Collaborative Network for Industry, Manufacturing, Business and Logistics in Europe

D6.3

Trust and Reputation Management

Project Acronym	NIMBLE
Project Title	Collaboration Network for Industry, Manufacturing, Business and Logistics in Europe
Project Number	723810
Work Package	WP6
Lead Beneficiary	INN
Editor	Marko Vujasinovic (INN), Violeta Damjanovic-Behrendt (SRFG)
Reviewers	Wernher Behrendt (SRFG)
Contributors	Developer teams of INN, SRFG, SRDC
Dissemination Level	PU
Contractual Delivery Date	31/10/2018
Actual Delivery Date	31/10/2018
Version	V1.0

Abstract

We report on the design and implementation of trust and reputation management functionality in the NIMBLE platform release 5.0. (October 2018). The document describes the design of services that support the creation of trust and reputation of platform participants, leading to measures of overall platform trustworthiness and thus, creating incentives for an increase of interactions. Initial trust and reputation requirements in NIMBLE are identified through the analysis of the four project use cases (D1.1 "Requirements and Collaboration Design for Manufacturing and Logistics in Four European Use Cases"), NIMBLE cybersecurity requirements (D6.1 "Security and Privacy Requirements"), and through additional trust-related interviews with the NIMBLE use case providers.

Based on the analysis of the existing front-end and backend services of the NIMBLE platform release 3.0, the design of platform services to capture trust and reputation features on the platform is documented in Appendix 2 and through UI mock-ups and wireframes. The detailed technical description of the Trust Service implementation and its interactions with other NIMBLE services, including Identity Service, Business Process Service, and Catalogue Service, is presented in the document.

In addition to the centralized trust management in NIMBLE, this document provides discussion of a decentralized trust approach by taking advantage of blockchain technology and the Federated Byzantine Agreement (FBA) algorithm. FBA is implemented in the Stellar consensus protocol, supporting open membership and promoting organic network growth as expected in NIMBLE.

NIMBLE in a Nutshell

NIMBLE stands for the collaborative **N**etwork for **I**ndustry, **M**anufacturing, **B**usiness and **L**ogistics in **E**urope. It will develop the infrastructure for a cloud-based, Industry 4.0, Internet-of-Things-enabled B2B platform on which European manufacturing firms can register, publish machine-readable catalogues for products and services, search for suitable supply chain partners, negotiate contracts and supply logistics. Participating companies can establish private and secure B2B and M2M information exchange channels to optimise business workflows. The infrastructure will be developed as open source software under an Apache, permissive license. The governance model is a federation of platforms for multi-sided trade, with mandatory interoperation functions and optional added-value business functions that can be provided by third parties. This will foster the growth of a net-centric business ecosystem for sustainable innovation and fair competition as envisaged by the Digital Agenda 2020. Prospective NIMBLE providers can take the open source infrastructure and bundle it with sectorial, regional or functional added value services and launch a new platform in the federation. The project started in October 2016 and will last for 42 months.

Document History

Version	Date	Comments
V0.1	12/12/2017	Initial version and assignments distribution. An additional trust questioner sent to use case partners.
V0.2	05/04/2018	Requirements analysis from DoW, project deliverables and Appendix 1.
V0.3	07/06/2018	State-of-the-art in trust and reputation; review of methods for trust and reputation modelling; review of the role of trust and reputation in multi-sided digital platforms.
V0.4	15/07/2018	Initial design and description of trust service and trust scoring.
V0.5	05/08/2018	Appendix 2 contains the analysis of NIMBLE UIs and existing services to support trust and negotiation methods and services.
V0.6	19/10/2018	Description of trust data model, UBL model extensions to incorporate trust metrics and indicators, trust component and sequence diagrams, and design of UI/UX front-ends through mock-ups and wireframes.
V0.7	22/10/2018	Document revision and draft finalization
V0.8	27/10/2018	Draft document sent for QA
V0.9	31/10/2018	Finalised QA
V1.0	31/10/2018	Submission

Table of Contents

List of Figures

List of Tables

Glossary

eIDAS	*electronic IDentification And trust Services*
Apache Kafka	*An open-source stream-processing software platform developed by the Apache Software Foundation*
BFT	*Byzantine Fault Tolerance*
DLT	*Distributed Ledger Technology*
FCB	*Federated Byzantine Agreement*
FOAF	*Friend Of A Friend*
GDPR	*The GDPR or General Data Protection Regulation 2016/679 is a regulation in EU law on data protection and privacy for all individuals within the European Union and the European Economic Area.*
GUI	*Graphical user interface*
IoT	*Internet of Things*
IPR	*Intellectual Property Rights*
QTSP	*Qualified Trust Service Provider*
REST	*Representational State Transfer*
Solr	*Open source enterprise search platform, written in Java, from the Apache Lucene project. Its major features include full-text search, faceted search, real-time indexing.*
Spring	*An application framework and inversion of control container for the Java platform and microservices*
UBL	*Universal Business Language*
XDR	*External Data Representation*
MCDM	*Multi-Criteria Decision Making*
TOPSIS	*Technique for Order of Preference by Similarity to Ideal Solution*
MVP	*Minimum Viable Product*

1 Introduction

The objective of the NIMBLE project is to establish a federated, cloud-based B2B collaborative platform for digital manufacturing and supply chain management in Europe. In NIMBLE, the platform services are designed to support registering of the users and their companies, publishing of catalogues of products and services, searching through catalogues, negotiation and matchmaking between platform participants. To support decentralized interaction and data exchange between machines (devices) carrying out business processes, platform services in NIMBLE are designed to enable the creation and customization of private IoT information exchange data channels.

In task T6.3, our motivation is to design specific NIMBLE MVP (Minimum Viable Product) platform services that support the creation of trust and reputation of platform participants, leading towards an overall platform trustworthiness and increase of interactions over the platform. In 2016, the *eIDAS regulation of the European Parliament and of the Council of the EU on electronic identification and trust services for electronic transactions in the internal market* came into force with a legal framework for interoperability and security of electronic trust services across the EU (see: https://ec.europa.eu/digital-single-market/node/50813). eIDAS electronic trust services include a range of services around digital signatures, digital certificates, electronic seals, timestamps, etc. [ENISA18]. Article 19 of the eIDAS regulation requires that providers of trust services (1) assess risks, (2) take appropriate security measures to mitigate the risks, and (3) notify the supervisory body (e.g. Qualified Trust Service Provider (QTSP)) about captured incidents (breaches) that have significant impact on trust services and the personal data contained therein. Therefore, the creation of trust, trust measures, policies and guarantees has to take an important place in the design of trust and reputation mechanisms in NIMBLE.

In general, trust might be considered as a wider concept than reputation; the reputation is an element of trust, and there can be other elements of trust e.g. mechanisms that secure platform and data privacy. For example, in [JOIB07] "reputation is [defined as] what is generally said or believed about a person's or thing's character or standing." However, the NIMBLE takes a reputation-based trust view, and here, reputation is closely related to trust, and used as the basis of a judgement as to whether to trust an individual or organization. (see Figure 6 Conceptual model of trust and trust elements on page 29).

Other traditional eCommerce marketplaces such as Amazon and eBay, also use reputation-based trust mechanisms, which accumulate users' reviews and ratings about products and sellers, helping the users to evaluate trustworthiness of their business decision. In traditional product and service discovery platforms, the trust relationship is in most cases unidirectional, e.g. buyers search for trustworthy products and services, while sellers do not consider reputation of buyers. In NIMBLE, trust could be beneficial, although not sufficient because the multi-sided platform participants typically require a greater trust control within their collaboration networks. Hence, in our approach, **trust is defined as an evaluated expectation that is available to all participants involved in a specific interaction on the platform, giving a trust-related insight about other interacting participants, before engaging into an interaction with them.** A trust negotiation in NIMBLE is an interaction process between

platform participants that is based on an evaluation whether they can engage into a mutual trustworthy relationship, according to their defined trust policies and trust guarantees.

The NIMBLE platform creates trust indices from measuring trust features, including users' reviews and ratings and users behaviour on the platform, and putting them through a weighting and normalising algorithm in order to project the trust values of companies onto a normalised scale from 0 to 1. Once the projected trust values are on a normalised 0 to 1 scale, the trust management services further sub-range that scale into the areas of one- star, two- star and three- star signs[1], where the three-star sign is a good reputation, two-stars is a medium reputation, one star is a low reputation, and no-stars would be an unknown reputation (for just-registered companies on the platform without any trust-related features provided). Therefore, the **reputation in NIMBLE platform is the mapping of trust measures on a normalised scale from 0 to 1 to a simplified and more meaningful scale of three levels of reputation** (good > medium > low).

1.1 Document Structure

Section 1 explains our motivation to address trust and reputation as integral elements of the NIMBLE platform functionality.

Section 2 describes the concept of trust as it has been defined in Computer Science, as trust related to systems (security driven definition) or trust related to users (based on trust values shared by users via specifically designed trust and reputation systems). Section 2 defines the use of concepts of trust, trustworthiness, its values and measures, trust indicators, reputation, reputation values and measures, for the purpose of presenting our approach in this report. Here we additionally analyse the dynamic and context-specific nature of trust and reputation in multi-sided digital B2B platforms, and through the state-of-the-art analysis, we look at methodologies for establishing trust either through direct interaction over the platform, or through third-party recommendation and reputation. Here, we also look at the trust evaluation methods that are focused on either the structure of a given social network measuring trust propagation among its members, or on the interaction among the members.
Finally, section 2 looks at several successful eCommerce platforms (e.g. Amazon.com, eBay.com, BizRate.com, and Alibaba.com) and their trustworthiness systems. It concludes by discussing recent the most prominent trust and reputation ranking algorithms.

Section 3 presents the trust and reputation modelling approach in NIMBLE. Here we start with the identification of trust and reputation requirements captured through analysis of NIMBLE use cases (see D1.1 "Requirements and Collaboration Design for Manufacturing and Logistics in Four European Use Cases") and NIMBLE cybersecurity requirements (see D6.1 "Security and Privacy Requirements"). We briefly explain our approach to trust and reputation, and continue with the design and implementation of the NIMBLE trust and reputation data model and the improvement of user interfaces (UIs) to support these new requirements.

[1] A NIMBLE front-end actually shows a heart symbol instead of a star symbol

Section 4 presents the implementation details of centralized trust management in NIMBLE**.** Architecture of Trust Service is introduced at the beginning of the section, followed by an explanation of trust scoring algorithm/method. Further, the section gives an overview of the needed user interfaces (UIs) for the selected use-cases that involve trust management in NIMBLE, then it provides the details of the trust database models, UML component and sequence diagrams, and finalize with the REST specification of the Trust Service.

Section 5 presents the decentralized trust approach by taking advantages of blockchain technology (this section is based on our published research paper "*Federated Byzantine Agreement to Ensure Trustworthiness of Digital Manufacturing Platforms*" [INDB18]).

Section 6 provides a conclusion.

2 Background

2.1 State-of-the-Art in Trust and Reputation

Digital B2B platforms require both **security methods** to protect platform assets and users from attacks, intrusions and vulnerabilities of the system, and **trust methods** which are required to build confidence of platform participants and business reputation through stability of the B2B platform environment. The notion of trust has various meanings in different fields of science, e.g. from a psychological point of view, a social perspective, a legal perspective, in commercial situations, etc. [Hartman03] classifies trust as *blue trust* (competence trust; e.g. can you do this job for me?), *yellow trust* (integrity trust, e.g. will you constantly look after my interests?), and *red trust* (intuitive trust, e.g. does this feel right?). In the computing literature, most of definitions of trust are focused on the action or behavioural aspects of trust, while some of them cover the context-dependent nature of trust [Huss04] or the dynamic nature of trust, e.g. over time or when the behaviour of a trusted party changes influencing the trust value of that party [Dillon04]. Trust is realized by the concept of **a trust relationship** that is determined by **a trust value**. Trust relationships could be unidirectional (between two trusted parties) or multidirectional trust (between multiple trusted parties). Furthermore, trust concepts have **trust attributes** and **trust methods** as functions, operations that calculate the values of attributes for each concept, creating measures of the **trustworthiness** of systems [Chang06] with trustworthiness scales that can range in their nature from Bayesian scales with two values, e.g. satisfying denoted by 1, and unsatisfying, denoted by 0 in [Wang03], to representing trust by 1 and "mistrust" by -1 in [Aberer03], etc.

The authors in [Chang06] define **a seven-level trustworthiness** metric that uses numeric and non-numeric measures for the evaluation of trustworthiness:

- level -1 (unknown agent); Quality of Service (QoS) rating: new agent;
- level 0 (very untrustworthy); percentage intervals: 0-19; QoS rating: terrible;
- level 1 (untrustworthy); percentage intervals: 20-39; QoS rating: bad;
- level 2 (partially trustworthy); percentage intervals: 40-59; QoS rating: average;
- level 3 (largely trustworthy); percentage intervals: 60-79; QoS rating: good;
- level 4 (trustworthy); percentage intervals: 80-90; QoS rating: very good;
- level 5 (very trustworthy); percentage intervals: 91-100; QoS rating: excellent.

According to the literature [CAFV13][ARGI07][BODO05][JOIB07][SHNP13], trust in Computer Science can be classified into two categories, depending on whether it refers to systems or users.

- Trust related to systems consists of security mechanisms involving policies, which describe the conditions necessary in a system to obtain trust.
- Trust related to users is based on trust values (direct or recommendations) gathered and shared by users in a distributed community, via trust and reputation systems.

2.1.1 Trust Related to Systems

Trust related to systems is supported by both software- and hardware-based solutions, which according to Bonatti et al. [BODO05], follow an approach based on security mechanisms to create *"trusted"* systems that overcome technical failures and malicious attacks [CAFV13]. This kind of **policy-based trust management** describes the conditions necessary to obtain trust, and also prescribes actions and outcomes if certain conditions are met. Blaze et al. [BLFL96] propose a comprehensive trust management scheme called **PolicyMaker** and present trust management policies that specify the trusted behaviors and trust relationships. In the field of pervasive computing, interesting works are presented in [KAZA05] (intrusion detection within pervasive computing environments), [WEIS05] (security of human–computer interactions), and [YUMH06] (a pervasive computing security system based on human activities analysis). A vision of trust based on traditional policy-based mechanisms has been criticized from the view of security, e.g. the authors in [NISS99] and [OSTE01] state that the level of security in a system does not necessarily affect trust.

2.1.2 Trust Related to Users

Trust related to eCommerce users is important as people are certainly more willing to engage online if they are assured that their personal and financial data are protected. According to a standard definition of trust, which is derived from psychology and sociology by Marsh [MARSH94] and Mui et al. in [MUIM02], trust is *"a subjective expectation an agent has about another's future behavior based on the history of their encounters.*" According to the **probabilistic definition of trust** provided by Gambetta in [GAMB90], trust is *"the subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends."* According to the **cognitive definition of trust** in [CAFA98], it is *"a mental state, a complex attitude of an agent x towards another agent y about the behavior/action relevant for the result (goal) g."* Trust can be lost quickly: the authors in [NIEL99] believe that "[trust] is hard to build and easy to lose: a single violation of trust can destroy years of slowly accumulated credibility." Trust can be direct or based on recommendations. **Direct trust** is based on the direct experience of the member with the other party. **Recommendation-based trust** is connected to reputation, which is a social evaluation or an assessment based on the history of interactions with, or observations of, an entity, either directly with the evaluator (personal experience) or as reported by others (recommendations or the 3rd party verification) [ARGI07]. Recommendations may be based on **collaborative filtering techniques** which unfortunately, perform poorly when there is insufficient previously available mutual rating between users; this is commonly known as the **cold start problem** [LVLD08]. To overcome this problem, the introduction of **purely trust-based approaches** to recommendation has emerged, which assume a trust network among users and make recommendations based on the ratings of the users that are directly or indirectly trusted by the target user.

In the literature, there are various techniques for the implementation of trust and reputation, which range from numerical/statistical and Machine Learning (ML) techniques, to heuristic and behavioral techniques. Numerical/statistical and ML techniques focus on mathematical models for trust management, e.g. Bayesian systems [COJOIS02], belief models [JOHP06][YUSI02], average of ratings [REZE02], or Artificial Neural Networks (ANNs) and

Hidden Markov Models (HMMs) for computing and predicting trust. Furthermore, heuristic techniques focus on defining practical models for implementing robust trust systems, while behavioral models focus on user behavior in the community, and differentiate between conversation trust and propagation trust. **Conversation trust** specifies how long and how frequently two members communicate with each other, while **propagation trust** refers to the propagation of information.

2.1.3 Trust Evaluation Models

Trust evaluation models vary from those (i) that consider only the structural properties of the social network and develop **structure-based trust models** (based on trust propagation techniques), (ii) that consider the interactions among users in the social network and **develop interaction-based trust models** (based on trust prediction) and (iii) those that take into consideration both aspects and techniques, generating in this way **hybrid trust models**.

Structure-based trust evaluation models exploit the social network structure in evaluating trust, e.g. trust values are either explicitly provided or where they can be inferred. In these models, a trust network is created for each member, representing the other members in the person's social network as nodes and the amount of trust s/he has for each of them as edges [SHNP13]. Buskens in [BUSK98] observes that high interconnection between members can yield a high level of trust.

Interaction-based trust evaluation models include the following models:

- Liu at al. in [LLLL08] observe that a user trusts another user either because of the good reputation or because of good personal interactions between the two users. They propose **a supervised learning approach for the automatic prediction of trust between a pair of users** exploiting evidence derived from (i) actions of individual users, as well as from (ii) interactions between pairs of users.
- Adali et al. in [AEGH10] quantitatively measure trust between two entities based on observed communication behavior in social networks. They evaluate behavioral trust by taking two social behaviors into account: **conversations and propagation of information from one person to another**.
- Nepal et al. in [NESP11] propose STrust, a **social network trust model designed to encourage positive interactions among network members**. They separate the interactions of users into two groups: popularity and engagement. Popularity-based interactions are based on the trustworthiness of a member in the community. Engagement-based interactions are based on how much one user trusts other users in the community, and show how frequently members visit the site/network, how many members they follow, how many posts they read and comment on, etc.
- Švec and Samek in [ŠVSA13] create a **model of trust that is based on the Marsh theory** [MARSH94] that considers the following elements: the interaction time span, number of interactions, number of characters, interaction regularity, photo tagging, group membership, and common interests.

Hybrid trust evaluation models combine both interactions and social network structure information to compute social trust. For example, Trifunovic et al. in [TRLA10] propose a

hybrid model for opportunistic networks that supports: explicit social trust (based on network structure) and implicit social trust (based on users' interactions in the network).

2.2 Trustworthiness Systems of Successful eCommerce Providers

In this section, we look at several successful eCommerce providers and their support systems for trustworthiness, e.g. Amazon.com, eBay.com, BizRate.com and Alibaba.com. The analysis is based on [Chang06] and information available from the platform websites.

2.2.1 Trustworthiness at Amazon.com

Amazon.com provides two kinds of trustworthiness: at the level of transaction partners and at the level of products.

- **Transactions partners** are defined as those who are directly involved in *buy-and-sell* type of transactions. The trustworthiness of sellers is measured by the quality of seller's services, and the buyers are provided with a view of the products being sold by different sellers, and the seller's trustworthiness level of rating. Sellers can leave their feedback on ratings that is not considered for the overall trustworthiness value of sellers. All comments and feedbacks can be viewed on the Amazon website.
  Apart from a 5-start rating of transactions, and leaving the comments and feedback on transactions, the Amazon Safe Buying Guarantee indicates that the conditions of the order are defined under the "A-to-Z Guarantee", which is created to protect the buyers and increase online purchases.
- At the level of **products**, the trustworthiness is implemented for two types of product reviews: customer review and spotlight review. Customer reviews allow ratings for electronic goods (games, software), and both reviews and ratings for other goods, i.e. books, apparel, etc. Customer reviews can be further rated and reviewers with the most positive votes are then classified as "Top Reviewers".

All ratings and reviews, along with the information about the reviewers are stored in the Amazon system, providing customers with a sort of information that builds confidence related to their purchasing decisions. Finally, to control the process of reviewing, Amazon.com is using a verification step for the customers, asking them to verify their identity using the verification code sent via e-mail.

2.2.2 Trustworthiness at eBay.com

The eBay online marketplace is an eCommerce platform for anyone to trade anything. It offers reputation rating systems that consider only the feedback from eBay transaction partners and calculate trust for eBay's trusted members. Comments and ratings are used as indicators of the reputations of eBay transaction partners, and are included in a member profile.

Reputation is determined using members' feedback, which is categorized as positive, negative, or neutral ratings. A total eBay score is calculated on the basis of the difference between the total number of members who left a positive rating and the total number of

members who left a negative rating. Based on the total number of points scored (that can be between 1 to over 100 000 points), the eBay member is assigned stars, e.g. yellow star (1-49 points), blue star (50-99 points), turquoise star (100-499 points), purple star (500-999 points), red star (1000-4999 points), green star (5000-9999 points), yellow shooting star (10000-24999 points), turquoise shooting star (25000-49999 points), purple shooting star (50000-99999 points), red shooting star (100000 over points). Here, star notation is represented as a star with corresponding colour, while shooting star is represented as a star with tail and with corresponding colour.

2.2.3 Trustworthiness at BizRate.com

BizRate.com is a shopping search engine that lists every online shop and online product around the world. It lists the product price, availability and rating information to support the purchase decisions of customers. It uses a rating algorithm called ShopRank that is determined by weighting price, popularity and availability of products against the reputation of merchants selling these products.

BizRate.com takes the latest feedback from BizRate point-of-sale survey network and combine it with the latest feedback from BizRate members to arrive at Store Ratings. The equation that determines each merchant's rating is calculated as follows for each dimension of service (see: http://about.bizrate.com/ratings):

```
(Average Survey Scores * Number of Surveys) + (Average Member Scores *
Number of Member Reviews) / (Number of Surveys + Number of Member Reviews)
```

BizRate collects feedback from more than one million online buyers and sellers each month. Since merchant ratings often change over time, time is also a very important element in rating calculations. BizRate only uses data from the latest 90 days when performing the calculations to arrive at a rating. That means that, the rating information on BizRate is never more than three months old. BizRate also needs a minimum of 20 surveys in the previous 90 days to regard the rating as valid.

Merchants can be found in *the Compare Prices and Stores* section of the BizRate.com, for any product, using a four-level scale (i.e. outstanding, good, satisfactory, poor). BizRate also allows users to vote if they find a particular review helpful or not. All reviews are stored in the BizRate system and are accessible to everyone, but are not further rated at the level of reviews or the reviewers. The system gives only information about the number of people who found the review helpful or otherwise.

2.2.4 Trustworthiness at Alibaba.com

Alibaba.com is a B2B portal with a mission to connect Western businesses and Chinese manufacturers [ALIBABA.COM-IPO]. Today, Alibaba.com accounts for more than 80% of all online purchases in China. The **Alibaba Business Trust System** is a solution for e-Commerce companies to exhibit their capabilities for a global audience (see: https://service.alibaba.com/buyer/faq_detail/20153570.htm). The aim of the systems is to promote e-Commerce by improving sourcing efficiency and reducing costs associated with finding trustworthy trade partners. The trust rating is calculated using Alibaba's proprietary

data technology and it depends on a number of measurements. Some ways to improve trust rating on Alibaba.com include: submit company certificates; accumulate transaction data; actively resolve any disputes with business partners; avoid false trading or IPR infringement; and maintain good financial practices. All suppliers belong to one of the following two categories: **1) Trade Assurance Supplier**: those who support Trade Assurance, a free service that protects orders from payment to delivery, and **2) Gold Plus Supplier**: a premium membership for high-level suppliers.

