Trust Blueprints and Use Cases

Deepak Vij, Ishita Majumdar, Naveen Dhar
FutureWei Technologies, Inc.
US Central Research Institute
Santa Clara, California, USA
e-mail: {Deepak.vij, Ishita.majumdar, Naveen.dhar}@huawei.com

George Vanecek, Jr.
FICO
Product and Technology Organization
San Jose, California, USA
e-mail: {georgeVanecek}@fico.com

***Abstract*—As the scope of current distributed computing model envisioned by the contemporary cloud computing environment enlarges to future federated *Intercloud* and ubiquitous and pervasive computing models such as *Internet of Things* (IoT), many difficult problems and challenges arise. Security is one of the most important concerns of such a computing environment. Current security mechanisms are very static, inflexible and not granular enough to make efficient and informed decisions in the *Service Provider* based computing environment. The conventional trust mechanisms in place are inadequate at addressing granular level trust issues in the highly distributed open environments. In this paper, we explore various Trust Management schemes and blueprints for enabling a framework that interested parties can use to determine the trustworthiness of disparate and heterogeneous computing entities. The paper also enumerates various business use case scenarios articulating how such a *Trust Management* framework would be invaluable for addressing the current as well as future computing environments needs.**

Keywords-Trust Management; Internet of Things; Cloud Computing; Device Mobility; Authentication; XACML; Authorization; Intercloud; Context-Awareness; Evidence-based Trust.

I. INTRODUCTION

In recent years, the contemporary highly distributed and heterogeneous cloud computing design pattern has ushered in an era of tremendous breakthroughs in geographical distribution, resource utilization efficiencies, and infrastructure automations. Yet, as the scope of current distributed computing model envisioned by the contemporary cloud computing environment enlarges to future federated *Intercloud* and ubiquitous and pervasive computing models such as *Internet of Things* (IoT), many difficult problems and challenges arise. Security is one of the most important concerns of such a computing environment. Current security mechanisms are very static, inflexible and not granular enough to make efficient and informed decisions in the *Service Provider* based computing environment. The conventional trust mechanisms in place are inadequate at addressing granular level trust issues in the highly distributed open environments.

Typically, security architecture facilitates the trust mechanisms between two entities whereby the *truster* is an entity that trusts another entity, the trustee, and the *trustee* is an entity that is being trusted. Traditional security architecture is built around regulating access to target resources or services by granting certain *authorization* rights to *authenticated* entities (*trustee*). Authentication and authorization processes work in tandem as part of the overall access management architecture.

Authentication is the process through which an entity (e.g., a person, device or service) provides sufficient credentials such as passwords, tokens, public key certificates (using public-key infrastructure - PKI) or secret keys to satisfy access requirements of a resource, based on a pre-existing membership of that entity. Authentication is essentially a process of ensuring irrefutable knowledge of the trustee (entity). It enables users, computers or devices to know with whom they are communicating.

Authorization, on the other hand, is the process used to determine *what* services or resources an irrefutably known authenticated user, computer or a device, can access. Authorization is a process for protecting resources and information while allowing seamless access for legitimate use of those resources. It allows security administrators to enact authorization entitlement policies in an easy to maintain and simple to monitor fashion.

Traditionally, authentication services helped a computer identify a person attempting to gain access, or to *log on*. In the last decade or so, authentication needs have evolved to go beyond the traditional scope of simple *log on* process. These new authentication schemes include PKI based *digital signatures* technique. Cryptographic algorithms-based digital signatures, as the name implies, mark an electronic document (digital certificate) to signify its association with an entity. A trusted third party that certifies the digital signature issues the digital certificate.

Irrespective of the authentication mechanism, a successful authentication process assigns a static/fixed *role* to the *trustee* (or *requester*). The authorization process, in turn, determines the access control based on the fixed role assignment. It is important to note that access control to resources is not assigned directly to the *requester* entities but to abstractions known as *roles*. As *entities* are assigned to different roles, they indirectly receive the relevant access control privileges.

With the distributed computing and cloud models moving towards a federated *Intercloud* model [1][2][3] along with the near ubiquity and pervasiveness of smart devices

 ISBN: 978-1-61208-324-7

and sensors (*Internet of Things*), these classic authentication and authorization methods pose challenges. With the humanization of Internet technologies whereby smart devices are increasingly taking on more intelligent and autonomous roles for their owners, it is equally important for services to obtain real-time and context-specific information about trustworthiness of its users.

Effective provisioning and delivery of application services in an efficient and more importantly, in a highly secured manner, are the key challenges faced going forward. It has become increasingly important to be able to generate dynamic, granular security policies for federated ubiquitous systems.

Current security techniques that are widely being employed include sand-boxing, PKI based cryptography, and other access control and authentication mechanisms. These mechanisms, however, are very static, inflexible and not granular enough in order to make efficient and informed decisions for the future computing environment.

Specifically, explicit *trust* [4][5][6], for the most part, is conspicuously left out of the contemporary fabric of the Internet. Contemporary rudimentary *trust* mechanism applies to individuals only and is not made integral part of the fabric of the Internet and the Web itself. Current conventional trust mechanisms are inadequate at addressing granular level and real-time, contextual trust issues in the highly decentralized open environments.

