THEBES: THE GRID MIDDLEWARE PROJECT

Project Overview, Status Report and Roadmap

Arnie Miles
Georgetown University
adm35@georgetown.edu
http://thebes.arc.georgetown.edu

The Thebes middleware project was created by Georgetown University's Advanced Research Computing team with seed funding from Sun Microsystems to build a community of interested parties around the design and creation of the grid. Some core concepts are scalability, security, accountability, backwards compatibility, and the creation of the working grid from the grassroots. The project has no interest in creating yet another limited scope experimental grid. The project reaches out to underserved organizations, smaller schools, corporations, etc.

THE GRID IS DEAD (LONG LIVE THE GRID!)

- Term falling into discredit
- Misuse by Corporate marketing departments, including multiple contradictory definitions
- Failing to live up to expectations
  - Built for specific community
  - Unscalable
  - Expansion efforts only make the situation more

Current distributed computational solutions were designed for specific target user groups, and specifically solves requirements of that user group. Attempts to expand existing solutions to solve more generalized problems lack scalability and extending existing solutions adds complexity while away accounting data.

The Anatomy of the Grid

“Grid computing does not imply unrestricted access to resources.”

Ian Foster
Carl Kesselman
Steven Tuecke

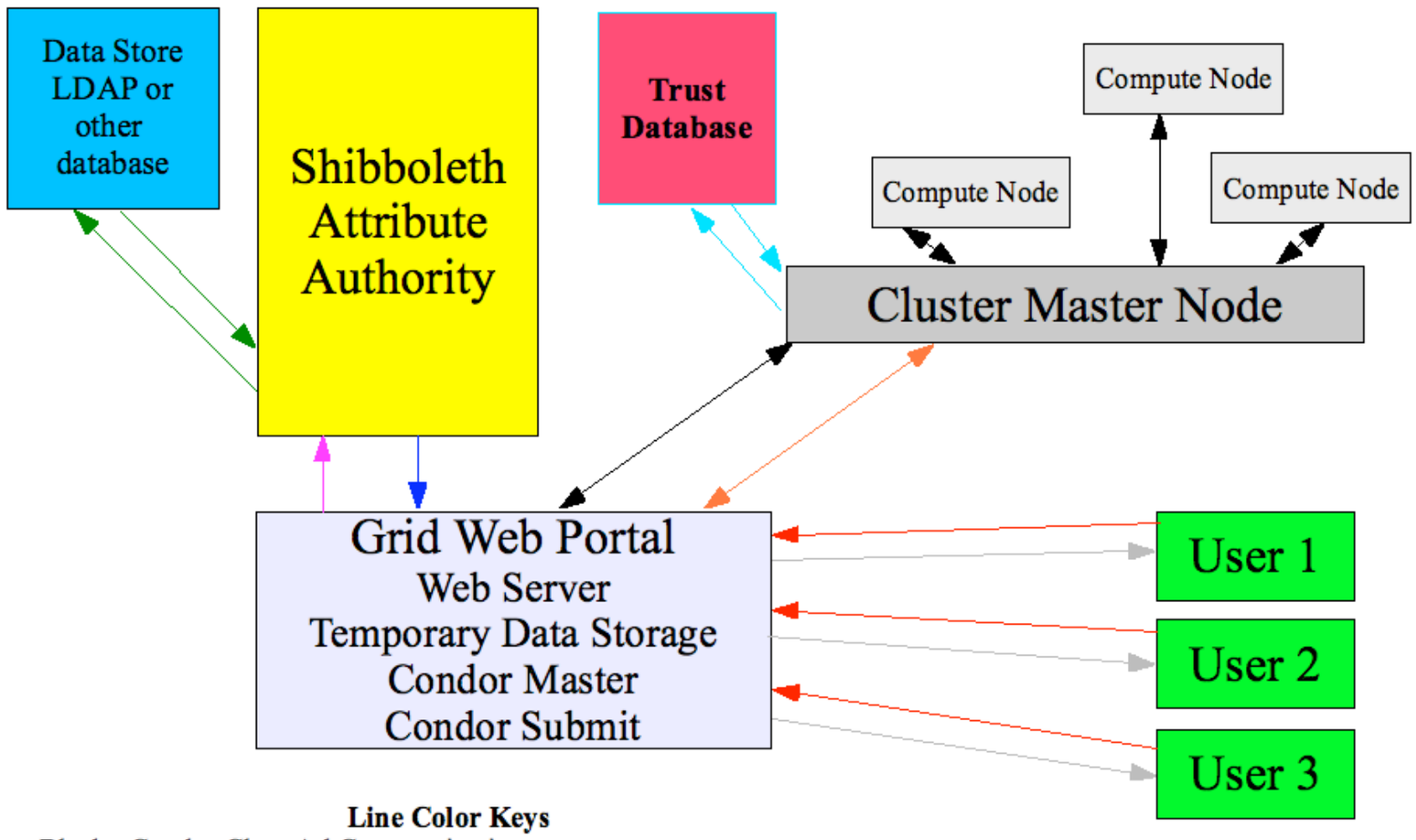
The grid requires authentication, authorization, and accounting. This is even becoming true in the academic space as funding agencies begin to demand an accounting of how money and resources are being consumed, but it has always been and will always be true in the commercial space. This cannot come at the expense of ease of use for either the consumer or the supplier of computational resources. Until now, this has been restricted to x.509 solutions.

