GWD-R
GGF Working Group Chairs and Steering Group

Joshua Apgar, Avaki Corporation
Andrew Grimshaw, Avaki Corporation
Steven Harris, Avaki Corporation
Marty Humphrey, University of Virginia
Anh Nguyen-Tuong, Avaki Corporation

January, 2002
Revised 5-February, 2002

Secure Grid Naming Protocol (SGNP): Draft Specification for Review and Comment

Status

This document is a draft specification of the Secure Grid Naming Protocol (SGNP), and is intended to afford the community an opportunity to evaluate the effectiveness and suitability of the SGNP as a logical naming convention (and associated protocols) in Grid Systems, to provide feedback to the authors, and to stimulate discussion of the SGNP as a potential standard

Copyright Notice

Abstract

We propose herein a Secure Grid Naming Protocol (SGNP), a new protocol that defines a means for system identification of resources that participate in large-scale distributed systems. The Secure Grid Naming Protocol (SGNP) introduces a straightforward approach to identifying resources that participate in Grid systems. The SGNP addresses the stringent requirements of large-scale, dynamic Grid systems—requirements that have not thus far been addressed by other naming approaches. Most importantly, SGNP does not introduce the need for a central trusted authority, thus enabling the creation of scalable Grid systems and facilitating a variety of practical models for Grid administration.

Table of Contents

1.0 Purpose of this Document

This document is a draft specification of the Secure Grid Naming Protocol (SGNP), a new protocol that defines a means for system identification of resources that participate in large-scale distributed systems. This detailed specification of the protocol is intended to:

- Enable readers to evaluate the effectiveness and suitability of the SGNP as a logical naming convention (and associated protocols) in large-scale distributed systems (Grid Systems), and to provide feedback to the authors
- Stimulate discussion of the SGNP as a potential standard

In addition to providing this draft specification, the authors and their colleagues are currently at work on a reference implementation of the SGNP.

1.1 The Grid Environment

The SGNP anticipates the requirements of large-scale distributed Grids while at the same time offering a practical solution for more modest Grids with similar challenges. The design of the SGNP is based on the following observations of Grid environments:[1,2]

- Grids, particularly those in multi-organizational or extended-enterprise settings, may be composed of a very large number of resources (billions, possibly trillions[3]) that need to be named, identified, and authenticated.
- Grids may contain resources that are managed and owned by multiple mutually distrustful organizations and individuals.
- Each organization may have its own security requirements, usage policies, and administration requirements.
- Resources may be geographically separated (different offices, different buildings, different campuses, different countries or continents).
- The resources and the network may be faulty.
- Over time, a resource may be migrated to a different network address, a different machine, or a different administrative domain.

2.0 What is the Secure Grid Naming Protocol?

The Secure Grid Naming Protocol (SGNP) defines a scheme for location-independent logical naming of grid resources, as well as a mechanism by which the identity associated with the names can be authenticated without strictly requiring a trusted third party.

2.1 Grid Resources and Their Communications

A Grid Resource is defined as a named computational element that operates in a Grid environment, has a defined interface, and provides service to other Grid Resources. The SGNP allows two Grid Resources, A and B, to name and securely communicate with one another even if:

- A or B moves (to another address, perhaps to a host with a different architecture)
- A or B is replicated
- A or B fails and is restarted, perhaps with another address
- A and B are on different architectures or operating systems
- A and B are in different organizations with different security policies
- There are trillions of endpoints, potentially moving about from place to place at a very high aggregate rate

"Communicate" means, at minimum, the ability to move byte vectors (messages) from one endpoint to another. A higher-level definition implies that, in addition to simply moving data from one endpoint to another, an action can be caused at the other end (a procedure call, for example).

"Securely" implies that identity and mutual authentication between endpoints can be established and attacks on the integrity of the data (snooping, replay, tampering) can be thwarted. This does not imply that mutual authentication between all pairs of Grid Resources is required.

All of the above must be accomplished in a highly scalable fashion (trillions of Grid Resources) in an environment in which the only constant is change, an arbitrarily large number of organizations will participate, and endpoints will be constantly created, destroyed, migrated, and replicated.

2.2 SGNP Names

The core idea driving SGNP is that *all Grid Resources* have an identity and that identity and security information should be indelibly linked to form a logical name. Further, this identity does not necessarily depend on trusted Certificate Authorities (CAs) or trusted third parties. The format of this security information is extensible. Currently-identified options include: (a) nothing, (b) RSA Public Key, (c) X.509 certificate, and (d) OpenPGP certificate (extended for use beyond just secure email).

