Towards A Personalized Trust Architecture

Hisham Salah, Mohamed Eltoweissy
The Bradley Department of Electrical and Computer Engineering
Virginia Tech
USA
hisham79@vt.edu, toweissy@vt.edu

***Abstract*—Advances in next generations distributed cyberspace technologies are expected to evolve a global cyberspace marketplace for resource and services. In marketplaces, particularly with the autonomy of users, trading parties differ in many respects, for example their declared agendas, honesty, owned or outsourced resources, and their products. Given these differences and the scale of a global marketplace, trusting the outcome of services or service compositions becomes a challenging endeavor. Personalized autonomic trust management constitutes a powerful solution to such issues. However, current trust systems do not generalize well beyond those problems they were designed for. In addition they are oblivious to individual users' trust requirements. In this paper we propose a generic trust management framework and architecture for trust personalization and autonomic trust management. Our main contributions include: defining and quantifying trust in terms of four parameters: intent capability, integrity and results, and a unified framework for autonomic and personalized trust management. In our simulations, we investigated using trust parameters in BitTorrent protocol. Results showed that, using trust parameters reduce download time and an increase number of finished peers.**

Keywords-trust management; trust personalization; autonomic systems; P2P networks; BitTorrent.

I. INTRODUCTION

Next generations distributed cyberspace technologies, to include cloud computing, social networking, and mobile applications, are driving the explosion in storage and processing power, with enhanced service offering, availability and mobility, while drastically reducing the cost of computing and communications. Such advances are expected to evolve a global cyberspace marketplace for resources and services.

However with this potential, new challenges arise. In marketplaces, particularly with the autonomy of users, trading parties differ in many respects, for example their declared agendas (what parties promise to provide through their services), honesty (if parties deliver what they promised), owned or outsourced resources (what assets parties have or can secure), and their products (what products they specialized to make), to name a few. Given these differences and the scale of a global marketplace, trusting the outcome of services or service compositions becomes a challenging endeavor. Additionally it creates diversities in requirements of different parties in the marketplace, hence the vital need for personalized trust management [1].

Definition of trust: we define trust as a party's belief or disbelief that another party, for a said subject of trust (SoT) in a given context, has the intent, integrity, results and capability to exhibit a set of acceptable actions in the future, for the welfare of the trusting party. Intent is a cognitive state representing willingness to execute a given action. Integrity is consistency of actions. Capability is whatever is needed to deliver services. Finally, Results is track record of production and delivery.

Personalized autonomic trust management constitutes a powerful solution to dilute some of the aforementioned issues. Personalized trust management endeavors the selection of best services that satisfy users' trust requirements among different alternatives. Autonomic management can dilute the inevitable increase in number of users with their varying requirements, and the amount of data gathered and analyzed. Personalization is necessary to accommodate the variety in requirements and services.

Current trust management solutions do not generalize well beyond those applications and domains they were designed for [2]. In addition, they are oblivious to individual users' trust requirements; which is the essence of trust [1]. Drawbacks of current solutions [2-8] may be summarized as follows:

- Hard-coded methodologies and computations: trust as well as trust-based decision support computations and methodologies are not personalized and are generally hard coded and non reconfigurable;
- Limited credentials: Contemporary trust/reputation systems primarily utilize direct or indirect experience, while other forms of credentials are hardly ever utilized;
- Oblivious to mutual trust requirements: most existing works focus on trust issues with service provides while neglecting the need to also trust the consumer.
- Lack of evaluation mechanisms for trust computations. Up to our knowledge no work exists in the literature towards a mechanism for quantitative assessment of trustworthiness computations.