Trust needs to be established from the viewpoint of both parties (*Service Requesters* and *Service Providers*). *Service Requester's* trust with respect to the *Service Provider* may be different from *Service Provider's* trust with respect to the requester. From *Service Requester's* perspective, trust towards the *Service Provider* signifies correct and faithful allocation of resources as part of the efficient execution environment with respect to established trust and other security policies. From *Service Provider's* perspective, trust towards *Service Requester* will generate a legitimate request consisting of virus free code and will not produce malicious results and does not temper other results/information/code present at *Service Provider's* end.

With this as the backdrop, this paper proposes detail blueprints of a *Trust Management* system describing the key components within the proposed system and how these components interact with each other. The paper explores various *Trust Management* schemes and blueprints for enabling a framework so that interested parties can determine the trustworthiness of disparate and heterogeneous computing entities. The paper also enumerates various business use case scenarios articulating how such a *Trust Management* framework would be invaluable for addressing the current as well as future computing environment needs.

This paper describes various components of the *Trust Management* system in detail and strives to provide a general foundation for building various constituents of the trust system. However, the paper does not delve deep as far as describing the actual mathematical algorithms/functions and in-depth technology details for underlying components. Our future work will publish such in-depth details for each and every components of the *Trust Management* system.

We will attempt to demonstrate our proposed Trust Management system's paradigm shift in comparison to the typical role-based access control computer security model. In the future, with open and highly decentralized environment where entities are dynamic in nature, the identity of every entity is not known in advance. In such an environment, traditional fixed *role* assignment becomes an irrational and ad-hoc exercise and not viable at all. Although, PKI based credentials mechanism implement a notion of trust, this trust is static and binary in nature. Access privileges are allowed or credentials are rejected and the *trustee* entity does not get the access rights. In such a highly de-centralized environment, the static role assignment needs to be evolved in such a manner that it enables a dynamic trust value assigned to a *trustee* entity. Trust based authorization mechanism, in turn, leverages the dynamic trust value assigned to the *trustee* entity and makes the access control decisions accordingly in a highly dynamic manner.

The rest of the paper is organized as follows: Section II outlines a brief description of *Trust based Paradigm Shift* as well as formal definitions related to *Trust Paradigm*. Section III outlines the proposed overall *Trust Management* system blueprints. Section IV enumerates various business use cases. Finally, Section V presents our conclusions.

II. TRUST BASED PARADIGM SHIFT – AN OVERVIEW

Trust reflects the expectation one actor has about another's future behavior to perform expected activities dependably, securely, and reliably based on experience collected from previous interactions and relevant external sources. Our definition of *Trust* is based on a paradigm shift assumption that formalizes trust so that trust considerations may be added to how future services and computer systems communicate amongst each other.

The key tenet of our proposed trust model is that the *truster* decides permissions based on Principle's set of attributes instead of principle's identities. Trust attributes may include *Evidence-based* as well as *Reputation-based* attributes whereby entities endow other unknown entities in order to gain access to services or resources in a highly federated distributed environment. Traditional mechanisms, on the other hand, are typically based on the key assumption that identity of every entity is known in advance.

This section explains the overall trust based paradigm that includes, trust properties, trust entities, trust contexts and situations and belief policies and intent.

A. Trust Properties

We define trust as possessing the following properties:

- Trust is not *Transitive*; if I trust Alice and Alice trusts John, that does not mean I should trust John. Essentially, trust relationship between two entities is a *vector* that consists of trust value in conjunction with direction.
- Trust is *Contextual* [7]. A truster may have different and independent sets of trust relationships given her different roles or configurations. For example, a person can be a tourist, a hobbyist, an employee, a father, a husband, a consultant, a teacher or a

volunteer, to name a few; a mobile device may be used in a security zone with restrictions or in a public place playing games. Trust relationships vary depending on such situations that arise from these contexts.

- Trust is *Granular*. Trust is an assessment of many trust-related distinct scores taken from the evidences provided, not just one cumulative and global score value.
- Trust is *Belief-based*. Different truster's have different beliefs of trust. Some trust until trust is broken; others distrust until trust is earned.
- Trust assessment is *Situational*. Which context applies to the question of "Do I trust?" depends on the situation.
- Trust assessment is *Intent-driven*. A situation defines the context, but the intent defines the trust scoring.
- Trust is *Continuously Reevaluated*. Yesterday one may trust, today they do not, while tomorrow they will again. Why? Situations, contexts, and evidence. Scores change based on continuous assessment of the trustee's relationships – Dynamic Trust Establishment. A trust-based paradigm shift takes the blind trust method and introduces a trust query allowing both the client and the server to proceed based on their latest and up-to-date understanding of the trust relationship between the two entities (as shown in Figure 1 below). In such a methodology, the trust is a property that leverages dynamic verification and updates for such trust relationships, taking contexts, and entity specific (e.g., personal) policies into account.

