Trust Cell: Towards the End-to-End Trust in Data-Oriented Scientific Computing

Sangmi Lee Pallickara, Beth Plale, Liang Fang, Dennis Gannon
Computer Science Department, Indiana University
{leesangm, plale, lafang, gannon}@cs.indiana.edu

***Abstract*[1]— Data-driven computational science on community computational resources is frequently of a magnitude and scale that it requires that computations be done remotely, generating resulting data collections that are too large to be shipped back to a user's workstation. Service-oriented middleware is well equipped to carry out actions on behalf of a user, but SOA middleware does not address user trust in the privacy of their actions and security of their data. In this paper we first identify the role of trust in a large distributed computation then develop a model that represents the trust relationship between the users and their remote resources in the Grid system. We show how one can construct a trusted relationship from the model, with an emphasis on the importance of context to a specific trust relationship. We provide a case study of a data-driven scientific application that executed across multiple organizations.**

***Index Terms*— Data-Oriented Grid Computing, LEAD, Grid Security, Grid Trust**

I. INTRODUCTION

Trust is a catalyst for human activity. From our everyday life to professional activities, trust allows people to interact spontaneously and enables voluntary participation in group activities. In computational and data-oriented grids, such as the Teragrid [1], a user may resist migrating his/her computations or data storage off a local resource to a remote resource because of an inherent distrust they may possess in intra-organizational resource sharing. Indeed, unless the intra-organization system is able to build trust in how it handles a user's computations and data products, the system can expect reluctance on the part of scientific users to adapt the system as their experimental environments.

Current computational and data-oriented grids pose multiple challenges to building user trust. First, each organization administers its own resources by means of an in-house security mechanism. Ideally, the user can distribute computation and data products across organizational boundaries by means of resource communications based on universal protocols. However, since the resources have different schemes for managing the user's identity, the system should provide interoperable interfaces between different schemes. The implication for building trust is that the trust relationship is posed over different security schemes. Second, resources are often utilized by multiple grid applications, and trust must be understood based on the context [2] in which it is deployed. Popular resources can be accessed by multiple applications from various scientific fields, at different times and for different durations. So a clear definition of the context and management is required. Third, computational resources such as services are often implemented as composed or nested services. While interoperable service interfaces offer access to the resources, the interface to the resource is simply the interface, and users or other services cannot be expected to simply trust the communication and management between the distributed resources behind the service interface. An end-to-end trusted relationship is needed.

Trust management systems such as KeyNote[3], PolicyMaker[4], Simple Public Key Infrastructure (SPKI)[5], and Simple Distributed Security Infrastructure (SDSI)[6] attempt to manage security in large-scale distributed networks through the use of credentials that delegate permissions. However, these approaches investigate trust management between the service interfaces, without consideration for the resources these service interfaces themselves access.

In this paper we start from a fundamental definition of trust, and its characterization in a large distributed computation. We describe our new model for representing the trust relationships between users and remote resources in the Grid system. Specifically, we extend the trust relationship of discoverable service interfaces to those resources that are accessed locally and located behind the discoverable service interfaces. The model is general and adaptable so that existing or emerging security schemes can be fit into the model and provide end-to-end trustworthiness to their Grid applications. Further, we show how one can construct a trusted relationship from the model, with an emphasis on the importance of context to a specific trust relationship. In the service layer of Grid application, the context should be clearly defined and verified for each of communication between the minimum trusted units, *Trust Cells*. Finally, we give a case study of a data-driven scientific application that executes across multiple organizations. In this case study, we show how the Trust Cell is defined, and show how the trusted relationship within the

[1] This work is funded by the National Science Foundation under Cooperative Agreements ATM-0331480 and CNS-0202048.

Trust Cell is ensured. The example also demonstrates that some of the Trust Cells are actually administrated by different organizations and managed with different security schemes. We also present a scheme for managing the trusted access based on context-specific situations in our system.

The structure of the remainder of the paper is as follows. Section 2 is an examination of trust concepts and characteristics. Section 3 specifies the trust challenges in Grid computing environments. In Section 4, we introduce the trust model which considers the end-to-end trusted relationship in Grid computing. In Section 5, we present our case study. Section 6 discusses the ability of Trust Cell model to model a new capability security scheme. Section 7 contains related work and Section 8 outlines open problems in trust management.

II. TRUST

Trust is a social phenomenon. Before computer scientists began to investigate the issue, trust was a topic of research in such academic fields as psychology [7], sociology [8, 9]. These approaches have influenced efforts in the computer science community to generalize and formalize the concept of trust [10]. In this section, we provide the definition of trust most instructive for our context, and outline the characteristics of trusting actions.

