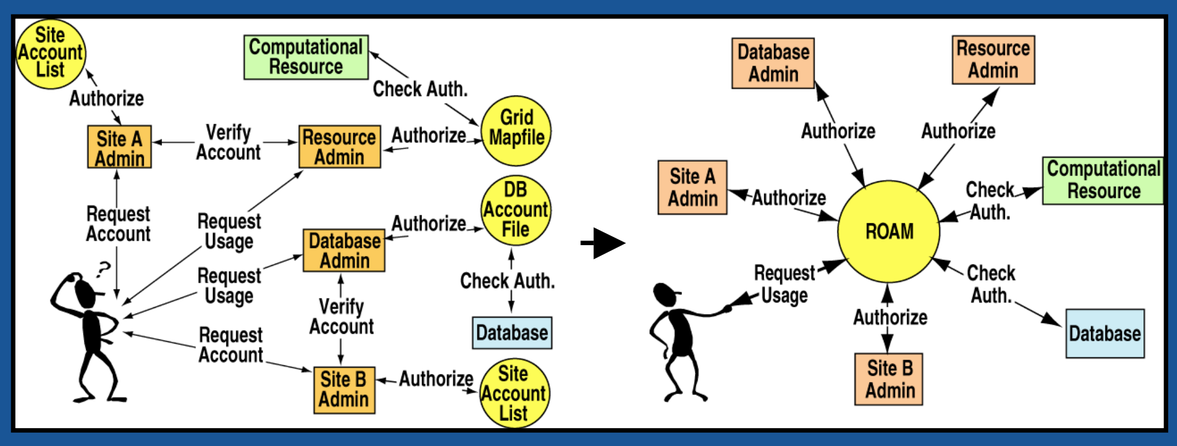
# THE RESOURCE-ORIENTED AUTHORIZATION MANAGER (ROAM)

J. R. BURRUSS[1], T. W. FREDIAN[2], M. R. THOMPSON[3]
[1]GENERAL ATOMICS
[2]MASSACHUSETTS INSTITUTE OF TECHNOLOGY
[3]LAWRENCE BERKELEY NATIONAL LABORATORY

## Summary


- ROAM was created to provide a simple but flexible authorization system for the National Fusion Grid (FusionGrid)
- ROAM is for authorization management in administratively separated organizations
- Allows resource stakeholders to control access to their resources

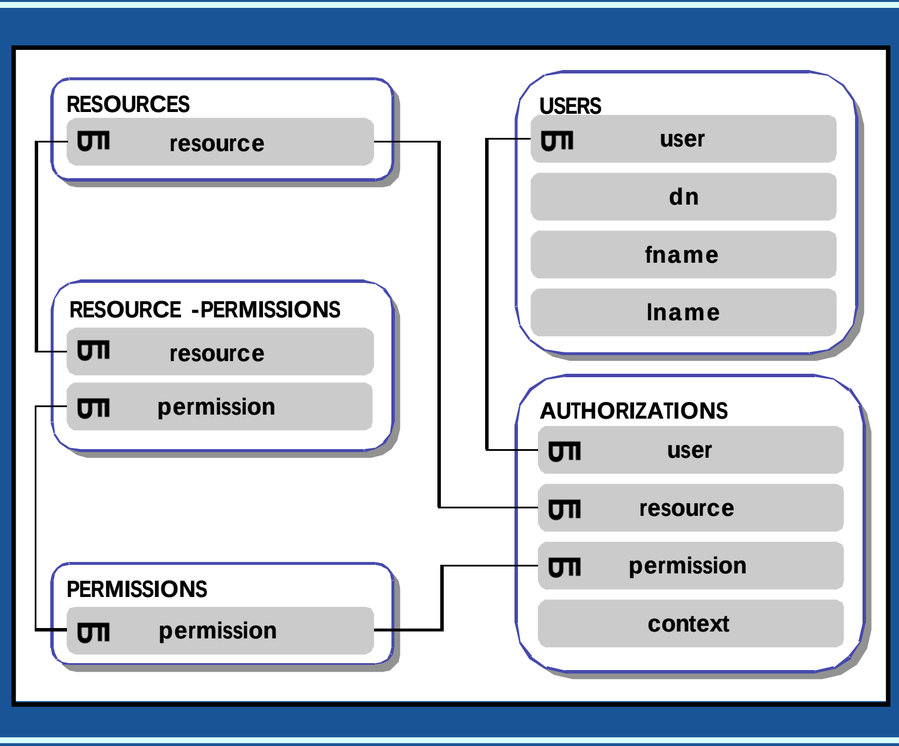




*Before & after: ROAM simplified the process of requesting authorization to use FusionGrid resources*

- Works with GSI & GRAM
  - –X.509 certificates
- Can do account mapping
  - –grid-mapfiles no longer needed

