specific, generic manner. An extract of the quality model of the VA is shown by Fig. 6. Total quality of the VA system is first decomposed into the four quality attributes: availability, scalability, security and "Other Attr." (this node covers the possibly unspecified attributes, for model completeness purpose). The X, Y, Z and V represent system quality attribute ratings, while u, v, g and j represent the normalized weights expressing each attribute's contribution to the total quality of VA. The attributes and their ratings are defined with respect to the VA system, in such a manner that they are orthogonal and together represent the total quality of the VA system. Thereafter, the sub-characteristics of each quality attribute are defined with respect to the

VA system, so that they are orthogonal to each other and represent a complete decomposition of the VA attribute in question. The sub-characteristics act as indicators, thus the dependency UML notation in Fig. 6. Values and importance of the sub-characteristics are DV node specific. Data encryption is for example more important when estimating QCF and EI of some nodes within a security DV, then others. Both the attribute ratings and the sub-characteristic metrics are defined so that their value lies between 0 (minimum) and 1 (maximum fulfillment). Qualitative and quantitative (VA specific) definitions of security and each one of its sub-characteristics are provided in the UML-based quality models.

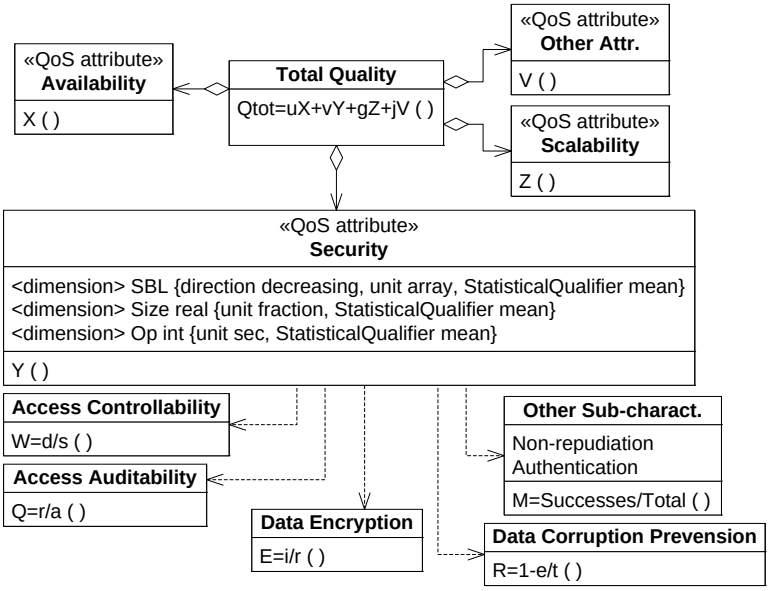

Fig. 6. An extract of the quality model

Consider the attribute "Security" in Fig. 6. Its weight is v (the coefficient of Y in the "total quality" class). The security attribute definition was based on [4]: *"The capability of the software product to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them."*. The security rating was based on [20]. The dimension-notation shown in the form of the security class attributes in Fig. 6, originates from [5]. The dimensions represent the variables constituting the security rating for the VA system. Given the attribute and rating definition, the appropriate (for the attribute and the VA system) sub-characteristics were retrieved from [3], where they are fully defined. Q, W, E, R and M represent the measure definitions of each sub-characteristic. Security depends on the five sub-characteristics displayed: access auditability, access controllability, data encryption, data corruption prevention and "other sub-characteristics" (for model completeness purpose). The access auditability sub-characteristic, for example, expresses how complete the implementation of access login is, con-

sidering the auditability requirements. Its measure is: Access auditability = $\frac{r}{a}$, where r = Nr. of information recording access log confirmed in review; a = Nr. of information requiring access log [3]. Development of quality models involved designing the structure and assigning the VA specific definitions to it. The quality models serve as a reference in the remainder of the analysis.

Map design models: The originally developed design models of the VA covered both high-level and low-level design aspects through use cases, class diagrams, composite structure diagrams, activity diagrams and sequence diagrams (SDs). A selection of the relevant models was made, and a mapping between them was undertaken. The model mapping resulted in a class diagram containing the selected elements (lifelines, classes, interfaces etc.) as classes, ownership as relations and interactions/dependencies/properties as class attributes. Due to only ownership representing a relation, this resulted in a tree-formed class diagram, so that the abstraction levels which need to be estimated are refined into the observable ones.

Create dependency views: A conceptual model (a tree-formed class diagram) with classes representing the selected elements and relations denoting ownership (with selected dependencies, interactions quality properties, association relationships or their equivalents represented as the class attributes), is deduced from:

1. the quality models, and
2. the above mentioned class diagram.

See Fig. 7 for an extract. It provided an overall model of the dependencies between quality and design aspects, traceable to the respective underlying quality and design models. The deduction of the relations was performed by following a systematic procedure: selecting the relevant parts of the underlying quality models and the design models mapping class diagram, and structuring them so that:

1. the already established relationships within quality and design models are not contradicted, and
2. the nodes having a common parent are orthogonal.

The dependent nodes were thus placed at different levels of this tree structure. The ownership relations in the conceptual model were mainly deduced from the selected generalization, realization, aggregation, composition in the (UML based) design and quality models. A generic DV – a directed tree representing relationships among quality aspects and design of the system was obtained by transforming the conceptual model. Each set of nodes having a common parent was supplemented with an additional node called “other”, for completeness purpose. Quality attribute specific DVs were then obtained by instantiating the generic DV and selecting the relevant subset of its structure. In addition, a total quality DV was instantiated from the quality models. The parametrization of

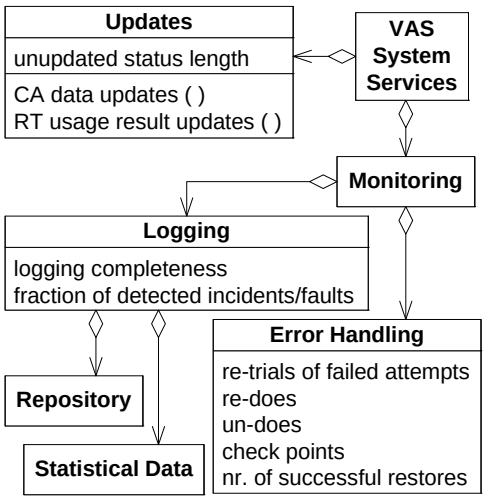

Fig. 7. An extract of the conceptual model.

the quality attribute specific DVs involved assigning a QCF value to each leaf node, followed by assigning EI values on arcs.

Totally four DVs were developed: one total quality DV and three quality attribute specific DVs, that is one for each quality attribute. The node "Other attributes" was deemed irrelevant. The DVs were then inserted into the tool we have built on top of Microsoft Excel for enabling automatic simulations within and across the DVs. DVs were, based on estimates provided by an expert panel, assigned parameter values. QCF values of attribute specific DVs were estimated by assigning a value of the quality attribute (which the DV under consideration is dedicated to) to each leaf node of the quality attribute specific DVs. The QCF value quantification involved revisiting quality models, and providing a quality attribute rating value to each node. The rating value expresses to what degree the quality attribute (given its system specific formal definition) is fulfilled within the scope of the node in question. This was followed by estimation of EI values on each arc of the quality attribute specific DVs, in a bottom-up manner. The EIs expressed the degree of influence of a child node on the parent node. The EI of an arc was assigned by evaluating the impact of the child node on its parent node, with respect to the sub-characteristics (defined in the quality models) of the attribute under consideration. Once a sum of the contributions of the sub-characteristics was obtained on each arc pointing to children nodes with a common parent, the EI values were normalized so that they sum up to 1 (due to model completeness).

Finally, the total quality DV (consisting of a root node for total quality and four children nodes, each for one of the quality attributes) was assigned weights (denoted by u, v, g and j in Fig. 6) on arcs. The weights were deduced by first evaluating the contribution (in monetary units for business goals of the

system owner) of each attribute (given its definitions) on the total quality of the system, and then normalizing the values. The leaf node QCFs correspond to the root node QCFs of the respective quality attribute DVs and are automatically reflected on the total quality DV.

Fig. 8 shows an extract of the Security attribute specific DV of the VA. The extract of the DV shown by Fig. 8 is assigned fictitious values on nodes as well as on arcs, for confidentiality reasons. In the case of the "Error detection" node of Fig. 8, the QCF value expresses the effectiveness of error detection with respect to security. The EI value on an arc expresses the degree of impact of the child node (which the arc is directed to) on the parent node, that is to what degree the error handling depends on the error detection. Thus, QCFs as well as EIs of this particular DV are estimated with reference to the definition of Security attribute and its sub-characteristics, respectively. The definitions are provided by the quality models exemplified in Fig. 6. Once estimates of leaf nodes' QCFs and all EIs are provided, the QCF values on the non-leaf nodes of all the DVs are automatically inferred based on the general DV propagation model.

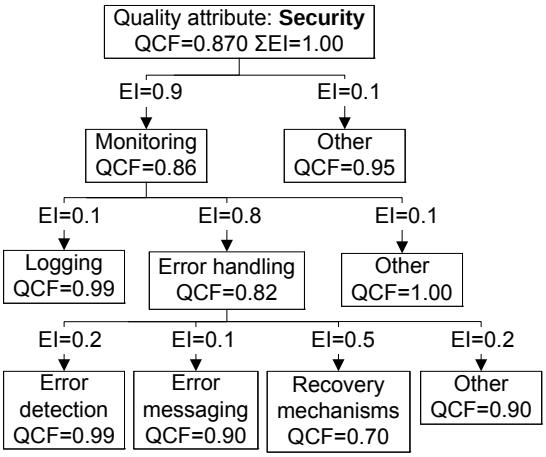

Fig. 8. A part of the Security DV.

3.2 Verification of prediction models

Verification of the prediction models for the VA relied on empirical input consisting of measurements and expert judgments.

Evaluation of models: Once the target modeling was completed, a measurement plan for the VA was developed. The uncertain and dominant parameters

were selected[1], and the appropriate measurements on the VA were precisely defined for each one. The expected values of the measurements were calculated by the analyst, but kept unrevealed for the participants involved in the measurement. Among the measurements performed on the VA were:

1. time to handle 30 simultaneous transactions of varying size and random kind (signature validation or certificate verification), and
2. time taken to download 60 CRLs (Certificate Revocation Lists) simultaneously.

The first one was repeated with different numbers of users. The second one distinguished the time taken by network, server and database. The measurement process was revisited until the empirical values of the parameters selected were obtained. The parameters and their empirical counterparts were then compared, when evaluating their magnitude of average deviation. The measurements were provided in Microsoft Excel sheets, where statistical analysis was performed in order to determine the impact of the file size, number of users, network, server, database and type of request. The mapping between the measurements and the parameters was made according to the quality model and the conceptual model. Microsoft Excel allowed relating the measurements to the parameters and expected values, and automatically analyzing their deviations in terms of measures of tendency and charts. The importance of the usage profile became evident, particularly for scheduling and performance, which are the main factors of the response time. In addition, a walkthrough of the selected internal nodes was made in order to test the automatically deduced values and to increase the statistical power. The measurements were collected and domain experts' estimates obtained, before the inferred parameter values and statistical analysis were revealed for the domain experts.

Fitting of the prediction models: The analysis of the deviations between the models and the measurements were presented in terms of statistical operators and charts. The walkthrough of the analysis resulted in 3-5 slightly fitted parameters per quality attribute specific DV. Specification of values is omitted for confidentiality reasons.

Approval of the final prediction models: The fitted parameters on the quality attribute specific DVs were automatically propagated upwards in the four DVs. The inferred values were then compared with the empirical or expected input and judged by the domain experts. Since the QCFs on the root nodes of the attribute specific DVs denote the rating values at the VA system level, their inferred values were compared with the known ones for availability, security and scalability, resulting in a negligible magnitude of average deviation. The approval

[1] The existing estimates of downtime, quality of logging, degree of security breaches, quality of encryption, access controllability, access auditability, security of the database, communication protocol security, etc., were known to be reliable.

of the fitted prediction models was obtained when the domain experts agreed that an acceptable model quality (in terms of the deviation threshold, contents and scope of the models) was reached.

3.3 Application of prediction models

The approved prediction models were applied for simulation of impacts of 14 specified architecture design changes on the VA quality. Each specified architecture design change was first applied on the affected design models, followed by the conceptual model and finally the DVs.

Specify a change: Totally 14 changes were specified, as shown by Table 1. Some of the changes (e.g. change 1) addressed specific architecture design aspects, others referred to the system in general (e.g. change 5), while the overall changes (e.g. changes 6 through 14) addressed parameter specifications of the DVs. The specification suggested each change being independently applied on the approved prediction models.

Nr.	Change specification
1	Split signature validation (SV) component into two redundant components, with load balancing.
2	Merge certificate verification (CV) and signature validation (SV) interfaces (not components).
3	Deploy redundant VA with external workload balancing.
4	Average size (nr. of data elements and data amount) of SV requests increases by 100%.
5	Use of gateway made mandatory.
6	Decrease error detection QCF for availability by 50%.
7	Decrease coupling QCF for availability by 50%.
8	Decrease upgrading QCF for availability by 50%.
9	Increase scalability QCF under modularity by 66%.
10	Increase scalability QCF upgrading by 56%.
11	Increase scalability QCF “Measures for protection of operational environment” by 37%.
12	Increase security QCF “Logging” by 16%.
13	Increase security QCF “Monitoring” by 16%.
14	Increase security QCF “Measures for protection of operational environment” by 5%.

Table 1. Change specification table

Apply the change on prediction models: All the 14 changes were successfully applied on the prediction models, thus showing to be within the scope of the models. Due to the space limitation, this subsection focuses on change nr.

3. We assume that the production environment of the VA is currently not redundant, and there is thus no load balancing. The change implies introducing VA redundancy (that is a duplicate installation of the entire VA system: client, server, database, etc.) and a separate load balancing mechanism. The latter involves distribution and scheduling of all requests by delegating them to one of the two VA installations. The change involved modification of:

1. a composite diagram of the VA environment
2. a class diagram with the VA interfaces, and
3. a sequence diagram (SD) showing the interaction between VA, CA and a relying party.

The last one referenced three additional SDs modeling detailed interactions in relation to certificate validation and signature verification. These modifications of the design models were sufficient for reflecting the specified change on the design diagrams. Since no structural adjustments of the design models were necessary, no overall prediction models were affected in terms of the structure. Next, the affected attributes of the conceptual model were identified and mapped to the corresponding parameters of the quality attribute specific DVs. The parameters were modified with reference to their definitions in the quality model and the changes were documented. Which DVs were affected and the extent of modification of the identified parameters, was dictated by the respective definitions in the quality models. In the case of change 3 from Table 1, all DVs were affected.

Change number: 3	**Availability (Initial QCF=X)**	**Security (Initial QCF=Y)**	**Scalability (Initial QCF=Z)**
QCF changed on leaf nodes:	SW recovery $a \rightarrow a'$ Modularity $b \rightarrow b'$ Monitoring $c \rightarrow c'$ Middleware $d \rightarrow d'$	Redundancy $m \rightarrow m'$ Modularity $n \rightarrow n'$	VA server $g \rightarrow g'$ Message routing $h \rightarrow h'$ Modularity $i \rightarrow i'$ Updates $j \rightarrow j'$
EI changed on arcs to nodes:	Hardware $e \rightarrow e'$ Upgrading $f \rightarrow f'$	Hardware $o \rightarrow o'$ Upgrading $p \rightarrow p'$	Hardware $k \rightarrow k'$ Upgrading $l \rightarrow l'$
New root node QCF:	X'	Y'	Z'

Table 2. An outline of the change simulation table

The result of the parameter modifications was a table of the form illustrated by Table 2 with modified EIs and leaf node QCFs on each DV. In the case of the security DV for change 3, two leaf node QCFs were increased due to the duplication of VA: Redundancy and Modularity. These two aspects fulfill (within their respective domains) the security attribute better, given this change. Both redundancy and modularity are now improved, with respect to security (given its VA specific definition from the quality models). Specification of the initial values (denoted by a-p and X-Z) and the values deduced from the above presented steps (denoted by a'-p' and X'-Z') are omitted due to their confidentiality.

Generally, for each quality attribute specific DV:

1. The DV structure was modified in order to maintain consistency with the modified conceptual model (tree formed class diagram mapping the design and quality models) which the DVs are instantiated from.
2. For those leaf nodes that were directly traceable to the affected attributes (which represent properties, interactions and dependencies in the design models) of the conceptual model illustrated by Fig. 7, the leaf node QCFs were modified by the analyst (based on the quality attribute rating definition) if appropriate for the DV in question (recall the quality model).
3. The affected arcs were identified, based on the affected attributes of the conceptual model (illustrated by Fig. 7). The EI values on the affected arcs were changed by the analyst, by re-evaluating the impact of the sub-characteristics of the attribute that the DV is dedicated to and normalizing them on the arcs pointing to the nodes having a common parent. Which DVs were affected and the extent of modification of the identified EIs on them, was determined by the quality models.
4. The modifications, their rationale and the results were documented.

Quality prediction: The tool support enabled automated change propagation within and among DVs. The propagation within the DVs involved automatic propagation of the affected QCF values and sensitivity charts, based on the general DV propagation model, while propagation among DVs involved reflecting modifications of attribute specific DVs, on the total quality DV. The impact of a change with respect to quality could be observed from the inferred parameters of the respective DVs. For each change, its predicted effects were documented by a snapshot of the initial prediction models, change application steps, and a change simulation table (an extract is illustrated by Table 2). X', Y' and Z' represent the new values (due to change 3) for availability, security and scalability, respectively. They are equivalent to the predicted root node QCF values on the respective quality attribute specific DVs, given that change 3 is deployed. The new total quality value was then automatically obtained from the total quality DV. Since our specification treated the 14 changes independently from each other, the prediction models were initialized (to the approved ones) prior to starting the simulation of each change.

Each DV contained two corresponding sensitivity charts, visualizing the degree of impact (on the root node QCF) of the individual nodes (QCF) and arcs (EI), respectively. One of the sensitivity charts, showing the impact of the QCF values[2] from some of the second top level nodes of the security attribute DV, is illustrated by Fig. 9. The sensitivity charts show the possible intersection points (and thus the corresponding values of the overall variables as well as the top node value) assuming a changed value of a parameter. A value update on a DV resulted in an automatic update of the sensitivity charts. Effects of each change with respect to quality at all levels above the modified ones, were automatically

[2] The input values the sensitivity chart is based on, are imaginary due to confidentiality of the DV parameter values.

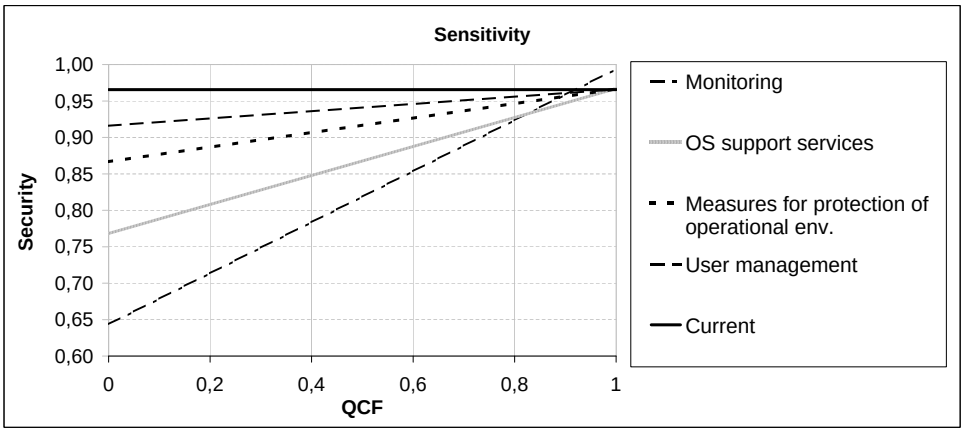

Fig. 9. An extract of a sensitivity chart for QCF of selected nodes

propagated. The impact of a change with respect to quality could be observed from:

1. the sensitivity charts showing slopes and intersections of the various QCFs and EIs at their respective abstraction levels, or
2. the inferred parameters of the respective DVs.

