

Contracts in a Grid Age

Russell Lock

Computing Department, Lancaster University

Email: r.lock@comp.lancs.ac.uk

Expected Thesis completion date 2004

Abstract

This position paper explores the issues surrounding authorization in modern grids, focusing on the development and use of Globus middleware(1). In particular this paper focuses on the future needs of inter-organisational business relationships in comparison to the currently available infrastructure. Finally an approach to resolving these issues is provided by building upon the existing security mechanisms.

Introduction

The emergence of VO's in industry and the need for increasingly complex grid systems have led to service based architectures taking hold in the grid research community. The current leading service based implementation is the Globus Toolkit v3 (Based on OGSA) which has been heavily overhauled from its previous incarnation v2 to allow for service based operation and development. OGSA enables more flexibility and considerably more transparency in grid services through numerous interface specifications developed in collaboration with the GGF group(2).

Previous incarnations of the Globus Toolkit whilst being very powerful have had limited application due mainly to their architectures, which favoured relatively small isolated grids working independently, preferably in an environment where its users were trusted. This situation is changing though, as the number of grids has grown with increasing need for interoperability; and a user base that has grown beyond the limits of a simple trust relationship.

However many of the higher level issues relating to security are only now being addressed. For example the Globus v3 security model has been carried over from v2 with only minor alterations to proxy layout. Given v2 was designed for isolated research settings it is clear that either extensions to, or the implementation of alternate authorization mechanisms is required to keep in line with VO grid development needs.

Security in Globus

The security model within Globus v2 and v3 (OGSA) relies heavily on X.509 certificates for authentication & authorization purposes.

In the traditional grid setting few have access to the grid itself and the turnover of staff involved is likely to be minimal over a given time span. Under these circumstances the model works well. However the model does not scale well due to its reliance on X.509 for both authentication and authorization. This prescribes a one to one relationship between a given users certificate and a relating manual entry in an ACL at the service itself. Looking to the future, within large dynamic VO's and companies this level of micromanagement would quickly become overwhelming. The grid development community have been aware of this issue for some time and more advanced systems have been designed, which will be covered further on.

Another important issue is the way in which services are utilized. The traditional grid setting tends towards one of two scenarios.

- 1) A large dataset accessed through a grid service
- 2) A large parallel programming task submitted to a grid for processing

These scenarios generally lead to a situation where a user either has access to a given service or not. This is increasingly becoming an invalid assumption as grids grow and offer variable levels of service on an economic basis. This highlights the need for more flexible authorization controls than a simple ACL. It is important to note that these issues do not point to a long-term problem with X.509 but specifically with its use for authorization as well as authentication in a business setting.

Alternate Authorization mechanisms

A number of alternate mechanisms exist for allowing access to a given system; many of which have been adapted for use in grids. The following section contains a brief overview of the main types.

Token based (limited time span)

The most dominant of these is Kerberos which is already Globus compatible.

RBAC (Role Based Access Control)

RBAC systems have never gained more than a niche in the market due to the difficulties in creating user groups to match people's roles.

Attribute based authorization

Attribute based authorization mechanisms such as Nereus(3) have also had limited impact partly because of the heterogeneous nature of users.

Issues relating to authorization and use of services are generally dealt with in the outside world through contracts and SLA's. Previous attempts to translate these heavily worded documents to allow QOS specifications within computer systems have met with limited success. A considerable amount of research into the dynamics and phases of electronic contract negotiation exist. Some of the more notable works include Research on B2B contracts by Goodchild(4) which deals with many of the issues relating to storage of contracts in XML format. Also Work by both Angelov(5) and Daskalopulu(6) which highlights the use of computers as tools (both automated and passive) to create contracts as well as highlighting the projects presently underway to develop tools for the future.

The SNAP(7) protocol designed at Argonne Laboratories alongside Globus relies on SLA's to allow QOS specifications to be addressed. However it takes a hierarchical view with a series of interconnected SLA's built up between resources. Currently the model recognises the need for negotiation and lays out a language for specification but does not address the specifics of negotiation at this time. Interestingly it also assumes relative trust between parties and currently does not as yet deal with the economic cost element of contracts.