II. PROPOSED PERSONALIZED TRUST MANAGEMENT FRAMEWORK

In this work we propose a generic trust management framework and architecture for trust personalization and autonomic trust management. The proposed solution comprises a number of reconfigurable components that can be used to implement a distributed or centralized trust management system according to problem requirements. Users' trust requirements are expressed using trust definition language (TDL). We assume service-to-service based environment where consumers, brokers and service providers (system users) are trading services to satisfy their own requirements. Trust requirements are satisfied either by:

- Selection of trust computation methodology based upon user trust preferences, which involve machine learning techniques for different trust computation methodologies used in the system; or
- Enable users to define their computation methodology.

The proposed framework consists of five main components as follows (Figure 1):

COLLABORATECOM 2011, October 15-18, Orlando, United States
Copyright © 2012 ICST
DOI 10.4108/icst.collaboratecom.2011.247164

Decision management (DS): implements trust-based decision policy of a trustor (party who establish trust in others), which includes acceptance/refusal of interaction parties, type of punishment and rehabilitation period for misbehaving parties, and type of rewards for well behaving parties;
Expectation Management (EM): accepts trustor's preferences towards a specific SoT and accordingly develops expectations of trustor in the required transaction(s) and evaluates their satisfaction after transaction;
Analytics Management (AN): analyzes different trustees (parties whose trustworthiness being evaluated) according with input from trustor's preferences. Analytics include select, define or redefine trust computation methodologies and trust data and metrics which suite trust preferences of users;
Monitoring Management (MM): creates monitoring threads to capture and collect trust data for trust computation;
Data Management (DM): implements storage, retrieval and replication policies necessary to manage system and trust data. Interaction among these components for querying trustworthiness about a said SoT is as follows: users send trust preferences and decision policies towards the required SoT to DS. In turn DS passes trust preferences to EM to indentify user trust expectations from transacting with parties and report satisfaction when interacting with them. EM extracts requirements needed for analytics from trust preferences document and develops satisfaction parameters thresholds. AN uses its stored knowledge about existing trust computation methodologies to recommend appropriate method(s) which suite(s) user trust preferences. Recommended method(s) are then forwarded to EM which solicits user acceptance for utilizing the recommended methodology(s). In case of no suitable method exists, EM is informed to solicit new computation methodology. AN is informed of user choice(s) and different parties are evaluated accordingly. Evaluation report is forwarded to DS for decisions. DS selects among parties for transaction and informs analytics for building trust, which in turn contacts MM to collect required data from transaction. Transaction report is delivered to EM which calculates satisfaction and forwards it DS to take appropriate actions and AN to update its knowledge about the computation method used.

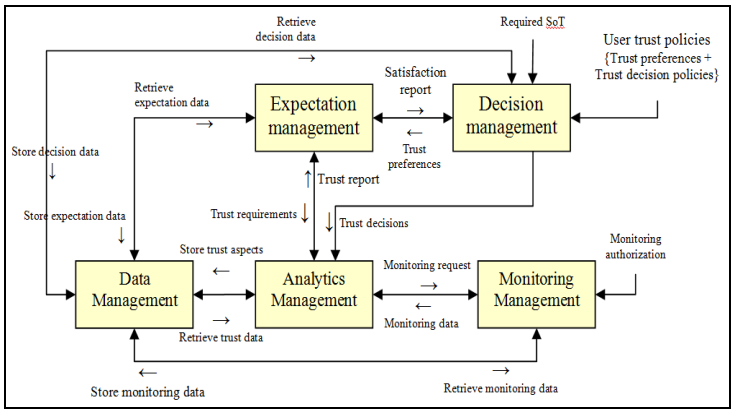

Figure 1 Generic Framework Components

III. TRUST ESTABLISHMENT PROTOCOL

Trust establishment protocol between two authenticated parties is as follows: trustor and trustee exchange expectations and declaration on SoT respectively. The trustor extracts available trust parameters concerning; intent, integrity, capability and results from trustee's declaration and collect missing and needed parameters from its experience with trustee and from others experience. Accordingly trustor develops the trust report necessary to evaluate trustee. In case the decision is to trust, the trustee is informed and a contract abiding trustee's provisioning is developed. During provisioning, the trustor monitors trustee's performance and accordingly evaluates him and updates trust parameters locally and decides whether to still trust or un trust.
Autonomic aspects of our proposed framework are as follows:
Self configuring: automatic selection of suitable trust computation model according to user trust preferences.
Self optimizing: apply sensitivity analysis to trust data and filter out the un-useful to reduce storage and analysis complexities.
Self protecting: automatic detection and penalty execution for adversaries launching Denial of Trust (DoT) attacks.