A. Definitions of Trust

A widely accepted definition of trust comes from psychology, given by Deutsh [7] in 1962:

- *"If an individual is confronted with an ambiguous path, this path can lead to an event that can be perceived to be beneficial (Va^+) or to an event that is perceived to be harmful (Va^-);*
- *He perceives that the occurrence of (Va^+) or (Va^-) is contingent on the behavior of another person; and*
- *He perceives the strength of (Va^-) to be greater than the strength of (Va^+).*
- *If he chooses to take an ambiguous path with such properties, I shall say he makes a trusting choice; if he chooses not to take the path, he makes a distrustful choice."*

Deutsh's definition implies that trust is a subjective quality which individuals place upon one another. The definition also implies that trusting decisions are based on a form of cost-benefit analysis. Different individuals' decisions to trust differ with each individual's perception of the estimated cost ($Va+$) and ($Va-$).

Later, Gambetta [11] integrates various approaches from diverse fields such as biology and economics to define trust as follows:

"[trust is] a particular level of the subjective probability with which an agent will perform a particular action, both before he can monitor such action and in a context in which it affects his own action"

In addition to the subjective nature indicated in Deutsh's definition, Gambetta's definition implies that the level of trust depends on how our own actions are in turn affected by the agent's action.

B. Trust Characteristics

From the diverse set of properties identified in the psychology and sociology literature, there have emerged specific properties of trust identified as most relevant to software systems [12, 13]. These are subjectivity, context, non-transitivity, ability to be measured, and dynamic reasoning.

The subjective nature of trust is one of the most challenging properties of building a trust relationship, because the parameters used in each trust-related process varies widely among different individuals.

Trust is also context-specific. The level of expectation of trustworthiness varies according to the purpose of the system. For instance, when we say that Alice trusts dentist Bob, it does not mean that Alice trusts Bob for her financial decision as well. Bob is trusted in the context of that Alice needs a decision of her dental treatments.

Trust is not implicitly transitive. If Alice trusts Bob and Bob trusts Cathy, it does not necessarily follow that Alice must trust Cathy by any degree. In a large distributed system, this non-transitivity is often approached by the *recommendation operation*. It is processed under the conditions that the recommendation and the recommenders are trustworthy. Also it requires the ability to judge the quality of recommender.

The value of trust can be measured. For example the trust model proposed in [12] provides four level of trust degrees such as 'very trustworthy' or 'untrustworthy' based on their measurement of trust value. Finally, trust reasoning is dynamic and non-monotonic. Additional evidence or experience at a later time may increase or decrease our degree of trust in another agent.

III. TRUST CHALLENGES IN DISTRIBUTED COMPUTING

Distributed computing in general, and grid computing specifically, facilitates coordinated resource sharing and problem solving in dynamic, multi-organizational Virtual Organizations [14]. Sharing, which includes giving direct access to computing units, software, data collections, and other resources to users in other organizations, requires that a resource provider have complete control of the resource, and make decisions based on consumer decisions about what is shared, who is allowed to access, and under what conditions sharing is to occur.

Data-specific scientific resources, such as collections of data products, are understood as intellectual properties that require exclusive access. For example, in the data-driven application, the resultant data (or analytical data) of critical experiments can be the supporting data behind a significant new contribution to science or profitable discovery. As grid computing is increasingly seen as a viable resource for larger

and more complex computational science investigations, it simultaneously introduces critical concerns of privacy and confidentiality. To make grid computing more palatable to broader groups of users, trustworthiness, which encompasses privacy and confidentiality, must be addressed.

To understand the trust challenges in grid computing, we start from Azzedin's classification of trust in the context of the Grid computing [15]: identity trust and behavior trust. *Identity trust* is concerned with verifying the authenticity of an entity and determining the authorizations that the entity is entitled to access and is based on techniques including encryption, data hiding, digital signatures, authentication protocols, and access control methods. On the other hand, *behavior-trust* deals with a wider notion of an entity's trustworthiness. For example, a digitally signed certificate does not convey whether or not the issuer is an industrial spy, and a digitally signed code does not convey whether or not the code is written by competent programmers [15].

