International Journal of Computer Applications (0975 - 8887)
Volume 133 - No.12, January 2016

A Survey on Trust Aware Security and Scheduling in Traditional and Ad Hoc Grids

Slavomír Kavecký
Department of Informatics
Faculty of Management Science and Informatics
University of Žilina
Slovakia

Penka Martincová
Department of Informatics
Faculty of Management Science and Informatics
University of Žilina
Slovakia

ABSTRACT

The grid infrastructure has emerged as a technology supporting long-term collaboration utilizing shared resources spread across multiple administrative domains. However, for a transient community that is interested only in short-term or one-time collaborations the establishment of a traditional grid environment can become too cumbersome. The ad hoc grid infrastructure was proposed as a consequence of the need for the short-term collaborations. The ad hoc grid infrastructure provides the same basic grid services as the traditional grid, but the implementation of these services may vary from the implementation in the traditional grid. The paper focuses mainly on the similarities and differences in the trust awareness of grid security and job scheduling in the traditional and ad hoc grid environments.

General Terms

Traditional and Ad Hoc Grid Security, Traditional and Ad Hoc Grid Scheduling

Keywords

Traditional grid, ad hoc grid, trust aware grid security, trust aware job scheduling

1. INTRODUCTION

Recently, large data processing and high-performance computing has become more available for the public. Grid[11], which is one of the leading technologies enabling these capabilities, is characterized by heterogeneity and geographical dispersion of its nodes serving as resources for job execution or as access points into the grid environment. According to the OGSA (Open Grid Services Architecture)[10] the following capabilities should be typical for any grid middleware: user tasks execution management, data manipulation management, shared resources allocation and management, secure job execution and resource sharing, information provision of executed tasks and shared resources, and finally support for the grid configuration.
As stated above, job scheduling and secure execution of jobs are core services enabling provision of the main grid capabilities – sharing and utilization of available dispersed resources. The traditional grid infrastructures (the most known traditional grid infrastructures are Globus Toolkit [16], Gridbus Middleware [13] and UNICORE [36]) implement scheduling and security in accordance to their centralized nature. In the ad hoc grids (the most known ad hoc grid infrastructures are OurGrid [4, 35] and MoGrid [14]), which are well known for structural independence and decentralized architecture, scheduling and security are managed by participating nodes without depending on any external infrastructure for assistance.
The aim of the paper is to present information about the main characteristics of traditional and ad hoc grid architectures, the similarities and differences between them, as well as to describe the fundamental features of security and scheduling services provided by both grid infrastructures extended with the trust management.
The reminder of the paper is organized as follows: The section 2 provides a description of traditional and ad hoc grid and lists the main differences between the two technologies; Section 3 overviews the implementation of security in the grid infrastructures; Section 4 describes the traditional and ad hoc grid scheduling process; Section 5 introduces a description of future work in the field of trust aware grid security and scheduling and Section 6 concludes the paper.

2. DESCRIPTION OF TRADITIONAL AND AD HOC GRID

The aim of the grid systems and applications is to integrate and manage resources and services within distributed, heterogeneous and dynamic virtual organizations. The execution of this goal requires the disintegration of barriers usually separating different computing systems within and across organizations, so that computers, services, data and other resources can be accessed regardless of physical location[10].

2.1 Traditional grid

The grid technology evolved from the already existing technologies as distributed computing, virtualization, web services, the Internet and various cryptography technologies. These technologies existed for some time and served for various purposes. The grid has taken features from these technologies to create a system providing computational resources for high-throughput computing.
Virtualization is one of the basic characteristic typical for any grid system and refers to integration of geographically dispersed and

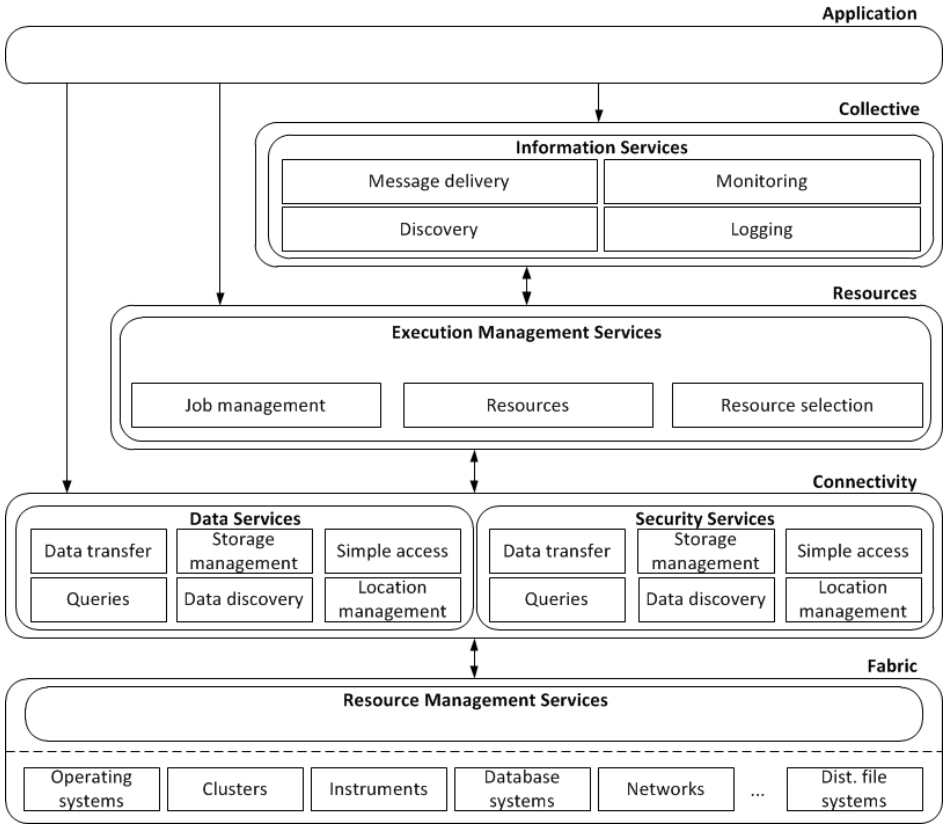

Fig. 1. Traditional grid architecture[11, 10]

heterogeneous systems. This enables the users to abstract from the underlying infrastructure (they do not need to be aware of the location of the resources, access protocols, etc.) and access the distributed systems through one access point. From the user's perspective, there is just one computational system, to which he needs to submit his service request. Then it is up to the grid system to discover and locate the appropriate grid resources, which can process the user's request.

The access to the distributed resources and services, to which the user submits requests via his access point, is managed by the grid middleware. The middleware combines services that enable access to the underlying resources, data access, data manipulation, security provision, job scheduling and job execution. These services are supported by the information services that provide capabilities like resource discovery, resource usage monitoring, logging and more. The architecture of the grid system firstly proposed in the "Anatomy of the Grid"[11] separates the grid system capabilities into layers as shown in Fig. 1. Each layer encapsulates a set of services and functions utilized by the upper layer and uses services of the lower layer. The bottommost layer manages access to the resources, instruments and other dispersed entities integrated into the grid infrastructure.

The organizations forming the grid community typically have different resources in terms of hardware, software, operating system and network bandwidth. A group of organizations, which is referred to as a virtual organization, utilize the grid system as a tool for achieving a shared goal (execution of particular set of tasks, processing of unique data, creation of specific services provided by the grid system, etc.). The virtual organization can become interested in services provided by other grid solutions. To make the various grid systems interoperable; every grid middleware must provide its services in a standardized manner. The unofficial set of standards and recommendations summarized in OGSA[10] defines that the grid system services must be able to overcome the boundaries usually separating them. In contrast with the above mentioned architecture, the grid architecture proposed in the OGSA standard is not layered, but the services defined in the OGSA standard are capable of mutual interaction and creation of new capabilities. This allows the OGSA services to be presented in a layered manner as shown in Fig. 1.