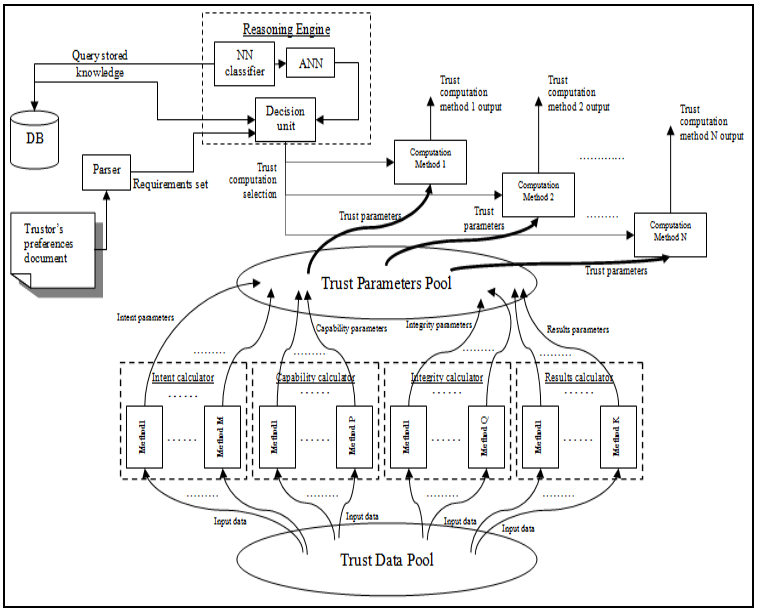

Figure 2 Block diagram illustrating trust computation process

IV. TRUST COMPUTATION

Trust computation is a dynamic process which mainly depends on trustor's preferences, where for each set of preferences for a specific trustor a trust computation methodology is needed to use/define the appropriate trust data and aggregation methods necessary to produce the desired trust report reflecting trustor's view of trust as illustrated in Figure 2. In Figure 2 we have 2 main pools storing trust data and trust parameters values namely trust data pool (TDP) and trust parameters pool (TPP). MM and DM are the sources of trust data in TDP, while the four trust attributes' calculators are the sources of trust parameters in TPP. Each one of the four calculators is comprised of number of methods for computing different parameters values. Users' have the option to define new parameters and add their own calculation method(s) for them. Computing each of the four attributes may be achieved as follows:
Intent: of a party is always issued by a motive; accordingly the party places an agenda that includes the actions to be taken in order to achieve that motive. Finally party's behavior constitutes

execution of these actions. In this scope, Intent of a trustee can be measured using the following: trustee's reputation: where having good reputation may constitute the agenda to increase party's gain. Trustee's customers' credibility: where having reputable clients may constitute an agenda to have high value contracts and referrals to many new clients. Warranty: executing warranties may constitute a behavior towards proving the good intent towards delivering a service. Gain and loss calculations: are the fundamental motives for transacting and can be measured by comparing popularity and diffusion of different trustees. The aforementioned parameters can be calculated as follows:

Reputation: is sum of different parties' opinions in a trustee. An opinion constitutes a set of parameters for evaluating a trustee.

$$\text{Re } p(O) = w * LocOp(O) + (1-w) * ExtOp(O, PV) \quad (1)$$

Rep(O): reputation of trustee O.
LocOp(O): function returning trustor's opinion about trustee O.
w: weight of local opinion.
ExtOp(O,PV): external opinion, which is a function returning different parties' opinions about trustee O according to trustor perspective.

$$ExtOp(O) = \sum_{i=0}^{i=M} \frac{1}{N} \sum_{p=0}^{p=N} PV(p) * fb(i,p) \quad (2)$$

M: total number of evaluation parameters chosen by trustor.
N: total number of evaluations provided by different parties concerning trustee O.
PV: vector holding weights representing trustor's interest in each evaluation parameter.
fb: vector holding evaluation parameters values provided by different parties provided in trustee O.