Identity trust has been actively investigated in the grid community. Public key cryptography (PKI) based Grid Security Infrastructure (GSI) [16] from the Globus Toolkit is widely used. GSI establishes the identity of users or services, protecting communications; and authorization of the user, as well as managing user credentials and maintaining group membership information. However, there remain issues such as key management and identity from different Certificate Authorities. Also, resources often have their own application layer policy and security mechanism. GSI does not ensure end-to-end trusted access (from the end-user to the end-resources) by itself. For example, Storage Repository Broker (SRB) [17] provides single point authentication to the users. To access distributed SRB networks across multiple organizations, the users must authenticate themselves to the first SRB that they access. Later, the first SRB authenticates to the second SRB with the representative identity, "SRB" instead of the user's identity. From this moment, the second and subsequent SRBs process the authentication with the representative identity that is agreed between their participants. The user's identity is transferred but processed only when it is required. This approach provides efficient authentication handling over distributed services, located in the different organizations within their context, based on the trusted participants. To provide trusted access to the end resource (files or streaming data), a Grid service must provide the end-to-end trust relationship facilitated by the service-level security and the application-level control scheme.

There are various approaches to behavioral trust in Grid activities. [18, 19] These approaches are more focused on trust between services. As the Grid is deployed as a new paradigm of scientific collaboration not only physically and technically, but also socially the Grid system should also be able to adapt to the social factors of the community members. For instance, a wider approach comprising the trust relationship between the user and services, or the user and collaborators should be followed.

IV. The End-To-End Trust Model

As a means to capture and formalize end-to-end trust, we constructed a trust model that extends the inter-service security infrastructure to encompass end-to-end trustworthiness between users and physical resources. It comprises modular trust domains across large distributed grid computing environments and trusted referrals between the global and local resource managements.

A. Trust Cells and Referring Services: Components of the Model

The model we propose builds on the notion of the *Trust Cell*, a minimum domain (or collection) of resources that is trusted and recommended by one or more discoverable services within the domain. Here we define the "recommendation" as providing enough information about the security and privacy management of the local resources. Therefore, if the application contains resources whose trust is agreed upon by the participating services of the application, the process of recommendation will be done during the time which agreements between the participants are being reached. For instance, in the initial stage of an application run, the participants will decide whether they will trust a particular resource for use in their application or not. If the application somehow permits access to resources of uncertain trustworthiness, the recommendation process should be much more dynamic, such as providing a trust measurements or leveraging with a centralized trust management system.

A trust cell has a *Referring Service* that is contained within the Trust Cell and accessible from the higher level service layer. An organization provides one or more Trust Cells and a Trust Cell can contain services from one or more organizations. Because the size of an organization can vary and since some active organizations are involved in multiple applications, the Trust Cell does not have to be mapped exactly to the organization. However if multiple organizations provide resources to one Trust Cell, the security policies of the organizations should be compatible. The trust relationship within the Trust Cell must be recommended by a globally trusted referral service. The referral service must be able to revoke communications with the services within the Trust Cell.

B. Trust Cell Model

The model is depicted graphically in Figure 1. There are multiple depths of visibility in Grid resources. At the top layer, the "Virtual Organization" layer, there are service interfaces that are discoverable by members of the VO. Participating organizations provide one or more virtual hosts to the VO. Virtualization of resources is achieved through WSDL service interface definitions. WSDL definitions are distinct from the protocol bindings used for service invocation. For users and other services, these interfaces are the only ones available to specific resources.

In the second layer, the "Services" layer, we capture the more detailed interactions that must be carried out between VO-level services and the locally available services that