The area of grid computing only encompasses a small subset of the area general contracts serve. This avoids many of the problems that expert systems and automated agent based contract negotiators have had in the past due to lack of domain knowledge. Therefore it is possible to conceive that by simplifying the contracts themselves beyond that of Goodchild's model whilst retaining the use of XML as a storage medium; coupled with a rule based language similar to that laid out in SNAP it would be possible to automate the process of grid contract creation. Such a system would rely on synchronous negotiation between contract negotiating software on both sides, with input from users only at the beginning and end of the process, removing the need for slower manual asynchronous negotiation techniques. The mechanisms needed to allow a user to utilize this type agreement for authorization are

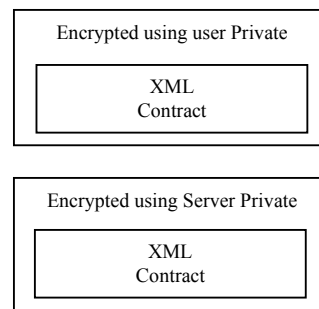
shown in the next section, starting with the securing of a given contract for authorization.

Contracts for grid Authorization

In order to understand how authorization can be achieved using this mechanism a brief outline of X.509(8) is shown below.

Globus authenticates using public key X.509 technology. An X.509 pair consists of a public key holding certificate, detailing information about the user including the CA that issued the certificate, as well as a corresponding private key. These allow a user to prove whom they are based on the sole use of a private key to decrypt public key encrypted messages sent to them. Assuming this technology is secure enough for the grid generally it is possible to make use of these public private key pairs to encrypt a contract laid out in XML, such that a mediator could prove that both users agreed to the same contract. The basic structure of this is illustrated below.

Fig 1:



If both match, they both agreed to the same contract. These can be authenticated by anyone with the public keys of both parties.

This model of authorization requires a considerably more lightweight server end authorization mechanism to process incoming requests due to its reliance on the actual information received (the contract) rather than a database of individual permissions stored locally. This allows for easier replication of authorizing mechanisms without the need to control multiple copies of databases.

Automating the Negotiation process

Many services offered by businesses appear to operate on a fixed cost basis. For example a can of beans costs a given amount in a certain place. But are in fact variable at most levels of the supply chain based upon how much of something you want. These economies of scale apply to grid computing and are a necessary part of the bartering process. Compromise is also an important factor in most negotiations. Meaning that the negotiator needs leeway on what is

acceptable from the user prior to negotiation if it is to be automated successfully.

The grid contract design allows for any number of discrete (yes, no etc) and variable (10, 20.5 etc) field types to be included in the negotiation process. An automated negotiator may need many iterations of offer and reply before a suitable compromise is met. However in contrast to a manual contracting process the negotiation phase would take considerably less time to complete.

The next section describes the structure of this iterative phase of contract creation within the prototype that has been developed.

Iteration Layouts

The contract iterations are split into two sections, the header and the main section. The contract iterations are full contracts marked "request", instead of "contract" in part of the header. The fields in the header are mostly compulsory and deal with matters such as, username, service names and descriptions etc. They also specify the creation and expiry dates for the contracts as well as a validity period statement for this particular offer. The main section consists of the user options / constraints as well as a few compulsory fields like cost. An example of an optional field could be Auto_Resubmission (on failure).

There are two main supported types of field layout; variable and fixed. For this paper variable relates to an option that uses a number system for decision making whereas a fixed uses string based discrete options. The reasons for this will become clear in the next section.

Note that besides the compulsory header and a few compulsory fields in the main section the user can add any other type of option they require. This flexibility in content means it is vital that a pre-negotiation phase takes place where all supported options are revealed to the user, to ensure only logistically feasible bartering stances are taken in the automated negotiation phase.

User Controls

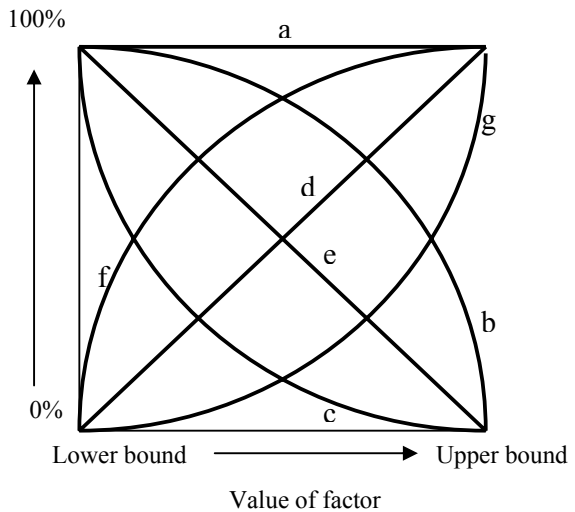
The users themselves should be removed from the negotiation itself as far as possible, with input only at the beginning of the process to indicate acceptable bounds and needs, and to accept or decline a contract at the end. This way a truly autonomous system can be achieved without user related bottlenecks once the process has commenced.