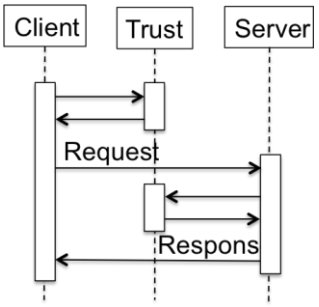

Figure 1. Dynamic Trust between client and a server

B. Entities

Entities are the objects between which trust is established and maintained. An *entity* is defined as any person, place, or thing with a distinct and independent existence that may trust or be trusted.

Each entity needs to be uniquely identified. One possible identification mechanism may be the *Extensible Resource Identifier* as defined by the XRI [8] Technical Committee at OASIS.

As shown in Figure 2 below, entities have a duality as either:

- a *truster* which positions the entity as the one that is trusting another (i.e., a trustee), or
- a *trustee* which positions the entity as the one that is being trusted by a truster.

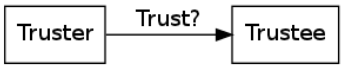

Figure 2. Trusters and Trustees

Trusters have a belief policy and one or more contexts.

C. Truster Contexts and Situations

Truster contexts are a way to partition an entity's singular notion of trust into different sets of related trust domains. To answer questions of trust, one first has to establish the context. The specific contexts are selected based on specific *situations* that are present at the time of trust determination. Consider, as an example, the many contexts that a person can be a part of, as the example in the Figure 3 below illustrates. All these examples are contexts. These differentiate in how a truster evaluates trust or risk assessment.

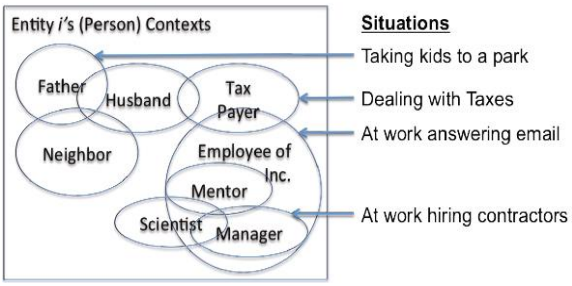

Figure 3. Examples of situations that select which truster's context applies to those situations

D. Belief Policies and Intent

Context and situations alone are not sufficient to assess trust. Each entity must be able to apply its own belief in trust assessment. A *Belief Policy* can be defined that helps determine how trust values are interpreted to derive a Boolean trust value for a specific scenario. A final trust score of 0.8 may signal one to trust but another not to. Belief Policies maintain trust value thresholds, and allow the entity to change its belief over time as trust is gained or reduced.

In addition to the Belief Policy, the intent of the situation has to also be taken into account. Consider an example. A situation in which a person is at work on Monday talking to a non-employee in a conference room with a human resources representative present may identify the context as that of an interview. But the interviewers (truster) intent may affect the trust determination of the interviewee (trustee). If the interviewer's intent is to hire a friend its risk acceptance is higher and so is his trust. If the interviewer's intent is to hire a replacement, their trust may be lower. Thus, intent is an adjustment to one's belief policy is important in allowing for more accurate trust assessment of a given context identified by a specific situation.

 ISBN: 978-1-61208-324-7

III. TRUST MANAGEMENT SYSTEM OVERVIEW

The following schematic as shown in Figure 4 below captures details for the *Trust Management System* as a whole. Subsequent subsections describe all these components in more detail.

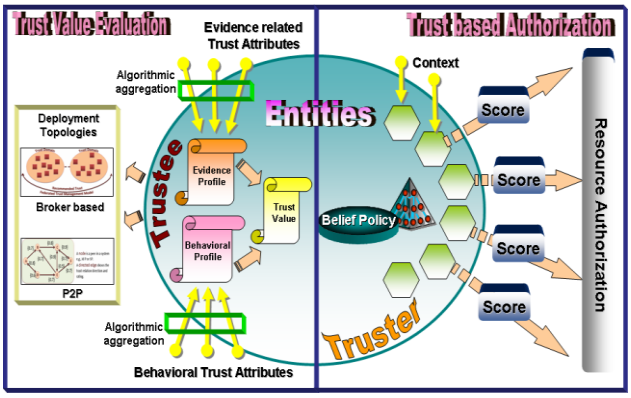

Figure 4. Trust Management System Overview

A. Trust Value Evaluation

As mentioned in the introduction, in a highly de-centralized environment, the contemporary static role assignment mechanism needs to be evolved in such a manner that it enables a dynamic trust value assignment to a *trustee* entity. Trust based authorization mechanism, in turn, leverages the dynamic trust value assigned to the *trustee* entity and makes the access control decisions accordingly in a highly dynamic manner.

Trust Value Evaluation process essentially entails collecting the relevant information necessary to establish trust relationship and, at the same time, dynamically monitors and adjusts the existing trust relationship. This process assigns a single-valued scalar numeric value in the range [0..1]. Lower trust value signifies lack of trust, while higher value denotes more trustworthiness of an entity. A trust value of 0 represents the condition with the highest risk for an entity and 1 representing the condition that is totally risk-free or fully trusted.