2.3 Popular Trust and Reputation Algorithms

Some popular algorithms for trust propagation include the following:

- **TidalTrust** [GOLD05] is a trust network inference algorithm based on the Friend Of A Friend (FOAF) vocabulary. It generates a recommendation about how much one person should trust the other, based on the paths that connect them in the network, and the trust ratings on those paths. Zhang et al. in [ZHCW06] expand Golbeck's **TidalTrust** model to include pair-wise trust ratings and reliability factors of the entities in the network, using an edge-weighted graph to calculate trust.
- **SUNNY** [KUGO07] is a trust inference algorithm that uses a probabilistic sampling technique to estimate the level of confidence in the trust information from some designated sources. It performs a probabilistic logic sampling procedure over the generated Bayesian network.
- The **gravity-based model** proposed by Maheswaran et al. in [MATG07], contains two stages: firstly, the strengths of the friendships are recomputed along with the extent of the trusted social neighborhood for each user. Secondly, the social neighborhood is used to compute the effective trust flow for users outside of the social neighborhood.
- Caverlee et al. in [CALW08] propose the **SocialTrust model** that exploits both social relationships and feedback to evaluate trust. Members provide feedback ratings after they have interacted with another member. The trust manager combines these feedback ratings to compute the social trust of the members.
- Hang and Singh in [HASI10] employ **a graph-based approach for measuring trust** that uses the similarity between graphs to make recommendations.
- The Jamali and Ester's approach, presented in [JAES10], makes recommendations for a user based on the ratings of the users that have direct or indirect social relations with the given user, employing matrix factorization techniques. The model also incorporates the mechanism of trust propagation.
- Guha et al. [GKRT04] develop a **formal framework of trust propagation schemes**, introducing the formal and computational treatment of distrust propagation. In their work, authors show that a small number of expressed trusts per individual allows the system to predict trust between any two people in the system with high accuracy.
- Wierzowiecki and Wierzbicki in [WIWI10] propose a trust/distrust propagation algorithm called **CloseLook**, which is capable of using the same kinds of trust propagation as the algorithm proposed by Guha et al. CloseLook has a lower complexity and reduces the amount of consumed computational and network resources by selecting the best paths to propagate trust and by stopping the trust propagation using scope parameters that can limit the number of considered nodes.

- Leskovec et al. [LEHK10] extends the algorithm proposed by Guha et al., using a machine-learning framework to enable the prediction task of positive/negative links. Guha et al. in [GKRT04] propose a method based on the **PageRank algorithm for propagating both trust and distrust**. They identify four different methods for propagating the net beliefs values, namely: direct propagation, co-citation, transpose, and coupling.
- **The Avogato maximum flow trust metric** [LEVI09] aims at discovering which users are trusted by members of an online community and which are not. Trust is computed through one centralized community server and considered relative to a seed of users enjoying supreme trust. Advogato assigns boolean values indicating presence or absence of trust. As it has been released under a free software license, it has become the basis of many research papers.

3 Trust and Reputation Requirements in NIMBLE

Trust and reputation requirements are identified based on the following sources:

- D1.1 "Requirements and Collaboration Design for Manufacturing and Logistics in Four European Use Cases";
- D6.1 "Security and Privacy Requirement"; and
- A questionnaire that was given to use case owners to express their additional requirements related to trust and reputation in NIMBLE (see Appendix 1).

In the following, we discuss the analysed results related to trust and reputation design and implementation.

3.1 Requirements Elicitation Based on D1.1 and D6.1

Use Case 1: Childcare Furniture Use Case (Micuna)
The Child Care Furniture use case is focused on the definition and configuration of an optimal value chain from a specific business ecosystem. This value chain covers both production needs and logistics. The ecosystem includes information about certifications of their members, as well as the option to access the normative and regulation awareness system provided by AIDIMME. The use case includes the following sub-scenarios:

- UC-1: Provider Search
- UC-2: Negotiation of the business conditions (financial, delivery, etc)
- UC-3: Awareness of normative and legislation to enter new markets
- UC-4: Publication of product catalogue
- UC-5: Product End-Of-Life

Here is the list of related trust requirements as identified in D1.1:

REQ_MIC_06 Reputation of potential provider may be assessed. The assessment of the potential partner is made internally. Optional, this might be published via NIMBLE.

REQ_MIC_30 Trust in collaboration
Focus on the definition and configuration of an optimal value chain from a rich and reliable business ecosystem.

The list of trust and reputation requirements for Micuna from D6.1:

Sec. Req. ID	UC Req. ID (D1.1)	Priority	Name	Description	Stakeholder/ Countermeasure
SEC_UC_04	*REQ_MIC_06*	*SHOULD*	*Trust & reputation assessment*	*Trust and reputation of users must be automatically calculated and managed*	*Trust and reputation mechanisms (to be based on mutual evaluation of business actors)*

Use Case 2: Textile Manufacturing Use Case (Piacenza)

This use case is focused on collaborative design and production; dynamic, real-time access to supplier virtual catalogues and inventories for fast de-sign development; full manufacturing, product traceability and real time vision of production to provide customers with information about their orders and deliveries; automatic preferential origin certificate declaration, including ethical and environmental fulfilment evidences.

The Piacenza use-case anticipates the following four scenarios:

- UC-6: Collaborative design and production
- UC-7: Virtual catalogues and services
- UC-8: IoT machine connection and data elaboration
- UC-9: Automatic origin certificate declaration

Piacenza trust and reputation requirements as identified in D1.1:

REQ_PIA_30 Trust of System
Users must be sure that the platform is intrinsically stable and secure. Bugs and service interrupts must be limited already from design phase by a proper strategy to minimize customer service in commercialization phase considering also the high number of expected users in different languages.

The list of trust and reputation requirements for Piacenza from D6.1:

Sec.Req. ID	UC Req. ID (D1.1)	Priority	Name	Description	Stakeholder/ Countermeasure
SEC_UC_15	*REQ_PIA_30*	*SHOULD*	*Trust & reputation*	*Trust and reputation must be calculated*	*Trust and reputation mechanisms; UC-08*

Use Case 3: Eco Houses Use Case (Lindbäcks)

Lindbäcks's need in NIMBLE is to improve their supply chains to seamless connect stakeholders and exchange data for manufacturing building. The key planned capability of Eco Houses is to build modularized buildings rapidly on a construction site, while assuring a quality-built apartment building (buildings that should last for 100 years). To that end, this use case has the following target scenarios:

- UC-10: 3D Product Configurator
- UC-11: IoT Measurements (in Bath Room)
- UC-12: Tracing/traceability Components
- UC-13: Quality Control Info

Lindbäcks trust and reputation related requirements as identified in D1.1:

There is no trust-related requirements from this use case in D1.1

The list of trust and reputation requirements for Lindbäcks from D6.1:

There is no trust-related requirements from this use case in D6.1

Use Case 4: White Goods Use Case (Whirlpool)

This use case aims to establish a collaborative environment for sharing product quality data collected from different sources (considering and consolidate complementary data sources) (e.g. UC-14: Regression Study), and to allow actors in the product lifecycle to improve their capability to take the right product related decisions (UC-15: Product Avatar).
These two scenarios are expected to make an impact on product quality, to increase the effectiveness in product field-failure resolution and, in long term, to improve an overall quality perception from the market.

Whirlpool trust and reputation related requirements identified in D1.1:

REQ_WHR_22 Security, Trust and Privacy Quality data are sensitive. Access to the system has to be granted using Whirlpool internal policy (LDAP).

The list of trust and reputation requirements for Whirpool from D6.1:

There is no trust-related requirements from this use case in D6.1

3.2 Requirements Elicitation Based on Trust Questionnaire

In project month M15, an additional questionnaire was sent to NIMBLE industrial partners (use-cases owners) in order to gather new requirements related to trust and reputation from them. The questionnaire is reproduced in Appendix 1. A short discussion on the results of this questionnaire is given below.

The questionnaire was focused to gather information about the following issues:

- Purpose and understanding of the trust concept, and expectations of trust and reputation functionality of the platform in a respective use-case;
- Identification of trust related elements and their relevance in a respective use-case (e.g., importance or relevance of user ratings, written opinions, company profile and evidence of history, security and privacy-related matters such as identity management, access control, etc.)
- Identification of non-NIMBLE services and/or data-sources that NIMBLE industrial partners use to evaluate trust and reputation in their everyday business collaborations.

First of all, the survey confirms our premise that of trust is a subjective, multi-faceted concept, and that trust is contemplated differently from one user to another, and also from one use-case to another. For example, Micuna has stated that in their use-case trust is related to the confidence in security access controls and identity management, and that a reputation of other business actors within NIMBLE should be based on available comments, ratings and opinions submitted by other business actors. For Micuna, a number of successful negotiations and similar metrics might be relevant too. On the other hand, Whirlpool stated the importance of trust with strong identification mechanism, while user ratings and reviews are not so relevant for their business.

3.3 Summary of Requirements

Table 1 and Table 2 summarize the above discussed trust and reputation requirements.

Table 1 Trust and Reputation Management - Functional requirements

Req.ID.	Name	Description	Priority	Relationship to D1.1. and D6.1 req. and use-cases
TRM_01	*Trust & reputation assessment*	*Trust and reputation of users must be automatically calculated and managed*	*SHOULD*	*SEC_TRM02, SEC_UC_15 - REQ_PIA_30, SEC_UC_04 - REQ_MC_06*
TRM_02	*Provider reputation assessment*	*Reputation assessment for providers that have their profile and historical data in Nimble should be provided using user ratings, user reviews, and other relevant inputs.*[2]	*SHOULD*	*SEC_UC_04 - REQ_MC_06*
TRM_03	*Trust criteria specification*	*Users should be able to specify their trust expectations in terms of different trust-related properties and their relevance (weights)*	*SHOULD*	*SEC_TRM02, SEC_UC_15 - REQ_PIA_30, SEC_UC_04 - REQ_MC_06*
TRM_04	*Trust calculation method*	*Trust calculation method should be take different trust-related properties as its input, specified according to the trust criteria and trust promises of trust parties.*	*SHOULD*	*SEC_TRM02, SEC_UC_15 - REQ_PIA_30, SEC_UC_04 - REQ_MC_06*
TRM_05	*Trust score value*	*Final trust score should be quantitative and normalized to 0-1 range.*	*SHOULD*	*SEC_TRM02, SEC_UC_15 - REQ_PIA_30, SEC_UC_04 - REQ_MC_06*
TRM_06	*Recommend-ation based on trust score*	*Platform should be able to rank NIMBLE actors and services according to their trust score*	*SHOULD*	*UC-1, UC-4*

[2] *Number of company relationships (contacts) arranged through the platform; number of negotiations made via the platform which had resulted in business collaboration (successful / less successful); number of products of the company with some rating or opinion in the platform; average rating of products of the company average rating of the company in the platform.*

Table 2 Trust and Reputation Management - Nonfunctional requirements

Req.ID.	Name	Description	Priority
NTRM_01	*Microservice Architecture*	*Component has to be developed as a microservice*	*MUST*
NTRM_02	*API*	*Component has to expose its services using Rest API*	*MUST*
NTRM_03	*Security*	*Access to trust services must be secured with authentications and authorisations controls.*	*MUST*

3.1 Examples of Trust Scenarios

3.3.1 Example 1: Trust and Reputation Enhanced Provider Search

NIMBLE's UC-1 *Provider Search* can be enhanced with trust and reputations management (see Figure 1). In its core, the Provider Search is about manufacturers who use NIMBLE to find providers (suppliers) who can provide required materials and operations. There can be multiple providers that match the essential search criteria, but the trust in the potential providers is an important point that has to be assessed before choosing the right provider (i.e. the most trustful one).

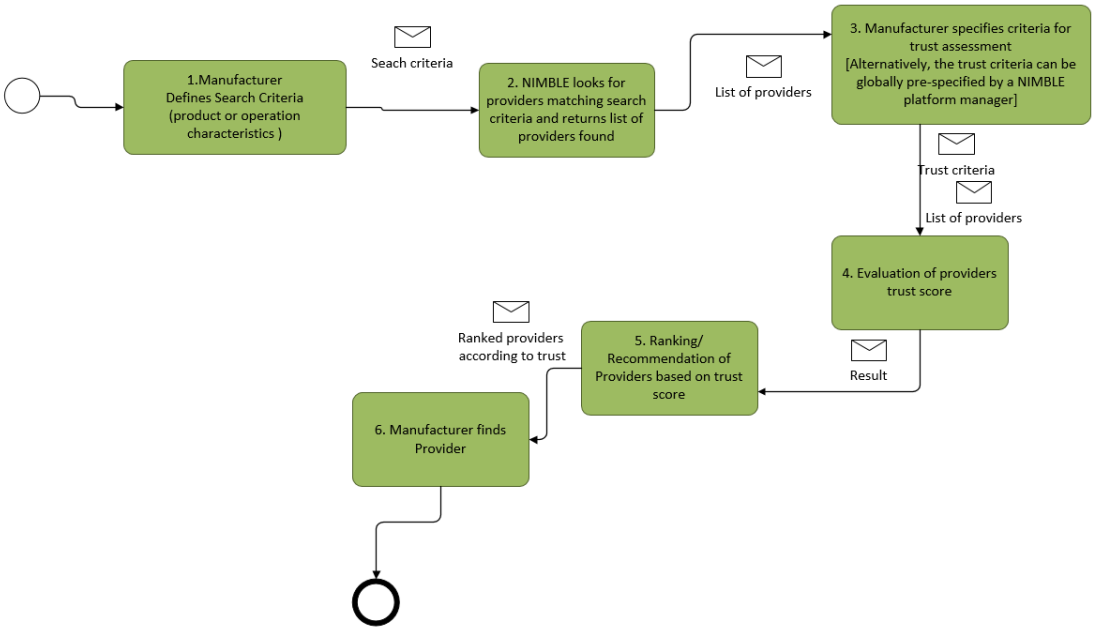

Figure 1 Ranking providers according to trust

Trust of providers can be derived from various sources including reviews or opinions, ratings, and possible other historical data about provided or stored in NIMBLE (e.g. number of company relationships arranged through the platform, number of successful / less successful negotiations made via the platform, number of products of the company with some rating or

opinion in the platform, average rating of products of the company, average rating of the company in the platform). Each of trust and reputation indications can be more or less relevant, depending on the manufacturer thinking what is and what is not relevant to assess the trust, in a specific Provider Search situation.
In the scenario depicted in Figure 1, a manufacturer specifies a search criterion to find providers according to desired materials or operations characteristics. Then, after NIMBLE returns providers that match the criteria, the manufacturer asks NIMBLE to rank providers according to their reputation and trust score. For example, a policy for the trust assessment may express that a high average rating of the provider in the platform and a high number of provider relationships are very important indicators of trust, while, for example, a number of products of the company is of minor or no relevance. Given the trust policy and a list of providers, NIMBLE aggregates all the needed trust evaluation inputs and automatically calculates the reputation of providers, and their final trust score. Finally, NIMBLE ranks providers according to their trust score and the manufacturer can choose the provider with the highest trust rank.

3.3.2 Example 2: Trust and Reputation Enhanced Product Search

This scenario is similar to the first one, but the trust and reputation in this scenario are assessed for products not for providers. Trust or reputation of the products can be derived from reviews or opinions about specific products, average rating of products, number of purchases, claims, etc.

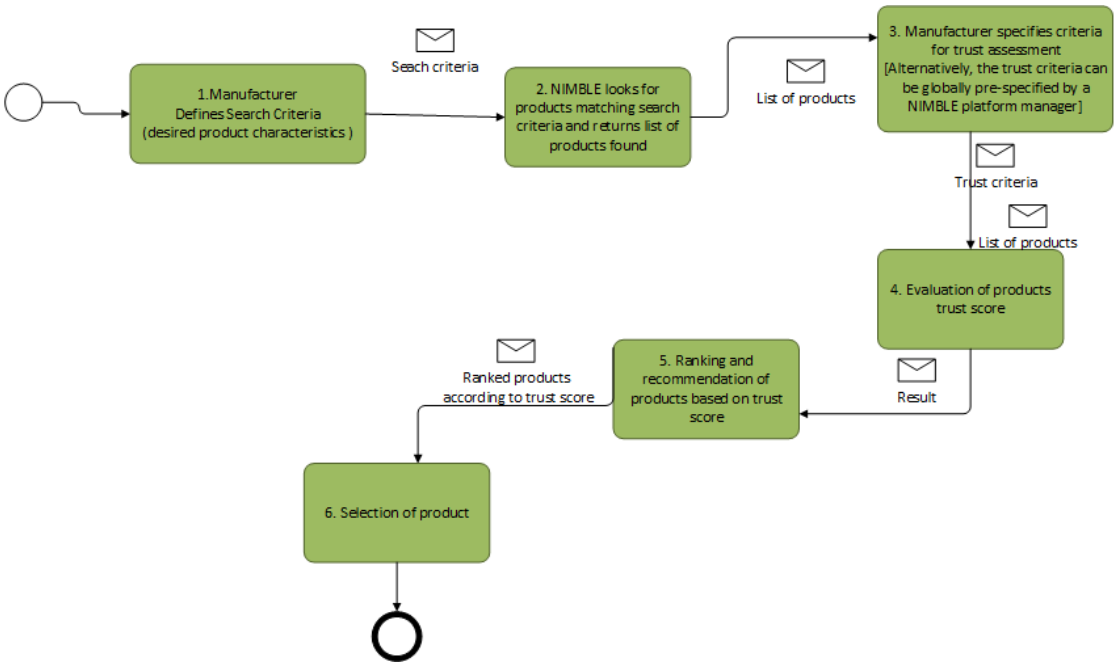

Figure 2 Ranking products according to trust

For example, in this scenario a manufacturer may expresses that a high average rating of the product and a large number of positive opinions about product are very important indicators of trust, while the company's business transaction history is less relevant for trust assessment.

4 Trust and Reputation Centralized Approach

Studies on trust architectures and trust propagation methods can be categorized into centralized, decentralized and distributed approaches. In the centralized approach, each trust request and service goes through a central node which can be further accessed by all other nodes in the domain [TJUL16]. Here, a central node is responsible for managing trust information including trust negotiation, calculation and decision making. One of the most prominent areas where centralized trust computation has been deployed is in social networks like Facebook.com, and e-markets like Amazon.com and eBay.com [BARB11][LEPE08], which calculate reputation as a function of the cumulative ratings of users by others.

In NIMBLE, we use and expand a trust approach from [VUGA14] where the centralized trust model was built on top of a trust goal classification technique introduced by the [GAGD07] for web services discovery. The [VUGA14] approach does not take a particular and single element of trust, but rather builds a generic trust model that allows to integrate and combine different trust-related elements into the trust evaluation. The evaluation of trustworthiness of relationship between two sides (i.e. the trust evaluation) is a process of collecting trust-related properties, and based on them calculating an aggregated trust index, in regards to the given trust policy.

A trust policy is defined as set of trust criterion, which are triples of `<desired trust attribute/indicator, desired value, relevance>`. `Desired trust attribute` is a desired evidence of trust, e.g. user rating, popularity of use, response time. The trust attribute's value is e.g. good or bad ratings, or response time less than 12 hours. Trust attribute's relevance is its weight in a specific context e.g. in a specific maturity stage of platform lifecycle.

In the following we briefly describe the NIMBLE Security Architecture from D6.1 that includes the Trust and Reputation system.

4.1 NIMBLE Security Architecture

Figure 1 illustrates the NIMBLE Security Architecture, which is based on the NIMBLE architecture specification (D2.1) [D2.1_17]. It addresses basic security controls and security best practices for each of the NIMBLE core components, which are FrontEnd, Open API, Data Store, Data Management, Services, Service Discovery, Service Registry, and Cloud Service Bus component. Specifically, the NIMBLE Security Architecture designs core Security and Privacy Controls including Identity Management, Access Control Management, Authorization, Data Provenance Management, Data Quality Management and Trust and Reputation Management. The NIMBLE platform runs in the cloud, which adds both the platform service provider and the cloud service provider requirements to the list of additional security controls, as described in D6.1 and D6.2.

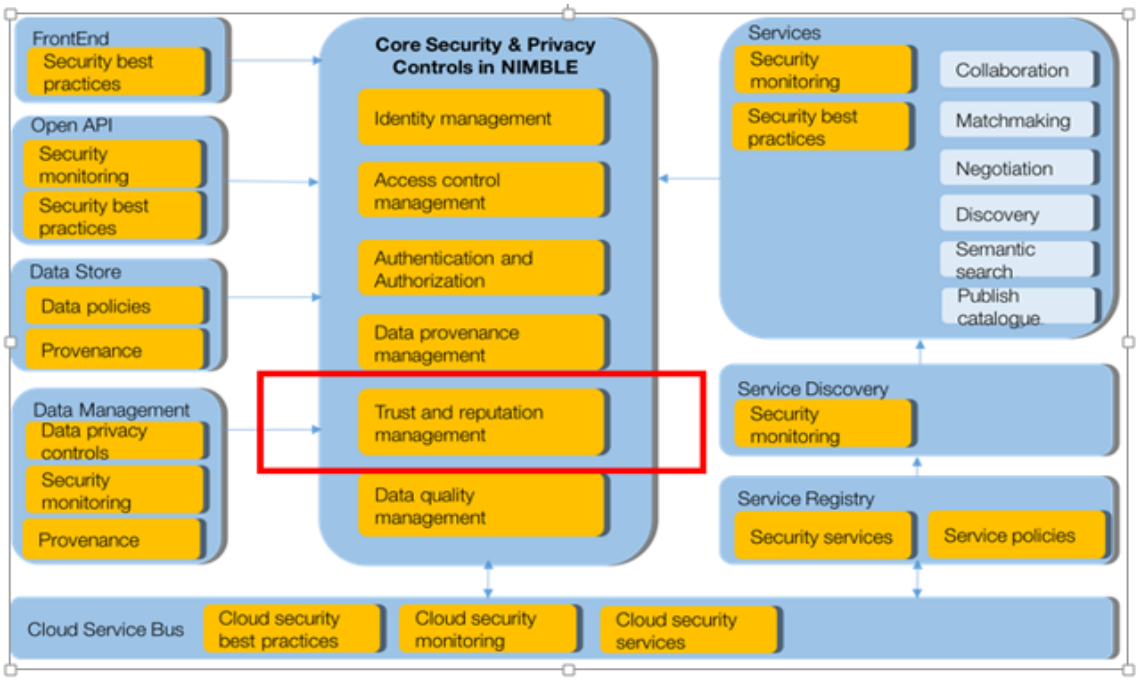

Figure 3 NIMBLE Security Architecture

The Trust and Reputation Management, as a component and service of NIMBLE Security Architecture is responsible for trust and reputation evaluation in NIMBLE. It interacts with other components that provide trust and reputation related data.

4.2 Trust Service Architecture

The figure below presents the architecture of NIMBE Trust Service with its major input, output and its relationships with other components.

The NIMBLE Trust Service is designed to support NIMBLE users in getting the answers to main two questions:

- Is a NIMBLE provider (seller) trusted, based on available information about it, including objective indicators (such as completed transactions, compliance with the terms established by both B2B parties, quality assurance, and similar) and subjective feedback that users leave on the platform?
- Is a NIMBLE provider (seller) trusted, on some scale relative to others NIMBLE providers?