Popularity: is number of parties who got good service from trustee in the current time sample (figure 3), to the total number of parties.

$$pop(Tu,1) = \frac{1}{N} \sum_{i=0}^{i=s} CFun\ (\sum_{j=0}^{j=T} DL_j(SL_i, Tu)) \quad (3)$$

Tu: trustee
pop(Tu,1): popularity f Tu at time sample number 1
N: total number of parties in the current time sample.
S: number of parties who transacted with trustee during current time sample.
SL: list of parties who transacted with trustee during current time sample.
SL_i: party i who transacted with trustee during current time.
T: number of transactions between party i and trustee in current time sample
DL: list of delivery status of trustees in transactions {1 delivered, -1 not delivered}
DL_j (SL_i,Tu) delivery status of trustee Tu in transaction j with party SL_i in current time sample
CFun: function returns 1 for +ve value and 0 otherwise.

Diffusion: average popularity across history

$$Dif(Tu) = \frac{1}{TSN} \sum_{i=1}^{i=TSN} pop(Tu,i) \quad (4)$$

Tu: trustee
TSN: number of time samples = History/TS
TS: time sample length (figure 3)

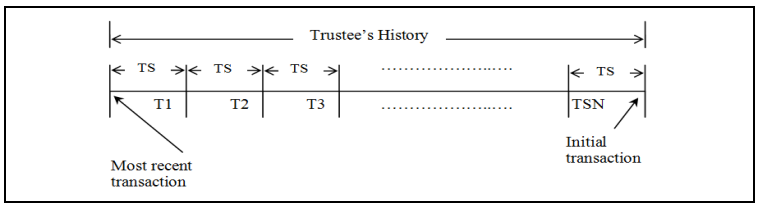

Figure 3 time diagram illustrating trustee's history

Credibility of customers: sum of diffusion of all parties who were served by trustee.

$$CredT(Tu) = \sum_{i=1}^{i=s} Dif(SL_i). \quad (5)$$

Tu: trustee
S: number of parties served by trustee
SL: set of parties served by trustee
Dif(SL_i): diffusion of party SL_i

Warranty Quality: is reputation of warranty and can be calculated using equation1

Integrity: of a party is mainly about delivering what the party advertised and being consistent in his delivery to all parties he interacts with. Consistency implies predictability of party's future behavior. Integrity parameters may include: satisfaction of different parties in service delivery, where parties' expectations of a service is to deliver what is advertised. Satisfaction can be measured by collecting opinions of different parties as illustrated in equation 1.

Capability*:* of a party mainly includes all the resources needed to deliver a service. These resources include: hardware, software, skill set and raw material. Capability can be measured using 2 metrics the first metric indentifies existence of the needed resources at the party under evaluation and the second measures quality of existing resources. Quality of Resource (QoR) can be measured by: reputation of resource and certificates from benchmarking bureaus or organizations evaluating the resource. It is up to user trust to include or define the parameters which are involved in evaluating reputation of a resource. For instance, Bob selects RPM, response time, reliability to evaluate the hard disk resource for file storage service. Reputation of a resource can be measured using equation 1.

Results: of a party are the history of production of that party. Results of a service are what it delivers to its consumers in transactions along its history. Both quality and quantity matter. Thus results may be measured using the weighted average of metrics of Quality of Product (QoP) and quantity. QoP can be measured by: reputation of product and certificates from benchmarking bureaus or organizations evaluating the product. It is up to user trust to include or define the parameters which are involved in evaluating reputation of a product. For instance, Bob selects availability, reliability QoS parameters to evaluate the SW code for file storage service. Reputation of a product can be measured using equation 1.