4 Evaluation Based on a Thought Experiment

As mentioned earlier, we basically did two evaluations. Firstly, we did a feasibility study, as described above, with the objective of evaluating whether PREDIQT could be used in a practical setting and deliver useful results. Secondly, we conducted a thought experiment. This section concentrates on the latter.

Set-up: Due to the involvement of several participants, a certain presence of the human factors was inevitable. Therefore, we briefly present the setup[3]. The analysis leader had more than six years of relevant experience in architecture design and software engineering. She served as the method developer and the analyst. She developed the VA prediction models, while interacting with the domain experts and then performed the quality simulations presented in Section 3.3. The domain expert panel consisted of four professionals from DNV [1], with many years of relevant experience within software engineering. Two of the domain expert panel members were involved in the development and verification of the VA prediction models.

[3] As of the initialization of the trial in February 2008

Research method: None of the changes specified had been deployed on this particular system previously, although some freely available experience factories from changes of this kind on similar systems could be recalled by the analysis participants. The simulation (presented in Section 3.3) of impacts of the 14 specified changes on quality of the VA system was performed by the analyst. Both the simulation approach and all the resulting predictions were documented, stored by an additional analysis participant, and kept unrevealed for the domain expert panel until their completion of the thought experiment. The domain expert panel was, by the analyst, given a brief presentation and handouts of the approved prediction models (which they had been involved in the development and verification of) for the VA. The changes were considered independently of each other. For each one of the 14 specified changes we went through the following three steps:

1. The change was presented by the analyst, by only providing the change facts.
2. The preliminary design model modifications (if any) for the change under consideration were proposed by the analyst and then further revised by the domain expert panel.
3. The domain experts were asked to estimate a new root node QCF value (represented by X', Y' and Z' in Table 2) for each quality attribute specific DV, assuming the change is deployed (the specific values obtained are confidential).

Results: The results expressing the magnitude of average deviation $AD = \frac{|E-S|}{E}$ between the PREDIQT based simulations S and the empirical counterparts (thought experiment) E, are presented by Table 3. The blank fields indicate that the quality attribute was unaffected by the change and does not influence the statistical values provided at the bottom. The specific E and S values obtained for the attributes are confidential and therefore not revealed here.

5 Discussion

This section discusses the results from both the feasibility study and the thought experiment, presented by Sections 3 and 4, respectively.

Feasibility study: A lightweight post-mortem analysis of the case study was conducted. The domain experts (who are the main stakeholders: service providers, quality engineers and system architects) have expressed that the development (structuring and parametrization) of DVs was relatively simple, thanks to the comprehensible DVs (including the structure of the DVs and the general DV propagation model) as well as the confined underlying quality and design models. One of the main points of the feedback was that the reasoning around DVs, particularly their parametrization, facilitated understanding the system design and reasoning about alternatives for potential improvements, as well as existing and potential weaknesses of architectural design, with respect to quality. We

	Magnitude of average deviation (AD)			
Change	**Availability**	**Scalability**	**Security**	**Tot. quality**
1	0.005	0.024	0.000	0.002
2	0.013	0.059	0.014	0.000
3	0.009	0.077	0.025	0.017
4	0.122	0.188	0.000	0.101
5	0.026	0.033	0.197	0.019
6	0.104			0.053
7	0.082			0.043
8	0.053			0.027
9		0.100		0.023
10		0.032		0.007
11		0.031	0.000	0.006
12			0.003	0.001
13			0.008	0.002
14			0.015	0.004
Average	0.052	0.068	0.029	0.022
Stdev	0.046	0.055	0.064	0.028
Max	0.122	0.188	0.197	0.101
Min	0.005	0.024	0.000	0.000

Table 3. Results of evaluation

managed to conduct all the steps of the PREDIQT method, with limited resources and within the planned time period (six half-a-day workshops with upto one man-labour week before each). The changes specified were deduced with the objective of addressing the most relevant parts of the prediction models, being diverse (with some intended overlaps – in order to check for consistency) and realistic. The fact that all the changes were within the scope of the prediction models and could be simulated, indicates feasibility of developing the prediction models with intended scope and quality. Overall, applying PREDIQT was feasible within the practical settings of this case. The models were relatively straight forward to develop, and judged to be fairly easy to use.

Thought experiment: The AD values and their standard deviations in Table 3 are quite low, although not negligible. Table 3 indicates a strong consistency between S and E values. AD was, rather than $|E - S|$ used as a measure of deviation, in order to compensate for the quality attribute ratings being limited to a value between 0 and 1. Being defined as the fraction between $|E - S|$ and E, AD is necessarily higher than $|E - S|$, thus taking into account the size of the change relative to the initial value of the quality attribute rating. We do not have hard evidence that the predictions were correct, but the results of the thought experiment are promising.

6 Threats to Validity and Reliability

The validity of the findings with respect to (1) the feasibility of the PREDIQT method; and (2) the results of the evaluation of the predictions based on the thought experiment, depends to a large extent on how well the threats have been handled. In addition to reliability threats, four types of validity threats, presented in [9], are addressed: construct validity, conclusion validity, internal validity and external validity. Reliability is concerned with demonstrating that the operations of the case study can be repeated with the same results. Construct validity concerns whether we measure what we believe we measure. Conclusion validity concerns the composition of participants and the statistical analysis. Internal validity concerns matters that may affect the causality of an independent variable, without the knowledge of the researcher. External validity concerns the generalization of findings of this case study to other contexts and environments.

The VA is representative for the systems intended to be within the scope of the PREDIQT method. No particular customizations of the method were needed for this trial. Thus, it should be possible to reapply PREDIQT in another context. The largest threat may be related to the conclusion validity since both the model development and the thought experiment relied on the subjective estimates provided by domain experts. There was turnover among the domain experts, but two of them participated throughout the case study. The simulations themselves were conducted by the method developer before and independently from the thought experiment. The change specification included diverse, non-overlapping changes covering major parts of the prediction models. The quality attribute specific DVs were relatively complex, which minimizes the possibility that the domain experts were able to remember the DVs and thus quickly calculate propagation of the changes 6-14 during the thought experiment. Still, the standard deviations of *AD* in Table 3 were quite small. All this considered, the risk that the prediction models, the change impact simulations and the thought experiment based estimates were consistently wrong, should be relatively small. Hard evidence in the form of measurements to validate the correctness of the predictions would have been desirable, but this was unfortunately impossible within the frame of this case study. Statistical power was low, due to low number of participants. The careful selection of experienced participants and the variety of the changes might compensate for some of this.

7 Related Work

Research on quantifiable system quality improvement is extensive. Most traditional system quality research has concentrated on defects modeling, and relied on statistical techniques such as regression analysis. [21] presents risk identification techniques like principal component analysis, discriminant analysis, tree-based reliability models and optimal set reduction. Common for all these techniques is that they analyze and, to some degree, predict defects, based on raw low-level data. [5] provides a UML notation for QoS modeling, which has also

been applied in our quality models. PREDIQT is compatible with the established software quality standard [3], which is applied in this case study. The goal/question/metric paradigm [7], [6] is a significant contribution to quality control which also is compatible with PREDIQT and can be used for development of quality models and design of a measurement plan [13], [11].

[14] introduces an approach for decomposing security objectives into security patterns, which again are decomposed into security measures, which serve as indicators. The resulting dependency graphs are only developed for the security quality attribute and with limited traceability to the system design and its quality notions. Still, the idea of pattern reuse would be useful in the PREDIQT context. Pattern based quantitative security assessment is also presented in [22], where threat coverage metrics are associated with security patterns and aggregated to an overall security indicator. Again, this is a solely security oriented approach for quantitative security assessment (and not prediction) with limited traceability to the underlying design and quality notions.

[10] and [17] provide surveys of the software architecture analysis methods (SAAM, ATAM, ALPSM, SAEM etc.). Compared to PREDIQT, they are more extensive and provide a more high-level based architecture assessment of mainly single quality attributes (maintainability or flexibility). Furthermore, they are not predictive, do not incorporate measurement, and quality is defined and quantified differently. ATAM [16], [15], [8] is, for example, more coarse-grained than PREDIQT in terms of both quality definitions and measurement. PREDIQT allows a more fine grained analysis of several quality attributes and their trade-offs simultaneously and with limited effort. Hence, an integration of the two may be worthwhile examining.

According to [12], most prediction models use size and complexity metrics to predict defects. Others are based on testing data, the quality of the development process, or take a multivariate approach. In many cases, there are fundamental statistical and data quality problems that undermine model validity. In fact, many prediction models tend to model only part of the underlying problem and seriously misspecify it.

[12] argues that traditional statistical approaches are inadequate and recommends causal, holistic models for software defect prediction, using Bayesian Belief Networks (BBNs). However, a drawback both statistical and BBN based models suffer, is their scalability. Providing sufficient data for statistical models requires tedious data acquisition effort and knowledge of statistical techniques. PREDIQT resolves such scalability issues, by relying on a generic process which results in comprehensible and feasible prediction models. No particular expert knowledge regarding theories or tools is necessary. We operate with multi-level prediction models, allowing a wide range of architecture design changes to be simulated and their impacts on quality predicted at multiple abstraction levels. The method can be carried out with limited effort, and still offer a sufficient level of detail. The incorporation of the established notations, techniques and standards in PREDIQT allows for reuse of the known (and possibly already adopted by the system or the organization involved in the analysis) processes and tools.

8 Conclusions and Future Work

This paper has presented PREDIQT – a method for model based prediction of impacts of architecture design changes on system quality. We have also reported on results from applying PREDIQT in a major industrial case study, as well as on a thought experiment. The case study focused on security and its trade-offs with the overall quality attributes of the target system. The results indicate that PREDIQT is feasible in practice, in the sense that it can be carried out on a realistic industrial case and with limited effort and resources. Moreover, the evaluation of PREDIQT based on the thought experiment is promising. We expect PREDIQT to be applicable in several domains of distributed systems with high quality and adaptability demands. Handling of inaccuracies in the prediction models, improving traceability of the models and design of an experience factory, are among the planned and partially initiated future developments.

Acknowledgments. This work has been conducted within the DIGIT (180052/S10) project, funded by the Research Council of Norway. The authors acknowledge the feedback from Olav Ligaarden, Birger Møller-Pedersen and Tormod Vaksvik Håvaldsrud.

Bibliography

[1] Det Norske Veritas, http://www.dnv.com/. Accessed on 20.02.2009.

[2] International Organization for Standardization: ISO 8402 - Quality Management and Quality Assurance – Vocabulary. 1994.

[3] International Organisation for Standardisation: ISO/IEC 9126 - Software engineering – Product quality. 2004.

[4] ISO/IEC 12207 Systems and Software Engineering – Software Life Cycle Processes. 2008.

[5] Object Management Group: UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms, v. 1.1. 2008.

[6] V. Basili, G. Caldiera, and H. Rombach. The Goal Question Metric Approach. *Encyclopedia of Software Engineering*, Wiley, 1994.

[7] V. R. Basili. Software Modeling and Measurement: the Goal/Question/Metric Paradigm. Technical Report TR-92-96, University of Maryland, 1992.

[8] C. Byrnes and I. Kyratzoglou. Applying Architecture Tradeoff Assessment Method (ATAM) as Part of Formal Software Architecture Review. Technical Report 07-0094, The MITRE Corporation, 2007.

[9] T. D. Cook and D. T. Campbell. *Quasi-Experimentation: Design and Analysis Issues for Field Settings.* Houghton Mifflin Company, 1979.

[10] L. Dobrica and E. Niemel. A Survey on Software Architecture Analysis Methods. *IEEE Transactions on Software Engineering*, 28(7):638–653, 2002.

[11] C. Ebert, R. Dumke, M. Bundschuh, A. Schmietendorf, and R. Dumke. *Best Practices in Software Measurement.* Springer Verlag, 2004.

[12] N. Fenton and M. Neil. A Critique of Software Defect Prediction Models. *IEEE Transactions on Software Engineering*, 25:675–689, 1999.

[13] N. E. Fenton and S. L. Pfleeger. *Software Metrics: A Rigorous and Practical Approach.* PWS Publishing Co., 1998.

[14] T. Heyman, R. Scandariato, C. Huygens, and W. Joosen. Using Security Patterns to Combine Security Metrics. In *ARES '08: Proceedings of the 2008 Third International Conference on Availability, Reliability and Security*, pages 1156–1163, Washington, DC, USA, 2008. IEEE Computer Society.

[15] R. Kazman, M. Barbacci, M. Klein, S. J. Carriere, and S. G. Woods. Experience with Performing Architecture Tradeoff Analysis. *International Conference on Software Engineering*, 0:54, 1999.

[16] R. Kazman, M. Klein, M. Barbacci, T. Longstaff, H. Lipson, and J. Carriere. The Architecture Tradeoff Analysis Method. In *Engineering of Complex Computer Systems, 1998. ICECCS '98. Proceedings. Fourth IEEE International Conference on*, pages 68–78, Aug 1998.

[17] M. Mattsson, H. Grahn, and F. Mårtensson. Software Architecture Evaluation Methods for Performance, Maintainability, Testability and Portability.

In *Second International Conference on the Quality of Software Architectures*, Vasterås, Sweden, June 2006.

[18] J. Ølnes. PKI Interoperability by an Independent, Trusted Validation Authority. In *5th Annual PKI R&D Workshop*, NIST Gaithersburg MD, USA, 2006.

[19] J. Ølnes, A. Andresen, L. Buene, O. Cerrato, and H. Grindheim. Making Digital Signatures Work across National Borders. *ISSE 2007 Conference: Securing Electronic Business Processes*, pages 287–296, Warszawa, Vieweg Verlag, 2007.

[20] A. Omerovic. *Design Guidelines for a Monitoring Environment Concerning Distributed Real-Time Systems.* Trondheim, Tapir Academic Press, 2004.

[21] J. Tian. *Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement.* Wiley-IEEE Computer Society Press, 1 edition, 2005.

[22] A. Yautsiukhin, R. Scandariato, T. Heyman, F. Massacci, and W. Joosen. Towards a Quantitative Assessment of Security in Software Architectures. In *13th Nordic Workshop on Secure IT Systems*, Copenhagen, Denmark, 2008.

Appendix 1: Organization of the PREDIQT trial on VA

The trial of PREDIQT on the VA system was organized in totally six workshops, which enabled the covering all the parts of the PREDIQT process. The workshops with the respective objectives, deliverables, prerequisites and required participants are summarized in Table 4.

Each workshop lasted approximately three hours and the average time between each workshop was about a month. In case the goals of the workshops or the prerequisites were not fulfilled according to the schedule or if clarifications were needed before proceeding, additional efforts and interactions took place between the planned workshops. At the beginning of each workshop, an update on the overall schedule for PREDIQT was given, the goals for the current meeting were presented, and information on the work done since the previous meeting was given. Each workshop ended by planning the forthcoming activities/tasks and by setting a date for the next workshop.

Work-shop	Objective	Prerequisites	Participants
1	– The customers presentation of the system. – Identify objective and scope of the analysis. – Identify goals of the system. – Characterize main quality characteristics, operational profile, expected lifetime and expected extensions of the system. – Provide the system documentation. – Appoint the communication points and establish communication routines. – Appoint the head of the customer group.	– A brief introduction of the PREDIQT method with process, objectives and capabilities, is given. – Commitment of the management and the domain experts. – The necessary resources made available. – No interest conflicts.	– Analyst – All domain experts who will participate throughout the trial (staff responsible for design architecture, testing, project management, system operation and integration),
2	– Presentation of the target system: scope, design models and initial quality models, initial dependency view structure - by the analyst. – Customers review of the models, followed by a revision, if needed. – Views decomposition up to the measurable and needed detail.	– Sufficient feedback from the domain experts for final update of the models (without estimates).	– Analyst. – All domain experts.
3	– Approval of the initial prediction models (without estimates) of the system. – Estimation of the parameter values on the dependency views.	– The prediction models are updated according to the feedback from the previous workshop.	– Analyst. – All domain experts.
4	– Presentation of the estimated prediction models of the system. – Preparation of a measurement plan regarding the current and the changed system.	– Estimation by several appointed domain experts is finalized.	– Analyst. – Head of the domain expert group. – Domain experts responsible for data acquisition (related to evaluation).
5	– Model evaluation and fitting.	– Measurement of the actual QoS values, completed. – Statistical analysis completed.	– Analyst. – All domain experts.
6	– Applying the method for simulation of change impacts on quality.	– The fitted prediction models are approved.	– Analyst. – All domain experts.

Table 4. Organization of the PREDIQT trial on the VA

Appendix 2: A selection of the design models of the VA

This section contains a selection of the main design models of the VA system, developed in "Rational Software Modeller 6.0.1." tool:

- The use case shown in Fig. 10 models the scope of the analysis with respect to the stakeholders and processes involved.
- The class diagram shown in Fig. 11 models the classification of the quality notions for certificates and signatures handled by the VA system.
- The class diagram shown in Fig. 12 models the gateway of the VA system and its environment, with the respective operations available.
- The class diagram shown in Fig. 13 models the VA and the CA (certificate authority) with the respective environments.
- The class diagram shown in Fig. 14 models the interfaces of the VA.
- The class diagram shown in Fig. 15 models the database of the VA.
- The class diagram shown in Fig. 16 models the interfaces of the gateway.
- The composite diagram shown in Fig. 17 models the VA environment with the interactions.
- The composite diagram shown in Fig. 18 models the VA Gateway environment with the interactions.
- The sequence diagram shown in Fig. 19 models the request types.
- The sequence diagram shown in Fig. 20 models the certificate verification request.
- The sequence diagram shown in Fig. 21 models the certificate verification response.
- The sequence diagram shown in Fig. 22 models the signature validation process.
- The activity diagram shown in Fig. 23 models the certificate verification process.
- The activity diagram shown in Fig. 24 models the signature validation process.
- The class diagram shown in Fig. 25 models the data format of the signature validation.

These models were developed in the beginning of the trial, and actively used during development of the conceptual model and the dependency views, as well as during simulation of the changes. When applying changes on the prediction models, a modification of the design models was the first step, in order to identify the affected design elements and interactions, before reflecting them on the overall prediction models.