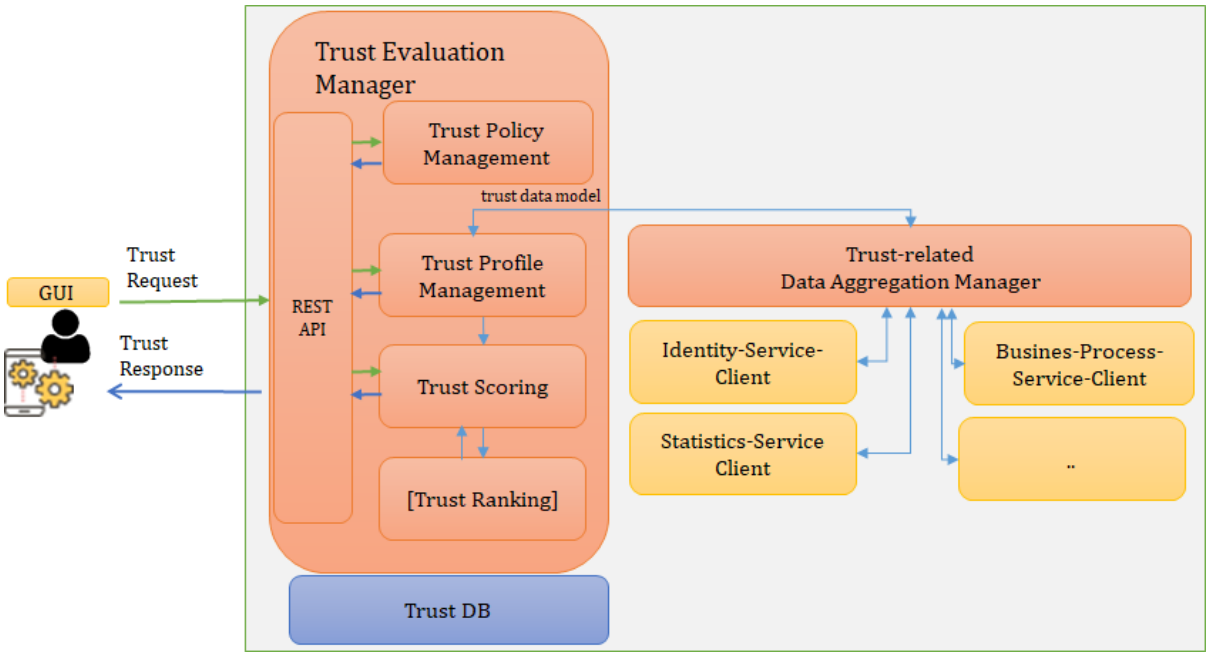

Figure 4 NIMBLE Trust Service Architecture

The NIMBLE Trust Service consists of two main high-level components, namely the Trust Evaluation Manager and the Trust Data Aggregation Manager.

- **Trust Evaluation Manager** is responsible for: (1) receiving and validating the trust evaluation requests; (2) trust policy configuration; (3) aggregation of trust-related information about NIMBLE companies into their trust profile; for (4) calculation of the trust scores and finally for (5) ranking according to trust score. Hence, the main subcomponents of Trust Evaluation Manager are:
  - Trust Policy Management
  - Trust Profile Management
  - Trust Scoring
  - Trust Ranking
- **Trust-related Data Aggregation Manager is** responsible to collect, upon a trust evaluation request, the trust–related data about the NIMBLE companies. The Trust Data Aggregation Manager collects this data by invoking other platform services that expose data of interest to the trust and reputation. For example, the trust service gets data from:
  - **Business-Process-Service.** After a business process is successful closed, the buyers may give their ratings and reviews about sellers on a five-star scale. Business-process-Service is responsible for the ratings management; to store the ratings and to provide statistical reports. In addition to providing ratings using a rating scale, buyers can provide comments and opinions about their further business experience with sellers. Such comments and opinions can be further analysed using a sentiment analysis algorithm to reveal a positive or negative sentiment.

- **Identity-service.** One possible indicator for the trust assessment is a company profile completeness rate. Through the registration process, by providing more verified and evidence-based details about the company, the company may get a higher trust score.
- **Statistics-Service** may provide an objective, statistical information about the presence of the companies on the platform, e.g., a number of transactions of a company, number of total transactions on the platform, trading volume, average response time, or average business process competition time.

- **Trust database** is a persistent storage where trust and reputation profiles (i.e., trust-related attributes and their values) of NIMBLE entities can be centrally kept for the purpose of trust evaluation. A trust profile of NIMBLE entity is populated by the data that are collected by the data aggregation manager. In addition to the trust profiles, the trust database contains a configuration of trust policy that is set on the platform by a platform manager.

The main data flow in the NIMBLE Trust Evaluation Service is provided in Figure 5, while the detailed sequence diagrams of cross-service interactions is described in Section 4.5.

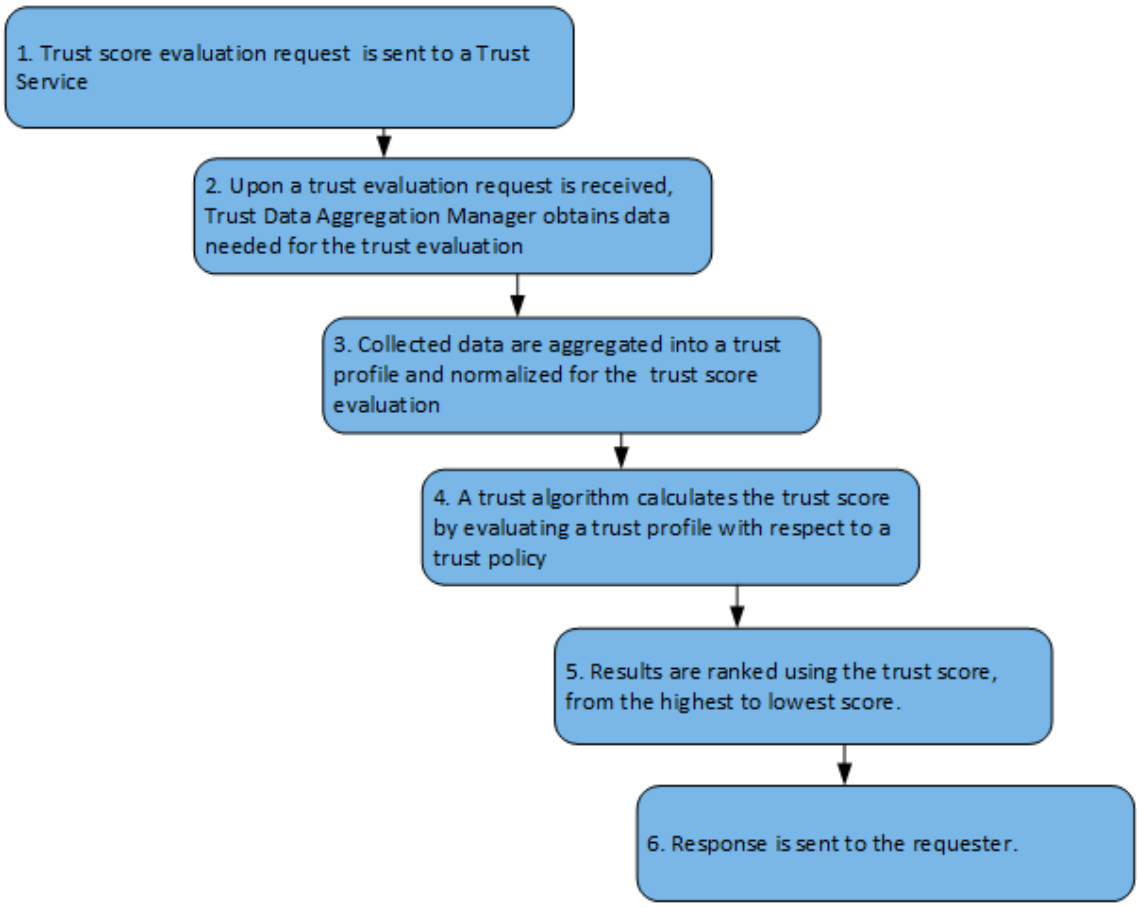

Figure 5 Sequence of interaction within NIMBLE Trust Evaluation Service

4.3 Conceptual Model of Trust and Trust Elements

The conceptual trust model in NIMBLE is based on a generic trust ontology that was developed within the COMPOSE FP7 project (see: http://www.compose-project.eu and [VUGA14]). The trust model introduced in COMPOSE/[VUGA14] is generic and was built to support requirements that are similar to those identified in the NIMBLE ecosystem. The model captures main trust concepts including an Agent (generic concept of any kind of entity with a trust score), TrustRelationship between two parties (namely, trustingParticipant and trustedParticipant), TrustCriteria (which represents a trust policy expressed as logical conjunction or disjunction of weighted TrustAttributes), and a TrustProfile that is as an aggregation of TrustAttributes belonging to certain Agent. A TrustAttribute can be a quantified MeasurableTrustAttribute, such as Rating or Popularity score, or descriptive NonMeasurableTrustAttribute such as quality assurance certification or security description. The model is a vocabulary to uniformly represent the trust policies, on the one hand, and trust profiles of entities on the other.

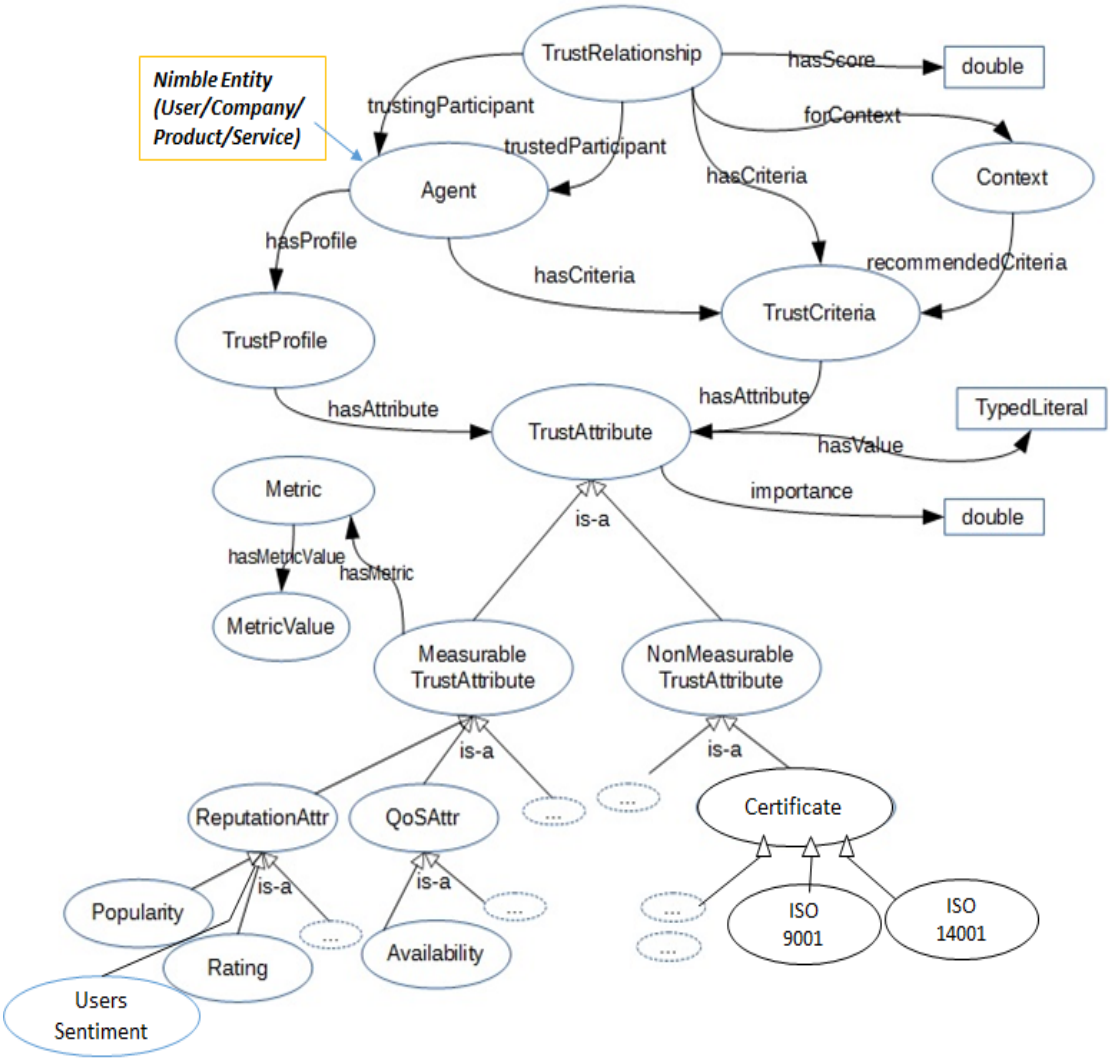

Figure 6 Conceptual model of trust and trust elements

Nimble Trust Elements:

Trust in the NIMBLE context can be considered as an estimated measure of the degree of trustworthiness in a B2B relationship between sellers and buyers. Considering the COMPOSE generic trust model, a buyer in NIMBLE acts as a trustingParticipant, while a seller in NIMBLE acts as a trustedParticipant in a TrustRelationship between two sides (Agents). A trustworthiness of the B2B relationship is quantified by a trust policy (TrustCriteria) evaluation.

Choosing the appropriate metrics (or trust attributes) for trust rating of vendors on multi-sided platforms such as NIMBLE is an important aspect of the platform management through its lifecycle. There can be a variety of metrics for trust rating, but not all are available or equally important across all stages of the life cycle of the platform.

We have made an analysis of possible metrics for different life cycle phases of the NIMBLE platform, and these are presented in Appendix 2.

For example, during the start-up phase of the platform, a trust rating can be based on:

- **Company registration profile completeness percentage**. Companies may be required to fill their profile with various details, which, if all available and verified, give a higher rating of trust. A company registration process can start by asking for basic information such as a company name, year of registration, verified VAT number or legal address, and then through several steps ask for more evidence of the company history and its trustworthiness. These additional details may include a list of company certifications by recognized certification authorities, a list of past and future visible events (conferences, fairs, etc.)
- **Average response time** on customer inquiries. A company could be rated with a higher trust score if it shows good behaviour on the platform, in terms of acceptable customer inquiry response time that is defined by a platform manager (e.g. within 12 hours).
- **Average negotiation time**. A company could be rated with a higher trust score if it is able to successfully close business transactions within an acceptable time range defined by a platform manager (e.g. within 24 hours). If needed, an average negotiation response time can be further structured into an average response time to complete different steps of the business process (to close the offer, to sign-in the fulfilled contract, etc).

Then, during the growth phase, additional metrics can be introduced, for example:

- **Trading volume** of a company
- **Number of successful contracts** closed via the platform

And finally, a trust rating in a mature phase of the platform can additionally include:

- Collaboration aspects. **User rating scores and reviews** (rating of the quality of negotiation; quality of ordering (packing and dispatching), and quality of the contract fulfilment (as described in Part 5 of Appendix 2)
- **Company presence and activities on the platform** tracked by audit logs

Implementation details of NIMBLE Trust service are given in Section 4.5.

4.4 Trust Evaluation Methods

4.4.1 Trust Scoring

The trust score (*t)* is a real number in the range of 0 to 1. It quantifies the match between the requested trust attributes specified in a trust policy and those corresponding characteristics of NIMBLE entities. The *t=1* means that given NIMBLE entity is a fully trusted, while *t=0* means it is distrusted, in respect to the trust criteria (or, trust policy). The trust score is calculated by the following equation (1).

$$t = \frac{\sum_{tp}^{n} \left(Ev(tp, D) \cdot relevance(tp) \right)}{\sum_{tp}^{n} relevance(tp)}; for each tp \in C. t \in [0,1] \qquad (1)$$

C is a trust criteria. Tp is a trust criterion (desired trust property).
D is a trust profile of NIMBLE entity.

The relation is a weighted sum of values obtained by Ev(tp, D) function. *Ev(tp, D)* evaluates a set of NIMBLE entity trust attributes with regards to the given trust criterion *tp* and returns a normalized value on a scale between 0 and 1.
Normalization is required as trust attributes or criteria can be of different dimensions and numerical scales. We use a linear normalization $\frac{value(tp, D)}{valueMax(tp)}$, where *value(tp, D)* is a value of a NIMBLE entity's property corresponding to the *tp*, and *valueMax(tp)* is the maximum [possible] value of the *tp* indicator.

For example, assume a company reputation as a trust property, with the reputation index on an ordinal ['bad', 'medium', 'high'] scale with a relative degree of difference between possible reputation values. Then, if a trust criterion is "reputation, at least medium", and a NIMBE entity has reputation index 'medium', then normalization returns 0.66, as $\frac{relativeDegreeOf'medium'}{relativeDegreeOf'high'} = \frac{2}{3} = 0.66$.
Or, if average rating, as a trust metrics, is in a range of [1, 10], and a NIMBE entity has average rating equal 5 , then normalized value of the rating will be $\frac{5}{10} = 0.5$.

An *Ev(tp,D)* evaluation in certain cases may require a more advanced computation, including semantic similarity computation to evaluate matches between offered and desired descriptive trust-related attributes. For example, a trust policy may state that companies with ISO certificates are more trusted than those without ISO certificates. In the platform, there can be a lightweight IS-A taxonomy of ISO certificates for description of company certifications. Therefore, for the trust score calculation there should be a way to evaluate semantic similarity of offered and by-trust-policy–expected certifications. A Trust Service uses a Semantic Measurement Library (SML) (see www.semantic-measures-library.org/), which offers a number of different algorithms for measuring semantic similarity between concepts.

4.4.2 Trust Ranking

The trust engine computes a NIMBLE entity's trustworthiness on a scale relative to other NIMBLE entities. This is required by the Search capability of the platform in cases where more than one company/ product/ service satisfies the search criteria of manufacturers, and hence, the manufacturers want to apply additional filtering or sorting to choose an entity with the highest levels of trust and reputation.

The ranking is defined as a Multi-Criteria Decision Making (MCDM) problem of different alternatives selection, with NIMBLE entities being considered as the alternatives, and with a given trust policy as a decision criterion. We implemented the ranking using two well-known MCDM methods: (1) Weighted sum model, and (2) Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS).

The weighted sum model is actually used for the trust scoring and different NIMBLE entities are just ranked according to their trust score, from the highest to the lowest score.

Using TOPSIS, the trust engine ranks NIMBLE entities in two steps. The first step is normalizing and weighting the evaluated score for each trust criterion, for each NIMBLE entity present in a ranking set. The second step is resolving an entity rank by calculating its relative geometric distance from the positive ideal and negative ideal solution. The positive ideal solution is one with the best score in each criterion, while the negative ideal solution is one with the worst score in each criterion. Alternatives are ranked according to the similarity to the best solution. The similarity is 1 only if the entity has the best trust offerings, while similarity is 0 only if the entity has the worst trust offering in all evaluated trust metrics.

4.5 Trust Service Design and Implementation

This section provides **implementation details of selected use-cases that involve trust management in NIMBLE**. The selection of use-cases and thus implementation is based on the Analysis of the NIMBLE Platform UIs and Services to Support User Ratings and Review Management (Appendix 2 of this document). This analysis has defined several use-cases that support user ratings and overall trust management in NIMBLE:

1) Company Registration and Profile Completeness
2) Product Publishing and Product Profile Completeness
3) Search with trust-based ranking/filtering, including Providers/Product Rating details
4) Rating and review of successfully ended business processes
5) Rating and review of unsuccessful negotiations
6) Trust policy management

Further in this section we first introduce the wireframes of needed user interfaces for the selected use-cases, then we provide the details of the NIMBLE trust database model, component and sequence diagrams, and finalize it with the REST interface specification of Trust Service.

Note that for the NIMBLE platform release 5.0, we have designed and implemented user ratings and reviews, and correspondent trust score calculations considering **those NIMBLE platform participants which are registered as either sellers or buyers, and their business processes**. While user ratings, reviews and trust scores for sellers are displayed at the NIMBLE platform, the correspondent user ratings, reviews and trust scores for buyers are kept internally for the purpose of platform participants management and possible negotiation resolutions.

The rating of individual products is not designed for the NIMBLE platform release 5.0. However, the current backend *Trust Service* can be extended to support trust scoring and ranking of products in the future.

The *Trust Service* in NIMBLE is implemented as an open source, licence free software and is available at: https://github.com/nimble-platform/trust-scoring-service