The virtual organizations are dynamic and can be joined and left by any organization per its requirements and convenience. This dynamic nature of virtual organizations makes the grid system structurally decentralized. However, it is important to note that the services provided by the grid middleware are managed and provided in a centralized manner, which makes the traditional grid infrastructure centralized not structurally but functionally.

2.2 Ad hoc grid

Traditional grids are based on centralized architecture, in which activities as maintenance of resources, monitoring and access control enforcement are performed by a dedicated administrative authority. All participants of the grid community share a non-conflicting objective and collaborations are executed under the control of agreed policies on usage. However, there also exists a need

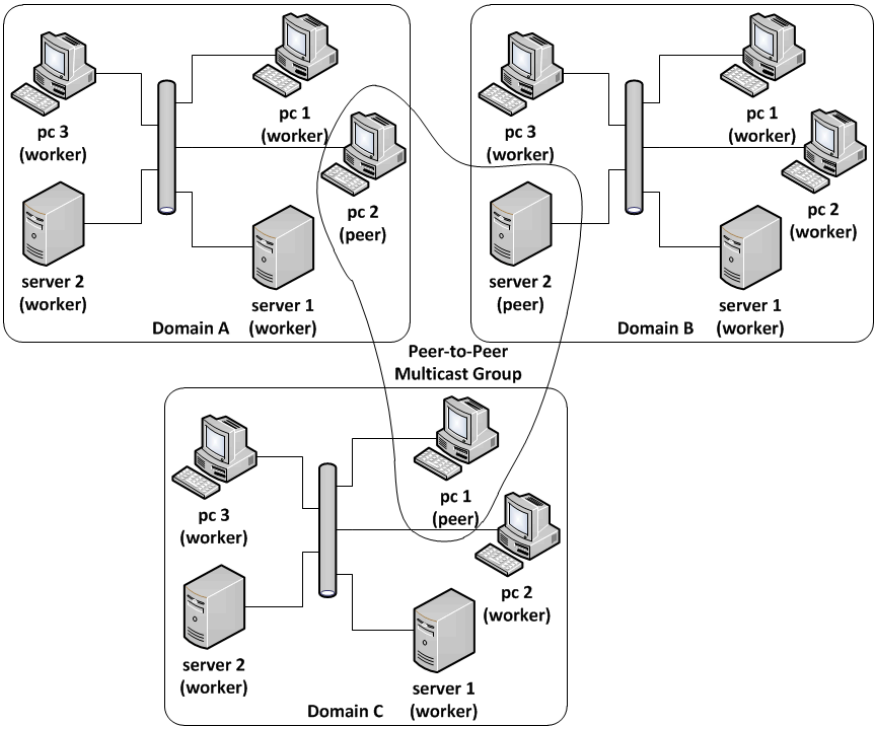

Fig. 2. Ad hoc grid topology[35]

for the support of sporadic and ad hoc communities and collaborations with dynamically changing members and access policies[3].
The motivation for ad hoc grid development is its ability to handle short-term collaborations. If a group of individuals need to pool resources and execute computation tasks in a one-time collaboration, then administrative overhead resulting from establishment of a traditional grid environment is not practical for such a transient community. In this scenario no individual can be entrusted with administrative privileges, but still all shared resources and provided services must be protected.
The ad hoc grid contains geographically dispersed resources with various management policies and in contrast with the traditional grid there is no centralized control. The **ad hoc grid** can be defined as: *distributed computing architecture offering structure-, technology-, and control independent grid solutions that support sporadic and ad hoc use modalities*[3]. The structural independence provides several benefits. It avoids a single point of failure that in the centralized architectures often leads to a failure of the whole system. It also enables participating nodes to start collaborations without depending on any external infrastructure for assistance. The technology independence in the ad hoc grid reflects the ability to support diverse grid technologies and protocols. The control independence allows the security to be managed in the absence of a central controller.
Ad hoc grid computing is not only a simple peer-to-peer communication and collaboration, but it emphasizes the integration of ad hoc paradigms into the grid domain. For example, this idea is implemented in the OurGrid toolkit[4, 35], which is focused on the sporadic ad hoc nature of the grid structure rather than the mobility of devices. In contrast with the traditional grid computing, there are no standards defined for the ad hoc grid architecture, thus the implementation of the grid services may vary. Nonetheless, in any ad hoc grid infrastructure implemented so far the nodes are capable to execute submitted tasks independently from any external services or infrastructures. For example, the ad hoc grid topology of the OurGrid toolkit depicted in Fig. 2 is one of the infrastructures, whose implementation is based on idea of the grid nodes independence.
The ad hoc grid is a relatively new technology. The existing ad hoc grid infrastructures have been developed with the aim to provide the basic grid capabilities as resource allocation, tasks management and information provision of executed tasks and shared resources. The security still remains an issue to be properly solved.

2.3 Comparison of traditional and ad hoc grid

The traditional grid is mainly used for high-performance computing and long-term collaborations. On the other hand, the ad hoc grid has just started to emerge in the beginning of the last decade. The basic ad hoc grid services as resource management, scheduling and job execution management were already implemented, but the ad hoc grid infrastructure is still lacking a full-fledged security implementation. This issue with security hinders a broader acceptance of the ad hoc grid technology.
In a traditional grid environment the scheduling process is responsible for assigning jobs, tasks and workflows on a set of available resources. The scheduling process receives the information about the availability and other characteristics of the resources through a centralized information service. Similarly, the grid security service handling the protection of job execution is supported by a trusted third party acting as an intermediary assuring that a relying party will not be harmed by malicious activity of a collaborating party. On the other hand, in an ad hoc grid environment the scheduling and security services are managed individually by each node. Fur-

Table 1. Comparison of the traditional and ad hoc grid infrastructure characteristics

Characteristic	Grid type	
	Traditional	Ad hoc
Grid architecture standards	Unofficial set of standards and recommendations summarized in the OGSA standard.	No standards defined for ad hoc grid architecture.
Grid organization	Structural decentralization achieved through dynamic nature of virtual organization and functional centralization resulting from provision of grid services in a centralized manner.	Full decentralization, services are provided in a decentralized manner and grid nodes are responsible for all their activities executed in the grid.
Type of shared resources	High performance computers (HPCs), measuring instruments and sensors.	Personal computers and mobile devices (but HPCs as well)
Goal of the grid community participants	Non-conflicting objective shared by all participants.	Participants of the grid community are interested in achieving only their own objectives.
Main usage of the grid	Long-term collaborations and high performance computing.	Short-term or one-time collaborations.
Support of fundamental grid functionalities	Nowadays grid infrastructures are capable to provide all basic grid services defined in the OGSA standard.	Support for job execution (i.e. resource allocation, scheduling, task management, information provision about resources and executing tasks) is handled by the ad hoc grid infrastructures, but a full-fledged implementation of other important services (e.g security of job execution) is still missing.

thermore, the scheduler must take into account that the nodes can join and leave the ad hoc grid community on the fly.
The traditional and ad hoc grid infrastructures undoubtedly share a large amount of similar characteristics regarding the execution of jobs on shared resources. However, they also differentiate in the architecture of provided services, type of collaborations executed through the infrastructures, type of shared resources, etc. The most significant differences of the both grid technologies are listed in the Table 1.

3. GRID SECURITY AND TRUST

The purpose of any grid security infrastructure is to protect shared resource from malicious actions of users and user's data from unauthorized access. The common security mechanisms utilized for the protection purposes are the processes of authentication and authorization. However, the security in the traditional and ad hoc grid infrastructures is implemented in a different manner in regard to the architectural aspects of both grid technologies. The subsections 3.1 - 3.3 explain this issue in more details.