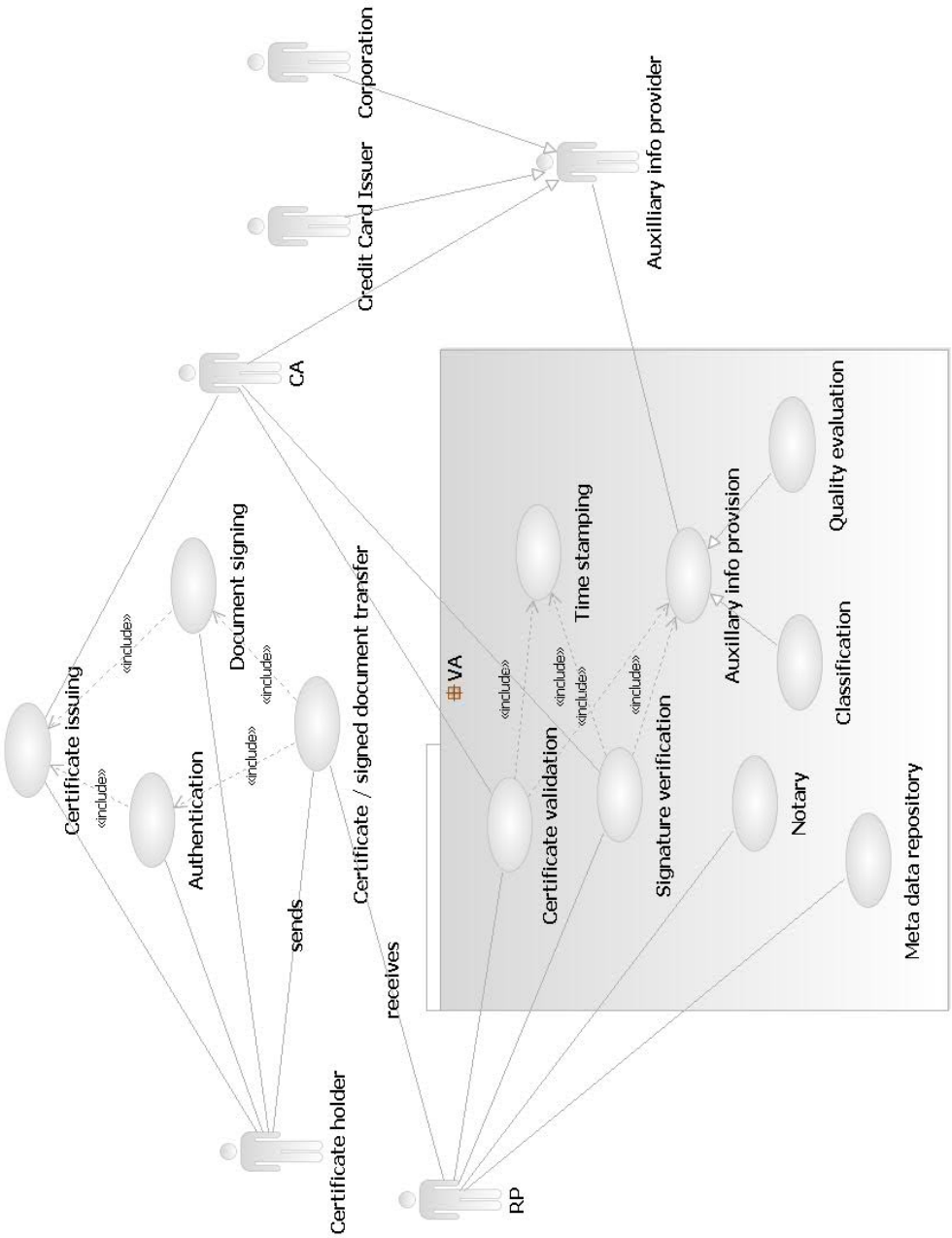

Fig. 10. A use case defining the scope, relative to stakeholders and processes

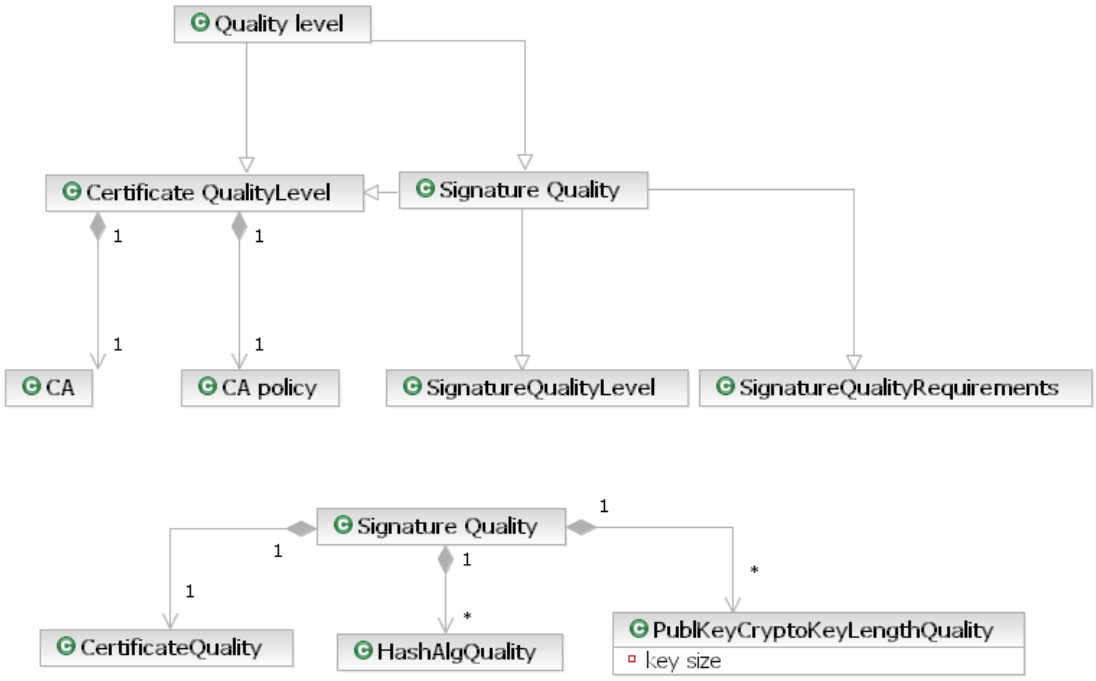

Fig. 11. Signature and certificate quality classification

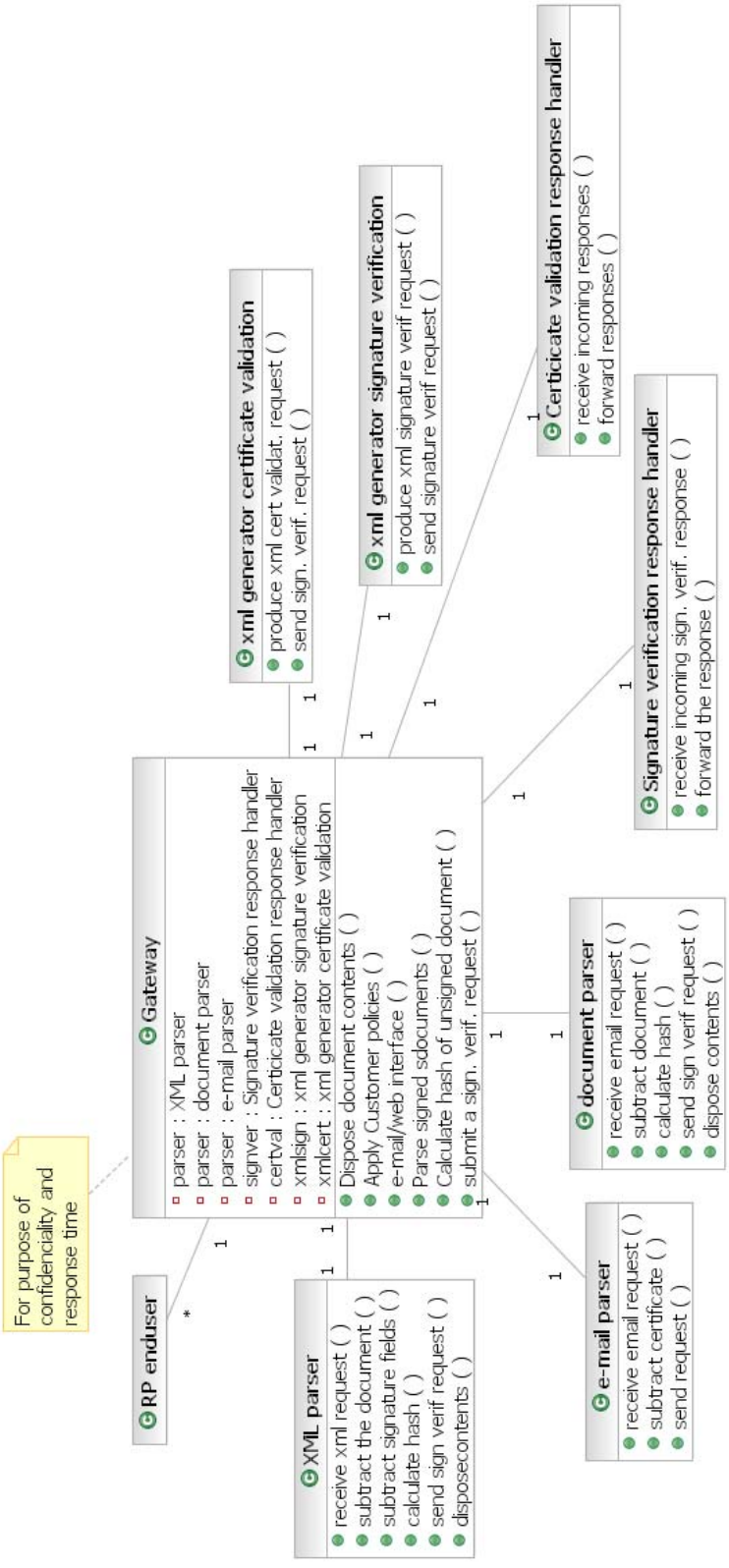

Fig. 12. Gateway with the environment

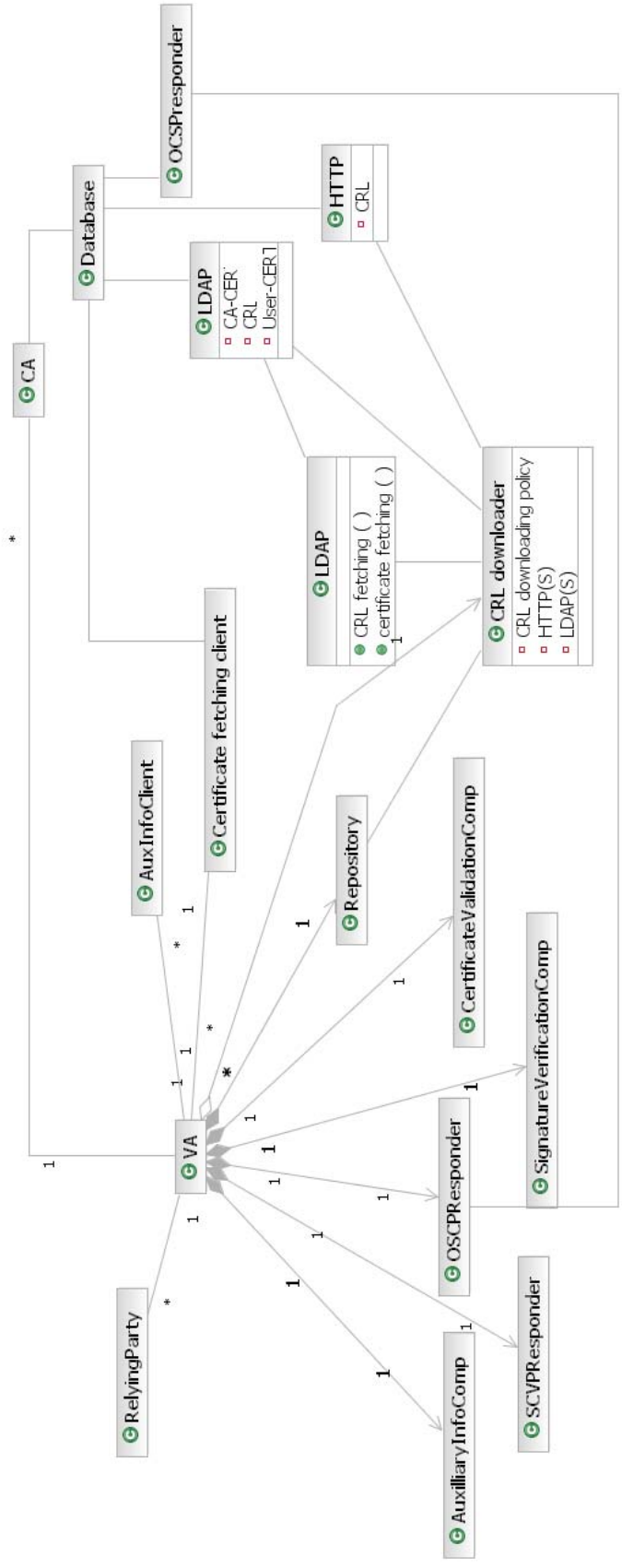

Fig. 13. VA and CA with environments

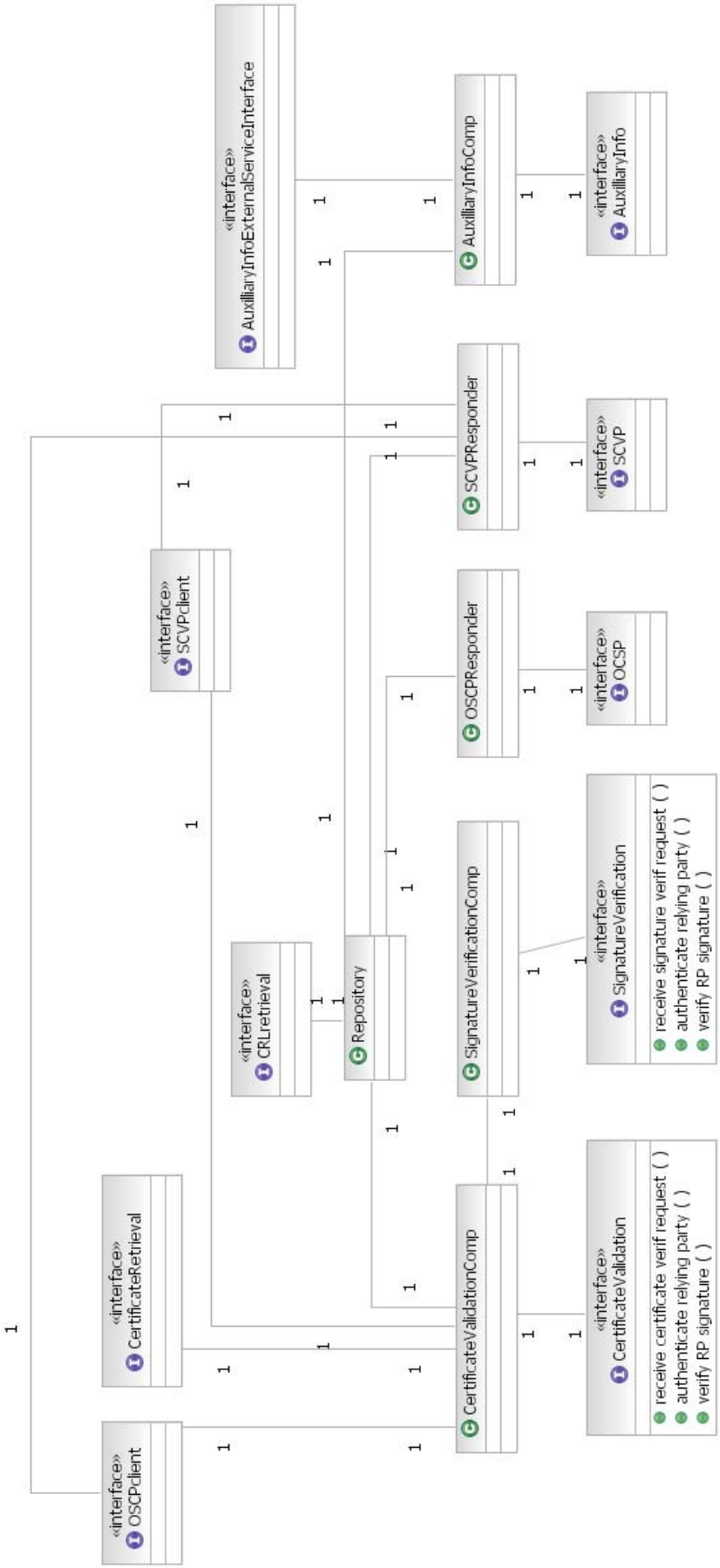

Fig. 14. VA interfaces

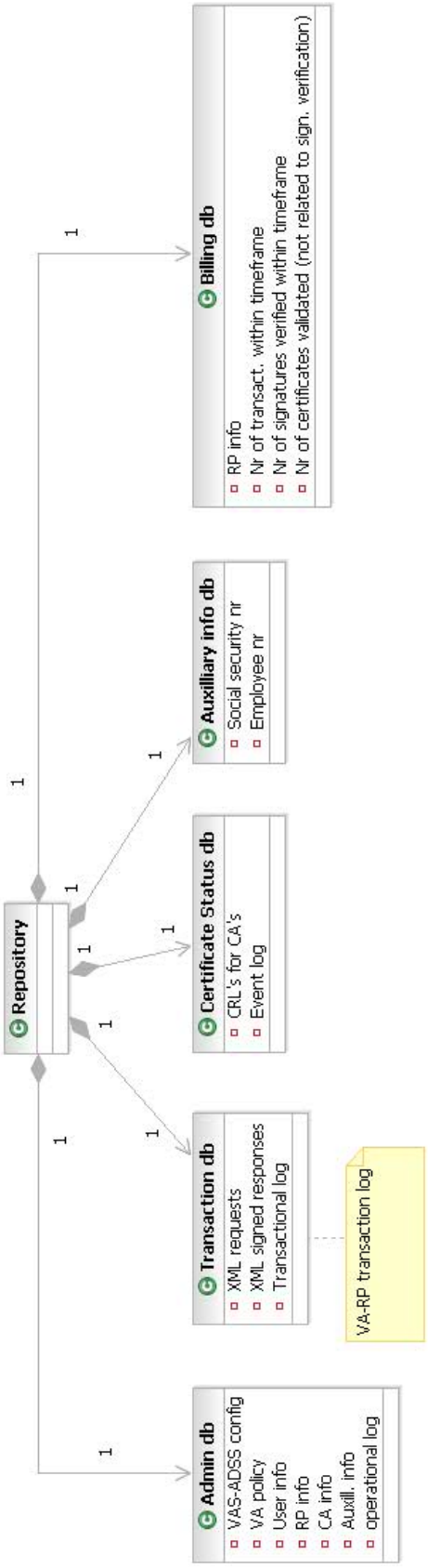

Fig. 15. VA database

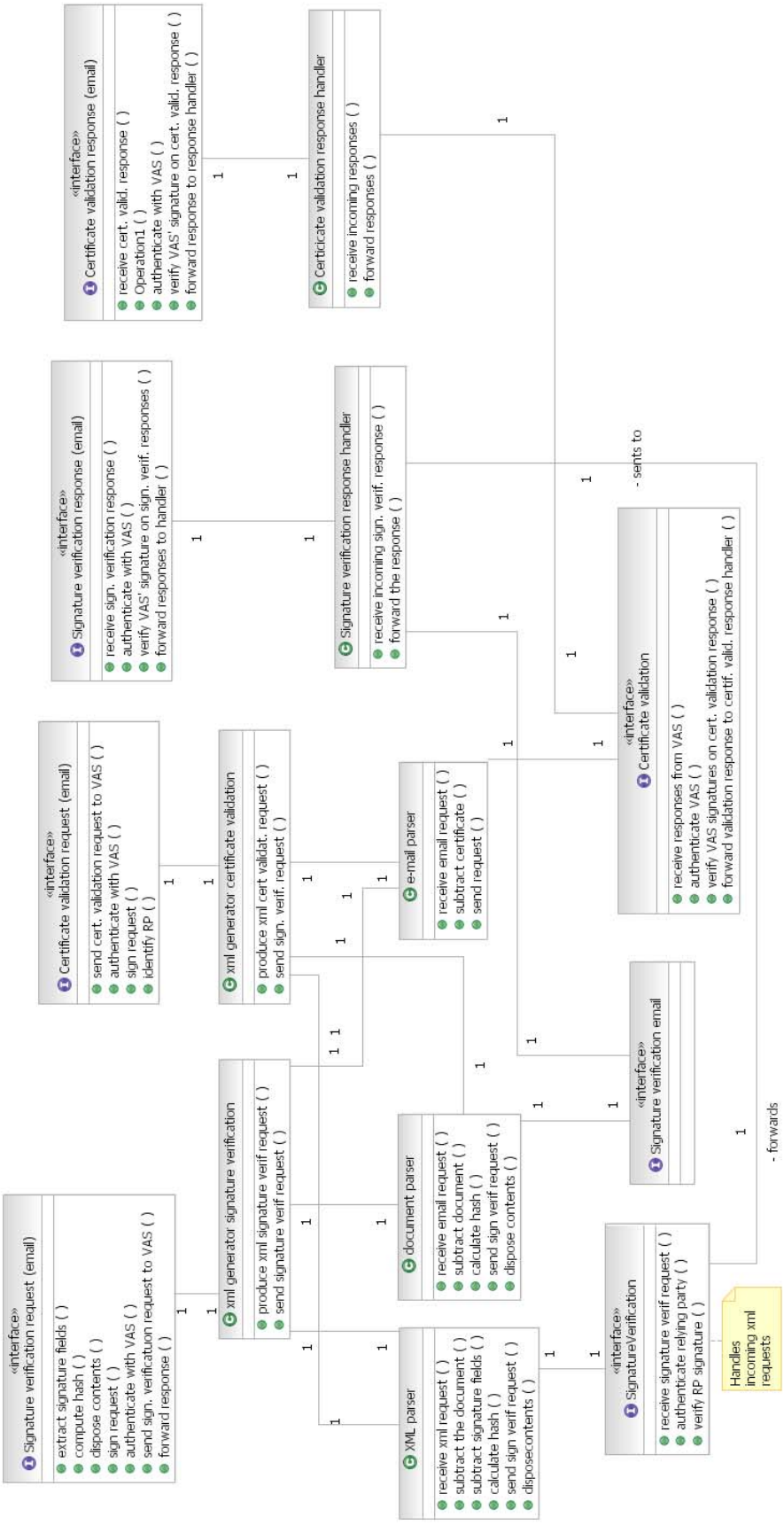

Fig. 16. Gateway interfaces

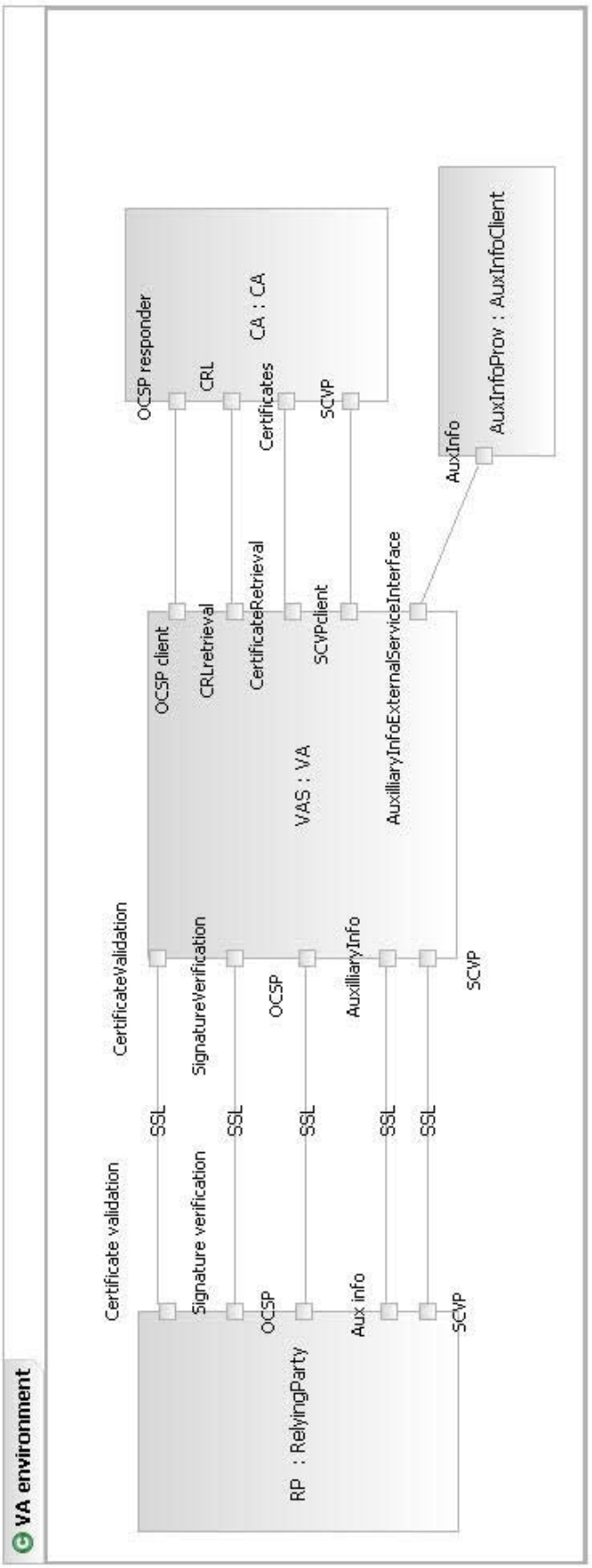

Fig. 17. VA environment composite diagram

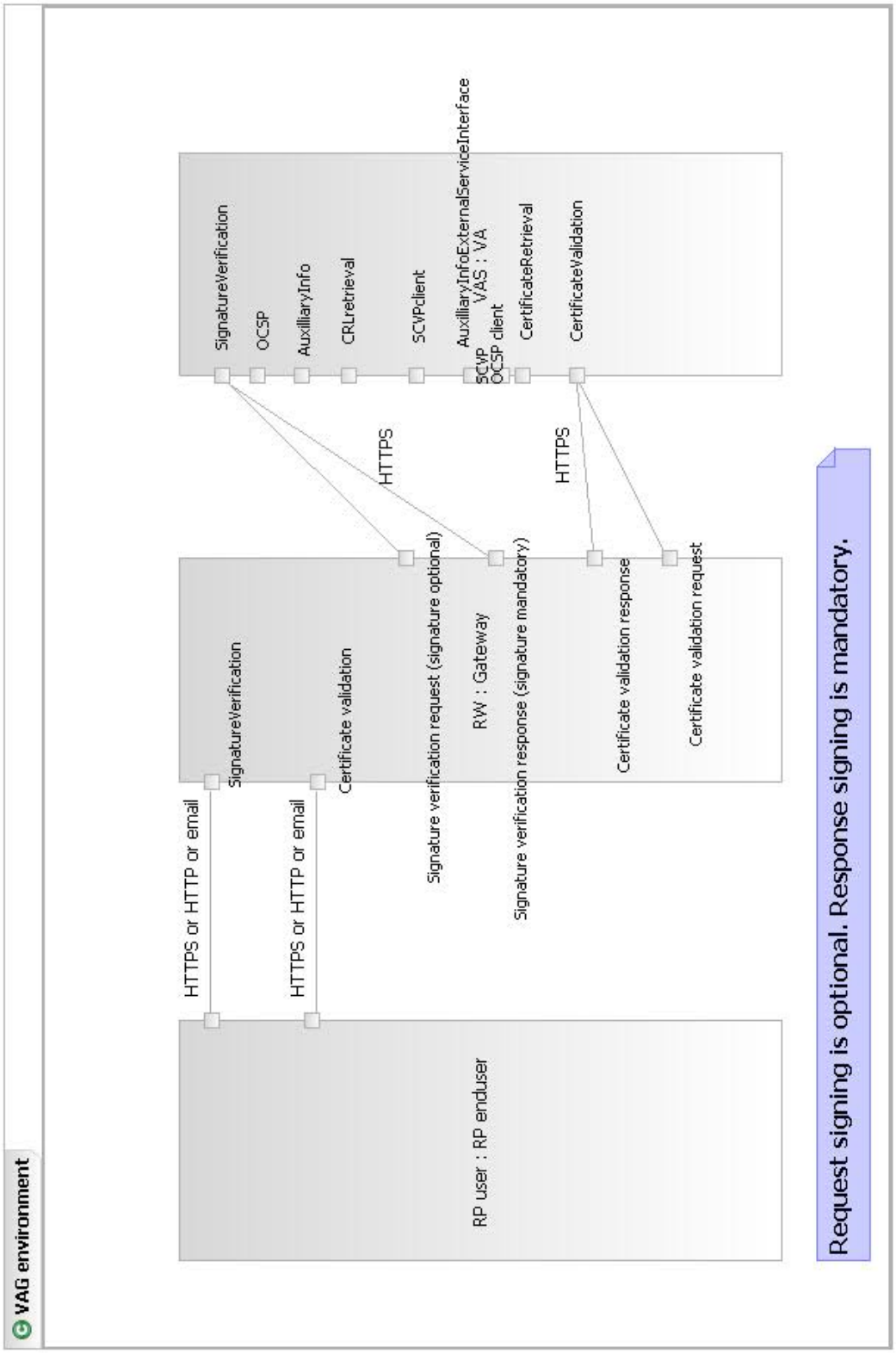

Fig. 18. VA Gateway environment composite diagram

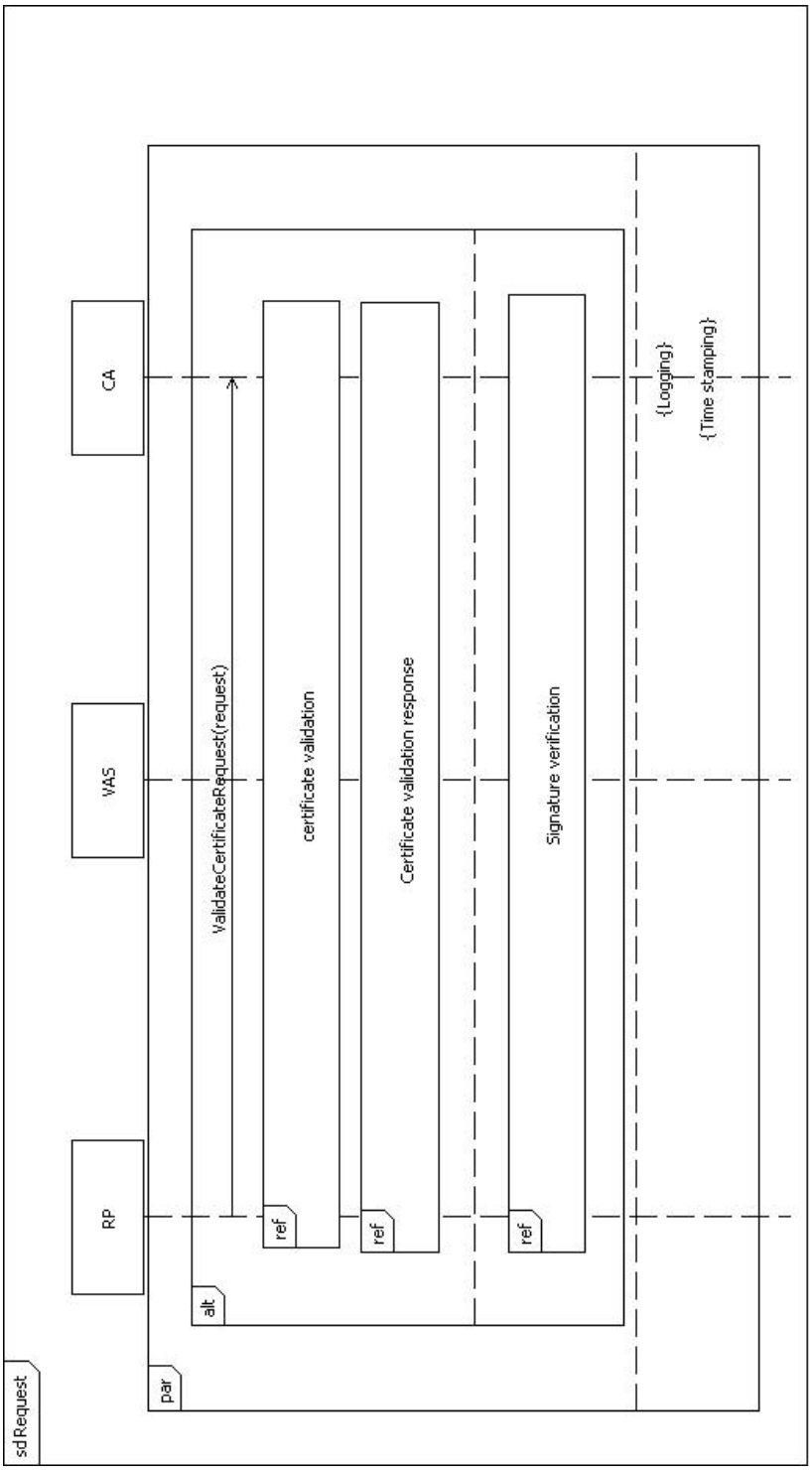

Fig. 19. Request types sequence diagram

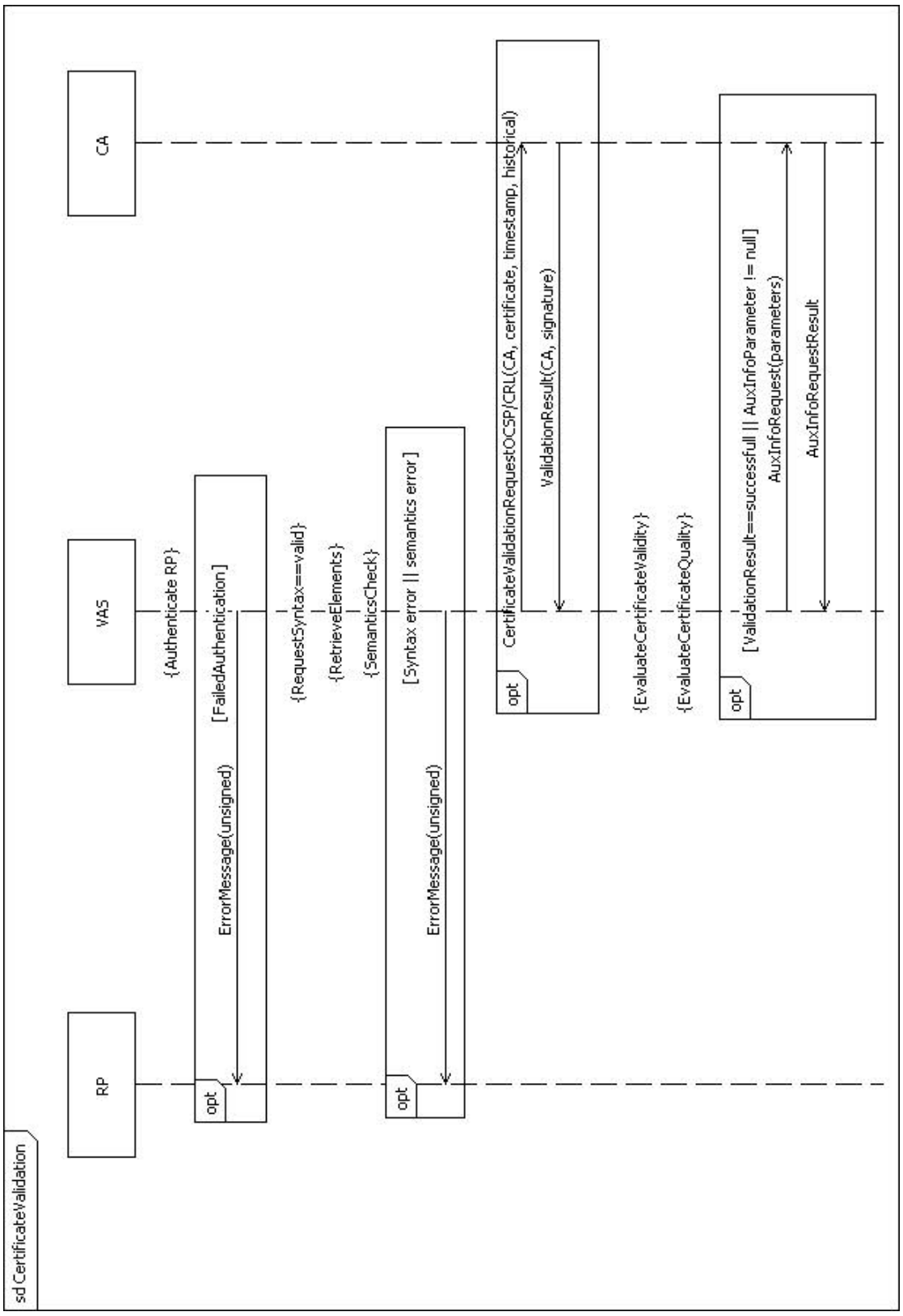

Fig. 20. Certificate verification request sequence diagram

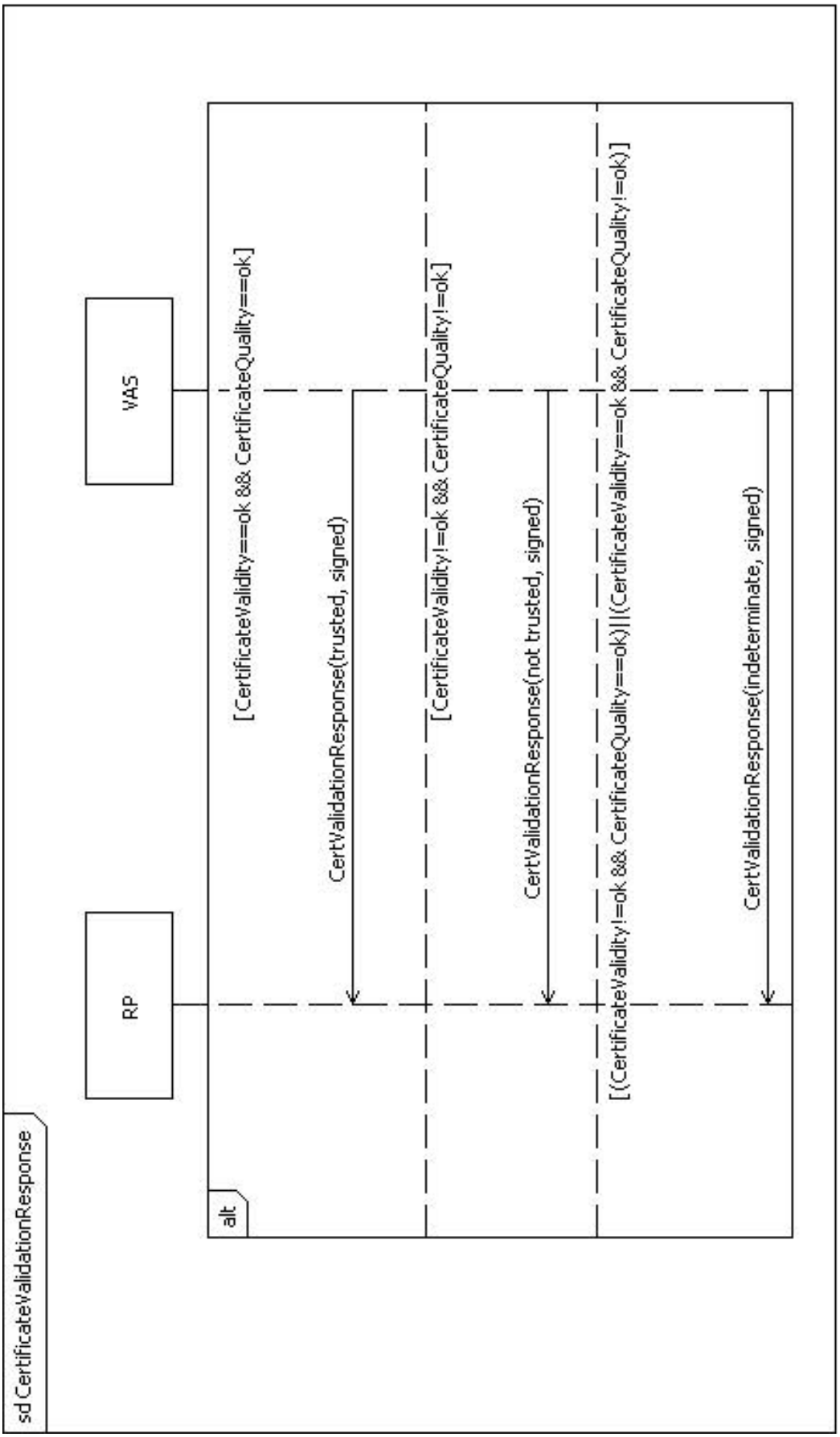

Fig. 21. Certificate verification response sequence diagram

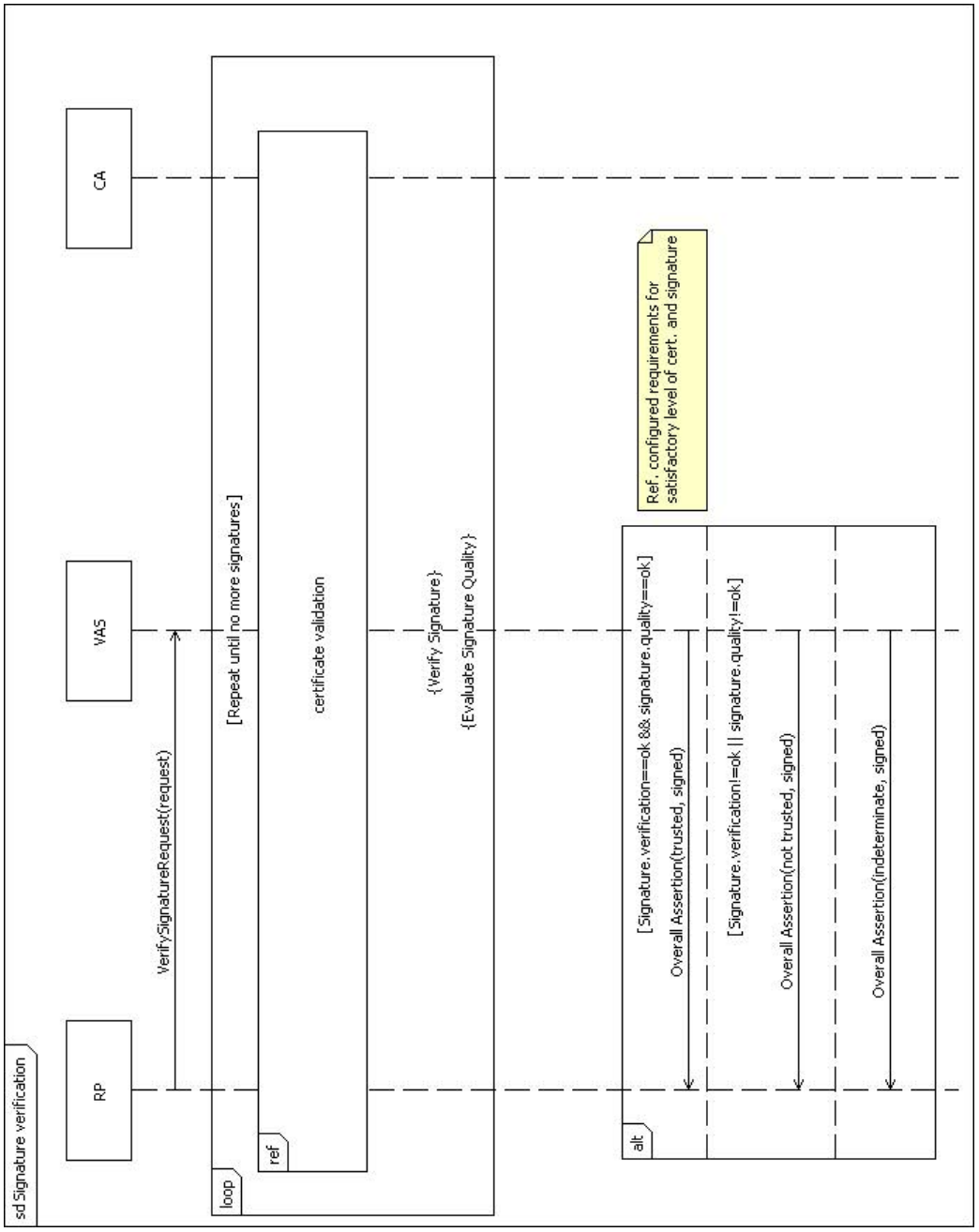

Fig. 22. Signature validation sequence diagram

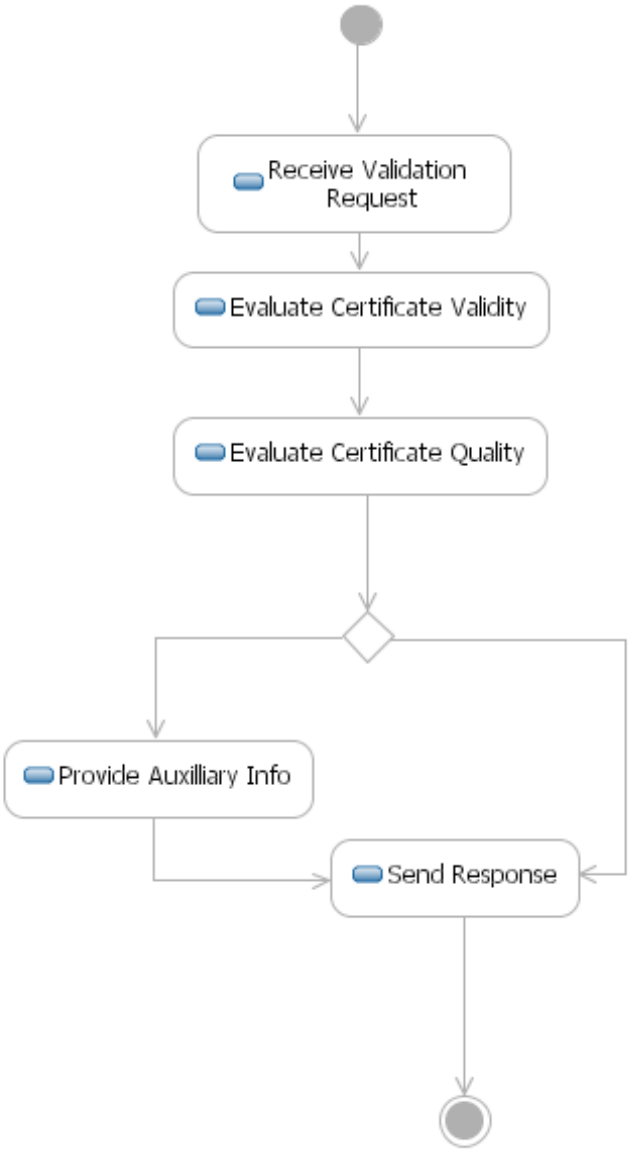

Fig. 23. Certificate verification activity diagram

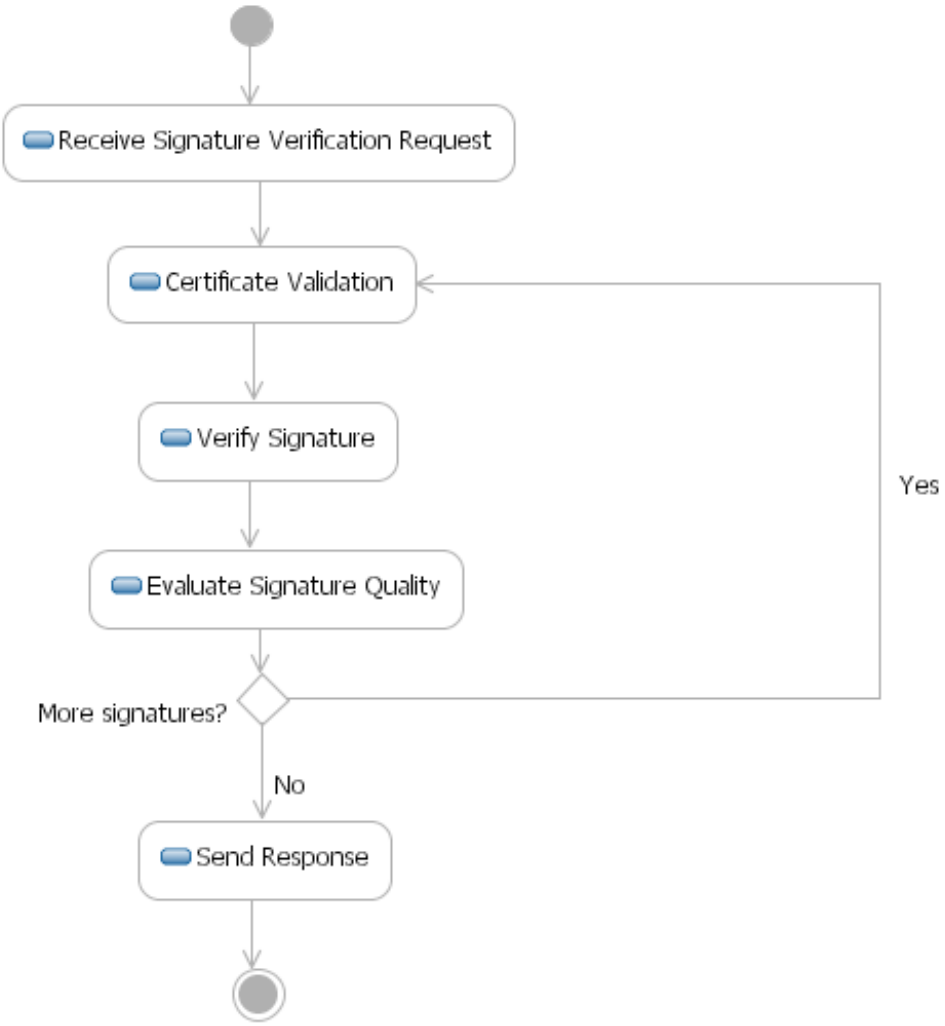

Fig. 24. Signature validation activity diagram

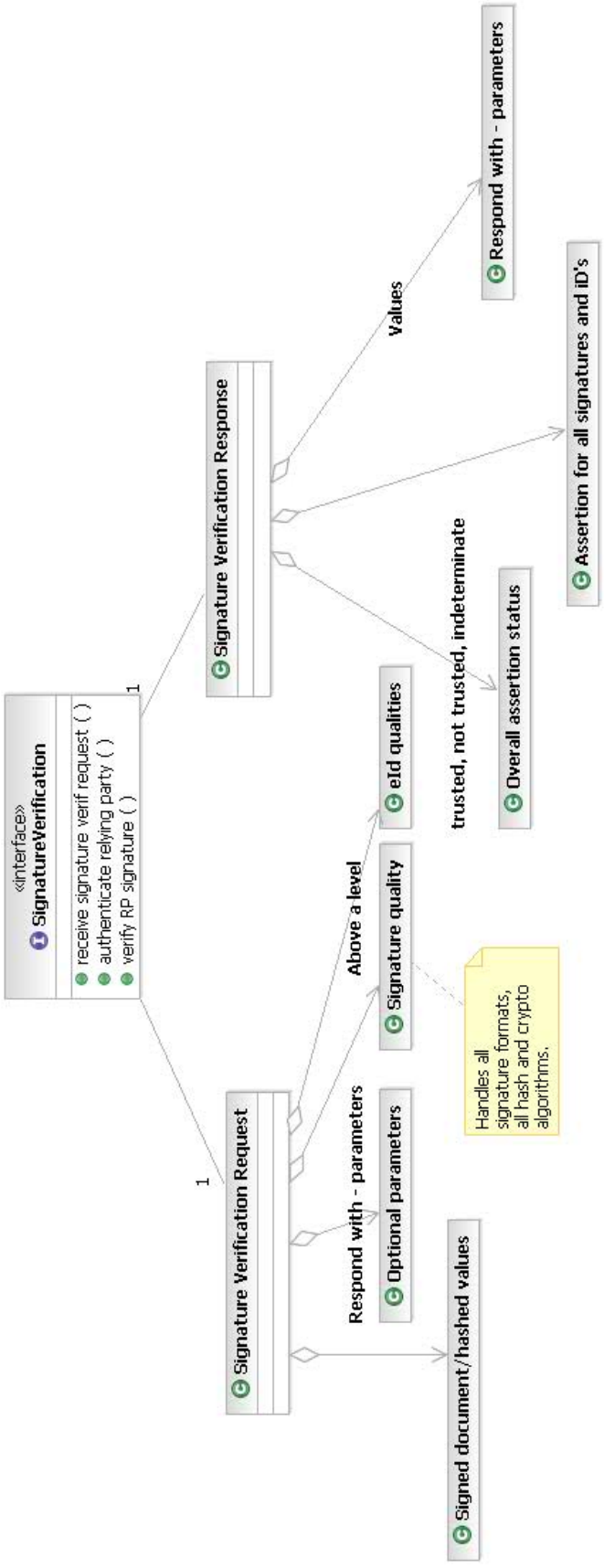

Fig. 25. Signature validation data format

Appendix 3: The quality models of the VA

The structure and definitions of the quality models for the VA system are provided below. The models are developed in "Rational Software Modeller 6.0.1." tool. The total quality is decomposed into the three quality attributes, as shown by Fig. 26. The total quality is defined as "The totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs" [2].

The "Security" quality attribute of the VA system is decomposed as shown by Fig. 27. The security attribute definition was based on [4]: *"The capability of the software product to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them."*. The security rating was:

$$\sum_{i}^{I} \sum_{j}^{J} \frac{Size(i) \cdot Op(j)}{Size(i) \cdot Op(j) + \forall i(SBL_i \cdot W)}$$

where:

- SBL is Security Breach Level
- W is weight
- i is the index for the component size
- j is the index for the operation
- I is the total number of the components
- J is the total number of the operations.

Most of the security sub-characteristics were assigned an internal and an external measure, based on [3].

For "Access auditability":

- Internal: "How complete is the implementation of access login instances considering the auditability requirements?" (nr of information recording access log confirmed in review)/(nr of information requiring access log) [3, Part 3, p. 7]
- External: "How complete is the audit trail concerning user access to the system and data" (nr of user accesses recorded)/(nr of user accesses done) [3, Part 2, p. 22-23]

For "Access controllability":

- Internal: "How complete is the detection of user access to the system" (nr of incorrect/illegal operations detected)/(nr of incorrect/illegal operations to be detected) [3, Part 3, p. 7]
- External: "How complete is the detection of user access to the system" (nr of incorrect/illegal operations detected)/(nr of incorrect/illegal operations anticipated in the specification) [3, Part 2, p. 22-23]

For "Data corruption prevention":

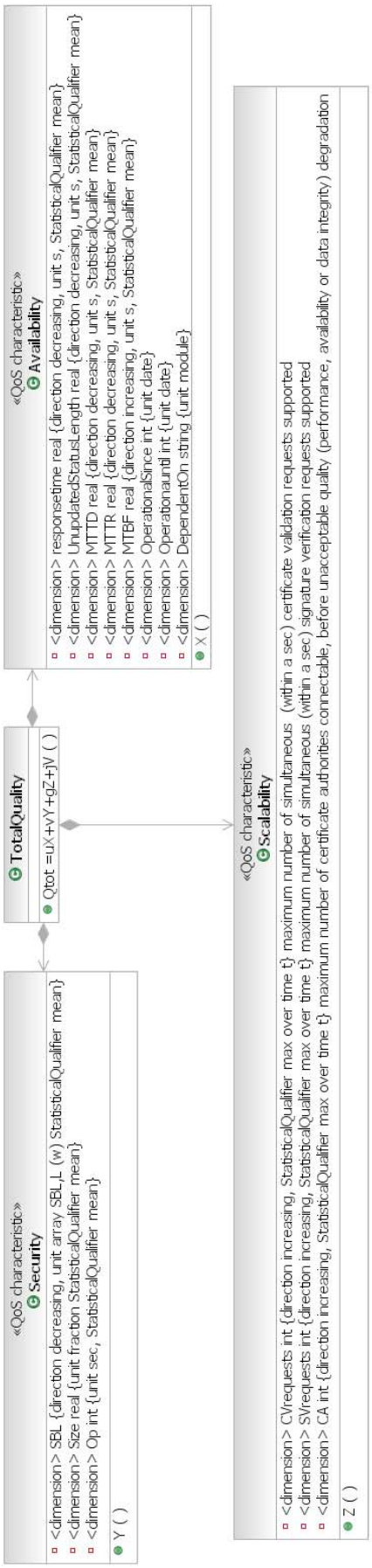

Fig. 26. Decomposition of the total quality of the VA, into attributes

- Internal: "The ratio of implemented data corruption prevention in operations to the total number of operations capable of corrupting data" [3, Part 3, p. 7]
- External: "How often do fatal data corruptions occur?" Frequency of data corruption events = 1-((nr of major data corruption events)/(nr of test cases tried to occur data corruption events)) [3, Part 2, p. 22-23]

For data encryption: Internal only: "What is the data encryption ratio?" (nr of data items implementing data encryption/detection facility)/(nr of data items requiring data encryption/decryption facility) [3, Part 2 and Part 3, p. 7]