4.5.1 Overview of UI Frames

4.5.1.1 UI Company Registration / Profile Completeness UC

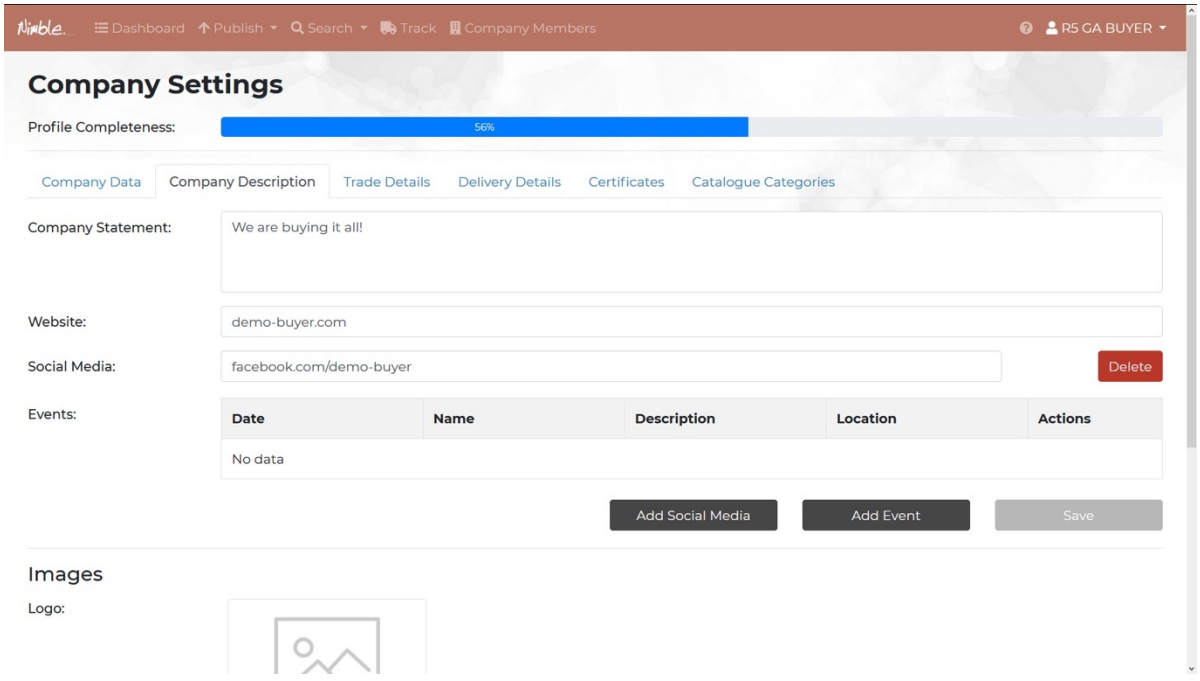

Figure 7 UI Company registration

4.5.1.2 UI for successful business process rating

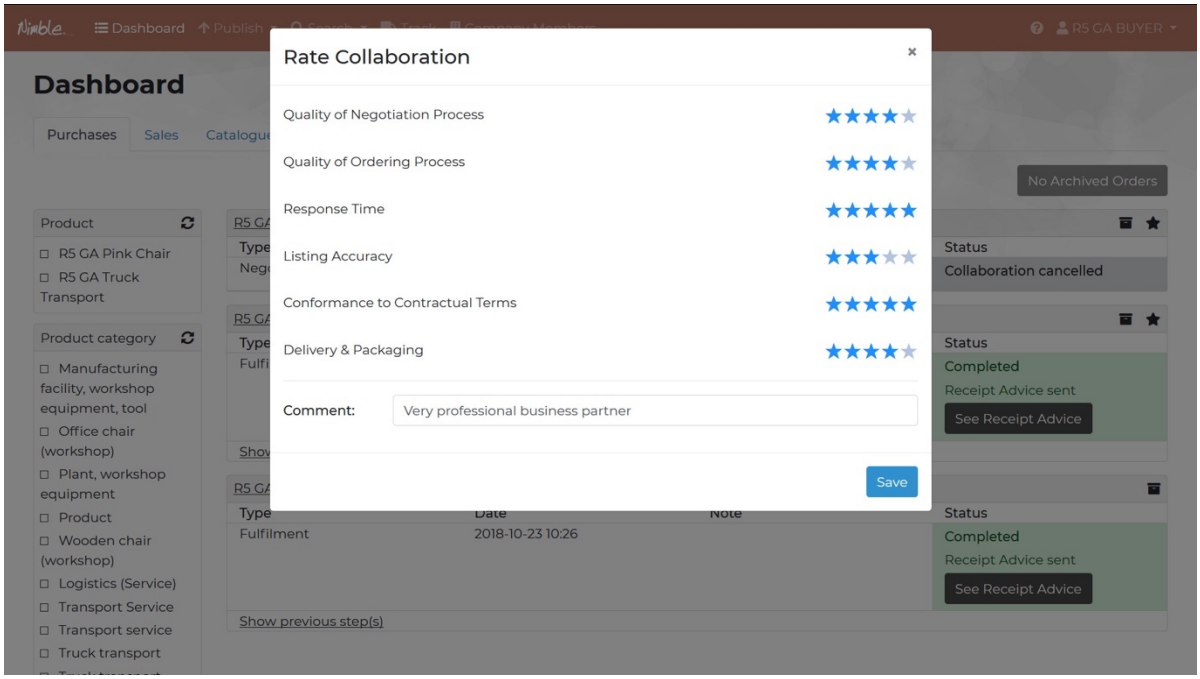

Figure 8 UI Rating of successfully completed business transaction

4.5.1.3 UI for rating of cancelled negotiation

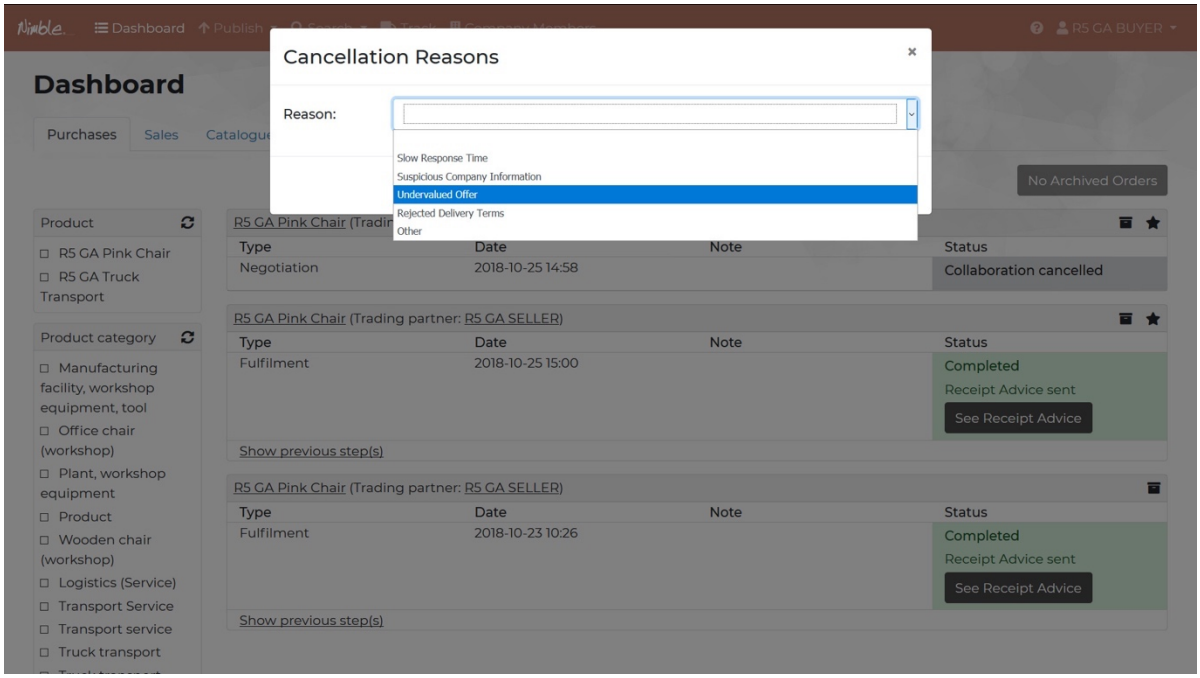

Figure 9 UI Rating cancelled business transaction

4.5.1.4 UI Search and Filtering

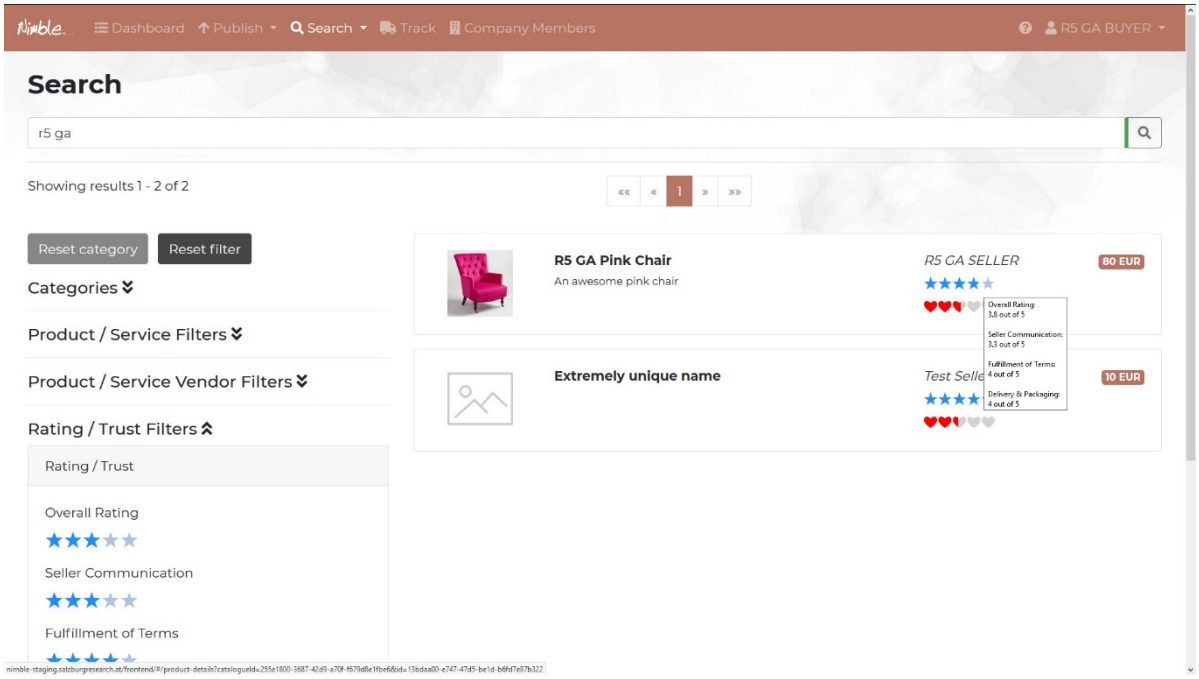

Figure 10 UI Search and filtering

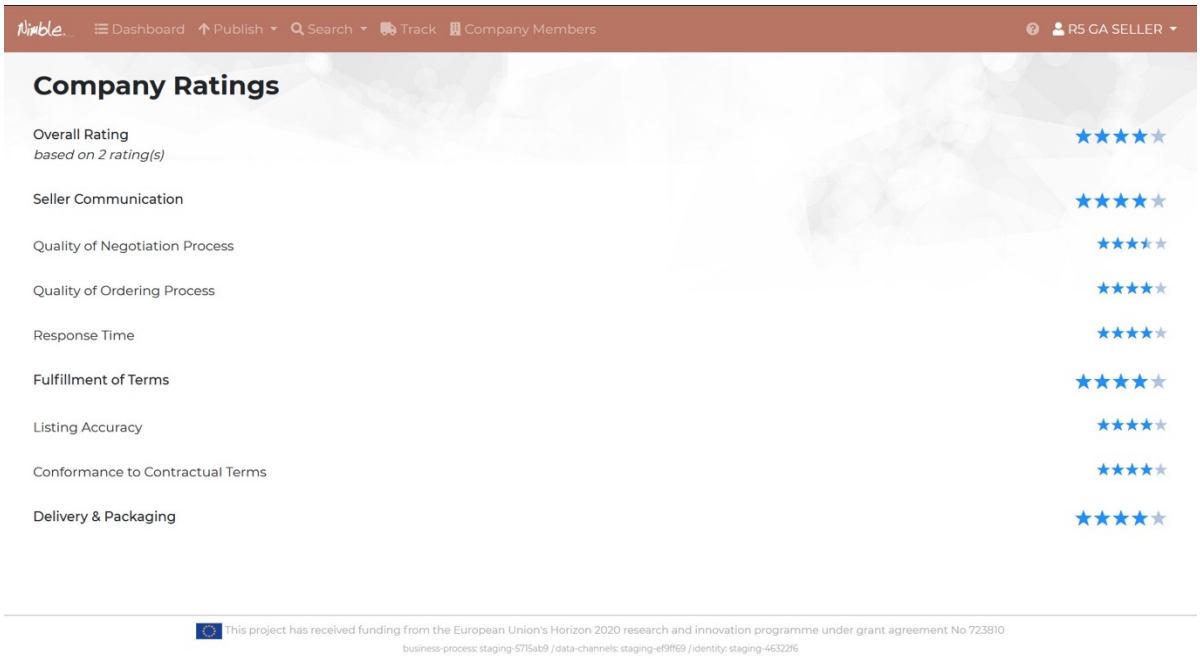

Figure 11 UI Company Ratings

4.5.1.5 UI Trust Policy Configuration

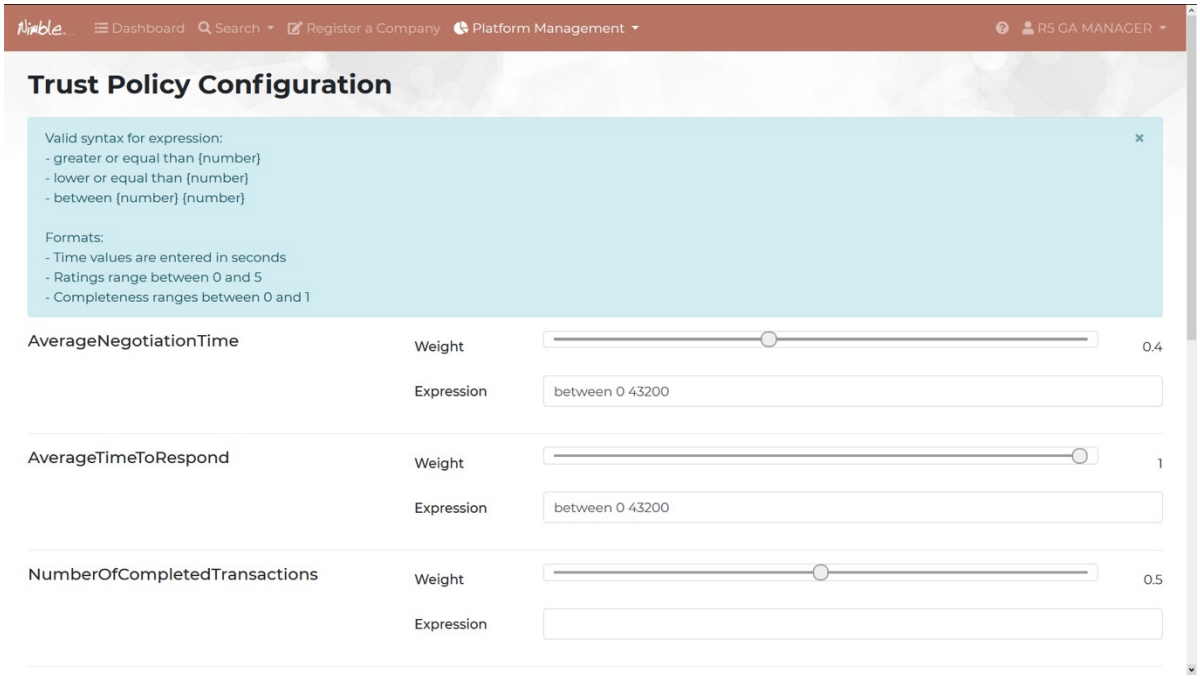

Figure 12 UI Trust policy configuration

4.5.2 Data Model Extensions to Support Trust Elements in NIMBLE

As overall NIMBLE data model is based on an UBL Schema, and as different microservices of NIMBLE platform are responsible for management of different parts of trust-related elements, we have provided a mapping of the desired trust elements to the corresponding NIMBLE's UBL data model. A data model of trust elements is specified based on Appendix 2 - Analysis of the NIMBLE Platform UIs and Services to Support User Ratings and Review Management.

Table 3 Trust elements mapping to UBL data model

Trust Element	UBL Data Model Element	Responsible Microservice/Database
CompDetails		
IDCompany	Party/ID	Identity Service @Persisted Field
CompanyLegalName	Party/Name	Identity Service @Persisted Field
VATNumber	Party/PartyTaxScheme/CompanyID[TaxScheme/TaxTypeCode=VAT]	Identity Service @Persisted Field
VerificationInfo	QualifyingParty/BusinessIdentityEvidenceID	Identity Service @Persisted Field
AddressStreet	Party/PostalAddress/StreetName	Identity Service @Persisted Field
AddressBuildingNumber	Party/PostalAddress/BuildingNumber	Identity Service @Persisted Field
City	Party/PostalAddress/CityName	Identity Service @Persisted Field
PostalCode	Party/PostalAddress/PostalZone	Identity Service @Persisted Field
Country	Party/PostalAddress/Country/Name	Identity Service @Persisted Field
BusinessType	Party/IndustryClassificationCode	Identity Service @Persisted Field
BusinessKeywords	QualifyingParty/BusinessClassificationScheme/Description	Identity Service @Persisted Field
YearOfCompanyRegistr	QualifyingParty/OperatingYearsQuantity	Identity Service @Persisted Field

NegotiationTime	Party/QualityIndicator[QualityParameter=NegotiationTime]/Quantity	BusinessProcessService @Calculated Field + TrustService @Persisted Field
TrustCompletnessProfile	Party/QualityIndicator[QualityParameter=ProfileCompletness]/Quantity	IdentityService @Calculated Field + TrustService @Persisted Field
TrustResponseTime	Party/QualityIndicator[QualityParameter=ResponseTime]/Quantity	BusinessProcessService @Calculated Field + TrustService @Persisted Field
TrustCompletOfDetails	QualifyingParty/ParticipationPercent	Identity Service@Calculated/Persisted Field + TrustService @Persisted Field?
TrustCompanyTrustScore	Party/QualityIndicator[QualityParameter=TrustScore]/Quantity	TrustService @Persisted Field
TrustCompanyRating	Party/QualityIndicator[QualityParameter=CompanyRating]/Quantity	TrustService @Persisted Field
TrustTradingVolume	Party/QualityIndicator[QualityParameter=TradingVolume]/Quantity	TrustService @Persisted Field
TrustOverallSellerCommun	Party/QualityIndicator[QualityParameter=SellerCommunication]/Quantity	TrustService @Persisted Field
TrustOverallFullfilOfTerms	Party/QualityIndicator[QualityParameter=FullfillmentOfTerms]/Quantity	TrustService @Persisted Field
TrustOverallDelPackaging	Party/QualityIndicator[QualityParameter=DeliveryPackaging]/Quantity	TrustService @Persisted Field
NumberOfTransactions	Party/QualityIndicator[QualityParameter=NumberOfTransactions]/Quantity	BusinessProcessService @Calculated Field + TrustService @Persisted Field
CompDescription		
CompanyStatement	QualifyingParty/EconomicOperatorRole/RoleDescription	Identity Service @Persisted Field
CompanyPhotosList	Party/DocumentReference[DocumentType=CompanyPhoto]/Attachment	Identity Service @Persisted Field
3DVirtualTour	QualifyingParty/BusinessClassificationScheme/URI	Identity Service @Persisted Field
Website	Party/WebsiteURI	Identity Service @Persisted Field
SocialMediaList	Party/Contact/OtherCommunication	Identity Service @Persisted Field
UpcomingEventsList	QualifyingParty/Event[CompletionIndicator=false]	Identity Service @Persisted Field
PastEventsList	QualifyingParty/Event[CompletionIndicator=true]	Identity Service @Persisted Field
TrustCompletnessOf CompanyDesc	Party/QualityIndicator[QualityParameter= CompletenessOfCompanyDescription]/Quantity	Identity Service@Calculated/ Persisted Field + TrustService @Persisted Field?
UpcomingEvent / PastEvent		
UpcomingEventName	QualifyingParty/Event/IdentificationID	Identity Service @Persisted Field
UpcomingEventPlace	QualifyingParty/Event/Address	Identity Service @Persisted Field
UpcomingEventDateFrom	QualifyingParty/Event/DurationPeriod/StartDate	Identity Service @Persisted Field
UpcomingEventDateTo	QualifyingParty/Event/DurationPeriod/EndDate	Identity Service @Persisted Field
UpcomingEventDescription	QualifyingParty/Event/Description	Identity Service @Persisted Field
CompCertifList		
IDCertifList	Party/Certificate/ID	Identity Service @Persisted Field
CertificateType	Party/Certificate/CertificateTypeCode	Identity Service @Persisted Field
CertificateName	Party/Certificate/CertificateType	Identity Service @Persisted Field
ValidityPeriod	Party/Certificate/DocumentReference/ValidityPeriod	Identity Service @Persisted Field
CertificateImage	Party/Certificate/DocumentReference/Attachment/ EmbeddedDocumentBinaryObject	Identity Service @Persisted Field
CertificateDescription	Party/Certificate/Remarks	Identity Service @Persisted Field
TrustCompletness OfCompCertif	Party/QualityIndicator[QualityParameter= TrustCompletnessOfCompanyCertif]/Quantity	Identity Service@Calculated/ Persisted Field + TrustService @Persisted Field?
CompTradeDetails		
`	QualifyingParty/Declaration[DeclarationTypeCode= MarketOptions]/Name	Identity Service @Persisted Field
AcceptedDelivery TermsOptions	QualifyingParty/Declaration[DeclarationTypeCode= AcceptedDeliveryTermsOptions]/Name	Identity Service @Persisted Field
AcceptedPaymentTypeOptions	QualifyingParty/Declaration[DeclarationTypeCode= AcceptedPaymentTypeOptions]/Name	Identity Service @Persisted Field

TrustCompletnessOf CompTradeDetails	Party/QualityIndicator[QualityParameter =TrustCompletnessOfCompTradeDetails]/Quantity	Identity Service@Calculated/ Persisted Field + TrustService @Persisted Field?
NegotiationHist		
IDNegotiationHistory	QualifyingParty/CompletedTask/AssociatedProcessInstanceID	BusinessProcessService @Persisted Field
TradingComp	QualifyingParty/CompletedTask/RecipientCustomerParty/Party/ID	BusinessProcessService @Persisted Field
NegOpenDate	QualifyingParty/CompletedTask/Period/StartDate	BusinessProcessService @Persisted Field
NegClosingDate	QualifyingParty/CompletedTask/Period/EndDate	BusinessProcessService @Persisted Field
NegotiationStatus	QualifyingParty/CompletedTask/Description	BusinessProcessService @Persisted Field
NegotiationRating (for successful negotiation)		
NumberOfStars	QualifyingParty/CompletedTask/EvidenceSupplied [ID=NumberOfStars]/ValueDecimal	BusinessProcessService @Persisted Field
QualityOfThe NegotiationProcess	QualifyingParty/CompletedTask/EvidenceSupplied [ID=QualityOfTheNegotiationProcess]/ValueDecimal	BusinessProcessService @Persisted Field
QualityOfThe OrderingProcess	QualifyingParty/CompletedTask/EvidenceSupplied [ID=QualityOfTheOrderingProcess]/ValueDecimal	BusinessProcessService @Persisted Field
ResponseTime	QualifyingParty/CompletedTask/EvidenceSupplied [ID=ResponseTime]/ValueDecimal	BusinessProcessService @Persisted Field
SellerCommunication	QualifyingParty/CompletedTask/EvidenceSupplied [ID=SellerCommunication]/ValueDecimal	BusinessProcessService @Persisted Field
ProductListingAccuracy	QualifyingParty/CompletedTask/EvidenceSupplied [ID=ProductListingAccuracy]/ValueDecimal	BusinessProcessService @Persisted Field
ConformanceToOther AgreedTerms	QualifyingParty/CompletedTask/EvidenceSupplied [ID=ConformanceToOtherAgreedTerms]/ValueDecimal	BusinessProcessService @Persisted Field
Fulfillment OfContractualTerms	QualifyingParty/CompletedTask/EvidenceSupplied [ID=FulfillmentOfContractualTerms]/ValueDecimal	BusinessProcessService @Persisted Field
DeliveryAndPackaging	QualifyingParty/CompletedTask/EvidenceSupplied [ID=DeliveryAndPackaging]/ValueDecimal	BusinessProcessService @Persisted Field
ReviewDescription	QualifyingParty/CompletedTask/Comment/Comment	BusinessProcessService @Persisted Field
NegotiationComments (for cancelled negotiations)		
NameCommentType	QualifyingParty/CompletedTask/Comment/TypeCode	BusinessProcessService @Persisted Field
BuyerOrSellerCommentType	QualifyingParty/CompletedTask/Comment/TypeCode[@name]	BusinessProcessService @Persisted Field
CommentValue	QualifyingParty/CompletedTask/Comment/Comment	BusinessProcessService @Persisted Field

As it can be seen from the table and column 'Responsible microservice', the trust elements of NIMBLE platform are dispersed across the different microservices / databases of the platform.

- The trust elements related to the company profile completeness are under the responsibility and database of Identity Service, as this service is in charge of registration of companies on the platform.
- Then, elements related to user ratings, reviews, and comments about business processes are under the responsibility and database of Business Process Service.
- Trust Service has its own database for persistence of providers trust profiles and trust scores, and for persistence of a trust policy configuration. Database transactions boundaries are kept within a single service (there are no two-phase commits) and synchronization of data between services/database is accomplished using a message publish-subscribe mechanism.

4.5.3 UML Component Diagram

The figure below shows a diagram of NIMBLE components (microservices) involved in the selected use-cases.