3.1 Traditional grid security mechanisms

The first traditional grid infrastructures were used by a small group of users with unnamed trust relationships among them. When the community of users grew bigger, the need for secure access to resources, secure communication and data manipulation has become an important issue. The grid developers proposed and implemented several authentication and authorization infrastructures as a solution to this problem. The following subsections contain a brief survey of the most widely known and/or used infrastructures.

3.1.1 Authentication infrastructures. The process of authentication checks whether or not the identity of an entity is right. Probably the most known authentication infrastructure is the **Public Key Infrastructure** (PKI)[38], which is based on the concept of public key cryptography. The trust in a user's identity is established through a trusted third party, thus a pre-established trust relationship between the third party, grid users and resource providers is assumed. The trusted mediator is called Certificate Authority and is responsible for allocating the user's home domain identity into the grid identity and for issuing certificates with the allocated identity. The issued certificates are used by the users as a means to authenticate to the resources shared in the grid community.
Kerberos[26] is another security infrastructure used for authentication of user's identity and is also based on pre-established trust relationships. The role of the trusted mediator is performed by the authentication server. The trust in the user's identity is mediated with session keys issued by the Authentication Server acting as the trusted third party. For the purpose of accessing the available services the Kerberos infrastructure uses special access tokens that carry the information about the identity of an entity and the groups to which the entity belongs.
Athens[27] is an authentication infrastructure developed to control access to a wide range of shared resources. Users have an account for each resource they wish to access and these accounts are managed centrally by the Account Server. An agent enforcing access control is installed in every site sharing resources. The user provides his user name and password in order to be granted the access to the requested resource. This step is repeated every time the user wants to access an available resource.

3.1.2 Authorization infrastructures. With the growth of grid popularity the enforcement of access control based only on user's identity become insufficient and more fine-grained access control was necessary. The process of authorization is used to determine who is allowed to use shared resources and under what conditions, whereby the user's identity (and other user's attributes) is considered before the final decision whether or not to allow the access to a particular resource is made. **Grid-Map Files** (GMFs) [18] is the first access control infrastructure based not only on user's identity. The main idea behind GMFs is the usage of access control lists. A list pairing distinguished names of authenticated grid

users and local user accounts, to which these names are mapped, is stored on each shared resource. It is then left to the resource operating system and application access control mechanism to enforce the access to the resource.

Virtual Organization Membership Service (VOMS)[2] mediates trust between users and resource providers through a trusted third party – VOMS server. All information about a user is managed on the VO level by the VO administrator centrally. The VOMS server provides the user with attributes needed to access a shared resource in the form of attribute certificate. The user presents his attribute certificate issued and signed by VOMS server to a resource in order to access it. The resource checks the validity of the certificate and the attributes it contains. Subsequently, local resource access policies are applied and the user is granted or refused the access to the resource.

Another example of an authorization infrastructure is **PrivilEdge and Role Management Infrastructure Standard** (PERMIS)[6]. In order to access a resource protected by the PERMIS infrastructure the user needs to present a role based attribute certificate. The attribute certificate is issued by a source of authority and contains the user's role and attributes. PERMIS enables distributed role management, whereby certificates can be stored locally on the sites that allocated them. Before a decision whether or not to allow an access to a resource is made, the resource checks the user's certificate, role assigned to the user, and whether the certificate was issued by a trusted source of authority. Then the user is granted or refused the access to a requested resource.

The **Akenti**[1] infrastructure defines a special type of trusted entities called stakeholders. The stakeholders are trusted to issue use-condition certificates, which place conditions on certificates the user has to obtain in order to gain access to a resource. Every stakeholder can define use-condition certificates independently from other stakeholders. Hence, one resource can be protected by more access control requirements.

3.2 Ad hoc grid security mechanisms

Generally, the grid security protects shared resources against malicious actions of users and other entities that could damage the resources or corrupt the integrity of data stored and processed on the resources. However, in many situations the users of the ad hoc grid have to be protected from those who offer the resources, so the issue is also vice-versa[19]. The security mechanisms described in the section 3.1 are unable to provide this type of protection.

Authentication and authorization, which are referred to as hard security mechanisms, do not allow any occurrence of risk or uncertainty (the user either is authenticated and authorized to access a shared resource or is not), but collaborations in an open environment are necessarily coupled with potential dangers that necessitate reasoning about risk and uncertainty. Trust was recognized as an important aspect of decision making in many distributed systems and is used as a mechanism for managing the dangers and learning from past interactions in order to reduce the risk exposure. For example, trust and reputation systems support decision making on the Internet provided services, which are based on a trust derived from ratings assigned to a certain provider by customers after completion of a transaction. Other parties can use the trust and reputation derived from the aggregated ratings to decide whether or not to run a transaction with the rated party in the future. Trust management, which is referred to as a soft security, represents the shift from attempting to provide absolute protection against potential dangers to accepting dangers as an intrinsic part of any global computing[9, 19].

3.2.1 Trust. Humans rely on the trust every day and it is easy to understand what the meaning of trust is. However, the term trust is vague in its nature and hard to define generally. Fortunately, the scope of trust can be reduced to a level where it concerns only online environments like the Internet and distributed online systems. The notion of trust is used with variety of meanings and without any unified definition. However, in the literature two common definitions with well understood distinction between them are used. The **reliability trust**[20, 19] is defined as follows: *Trust is a subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends.* The **decision trust**[20, 19] is defined as follows: *Trust is an extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible.*

The reliability trust enables to make decisions whether or not to start a collaboration only on the basis of the collaborator's reliability estimation. On the other hand, the decision trust defines context as a part of the trust value and binds the estimation of the collaborator's reliability with the risk that arises from the uncertain outcome of the collaboration. Therefore, the decision trust seems to be a better choice for the purpose of trust modelling.

Trust is a directional relationship between a trustor and a trustee, whereby the trustor is a thinking entity making decisions whether or not to start collaboration with the trustee on the basis of the trustee's trustworthiness. In a grid the trust between the grid user and the resource provider is a mutual bidirectional relationship, because the user and the provider must be trustworthy to each other, otherwise the collaboration is not possible.

The mutual trust relationship can be described by the trust classes[19] as follows:

Provision trust describes the user's trust in a service or in a resource provider. The user trusts the provider to provide services that implement the advertised functionality and do not harm the user's resources. The provision trust ensures the reliability of the provider and is related to the integrity of the user's data stored in and/or obtained from the provided resources.

Access trust describes the provider's trust in the user intending to access to the offered resource, i.e. the provider trusts the user to use the resource in an agreed manner. This relates to the access control paradigm which is a central element in a computer security.

Delegation trust describes the trust in an agent, who acts and makes decisions on behalf of the relying party. The delegation trust can be seen as a special case of the provision trust, because the relying party trusts the delegate not to misuse the delegated rights.

Identity trust describes the belief that the claimed identity of an entity is true.

Context trust describes the extent to which the trusting party believes that the distributed system contains mechanisms necessary to support the transaction in a case something goes wrong.

3.2.2 Trust management. Trust between two entities is a bidirectional relationship and can be seen from two sides. The success and survival of an entity is dependent on the willingness of other entities to collaborate. Humans tend to collaborate only with trusted entities. Hence, the ability to gain trust of other entities is an important measure. There are many genetically determined or culturally acquired strategies helping the people to appear reliable and trustworthy. The easiest and probably most used strategy for gaining trust is simply to behave in a trustworthy and reliable manner.

However, the attempt to give a false impression of trustworthiness for the purpose of a personal gain is not uncommon. Therefore, it is not important only to represent own trustworthiness, but to correctly determine the trustworthiness of the target entities as well.
According to the twofold nature of trust relationships the **trust management**[20] is defined as follows: *The activity of creating systems and methods that allow relying parties to make assessments and decisions regarding the dependability of potential transactions involving risk, and that also allow the players and system administrators to increase and correctly represent the reliability of themselves and their systems.*
The parties in a computer mediated communication and collaboration are in a need for methodologies enabling them to properly assess the trustworthiness of remote parties and at the same time to be recognized as trustworthy entities. This need arises due to the fact that the computer networks move human collaborators away from a direct style of interaction. They can collaborate with collaborators they have never met and that they might never meet in person. The traditional methods of trustworthiness assessment and representation used in a physical world can therefore no longer be used. Simply expressed, the application of methodologies that enable such trusted collaboration in an online environment can be called trust management[20].