The "Availability" quality attribute of the VA system is decomposed as shown by Fig. 28. The availability attribute definition was "The degree to which the VA system and its parts are operational and accessible when required for use". Uptime, as well as service continuity had to be taken into account, and downtime was defined as "incorrect operation time (planned or unplanned), or lasting of a failure". The availability rating was:

$$Availability = \frac{uptime}{uptime + downtime}$$

The measures of the sub-characteristics under availability were as follows:

- Mean down time: "How long is usually system down?" (Total down time)/(nr. of observed breakdowns) [3, Part 2, p. 34]
- Recovery: "How long will it take to recover if system went down?" Mean time to recover = (total sum of time to recover)/(number of recovery attempts) [3, Part 2, p. 35]
- Restorability: "How is the product capable to restore in defined cases?" (nr. of cases capable to restore)/ (required nr. of cases capable to restore) [3, Part 3, p. 13]
- Restore effectiveness: "How effective will the restoration process be?" (nr. of restore cases meeting the target time)/(number of restore cases which shall meet target restore time) [3, Part 3, p. 13]

The "Scalability" quality attribute of the VA system is decomposed as shown by Fig. 29. The scalability attribute definition was "The ability of the system to support rapid and extensive variations in the number of users, without requiring any changes" [20]. The scalability rating was two-fold:

- "The maximum number of simultaneous inquiries (Certificate verification + signature validation requests) supported by the system (without unacceptable degradation of quality of service level), before any changes need to be made (distinguishing between http and ssl related transaction loads and taking into account number of internal transactions)."
- "The maximum number of certificate authorities supportable by the system (without unacceptable degradation of quality of service level), before any changes need to be made."