As mentioned earlier, trust is always related to a particular context. An entity A needs not trust another entity B completely. Entity A only needs to calculate the trust associated with B in some context pertinent to a situation. The specific context will depend on the nature of application and can be defined accordingly. Based on our current model, trust is evaluated under a single context only.

Trust Value for an entity is determined by a combination of the following two models:

- ***Evidence-based*** model, an appropriate trust value is assigned to an entity based on some evidence such as self-defense evidence etc. explicitly manifested by the entity.
- ***Reputation-based*** model [9][10][11]12], in which Direct Experience coupled with Indirect Recommendation/s establishes the trust value of an entity.

Based on these two criteria, the trust rating value could be obtained by applying different mathematical functions/algorithms to all the relevant trust attributes applicable for an entity. All of the trust attributes (*Evidence-based* as well as *Reputation-based* attributes) would be assigned respective weights as part of the trust calculation algorithm.

The following three sub-sections describe brief summary of various trust value evaluation models. These sub-sections provide general foundation and grounding for these models and complexities involved. As part of our future work, we will delve deeper as far as describing the actual mathematical algorithms/functions and in-depth technology details for these trust value evaluation models.

1) Evidence-based Trust Model

In the ***Evidence-based*** model, trust is considered as a set of relationships established with the support of evidence. Evidence can be anything a policy requires to establish a trust relationship, such as attendance list, annual report, or access history. For example, in case of a web service resource, the intrinsic trust value calculation algorithm may factor in web service attributes such as:

- Dependability characteristics such as Accessibility, Availability, Accuracy, Reliability, Capacity, Flexibility etc.
- Self-Defense characteristics such as Authentication, Authorization, Non-repudiation, encryption, privacy, Anti-Virus Capabilities, Firewall Capabilities, Intrusion Detection Capabilities etc.
- Performance characteristics such as Latency, Throughput etc.
- and much more …

2) Reputation-based Trust Model

In the ***Reputation-based*** model, on the other hand, trust is motivated from human society, where human beings get to know each other via direct interaction and through a grapevine of relationships. In a large distributed system, every entity can not obtain first-hand information about all other entities. As an option, entities can rely on second-hand information or recommendations. Reputation is defined as "perception that an entity creates through past actions about other entity's intentions and track record".

The reputation assessment of an evaluated entity by an evaluator entity involves collecting information such as:

- *Direct Trust*, the evaluator's own interaction experiences with the evaluated entity; if the evaluator entity has first-hand experience of interacting with evaluated entity in the past.
- *Recommender Trust*, recommendation from peers who have interacted with the evaluated entity before. Attributes such as *Prior Success Rate*, *Turnaround Time*, *Cumulative Site Utilization* etc. are few examples of Reputation trust. Time is the key dimension for reputation. Reputation builds with time – reputation enhances or decays as the time goes by.

The recommendation protocol is straightforward. For example, entity A needs a service from entity D. A knows

 ISBN: 978-1-61208-324-7

nothing about the quality of D's service, so A asks B for a recommendation with respect to the service category, assuming that A trusts B's recommendation within this category. When B receives this request and finds that it doesn't know D either, B forwards A's request to C, which has D's trustworthiness information within the service category. C sends a reply to A with D's trust value. The path (A)X(B)X(C)X(D) is the *recommendation path*. When multiple recommendation paths exist between the requester and the target, the target's eventual trust value may be the average of the values calculated from different paths.

As mentioned earlier, *Time* is the key dimension for reputation. As in relationship, trust may decrease with time. For example, if an entity has not interacted with another entity for some time, then the trust value between these two entities is likely to be weaker. To account for *Time* dimension, a *time decay factor* needs to be included as part of the trust calculation algorithm.

3) Trust Normalization Policy and Unit of Measure (UOM) Standardization

As mentioned earlier, all of the trust attributes (*Evidence-based* as well as *Reputation-based* attributes) would be assigned respective weights as part of the trust calculation algorithm. However, lack of a *standard unit of measure* for quality of these attributes may pose a huge challenge. Also, without trust normalization policy, it would be difficult to deterministically determine the weights and the correct set of attributes to be included at the time of trust value calculation process. Such a *deterministic* approach would be a daunting task, nonetheless.

During evaluation of a trust value, a truster may assign different weights to the different factors that influence trust. The weights will depend on the trust evaluation policy of the truster. So, if two different trusters assign two different sets of weights, then the resulting trust value will be different. The trust normalization policy addresses this particular issue. The trust normalization policy to go along with the *Evidence-based* model and *Reputation-based* model forms the complete *truster's* trust evaluation policy.

B. Trust Management Topologies

The previous section explains all the complexities of determining trust value of an entity. There are primarily two topologies to support such a trust value evaluation process.

- A centralized broker-based trust aggregation topology.
- A trust overlay network based peer-to-peer decentralized topology.