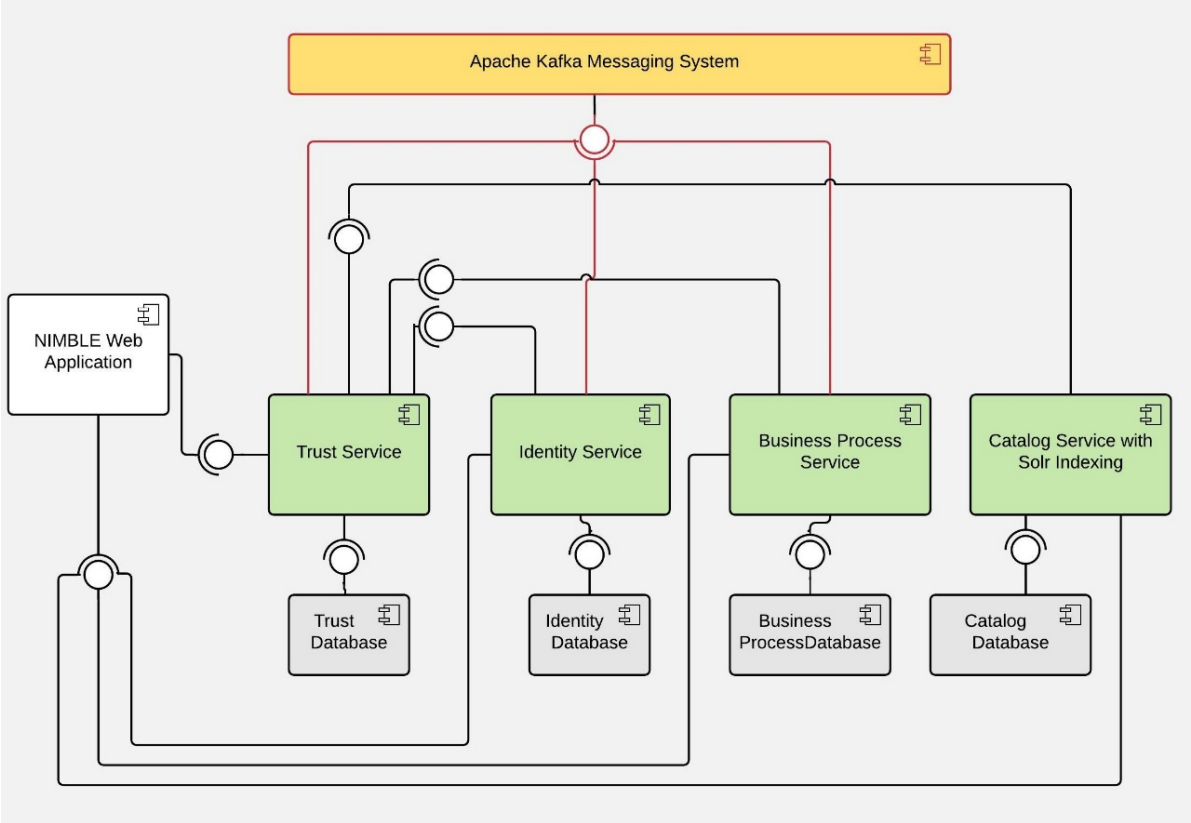

Figure 13 UML Component Diagram of Trust Management in NIMBLE

The main component responsible for the trust evaluation and management is the *Trust Service*. It provides an interface to a NIMBLE Web application for the trust governance tasks. Then it provides an interface to the *Catalog Service* for Solr indexing of trust metrics in order to enable filtering and ranking of product search results according the trust metrics.

The *Identity Service* provides an interface to the Trust Service in order to supply profile completeness details from the identity service into the trust database.

The *Business Process Service* provides an interface to the Trust Service in order to supply the trust service with ratings of business processes, trading and transaction volumes, and with other relevant statistics.

All services have their own databases. Change-of-data notifications for fault-tolerant and non-blocking synchronization of trust-related data and recalculation of trust scores are managed by an Apache Kafka messaging system.

Kafka acts as a topic-based publish-subscribe system. For example, there can be a topic named "company-details-changes" with Identity Service as a data change notifications publisher.

Using Kafka, a Trust Service can subscribe to the "company-details-changes" topic and will receive (consume) the company data change notifications. A receipt of such a message will further trigger the trust score update process. Similarly, there can be a "ratings-update" topic with feeds produced by a Business Process Service.

4.5.4 UML Sequence Diagrams

4.5.4.1 Company Profile Completeness

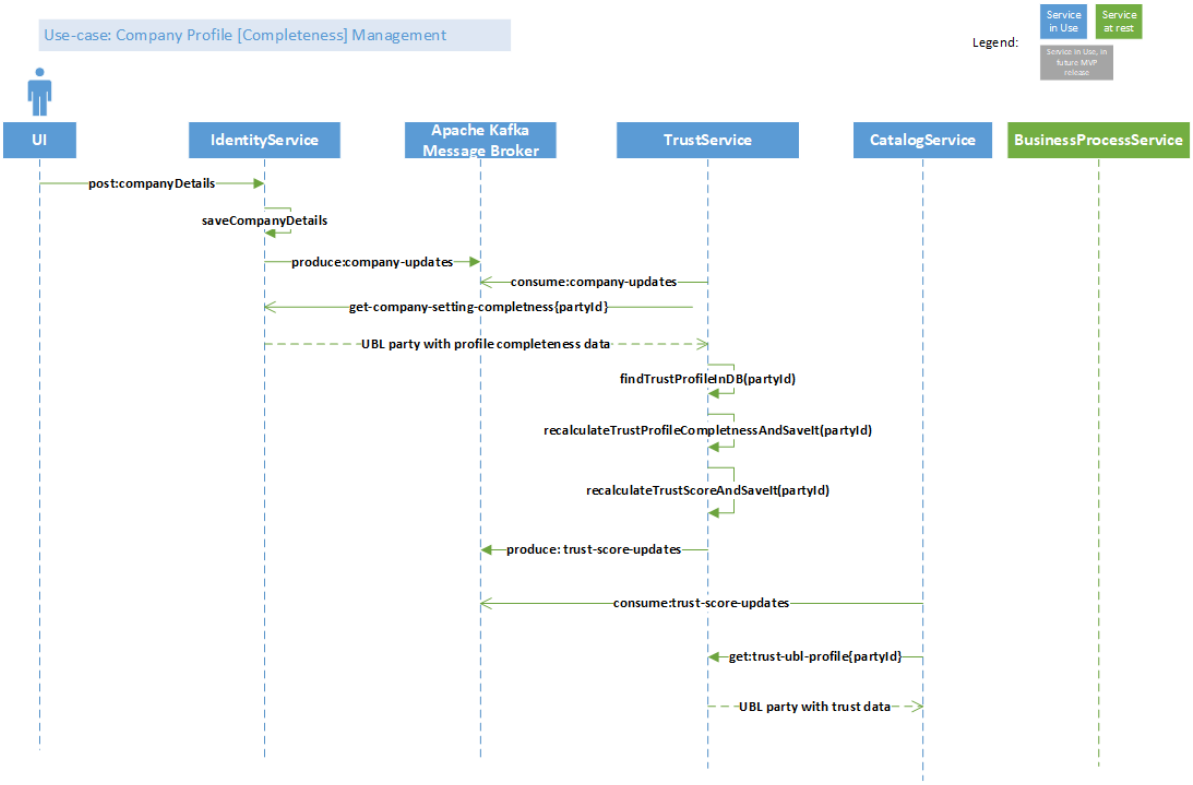

Figure 14 Sequence Diagram: Company Profile Completeness

4.5.4.2 Ratings and Reviews for Cancelled Negotiation

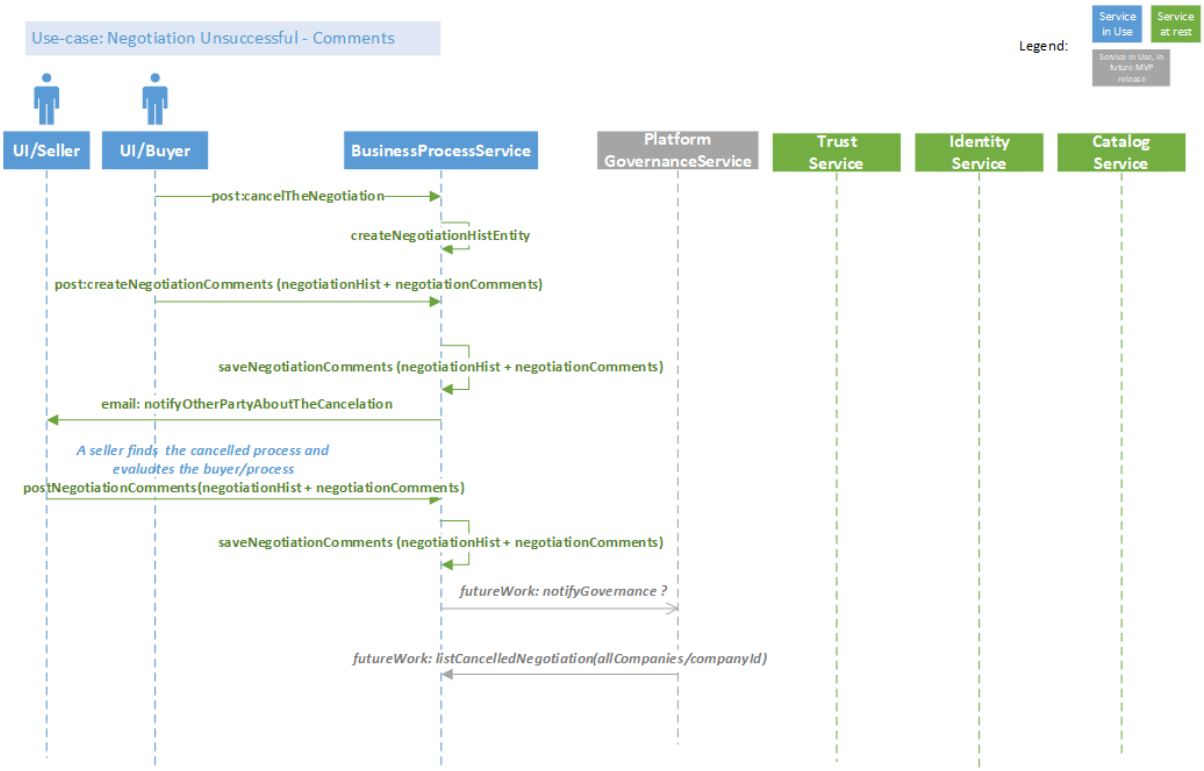

Figure 15 Sequence Diagram: Review of cancelled negotiation

4.5.4.3 Ratings and Reviews for Successful Negotiation

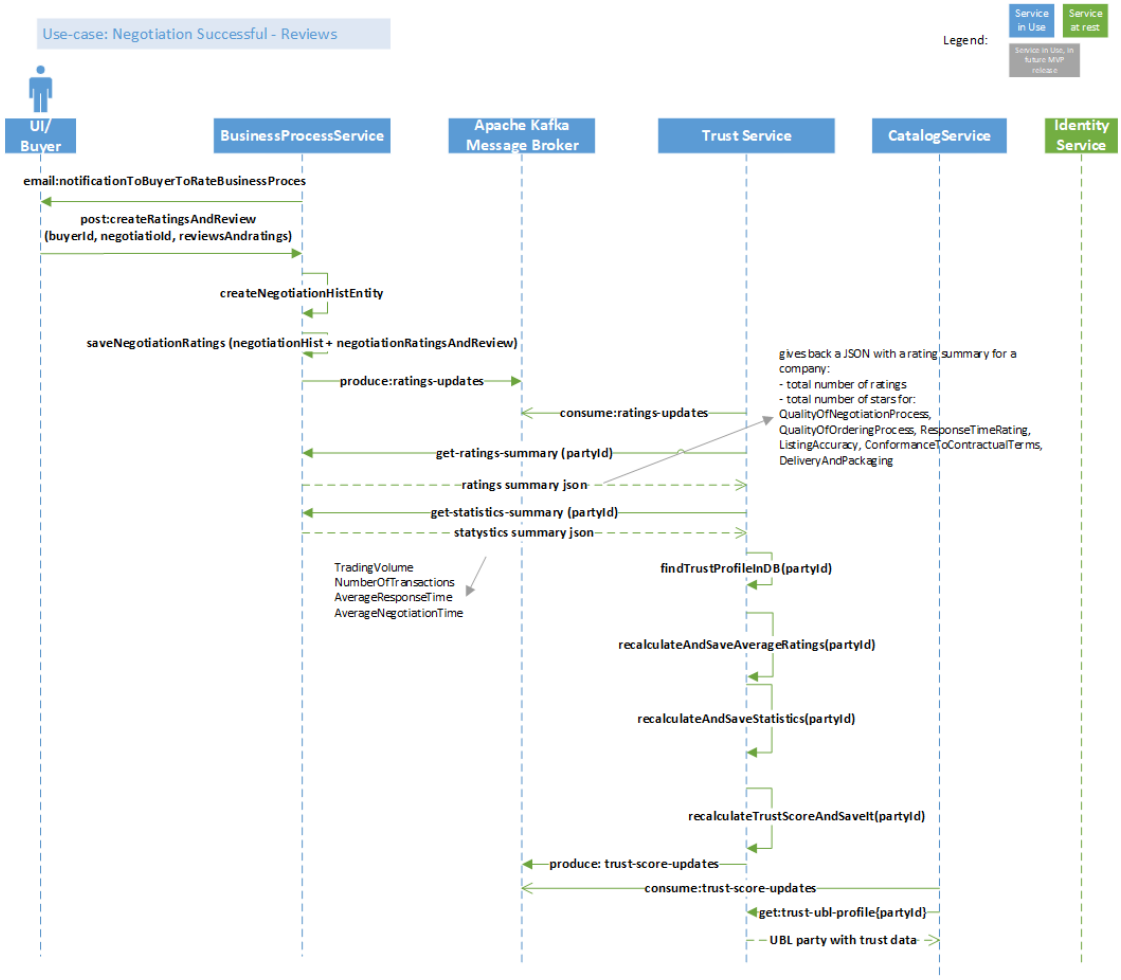

Figure 16 Sequence Diagram: Rating of successful negotiation

4.5.4.4 Global Trust Policy Management

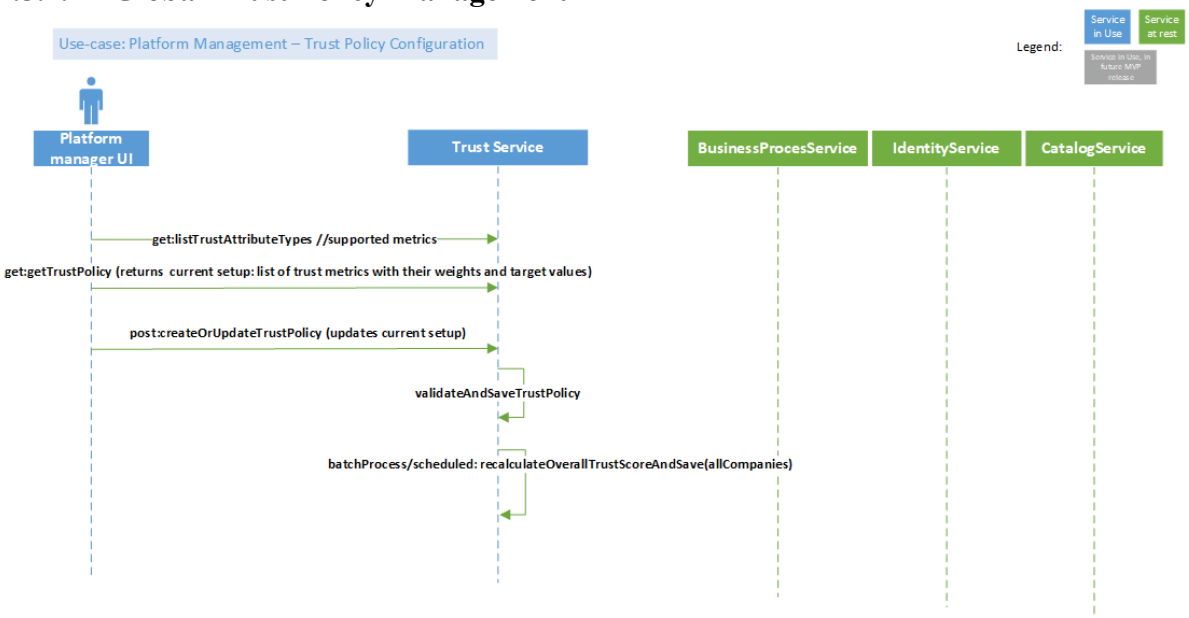

Figure 17 Sequence Diagram: Trust policy management

4.5.4.5 Trust-based ranking of search results

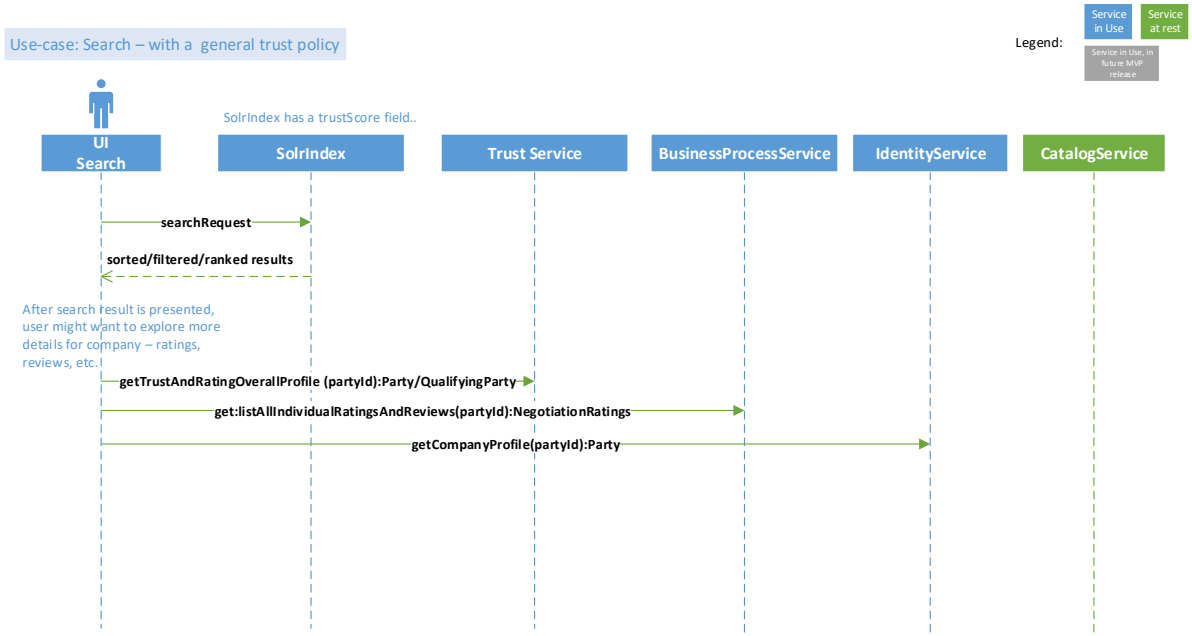

Figure 18 Sequence Diagram: Trust-based ranking of search results

4.5.4.6 Trust-based ranking of search results using customized trust policy

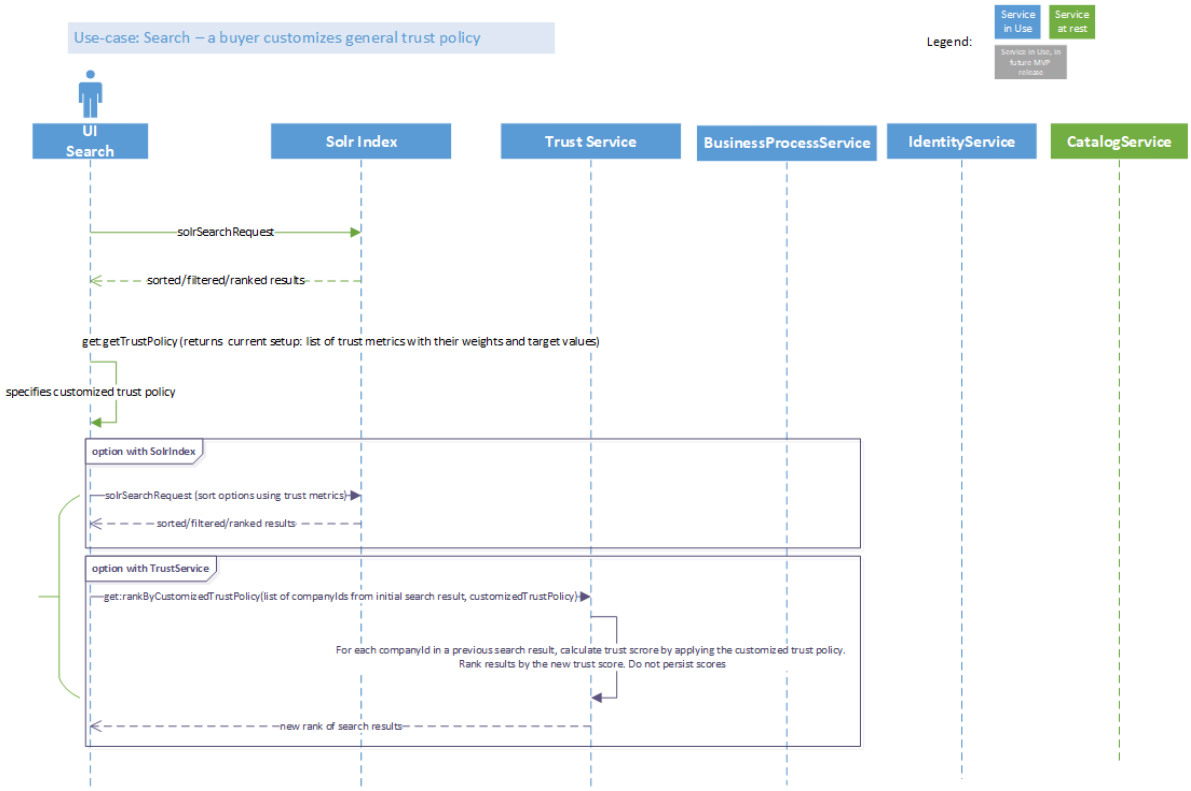

Figure 19 Sequence Diagram: Trust-based ranking using customized trust policy

4.5.5 Implementation Technology

The NIMBLE Trust Service is implemented as a microservice system using Spring Boot and Spring Cloud technologies to adhere to the NIMBLE platform architecture.

4.5.6 Internal Database Model

The Trust Service has its own database for aggregation of trust profiles and for persistence of a global trust policy. The database of the Trust Service implements the conceptual trust model introduced in Section 4.3, as shown in the Figure bellow.

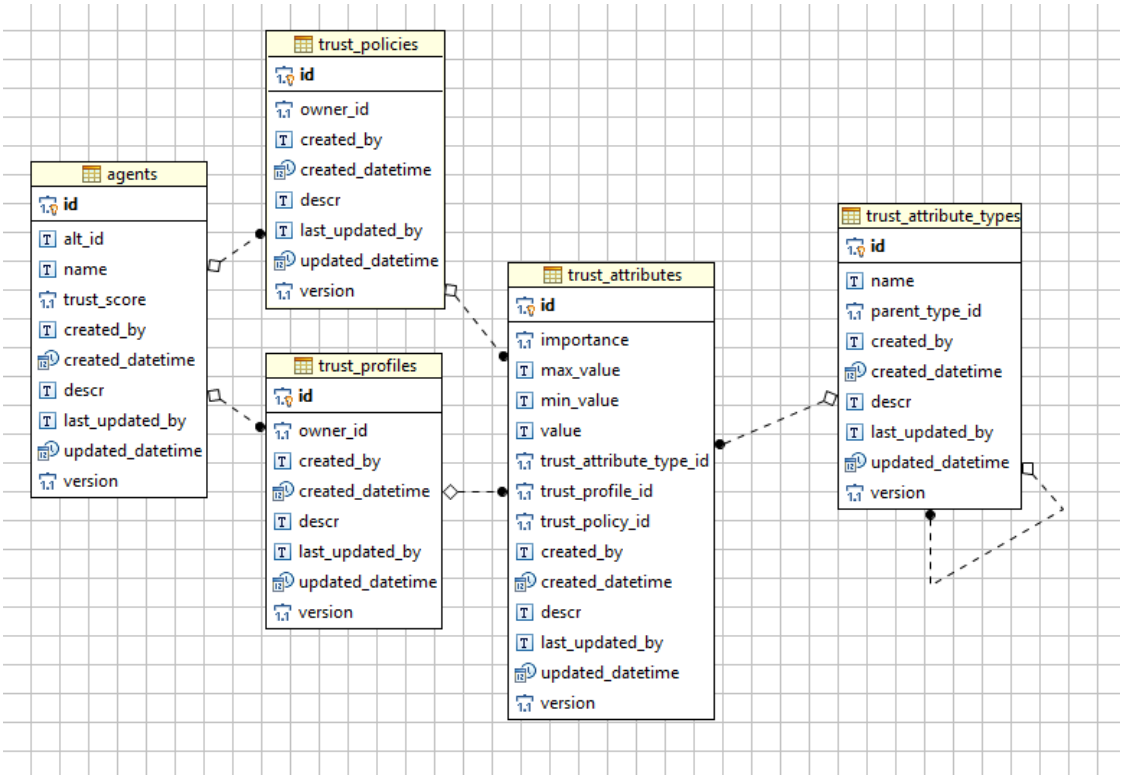

Figure 20 Internal database of the Trust Service

4.5.7 REST Interface Documentation

The REST interface of Trust Services provides a set operations for (1) management of trust policy in the platform, (2) trust scoring and ranking.