The measures of the sub-characteristics under scalability were as follows:

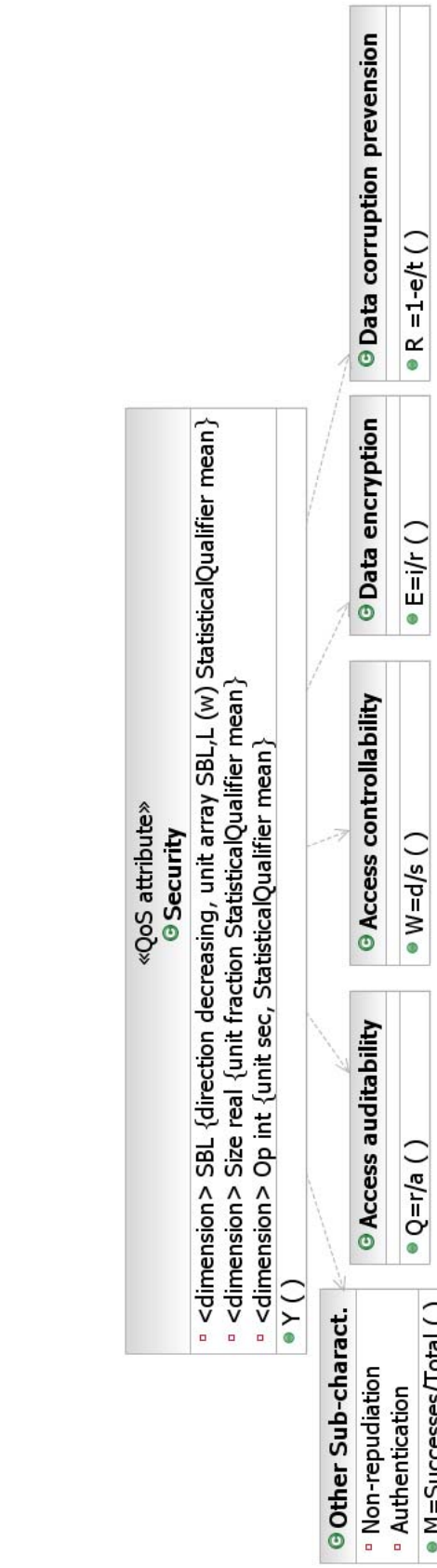

Fig. 27. Decomposition of the security quality attribute for the VA, into sub-characteristics

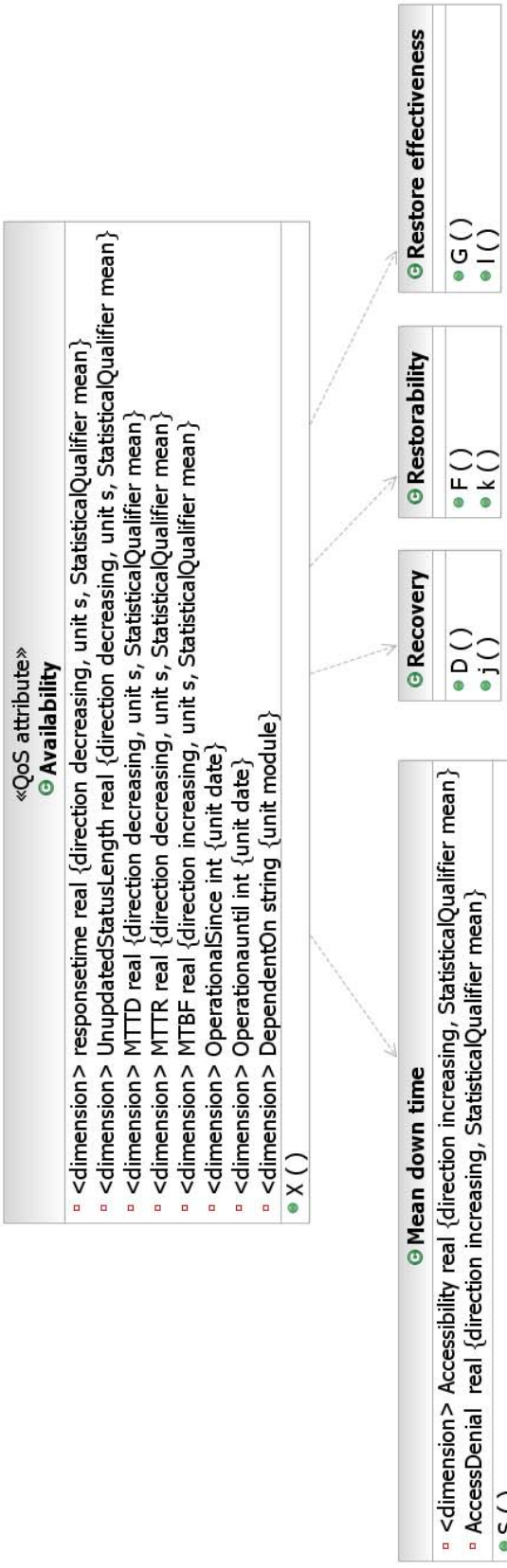

Fig. 28. Decomposition of the availability quality attribute for the VA, into sub-characteristics

- Response time: "The estimated time to complete a specified task." [3, Part 3, p. 23]
- Throughput time: "The estimated number of tasks that can be performed over a unit of time." [3, Part 3, p. 23]
- Turnaround time: "The estimated time to complete a group of related tasks as a job lot." [3, Part 3, p. 23]

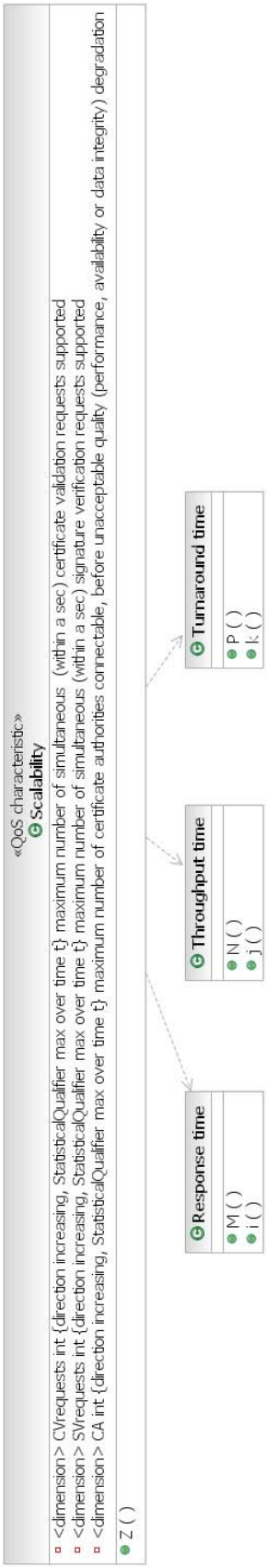

Fig. 29. Decomposition of the scalability quality attribute for the VA, into sub-characteristics

Appendix 4: The conceptual model of the VA

The conceptual model of the VA is displayed in Fig. 30. The conceptual model merges the quality and the design models, prior to its transformation to the generic DV, which is then instantiated into the respective quality attribute specific DVs. The conceptual model was actively used in relation to model fitting and particularly when applying the prediction models, in order to identify relationships between the different parts of the prediction models of the VA system.

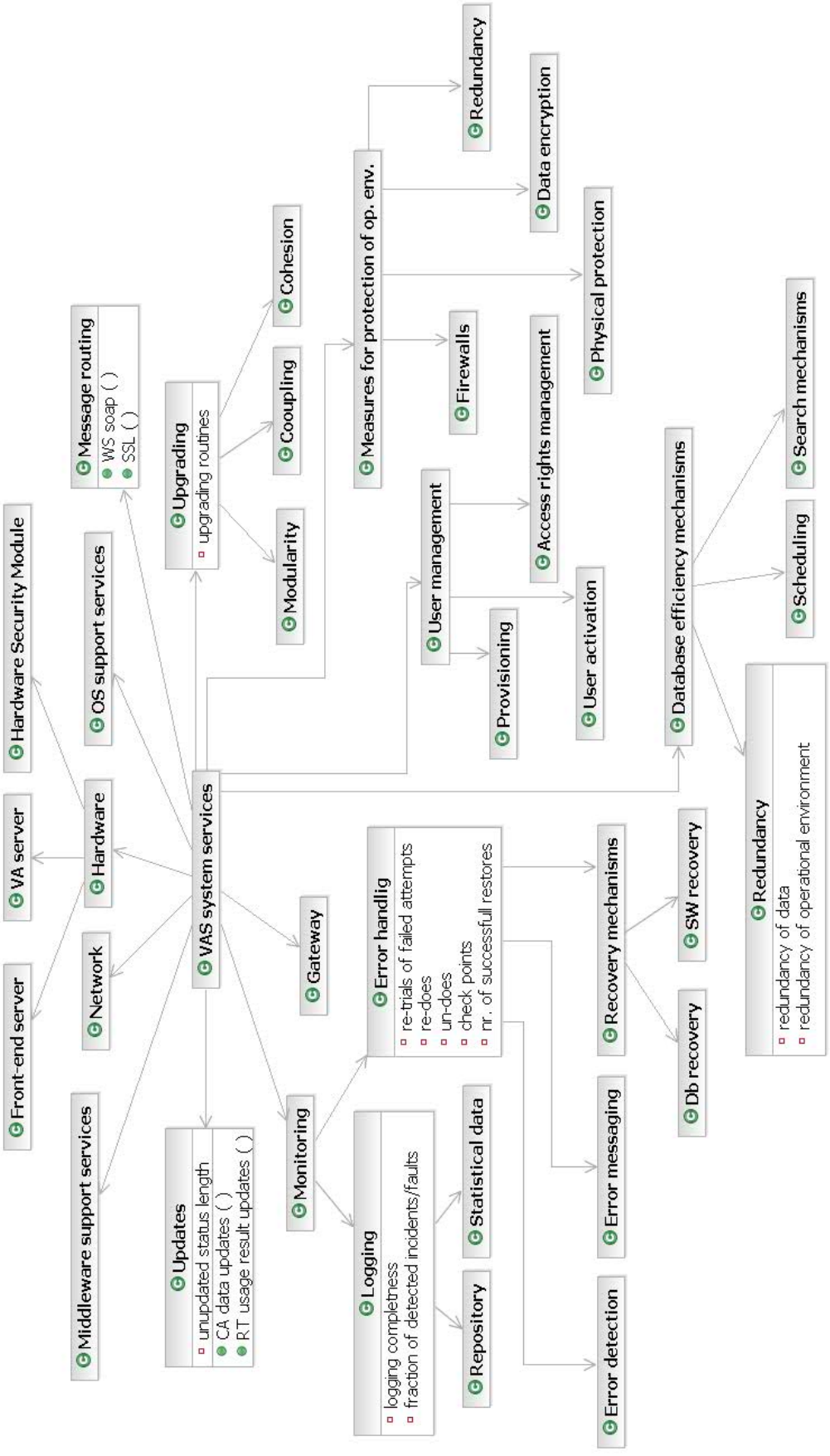

Fig. 30. The conceptual model of the VA

Appendix 5: Structure of the DVs

Since the parameter values are confidential, we only provide the structure of the quality attribute specific DVs (same structure for all the three quality attribute specific DV of the VA, as shown in Fig. 31) and the total quality DV (shown in Fig. 32). The former is an instantiation of the conceptual model, while the latter is an instantiation of the top two levels of the quality model. On both DVs shown in Figures 31 and 32, the parameter values are initialized due to their confidentiality.

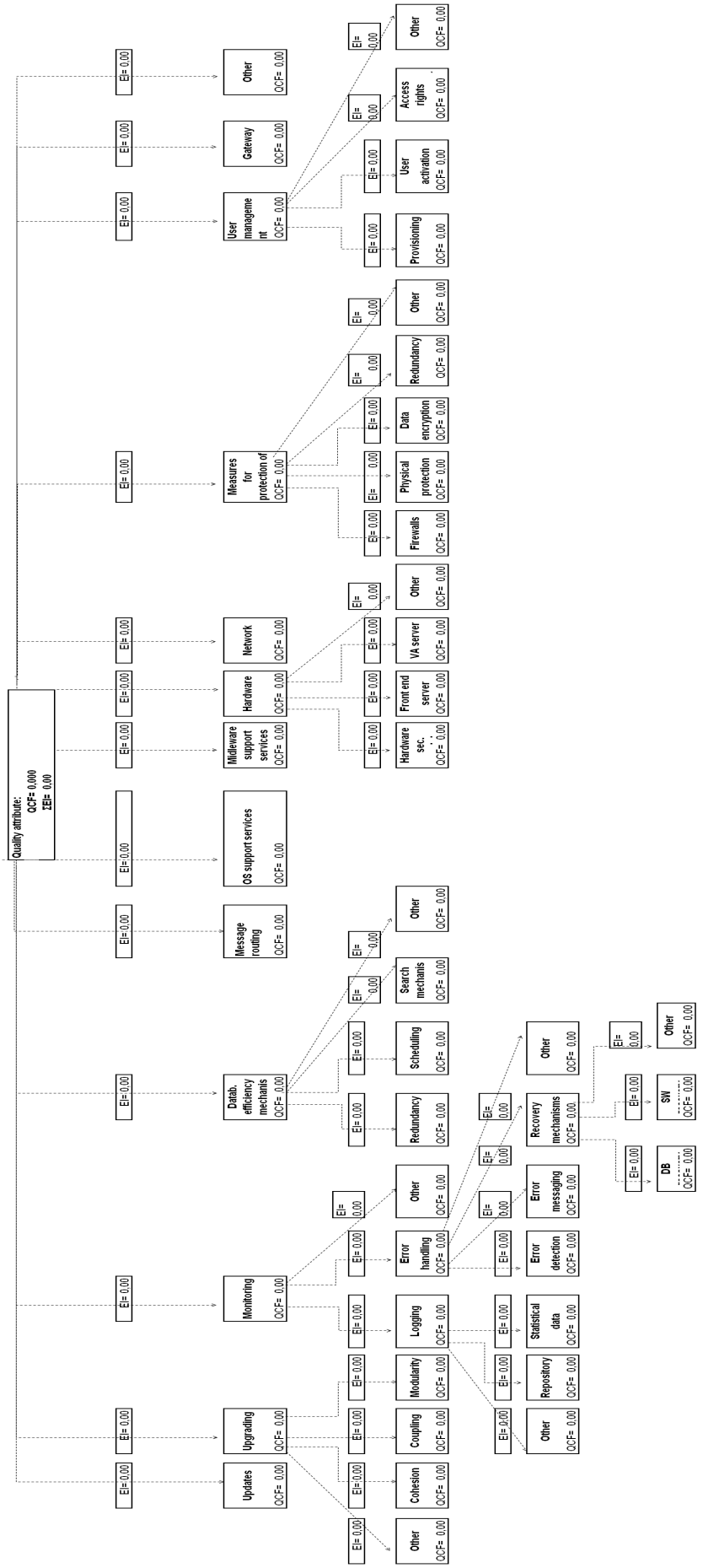

Fig. 31. Structure of a quality attribute specific DV of the VA

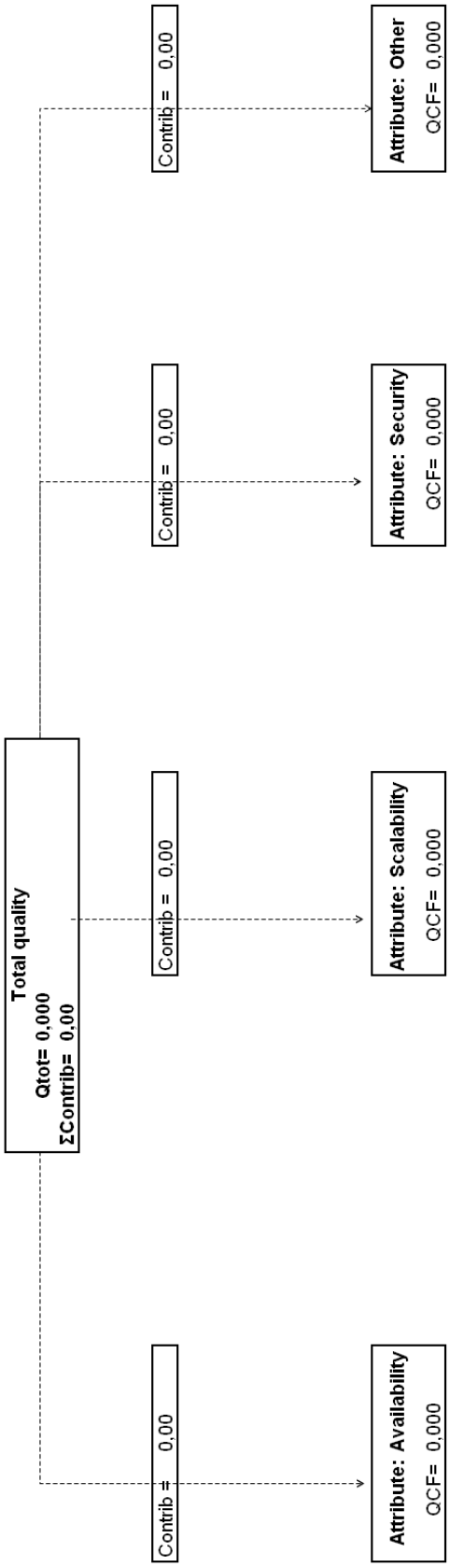

Fig. 32. Structure of the total quality DV of the VA

Appendix 6: Schema for documentation of results of the change simulation

The results of simulation of the 14 specified changes were documented in a table, shown by Figures 33, 34 and 35. All the updated parameters are displayed here. The simulation took place prior to the thought experiment, was performed by the analyst (who did not perform the thought experiment), stored by an additional (independent) participant, and kept unrevealed until completion of the thought experiment. Each change was simulated independently. The actual values are not shown due to their confidentiality. The letters are displayed instead and their (re)use is coincidental. As a result, it is here invisible whether the updates involved increase or decrease of the original (unmarked) parameter values, and to what degree. The marked letters represent the updated parameter values.

In the case of change 1, three design models were modified, as shown by Figures 36, 37 and 38, respectively.

In the case of change 2, two design models were modified, as shown by Figures 39 and 40, respectively.

In the case of change 3, one sequence diagram was modified, as shown by Fig. 41. Change 3 also involved a modification of the diagrams shown by Figures 14 and 17, in form of a duplication of all the elements included in these two figures.

In the case of change 4, one design model was modified, as shown by Fig. 42.

The overall changes (5-14) did not require modifications of the design models.