THE PROBLEM

- One size does not fit all.
- Current x.509 based solutions are excellent in sole cases to build “a” grid, but even with more and more middleware stacked on top, will never be the solution that provides “the” grid.

We first ran into the scalability issues of current grid solutions when we tried to share several hundred processors broken up into 5 clusters and 5 administrative domains between researchers. Globus and Condor-G implied there was a solution, but it was immediately clear that this solution was not scalable, nor was it production ready. One of the primary problems with grid technology today is that a solution that very nicely solved one community’s problems is being forced into service for all communities.

Condor-Shib: A Successful Test



Line Color Keys

Black: Condor Class Ad Communications
Red: Requests for Services from Web Portal
Magenta: Attribute Query
Green Shib Authority queries for information about the user
Blue: Attribute Response
Orange: Job submitted to compute node(s) and data returned to Portal
Light Blue: Check signature against Trust File
Grey: Data stored on Portal until retrieved by User

Our test case came to be known as Condor-Shib, as it was a merger between the Condor Job Scheduler and Shibboleth. In the Condor-Shib example, the portal also acts as the interface to Shibboleth. Users makes requests of the portals, which then asks the users to select a home institution from a drop down list provided by a “where are you from” server. The users is passed off to their home institution, where they log in as they normally would. The attribute authority sends a token back to the portal, which is embedded in the job submission. The portal contains a Condor Master, which acts as the secure entry point to any Condor cluster. The job is submitted to any pool of resources the Condor Master is aware of, the Cluster Master verifies that it trusts the source, and work is done.

The Ugly

What's Missing??

- After the excitement was over, we asked what's missing?
  - The final product cannot be specific to any one scheduler.
  - The final product must be pluggable into any grid technology, not just computational.
    - License management
    - Applications
    - Storage
    - ?
  - Shibboleth is web based only, while SAML 2.0 allows for command line and application support.
  - The product must support metascheduling.
  - Access must be transparent to the users.