## Data Model

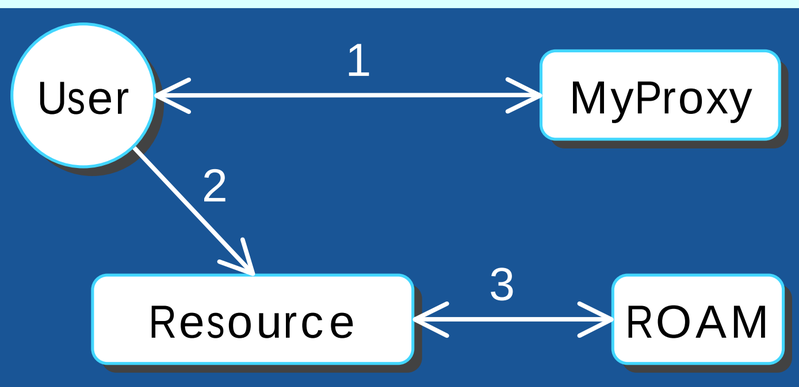
- Focus is on the resources of FusionGrid
- Resources include anything where access control is required
  - –codes
  - –data
  - –entire sites
- If you have to sign a form to use it, then you can probably model it as a resource
- Each resource has a set of valid permissions
  - –e.g. “execute”, “access”, “admin”
- Users have a unique FusionGrid username
  - –used to get MyProxy credentials
- An authorization is the grant of a permission for a resource to a user
  - –binary in nature
- Authorization has an optional context
  - –typically used for account/group mapping



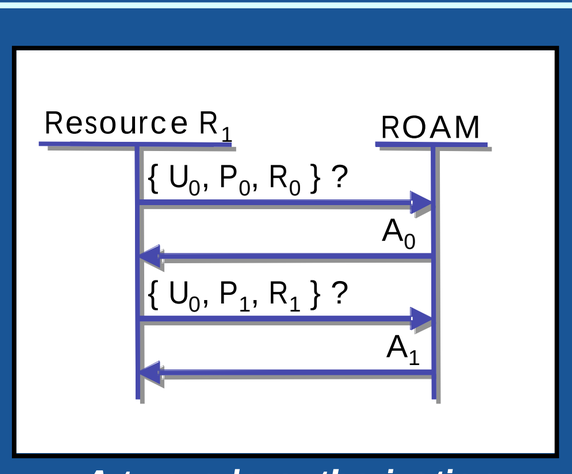

## Architecture

- ROAM avoids the “push” model of authorization used by other authorization systems
- Instead, resources pull authorization information from ROAM
- No special tokens required

**Resources query ROAM for authorization information**

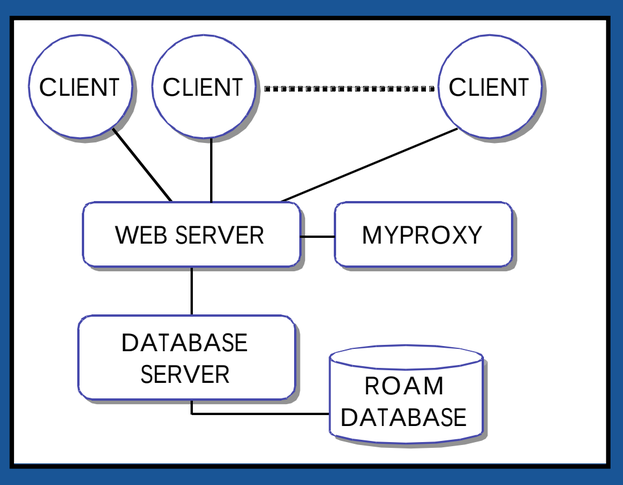



1. User “signs in” to FusionGrid as normal
2. Uses resource as normal
3. Resource queries ROAM to check authorization




*A two-rule authorization check requires two queries*

- Authorizations are binary in nature, and so are queries
- Typical FusionGrid authorization policy has two rules
  - – “execute” permission on code plus “access” permission on site
- Can handle more than two
- GRAM callout used for gatekeeper authorization checking
- Queries go through web intermediary
  - –HTTPS or HTTP




*ROAM consists of a web front end and a database back end*

## Interface

- Users and administrators request and grant permissions through a web page
- Either use a certificate in your browser, or enter MyProxy username & password
- View log of queries
- Do account mapping

**Welcome Justin Burruss to the FusionGrid Resource Authorization System**

You are currently authorized for the following resources:

| Resource | Permission | Description | Action | Logs |
|---|---|---|---|---|
| CMOD Data | Write | Access to CMOD experiment data | Submit Inquiry | Show log |
| CMOD Data | Admin | Access to CMOD experiment data | Administer Resource | Show log |
| CMOD-Jobs | Admin | Permission to execute CMOD-Jobs | Administer Resource | Show log |
| DIIID-MDSplus | Read | MDSplus at GA | Submit Inquiry | Show log |
| DIIID-MDSplus | Admin | MDSplus at GA | Administer Resource | Show log |

*Users and administrators interact with ROAM through a web interface*

## Conclusion

- The ROAM data model brought coherence to FusionGrid authorization
- The web interface proved to be easier for users and administrators
- Improved upon grid-mapfiles
  - – quickly change mappings for multiple computers
  - – change mappings on per use-case basis
  - – no typos (web interface)
- Politically easier to adopt than other centralized authorization systems because it’s “consultative”
  - – ask ROAM for authorization information
  - – let resource make the actual decision
- Leaves room for innovation by service developers
  - – example: interesting uses of context field
- It works and is being used

This work supported by the SciDAC Project, U.S. Department of Energy, Office of Advanced Scientific Computing Research and the Office of Fusion Energy Sciences.

http://web.gat.com/~burruss
burruss@fusion.gat.com