3.3 Comparison of traditional and ad hoc grid security

The traditional grid applications are hindered by the lack of security assurance from remote sites providing computing resources or other services. Trust models implemented in the grid environments support the user authentication and the single sign-on operations. However, the existing mechanisms are still inadequate to access local security conditions at sites participating in the grid community[32]. The situation in the ad hoc grid environments is similar. Grid nodes must assert the trustworthiness of a remote node according to the past experiences with the particular node and other considerable factors before the decision about the collaboration can be met.
In the traditional grid security infrastructures the identity trust, access trust and delegation trust are implemented as the processes of authentication, authorization and delegation of rights among the grid sites. Explicit implementations of the provision trust and context trust are missing. The proper behaviour of the resource provider and the user is guaranteed only by the third party acting as a trusted mediator. However, the trusted third party has no real mechanisms to ensure such behaviour.
In the traditional grid infrastructures trust is coupled with the establishment of the grid environment and is understood as an implicit part of the collaborations. On the other hand, in the ad hoc grids there are no implicit trust relationships among grid nodes. In the future the trust management mechanism could become responsible for the execution of collaborations in a presence of mutual trust.
The traditional and ad hoc grid security share a couple of similar features, e.g. the process of authentication is a prerequisite for other provided security services. However, the traditional and ad hoc grid security also differ in various characteristic features in regard to the grid architecture and the security needs of grid community participants. The most significant security characteristics of the both grid technologies are listed in the Table 2.

3.4 Trust value structure and modelling

Trust as a mechanism for managing the dangers and learning from past interactions is already recognized as an important aspect of decision making in the trust and reputation systems. Recently, the grid computing has recognized the trust management as an important part of the core grid functionalities as well. Several trust models enhancing the grid computing with the trust management were already proposed. The common characteristics and features of these models enable to derive the structure and the modelling approach of trust value as presented in the following subsections.

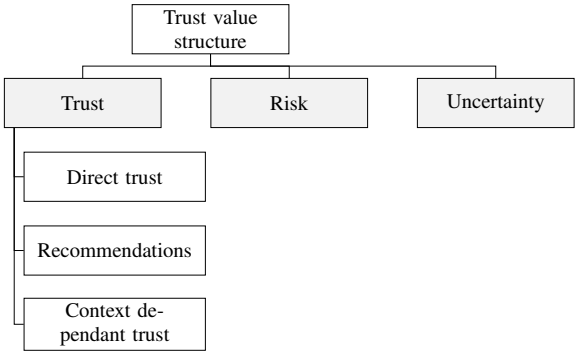

Fig. 3. Structure of the trust value and various trust relationship types among grid entities

3.4.1 Trust value structure. Trust between two entities is formed on the basis of direct interactions between the entities and recommendations resulting from the interactions the two entities had with other entities in the grid community. Generally, each event influencing the degree of the trust is interpreted by a trusting entity as either a positive or a negative experience. If the event is interpreted as a negative experience, the trustworthiness of the trusted entity is lowered and if the event is interpreted as positive, the trustworthiness is increased by some degree[31]. The current state of the system also influences the degree of trust of the trusting entity. Therefore, the direct experiences, recommendations and the state of the system are considered as factors determining the overall degree of trust and specify the various trust relationships among grid entities as depicted in Fig. 3.
The **direct trust** between two entities is mainly formed as a result of their previous interactions and is a part of the overall trust value in the majority of the trust models developed so far. However, the models differ in the method how the direct trust value is calculated. The model proposed by Song et al. [33, 32] specifies a prior job success rate, firewall capabilities, anti-virus capabilities and capabilities of intrusion detection system as a part of the final direct trust value. In the model proposed by Azzedin and Maheswaran [5] the direct trust is evaluated according to the behaviour of the evaluated entity, which is expressed as the willingness to abide requirements declared by the trusting entity and any violation of these requirements leads to a penalty in the direct trust.
The **recommendation trust** is referred to as a reputation of the trusted entity and can be described as everything that is generally said or believed about the entity's character or standing. If the trusting entity is aware of the trusted entity's reputation it can base its trust on that reputation, i.e. the trusted entity is trusted because of its good reputation. On the other hand, if the trusting entity has a private knowledge about the trusted entity (e.g. through direct experience) and the private information overrules any reputation the trusted entity might have, then the trusted entity can be trusted despite its bad reputation. Entities reveal and obtain reputation for the purpose of a decision making in several ways. The model proposed by Ding et al. [7] calculates the overall trust value according to the recommendation trust among VOs instead the grid entities. The

Table 2. Comparison of the traditional and ad hoc grid security characteristics

Characteristic	Grid type	
	Traditional	Ad hoc
Goal of grid security	Protection of shared resources and user's data from unauthorized access.	Protection of provided resources from malicious actions of users and protection of users from those providing the resources.
Prerequisites for provision of security services	Authentication of user's identity, secure communication and data transfer based on various cryptography technologies.	Authentication of collaborating entity's identity, secure communication and data transfer based on various cryptography technologies.
Provision of security services	Provision of security services is based on authentication and authorization mechanisms.	Provision of security is based on trust management.
Type of trust relationship	Implicit and unnamed trust relationships among grid community participants mediated through trusted third party.	Explicit trust relationship among ad hoc grid entities managed by each entity independently.
Supported trust classes	Identity trust as process of authentication, access trust as process of authorization and delegation trust as single sign-on operations.	Mutual trust relationship between two grid entities based on trust supporting all defined trust classes.

reason for this approach is the fact that the number of VOs is smaller than the number of grid entities. Hence, the reputation can be managed in a more scalable manner. The model proposed by Ryutov et al. [29] monitors behaviour of entities and if some action on one grid entity is regarded as insecure, the same behaviour is likely to be insecure to other similar entities as well. Therefore, the grid entities distribute warnings to other entities as soon as a threat was detected.

In the trust models the **situational trust** is not yet fully recognized as a factor influencing the overall trust the relying entity has in the trusted entity. This type of trust relationship can be described with the following example[20]: *Consider a person who distrusts a rope for climbing from the third floor of a house during a fire exercise. Imagine now that the same person is trapped in a real fire in the same house, and that the only escape is to climb from the third floor window with the same old rope. In a real fire, most people would trust the rope.* In the example, the reliability trust during the fire exercise is equal to the reliability trust in a case of fire: the rope is old and therefore distrusted. However, the definition of decision trust (in the section *3.2.1*) implies that the context of a current situation in a system influences the overall trust as well. Therefore, in case of a fire the decision trust is high enough to try to use the rope to escape from the building.

The principle of decision making whether or not a relying entity can collaborate with a trusted entity is mainly based on the evaluation of trust the relying entity has in the trusted entity. However, the definition of decision trust implies that trust in form of direct experiences, recommendations and context dependant information is not sufficient to determine the trustworthiness of the trusted entity. Therefore, the structure of trust value corresponding to the trustworthiness of the trusted entity is built from trust, as well as risk end uncertainty as depicted in Fig. 3.