http://thebes.arc.georgetown.edu/node/9

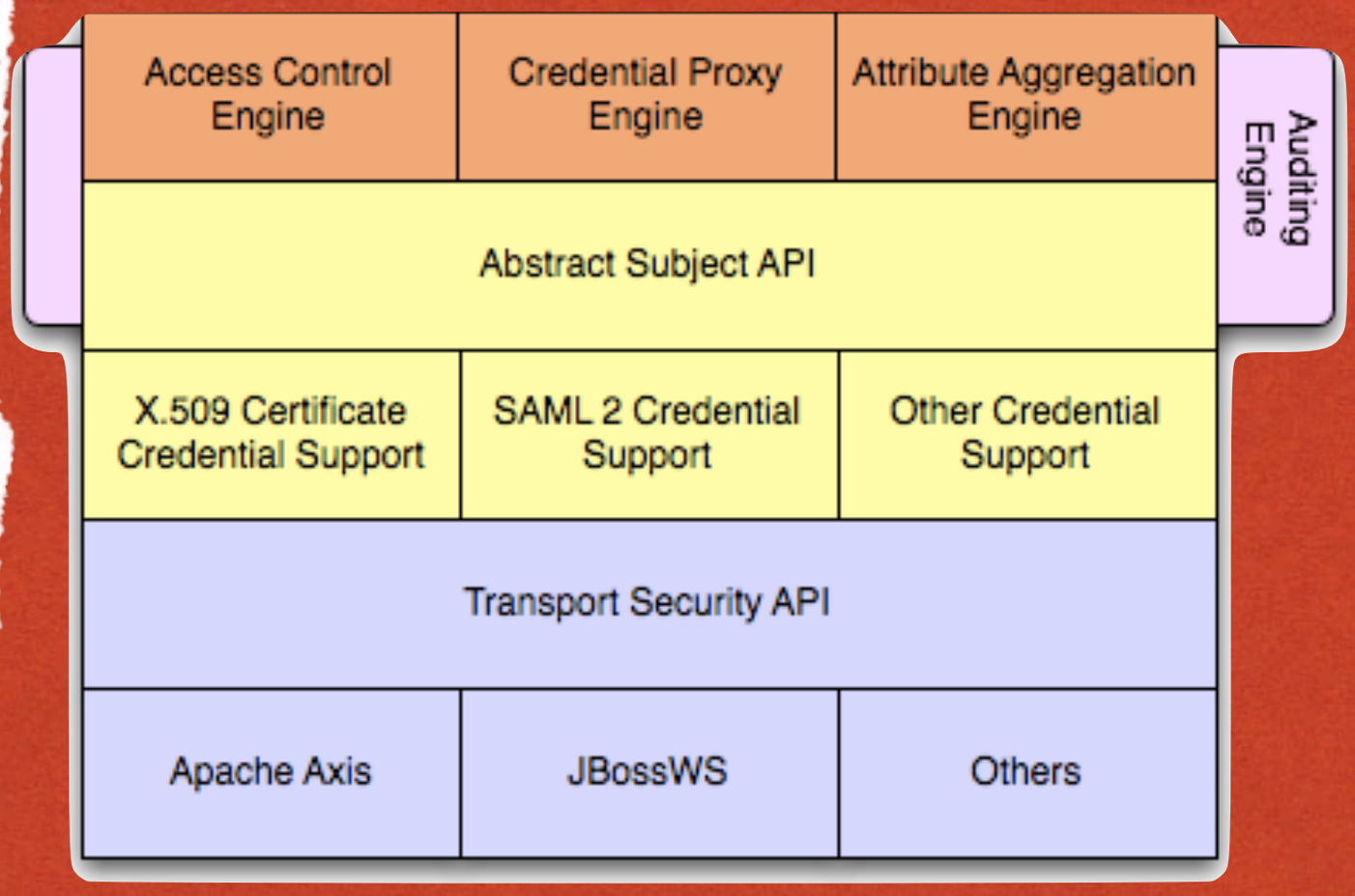
So, we were pretty pleased with our initial trial, but realized even then that much was missing. We proved that SAML in the form of Shibboleth can provide a single sign-on method for accessing and sharing attributes about users to resources, and that Condor can consume the attributes provided by Shibboleth. But this is far from being the next grid, there was too much missing.

THEBES: PROVIDING 100(+) GATES

- Homer describes it as having 100 gates and he must have been told that by Phoenicians who traded there, since he himself was never there and he never spoke of the rest of Egypt. He also told us that out of each of the gates, Thebes could send 200 chariots to oppose an enemy which was an exaggeration he must have gotten from the same people.
  - http://mysaga.net/regboard/viewtopic.php?t=52

Thebes grew out of the work done in Condor-Shib. We argue that current one-size-fits-all grid solutions are fine for creating “a grid,” but “a grid” is no longer very interesting. To accomplish this, we intend to use what works, retrofit what can be salvaged, and create what is missing.