Whether a trust topology is centralized or decentralized determines the feasibility and complexity of a trust value evaluation mechanism. In a centralized system, a central node will take all the responsibilities of managing reputations for all the members. In a decentralized system, e.g., a peer-to-peer system, there is no central node. The members in the system have to cooperate and share the responsibilities to manage reputation.

Generally speaking, the mechanisms in centralized systems are less complex and easier to implement than those in decentralized systems. But, they need powerful and reliable centralized servers and a lot of bandwidth for computing, data storage, and communication.

The following two sub-sections give a brief summary of these two topologies. These sub-sections provide general foundation and grounding for various topologies and complexities involved. As part of our future work, we will delve deep as far as describing the actual in-depth mathematical algorithms/functions and technology details for these deployment topologies.

1) Trust Broker Topology

As shown in Figure 5 below, in a centralized broker-based [13] trust aggregation topology, the entire trust landscape is divided into trust domains. Trust agents/entities inherit the trust properties of the domain they are associated with. This increases the scalability of the overall approach.

Trust entities rely on the trust broker to manage trust. As Domain trust agents, trust brokers store other domain's trust information for inter-domain cooperation. Essentially, the trust information stored reflects trust value for a particular resource type (compute, storage, etc.) for each domain. Trust Brokers also recommend other domains trust levels for the first time inter-domain interaction.

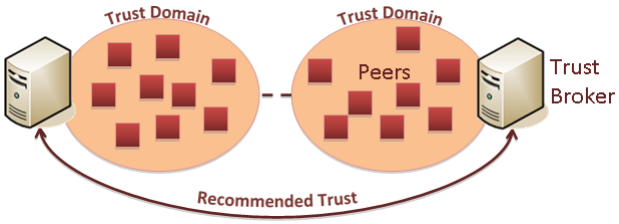

Figure 5. Trust Broker based Federated Trust Management Topology

A decentralized Distributed Hash Table (DHT) based 3rd Party Trust Management may be used for efficiently managing various trust domains. Individual entities themselves do not need to take any responsibilities for managing the trust model. Instead, the responsibility is delegated to the 3rd party trust broker node. However, this approach has classical disadvantages of a typical centralized methodology – performance bottlenecks, single point of failure etc.

2) P2P Topology

Peer-to-Peer (P2P) Trust Management topology [14][15][16][17], on the other hand, does not employ any centralized server. As shown in Figure 6 below, each peer maintains a local trust table to store trust information of neighboring nodes. Trust Vector Aggregation Algorithm can infer indirect trust among peers. Each member entity itself has to cooperate and share responsibilities to manage the local level trust index. Trust value for all nodes is determined algorithmically.

 ISBN: 978-1-61208-324-7

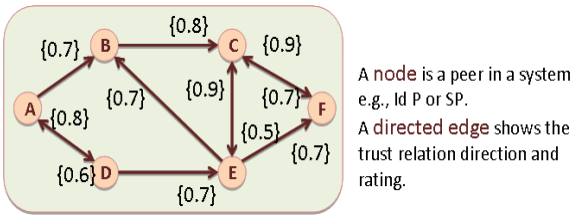

Figure 6. P2P Topology

In such a decentralized environment, finding *Most Trustable Path* so that the trust path yields the highest trust value from thousands or millions of peers is a mammoth challenge, to say the least. Also, trust propagation to each peer in a vast network of peers is yet another challenge that needs to be addressed as part of the overall Peer-to-Peer (P2P) topology.

C. Trust based Authorization

As stated earlier in the introduction section, in an open and highly decentralized environment where entities are dynamic in nature, the identity of every entity is not known in advance. In such an environment, the static role assignment needs to be evolved in such a manner that it enables a dynamic trust value assignment to a *trustee* entity. Trust based authorization mechanism, in turn, leverages the dynamic trust value assigned to the *trustee* entity and makes the access control decisions accordingly in a highly dynamic manner. As mentioned earlier, the *truster* decides permissions based on Principle's set of attributes instead of principle's identities. Trust attributes may include *Evidence-based* as well as *Reputation-based* attributes as covered in the previous section.

In very simplistic terms, *Trust based Authorization* process is a mathematical equation. On one side of the equation is the *Security Demand* (**SD**) of an entity. On the other side of the equation is the *Trust Value* (**TV**) that reveals of another entity. These two must satisfy a security assurance condition so that ***TV*** >= ***SD***.

As mentioned earlier, trust relationship between two entities is a *vector* and is always related to a particular context. The trust vector is a vector of trust value and trust direction, where trust value is defined as a real number in the range [0..1] and direction is defined as a directed edge in the trust graph. The edge in a graph represents the rating for a combination of all direct transactions between two peers. Trust value itself is composed of three key components – Evidence, Direct Experience, and Recommendations from others. As shown in equation (1) below, trust relationship between entity A and B in simple terms can be described as:

$$TV(A \rightarrow^{c} B) = [{}_{A}E^{c}_{B}, {}_{A}D^{c}_{B}, {}_{O}R^{c}_{B}] \quad (1)$$

Here the value ${}_{A}E^{c}_{B}$ represents the level of evidence demonstrated by entity B to entity A under context c. The value ${}_{A}D^{c}_{B}$ represents the magnitude of direct experience of entity A in relation to entity B under context c. The value ${}_{O}R^{c}_{B}$ represents the cumulative effect of all recommendations from all other entities for entity B under context c. Each of these three components is expressed in terms of a numeric value in the range [0..1].

