Dynamic Accounts: Identity Management for Site Operations

Kate Keahey

R. Ananthakrishnan,

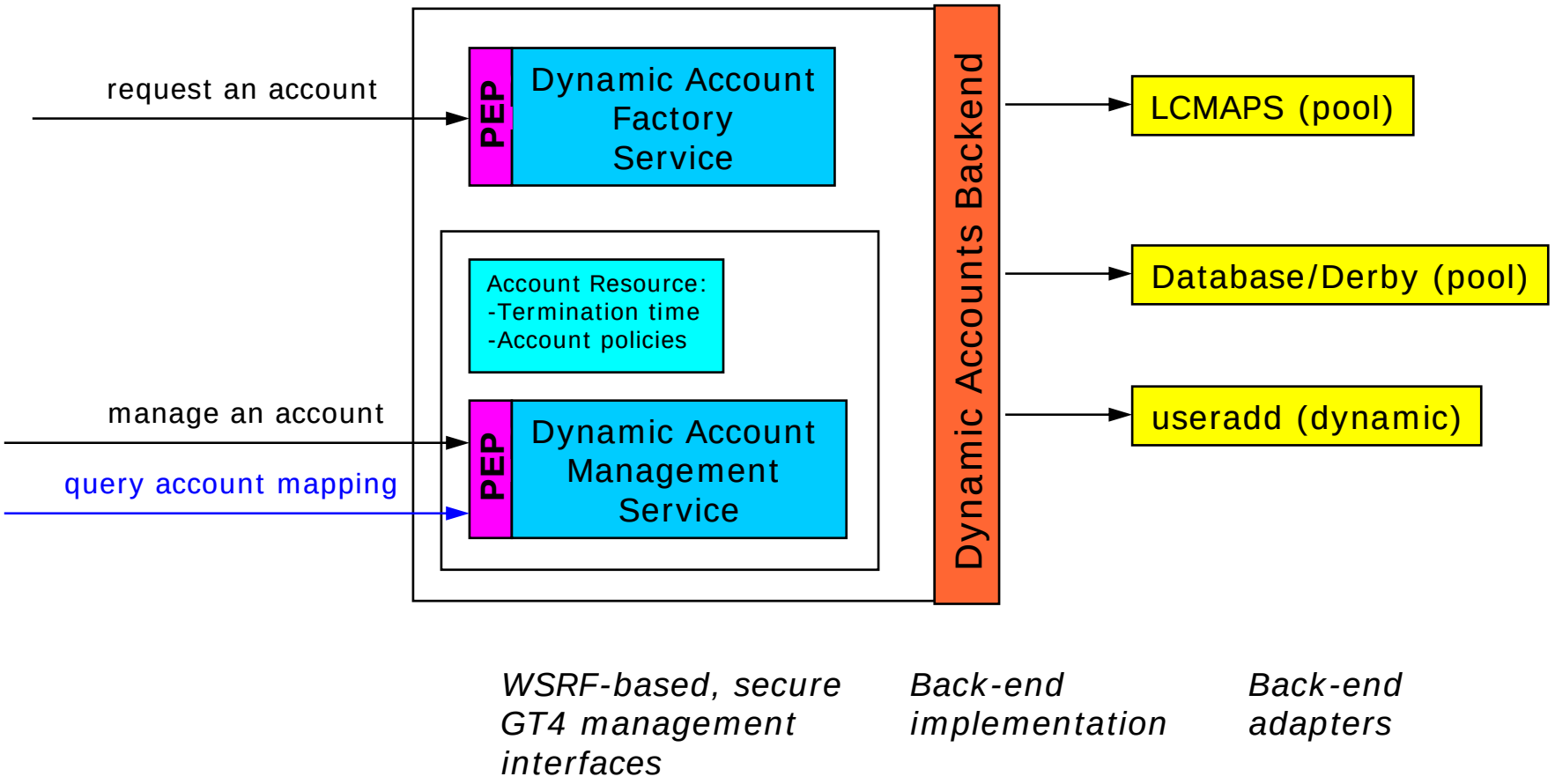
T. Freeman, R. Madduri,

F. Siebenlist

Requirements

- Mapping PKI credentials to local accounts
  - Mapping attributes into account attributes
- Creating a mapping creates an implied policy
  - The PKI credential mapped to an account can access and manage this account
- Resource allocation aspect
  - Account is a resource
  - Allocate accounts from a pool, manage those pools
  - Pools may be allocated on a per-attribute basis
- Policy management aspect
  - Account access policies: mapping multiple identities to a local account
  - Account management policies: who can manage this mapping

Dynamic Account Services



Account creation within a Trusted Computing Base (TCB)