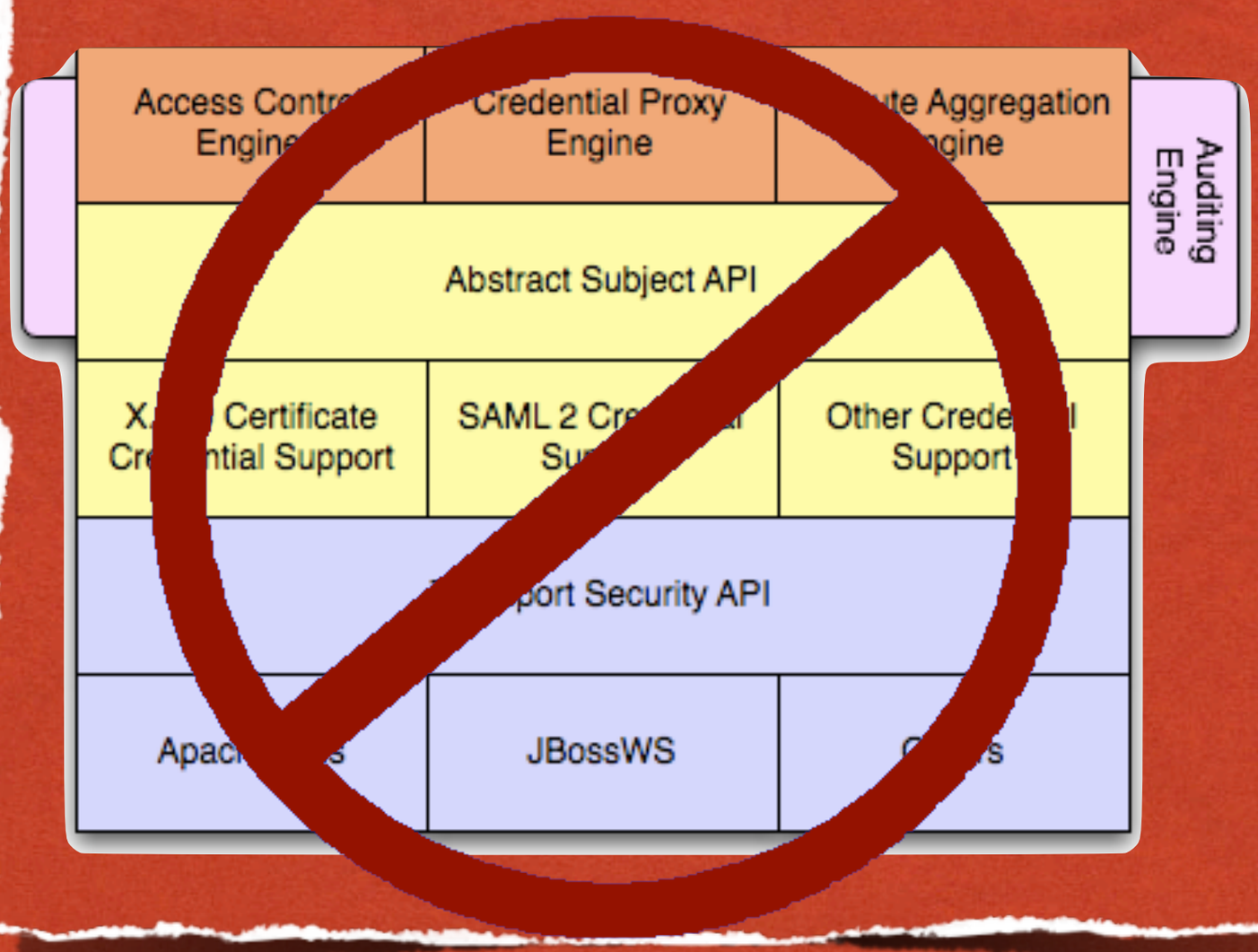
Sun Microsystems has generously invested in the creation of this consortium of interested parties. It is in our interest to get others interested in grid computing to participate.
Success in this project will further growth grids, identity management, high performance clusters, and cycle scavenging mechanisms.



FIRST PASS: AN EXPANDED VERSION OF THE GSI

Current GSI solutions leverage various X.509 specific solutions, where all entities on the grid must have a certificate, or belong to a virtual organization that aggregates users and provides said certificates on their behalf. We argue that one size does not fit all, and proposed a new GSI that will continue to allow x.509 certificates, but added, at a minimum, SAML 2.0 credential support. The policy grid service would be built upon the new GSI library and leverage its abstract subject interface and policy evaluation hooks.

Then came SWITCH and the Swiss Grid contributions to EGEE.



FIRST PASS: AN EXPANDED VERSION OF THE GSI

Current GSI solutions leverage various X.509 specific solutions, where all entities on the grid must have a certificate, or belong to a virtual organization that aggregates users and provides said certificates on their behalf. We argue that one size does not fit all, and proposed a new GSI that will continue to allow x.509 certificates, but added, at a minimum, SAML 2.0 credential support. The policy grid service would be built upon the new GSI library and leverage its abstract subject interface and policy evaluation hooks.

Then came SWITCH and the Swiss Grid contributions to EGEE.