A P2P Trust Management Architecture (TMA) using the proposed framework is illustrated in figure 4. In the figure, each peer implements the five main components of the framework and each component collaborate with its counter parts on other peers to from an overlay network for decision management, expectation management, analytics (awareness) management, monitoring and data management.

Our work is different from other work in literature in the following aspects:

1. User-centric trust management system; the proposed framework is a flexible and open system which enables users-trust to define, redefine or select any form of trust computation to produce trust reports which suites their trust requirements, hence personalize trust.
2. Generic Framework for trust management; the proposed framework include generic components which can be used to implement any trust management system.
3. Defining and quantifying trust in terms of Intent, Integrity, Capability and Results; the proposed framework doesn't depend solely on past behavior, rather it evaluates resources, declared agenda, honesty and production history to establish trust.

4. Algorithm for quantitative evaluation of trust computation methodologies.

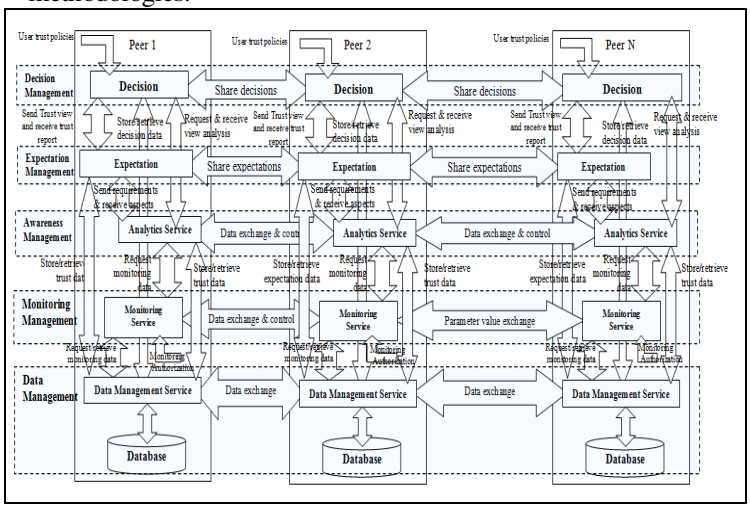

Figure 4 P2P TMS overlay network

V. EXPERIMENTS AND RESULTS

To demonstrate the effect of using trust parameters, we used integrity and results parameters in BitTorrent protocol and illustrate sample results showing protocol performance. Integrity parameters included: Does the peer seed after he finishes download or quits immediately after? Does the peer upload to other peers during download? Does the peer upload valid chunks or sends bogus ones? While results included: number of seeds (finished peers) created. Our evaluation metrics for protocol performance included: download time of peers and total number of peers who finished download. In simulation we implemented 3 types of misbehavior: chunk malicious, free riding and selfish seed behavior. Chunk malicious includes uploading fake chunks to other peers in order to poison their downloaded pieces and to receive illegitimate bandwidth from others. Free riding includes connecting to seeds only and no sharing of downloaded content. While selfish seed includes immediate exit after finish download. Total population used in experiments was 120 peers(90 peers at start of simulation and 30 peers were added after 100k msec). During simulation and for every 100k mSec, 5 peers are either added or removed with equal probability. A condition was set such that total population does not exceed 120 peers. Upload bandwidth value of a peer is quarter that of its download bandwidth value. Download bandwidth distribution among peers is 640 Kbps with probability 1/4, 1 Mbps with probability 1/4, 2 Mbps with probability 1/4 and 4 Mbps with probability 1/4. Peers in simulation are chunk malicious with probability 1/8, free riders with probability 1/8 and are selfish seed with probability 1/8. A condition was set such that free riders cannot be chunk malicious and vice versa. Number of seeds in simulation was 15% of total population, number of old peers (i.e. peers who already downloaded x % of file) was 5% of total population and number of new peers (i.e. peers haven't downloaded any of the file) was 80% of total population. The value of x for old peers is randomly set at initialization. Peers download a 100M byte file. Figure 5 and 6 illustrates sample of results. From the figures we may observe the following: In figures 5, for new peers, download time falls between 0.5k and 3.75k sec when using trust parameters, while it falls between 1.5k and 7k sec in case of no trust parameters used. For old peers, download time falls between 0.2k and 1.5k sec when using trust parameters, while it falls between 0.2k and 5k sec in case of no trust parameters used. In figure 6, we may observe that number of new peers in case of using trust is 85% larger than that of not using trust, while number of old peers in case of using trust is 11% less than that of not using trust. The 11% difference is caused by number of old peers who are added to the network already owning large portion of the file.