4.5.7.1 Trust-policy-controller

Trust policy controller provides a set of operations for management of global trust policy. These are the operations:

- *GET /policy/global* -- Returns current global trust policy in JSON format. For example, it may return JSON like this one:

```
{
  "trustAttributes": [
    {
      "id": 18,
      "weight": 0,
      "expression": "",
      "attributeType": {
        "name": "NumberOfUncompletedTransactions"
      }
    },
    {
      "id": 21,
      "weight": 1,
      "expression": "between 0 42000",
      "attributeType": {
        "name": "AverageNegotiationTime"
      }
    },
    {
```

```
        "id": 22,
        "weight": 1,
        "expression": "",
        "attributeType": {
          "name": "OverallProfileCompletness"
        }
      },
      {
        "id": 16,
        "weight": 1,
        "expression": "",
        "attributeType": {
          "name": "OverallCompanyRating"
        }
      }
    ]
}
```

- *GET /metrictypes/all* -- Lists all supported 'root' trust metrics. It returns a JSON model with a list of root metric types. Please note that metrics and their implementations are built-in the platform. An introduction of new trust metrics requires an implementation of data aggregation for the new metric and metrics calculation. Successful return response model schema of this operation is:

```
{
  "hasSubTypes": true,
  "id": "string",
  "isRoot": true,
  "name": "string",
  "nameLocalized": "string"
}
```

- *POST /policy/global/initialize* -- Initializes a new global trust policy using built-in trust metrics definitions on the platform. Usually this operation is used for an initial creation of global trust policy.

- *POST /policy/global/update* -- This operation is responsible for a global trust policy update. Platform manager provides a new policy in a JSON form using the following schema:

```
{
  "id": 0,
  "recalculateScoresWhenUpdated": true,
  "trustAttributes": [
    {
      "attributeType": {
        "hasSubTypes": true,
        "id": "string",
        "isRoot": true,
        "name": "string",
        "nameLocalized": "string"
      },
      "expression": "string",
      "id": 0,
      "weight": 0
    }
]
}
```

For an example, if platform manager wants to change an acceptable average negotiation time to 12 hours, then the new policy will state that AverageNegotiationTime is *between 0 and 43200*:

```
{
```

```
  "trustAttributes": [
    {
      "id": 18,
      "weight": 0,
      "expression": "",
      "attributeType": {
        "name": "NumberOfUncompletedTransactions"
      }
    },
    {
      "id": 21,
      "weight": 1,
      "expression": "between 0 43200",
      "attributeType": {
        "name": "AverageNegotiationTime"
      }
    },
    {
      "id": 22,
      "weight": 1,
      "expression": "",
      "attributeType": {
        "name": "OverallProfileCompletness"
      }
    },
    {
      "id": 16,
      "weight": 1,
      "expression": "",
      "attributeType": {
        "name": "OverallCompanyRating"
      }
    }
  ]
}
```


4.5.7.2 Trust-score-controller

Trust score controller provides a set of operations for trust scoring and ranking of NIMBLE companies (sellers). These are the operations:

- *POST /notifyChange* -- This non-blocking operation should be used to notify the trust service about trust-related data changes in other services. After calling this operation, trust service will collect updates and will recalculate the trust score. Valid options for changeType are 'ratings-update','company_details', 'company_description', 'company_certificates', 'company_trade'. Returns 200 if successful.

- *POST /calculate/global/{partyId}* -- This operation executes the trust calculation for a NIMBLE providers with partyId, using a global trust policy. Returns 200 if successful.

- *POST /recalculate/batch* -- Non-blocking, asynchronous batch operation that recalculates trust score using global policy for all NIMBLE providers that are available (that have trust profiles) in a trust-service database. This operation is typically called after a global trust policy has been changed, in order to recalculate the scores according to new trust policy. Returns 200 if successful.

- *POST /fetch-all-calculate/batch* -- This non-blocking, asynchronous batch operation creates a trust profile for all NIMBLE providers that are register in a identify-service database. Returns 200 if successful.

- *GET /party/{partyId}/trust* -- This operations returns a UBL Party model for a NIMBLE provider with partyId populated with trust-related values such as trust score, company rating, average response time, average negotiation time, trading volume, number of transactions, and other available values aggregated by the platform. A return response JSON example is provided here:


```
{
  "id": "5",
  "nace": [],
  "partyTaxScheme": [],
  "certificate": [],
  "qualityIndicator": [
    {
      "qualityParameter": "COMPLETENESS_OF_COMPANY_GENERAL_DETAILS",
      "quantity": {
        "value": 0.1
      }
    },
    {
      "qualityParameter": "COMPLETENESS_OF_COMPANY_DESCRIPTION",
      "quantity": {
        "value": 0.5
      }
    },
    {
      "qualityParameter": "COMPLETENESS_OF_COMPANY_CERTIFICATE_DETAILS",
      "quantity": {
        "value": 0.66
      }
    },
    {
      "qualityParameter": "COMPLETENESS_OF_COMPANY_TRADE_DETAILS",
      "quantity": {
        "value": 0.6
      }
    },
    {
      "qualityParameter": "PROFILE_COMPLETENESS",
      "quantity": {
        "value": 0.465
      }
    },
    {
      "qualityParameter": "SELLER_COMMUNICATION",
      "quantity": {
        "value": 0
      }
    },
    {
      "qualityParameter": "FULFILLMENT_OF_TERMS",
      "quantity": {
        "value": 0
      }
    },
    {
      "qualityParameter": "DELIVERY_PACKAGING",
      "quantity": {
        "value": 0
      }
    },
    {
      "qualityParameter": "COMPANY_RATING",
      "quantity": {
        "value": 0
      }
    },
    {
      "qualityParameter": "TRUST_SCORE",
      "quantity": {
        "value": 0.0775
      }
```

```
      }
    ],
    "ppapDocumentReference": [],
    "documentReference": [],
    "industrySector": []
}
```

- *POST /calculate/custom* -- An operation for trust scoring and ranking of a set of NIMBLE providers by using a custom trust policy in JSON form. The Request body has the following JSON schema:

```
{
    "parties": [{ <Nimble-partyID>, <Nimble- partyID >,
     <Nimble- partyID >…}],
    "parameters": {
        "attributes": [<type, expression, weight>,
                       <type, expression, weight>…] },
    "strategy": <strategy>
}
```

A field "parties" is an array of ids of NIMBLE providers for which the trust score needs to be calculated according the supplied trust policy. The "attributes" is a specification of policy for trust evaluation, and it is represented as a JSON array of desired trust-related attributes, their desired values and weights. Then, the "strategy" field tells the trust engine which trust scoring strategy to apply. The strategy may be either a "standard" (refers to a weighted sum model) or "topsis" strategy.
A response of this operation is a JSON message with schema:
Response - Content-Type:application/json; Status code: 200

```
{
  "success": "true",
  "result": [ <entity trust score>, < entity trust score>]
}
```

where "result" is an array of < entity trust > attributes that contains a Nimble providers ID (field "Nimble-partyID"), its trust score (field "index") and its relative rank (field "rank") among other providers which were sent to the trust evaluation using this REST operation. Example response is:

```
{
"success": "true",
"result": [{
   "Nimble-partyID": "789",
      "index": 0.9,
      "rank": 1
   },
  {
 "Nimble- partyID ":"52",
 "index": 0.8,
 "rank": 2
}]
}
```

- *POST /filter/threshold* -- Operation for the trust filtering using a trust score threshold (pre-set at 0.5). NIMBLE providers whose trust score is less then a treshold will be filtered out from the result set. *Request Header Content-Type:application/json. Request Body is* same as in the /calculate/custom operation, but without the "strategy" attribute. A response content type is *:application/json; Status code: 200*

```
{
  "success": "true",
  "result": [ <Nimble-entityID>]
}
```

"result" is an array of <Nimble-entityID> attributes that are IDs of NIMBLE providers evaluated as trusted by the filter. Example response is

```
{
  "success": "true",
  "result": [
      {
```

```
        "Nimble-entityID":"52"
         },
         {
         "Nimble-entityID":"78"
          }
       ]
}
```

- *POST /filter/exclusion* -- Operation for filtering out NIMBLE providers whose trust profiles do not satisfy at least one of the trust-related attributes specified in a trust policy. For example, if trust policy states that provider average rating has to be minimally 3, then all the NIMBLE providers that do not satisfy the criteria will be evaluated as non-trusted (trust score will be zero).
  Request Header: Content-Type:application/json
  Request Body: same as the above, also without the "strategy" attribute.
  Response - Content-Type:application/json; Status code: 200 – JSON same as the JSON response from POST /trust/filter/threshold

  In a case of internal errors, the trust scoring controller responds a JSON message:

  Response: Content-Type:application/json; charset=UTF-8, Status Code: 500

```
{
  "success" : "false",
  "message" : "error message text here"
}
```

4.5.7.3 JSON Syntax for Trust Policy

As shown in the examples of the REST services request bodies, a trust policy can be specified using the following JSON syntax.

```
{
  "attributes": [
    {
      "type":  "trust attribute type name : string"
      "expression": "evaluation expression: string",
      "weight": double
    }
]
}
```

The field "attributes" captures a trust policy as an array of desired trust-related attributes, their desired values and their importance (weight 0 to 1). A vocabulary for the trust attribute types name (or trust metric names) is defined in NIMBLE Trust Data Model, and "type" refers to concepts in that model. Currently supported trust metrics are:

- AverageNegotiationTime
- AverageTimeToRespond
- NumberOfCompletedTransactions
- NumberOfUncompletedTransactions
- OverallCompanyRating
- OverallProfileCompletness
- TradingVolume

For each trust metric, its desired value, if needed, can be specified in a field "expression". Usually, the expression specifies the situation where the value of certain trust metric has to be

within some range in order to be evaluated as trusted or distrusted. Expressions must follow a custom, domain-specific syntax of numerical comparison operators: `equal {value}, less than {value}, greater than {value}, between {value one} {value two}.`

For example, a NIMBLE user or platform manager would express a trust expectation "I trust to providers which trading volume is greater than 50,000 Euro and which number of transactions is greater than 1000, and with average response time within 12 hours" by this expression:

```
trustAttributes": [
    {
      "weight": 1,
      "expression": "greater than 50000",
      "type": "TradingVolume"
    },
    {
      "weight": 1,
      "expression": "greater than 1000",
      "type": "NumberOfCompletedTransactions"
    },
    {
      "weight": 1,
      "expression": "between 0 43200",
      "type":"AverageNegotiationTime"
….
]
```

Therefore, if some provider has a trading volume lower than 50,000 euro, but still have more than 1000 transactions and responds within 12 hours, it be evaluated with a score 0 for the trading volume expectation, with a score 1 for a number of transactions, and with score 1 for a response time. An overall trust score, in this example case, with the same weight of trust metrics, would be ((0 x 1 + 1 x 1 + 1 x 1)/(1+1+1))=2/3 = 0.667.

4.5.8 GitHub Repository

A source code of the Trust Microservice is published on a GitHub, https://github.com/nimble-platform/trust-scoring-service, under Apache Licence 2.0[3]

[3] https://en.wikipedia.org/wiki/Apache_License

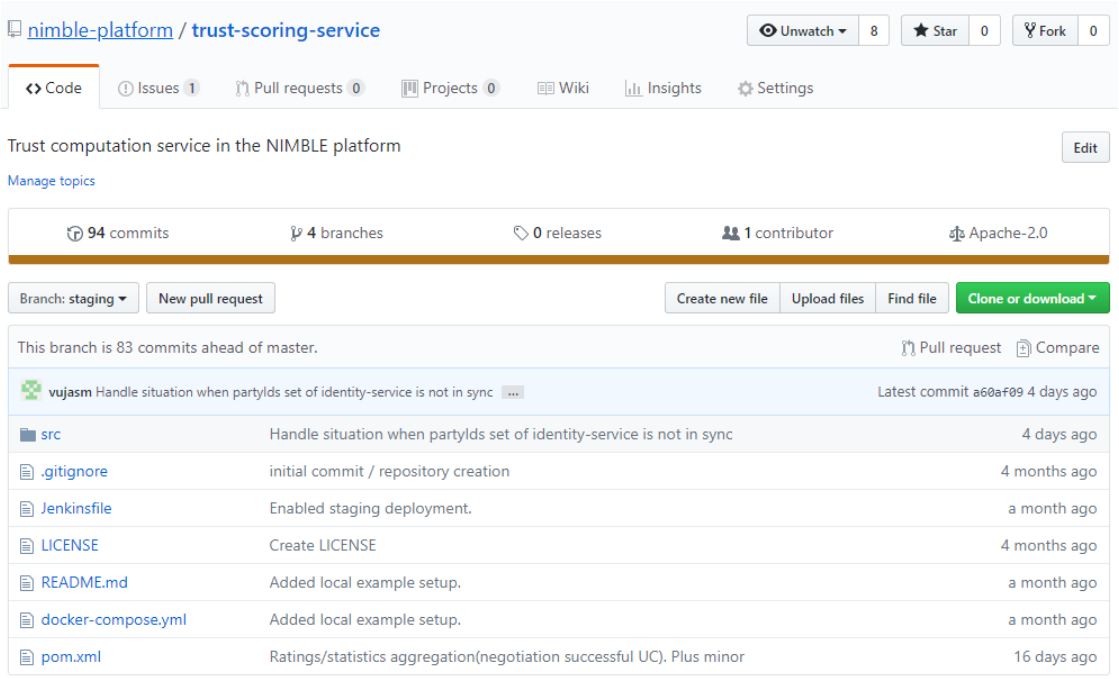

Figure 21 Github repository of Trust service

4.6 Demonstration

NIMBLE release 5.0 (October 2018) provided an implementation of all the following use-cases related to the trust management in NIMBLE:

1) Company Registration and Profile Completeness
2) User rating and review of successfully ended business processes
3) User rating and review of cancelled negotiations
4) Trust policy management
5) Search with trust-based ranking/filtering, including providers trust details

Our demonstration includes the following scenario:

A. Company Registration and Profile Completeness progress → Kafka notification about profile changes → Trust score update → Kafka notification about trust score update → Reindexing of Catalogues / Solr
B. Buyer's rating and review of successfully ended business processes → Kafka notification about a new rating → Trust score update → Kafka notification about trust score update → Reindexing of Catalogues/Solr
C. Buyer's and Seller's feedback of cancelled business processes
D. Product Search → Exploring search results using trust features for filtering and ranking of results → View trust measures and user reviews of sellers
E. Trust policy change → Batch (re)calculation of trust scores using new policy → Kafka notification about trust score update → Reindexing of Catalogues /Solr

5 Trust and Reputation Decentralized Approach

Apart from centralized trust and reputation models, NIMBLE is also exploring the decentralized trust approach by taking advantage of blockchain technology (see our research paper "*Federated Byzantine Agreement to Ensure Trustworthiness of Digital Manufacturing Platforms*" [INDB18]). One of the major challenges of federated digital manufacturing platforms is to ensure trust between platform instances (nodes) and their participants, in a way that enforces trustworthiness of collaboration platforms, the integrity of performed actions and measurements, and correctness of their recording, which further encourages new organizations to join and extend their businesses to new collaboration models and new communities. Since some of the key benefits of the Distributed Ledger Technology (DLT) relate to trust and transaction acceleration for the Internet of Things (IoT), we explore the use of distributed ledgers and consensus protocols to ensure trust and reputation between various participants collaborating via the NIMBLE collaborative manufacturing platform. Specifically, we explore the use of a consensus mechanism that employs the Federated Byzantine Agreement (FBA) algorithm, which is implemented in the Stellar consensus protocol (website: https://www.stellar.org/).

5.1 Related Work in Decentralized Trust Methods

Some background mechanisms for DLT include **community consensus mechanisms**, e.g. Proof-of-Work, Proof-of-Stake, Proof-of-Importance, Byzantine Agreement and Federated Byzantine Agreement, and **consensus protocols**, e.g. Ripple and Stellar.

5.1.1 Byzantine Fault Tolerance (BFT) in Distributed Systems

The BFT algorithm is created to address the Byzantine Generals Problem, which is a logical dilemma explained in [LASM82]. It suggests a scenario in which a group of Byzantine generals and their armies surround an enemy city that they plan to attack. The attack preparation involves sending a messenger from one army to the next, because in order to be successful, all armies must attack at the same time. However, the generals know that there are one or more traitors involved in the communication, who will try to confuse the others. The BFT algorithm ensures that the agreement for attack will not be compromised through untrustworthy messages. In other words, it needs to guarantee that (i) all loyal parties decide upon the same plan of actions, and (ii) a small number of traitors cannot cause the loyal parties to adopt a bad plan [LASM82]. The potential solutions could range from a solution with *oral messages* (every message that is sent is delivered correctly; the receiver of the message knows who is the sender; the absence of a message can be detected) to a solution with *signed messages* where anyone can verify the authenticity of the loyal party's signature.
Similarly, in distributed digital manufacturing environments with multiple actors, the Byzantine Generals Problem can be used to simulate the risk of producing incorrect or inconsistent outputs that can lead to a breakdown of the system. The failures in distributed systems can occur either as (i) *omission failure* i.e. not receiving a request, or failing to respond to a request, and (ii) *execution failure*, due to sending incorrect or inconsistent data, or responding to a request incorrectly. The authors in [LASM82] showed that Byzantine resilient (fault tolerant) systems that implement BFT solutions are expensive in traditional

networks: they require significant amounts of time and numbers of messages in order to guarantee the reliability of the system.

5.1.2 Distributed Ledger Technology

Distributed Ledger Technology (DLT), commonly called blockchain, is an emerging distributed data architecture for processing digital transactions over a business network. It tracks both tangible (i.e. car, house) and intangible (i.e. brand, copyright) assets involved in transactions, and facilitates the process of recording performed transactions. It can be also seen as a critical enabler of Digital Identity with the potential to minimize fraud and enable asset provenance and full transaction history [MCWA16]. Other key utilities of DLT and blockchains are *contract management* between two parties involved, *regulatory compliance*, *tokenization* for the authentication of physical items, when the items are paired with a corresponding digital token. The authors in [MCWA16] emphasize the following benefits of blockchains in financial scenarios:

- *Transaction immutability* – eliminates inclusion of an enforcer of trust in the ecosystem;
- *Transparency between all participants* – provides transparency for historical and real time transactions;
- *Transaction autonomy* – guarantees transaction execution under mutually agreed conditions and accelerates business outcomes.

5.1.3 Community Consensus Mechanisms

DLT and blockchain uses BFT and community consensus to legitimate transactions. In blockchain, new transactions are added into new blocks, to the end of the chain, broadcast to all the nodes, and can never be changed or removed once accepted by the network. If members of the community send inconsistent, inaccurate or malicious transactions information to others, the reliability of the blockchain breaks down. Hence, the consensus mechanisms are necessary in blockchain systems, to protect against the Byzantine Generals Problem. There are several approaches to consensus mechanisms, supporting both reputation and trusted identity claims [MAZI15]:

- **Proof-of-Work (PoW) algorithm** [DWLS88][DWNA92] is designed to protect against ill-behaviour of the participants who do not possess the majority of the system's computing power. PoW is the basis of Bitcoin, requiring from anyone who wants to add new information to the blockchain to perform a work-intensive task, e.g. must use information from the existing blockchain [TOZZ17]. PoW takes a fair amount of time to execute, which guarantees a practical protection against manipulation of the blockchain, enjoying a measure of protection against "51% attacks" [MAZI15][EYSI13]. An alternative solution for PoW relies on node votes and majority consensus in order to root out faults. The downside to this strategy is that it provides protection against Byzantine faults only as long as a relatively large majority of nodes on the blockchain continues to act legitimately [TOZZ17]. Although BFT has been studied in Distributed Systems for a long time, after the Practical BFT (PBFT) was introduced in 1999 [CALI99], there were no practical implementations of BFT until the emergence of the PoW algorithm.

- **Proof-of-Stake (PoS) algorithm** [KINA12] calculates consensus based on parties that have posted some collateral to prove their value. This opens the possibility of so-called "nothing at stake" attacks, in which parties that previously posted some collateral but later spent the money, can go back and rewrite history from a point where they still had stake. To mitigate such attacks, systems combine PoS and PoW, or delay refunding collateral long enough for some other consensus mechanism to establish a checkpoint. Some other approaches based on PoS are Leased Proof-of-Stake (LPoS) and Delegated Proof-of-Stake (DPoS) [KOST17]. LPoS allows holders to lease their balances to staking nodes, which increases the weight of the staking nodes and their chances of being allowed to add a block of transactions to the blockchain. DPoS enables holders to use their balances to elect a list of nodes with the opportunity to stake blocks of new transactions and add them to the blockchain.
- **Byzantine Agreement** [PESL80] ensures consensus in a fast and efficient way, enforcing trust and helping a small non-profit organization to keep more powerful organizations, such as banks or CAs, honest. Complicating matters, however, all parties must agree on the exact list of participants, and attackers must be prevented from joining multiple times and exceeding the system's failure tolerance.
- **Federated Byzantine Agreement (FBA)** overcomes situations in which malicious parties are joining many times in order to outnumber the well-behaved nodes, and create the Byzantine General Problem. FBA determines decentralized quorums by allowing each node to select quorum slices – individual trust decisions made by each node that together determine system-level quorums. FBA avoids complete lists of accepted participants that are necessary for ensuring consensus on a system level, and supports open membership that promotes organic network growth. It also has modest computing and financial requirements, in comparison to PoW and PoS.
- **Stellar Consensus Protocol** allows for solving problems through reaching consensus among network nodes [MAZI15]. It targets individual participants, rather than financial institutions. Stellar has a strong focus on technology, and uses an API based on the External Data Representation (XDR) standard [XDRS].

5.2 Federated Byzantine Agreement for Trust and Reputation in Business Platforms

Since in decentralized and distributed systems there are no central authorities to govern interactions and agreements between participants, trust and reputation of participant parties is still a major challenge. Similar problems occur in federated ecosystems, where low entry barriers should spur the organic growth of the systems. The NIMBLE platform is designed to support collaboration between companies interacting with each other on a daily basis, and here, trust and reputation are becoming major concerns.

To support trustworthiness between companies registered in one platform instance and collaborating with other companies either from the same instance or from another NIMBLE instance, it could be necessary to associate a federated identity to each company and provide appropriate mechanisms which allow multiple authorities to access and validate globally recognized entities. Depending on the business context, federated identities can vary and have a different representation across platform instances. Hence certain collaboration scenarios may require additional authentication and verification mechanisms.