Security Token Service

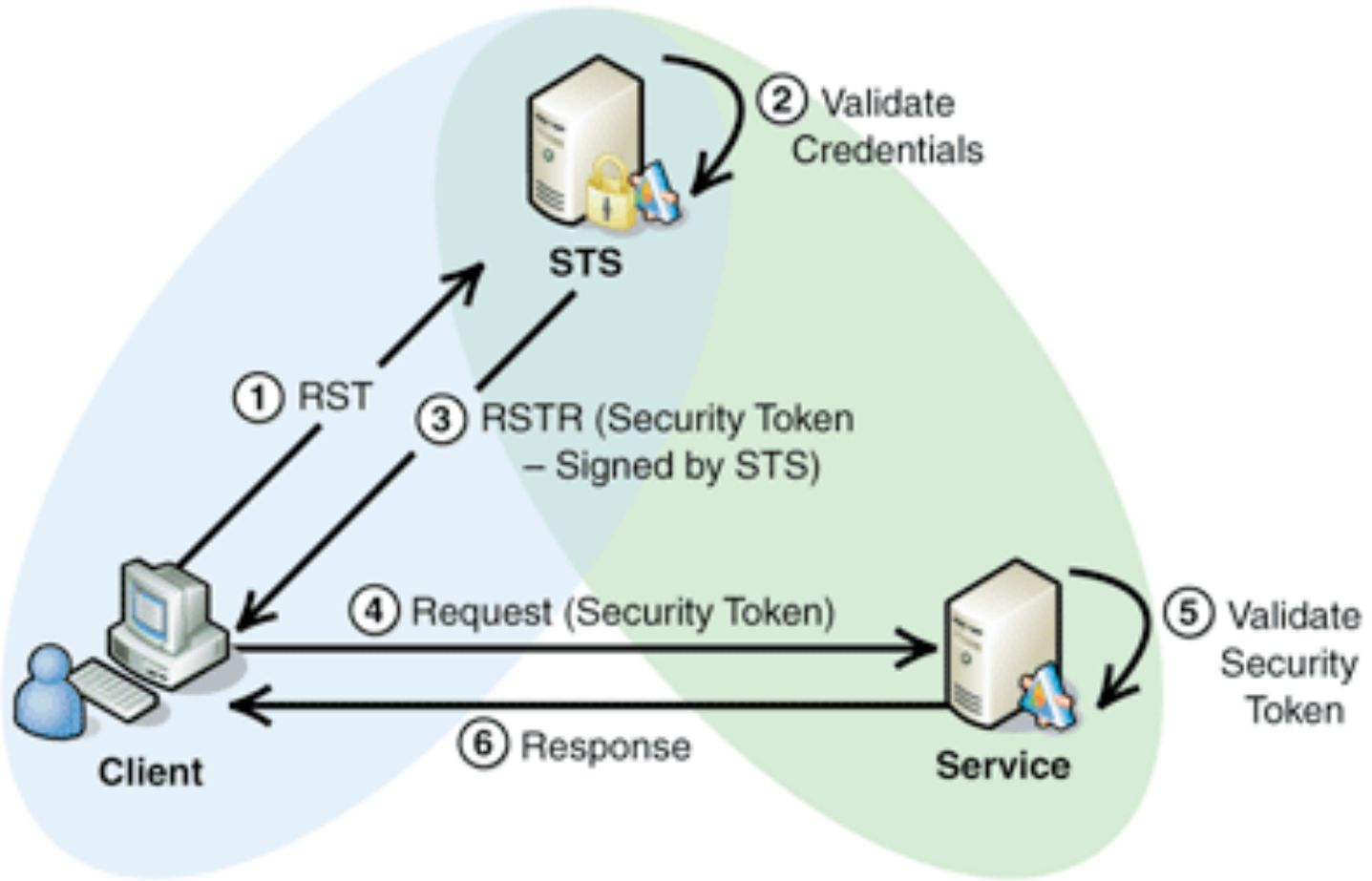
Valéry Tschopp - SWITCH

SWITCH Middleware Meeting

The next few slides are used by permission from SWITCH, and discuss the Security Token Service. It is no coincidence that SWITCH's work is consistent with THEBES, as one of the principal authors of the original white paper describing Thebes was Chad La Joie, who was hired by SWITCH in part to implement some of his ideas. This presentation on the Security Toekn Service was given by Valery Tschopp at a SWITCH Middleware Meeting