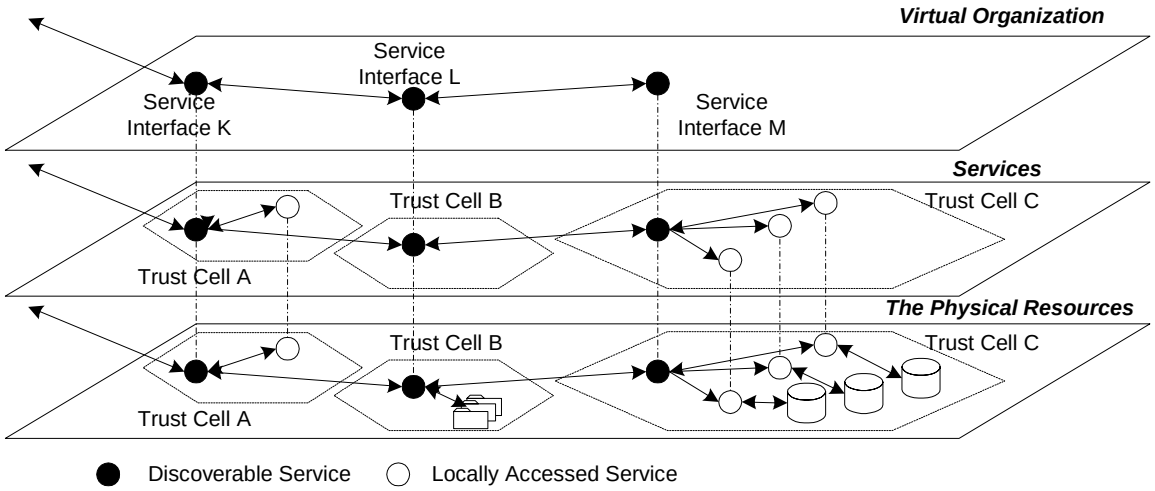

Figure 1. End-to-end Trust Model. Service interfaces at the virtual organization level implement functionality through local services and resources. The trust relationship propagates from trust cells up to the service interface.

implement that higher-level service. A VO-level service interface might be very bound directly to a computing resource; hence there is no communication with additional services or computing resources. This is depicted for "Service Interface L". On the other hand, VO-level services might interact with numerous local services which in turn must access multiple physical-level resources. The bottom layer captures service communication with physical resources. The service interface M interacts with a distributed database for instance. Likewise, service interface L interacts with a file system.

Therefore although the VO-level service interfaces provide an access point to the resources, they cannot fully represent the end-resources which are directly related to a user's privacy or confidentiality.

C. Trusted Referrals in Trust Cells

The role of the referral service in the Trust Cell model is important. The referral service recommends the local services administrated by the same trust cell. To be a trusted referral service, the referral service must be,

- A local and global trusted delegator in the predefined context.
- Able to verify the competence of a requester in order to delegate him/her to work in the context.
- Contains revocation power over the released delegation certificates.

To satisfy the above requirements, the referral service should leverage the local security scheme and cross domain security infrastructure. For instance, the distributed system should provide the management of the predefined context. It can be a policy management or capability management in the current grid architecture. Likewise, the distributed system should provide identity and authentication management to verify the competence of the requester.

Meanwhile, for the local resources, the referral services must be able to verify the trustworthiness of access to the local resources. It can be a local security scheme such as X.509 [20] based authentication, local capability manager, or a local reputation system. The referral service should be able to revoke communications within the trust cell if the activity is not trusted enough.

D. Authenticity Handover and Delegation

For communications between the trust cells, the trust cell model follows the security infrastructure agreed by the Grid community, such as the GSI infrastructure [16]. If the system follows GSI, the X.509 based certificate is required. The GSI provides a delegation capability. Therefore, if a Grid application requires that several Grid resources be used or agents requesting services on behalf of a user, the need to re-enter the user's pass phrase can be avoided by creating a proxy. When proxies are used, the remote party receives the owner's certificate with the proxy certificate. During mutual authentication, the owner's public key is obtained from the owner's certificate and used to validate the signature of the proxy certificate. In addition, the owner's certificate is validated by the Certificate Authority's (CA) public key. This process is repeated during the delegation and establishes a chain of trust from the CA to the proxy through the owner.

The delegation scheme of GSI provides a trust chain to the Grid application. However, especially for the large-scale Grid application which interacts with a large number of Grid resources and maintains large number of users, applying GSI-style delegation to each of the Grid resources and users is not practically straightforward. Therefore, some Grid applications will apply GSI-style delegation until a particular point during the application flow at which time it will hand over the user's authenticity to more generalized authenticity, such as the service provider's identity. A Grid application agrees to use a certain set of service provider's certificates initially, and

participants accept those certificate without a delegation process.

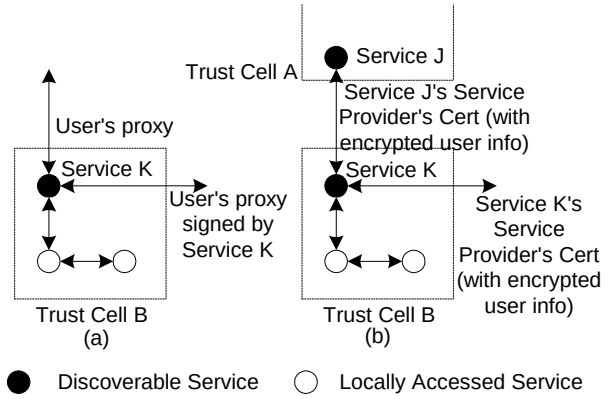

Figure 2 Communication between Trust Cells. In (a), the trust cell B delegates the user's certificate. In (b), the trust cell B hands over the user's identity from Trust Cell A to the next Trust Cell.