Furthermore, it would be necessary to make a clear distinction between delegated identities and federated identities. In a system with delegated identities, the identity management is outsourced to another system, while in a system with federated identities, every participant can keep its own entity information in multiple nodes (e.g. NIMBLE instances). In the following, we explore the applicability of FBA to ensure trust and reputation between the platform instances and their participants in the NIMBLE ecosystem.

5.2.1 FBA background mechanisms

In FBA, a consensus protocol ensures that all participants agree on updating a replicated state that is called slot (e.g. transaction ledger), which helps participants to avoid contradictory states [MAZI15]. Each participant in the FBA system can safely apply update *x* in a specific slot when it has safely applied updates in all other slots upon which a specific slot depends, and when it believes that all other participants will agree on update *x* for that specific slot. In FBA language, this is described as "participant has *externalized* update *x* for a specific slot".

5.2.1.1 Quorum slices and quorum intersections

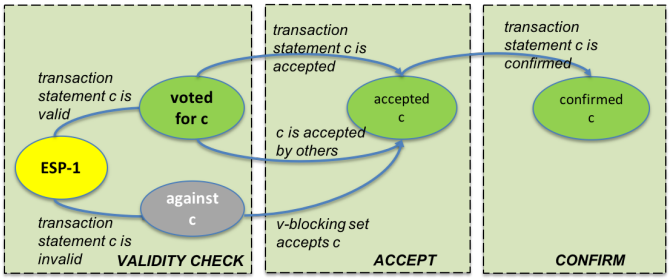

Figure 22 The consensus phases and agreement of an accepted statement c at a single node ESP-1

Agreement in FBA is accomplished by allowing every participant to decide on its own set of trusted neighbors, some of which may exhibit various types of non-rational behavior e.g. malicious behavior, unavailability, random errors, etc. A set of participants that is sufficient to reach agreement is called *quorum*, while a *quorum slice* is a subset of the quorum that can be selected based on arbitrary criteria, e.g. reputation or financial arrangements [MAZI15]. A participant agrees to a specific statement if there exists at least one quorum slice, which also agrees to the same statement. Another important property for ensuring the safety of an FBA-based system is *quorum intersection*. If a system lacks quorum intersection, quorums can independently agree on contradictory statements. In other words, quorum intersection exists iff any two of quorums share at least one node [MAZI15].

5.2.1.2 Tiered Quorum in Federated Platforms

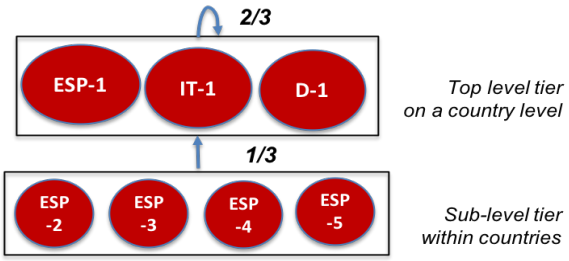

Figure 23 Tiered quorum structure for ensuring trust between federated instances in NIMBLE

In NIMBLE, we apply a tiered quorum structure, in which each platform instance is represented by a node (see Figure 2). The top-level tier is composed of instances (e.g. *ESP-1 (Spain), IT-1 (Italy)*, and *D-1 (Germany)*), which are governed by well-known and trusted state authorities and, therefore, enjoy a high level of trust. In the example in Figure 2, every top-level node agrees to a statement iff at least two other nodes at the same level agree on the same

statement. Sub-level tiers are constituted from nodes within a specific country (e.g. *ESP-2*, *ESP-3, ESP-4, ESP-5*), and must find trust from at least one node in the top-level tier.
The above presented tiered architecture increases trust, since at least two instances from the top-level are necessary to ensure system-wide agreements. In addition, not every single instance needs to be constantly available, which results in a more fault-tolerant system. FBA guarantees that all well-behaving nodes will externalize the same statement even in the presence of ill-behaved nodes.

5.2.2 FBA Consensus Phases

Agreement to a specific statement *c* requires the exchange of messages between participants (nodes) (Figure 3). The process of consensus at the level of a single node evolves in three phases, from (i) unknown, when nodes "vote for statement *c*", via (ii) accepted, when two nodes either succeed in agreement or show that statement *c* is contradictory, to (iii) confirmed, when both nodes send acceptance messages and confirm that statement *c* is true.

5.2.2.1 Federated statement acceptance

Federated agreement at a system-wide level allows open membership, but this set-up bears the risk that a majority of well-behaved nodes can be broken. The challenge here is for the well-behaved nodes to discover ill-behaved ones and to arrive at a quorum intersection of well-behaved nodes. In FBA, there is a term called *v-blocking* that identifies failed nodes (Figure 3) [MAZI15].

5.2.2.2 Federated statement confirmation

Statement *confirmation* means that a node *v* claims to accept statement *c* and confirms *c* iff an intact node *v* enjoys a quorum intersection. According to Theorem 11 in [MAZI15], once sufficient messages are delivered and checked, every intact node *v* will accept and confirm statement *c*.

5.2.3 FBA safety, liveness and fault tolerance

A distributed consensus protocol has to ensure system-wide safety, liveness and fault tolerance [MAZI15]. Safety is achieved if all correct instances agree or disagree on a certain statement that was initially proposed by one of the instances. The SCP solves this issue by attaching full sets of quorum slices to each propagated message.

Another important feature of the protocol is known as liveness of the system. In an FBA system, participants are not allowed to change their decisions for a statement after it was distributed to other participants. This may lead to a situation where an agreement on a statement gets stuck. Therefore, the consensus protocol has to ensure that the system agrees or disagrees with a statement after a finite amount of time. The SCP supports a federated voting mechanism (see Section 4.2), with voting of the nodes starting in a *bivalent* state (neither agrees nor disagrees with *a*). After enough votes were cast the state of the system changes to either *a-valent* (nodes vote for statement *c*) or a contradicting *ā-valent* state (nodes vote against statement *c*). The system can also end up in a stuck state, when it is not capable of finding a solution, due to the fact that nodes are not allowed to change their votes in a later phase. The SCP [MAZI15] avoids stuck states by applying neutralizable statements, which overcomes this problem.

The third feature of the protocol is known as fault tolerance. At any point in the execution of the protocol, the system should be able to recover from a failure of a node.

The authors in [CALLI99] present a fail-stop model that describes situations where a node crashes and stops sending messages to other nodes. In BFT, it can be assumed that nodes fail by behaving arbitrarily, e.g. the node is taken over by an attacker and sends compromising messages to the system.

5.3 Employing Stellar Consensus for Federated Business Platforms

Our objective is to enable a federated system like NIMBLE to agree on statements in a decentralized manner. The role of SCP is to define well-structured communication and message exchange between distributed platform instances and their participants. Figure 4 illustrates the message flow between application logic and Stellar consensus logic.

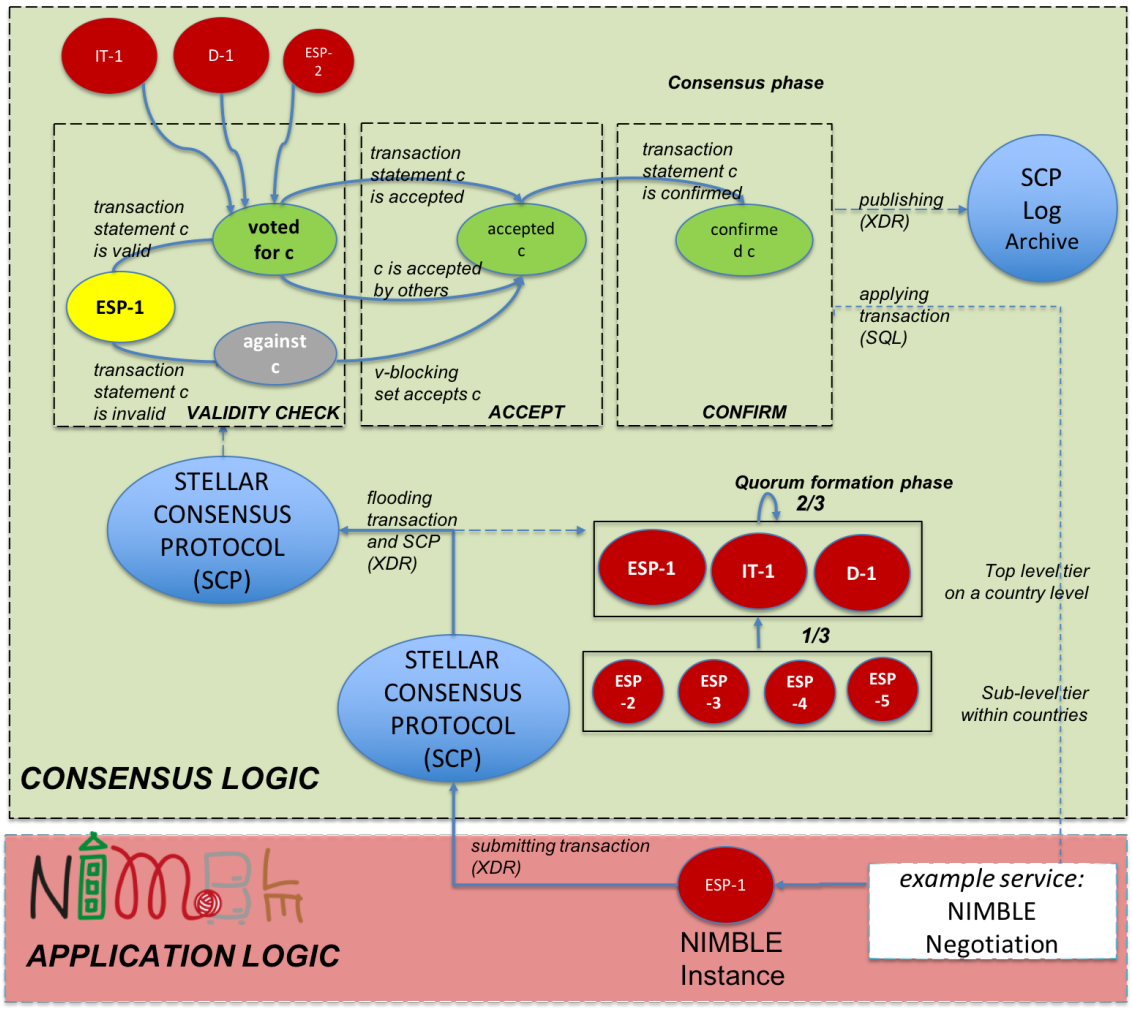

Figure 24 message flow between NIMBLE application logic and Stellar consensus logic

A specific NIMBLE platform instance in Spain, ESP-1 in Figure 4, is in the process of negotiating logistics details with a partner organization in Italy. To check trustworthiness of that partner, ESP-1 needs to submit a transaction to SCP. If ESP-1 is an external client to SCP submitting this new transaction, SCP contacts peers (through HTTP), submits an XDR transaction representation, and ESP-1 receives a status code of either "rejected" or "pending" [STELLAR]. If ESP-1 is not an external client to SCP but peer that already holds TCP connections to other peers, it has already defined quorum slices at the country level (top level tier). ESP-1 submits a transaction message in XDR format which is repeated to all peers (called "flooding" in [STELLAR]). SCP decides on the consensus state and the results are recorded in the SCP Log Archive (in XDR format) and sent back to the application (in our case, the NIMBLE negotiation service).

5.3.1 Embedded Architectural Components

Implementing SCP in NIMBLE requires several components to be added to the existing microservice architecture of the NIMBLE platform. Figure 25 shows the composition of the consensus component (green) and the platform services (red). Each consensus component will be realised in loosely coupled units, with inter-component communication executed via HTTP. The *Consensus Logic* component exchanges votes with other instances in the federated network and manages the formation of a quorum.

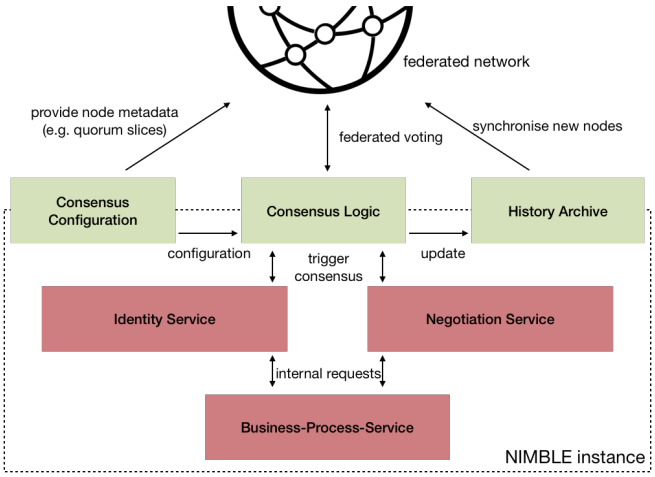

Figure 25 Consensus components embedded in the NIMBLE architecture.

Platform services, e.g. *Identity Service* and *Negotiation Service*, communicate with the *Consensus Logic* in order to find system-wide agreements for new statements. Configurable metadata of individual nodes is saved in the *Consensus Configuration* component, whose role is also to provide necessary information (i.e. quorum slices) for finding consensus. Metadata of nodes is at the same time shared internally with the consensus logic and is publicly available for other nodes. Each agreement is stored in the *History Archive*, which provides historical information for synchronising new nodes in the network.

6 Conclusion

This deliverable presents our approach to the design and implementation of trust and reputation management functionalities for the NIMBLE MVP platform release 5.0. We define trust as an evaluated measure that is available to all participants involved into a specific interaction via the platform, giving a trust-related insight about other interacting participants. Reputation is closely related to trust and used as the basis for a judgement as to whether an individual or organization can be trusted, before engaging into an interaction with them.

The trust and reputation management services in NIMBLE are built on a flexible trust model that combines different trust-related elements into the overall trust evaluation. The evaluation of trustworthiness of relationship and interaction between the two NIMBLE platform participants, e.g. buyer and seller, is a process of collecting trust-related elements, and based on the trust policy of the platform, calculating an aggregated trust index. The trust policy defines desired trust elements, their weights and trusted values, and can be customised by the platform managers. There is a need for trust policy governance in NIMBLE because of the federated and multi-sided nature of the platform. To overcome the cold start problem, when there are insufficient available ratings between users, the trust policy governance in NIMBLE defines different trust weight metrics for the platform start-up phase, for its growth phase and for the platform maturity phase.

The NIMBLE platform release 5.0 implements user ratings, reviews and trust ranking calculations for sellers and buyers registered at the NIMBLE platform. Trust related elements are directly exploited by the NIMBLE Search engine and therefore, presented at the platform only for sellers. While sellers and buyers can rank each other during the negotiation period, the trust related elements for buyers are – in the present implementation - not publicly visible to the sellers.

Finally, the rating of products and services of NIMBLE platform participants is currently not covered in release 5.0, but is planned as the platform usage grows over time. In addition, this deliverable discusses the potential of a decentralized trust approach based on blockchain technology and the Stellar consensus protocol, which can be seen as an advanced trust model that could be a possible extension of the platform in the future.

References

[Aberer03] Aberere, K. & Despotovic, Z. (2003). Managing Trust in a P2P Information System.

[AEGH10] S. Adali, R. Escriva, M. K. Goldberg, M. Hayvanovych, M. Magdon-Ismail, B. K. Szymanski, W. A. Wallace, and G. Williams. Measuring behavioral trust in social networks. In Intelligence and Security Informatics (ISI), 2010 IEEE International Conference on, pages 150–152, 2010

[ALIBABA.COM-IPO] Alibaba IPO: https://bit.ly/2SoDZv2

[ARGI07] D. Artz and Y. Gil. A Survey of Trust in Computer Science and the Semantic Web. Journal of Web Semantics, 5(2):58–71, June 2007. DOI: 10.1016/j.websem.2007.03.002. 20.

[BARB11] G. Barbian, "Trust Centrality in Online Social Networks," in Intelligence and Security Informatics Conference (EISIC), 2011 European, 2011, pp. 372-377. [14] W. Yan, [LEPE08] L. Lei, and L. Ee-Peng, "Price Trust Evaluation in E-service Oriented Applications," in E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, 10th IEEE Conference on, 2008, pp. 165-172.

[BLFL96] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized Trust Management. In Proceedings of the 1996 IEEE Symposium on Security and Privacy, SP '96, pages 164–, Washington, DC, USA, 1996. IEEE Computer Society. DOI: 10.1109/SECPRI.1996.502679.

[BODO05] P. A. Bonatti, C. Duma, D. Olmedilla, and N. Shahmehri. An Integration of Reputation-based and Policy-based Trust Management. In Proceedings of Semantic Web Policy Workshop, Galway, Ireland (7th November 2005), 2005.

[BUSK98] V. Buskens. The social structure of trust. Social Networks, 20(3):265–289, 1998

[CAFA98] C. Castelfranchi and R. Falcone. Principles of Trust for MAS: Cognitive Anatomy, Social Importance, and Quantification. In Proceedings of the 3rd International Conference on Multi Agent Systems, ICMAS '98, pages 72–, Washington, DC, USA, 1998. IEEE Computer Society.

[CAFV13] Barbara Carminati, Elena Ferrari, Marco Viviani (2013). Security and Trust in Online Social Networks. Synthesis Lectures on Information Security, Privacy, and Trust, December 2013, Vol. 4, No. 3, pp. 1-120. https://doi.org/10.2200/S00549ED1V01Y201311SPT008

[CALI99] Castro, M. and Liskov, B., 1999. Practical Byzantine Fault Tolerance. In *Proceedings of the 3rd Symposium on Operating Sys. Design and Implem.*, pp. 173–186.

[CALW08] J. Caverlee, L. Liu, and S. Webb. Socialtrust: tamper-resilient trust establishment in online communities. In Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries, JCDL '08, pages 104–114, New York, NY, USA, 2008. ACM

[Chang06] Chang, E., Dillon, T.S. & Hussain, F.K. (2006). "Trust and Reputation for Service-Oriented Environments". John Wiley & Sons, Ltd., ISBN-13: 978-0-470-01547-6.

[COJOIS02] B. E. Commerce, A. Jøsang, and R. Ismail. The Beta Reputation System. In In Proceedings of the 15th Bled Electronic Commerce Conference, 2002.

[Dillon04] Dillon, T.S., Chang, E. & Hussain, F.K. (2004). "Managing the Dynamic Nature of Trust", in IEEE Transaction of Intelligent Systems, vol. 19, no. 5, pp. 77—88.

[DWLS88] Dwork, C., Lynch, N. and Stockmeyer, L, 1988. Consensus in the Presence of Partial Synchrony. Journal of the ACM 35, pp. 288–323.

[DWNA92] Dwork, C. and Naor, M., 1992. Pricing via Processing or Combatting Junk Mail. In *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, pp. 139–147.

[ENISA18] ENISA, 2018. Annual Report Trust Services Security Incidents 2017. Online available: https://www.enisa.europa.eu/publications/annual-report-trust-services-security-incidents-2017 (last accessed: October 2018).

[EUR-LEX14] Regulation (EU) No 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market (23 July 2014). Online available: http://eur-lex.europa.eu/eli/reg/2014/910/oj

[EYSI13] Eyal, I. and Sirer, E.G., 2013. Majority is not Enough: Bitcoin Mining is Vulnerable. Online available from: http://arxiv.org/abs/1311.0243 Last access Feb. 2018.

[GAGD07] S. Galizia, A. Gugliotta, and J. Domingue (2007). A Trust Based Methodology for Web Service Selection. International Conference on Semantic Computing, IEEE, 2007.

[GAMB90] D. Gambetta. Trust: Making and Breaking Cooperative Relations. Blackwell Publishers, 1990.

[GDPR95] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Online available: https://www.cdt.org/files/privacy/eudirective/EU_Directive_.html Last access: October 2018

[GKRT04] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In Proceedings of the 13th international conference on World Wide Web, WWW '04, pages 403–412, New York, NY, USA, 2004. ACM.

[GKRT04] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In Proceedings of the 13th international conference on World Wide Web, WWW '04, pages 403–412, New York, NY, USA, 2004. ACM

[GOLD05] J. A. Golbeck. Computing and applying trust in web-based social networks. PhD thesis, University of Maryland at College Park, College Park, MD, USA, 2005.

[Hartman03] Hartman, F. (2003) "The role of trust in successful system development and deployment", Proceedings of IEEE Conference on Industrial Informatics 2003, Banff, Canada.

[HASI10] C.-W. Hang and M. P. Singh. Trust-based recommendation based on graph similarity. In in AAMAS Workshop on Trust in Agent Societies (Trust), 2010.

[Huss04] Hussain, F.K., Chang, E. & Dillon, T.S. (2004), "Taxonomy of Trust Relationships in P2P Communication", Proceedings of the 2nd International Workshop on Security in Information Systems, Porto, Portugal, pp. 99—103.

[INDB18] J. Innerbichler, V. Damjanovic-Behrendt, "Federated Byzantine Agreement to Ensure Trustworthiness of Digital Manufacturing Platforms", In Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems (CryBlock 2018) (co-located with the 16th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys 2018)) June 15th, 2018, Munich, Germany. Online: https://cryblock18.hotcrp.com/paper/27?cap=027aCQyRL51MXJs

[JAES10] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In Proceedings of the fourth ACM conference on Recommender systems, RecSys '10, pages 135–142, New York, NY, USA, 2010. ACM.

[JOHP06] A. Jøsang, R. Hayward, and S. Pope. Trust network analysis with subjective logic. In Proceedings of the 29th Australasian Computer Science Conference - Volume 48, ACSC '06, pages 85–94, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc

[JOIB07] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. Decision Support Systems, 43(2):618 – 644, 2007. DOI: 10.1016/j.dss.2005.05.019.

[KAZA05] P. Kannadiga, M. Zulkernine, and S. I. Ahamed. Towards an Intrusion Detection System for Pervasive Computing Environments. In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II - Volume 02, ITCC '05, pages 277–282, Washington, DC, USA, 2005. IEEE Computer Society.

[KINA12] King, S. and Nadal, S, 2012. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. Online: http://peercoin.net/assets/paper/peercoin-paper.pdf.

[KOST17] Kostarev, G., 2017. Review of Blockchain Consensus Mechanisms, Waves Platform. Online available: https://blog.wavesplatform.com/review-of-blockchain-consensus-mechanisms-f575afae38f2 Accessed 2018.

[KUGO07] U. Kuter and J. Golbeck. SUNNY: A New Algorithm for Trust Inference in Social Networks Using Probabilistic Confidence Models. In Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada, pages 1377–1382. AAAI Press, 2007.

[LASM82] L. Lamport, R. Shostak, and M. Pease, 1982. The Byzantine Generals Problem. ACM Transactions on Programming Lang. and Sys., Vol. 4, No. 3,382—401.

[LEHK10] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting Positive and Negative Links in Online Social Networks. In Proceedings of the 19th international conference on World wide web, WWW '10, pages 641–650, New York, NY, USA, 2010. ACM

[LEVI09] R. Levien. Attack-Resistant Trust Metrics. In J. Golbeck, editor, Computing with Social Trust, Human-Computer Interaction Series, pages 121–132. Springer London, 2009.

[LLLL08] H. Liu, E.-P. Lim, H. W. Lauw, M.-T. Le, A. Sun, J. Srivastava, and Y. A. Kim. Predicting trusts among users of online communities: An Epinions case study. In Proceedings of the 9th ACM conference on Electronic commerce, EC '08, pages 310–319, New York, NY, USA, 2008. ACM.

[LVLD08] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong. Addressing cold-start problem in recommendation systems. In Proceedings of the 2nd international conference on Ubiquitous information management and communication, ICUIMC '08, pages 208–211, New York, NY, USA, 2008. ACM.

[MARSH94] S. P. Marsh. Formalising Trust as a Computational Concept. PhD thesis, University of Stirling, April 1994.