Change number	Change description.	Availability (Current QCF=X)	Scalability (Current QCF=Z)	Security (Current QCF=Y)
1	Split signature verif. component into two redundant components, with load balancing	QCF changed on nodes: Modularity a->a' Redundancy b->b' Middleware support servies c->c' Sw recovery d->d' EI changed on nodes: => Avail QCF = X'	QCF changed on nodes: Updates e->e' Modularity a->a' message routing f->f' EI changed on nodes: Measures for prot of op env g->g' Message routing h->h' => Scal QCF = Z'	Unchanged.
2	Merge CV and SV interfaces (not components)	QCF changed on nodes: Modularity a->a' Scheduling i->i' Middleware supp services c->c' Message routing f->f' Updates j->j' EI changed on nodes: none => Avail QCF = X'	QCF changed on nodes: Updates j->j' Modularity k->k' Coupling l->l' Cohesion m->m' Scheduling n->n' Middleware supp services p->p' EI changed on nodes: none => Scal QCF = Z'	QCF changed on nodes: Access rights a->a' Data encryption b->b' Message routing c->c' Error detection d->d' Error messaging e->e' Modularity f->f' EI changed on nodes: none => Sec QCF = Y'
3	Redundant VAS with external workload balancing	QCF changed on nodes: Sw recovery a->a' Modularity b->b' Redundancy c->c' Front end server d->d'	QCF changed on nodes: Redundancy (meas. for prot. of op env) a->a' VA server b->b' Message routing c->c'	QCF changed on nodes: Measures for prot of op env (redundancy) a->a' Modularity b->b'

Fig. 33. Change simulation table - Part 1

Change number	Change description.	Availability (Current QCF=X)	Scalability (Current QCF=Z)	Security (Current QCF=Y)
		Monitoring e->e' Middleware supp services f->f' EI changed on nodes: Hardware g->g' Middleware support serv.h-> h' Upgrading i->i' => Avail QCF =X'	Scheduling d->d' Datab. redundancy e->e' SW recovery f->f' Modularity g->g' Coupling h->h' Cohesion i->I' Updates j->j' EI changed on nodes: Hardware k->k' Middleware support services l->l' Upgrading m->m' => Scal QCF = Z'	EI changed on nodes: Hardware c->c' Upgrading d->d' Measures for prot. of op env e->e' => Sec QCF = Y'
4	Average size (nr. of data elements and data amount) of SV requests increases by 100%.	QCF changed on nodes: Updates a->a' Db recovery b->b' Scheduling c->c' Message routing d->d' Network e->e' => Avail QCF = X'	QCF changed on nodes: Db recovery a->a' Scheduling b->b' Message routing c->c' Network d->d' EI changed on nodes: Network e->e' Measures for prot. of op.env. f->f' Message routing g->g' => Scal QCF = Z'	Unchanged
5	Use of gateway made mandatory.	QCF changed on nodes:	QCF changed on nodes:	QCF changed on nodes:

Fig. 34. Change simulation table - Part 2

Change number	Change description.	Availability (Current QCF=X)	Scalability (Current QCF=Z)	Security (Current QCF=Y)
		Redundancy a -> a' Coupling b->b' Cohesion c->c' Modularity d->d' EI changed on nodes: Gateway e->e' Upgrading f ->f' Monitoring g->g' Network h->h' Middleware support serv. i-> I' => Avail QCF = X'	Message routing a->a' Modularity b->b' Coupling c->c' Cohesion d->d' Gateway e->e' EI changed on nodes: Gateway f->f' Upgrading g->g' Monitoring h->h' => Scal QCF = Z'	Message routing a->a' EI changed on nodes: Measures for prot. of op.env. b->b' Upgrading c->b' Gateway d->d' => Sec QCF = Y'
6	Decrease error detection QCF for availability by 50%	X'		
7	Decrease coupling QCF for availability by 50%	X'		
8	Decrease upgrading QCF for availability by 50%	X'		
9	Increase scalability QCF modularity by 66%		Z'	
10	Increase scalability QCF upgrading by 56%		Z'	
11	Increase scalability QCF "measures for protection of operational environment" by 37%		Z'	
12	Increase security QCF "logging" by 16%			Y'
13	Increase security QCF "monitoring" by 16%			Y'
14	Increase security QCF "measures for protection of operational environment" by 5%			Y'