We propose a XACML-compliant policy management system as part of the trust based authorization scheme. XACML provides a standardized language and method of access control and policy enforcement.

eXtensible Access Control Markup Language (XACML) [18] is an XML-based language for access control that has been standardized in OASIS. XACML describes both an access control policy language and a request/response language. The policy language is used to express access control policies (who can do what when). The request/response language expresses queries about whether a particular access should be allowed (requests) and describes answers to those queries (responses).

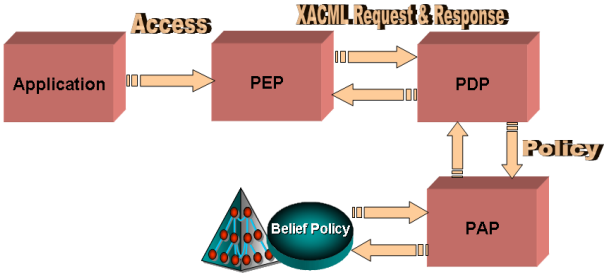

Figure 7. OASIS XACML Authorization Environment

In a typical XACML usage scenario, a subject (e.g. human user, device) wants to take some action on a particular resource. As shown in Figure 7 above, the subject submits its query to the entity protecting the resource. This entity is called a Policy Enforcement Point (PEP). The PEP forms a request (using the XACML request language) based on the attributes of the subject (trust value in our case), action, resource, and other relevant information. As shown in Figure 8 below, the PEP then sends this request to a Policy Decision Point (PDP), which examines the request, retrieves policies (written in the XACML policy language) that are applicable to this request, and determines whether access should be granted according to the XACML rules for evaluating policies. That answer (expressed in the XACML response language) is returned to the PEP, which can then allow or deny access to the requester. Policy Administration Point (PAP) is used to get to the policies; the PDP uses the PAP where policies are authored and stored in an appropriate repository.

 ISBN: 978-1-61208-324-7

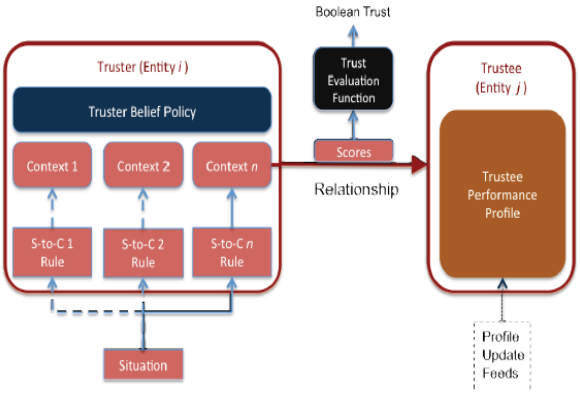

Figure 8. Trust based Decisioning Process

In this section, we described a brief summary of trust based authorization framework. The section provided general foundation and grounding for various complexities involved.

D. *Trust Management Conceptual Layered Architecture*

As shown in Figure 9 below, the key ingredients of the conceptual architecture are:

- *Trust Rating Layer*
- *Trust Aggregation Layer*
- *Trust Access Layer*

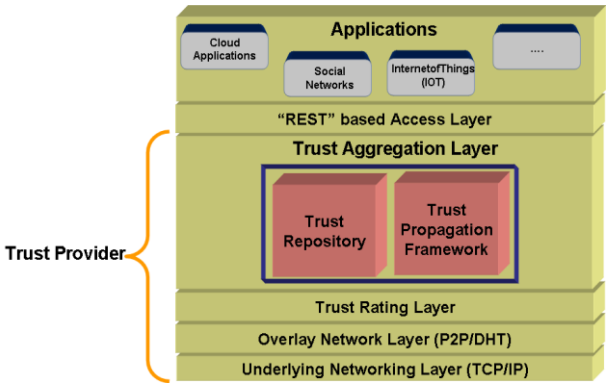

Figure 9. Trust Management Conceptual Architecture

The details for *Trust Rating Layer* have already been described earlier, as part of the ***Trust Value Evaluation*** section.

Trust Aggregation Layer is responsible for aggregation of distributed trust scores in a peer-to-peer environment. It is based on mathematical algorithm for fast and lightweight trust score aggregation.

Trust Access Layer provides entities to extract trust information from the trust model. This REpresentational State Transfer (REST) API specification is for the interface of the Trust System. The API set consists of methods related to:

- Entities (Create, List, Find, Entity Details, Modify, Delete)
- Entity Context (Create, List, Find, Entity Details, Modify, Delete)
- Entity Belief Policy (Get, Modify)
- Entity Relationship (Create, Find, List, Get, Modify)
- Trust Determination
- etc.