Dangers are an inevitable part of any global computing system. Therefore, an explicit reasoning about the dangers causing a damage to the relying party is necessary during the process of decision making. The more important a flawless job execution is the more severe the damage in a case of failure becomes. The likelihood of a failure occurrence and the cost it incurs to the relying party is referred to as a risk. **Risk** and **trust** are related in the sense that there is no need for a trusting decision unless there is risk involved. Two alternative views of the relationship between trust and risk exist: risk determining level of trust and trust determining level of risk. The first view can be described as follows: in a particular situation or a particular action with a certain level of risk a principal should be trustworthy in order to be allowed to enter the situation or carry out the action, i.e. the level of risk determines the minimal level of required trustworthiness. The latter view is described as follows: in a particular situation or a particular action involving a principal with a certain level of trustworthiness the risk should be low enough in order to allow the principal to enter the situation or carry out the action, i.e. the level or trustworthiness determines the maximal level of acceptable risk[9]. The latter view seems more appropriate for the risk evaluation if the costs and the benefits of the entered situation are quantifiable.

Uncertainty refers to a situation where the relying party cannot be fully certain about the accuracy of the decision. In other words, the decision about possible future collaboration is done in a situation where the complete information is not present. For example, a situation can occur where two completely unknown entities have to collaborate, but they have neither the experiences with each other, nor recommendations from other entities are available. A similar situation can also occur if only a part of the information is available and other decision factors are missing. The lack of information must not necessarily result in a change of trust in the trusted entity. However, it can change the certainty about the final decision. Therefore, if the certainty is changed significantly, the level of trust is changed as well[9].

3.4.2 Trust value modelling. The notion of the trust can be expressed with a variety of different meanings and, similarly, the trust value can be evaluated with different approaches. The evaluation of the trust can be based on its vague nature, it can be modelled as a prediction of future collaboration outcomes or it can be obtained as a deterministically calculated value. The modelling approaches used for calculation of the overall trust value are fuzzy logic based approach, probability theory based approach and approach based on other mathematical methods as depicted in Fig. 4.

Trust in the **fuzzy logic based models** [23, 24, 33, 32, 34] is not modelled as an objective property of an entity. Instead, it is a subjective belief of the relying party about that entity and the trust value expresses to what extent the relying party is willing to depend on the other entity. Fuzzy logic models use linguistic terms

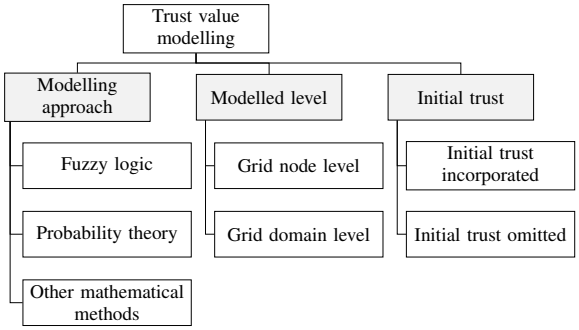

Fig. 4. Trust value modelling approach, levels of the modelled trust value in regard to the structure of the grid and incorporation of initial trust into trust modelling

rather than exactly calculated trust values to state how much believe an entity has in the collaborating entity. The relying party can describe the trusted entity as "Very trustworthy", "Trustworthy", "Untrustworthy" or "Very untrustworthy". In fact, the granularity of the expressions used for entity trustworthiness evaluation can vary. The expressions can either be defined directly in the model or the value specifications of the trust variable are left to the grid node's access policy.

The modelling approach in the fuzzy logic trust models is based on a fuzzy inference system. Grid node attributes and other relevant properties (e.g. direct trust and recommendation trust) are at first transformed from crisp values into membership grades for linguistic terms of fuzzy sets. The membership functions are a subject of the designer's choice. The transformed values are processed by applying fuzzy rules provided by experts or extracted from numerical data. The output fuzzy set is processed through the process of defuzzification and the output fuzzy values are transformed into crisp values. The output values obtained from the fuzzy inference system enable to make decision on whether or not the relying party should start the transaction with other entity. In the model proposed by Song et al. [33, 32] the calculated crisp value is called the trust index (TI) and represents the overall trustworthiness of the trusted entity. The relying party demands the trusted entity to provide security assurance by issuing a security demand (SD), which represents the required minimum trustworthiness. The transaction can then start, only when these two parameters satisfy the security-assurance condition: $TI \geq SD$.

The outcome of actions that are executed in the grid environment by the collaborating parties is unknown in advance. In the **models based on the probability theory** [7, 31] the trust is related to the prediction of what the outcome will probably be and is estimated on the basis of previous observations. In the model proposed by Shi et al. [31] the probability that the next execution of actions will be a point within a space of possible outcomes is described by a probability distribution called the outcome distribution, which enables to predicted a trust value applicable to several utility models for the purpose of decision making. Interesting is also the concept of suspicion level defined by Ryutov et al. [29]. The suspicion level indicates how likely an entity will act improperly and changes trust established between the two collaborating entities accordingly.

In the **models based on other mathematical methods** [5, 22] trust reflects the belief one entity has about the other one. i.e. the trusting entity expects the trusted entity to act in a certain manner. This expectation is based on the information about the trusted entity's attributes (e.g. technical capabilities and skills), previous experiences with the entity and recommendations from other trusted entities.

One of the requirements imposed on trust models is the ability to evaluate the trustworthiness of a grid node in a scalable manner. However, only few models consider the growing number of grid community participants as a factor influencing the performance of trust value calculation. In dependence to the level, on which the trust value calculation takes place, the trust modelling levels are divided into grid node level and grid domain level as depicted in Fig. 4.

In the most models the trust evaluation is performed on a **grid node level**. However, should the number of grid nodes participating in the grid community grow drastically; optimization of the trust value calculation may be needed. For example, the model proposed by Ying and Jiang [41] manages recommendations on the **domain level** and trustworthiness of entities is managed in their respective domains. The overall trust is determined according to the trustworthiness of an entity within the domain and the recommendation trust associated with the domain. The idea of reputation management on a higher level is also part of the model proposed by Ding et al. [7]. In this model the overall trust corresponds to the direct trust of the evaluated entity and the reputation of the virtual organization, of which the entity is part.

In an open environment it is not uncommon to start a collaboration with a completely unknown entity. However, the relying entity needs to make a decision whether or not to collaborate with that entity. If the entities in the community have no previous experiences with the unknown entity and recommendations are not available, the relying entity requires other means to evaluate the trustworthiness of the unknown entity. It could be reasoned that the lack of information about the entity makes the collaboration too risky to start. However, this precedent would disqualify any new entity joining the community from collaborating with entities already present in the community. Therefore, the **initial trust** was proposed as a means for determining the trustworthiness of an entity without a need for previous experiences and recommendations.

The issue of initial trust modelling in many trust models is addressed only indirectly[29, 33, 32], or is completely omitted [34]. The Fuzzy model proposed by Song et al. [33, 32] addresses the issue of initial trust value modelling indirectly. The overall trust value is calculated according to the past experiences and attributes characterizing the evaluated entity. New entity joining the community cannot be assigned a rating corresponding to the entity's behaviour. Therefore, only the attributes associated with the entity are considered during the process of decision making. In this context, the trust value calculated according to the entity's attributes may be understood as the entity's initial trust value.

4. GRID SCHEDULING

The integration of trust management into a traditional or ad hoc grid infrastructure requires cooperation of the grid security and scheduling services, i.e. the grid security service identifies and specifies trusted entities and the scheduling service provides task execution schedule according to the trustworthiness of available grid resources. This section and the following subsection present a description of the scheduling process and the integration of trust awareness into this process.

Grid scheduling (in traditional an ad hoc grids) is a process of assigning jobs, tasks or workflows on a set of heterogeneous resources scattered over multiple administrative domains and is categorized by architecture layout under three groups[28]: (i) centralized, (ii) decentralized, (iii) and hybrid scheduling.

In the **centralized grid scheduling** architecture the scheduling operations executed in the grid are managed by one central controller. The resources shared in the distributed environment must inform this controller about their static information and the current state. The main drawbacks of this approach are a single point of a failure represented by the controller and issues with scalability resulting from the growing number of shared resources. In the **decentralized grid scheduling** architecture the scheduling is performed on each grid node autonomously. A dedicated module contained in the node makes decision on the resource, on which the job is scheduled. The **hybrid grid scheduling** combines the centralized and decentralized approach. One scheduler manages a group of nodes subscribed to it. This scheduler communicates with other schedulers managing their own group of nodes. And if the job cannot be scheduled in the local group the scheduler delegates the job to the collaborating schedulers.