Fig. 35. Change simulation table - Part 3

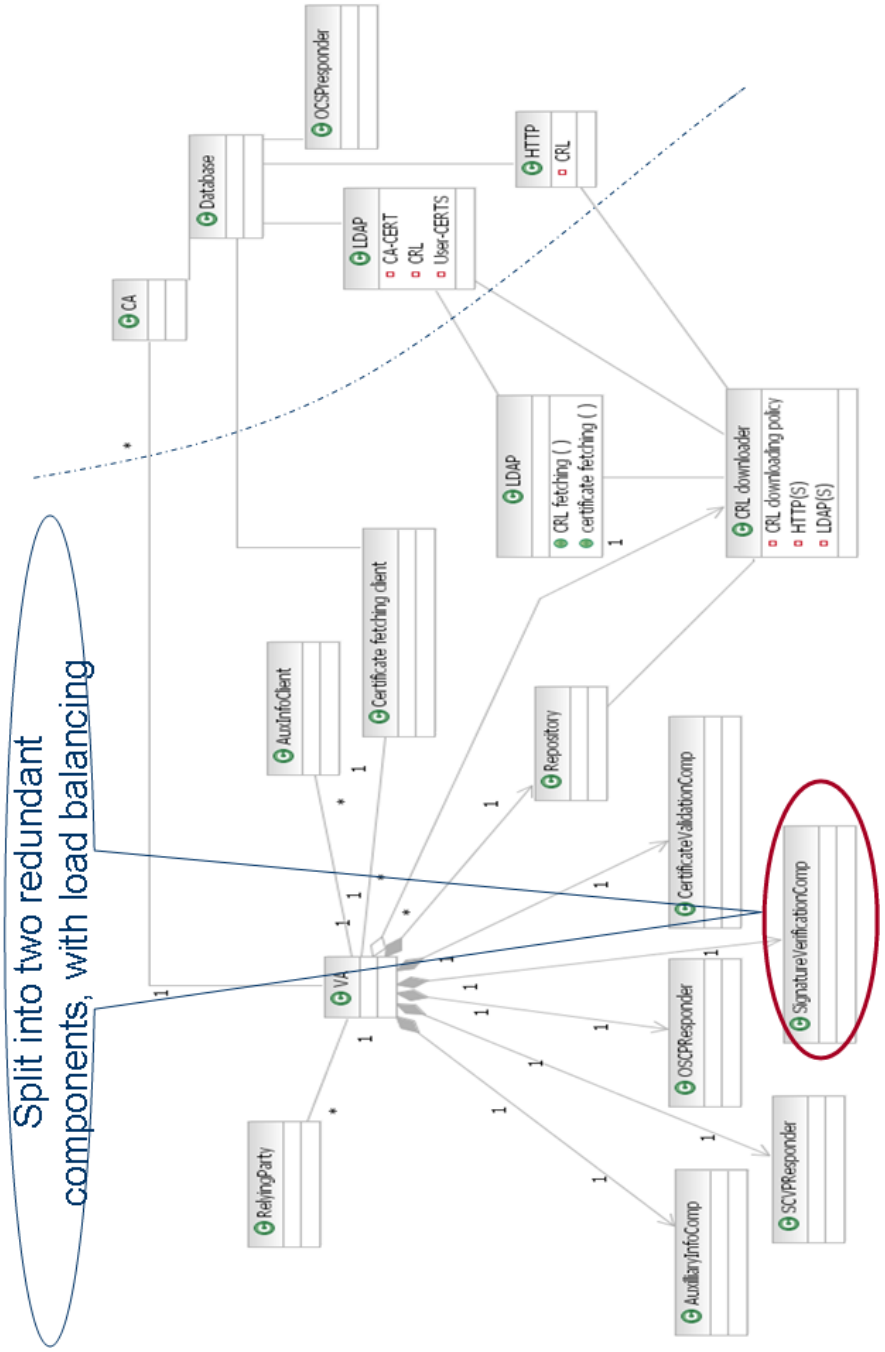

Fig. 36. Design modification due to change 1 - Part 1

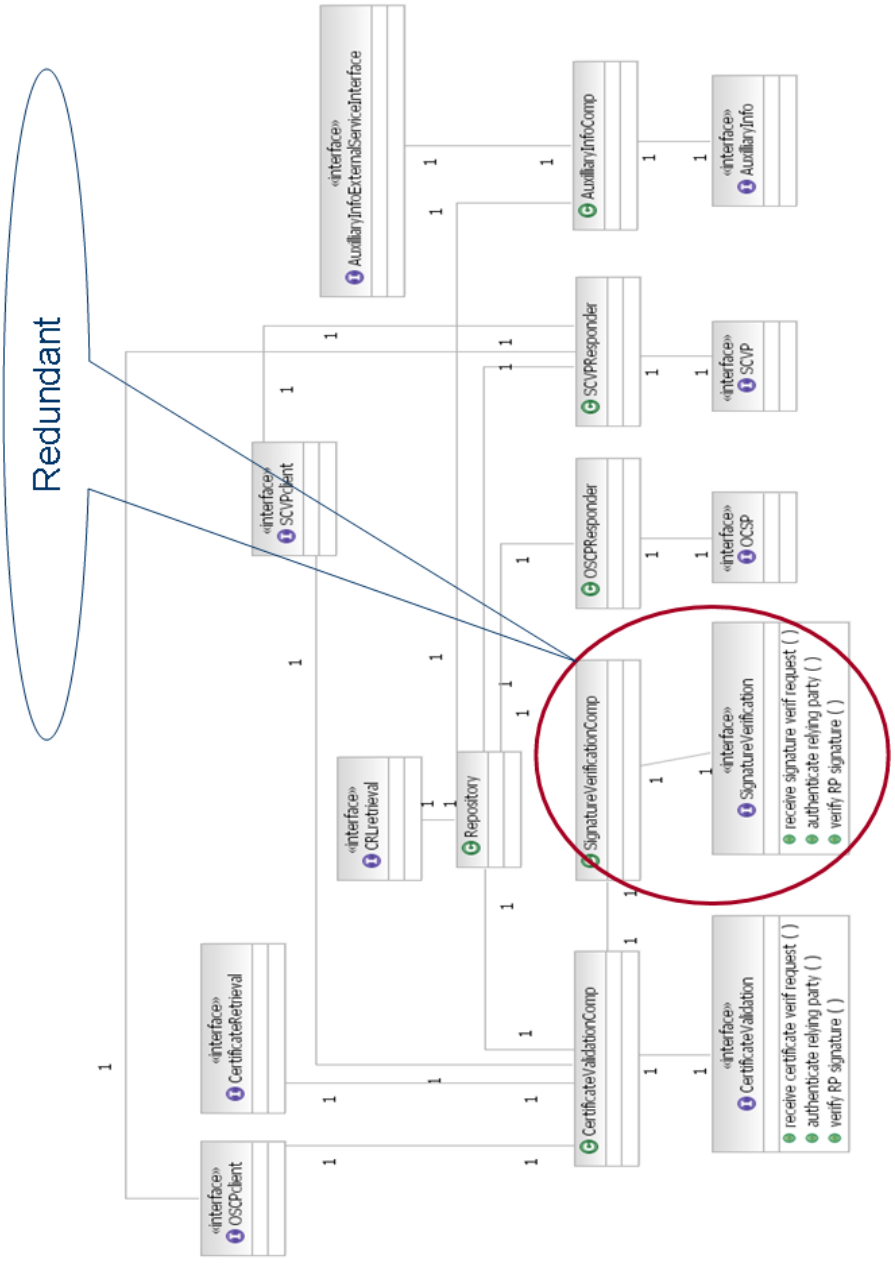

Fig. 37. Design modification due to change 1 - Part 2

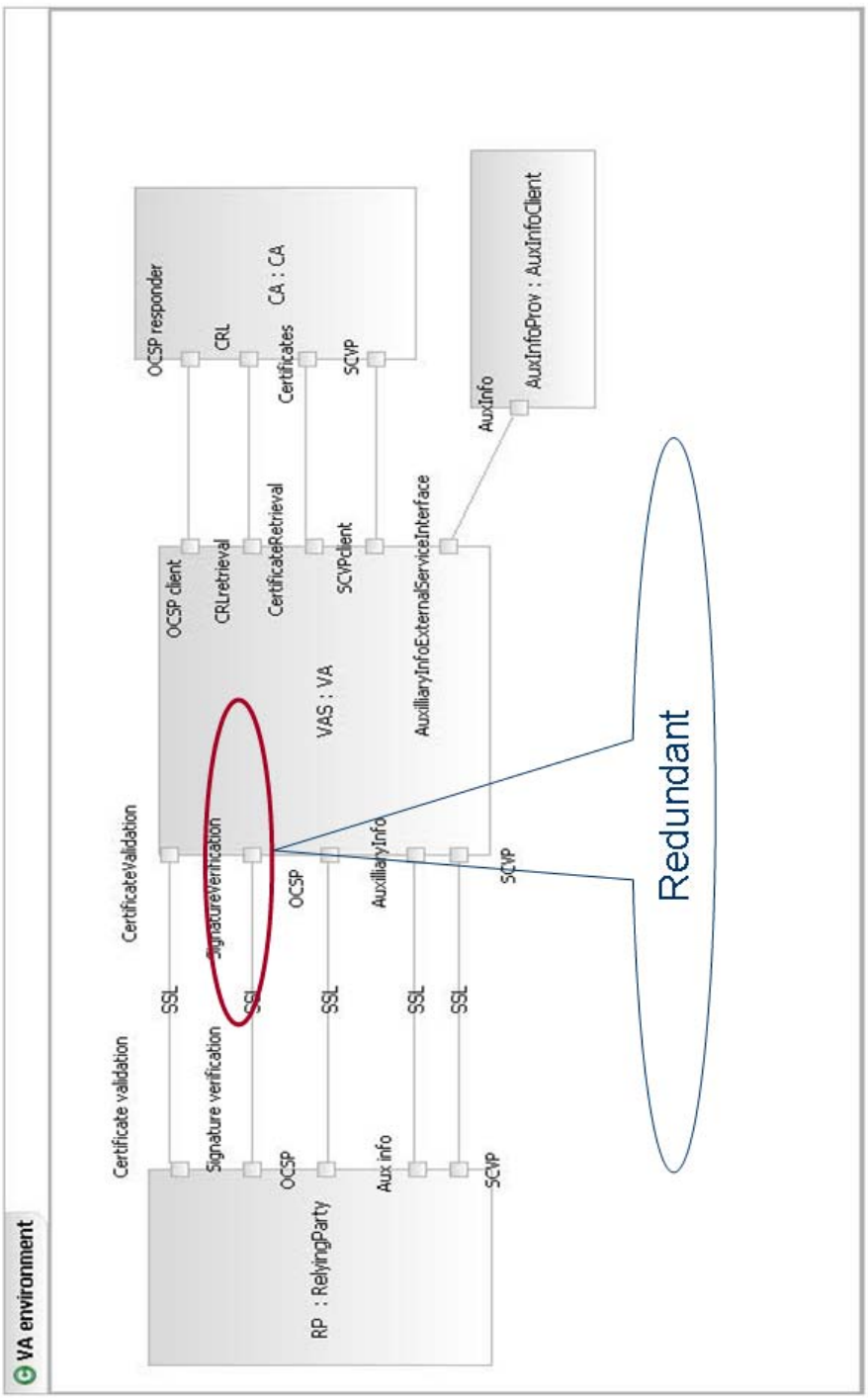

Fig. 38. Design modification due to change 1 - Part 3

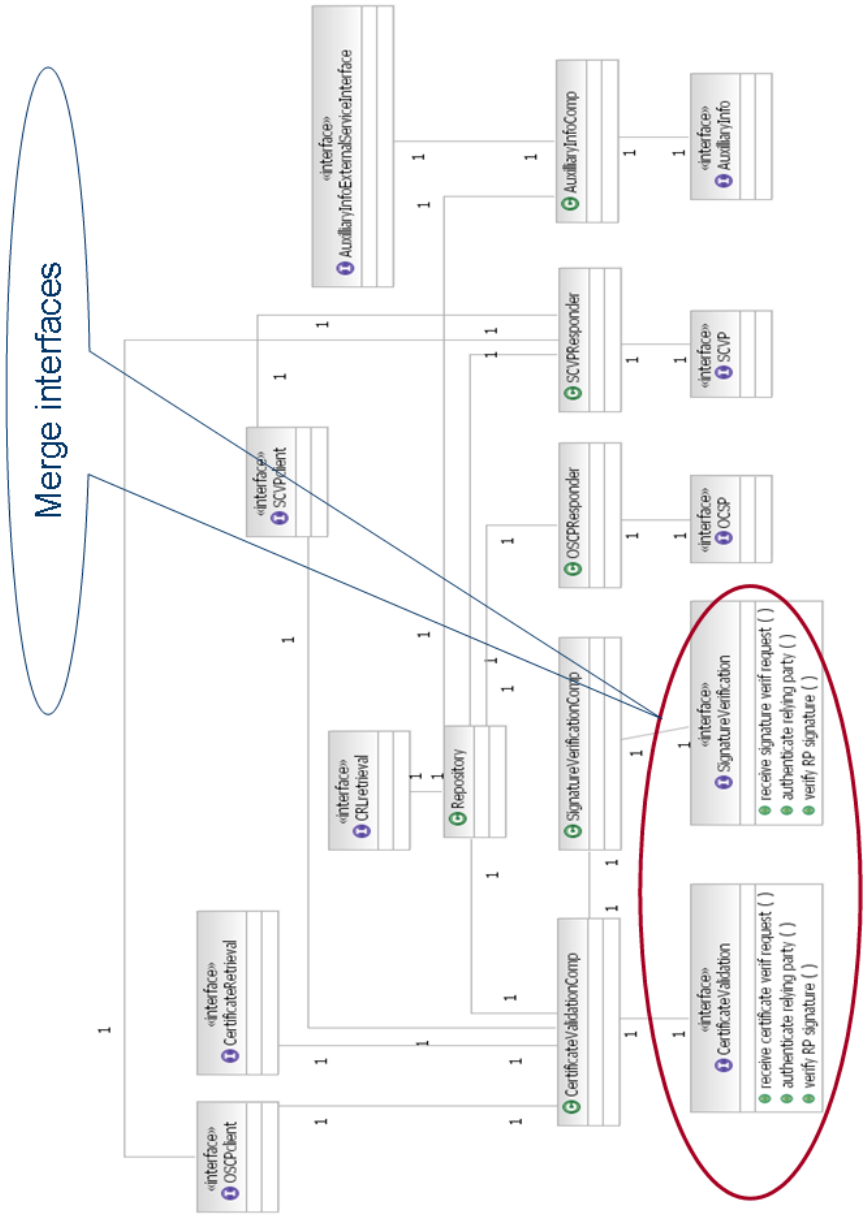

Fig. 39. Design modification due to change 2 - Part 1

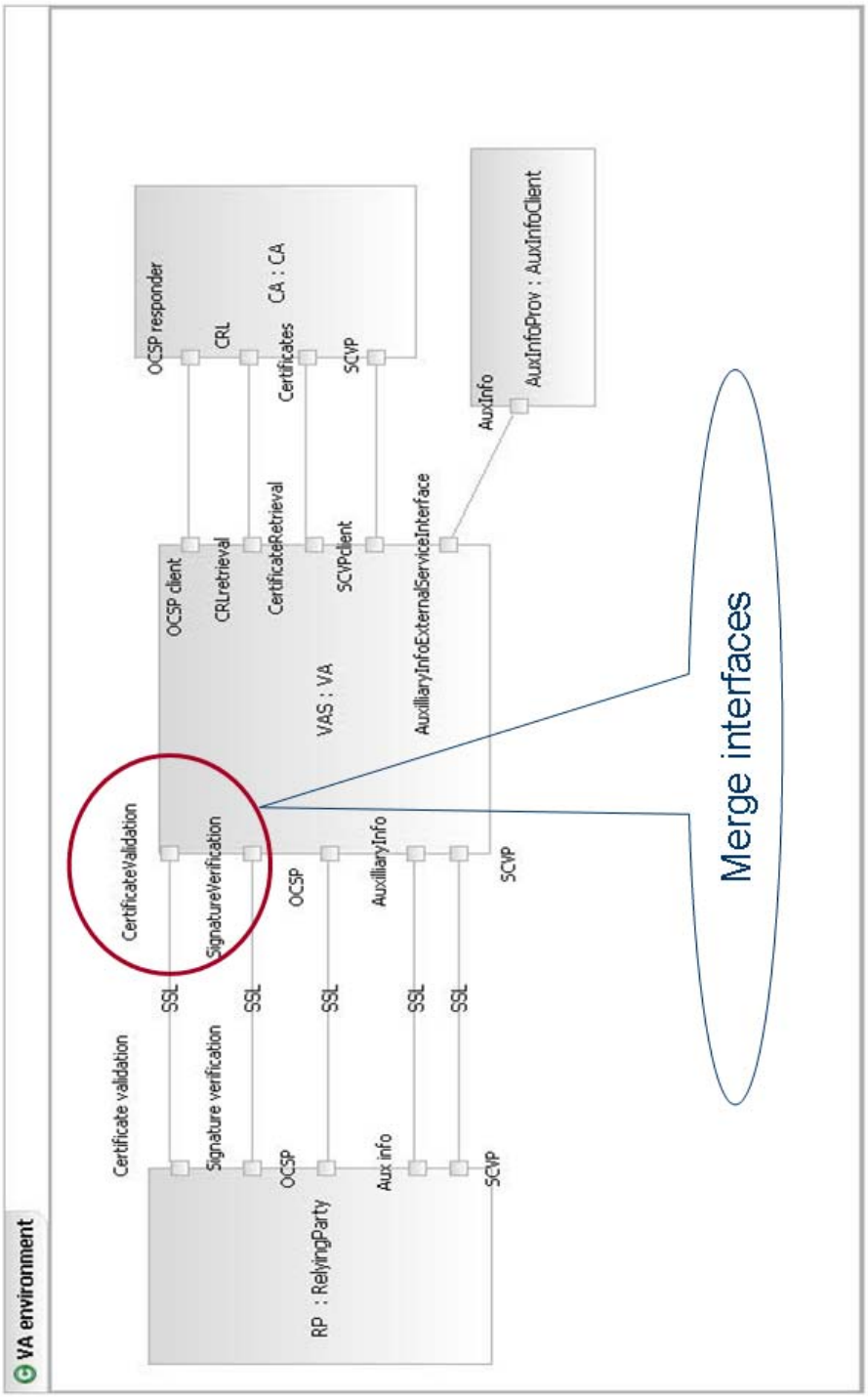

Fig. 40. Design modification due to change 2 - Part 2

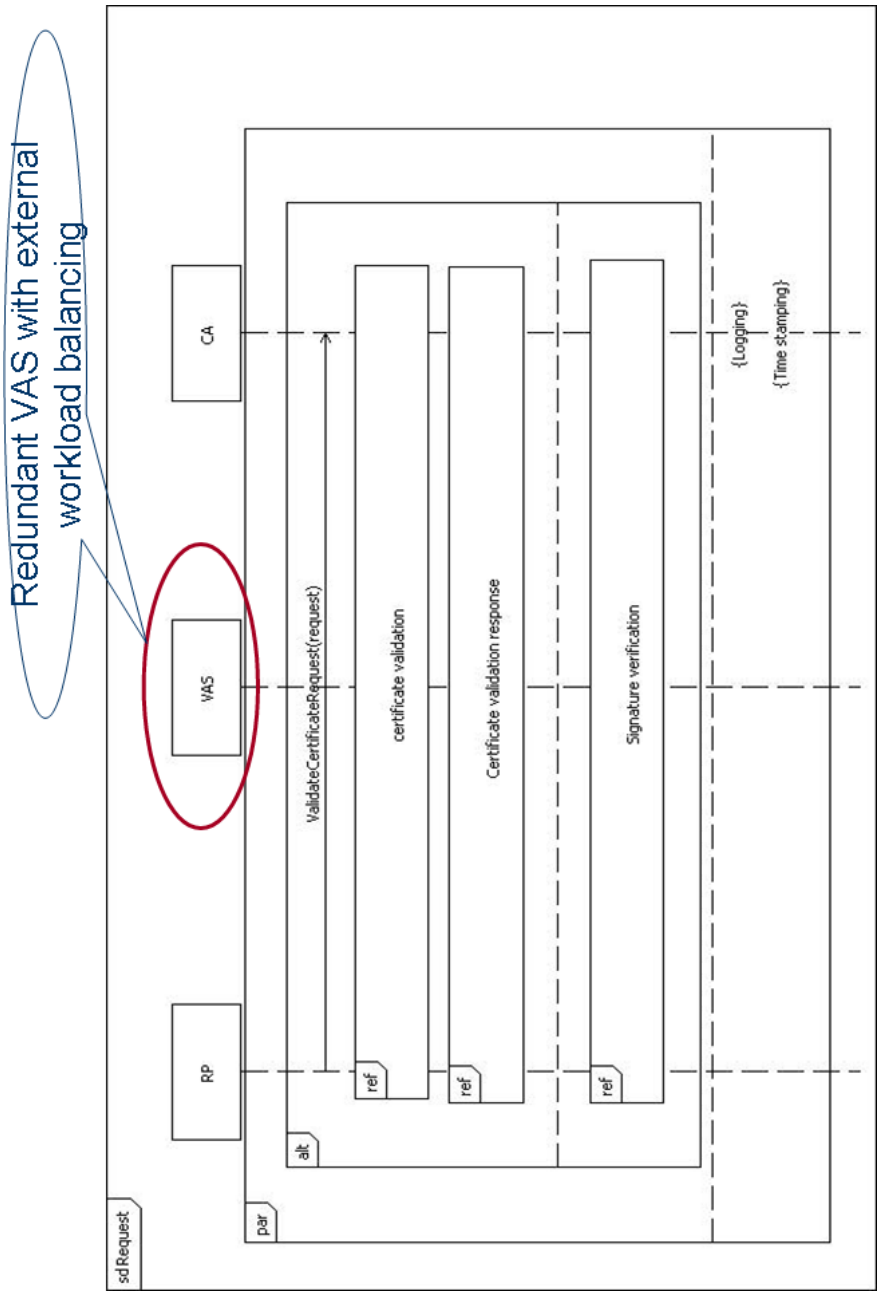

Fig. 41. Design modification due to change 3

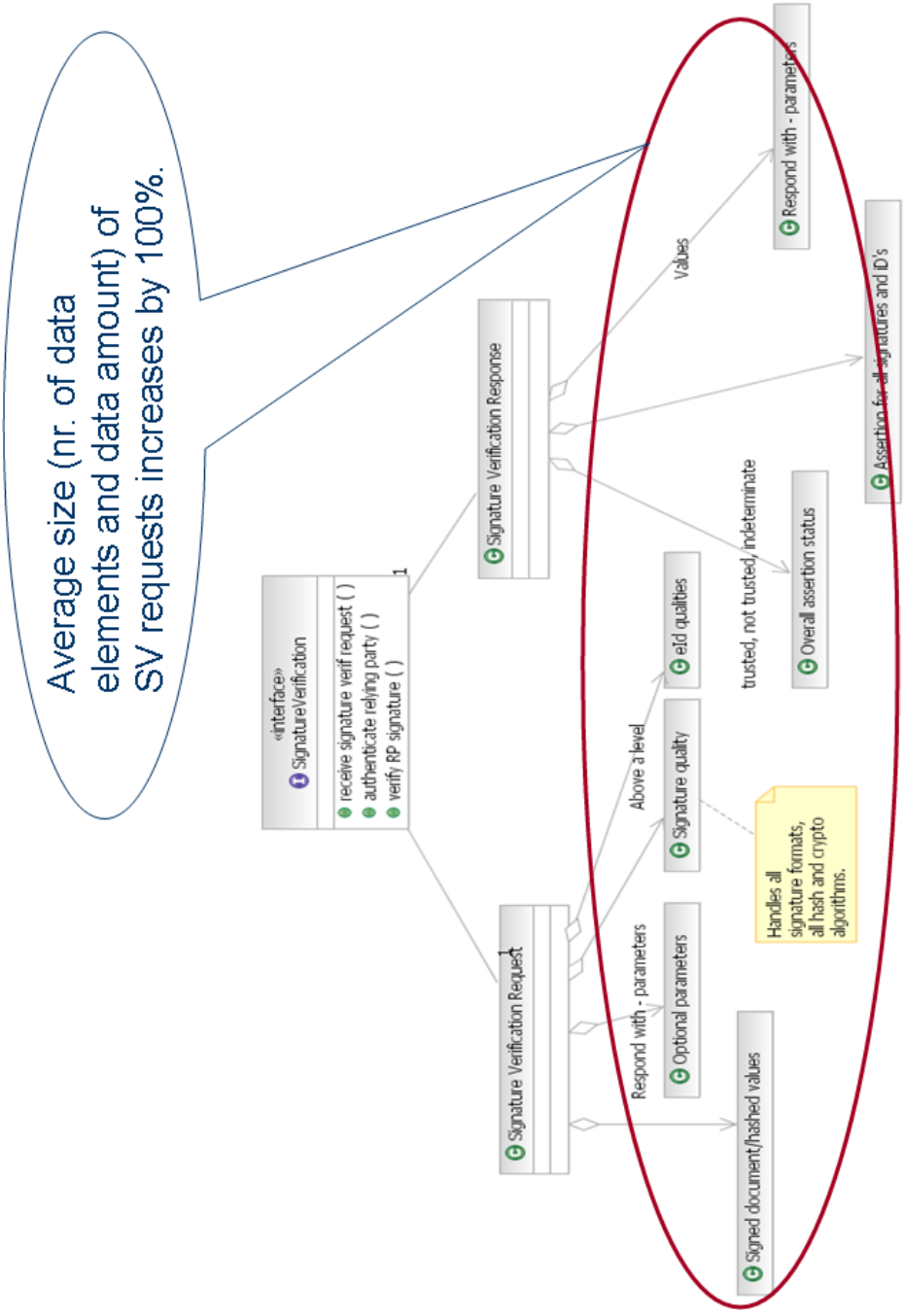

Fig. 42. Design modification due to change 4

Appendix 7: The measurement plan for VA

The measurement plan for the VA was based on the needs for evaluation, and feasibility of the measurement. The latter caused several revisions of the plan, the final version of which is shown by Figures 43, 44 and 45, respectively.

Node/ DV	Availability	Scalability	Security
Overall	Downtime over a period of time. May use a scale: - 25 % of system's total extent - 50 % of system's total extent - 75 % of system's total extent - 100 % of system's total extent	- Transaction times of certificate validations and signature verification (Dependent on how large document (SV) and which RespondWith Items that are included) - How many concurrent transactions (load)? - Nr of concurrent users	Nr. of security breaches over the past x period: - Level 1 - Level 2 - Level 3 A breach: unelgible system or data acess, system or data corruption etc.
Updates	Measure the time the system is unavailable due to updates over a period of time	- Downloading time of CRLs When heavy download – many concurrent CRL downloadings	Number of missing or incorrect updates of users, messages CRLs etc. over period x.
Upgrading	The time the system is unavailable due to upgrading – this can be found in ADSS log files. Might be difficult to differentiate between the different nodes cohesion, coupling and modularity.	The degree (in terms of level and duration) to which upgrading has been a hinder to the needed scalability.	Number (per degree) of security breaches due to upgrading.
Coupling			Number of security breaches due to unsatisfactory or unnecessary cross-component interactions?
Monitoring	To what degree is the system unavailable due to weaknesses of the monitoring mechanism (logging and error handling)?	To what degree is the system unable to scale due to weaknesses of the monitoring mechanism (logging and error handling)?	To what degree is the system unable to handle security breaches due to weaknesses of the monitoring mechanism (logging and error handling)?
Repository (Logging)	Percentage of repository availability.		
Statistical data (Logging)	Percentage of availability relevant statistics.		
Db recovery (Recovery mechanisms)	Loss of availability due to insufficient db recovery.		
SW recovery	Loss of availability due to		

Fig. 43. The measurement plan for VA - Part 1

Node/ DV	**Availability**	**Scalability**	**Security**
(Recovery mechanisms)	insufficient sw recovery.		
Error detection (Error handling)	Nr. of availability relevant errors detected, in relation to the number of the actual ones.	Nr. of scalability relevant errors detected, in relation to the number of the actual ones.	Nr. of security relevant errors detected, in relation to the number of the actual ones.
Error messaging (error handling)	Nr of availability relevant errors which are timely reported, in relation to the number of the detected ones.	Nr of scalability relevant errors which are timely reported, in relation to the number of the detected ones.	Nr of security relevant errors which are timely reported, in relation to the number of the detected ones.
Database efficiency mechanisms		To what degree (percentage in duration per scale) is the system unable to scale due to limitations in database?	Number of security breaches (per security level) due to database related weaknesses.
Scheduling (Database efficiency mechanisms)	Percentage of detected availability relevant errors caused by weak scheduling.	Percentage of detected performance relevant errors caused by weak scheduling.	Nr of message or data corruption events due to scheduling
Search mechanisms (Database efficiency mechanisms)	Time to look up in the database? – Can be taken from log files		Search mechanisms delivering incorrect, or unauthorised data?
Message routing	Transaction time?	Messages delayed due to routing problems?	Messages lost or incorrectly delivered due to routing problems?
OS support services			
Middleware support services	Percentage of availability relevant errors due to middleware weaknesses.	Percentage of scalability relevant errors due to middleware weaknesses.	Percentage of security relevant errors due to middleware weaknesses.
Hardware security module (hardware)	Percentage of downtime due to HSM weaknesses.		
Front end server (hardware)	Percentage of downtime due to front end server weaknesses		
VA server (hardware)	Percentage of downtime due to VA server weaknesses		
Network	Networks availability – any downtime is reported in the monthly service report		
Measures for protection of op. env.			Percentage of the security breaches due to operational env. weaknesses, or that could have

Fig. 44. The measurement plan for VA - Part 2

Node/ DV	**Availability**	**Scalability**	**Security**
			been prevented by the operational environment.
Firewalls (VAS: Measures for protection of op. env.)	Percentage of the downtime caused by the firewall.	How soon a new firewall opening can be in place? (or is that update?)	Percentage of the security breaches due to firewall weaknesses, or that could have been prevented by firewall.
Physical protection (VAS: Measures for protection of op. env.)	Percentage of availability degradation due to weak physical security.	Percentage of scalability degradation due to weak physical security.	Percentage of security breaches due to weak physical security.
Data encryption (VAS: Measures for protection of op. env.)	Percentage of availability degradation due to encryption weaknesses.	How many concurrent SSL connections to the service (stress testing)?	Percentage of security breaches due to weak data encryption.
Redundancy (VAS: Measures for protection of op. env.)	Percentage of availability degradation due to weak redundancy.		
User management	Percentage of availability degradation due to general user management weaknesses.	- Time to add new users	Percentage of the total number (scaled) security breached caused by weaknesses in user management (provisioning, user activation and access right management).
Provisioning (User management)	Percentage of availability degradation due to provisioning weaknesses.		Fraction of the cases of incorrect provisioning, in relation to the total number of provisioning jobs.
User activation (User management)	How fast a user can be added an activated?		Fraction of the cases of incorrect (or missing) user activation, in relation to the total number of user activations.
VAS: Access rights management (User management)	Percentage of availability degradation due to access rights weaknesses.		Fraction of the cases of incorrect (or missing) access rights activities, in relation to the total number of access rights activities.
Gateway	Percentage of availability degradation due to gateway weaknesses.		Percentage of gateway failures, in relation to the nr of successfully performed tasks.

Fig. 45. The measurement plan for VA - Part3

Appendix 8: Schema for documentation of results of the thought experiment

The schema for documentation of results of the thought experiment is shown in Fig. 46. The domain expert group was asked to consider each specified change independently, agree upon the design model modifications (if any), and finally to agree upon an estimated new value of the root node QCF (represented by the question marks on Fig. 46) on the respective DVs, which could have possibly been affected. Some of the changes explicitly implied modifications of only a single specified DV, which is why some of the fields on Fig. 46 do not contain question marks. The approved prediction models were available to each domain expert. In addition, the relevant current root node QCF was informed about by the analyst. The discussion about each estimate took approx. 2-5 minutes.

Change number	Change description.	Availability (Current QCF=X)	Scalability (Current QCF=Z)	Security (Current QCF=Y)
1	Split signature verification component into two redundant components, with load balancing.	?	?	?
2	Merge CV and SV interfaces (not components).	?	?	?
3	Introduce redundant VAS with external workload balancing.	?	?	?
4	Average size (nr. of data elements and data amount) of SV requests increases by 100%.	?	?	?
5	Use of gateway made mandatory.	?	?	?
6	Decrease error detection QCF for availability by 50%.	?		
7	Decrease coupling QCF for availability by 50%.	?		
8	Decrease upgrading QCF for availability by 50%.	?		
9	Increase scalability QCF under modularity by 66%.		?	
10	Increase scalability QCF upgrading by 56%.		?	
11	Increase scalability QCF "measures for protection of operational environment" by 37%.		?	
12	Increase security QCF "logging" by16%.			?
13	Increase security QCF "monitoring" by 16%.			?
14	Increase security QCF "measures for protection of operational environment" by 5%.			?

Fig. 46. Schema for documentation of results of the thought experiment

Appendix 9: The process diagram for evaluation of the PREDIQT method

Fig. 47 shows an activity diagram for the overall plan for evaluation of PREDIQT. The method evaluation model shown assumes that the prediction models are approved.

The thought experiment based evaluation was, for each change in the change specification, carried out according to Fig. 47. The results of PREDIQT simulations were only presented when all the planned thought experiments and measurements had taken place.

Two aspects are crucial when performing a taught experiment:

- The participants' insecurity should be kept within an acceptable threshold.
- The simulation results from prediction models should not be revealed for the participants prior to the taught experiment execution.

In general, the acceptable number of repetitions can be calculated from statistical power analysis. The number obtained is in most cases much higher that what is feasible or realistic. Therefore, the selection of changes and the number of repetitions should be result of a thorough planning.

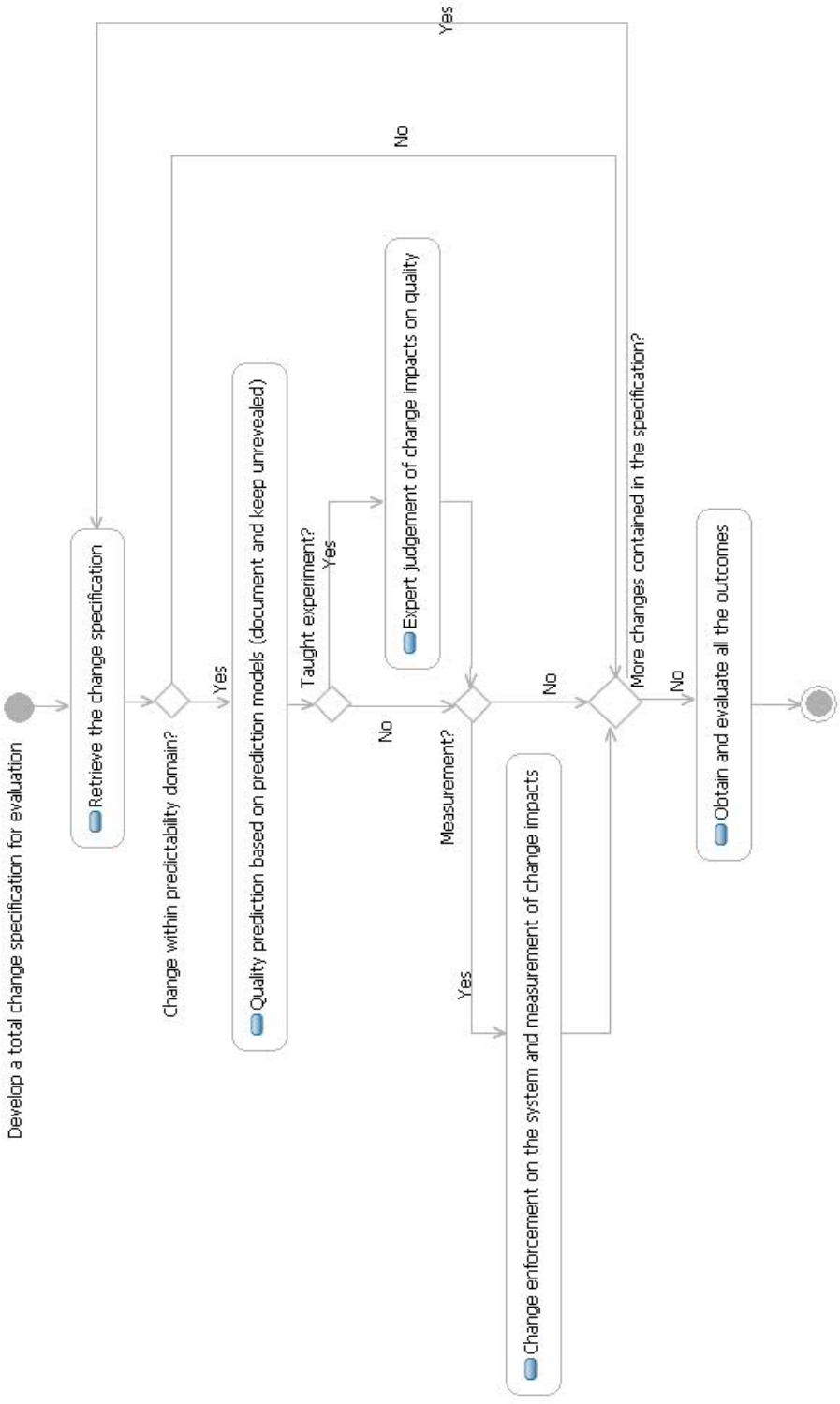

Fig. 47. The PREDIQT method evaluation plan