Figure 2 shows how the Trust Cell model applies in both the aforementioned cases. Figure 2-(a) presents the GSI style delegation between Trust Cells. User's proxy is signed by the referral service interface and passed to the next Trust Cell. The referral service of the next Trust Cell verifies the user's delegation with the proxy and the CA's public key. The trust chain between the Trust Cells is built by the delegation process of the GSI.

Figure 2-(b) depicts how the trust cells communicate with the service provider's certificate. Trust Cell A and B can have the same or different service certificates, but those certificates must be agreed within the Grid application. This type of communication is efficient in the sense of managing the keys and the user's certificate. However, the service provider's certificate is not enough to verify the competence of a requester which is one of the requirements of functionalities of the referral services. To delegate the user's behavior, each of the access must be under the specific context that defined and agreed by the Grid application. Therefore, to apply the Trust Cell model, if the Grid application wants to utilize the service provider's certificate for their practical purpose, it must define the context specific authorizations for each activity.

In addition, the user's privacy related information needs to be secured. The referral service must have the revocation power in local and global. Even though the Trust Cells are authenticating with the service provider's certificate, if the communication is not secure enough, the referral service is able to revoke the communication. Therefore, the user's identity or privacy related information should be properly secured when it is delivered between the Trust Cells. For example, Figure 2-(b) shows Trust Cell B passes the user's identity to the next Trust Cell. It encrypts the user's identity with its private key before sending the request. By doing this, although the communication between the Trust Cells is authenticated by the service provider's identity, user's private information is secured until it is utilized later.

V. Case Study: Application of Trust Cell Model to LEAD Data Subsystem

Linked Environments for Atmospheric Discovery (LEAD) is a large NSF funded Information Technology Research (ITR) project [21]. LEAD is a multidisciplinary effort involving nine institutions and more than 100 scientists, students, and technical staffs in meteorology, computer science, social science, and education. LEAD addresses the fundamental research challenges needed to create an integrated, scalable framework for adaptively analyzing and predicting the atmosphere.

LEAD's foundation is dynamic workflow orchestration and data management in a Web service framework. These capabilities provide for the use of analysis tools, forecast models, and data repositories, not in fixed configurations or as static recipients of data but rather as dynamically adaptive, on-demand systems that respond to weather as it evolves. Although mesoscale meteorology is the particular problem to which we've applied the LEAD concept, the methodologies and infrastructures we've developed are extensible to other domains such as medicine, ecology, oceanography, and biology.

A. LEAD Data Subsystem

The LEAD data subsystem allocates storage space on the LEAD grid to an individual user for purposes of storing the data products resulting from the meteorological investigations they carry out on the LEAD grid and the Teragrid [1].This space is cast as a user's *personal workspace* and is managed through a service called myLEAD [22]. In addition to the myLEAD service, the LEAD data subsystem comprises a Query service, a Storage Repository service, an Ontology Service, and a portal service providing a client with web access to the subsystem. As shown in Figure 3, these services reside all over the site of LEAD grid. Each site and resource follows a specific in-house security scheme.

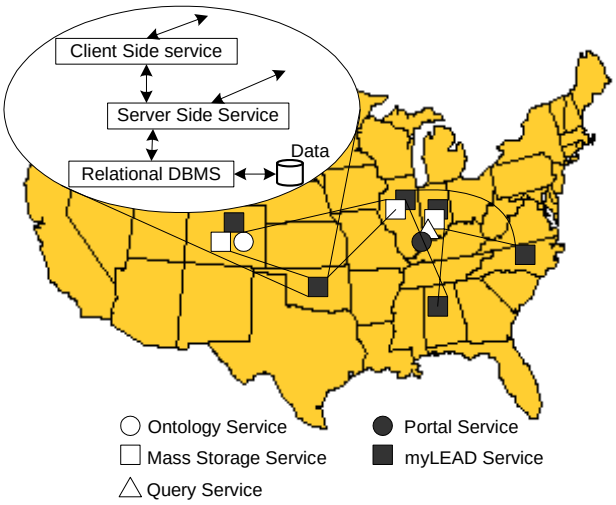

Figure 3 LEAD Data Subsystem deployed on the LEAD grid

One of the key components of the personal workspace is a metadata catalog that stores the metadata associated with data products generated and used in the course of scientific investigation. The data products, many of which are hundreds of megabytes to a gigabyte in size and in a binary format, reside either in the database along with the metadata or in a separate storage repository. A strength of the workspace manager is that it organizes and tracks the critical information about a workflow computation, including a users' preferred computer servers, scripts and input files, associated documentation, the generated data products' provenance, and run status.