The initial decisions allowed by the user include the following mechanisms.

- To specify bounds (upper and lower) for variable options
- To specify specific options for fixed (or to specify no opinion)
- To specify bias via a slider bar to indicate the relative importance of different fields by the user
- To indicate for each variable option a graphical model to be used in negotiation (see below)

The system itself relies upon a series of mathematical formulas to deduce the relative acceptability of contract offers during the negotiation itself. These are manifested to the user as a series of pictorial graphs from which they pick an acceptable starting formula. An example set of these is shown below with descriptions as to their exact meanings.

Acceptability



Key:

- This straight line would represent a situation where all acceptable bounded values elicit a 100% acceptability rating
- This curved line represents a gradual decreases over lower values but exponentially towards higher ones
- This curve is the pair of B with gradual decreases at higher values but high decreases at low ones
- A direct correlation between value and acceptability where a higher value is better

- E) The inverse of D with lower values eliciting more acceptability
- F) This curve is the pair of G with gradual increases at higher values but high increases at low ones
- G) This curved line represents a gradual increases over lower values but higher increases at high ones

There is also a mechanism to allow the user to program more fine grained decisions. A simple form of rule input of the following form can be entered into the system to allow more complex dependencies and factors. The purpose and complexity of these entries vary but they are constructed from standard operators such as: >, <, =, AND, OR, SHOULD, SHOULDNOT

These can be used to construct rules for the system to ensure bounds aren't exceeded under certain circumstances, for example.

IF Cost > 50 THEN Availability SHOULDNOT < 10

In the example above the operator SHOULDNOT is used to indicate that the negotiator has leeway if it finds the situation impossible to work around.

The system then negotiates continually altering its offers based upon an acceptability quotient calculated from these graphs, the equations and the user bound inputs. Once a certain threshold is achieved the system completes the negotiations.

External Factors

In order to concentrate upon the core negotiation process shown above some parts of the architecture required to allow the type of system to operate have been left as external issues. These include the need to provide for revocation of contracts, mediation procedures for failed contracts, brokers to locate services and repositories for public keys.

Conclusions

The model shown here is currently in a prototype stage with implementation through XML with SOAP for information transfer. It is based upon Java code and utilizes the same X.509 certificates as the current Globus v3.

Further Work

In the future a more extensive rule set for bias inputting, and a more advanced bartering engine will allow considerably more precision in conveying constraints and negotiating deals. Currently the prototype is being extended to barter simultaneously with multiple parties.

References

- [1] Globus Middleware
www.globus.org
- [2] GGF (Global Grid Forum)
www.ggf.org
- [3] Nereus:
Miklos, Z. A decentralized Authorization Mechanism for E-business Applications.
http://www.infosys.tuwien.ac.at/Staff/mz/miklosz_decentralized.pdf
- [4] Goodchild
Goodchild, A. Herring, C. Milosevic, Z. Business contracts for B2B
<http://titanium.dstc.edu.au/papers/ISDO00.pdf>
- [5] Angelov
Angelov, S. Grefen, P. B2B eContract handling.
<http://www.ub.utwente.nl/webdocs/ctit/1/000005e.pdf>
- [6] Daskalopulu
Daskalopulu, A. Sergot M. The Representation of Legal Contracts
<http://arxiv.org/ftp/cs/papers/0106/0106005.pdf>
- [7] SNAP
Czajkowski, K. Foster, I. Kesselmann, C. Sander, V. Tuecke, S. SNAP.
<http://citeseer.nj.nec.com/538954.html>
- [8] X.509 overview
<http://www.ipa.go.jp/security/rfc/RFC3280-01EN.html>

ACKNOWLEDGMENTS

I would like to thank the UK Engineering and Physical Sciences Research Council, grant number GR/M52786 and the Dependability Interdisciplinary Research Collaboration (DIRC).