4.1 Traditional grid scheduling

In general, the grid scheduling is performed by one grid scheduler and more local schedulers. The main difference between a grid scheduler and a local scheduler is the fact that the grid scheduler itself has no direct control over dispersed resources. Therefore, it requires interactions with remote sites and their local scheduling systems. The grid scheduler delegates scheduling requests to the lower-level schedulers, which either control their local resources directly or have some kind of access to the resources. However, the concept of grid scheduling is not restricted only to the two levels. The lower-level scheduler might be represented either by a local scheduler managing a single resource or by a scheduling system managing several resources at once.
The grid scheduling is executed in three phases[30]: (i) resource discovery, (ii) system selection (iii) and job execution. **Resource discovery** involves determining which resources are available and creates a set of resources that passed minimal feasibility requirements. During the **system selection** a single resource (or a single resource set), on which the job is scheduled, is selected from the resources meeting the minimal requirements. Finally, the job is submitted to the scheduled resource and its **execution** is started.
No common and generic grid scheduler yet exists, but several common aspects can be found examining the grid scheduling use cases[15]. According to those aspects the grid scheduling phases can be split into the following steps[30, 40]:

(1) **Authorization filtering** corresponds to the fact that it does not make a sense to schedule a job on an unauthorized resource. Without authorization to use the resource, the job will simply not run. Therefore, authorization filtering is responsible to determine a set of resources, to which the user submitting the job has access.
(2) **Requirements definition** of the user's job is used to further filter the set of feasible resources. The definition may include static details (e.g. operating system) as well as dynamic details (e.g. minimum RAM available). However, the more details are included in the definition, the better the resource selection can be.
(3) **Minimal requirement filtering** performs the selection of resources that meet the minimal job requirements stated by the definition. The resources not meeting the requirements are filtered out.
(4) **Dynamic information gathering** is responsible to collect detailed dynamic information about resources needed to make the best possible job - resource match.
(5) **System selection** executes the actual resource selection, on which the job is scheduled. The selection involves the creation of a schedule optimizing the task completion time and/or other criteria corresponding to the quality of services demanded by the user.
(6) **Advance reservation** is an optional step and is meant (if the resource enables it) to reserve a specific time frame on the selected resource. Once the time of the job execution comes, the resource is claimed and the job is executed.
(7) **Job submission** represents the process of moving the user's job to the selected resource together with the job definition, application executables and data.
(8) **Preparation tasks** refer to actions that must be executed before the execution of the user's job can start. For example, this step might involve actions as claiming of the reserved resource, staging the job to the resource or moving data to the resource.
(9) **Monitoring** the **progress** of the job execution is needed in case some event occurs that causes to interrupt the execution and reschedule the job; or when the user decides to move the job to other resource.
(10) **Job completion** involves notifying the user that the job finished and he can obtain the results produced by the job.
(11) **Cleanup tasks** are used for retrieving calculated data from the resource, removing temporary settings, and so forth.

4.2 Ad hoc grid scheduling

In the section 2.2 the ad hoc grid is defined as an architecture with structural independence, which enables the participating nodes to collaborate without depending on any external infrastructure for assistance. Hence, the centralized grid scheduling architecture is unsuitable for implementation in the decentralized ad hoc grid environment. Contrariwise, several decentralized and hybrid scheduling architectures were proposed and implemented.
In the hybrid scheduling architectures [4, 25, 35] the nodes are organized into several clusters. A cluster is a group of nodes that are in a certain geographic proximity[4, 35]; or are part of the same local network[25] that allows them to communicate directly with the local cluster's dedicated operator. The operator manages its local cluster and handles subscription of nodes joining the cluster and scheduling of tasks.
In the decentralized scheduling architectures [14, 17, 8, 37] the resource discovery and information services are provided locally on each node instead of the centralized server in the centralized architecture or the operator node in the hybrid architecture. To schedule a task on a resource the scheduler needs information about available resources and the current state of the resources. The MoGrid infrastructure[14, 8] utilizes a mechanism of flooding messages sent from a node searching for available resources to its neighbour nodes. The node receiving the request message also propagates the message to its neighbours until a certain value of propagations is achieved. The receiving nodes reply to the request message according to their willingness to collaborate as a resource provider. Other approaches to obtain a task - resource pair in the decentralized environment are the usage of mobile agents[37] and virtual shared memory[17].

4.3 Trust awareness in grid scheduling

The scheduling of task in the heterogeneous distributed systems is NP-complete optimization problem[12]. Therefore, a good approach is to search for suboptimal solutions, where sufficiently efficient algorithms exist. The traditional scheduling algorithms proposed for the purpose of task completion time minimization in-

clude the following (i) algorithms: Min-Min, Max-Min, Minimum Completion Time, Minimum Execution Time, Highest Response Time, Hill Climbing, Tabu Search (ii) and heuristics: Simulated Annealing, Ant Colony, Genetic Algorithm, Particle Swarm Optimization and other. It is important to note that these algorithms and metaheuristics do not consider trust management as a part of the scheduling process.

Grid infrastructure was designed primarily for the cooperation of scientists knowing each other. Therefore, there existed implicit trust relationship among them. If the resources are shared among unknown parties (e.g. the grid is used for business) then the implicit trust relationships are missing. However, the resource consumers want to allocate their applications only to those resources that are owned and/or managed by trusted resource providers. Similarly, the resource providers want to allow the access to their resources only to trusted resource consumers.

Trust has been recognized as an important aspect of the security provided by the grid. However, the process of grid scheduling must be aware of the trust relationships among the collaborating parties in order to integrate trust management into the grid infrastructure. An example of such integration is the scheduling model proposed by Kashyap and Vidyarthi [21]. The aim of the model is to maximize the security offered to scheduled tasks and to minimize the security overhead resulting from the usage of mechanisms providing that security. The two objectives are scheduled with a dual objective scheduling algorithm, i.e. when one objective is optimized the other is taken as a constraint an vice-versa. To view more trust aware scheduling algorithms the interested reader is referred to algorithms proposed in [42, 39].

4.4 Trust awareness in ad hoc grid scheduling

The purpose of the trust management in the ad hoc grid environment is to guarantee the quality of services provided by the grid nodes and the quality of user's behaviour. The integration of trust into the ad hoc grid infrastructure is coupled inseparably with the scheduling of jobs on the provided resources. However, there are some differences between the ad hoc and the traditional grid scheduling when the trust management is involved.

In order to integrate trust management into the ad hoc scheduling process, the steps executed during the resource discovery, system selection and job execution (described in the section 4.1) must perform the following additional tasks: (i) definition of minimal trustworthiness needed to begin the collaboration between the user and the resource provider, (ii) determining the current trustworthiness of the involved nodes, (iii) and update of trustworthiness after the job completion. It is evident that these tasks impose new requirements on the ad hoc grid architecture. The architecture depicted in the Fig. 5 introduces the trust manager module as a solution to meet the imposed requirements.

During the phase of resource selection the user defines the job and the requirements needed for the job to run. To select a trustworthy resource, the user defines the security demand[33, 32], which is taken as a constraint during the system selection step. The security demand is determined either directly by the user as one of the job requirements, or by the trust manager according to the parameters provided by the user.

The resources that passed the authorization and minimal requirements filtering are assigned a trust index [33, 32] determined by the trust manager on the basis of static and dynamic information about the resources, job definition parameters and other factors managed by the trust manager as depicted in Fig. 6. The trust index is a combination of more parameters, but what parameters and how exactly they are used for the trust index evaluation depends on the used trust model. The minimal components of the calculated value are the direct trust and the recommendations. However, other factors as risk, uncertainty and context dependant information should be included in the trust index as well. It is important to note that the scheduler uses the security demand and the trust indexes obtained from the trust manager only to exclude the untrustworthy resources. The schedule optimization itself is not affected and still corresponds only to the quality of services demanded by the user.