IV. TRUST MANAGEMENT – USE CASES

This section enumerates various business use case scenarios articulating how such a *Trust Management* framework would be invaluable for addressing the current as well as future computing environment needs. The following are few of the business use cases in which the proposed *Trust Management* framework can play a huge role as part of the next generation highly distributed computing environment.

A. *NextGen Trust aware Federated Identity Management*

Federated Identity Management has existed for a while. However, almost all existing approaches to identity federation are based on static relationships. In a static federation, relationships among identity providers (IdPs) and *Service Providers* (SPs) are manually pre-configured in their metadata repository. The question of whether an entity can trust another depends on if they can find each other in the pre-wired metadata repository, thus this question cannot be answered in a dynamic manner due to the static nature of the meta data.

Current *Federated Identity Management* solutions lead to problems with scalability and deployment in real-time dynamic environment such as mobile networks and *Internet of Things* in general. Firstly, every new relationship between any two entities must be added manually as such a static federation cannot be quickly and easily expanded to accommodate hundreds, thousands or even millions of IdPs and SPs nodes. In essence, a static *Identity Federation* cannot be deployed in a real-time environment like a mobile network or in *IoT* environment where devices may potentially access each other across federation boundaries and at any time.

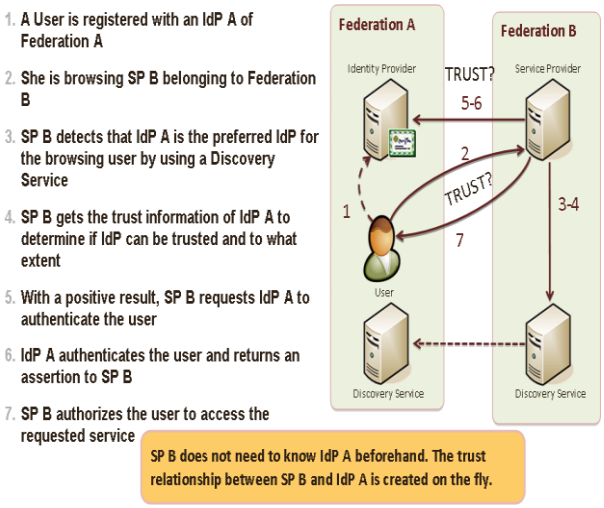

Figure 10. Trust aware Federated Identity Management

The proposed *Trust Model* enables a dynamic federation environment, in which the IdPs and SPs will be regarded as peers of a trusted network that evolves over time. A trust relationship between two entities is regarded as a network connection. As shown in Figure 10 above, in such a dynamic federation, an SP does not need to know an IdP beforehand. A trust relationship will be created on demand and the trust value, namely how much an IdP can be trusted will be determined on the fly.

B. *Trust aware Network Virtualization*

Network Virtualization enabled Bandwidth Trading - Most network traffic does not flow in steady and easily predictable streams, but in short bursts, separated by longer periods of inactivity. This pattern makes it difficult to predict peak loads. *Bandwidth on Demand* is useful for applications, such as backups, files transfers, synchronization of data bases, and videoconferencing, and allows the user to pay for only the amount of bandwidth used. It is a technique that allows the user to add additional bandwidth as the application requires it.

Traditionally, in a network virtualization environment, trust, if addressed, is generally addressed from the security and privacy point of view only. Authentication, authorization, access control, ensuring integrity of information and protecting the source of information are used to provide a secure virtual network. However, there are other trust related aspects that need to be taken into consideration. For example, we should be able to trust that an underlying infrastructure provider will fulfill its part of the SLA by providing the agreed Quality of Service (QoS).

A SP assesses the quality of service of an infrastructure network provider involved in a virtual network in terms of availability of resources, reliability, confidentiality and integrity, and adaptability to network conditions. The feedbacks sent by different *Service Providers* are gathered and stored. A *Trust Management Service* is used to keep track of trust data of infrastructure providers. As shown in Figure 11 below, while mapping a virtual network, the SP will take into consideration the reputation of the infrastructure providers.

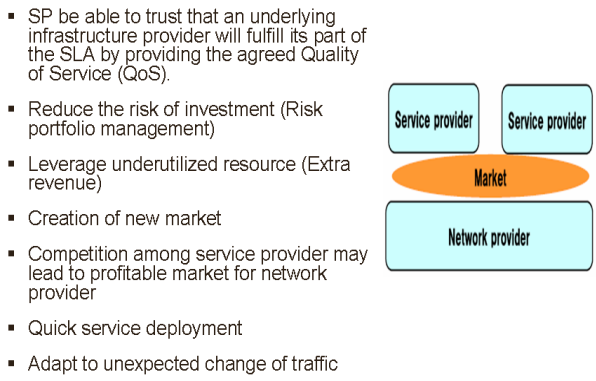

Figure 11. Trust aware Network Virtualization

Mapping a virtual network request requires the selection of specific nodes and links according to the requirement of a *Service Provider* in terms of resources (e.g., location and CPU of the nodes, and the bandwidth of the links) and cost. If *Service Providers* consider only the cost, the infrastructure providers may be tempted to reduce the price by minimizing the quality of the physical underlying network.