The manager of the user's workspace is a implemented as multiple distributed instances of the myLEAD server, where a service instance resides at each site in the LEAD testbed. Each of the sites in the LEAD testbed will run a persistent server-side service, and client-side service, which are shown in the oval of Figure 3. Portal access is through a single portal. Users local to a site will have their "personal metadata catalog" managed by the myLEAD service at that site. User workspaces are distributed (partitioned) across the sites. One of the sites is elected to run a master instance of the myLEAD service. This master instance serves as a replica to all the satellite sites. Replication synchronization with the master occurs on a schedule; the current plan is for nightly updates.

The server-side service instance is a persistent Grid service built on top of a relational database. It extends the Globus Toolkit MCS [23] and OGSA-DAI[24] with performance optimizations, support for complex attributes, publishing and sharing, third party administration, and versioning of experiments.

B. Mapping Case Study to Trust Cell Model

In this section, we present a mapping case of the Trust Cell Model especially for the myLEAD system which is the most highly distributed service within the LEAD data subsystem. The myLEAD systems are deployed in the major testbeds in the LEAD system. Each of the myLEAD system forms a Trust Cell as shown in Figure 4. The MyLEAD Trust Cell contains two referral services, Client-side service and Server-side service. Both services are accessible by external users or services with proper authentication. Inside the myLEAD Trust Cell, each constituent component follows the myLEAD security scheme. First, myLEAD requires designated deployment of the subsystems. As we depict inside Figure 4, myLEAD comprises multiple local services and resources. Here, the database system of myLEAD is designed to communicate with only the myLEAD system. Access from other requesters is ignored. Second, the referral services propagate the user's trust only in the pre-specified context by means of leveraging with the LEAD security scheme based on its capability management. Any other attempts are ignored. The capability management will be discussed in the following section in detail. Third, the myLEAD referral services manage fine-grained authorization. For example, the Client-side service, which provides user's personalized activities during the investigation, limits the user's accessibility to the functionality by his/her role. If there is any other attempt, it revokes the request.

MyLEAD does not allow accesses to the unknown resources. Based on above security scheme, the LEAD community agrees on the trustworthiness of the myLEAD system. Therefore, there is no requirement for the dynamic recommendation to the referral service such as trust measurements or scoring. The myLEAD Trust Cell provides sufficient trustworthiness inside of the Trust Cell for the LEAD grid and it communicates with other services and Trust Cells by means of the LEAD security scheme, which provides context specific authentication and authorizations. Therefore, the myLEAD Trust Cell successfully propagates a user's trust to the physical resources.

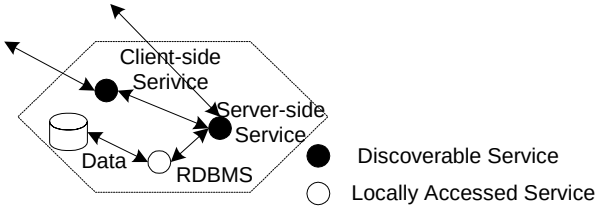

Figure 4 Applying Trust Cell Model to myLEAD

VI. ABILITY OF TRUST CELL MODEL TO MODEL ALTERNATE SECURITY SCHEME

The Capability Manager (hereafter Capman) [25] implements a capability-based access control model in the domain of Web services in the Grid system. Capman is the core service supporting XPOLA [25], the fine-grained authorization scheme. XPOLA a notion of two roles among the Grid users: *service providers*, and *service users*. Service providers start and maintain the services under their own account provided by the Grid application. Service providers make their services available to other Service users by issuing the "capabilities" of their services. The "capability" is a delegated right from the service provider, for a group of designated service users, on accessing specific operations of an appointed service instance. A "capability" contains information described in XPOLA's capability policy definition.

Capman manages the service level access control in the LEAD grid. The features of Capman leverages with the Trust cell model seamlessly. One of the major concerns of propagating the trust relationship is whether the context is clearly defined for each relationship. Capman provide a fine-grained access control in a specific context for each access to the service instances by means of the capability policy definition. For instance, the "capability" contains information about the accessibility which is defined as follows

$$C = [X, P, R, S, T + \Delta t]$$

where resource provider X creates a capability C to delegate a set of rights R on the service identified as S to a group of users P after the time T for the time duration in the future of Δ t.

The information in the capability is sufficient to specify the context which propagates the trust relationship between the resource provider and the user in the LEAD grid. Therefore the new security scheme, Capman, performs its role in Trust Cell model and cooperates with other trust properties required in Trust Cell model.