Security Token Service

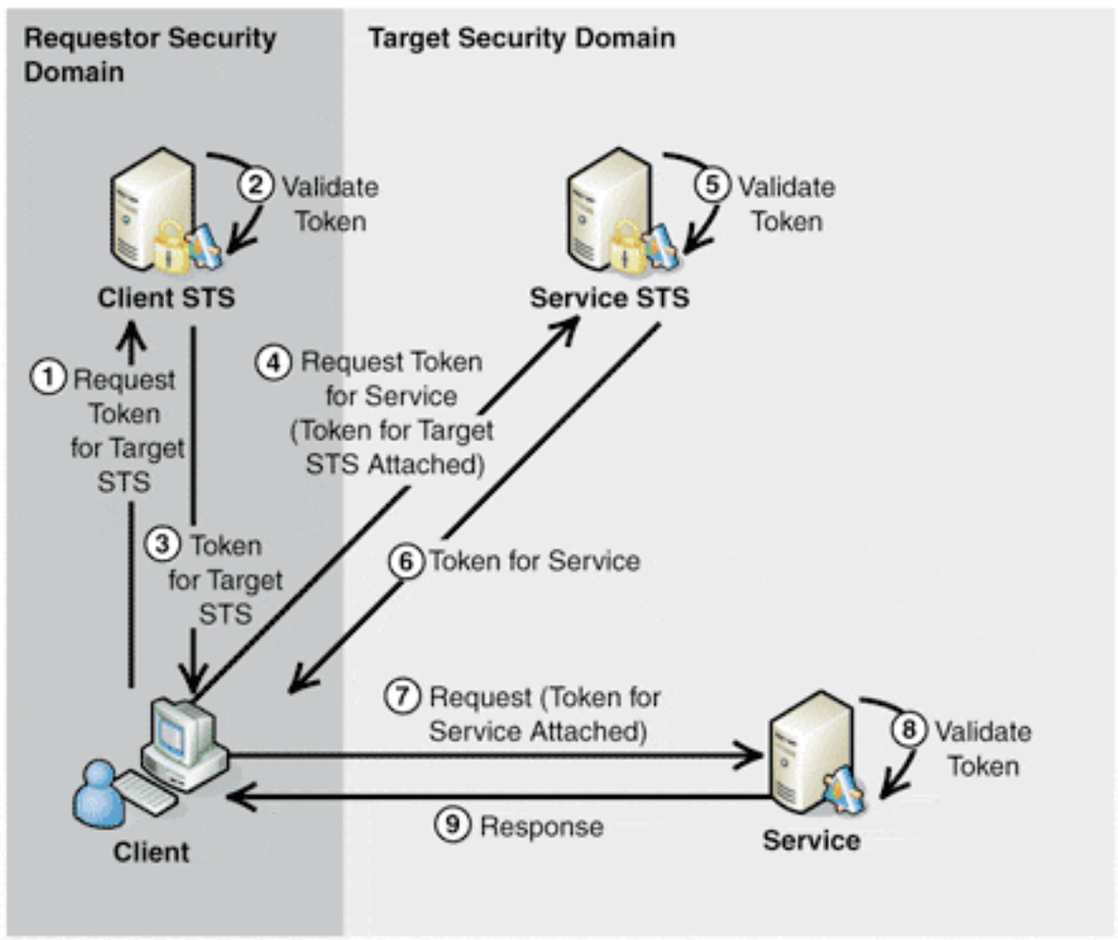
- WS-Trust defines mechanisms for brokering trust to an authority called Security Token Service (STS)
- The Security Token Service have a trust relationship with both the client and the service.



SWITCH has taken an improved approach to solve the problem of connecting users with one set of credentials to resources that have one of a number of different credential requirements. This is called the Security Token Service. In this example, the user has access to credentials (e.g. SAML) and wants to access a service that requires something different (e.g. x.509). The users approaches the STS with SAML credentials and requests x.509 credentials, which are then provided to the resource. The client and the service both have a trust relationship with the STS, and the client needs to know what sort of credential the service needs.

Multiple Security Domains

- **A client may need to communicate with services that operate across trust boundaries**
  - Shibboleth SAML vs Grid X.509
- **Multiple STS can be used in a trust chain across security domains (delegated trust)**