[MARSH94] S. P. Marsh. Formalising Trust as a Computational Concept. PhD thesis, University of Stirling, April 1994.

[MATG07] M. Maheswaran, H. C. Tang, and A. Ghunaim. Towards a gravity-based trust model for social networking systems. In Proceedings of the 27th International Conference on Distributed Computing Systems Workshops, ICDCSW '07, pages 24–, Washington, DC, USA, 2007. IEEE Computer Society

[MAZI15] Mazières, D., 2015. The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus. Online: https://www.stellar.org/papers/stellar-consensus-protocol.pdf Last accessed in March 2018.

[MCWA16] McWaters, R.J., 2016. The future of Financial Infrastructure. World Economic Forum 2016. Online available from: http://www3.weforum.org/docs/WEF_The_future_of_financial_infrastructure.pdf Last accessed March 2018.

[MUIM02] L. Mui, M. Mohtashemi, and A. Halberstadt. A Computational Model of Trust and Reputation for E-businesses. In Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02) - Volume 7, HICSS '02, pages 188–, Washington, DC, USA, 2002. IEEE Computer Society.

[NESP11] S. Nepal, W. Sherchan, and C. Paris. STrust: A Trust Model for Social Networks. In Proceedings of the 2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, TRUSTCOM '11, pages 841–846, Washington, DC, USA, 2011. IEEE Computer Society

[NIEL99] J. Nielsen. Trust or Bust: Communicating Trustworthiness in Web Design. Technical report, March 1999. http://www.nngroup.com/articles/trust-or-bustcommunicating-trustworthiness-in-web-design/

[NISS99] H. Nissenbaum. Can Trust be Secured Online? A theoretical Perspective. Etica e Politica, 1(2), December, 1999.

[OSTE01] D. Osterwalder. Trust through evaluation and certification? Social Science Computer Review, 19(1):32–46, 2001

[PESL80] Pease, M., Shostak, R., and Lamport, L., 1980. Reaching Agreement in the Presence of Faults. Journal of the ACM 27, 228–234.

[REZE02] P. Resnick and R. Zeckhauser. Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. In M. R. Baye, editor, The Economics of the Internet and E-Commerce, pages 127–157. Elsevier Science, November, 2002.

[SHNP13] W. Sherchan, S. Nepal, and C. Paris. A Survey of Trust in Social Networks. ACM Computing Surveys, 45(4):47:1–47:33, August, 2013. DOI: 10.1145/2501654.2501661.

[SHNP13] W. Sherchan, S. Nepal, and C. Paris. A Survey of Trust in Social Networks. ACM Computing Surveys, 45(4):47:1–47:33, August, 2013.

[STELLAR] Stellar Core Data Flow. Online available from: https://www.stellar.org/developers/stellar-core/software/core-data-flow.pdf

[ŠVSA13] T. Švec and J. Samek. Trust evaluation on Facebook using multiple contexts. In 3rd Workshop on Trust, Reputation and User Modeling (TRUM'13), Proceedings, 2013

[TJUL16] Truong, N., Jayasinghe, U., Um, Tai-Won, Lee, G.M., "A Survey on Trust Computation in the Internet of Things." The Journal of Korean Institute of Communications and Information Sciences, 33, 1ß-27, 2016.

[TOZZ17] C. Tozzi, 2017. Byzantine Fault Tolerance: The Key for Blockchain. Online available from: https://www.nasdaq.com/article/byzantine-fault-tolerance-the-key-for-blockchains-cm810058

[TRLA10] S. Trifunovic, F. Legendre, and C. Anastasiades. Social Trust in Opportunistic Networks. In INFOCOM IEEE Conference on Computer Communications Workshops , 2010, pages 1–6, 2010.

[VUGA14] M. Vujasinovic, A. Gugliotta. Trust-based Discovery for Web of Things Markets, W3C Workshop on the Web of Things – Enablers and services for an open Web of Devices. 25–26 June 2014, Berlin, Germany

[Wang03] Wang, Y. & Vassileva, J. (2003). "Bayesian Network Trust Model in P2P Networks," In Proceedings of the 2nd Int. Workshop on Agent and P2P Computing, Melbourne, pp. 24—24.

[WEIS05] S. A. Weis. Security Parallels between People and Pervasive Devices. In Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOMW '05, pages 105–109, Washington, DC, USA, 2005. IEEE Computer Society.

[WIWI10] G. Wierzowiecki and A. Wierzbicki. Efficient and Correct Trust Propagation Using CloseLook. In Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01, WI-IAT '10, pages 676–681, Washington, DC, USA, 2010. IEEE Computer Society

[XDRS] External Data Representation (XDR) Standard. Online: https://tools.ietf.org/html/rfc4506.html

[YUMH06] Y. Yuan, Z. Miao, and S. Hu. A pervasive computing security system based on human activities analysis. In TENCON 2006. 2006 IEEE Region 10 Conference, pages 1–4, 2006.

[YUSI02] B. Yu and M. P. Singh. An evidential model of distributed reputation management. In Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1, AAMAS '02, pages 294–301, New York, NY, USA, 2002. ACM.

[ZHCW06] Y. Zhang, H. Chen, and Z. Wu. A social network-based trust model for the semantic web. In L. Yang, H. Jin, J. Ma, and T. Ungerer, editors, Autonomic and Trusted Computing, volume 4158 of Lecture Notes in Computer Science, pages 183–192. Springer Berlin Heidelberg, 2006.

Appendix 1. Trust and Reputation Questionnaire

Trust and reputation questions	**Micuna/AIDIMME**	**PIA**	**LIND**	**WHR**
What is the purpose of trust in your scenarios? ● Access trust (supporting accessing resources owned by or under the responsibility of the relaying party) ● Identity trust (the belief that an agent identity is as claimed) ● Delegation trust (trust in an agent (the delegate) that acts and makes decision on behalf of the relying party) ● Context trust (the extent to which the relying party believes that the necessary systems and institutions are in place, to support the transaction and provide a safety in case something should go wrong)	Access trust Identity trust	X	X	Identity Trust
In the context of your use cases, what makes the business actors to trust each other?	Comments/opinions submitted by business actors Num. of successful negotiations	X	X	Strong identification
Is trust assessment based on technical, security related issues (e.g. data exchange security/ level of encryption)?	Yes	X	X	Yes
Is reputation based on user ratings/ review?	Only in review (comments, opinions)	X	X	No
Is reputation based on an additional information, e.g. a history of the business (year of foundation, number of successful contracts, stability of profit over certain period of time, etc.)?	Yes. Num. of successful negotiations should be considered	X	X	No
Are you aware of any online supplier/ manufacturer service rating platform?	No	X	X	No

Appendix 2. Analysis of the NIMBLE Platform UIs to Support User Ratings and Review Management

This section provides an analysis of the NIMBLE UIs and services, with a view on enabling trust and reputation measures on the platform. We analyse the interaction steps from the current demo based on the platform release 3: https://www.youtube.com/watch?v=T-eVQDlhijM, and describe the details related to improvement of the existing UIs and services in order to allow for building trust and reputation on the platform.
In the report D4.4 "*Platform User Experience - Platform Manager's Point of View*", platform manager's needs to monitor and administer the platform are summarized, including advices related to building trust and reputation through various phases of platform development. For example, D4.4 suggests to differentiate between the following three maturity levels of trust metrics (see Section 2.2 of this Appendix):

- metrics for the start-up phase,
- metrics for the platform growth phase, and
- metrics for the platform mature phase.

Hence, in this section we also define trust elements to be collected through different maturity phases of the platform development.

Designing Trust and Reputation from the view of NIMBLE UIs

Note the analysis of NIMBLE UIs in order to cover changes required to support trust and reputation mechanisms of the platform, is based on the platform release 3. The proposed UIs are given at the end of this subsection.

PART 1: Company Registration

1. **COMPANY REGISTRATION.** This form should contain (sub-)form(s) for the description of company trade details, relevant company description, certification and trademarks. A progress bar illustrating the completeness of the registration process with respect to added company details, needs to be presented to users. The same progress bar will be used as an evaluation method for the overall trustability of the user from the perspective of the platform. For example, if progress bar shows between 80 - 100% of completeness, the registered company earns 5 out of 5 in terms of its trustability on the platform; for 60 - 79% of completeness, the company earns 4 out of 5 for its trustability, etc.
    a. **COMPANY REGISTRATION UI** needs some minor updates:
        i. **Street, Building number, ZIP/Postal code, Country** - must be mandatory for business entities;
        ii. **Business Type** should be added during the company registration process, preventing companies to change their domains and responsibilities during the transaction execution period, e.g. a user would like to order products directly from the manufacturer, not from its agent. Some examples of Business Types are Manufacturer, Logistic Provider, Government Body, Agent, Associations.

iii. **Business Keywords** could help with the description of businesses of the registered companies. We suggest to add keywords during the registration process, allowing for better search performances based on entered keywords for all registered entities, and regardless of the completeness of their company profiles. Some examples of business keywords are: transport, policies, automotive industry, etc. We should allow for a minimum range of 5 mandatory keywords, up to maximum of 20.

iv. **Year of Company Registration** - a year when company has been (offline) registered.

b. **COMPANY REGISTRATION services**. Some new background services could be contributing here:

i. VAT automated checking service should be investigated (see: http://ec.europa.eu/taxation_customs/vies/)

ii. A year (date) when company was registered on the platform (note that this is not the same as a Year of Company Registration (a(iv))). It could be used to calculate a number of years of company being continuously present at the platform. Such detail can be an indicator of trustability between the registered company and the platform.

iii. For Business Keywords, we should consider integrating a standard-based taxonomy, e.g. eurostat RAMON (Reference And Management Of Nomenclatures): https://goo.gl/6DfcNh

c. **COMPANY TRADE DETAILS UI (new UI):**

i. **Markets**, e.g. Western Europe, South Europe, Central Europe, North Europe, Eastern Europe, Across Europe.

ii. **Accepted Delivery Terms**, e.g. FOB, CFR, CIF, etc.

iii. **Accepted Payment Type**, e.g. PayPal, Credit Card, Western Union, etc.

The above details can contribute to more precise search e.g. by searching for a company that accepts PayPal and delivers products across Europe.

d. **COMPANY DESCRIPTION UI (new UI):**

i. Company logo
ii. Company statement (description in free form)
iii. Photos
iv. 3D Virtual Tour
v. Links to various social media, e.g. Twitter, LinkedIn, Pinterest
vi. Upcoming (offline) events (list of events with name, date, location, scope of the event, e.g. demonstration of new products, educational workshop, etc.)
vii. Past events (list of past events with the same structure as the description of the above upcoming events)

e. **COMPANY CERTIFICATION AND TRADEMARKS UI (new UI):**

i. Type of Certification
ii. Reference number
iii. Certificate Name
iv. Issued by
v. Validity period (start date + end date)
vi. Image

vii. **Description**

PART 2: Product Publishing

1. **PUBLISH PRODUCT UI** should incorporate a progress bar showing the completeness of the product description during the product publishing steps.
2. **PUBLISH PRODUCT service** translates the product description completeness into trust value, which can be used for platform's trustworthiness (towards the product owner); it means that such trust value won't be publicly presented, and the sellers need to be informed that adding more details and descriptions about their companies, products and services will add to their internal ratings (within the platform, e.g. such products and services would be more likely recommended by the platform in comparison to those with weak or missing descriptions).

PART 3: Search

1. **SEARCH UI** (see Figure A2-1) should present (i) Company (average) Rating Score and Reviews (link to a page with all individual reviews), and (ii) Product/Service (average) Rating Score and Reviews (link to a page with all individual reviews related to the product).

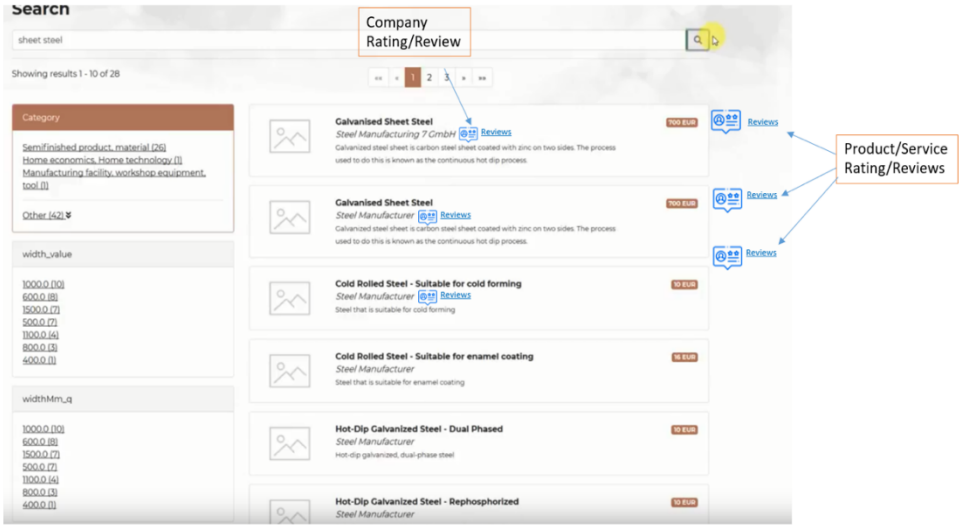

Figure A2-1. SEARCH UI with average rating scores and reviews for company and for its products/services

2. **SEARCH UI** (see Figure A2-2) should include a button to enable the user (buyer) to select relevant trust metrics elements. By using such button, the buyers can adjust the search results according to those trust metrics elements that they value the most in their business, e.g. trust elements such as response time, user ratings of the product/service, company profile completeness, trading volume, number of transactions performed over the platform. By default, trust metrics elements should have mid values (if the buyer is not using this button).

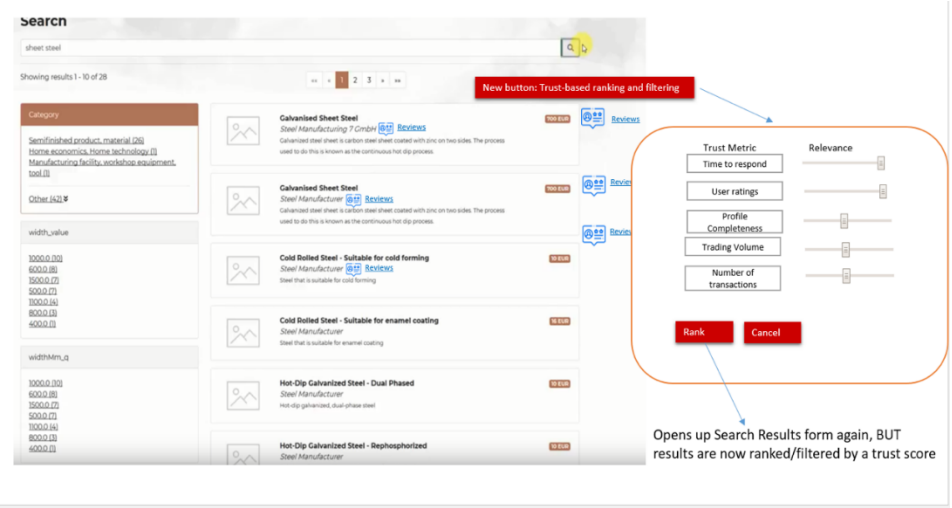

Figure A2-2. SEARCH UI with trust metrics control button

3. **PRODUCT PAGE UI/ PRODUCT DETAILS** (8:44 in the demo https://www.youtube.com/watch?v=T-eVQDlhijM) (see Figure A2-3) should present
   a. Product/Service Rating and Reviews (link to a page with all individual reviews), and
   b. Company Rating and Reviews.

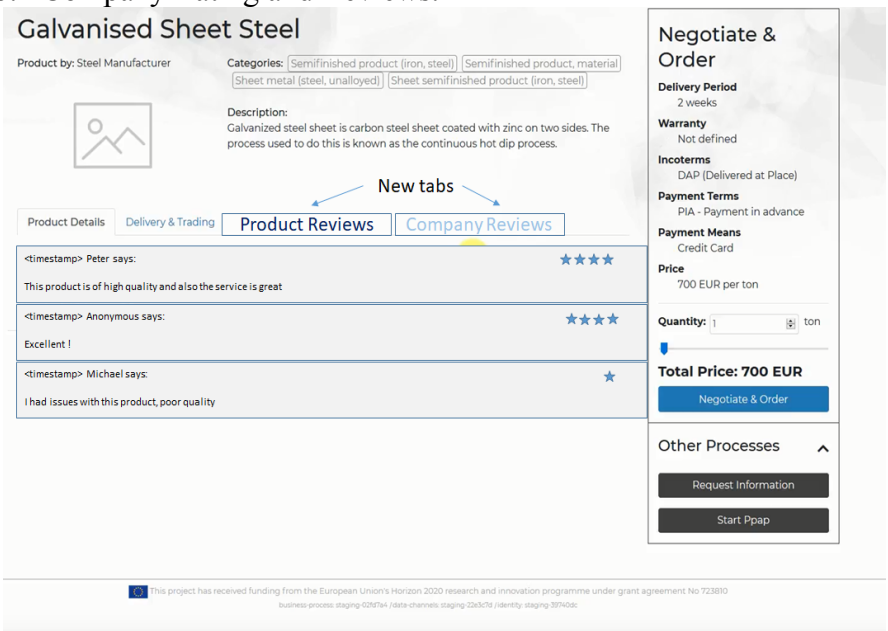

Figure A2-3. PRODUCT DETAILS UI with Product and Company Reviews tab

4. **EXPLORATIVE SEARCH UI/ PROPERTIES OF THE EXPLORATIVE SEARCH RESULTS UI** (9:36 in the demo https://www.youtube.com/watch?v=T-eVQDlhijM) - list of properties and their specification should display the average rating score for the product/service.

PART 4: Negotiation

1. **SELLER SIDE/ Dashboard** - The seller is able to preview the coming request in his dashboard. He should be able to identify the buyer, who created and sent the request (we should display: buyer's name, address, country, and average rating score).
   Note: Buyer and seller need to be aware of each other from step 1. For example, seller might want to investigate the buyer's company profile before sending out more details about their products and production processes. Hence, it would be important to display the buyer's rating score too.

2. **RATING OF NEGOTIATION PROCESSES THAT DIDN'T SUCCEED** - Regarding the rating of negotiation that didn't succeed, we would need a button "Comment on this process" on both sides Seller's and Buyer's side, that would invite them to provide their comments related to the interaction. This button would open a new UI enabling the following steps:
   a. **on the BUYER side:**
      i. Buyer can rate (i) accuracy/ reliability of information provided and (ii) satisfaction with the response time. In parallel, system can calculate response-time rate and verify if it is within the scope of the average/ideal response-time on the platform, better or worse than average/ideal time.
      ii. If the buyer indicates "Accuracy/reliability of information provided" as being poor, system offers to the buyer to select one or more options from the following checklist:
         1. not satisfied with the price
         2. not satisfied with the delivery
         3. not satisfied with the payment options
         4. inaccurate product information
         5. suspicious company information
      iii. System analyses buyer's feedback and for ticked 4) and/or 5) as described above, the system adds negative points to the rated company and the specific product.
   b. **on the SELLER side:**
      The seller also has an opportunity to provide his view on the negotiation by rating the following:
         1. slow response time
         2. suspicious company information (report here <link>)
         3. undervalued offer
         4. rejected delivery terms.
   c. **Incentives for Sellers and Buyers to provide their comments on failed negotiation.** Both Sellers and Buyers earn additional points for their trust/reputation if they provide reviews.

PART 5: Rating of the Contract Fulfillment

1. **RATINGS AND REVIEWS (new services):** After closing an order, based on the estimated delivery time for the order, the buyer should be invited by the platform to rate (i) previously purchased products/services and (ii) the seller. The rating process is based on the overall trading experience gained over the platform. Products/services and company-related rating scores should have quantitative values, and could be used for featuring products and companies, and their possible promotions via the platform (future platform services).
    a. The platform sends a message to the buyer (after the estimated delivery time) with a link to the Ratings & Reviews UI.
    b. All buyer's ratings and reviews of sellers should be public;
    c. Buyer's ratings and reviews of sellers should include the following elements:
        - Seller's communication (to be evaluated on a scale from 1 - 5) for each of the following sub-elements:
            - Quality of the negotiation process
            - Quality of the ordering process
            - Response time
        - Fulfillment of contractual terms
            - Product listing accuracy
            - Conformance to other agreed contractual terms
        - Delivery and packaging

Note that after closing an order, the platform invites sellers too, to rate their interaction and communication with the buyers. All ratings and reviews related to buyers should be available only to sellers and via platform management UIs (for the purpose of platform management). If the buyer appears on the platform as a seller too, the presentation of ratings and reviews about himself as a buyer should be prevented. It should be a matter of the platform to keep rating scores and reviews about buyers only for the platform management services.

2. **RATINGS AND REVIEWS (new UI):** This UI should contain the following elements:
    i. **Aggregated values based on all reviews, showing the following** (see Figure A2-4 just as an example):
        1. overall rating as numerical value (e.g. 4.9),
        2. five stars-based value for overall rating (on the left)
        3. 5 stars: with a progress bar and a number of 5 stars reviews in total,
        4. 4 stars: with a progress bar and a total number of 4 stars reviews,
        5. 3 stars: with a progress bar and a total number of 3 stars reviews,
        6. 2 stars: with a progress bar and a total number of 2 stars reviews,
        7. 1 star: with a progress bar and a total number of 1 star reviews.

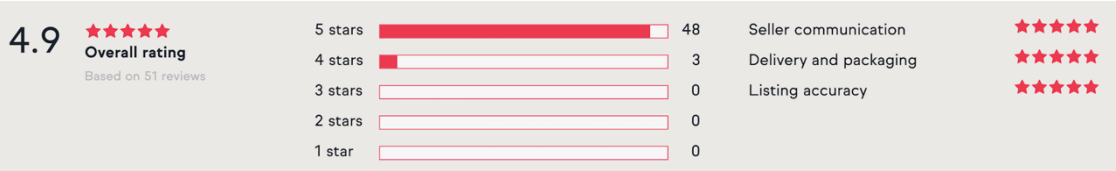

Figure A2-4. Trust metrics visualization (an example)

ii. List of all public reviews, showing the following details:

1. Name of the company/person who provided the review;
2. Comment from buyers (text);
3. Response on review from sellers (text);
4. **Overall rating related to the specific review, and based on trust metrics (as described in Part 5, c:** Seller's communication, Fulfillment of contractual terms, and Delivery and packaging).

Discussing Trust Metrics vs. Corresponding Platform Maturity

SELLER	Description of trust metrics	Platform maturity level
	Registration profile completeness	start-up phase
	Location relevance	start-up phase
	Certifications relevance	growth phase
	Collaboration aspects **Response time** e.g. Answering in less than 12 hours vs. 36 hours, vs. 48 hours…; Closing the offer in less than 12 hours vs. 36 hours, vs. 48 hours…; Signing the fulfilled contract in less than 12 hours vs. 36 hours, vs. 48 hours…	start-up phase
	Collaboration aspects **Trading volume** e.g. we not show exact numbers due privacy restrictions but categorize the trading volume into 'low', 'medium', 'high'	growth phase
	Collaboration aspects **Number of contracts successfully closed via the platform**	mature phase
	Collaboration aspects **User rating scores and reviews (rating of the quality of negotiation; quality of ordering (packing and dispatching) , and quality of the contract fulfilment (as described in Part 5 of Appendix 2)**	mature phase
	Collaboration aspects **Average rating scores for product/services/companies**	mature phase
	Company presence and activities on the platform	growth phase
BUYER		
	same as for the seller	
PRODUCTS/SERVICES		
	User ratings of	mature phase
	User reviews	mature phase