An SGNP name is a Location-independent Object Identifier (LOID) that uniquely identifies a Grid Resource. The actual location (address) of a Grid Resource is determined by associating the LOID with one or more communication protocols and network endpoints. Collectively the set of communication protocols and network

endpoints for a Grid Resource is called its Binding. The SGNP describes the specific protocol by which LOIDs are resolved to Bindings.

3.0 Why Include Security Information in the LOID?

One of the principal challenges in building distributed Grids is resolving a set of issues related to trust in heterogeneous and possibly distrustful policy domains. Use of Public Key Infrastructure (PKI) X.509 certificates has been proposed as one way to address this challenge. But X.509-based PKI alone is difficult to scale (e.g., 50 000 users) and difficult operationally precisely *because* the approach presupposes a set of social and operational arrangements between Certificate Authorities (CAs)—arrangements of just the sort that would make Grids more difficult to create and administer. In addition, relying on the typical security protocol for X.509-based PKI (Transport Layer Security (TLS)) incurs a non-negligible run-time overhead that hurts performance in many situations. In general, without the SGNP, a client must first explicitly ask a Grid Resource for the enumeration of security protocols that the Grid Resource supports, a protocol must be chosen, the parameters of the chosen security protocol must be exchanged, and *then* the client must decide if the Grid Resource is trusted. The goal of SGNP is to flexibly support a mix of X.509-based and non-X.509-based, standards-based security protocols, lower the on-line cost of security protocols, and increase the overall security of emerging Grid and eBusiness applications. By doing so, SGNP supports the dynamic creation of user- and (virtual-)organization-centric trust relationships without being confined to the rigid requirements of CA-based distribution of public keys.

We incur different benefits based on the type of security information incorporated in Grid Resource names. If the client and Grid Resource both contain only an RSA public key in their names, then *as long as the client trusts from whom it received the name of the Grid Resource* (note that the possession of keys is orthogonal to the issue of trust), a client can immediately engage in a mutually-authenticated, confidential and/or integrity-checked dialogue with the Grid Resource:

- The client is assured of the authenticity of the Grid Resource because only the possessor of the private key (corresponding to the public key in the name of the Grid Resource) can decrypt the contents of a message to the Grid Resource (the message is encoded in the Grid Resource's public key)

- The Grid Resource is assured of the authenticity of the client because of the digital signature on the incoming message

We argue for the rapid adoption of the SGNP schemes in which some form of security information is included in the logical names. However, to ensure compatibility with current logical naming systems (and those under development), SGNP provides an option for *no* security information in the name. SGNP facilitates clients and Grid Resources immediately recognizing *without message exchange* when they cannot possibly meet security requirements (perhaps due to different supported security protocols and/or trust

relationships). In situations where this is not immediately the case, SGNP can enhance run-time performance, as described above.

A compelling argument can be made for how SGNP can enhance the efficiency, scalability, and security of authorization decisions, although the details of this complex discussion are beyond the scope of this document. SGNP also builds upon and augments the evidence-based security model of the .NET Framework and the security model of J2EE.

The remainder of this document describes an approach where the security information included in the LOID is an RSA Public Key. The SGNP reference implementation also uses that approach.

4.0 Requirements

The SGNP is guided by the following set of requirements, which are based upon observations of existing and proposed Grid systems:

4.1 Scalable Naming

The SGNP must scale to very large numbers of named Grid Resources distributed across a wide area network. The SGNP must not introduce bottlenecks into the creation of LOIDs, the association of LOIDs to Bindings, or the resolution of LOIDs to Bindings.

4.2 Scalable Secure Identity

The SGNP must scale to very large numbers of unique identities, at least one per Grid Resource. The ability to authenticate identities and associate them with policy-based access control must scale across many mutually distrustful security domains.

4.3 Scalable Administration

It must be possible to administer the naming and binding of a very large number of named Grid Resources. This administration should be simple even for very large Grids. The SGNP must allow multiple mutually distrustful organizations to administer portions of the namespace with local autonomy.