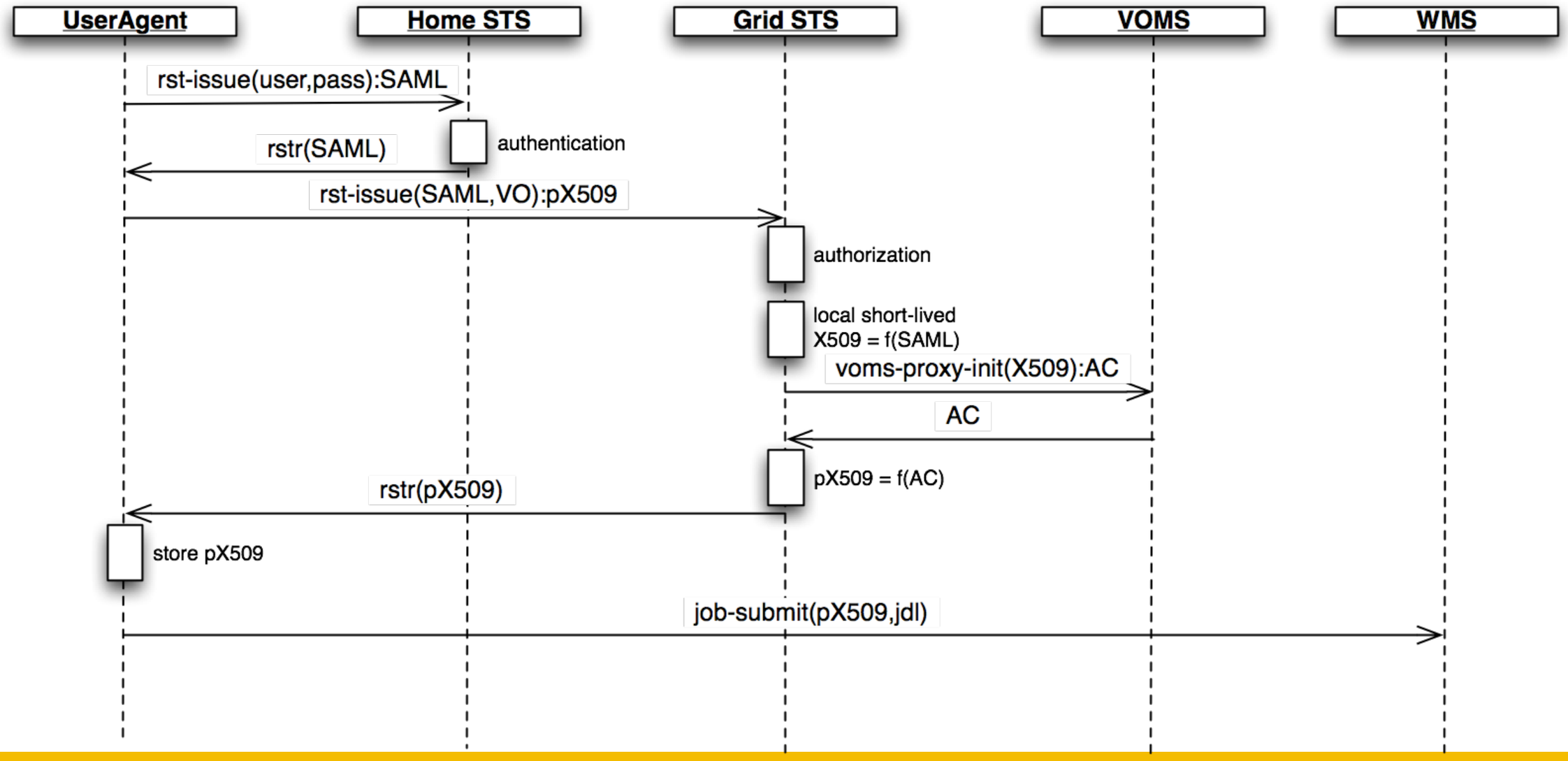


Communications across trust boundaries are done with multiple STS's configured in trust chains across security domains. The STS's trust each other, so the user can use their client STS supplied credentials to present to the Service STS to obtain credentials the Service will accept.

All this will require that credential requirements be published by resources.

Scenario: Issue a proxy X.509

- User authenticates with his credential to a Shibboleth IdP STS and receives a SAML security token
- User requests a proxy X.509 from a Grid STS using the SAML token



Friday, May 9, 2008 12

Here we see the user authenticating with his local credentials to a Home STS that accepts Shibboleth, and the user receives a SAML security token. The user uses this token to request an x.509 proxy from the Grid STS, which consumes the attributes it needs from the SAML token to make a voms-proxy-init call. When the user receives this x.509 proxy, jobs can be submitted to the service.

A PATH TO METASCHEDULING

- To coordinate resources under the control of different job schedulers, a metascheduler will require a harmonization of resource descriptions.
- A Resource Discovery Network will collect traditional static data about resources as well as the dynamic data made available by a standardized Resource Description Language.

With that problems in the hands of professionals, we have the opportunity to deal with other issues. Two that must be resolved simultaneously are the harmonization of a resource discovery language and replacing the single-point-of-failure Grid Indexing Services with something more robust.

STEPS DOWN THE PATH

- Work with leaders of job scheduling community and Open Grid Forum standardization to harmonize resource description to a single standard language.
- Simultaneously work on a Resource Discovery Network that is able to collect static data from any existing grid indexing service client.

I have been talking with the leaders of some of the most popular job scheduling packages out there, who represent the most experienced minds in this field. It seems that there have been multiple attempts through OGF to do this harmonization work, none of which have been all that successful. We cannot fail at this.

The various grid index service clients are pretty easy to extract data from, so while one team concentrates on working with the experts in job scheduling to harmonize the language that describes resources, including credential requirements and policy publication, another team will work on creating a resource discovery network that can consume the existing static data from grid index clients.