To make the right decisions, trust information of the infrastructure providers is taken into account while performing a Virtual Network (VN) mapping. Avoiding un-trusted physical network providers, where failure of nodes and links could easily happen, will improve the service provided to the users. *Service Providers* may reward reputable infrastructure providers by higher priority/probability of involvement in future VN mapping requests.

C. *Trust Model for Device Mobility*

There is a need for Trust-based Mobile Device Control Management for Enterprises

- Mobile devices are set up for only one security domain with static access policy limit usability and increases costs.
- Enterprises are adopting hybrid public/private cloud services.
- Enterprise security needs must balance personal privacy needs and usability.
- Enterprises must accept the coexistence of personal and corporate apps and data.
- Enterprises can adopt dynamic and real-time control policies based on managing risk with trust.
    - Granular Trust Attributes are defined for users, devices, apps, etc.
    - Trust is learned and continually verified and adjusted.
    - Trust is mutual and bi-directional, so are the policies.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we described various components of the *Trust Management* system in great detail and strived to provide a general foundation for building various constituents of the trust system. However, it does not delve deep as far as describing the actual mathematical algorithms/functions and in-depth technology details for underlying components. Our future work will publish such in-depth details for each and every components of the *Trust Management* system.

In order to make this a reality, an operational trust management system must be experimented with in a live public trial. To that regard, we are working towards establishing the *Trust Management Testbed* by collaborating with various well known academic institutions and industry leaders.

ACKNOWLEDGMENT

We would like to thank:

- Farag Azzedin & Muthucumaru Maheswaran of University of Manitoba and TR*Labs* Winnipeg, Manitoba, Canada, for their work on "Evolving and Managing Trust in Grid Computing Systems".
- Huaizhi Li and Mukesh Singhal of University of Kentucky for their work on "Trust Management in Distributed Systems".

REFERENCES

[1] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond, and M. Morrow, Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability, in Proceedings of ICIW '09, the Fourth International Conference on Internet and Web Applications and Services, 2009, pp. 328-336.

[2] R. Buyya, S. Pandey, and C. Vecchiola: Cloudbus toolkit for market-oriented cloud computing, in Proceedings of 1st International Conference on Cloud Computing (CloudCom), 2009.

[3] D. Bernstein and D. Vij, Intercloud Exchanges and Roots Topology and Trust Blueprint, in Proceedings of the IEEE 2011 International Conference on Internet Computing, Las Vegas, USA, 2011.

[4] E. F., Chrchill, On Trust Your Socks to Find Each Other, Yahoo Interactions, March 2009.

[5] K. Thompson, Reflections on Trusting Trust, Communications of the ACM, August 1984.

[6] L. J. Hoffman, K. Lawson-Jenkins, and J. Blum, Trust Beyond Security: An Expanded Trust Model, Communications of the ACM, July 2006, 49(7):94-101.

[7] M. C Huebscher and J. A McCann, A Learning Model for Trustworthiness of Context-awareness Services, Proceedings of the 3rd Int'l Conf. on Pervasive Computing and Communications Workshops, 2005, pp. 120-124.

[8] OASIS Extensible Resource Identifier (XRI) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xri, [retrieved: December, 2013].

[9] J. Goldbeck and J. Hendler, Inferring Reputation on the Semantic Web, in Proceedings of the 13th International World Wide Web Conference, May 2004.

[10] F. G. Marmol and G. M. Perez, Security threats scenarios in trust and reputation models for distributed systems, Elsevier, Computers & Security 28, 2009, pp. 545-556.

[11] S. Ramchurn, C. Sierra, L. Godo, and N. Jennings. Devising a trust model for multiagent interactions using confidence and reputation, International Journal of Applied Artificial Intelligence, 2005, 18(9–10):91–204.

[12] J. Goldbeck, Semantic Web Interaction through Trust Network Recommender Systems in end user semantic web interaction workshop at the 4th international semantic web conference, 2005.

[13] K-J. Lin, H. Lu, T. Yu, and C. Tai, Reputation and Trust Management Broker Framework for Web Applications, in e-Technology, e-Commerce and e-Service, 2005. EEE '05, The 2005 IEEE International Conference, 2005.

[14] R. Zhou and K.Hwang, Trust Overlay Networks for Global Reputation Aggregation in P2P Grid Computing, *IEEE IPDPS*, 2006.

[15] T. Repantis and V. Kalogeraki, Decentralized Trust Management for Ad-Hoc Peer-to-Peer Networks, ACM MPAC, 2006.

[16] S. Ayyasamy and S. N. Sivanandam, Trust Based Content Distribution for Peer-to-Peer Overlay Network in International Journal of Network Security & Its Applications (IJNSA), Volume 2, Number 2, April 2010.

[17] G. H. Nguyen, P. Chatalic, and M. C. Rousset, A Probabilistic Trust Model for Semantic Peer to Peer Systems, *DAMAP '08,* March 2008.

[18] OASIS xEtensible Access Control Markup Language (XACML) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml, [retrieved: December, 2013].

 ISBN: 978-1-61208-324-7