The SGNP must support evolving organizations, providing mechanisms for at least:

- The division of a name space into multiple administrative domains
- The merger of multiple administrative domains into a single domain
- The federation of multiple administrative domains

To satisfy these requirements, the SGNP must provide mechanisms to:

- Locally generate globally unique LOIDs

- Identify the administrative domain that issued the LOID

4.4 Location-, Migration-, Replication-, and Failure-Transparent Naming

The SGNP must provide names that are robust with respect to Grid Resource replication, migration, and failure. The SGNP must support at least the following cases:

- A Grid Resource is migrated to another address, perhaps supporting a different set of protocols
- A Grid Resource is replicated, and replicas are dynamically created and destroyed
- A Grid Resource is quiesced, fails, or is otherwise stopped and restarted, possibly with a different physical address

4.5 Support For Long-Lived Names

The SGNP name for a Grid Resource must not change for the lifetime of a Grid Resource. The lifetime may include events such as Grid Resource migration, Grid Resource reactivation, or organizational restructuring. In practice, the lifetime of a Grid Resource may be years.

4.6 Extensibility and Upgrade Path

SGNP names must be extensible to support future revisions of the SGNP. Newer implementations of the SGNP must be able to resolve legacy names.

5.0 Naming Overview

The SGNP is a two-layer naming scheme for obtaining Bindings to Grid Resources. For the purpose of this document, Grid Resources are named objects and services that have identity and can communicate over a network via some remote procedure call (RPC) mechanism. The SGNP requires that every Grid Resource must have a Location-independent Object Identifier (LOID) that identifies that Grid Resource. The LOID of a Grid Resource must be globally unique and immutable for the lifetime of that Grid Resource. The LOID of a Grid Resource can be mapped onto a lower-level Binding. A Binding is the current set of network endpoints and communication protocols that a Grid Resource supports. Unlike a LOID, a Binding may change over time.

The advantage of a two-level naming scheme is that it allows a Grid Resource to be migrated both spatially and temporally while preserving the identity of the Grid Resource. Consider the case where a Grid Resource X is running on server A, which is scheduled to be shut down for maintenance. X can be migrated to server B, and the binding associated with X can be updated to reflect its new location. The new running instance of X is not a new Grid Resource, but is instead a new incarnation of the same Grid Resource. (Two incarnations of a Grid Resource with the same LOID are considered to be the same Grid Resource.) In this way Grid Resources maintain their identity within

the system regardless of their physical location. Similarly a Grid Resource Y might become inactive and have no binding for an extended period of time. It should be possible to reactivate Y, possibly in a different location, while maintaining its original identity.

The SGNP incorporates the following elements, which are described in detail in the sections that follow:

- **Data Types**: LOID, Binding
- **Grid Resources**: Grid Naming Service, Resolver Hierarchy
- **Protocols**: Bind Protocol, Rebind Protocol

6.0 SGNP Data Types

The SGNP defines two data types: LOID and Binding. The LOID is the immutable name of a Grid Resource, and contains the Grid Resource's security information. A Binding is the current set of network endpoints and protocols that a given Grid Resource supports.

6.1 Location-independent Object Identifier (LOID)

A Location-independent Object Identifier (LOID) is a globally unique name for a Grid Resource. Using the Bind and Rebind protocols described in SGNP Protocols, the LOID of an object can be mapped onto the current Binding for the Grid Resource. The LOID of a Grid Resource is globally unique and immutable for the lifetime of the Grid Resource. Uniqueness is achieved by uniquely assigning a DomainResolverID to an administrative domain and by requiring that the tuple (BindingResolverID, ObjectID) be unique within that domain.

A simple mechanism for assigning the DomainResolverID field is to use the existing domain name infrastructure. For example, the organization Avaki Corporation owns the domain name avaki.com and could thus use "avaki.com" as the DomainResolverID in all LOIDs generated by the Avaki Corporation. The algorithm for assigning the other fields of a LOID is not specified and is left open to each organization. This structure supports the requirement of scalable administration, allowing each organization to generate names locally.

A LOID can be represented as a Uniform Resource Identifier (URI) with the following format:

LOID://<*LOIDType*>/<*DomainResolverID*>/<*BindingResolverID*>/<*ObjectID*>/<*SecurityInfo*>

The fields of the LOID URI are outlined in Table 1.