AND THAT BRINGS US TO

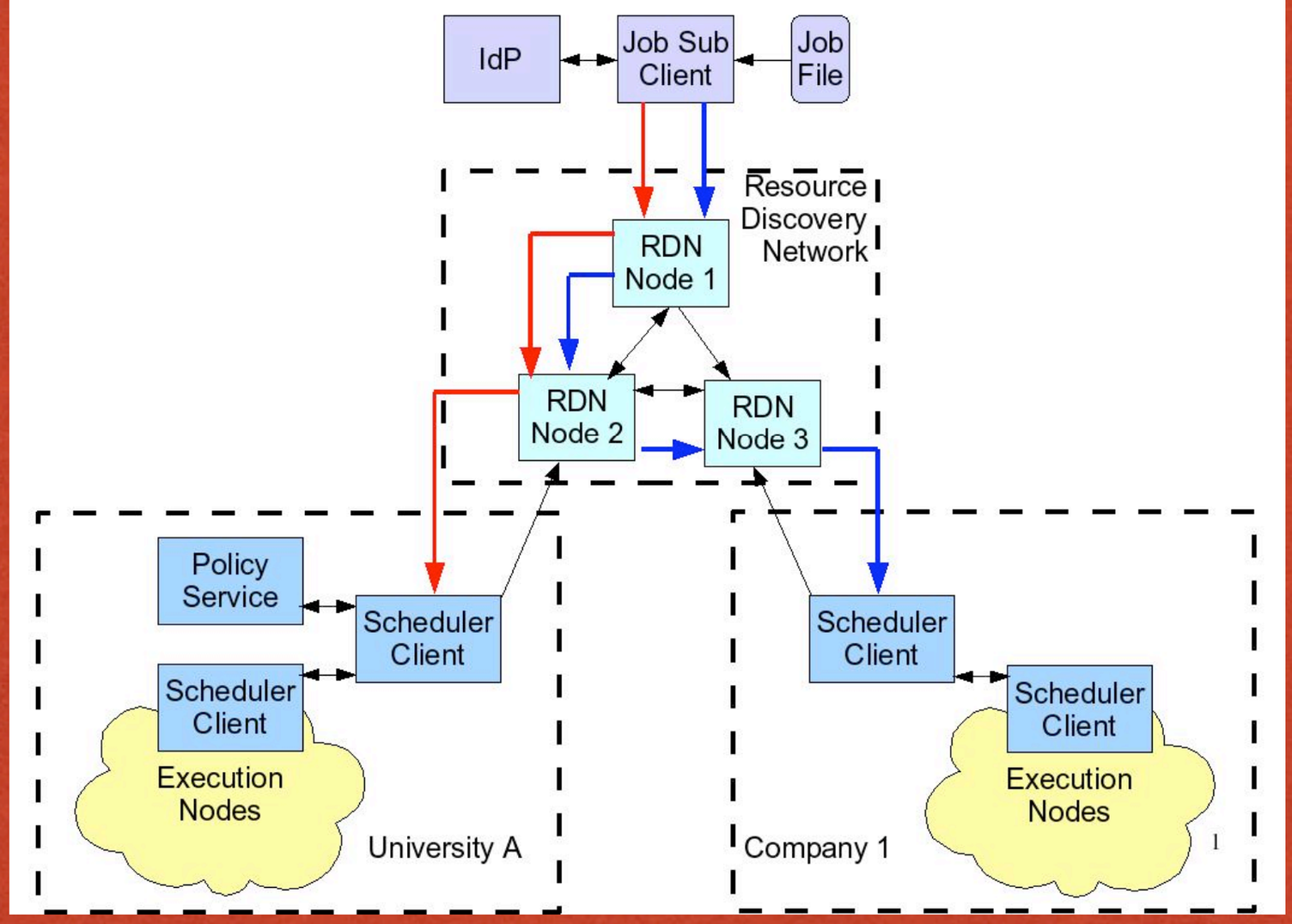
- With these two steps in place, the next logical step will be to use the new RDL to expand the new connections between existing grid index clients and the Resource Discovery Network to include dynamic information published by the local job schedulers.
- Users will know how to express their jobs so any appropriate resource can accept them.

Once we have the grid indexing service clients reporting to a resource discovery network, and have created a robust standardized resource description language, we can use these tools to begin publishing resource data in a useful, real-time fashion, which will expose it to consumption by a metascheduler. All static and dynamic data about a resource, including policies and credential requirements can be made available to the user via this RDN and metascheduler.

THE METASCHEDULER

- Resource Description Language in Place
- Resource Discovery Network receiving static, dynamic, credential, and policy data from Resources
- Metascheduler to connect jobs to resources becomes possible.

Once the disparate pieces are constructed, we can move towards a metascheduler for the grid. The metascheduler should be able to consume everything published to the Resource Discovery Network and use this data to match work to services in the same fashion that cluster level schedulers match. If the RDL is properly harmonized, the local resources should already be aware of how to consume this scheduling information.



RESOURCE DISCOVERY NETWORK

This very early drawing still does a pretty good job of depicting a Resource Discovery Network. This framework of peer-to-peer (and perhaps some heirarchical) nodes will be used to publish client needs and service availability. Adding

THANKS TO

- Sun Microsystems
- SWITCH
- Kansas City Federal Reserve Bank
- UCLA Grid Portal team
- The Condor team.
- All members of Thebes Consortium

CONTACT INFORMATION

- http://thebes.arc.georgetown.edu
- Join: thebes-l@georgetown.edu (signing up for the portal will get you on the mailing list)
- My contact information: adm35@georgetown.edu

Thebes is a community to design and build the grid. A community is only as good as it's members. So please join and participate.

QUESTIONS?

Arlo the Mastiff knows all and tells all. I, on the other hand, will do my best.