VII. Related works

The approaches of the Trust models in the Grid computing are more concentrated on identity trust such as those based on the credential. There have been Trust managements systems [3, 4, 5, 6] which focus on identity trust in large distributed systems. PolicyMaker [4] is a trust management system that facilitates privacy and authenticity for various network applications. The PolicyMaker service determines whether the request is acceptable based on the policy which is defined in the *PolicyMaker Language*. The policy contains local policy statements, a collection of credentials, and proposed trusted actions. KeyNote [3] is also identity trust management systems. KeyNote is based on the same principles as PolicyMaker except its trust engine verifies the signatures as well. These credential-based trust management systems are based on the assumption that service providers and their services are fully trusted. However, the resource requesters are not trusted and have to verify their authenticity each time. Furthermore, the credential-based trust management does not consider the trusted access to the resources that reside behind the access point. In our approach, we provide a conceptual model of the trust relationship which is propagated to the resources accessed by the service interfaces, so that a user can achieve an end-to-end trustworthy Grid application.

On the other hand, there have been models for supporting behavior trust based on experience and reputation such as [12]. This trust-based model allows entities to decide which other entities are trustworthy and also allows entities to tune their understanding of another entity's recommendation. This issue is more actively investigated in the decentralized network systems. Reputation-based trust management systems establish trust relationships with other peers and assign trust values to these relationships. The XREP[26] approach concentrates on P2P file-sharing applications. Each peer evaluates resources accessed from peers and a distributed polling algorithm is used to allow these reputation values to be shared among peers. The P-Grid trust management focuses on an efficient data management technique to provide scalable trust model for decentralized applications [27]. To achieve the scalability, P-Grid divides the problem of decentralized trust management into three generic sub-problems. First P-Grid defines a global trust model that determines whether a peer can be trusted or not. Second, P-Grid determines the local efficient computation that each peer needs to execute in order to approximately determine the trust in another peer. The last sub-problem is to consider the effect of this local trust algorithm on the actions of malicious peers. Therefore, trust computed by using P-Grid is only based on the "complaints" about the malicious peers. These approaches have contributed in the mechanism of the measuring the trust and developing the algorithm to calculate and share the values between peers. Although these approaches do not consider the end-to-end trust management, we consider these approaches will be able to leverage with Trust Cell model so that applications can ensure the end-to-end trustworthiness in the open grid environments.

Another interesting approach is social networks-based trust systems. [28] was an early investigation about the effect of social relationships of peers belonging to an online community on reputation in decentralized scenarios. It models an electronic community as a social network. When a peer determines the reputation of other peer, the value of the trust is calculated based on the direct reputation from the accessing peer and its neighbor's reputation of targeted peer. The Regret [29] is similar in concept to [28]. Regret specifies three dimensions of reputation: individual, social, and ontological. It combines three different dimensions to calculate a single value of reputation. As we consider the Grid society is a group of resources which closely collaborate with each other, the social networks-based trust system is able to be applied. Similar to other approaches, to apply to the grid environments, these concepts should be adjusted to the characteristics of the grid computing.

VIII. Conclusion and Open Problems

The data-centric scientific computing is facing to more and more sophisticated requirements such as more dynamic and flexible data management along with its continuous increasing data product sizes. In this paper, we presented a trust model, Trust Cell model, which provides end-to-end trustworthiness to the scientific application. The Trust Cell model extends the inter-service security infrastructure to the end-to-end trusted relationship between the user and the physical resources. It comprises modular trust domain across large-scale distributed computing environments and trusted referrals between the global and local resource managements.

We also provide a case study of a data-driven scientific application that executes across multiple organizations. In this case study, we presented how the Trust Cell is defined and the trust relationship within the Trust Cell is ensured. In addition, we also showed how the Trust Cells propagates the trusted relationship within a specific context.

The end-to-end trust approach within the Grid environment exposes open problems, such as how we measure trustworthiness, and related issues of collusions to inflate trust scores. We intend to investigate these issues in the future.