Field Name	Description
LOIDType	This field differentiates between different types of LOID and supports the requirement of extensible naming. This field accommodates future extensions to the naming protocol.
DomainResolverID	This field identifies the Domain Resolver that has ownership of the Grid Resource.
BindingResolverID	This field identifies the Binding Resolver within a Domain that has ownership of the Grid Resource Binding.
ObjectID	This field identifies a Grid Resource Binding within a Binding Resolver.
SecurityInfo	This field contains the security information for a Grid Resource (base 64 encoded). As discussed earlier, one approach is to use an RSA Public Key.

Table 1 Description of the LOID URI fields

6.2 Binding

A Binding is the collection of network endpoints and communication protocols that a Grid Resource currently supports. A Binding is easily represented as a Web Service Definition Language (WSDL) document. The protocols that the Grid Resource Supports are represented as WSDL <binding> tags. The network endpoints that support these protocols are represented as <port> tags.

It is important to note that the data contained in a Binding may become invalid. The Rebind Protocol allows a client to get an updated binding for a given Grid Resource.

7.0 SGNP Grid Resources

The SGNP defines a special set of Grid Resources: the Grid Naming Service, which provides client access to the SGNP naming services, and the Resolver Hierarchy, which maintains the authoritative LOID-to-Binding mappings. These SGNP Grid Resources will support SOAP over HTTP/HTTPS, as well as other language-specific bindings such as Java Remote Method Invocation (RMI). SGNP Grid Resources will allow access control based on the source LOID. In addition they may be replicated to provide for high availability and load balancing.

7.1 Grid Naming Service

Clients of the SGNP interact with it through an instance of Grid Naming Service (GNS). The GNS is a Grid Resource with a LOID and a well-known Binding. A client can use

this Grid Resource to obtain a Binding to another Grid Resource for which it knows the LOID. The GNS obtains the Binding by traversing the Resolver Hierarchy. This process is discussed in more detail in SGNP Protocols. An implementation of GNS provides at a minimum the following methods:

```
LOID addBinding(Binding newbinding)
Binding lookupBinding(LOID id)
Binding lookupBinding(LOID id, Binding oldbinding)
void updateBinding(Binding newbinding, LOID id)
```

As with any Grid Resource, the implementation of the GNS may require messages to be signed by the source LOID's private key, and use that signature as the basis for access control.

7.2 Resolver Hierarchy

Behind the GNS is a tree of Grid Resources known as Resolvers that resolve increasingly specific portions of LOIDs. Grid Resources in this tree are collectively known as the Resolver Hierarchy. Each level in the hierarchy represents a smaller administrative domain.

At the top of the hierarchy is the LOIDResolver. The LOIDResolver has a well-known Binding, and maintains the authoritative mappings from DomainResolverID to DomainResolver Bindings. Logically this represents the root of LOID space, and provides administrators with a single integration point for merging domains.

The next level in the Resolver Hierarchy contains the DomainResolvers. The DomainResolvers maintain the mapping of BindingResolverID to BindingResolver Bindings. Logically the Domain Resolver represents the portion of LOID space that is owned by a single administrative domain.

At the bottom of the Resolver Hierarchy are the BindingResolvers. The BindingResolver maintains the authoritative mappings from ObjectID to Grid Resource Binding. Logically the BindingResolver represents a single repository for Bindings. The class diagram in Figure 1 shows the relationship between the GNS and the Resolver Hierarchy. As shown in the diagram, a valid Resolver Hierarchy must have at least one resolver of each type.