The resource provider demands a certain level of trustworthiness as well as the user. Therefore, after the exclusion of untrustworthy resources the scheduler requests the most optimal and trusted resource to consent to the future collaboration. The decision whether or not to accept the collaboration is based on the resource's security demand and the trust index assigned to the requesting node. Both values are obtained from the resource's trust manager on basis of job parameters included in the request, recommendations, previous experiences, uncertainty, risk and other factors. The decision on the collaboration is responded back to the scheduler. In case of negative response the scheduler sends the request for consent to next most optimal resource until an affirmative answer is received.

The job scheduled with the help of the trust manager is forwarded to a module responsible for job submission and execution. After the job completion the result of the execution is transferred to the requester node. The trust update is the final step involving the trust manager module on the requester node as well as the resource node. The update is performed according to a positive or a negative experience resulting from the job execution and is necessary for correct representation of trust in the collaborating parties.

5. FUTURE WORK

The participants of the grid community collaborating in the grid environment have different requirements for the grid security infrastructure. The user is interested in competence of the shared resources to reliably execute the user's code and to protect his data from unauthorized access or modification. Similarly, the resource provider wants to collaborate only with reliable and authenticated users not compromising the shared resource or the integrity of unauthorized data.

Each participant of a collaboration has his own set of expectations for the quality and performance of the collaboration and is satisfied with the executed collaboration only if the required expectations were met. Trust in this context can be used to express the confidence of the relying entity that a collaborating party will meet the desired expectations. The expectations for the quality of collaboration placed by the users and resource providers can be mapped to various system parameters and capabilities.

The task for the future research is to determine what system parameters of the collaborating entities are relevant for the trust evaluation and what are the relations among the parameters, risk and uncertainty. The research should also specify the method for parameters measuring and define the procedure for combination of parameters with risk and uncertainty.

6. CONCLUSION

The traditional and ad hoc grid infrastructures differ in the structure of their respective architecture and the amount of control over the job execution managed by the participating nodes. Therefore, the both infrastructures require a different approach to implement the fundamental set of grid functional capabilities. The capabilit-

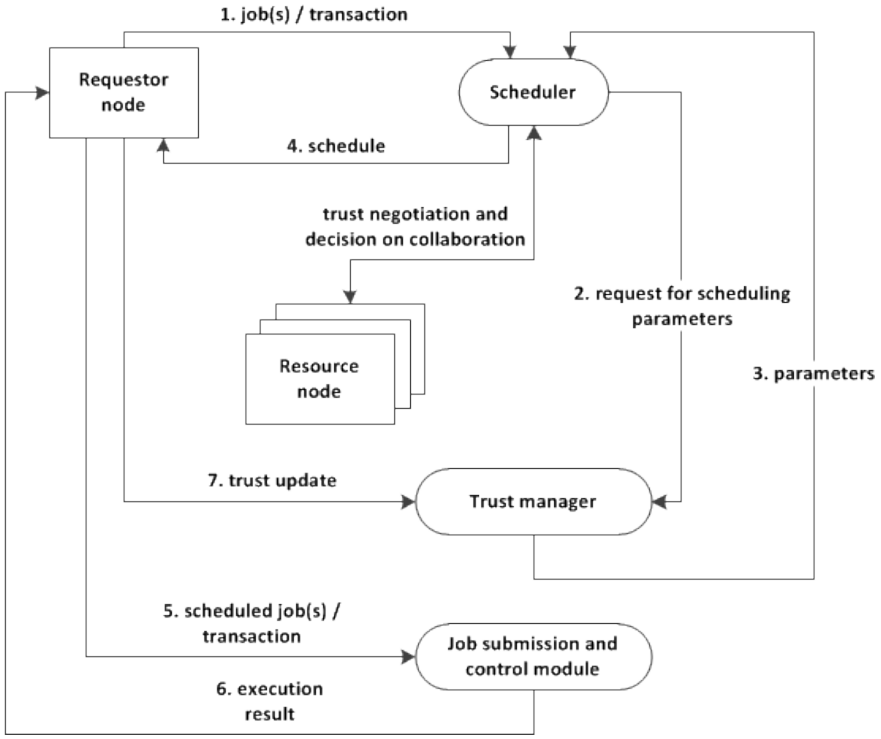

Fig. 5. Trust manager integration into the ad hoc grid infrastructure from the requester's point of view

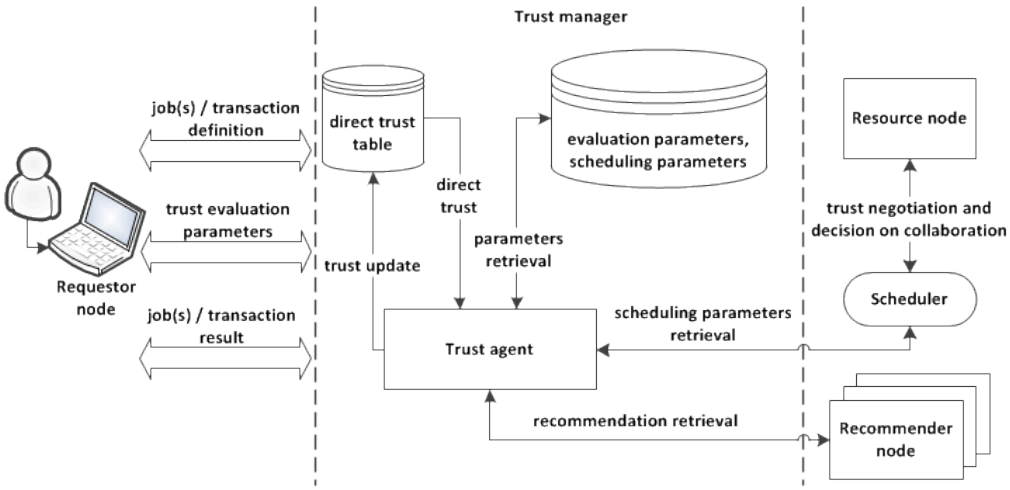

Fig. 6. Trust manager architecture from the requester's point of view

ies surveyed by the paper are the task scheduling process and the provision of security.

The steps executed during the resource discovery, system selection and job execution are similar in both types of grid infrastructures, but the implementation of scheduling differs in the actor performing those tasks (grid node, dedicated operator or centralized grid scheduler) and the amount of involved trust.

In the traditional grid infrastructure the trust is managed by the trusted third party, which is the basis for the authentication of user's identity. Another key paradigm coupled with the security provision is the authorized access to the available resources. The paper describes several infrastructures currently used to integrate trust into the traditional grid solutions.

The structure and control independent nature of the ad hoc grid necessitates to manage the security in the absence of a central controller and the responsibility for the protection against malicious collaborators is left to the grid nodes. The issue of security provision in the ad hoc grid environment can be addressed with the integration of trust management into the scheduling process. However, the integration of trust management results in changes in the schedul-

ing process and also necessitates enhancements in the ad hoc grid architecture. The paper presents an ad hoc grid architecture integrating a trust manager module taking over tasks as trustworthiness assessment of collaborating nodes and update of trustworthiness after a job completion.

7. REFERENCES

[1] Akenti, 2015.

[2] Roberto Alfieri, Roberto Cecchini, Vincenzo Ciaschini, Luca dell'Agnello, Alberto Gianoli, Fabio Spataro, Franck Bonnassieux, Philippa J. Broadfoot, Gavin Lowe, Linda Cornwall, Jens Jensen, David P. Kelsey, Ákos Frohner, David L. Groep, Wim Som de Cerff, Martijn Steenbakkers, Gerben Venekamp, Daniel Kouril, Andrew McNab, Olle Mulmo, Mika Silander, Joni Hahkala, and Károly Lörentey. Managing dynamic user communities in a grid of autonomous resources. *CoRR*, cs.DC/0306004, 2003.