References

[1] Andrews, P., T. Sherwin, and B. Banister "A Centralized Data Access Model for Grid Computing," in *Proceedings* of IEEE Symposium on Mass Storage Systems, April 2003

[2] Patrick, A. S., "Building Trustworthy Software Agents," *IEEE Internet Computing*, 2002, Vol.6, No.6, pp.46~53

[3] Blaze, M.,"Using the KeyNote trust management system," Technical Report, AT&T Research Labs, 1999

[4] Blaze, M., J. Feigenbaum, and J. Lacy,"Decentralized trust management," in *Proceedings* of IEEE Conference on Security and Provacy, 1996

[5] Ellison, C. et al.," SPKI Certificate Theory," RFC2693, Network Working Group, 1999

[6] Rivest, R., and B. Lampson,"SDSI—A Simple Distributed Security Infrastructure Version 1.1," presented at CRYPTO '96 Rumpsession, 1996

[7] Deutsh, M., "Cooperation and Trust: Some theoretical notes," in *Nebraska Symposium on Motivation*, M.R. Jones, Ed. Nebraska University Press, 1962

[8] Luhmann, N., Trust and Power. Wiley, 1979

[9] Barber, B, Logic and Limits of Trust. New Jersey: Rutgers University Press, 1983

[10] S. Marsh, "Formalizing Trust as a Computer Concept," Ph.D. dissertation, University of Stirling, Department of Computer Science and Mathematics, 1994

[11] Gambetta, D., Can We Trust Trust? Dept. of Sociology, University of Oxford, 2000 ch.13, pp. 213-237

[12] Abdul-Rahman, A., and S. Hailes, "Supporting Trust in Virtual Communities", in *Proceedings Hawaii International Conference on System Science* 33, 2000

[13] Christianson, B. and W. Harbison, "Why Isn't Trust Transitive?," in *Security Protocols Workshop*, 1996, pp.171-176

[14] Foster, I., C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," International Journal of Supercomputer Applications, 15(3), 2001

[15] Azzedin, F. and M. Maheswaran, "Evolving and Managing Trust in Grid Computing Systems," IEEE Canadian Conference on Electrical & Computer Engineering (CCECE '02), May 2002

[16] Welch, V., F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, S. Tuecke,"Security for Grid Services," in Proceedings of Twelfth International Symposium on High Performance Distributed Computing (HPDC-12), IEEE Press, to appear June 2003.

[17] Rajasekar, A. et al, "MySRB and SRB—Components of a Data Grid," *Proc.* 11th IEEE High Performance Distributed Computing (HPDC), IEEE CS Press, 2002, pp.301-310

[18] Alunkal, B., I. Veljkovic, G. Laszewski, K. Amin,"Reputation—Based Grid Resource Selection," Workshop on Adaptive Grid Middleware (AGridM 2003), September 28, 2003, New Orleans LA

[19] Groth, P., M. Luck, and L. Moreau,"A Protocol for recording provenance in service-oriented Grids," In Proceedings of the 8th International Conference on Principles of Distributed Systems (OPODIS'04), Grenoble, France, December, 2004

[20] Adams, C., and S. Farral, "RFC2510 – Internet X.509 public key infrastructure certificate management protocols," RFC2510, 1999

[21] Droegemeier, K. et al., "Service-oriented environments for dynamically interacting with mesoscale weather", Computing in Science and Engineering, IEEE Computer Society Press and American Institute of Physics, Vol. 7, No. 6, pp. 12-29, 2005

[22] Plale, B. et al., "Active Management of Scientific Data," IEEE Internet Computing special issue on Internet Access to Scientific Data, Vol. 9, No. 1, Jan/Feb 2005, pp. 27-34

[23] Singh, G., et al., "A Metadata Catalog Service for Data Intensive Applications," Proc. ACM/IEEE Supercomputing 2004, IEEE CS Press, 2003, pp. 33-49

[24] Antonioletti, M. et al., "Experiences of Designing and Implementing Grid Database Services in the OGSA-DAI Project," Proc. Global Grid Forum Workshop on Designing and Building Grid Services, Global Grid Forun, 2003

[25] Fang, L. et al., "XPOLA: An Extensible Capability-based Authorization Infrastructure for Grids," Proc. the 4th Annual PKI R&D Workshop, April 2005

[26] Damiani, E., et al. "Reputation-based Approach for Choosing Reliable Resources in Peer-to=Peer Networks," in *Proc*. ACM Conference on Computer and Communications Security, Washington DC, 2002

[27] Aberer, K., "P-Grid: A self-organizing access structure for P2P information systems.," in Proc. 9th International Conference on Cooperative Information Systems, Trento, Italy, 2001

[28] Yu, B., and M.P. Singh, " A social mechanism of reputation management in electronic communities.," In Proc. 4th International Workshop on Cooperative Information Agents, 2000

[29] Sabater, J., and C. Sierra, " Reputation and social network analysis in multi-agent systems.," in Proc. First International Joint Conference on Autonomous Agents and Multi-Agent Systems, 2002, Bologna, Italy