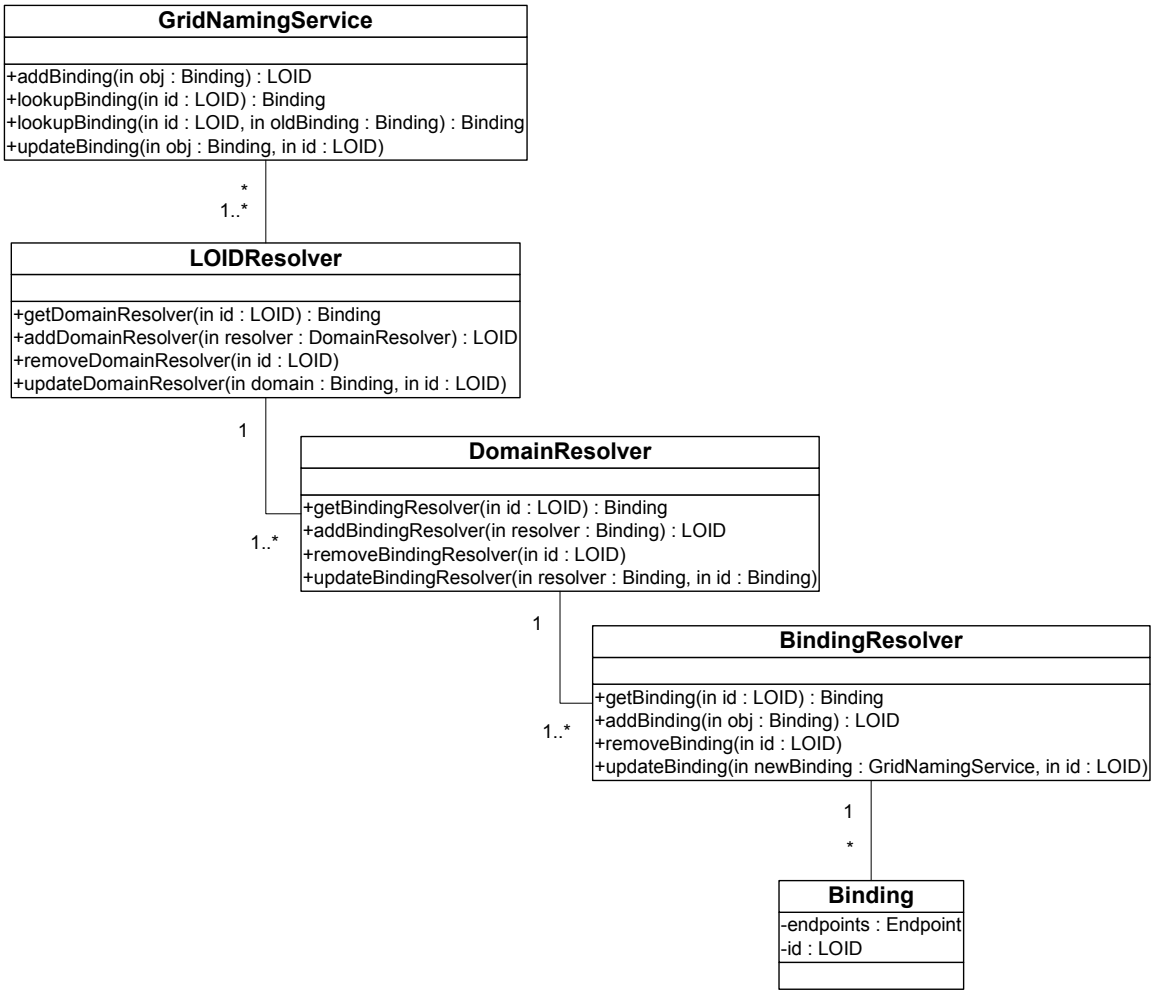

Figure 1 A UML class diagram of the Resolver Hierarchy

8.0 SGNP Protocols

The SGNP defines two protocols for mapping LOIDs to Bindings. The Bind Protocol provides a mechanism for obtaining a Binding to a Grid Resource given the Grid Resource's LOID. The Rebind Protocol provides a mechanism for obtaining an updated Binding given a Grid Resource's LOID and the old Binding. Clients of the SGNP interact only with the GNS, and are therefore insulated from the details of these protocols.

8.1 Bind Protocol

The Bind Protocol details the mechanism the GNS will use to resolve a LOID to a Binding. In the protocol the client makes a request to the GNS to resolve a particular LOID. The GNS then makes a call to the LOID Resolver to get the DomainResolver for the LOID. The GNS then makes a call to the DomainResolver to get the BindingResolver

for the LOID. Finally, the GNS calls the BindingResolver to get the Binding for the Grid Resource.

To improve performance, and to remove contention on the higher levels of the Resolver Hierarchy, the GNS may cache the Bindings to Grid Resources, including the Bindings of DomainResolvers and BindingResolvers. This allows the GNS to route requests for Grid Resources managed by the same BindingResolver directly to that BindingResolver without walking the upper portion of the Resolver Hierarchy. An example of this type of caching is shown in Figure 2.