The results show that:

1. Peers' download time is reduced when using integrity and results parameters.
2. Increase number of peers who finish download when using integrity and results parameters.

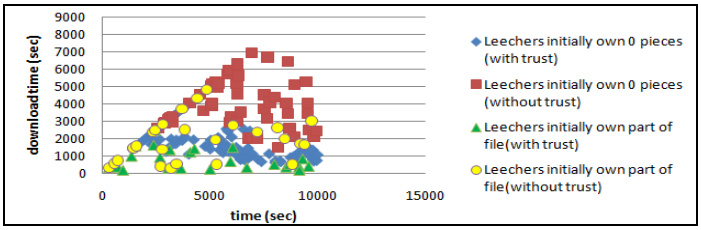

Figure 5 Download time for peers who finished download.

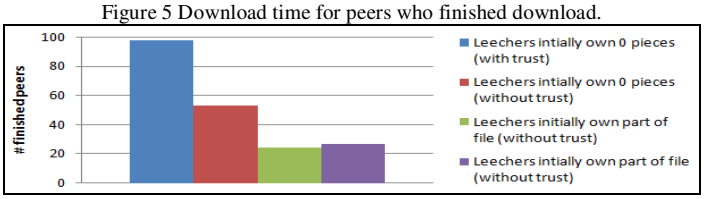

Figure 6 Number of peers who finished download

VI. ONGOING AND OPEN RESEARCH

Includes the following:

- What are the indicators of trust data being stale or useless?
- How can we manage collective trust versus individual trust in various forms of compositions and cooperation?
- How can we immunize the system against different forms of denial of trust (DoT) attacks, including but not limited to: collusion, white washing and trust exploitation.

REFERENCES

[1]. D. Gambetta "Can we trust trust?", In D. Gambetta, editor, Trust: Making and Breaking Cooperative Relations, pages 213{237. Department of Sociology, University of Oxford, 2000

[2]. Trung Dong Huynh, "A Personalized Framework for Trust Assessment", ACM Symposium on Applied Computing, 2009.

[3]. Audun Jøsang and Claudia Keser and Theo Dimitrakos, "Can We Manage Trust?", Third International Conference (iTrust), 2005.

[4]. Wang Yu Zhang Qiuyue Jiang Ying YangZhou Univ and Yangzhou , "A Trust Management Model Based on Multi-agent System ", CCCM '08, 2008.

[5]. Chavez, A, and Maes, P. Kasbah, "An Agent Marketplace for Buying and Selling Goods", International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96), 1996.

[6]. Thomas Repantis and Vana Kalogeraki, "Decentralized Trust Management for Ad-Hoc Peer-to-Peer Networks", proceedings of the 4th international workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC 2006), 2006.

[7]. Barry Smyth and Dennis McLeod, "Personalizing Trust in Online Auctions", proceedings of the third starting AI researchers' symposium, 2006.

[8]. Daeseon Choi, Seunghun, Younho Lee and Yongsu Park, "Personalized Eigen Trust with the Beta Distribution", ETRI journal, volume 32, number 2, 2010