[3] Kaizar Amin, Gregor von Laszewski, and Armin R. Mikler. Toward an architecture for ad hoc grids. In *12th International Conference on Advanced Computing and Communications (ADCOM 2004), Ahmedabad*, pages 15–18, 2004.

[4] Nazareno Andrade, Lauro Costa, Guilherme Germóglio, and Walfredo Cirne. Peer-to-peer grid computing with the ourgrid community. In *23rd Brazilian Symposium on Computer Networks (SBRC 2005) - 4th Special Tools Session*, 2005.

[5] F. Azzedin and M. Maheswaran. Evolving and managing trust in grid computing systems. In *Canadian Conference on Electrical and Computer Engineering, IEEE CCECE 2002*, volume 3, pages 1424–1429, 2002.

[6] David W. Chadwick, Alexander Otenko, and Edward Ball. Role-based access control with x.509 attribute certificates. *IEEE Internet Computing*, 7(2):62–69, March 2003.

[7] Changsong Ding, Yi Fu, Zhigang Hu, and Peng Xiao. A novel trust model based on bayesian network for service-oriented grid. In *ACIS-ICIS*, pages 494–499. IEEE Computer Society, 2009.

[8] Luciana dos S. Lima, Antônio T. A. Gomes, Artur Ziviani, Markus Endler, Luiz F. G. Soares, and Bruno Schulze. Peer-to-peer resource discovery in mobile grids. In *Proceedings of the 3rd International Workshop on Middleware for Grid Computing*, MGC '05, pages 1–6. ACM, 2005.

[9] Colin English, Sotirios Terzis, and Waleed Wagealla. Engineering trust based collaborations in a global computing environment. In *iTrust*, volume 2995 of *Lecture Notes in Computer Science*, pages 120–134, 2004.

[10] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, and J. Von Reich. The open grid services architecture, version 1.5, July 2006.

[11] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of High Performance Computing Applications*, 15(3):200–222, August 2001.

[12] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[13] Gridbus, 2015.

[14] Antônio Tadeu A. Gomes, Artur Ziviani, Luciana dos S. Lima, and Markus Endler. Performance evaluation of a discovery and scheduling protocol for multihop ad hoc mobile grids. *Journal of the Brazilian Computer Society*, 15(4):15–29, 2009.

[15] Christian Grimme, Joachim Lepping, Alexander Papaspyrou, Philipp Wieder, Ramin Yahyapour, Ariel Oleksiak, Oliver Wäldrich, and Wolfgang Ziegler. Towards a standards-based grid scheduling architecture. In *Grid Computing*, pages 147–158. Springer US, 2008.

[16] Globus toolkit, 2015.

[17] K.A. Hummel and G. Jelleschitz. A robust decentralized job scheduling approach for mobile peers in ad-hoc grids. In *Seventh IEEE International Symposium on Cluster Computing and the Grid, 2007. CCGRID 2007.*, pages 461–470, May 2007.

[18] Wei Jie, Junaid Arshad, Richard Sinnott, Paul Townend, and Zhou Lei. A review of grid authentication and authorization technologies and support for federated access control. *ACM Computing Surveys*, 43(2):12:1–12:26, February 2011.

[19] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, mar 2007.

[20] Audun Jøsang, Claudia Keser, and Theo Dimitrakos. Can we manage trust? In *Trust Management*, volume 3477 of *Lecture Notes in Computer Science*, pages 93–107, 2005.

[21] R. Kashyap and D. P. Vidyarthi. Dual objective security driven scheduling model for computational grid using ga. *IAENG International Journal of Computer Science*, 39(1):71–79, 2012.

[22] D. Kaur and J. SenGupta. A trust model based on p2p trust models for secure global grids. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 1103–1108, June 2012.

[23] Hongmei Liao, Qianping Wang, and Guoxin Li. A fuzzy logic-based trust model in grid. In *Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing - Volume 01*, NSWCTC '09, pages 608–614, 2009.

[24] Hongmei Liao, Qianping Wang, and Guoxin Li. A reliable fuzzy theory based reputation system in grid. *Journal of Computers*, 5(5):782–790, 2010.

[25] H. Morsy and H. El-Rewini. Adaptive scheduling in a mobile ad-hoc grid for time-sensitive computing. In *2013 ACS International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8, May 2013.

[26] B. C. Neuman and T. Ts'o. Kerberos: An authentication service for computer networks. *IEEE Communications Magazine*, 32(9):33–38, September 1994.

[27] Openathens, 2015.

[28] R. Ranjan, A. Harwood, and R. Buyya. Peer-to-peer-based resource discovery in global grids: A tutorial. *Communications Surveys and Tutorials*, 10(2):6–33, April 2008.

[29] Tatyana Ryutov, Li Zhou, Clifford Neuman, Noria Foukia, Travis Leithead, and Kent E. Seamons. Adaptive trust negotiation and access control for grids. In *GRID*, pages 55–62. IEEE, 2005.

[30] Jennifer M. Schopf. Ten actions when grid scheduling: The user as a grid scheduler. In *Grid Resource Management*, pages 15–23, Norwell, MA, USA, 2004. Kluwer Academic Publishers.

[31] Jianqiang Shi, Gregor Bochmann, and Carlisle Adams. A trust model with statistical foundation. In *Formal Aspects in Security and Trust*, volume 173 of *IFIP International Federation for Information Processing*, pages 145–158. Springer US, 2005.

[32] Shanshan Song, Kai Hwang, and Yu-Kwong Kwok. Trusted grid computing with security binding and trust integration. *Journal of Grid Computing*, 3(1-2):53–73, 2005.

[33] Shanshan Song, Kai Hwang, and Mikin Macwan. Fuzzy trust integration for security enforcement in grid computing. In *Network and Parallel Computing*, volume 3222 of *Lecture Notes in Computer Science*, pages 9–21. Springer Berlin Heidelberg, 2004.

[34] P. Suresh Kumar and S. Ramachandram. User satisfaction based quantification of direct trust in t-grid computational model. In *Computer, Communications, and Control Technology (I4CT), 2014 International Conference on*, pages 438–442, Sept 2014.

[35] Pablo G. S. Tiburcio and Marco Aurélio Spohn. Ad hoc grid: An adaptive and self-organizing peer-to-peer computing grid. In *IEEE 10th International Conference on Computer and Information Technology (CIT)*, pages 225–232. IEEE Computer Society, 2010.

[36] Uniform interface to computing resources, 2015.

[37] Zhi Wang, Qi Chen, and Chuanshan Gao. Implementing grid computing over mobile ad-hoc networks based on mobile agent. In *Fifth International Conference on Grid and Cooperative Computing Workshops, 2006. GCCW '06.*, pages 321–326, Oct 2006.

[38] Joel Weise. Public key infrastructure overview, 2001.

[39] Yujiex Xu and Wenyu Qu. A trust model-based task scheduling algorithm for data-intensive application. In *2011 Sixth Annual Chinagrid Conference (ChinaGrid)*, pages 227–233, Aug 2011.

[40] R. Yahyapour and Ph. Wieder. Grid scheduling use cases, March 2006.

[41] Gao Ying and Zhan Jiang. A layered trust model based on behavior in service grid. In *2010 2nd International Conference on Advanced Computer Control (ICACC)*, volume 5, pages 511–515, March 2010.

[42] Shanyu Zhao, Virginia Lo, and Chris GauthierDickey. Result verification and trust-based scheduling in peer-to-peer grids. In *Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing*, P2P '05, pages 31–38, Washington, DC, USA, 2005. IEEE Computer Society.