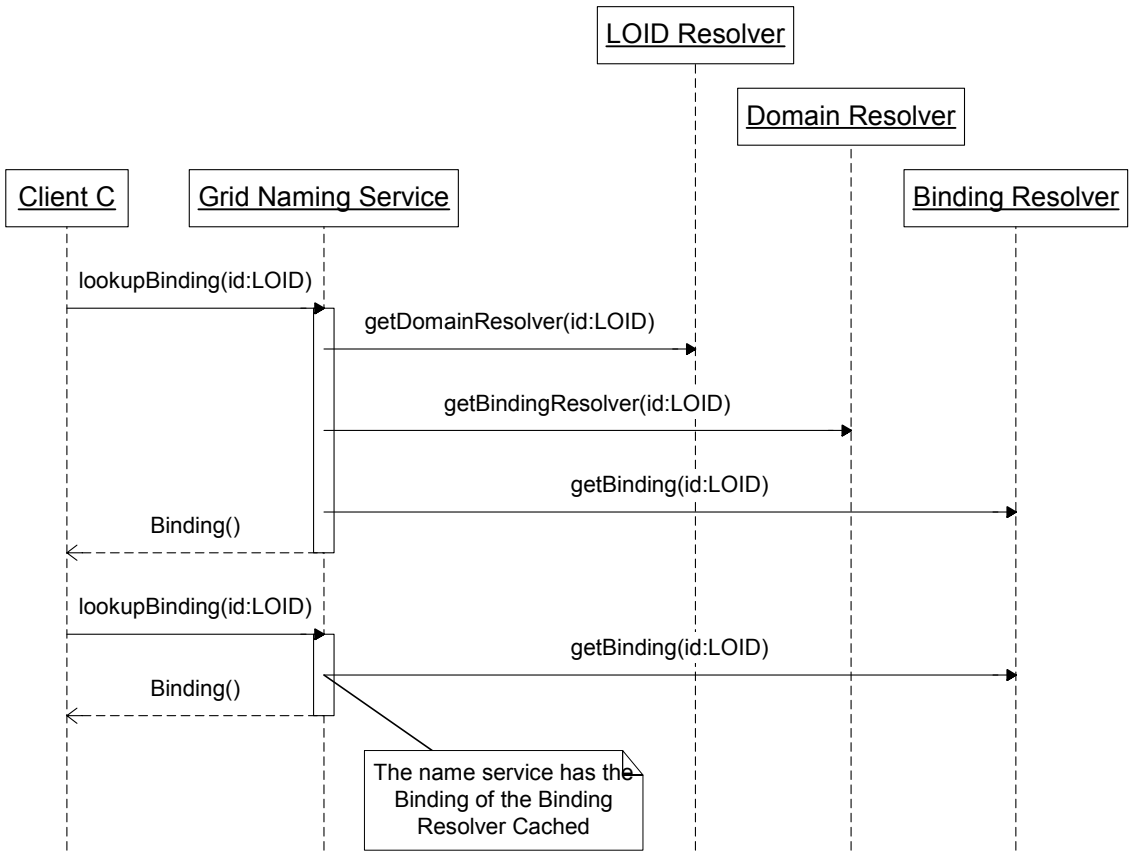

Figure 2 A UML sequence diagram of the Bind Protocol showing caching of the BindingResolver Binding

8.2 Rebind Protocol

The Rebind Protocol provides a mechanism for a client to obtain a new Binding to a Grid Resource in the event that the client's current Binding to the Grid Resource is invalid. For example, a client may have a Binding to a Grid Resource that was migrated after the Binding was obtained. Figure 3 is a sequence diagram detailing the Rebind Protocol. Note that in the Rebind Protocol, the client passes the old Binding to the GNS. This allows the GNS to issue an updated Binding from its cache without necessarily contacting the authoritative BindingRevolver for the Grid Resource.

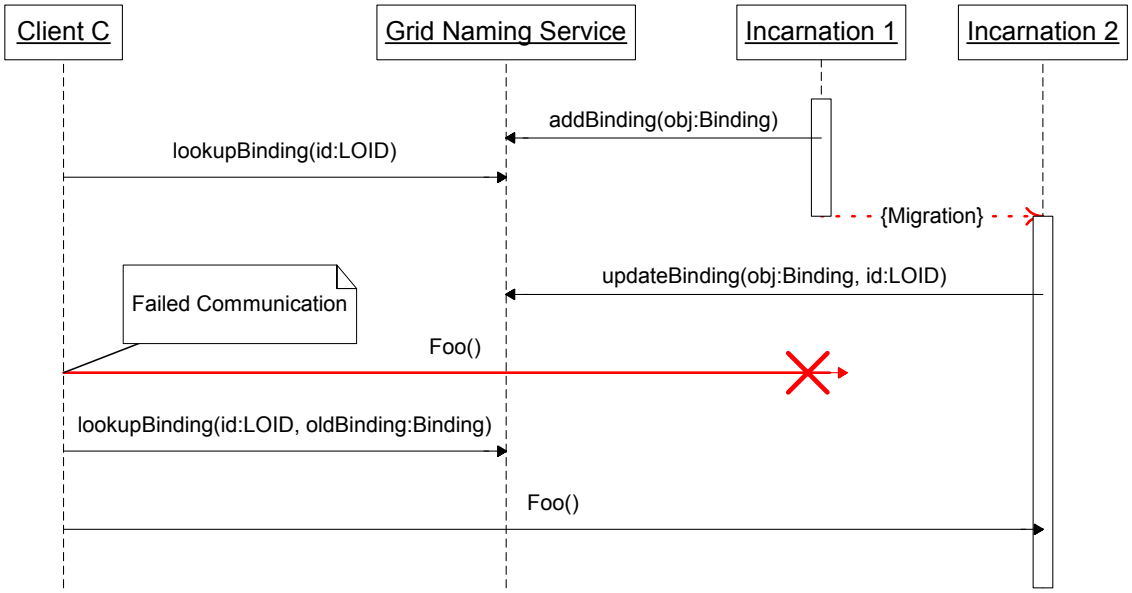

Figure 3 UML sequence diagram of the Rebind Protocol showing rebinding after object migration

8.3 Relationship to Communications Protocols

Although it is beyond the scope of this document to describe a communications protocol in detail, we expect that communications protocols built on top of SGNP will provide access that is transparent with respect to location, migration, replication, and failure. One way this could be achieved is for the communications protocol to provide a stub that wraps a Binding and provides automatic rebind semantics.

For example, a client C wishes to be able access application X even if X moves from place to place, fails, or is replicated in some way. The communication protocol could provide the client with a stub that knows how to bind and rebind to X. In the event that X shuts down and is reactivated on another host, the stub will detect that X is no longer in its old location, make a rebind call to the GNS to get a new binding, and reissue the call to the new location, potentially over a new protocol. In this way a communications wrapper layer can use the SGNP protocols to mask the failure, and provide the client with uninterrupted service.

In the SGNP reference implementation, we use JAVA Proxies to insert an automatic rebind layer between the client code and the RMI stub.

9.0 Conclusion

The Secure Grid Naming Protocol (SGNP) introduces a straightforward approach to identifying resources that participate in Grid systems. The SGNP addresses the stringent requirements of large-scale, dynamic Grid systems—requirements that have not thus far been addressed by other naming approaches. Most importantly, SGNP does not introduce the need for a central trusted authority, thus enabling the creation of scalable Grid systems and facilitating a variety of practical models for Grid administration.

The SGNP and the set of requirements that motivate it are informed by extensive practical experience building Grids in both academic and commercial environments. We believe that implementation of a protocol such as SGNP will accelerate the adoption of Grid technology by enabling Grids to conform and adapt to diverse practical requirements.

10.0 Author Contact Information

Joshua Apgar (Primary author)
japgar@avaki.com
617-374-2518

Andrew Grimshaw
agrimshaw@avaki.com
617-374-2508

Steven Harris
sharris@avaki.com
617-374-2549

Marty Humphrey
humphrey@cs.virginia.edu
(434) 982-2258

Anh Nguyen-Tuong
anh@avaki.com
434-951-0158

11.0 Copyright Notice

Endnotes

---

1 Andrew S. Grimshaw and William A. Wulf, "The Legion Vision of a Worldwide Virtual Machine", Communications of the ACM, 40(1):39-45, January 1997.

2 Ian Foster and Carl Kesselman (editors), "The Grid: Blueprint for a New Computing Infrastructure", Morgan Kaufmann Publishers, November 1998.

3 While some readers may initially view "trillions" as an absurdly large number, one should consider the burgeoning scale of devices and related entities involved in – or eligible to participate in – distributed systems: There exist approximately half a billion computers in the world (not counting devices such as PDAs, cell phones, and pagers); Web pages currently number in the billions; and even within a given enterprise, a single *database* may contain trillions of records.

Note:

Marty Humphrey was supported in part by the Next Generation Software program of the National Science Foundation under grant EIA-9974968, the NPACI program of the National Science Foundation, and by the NASA Information Power Grid program.