Authorizing Workspace Use

- Authorization based on VOMS proxy attributes
- Creation (Factory)
  - Authorization via a DN ACL

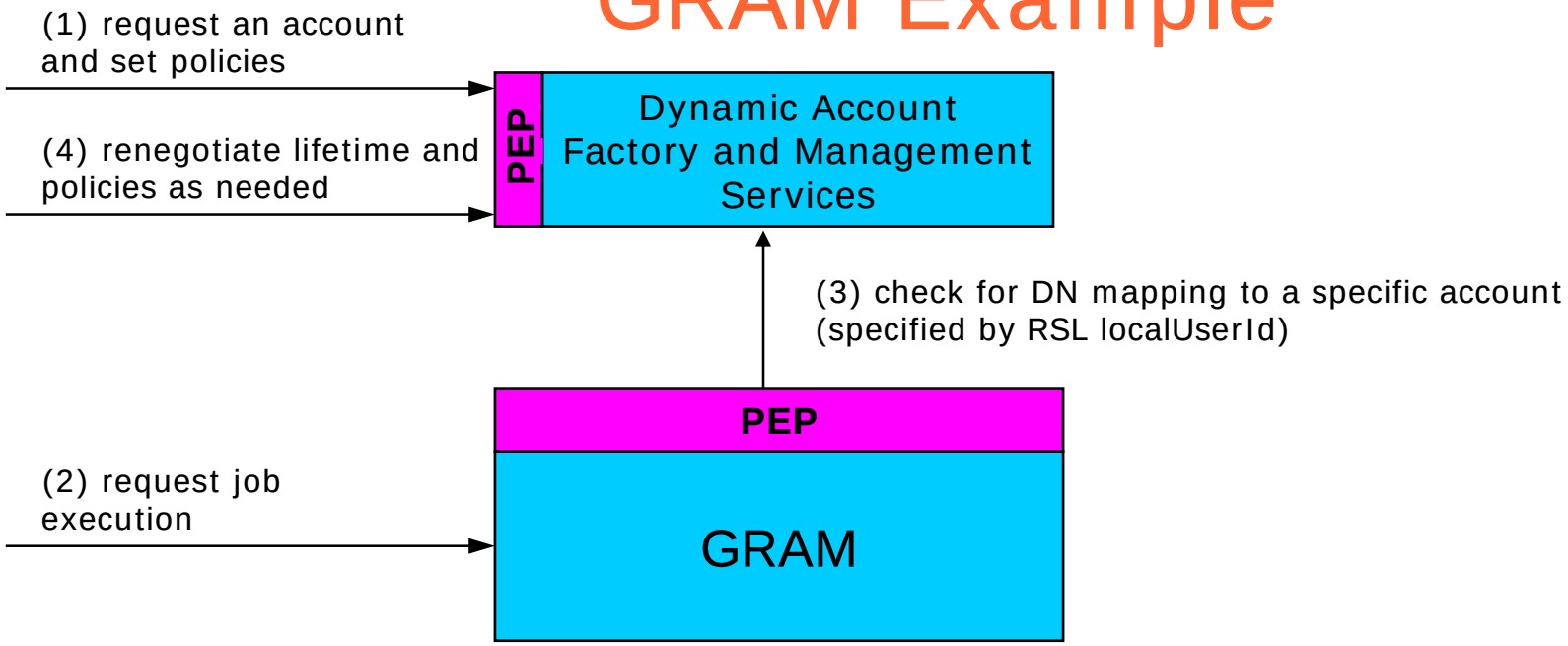
```
/DC=org/DC=doegrids/OU=People/CN=Timothy Freeman 964650
/O=Grid/OU=GlobusTest/OU=simpleCA-prnb3/CN=TimF
```

  - Authorization via an attribute ACL

```
/EGEE/egee/manager
/egtest/mytemp/Role=NULL/Capability=NULL
```

- Management and Inspection (Service)
  - Management functions accessible based on management policies
- Authorization callouts are customizable
  - LCAS

Using Dynamic Accounts Service: GRAM Example



- A service needs to be configured to work with the DA Service (configurable in GT4 GRAM)
- Prototype extended SAML interface to enable GUMS substitution
  - *Paper: Authorization Attributes, Obligations and Flexible Account Management in the OpenScienceGrid, GRID 2005*

Managing Accounts: Resource Aspect

- Pool Accounts
  - A site admin creates a finite pool of accounts
  - Accounts are assigned from a pool and potentially restored to the pool after they have been used
    - The same account may get assigned to multiple users -- audit issue
    - The number of available accounts is limited
    - How do we “clean” accounts?
    - How and when can accounts be quarantined?
- Truly dynamic accounts
  - Created by UNIX useradd call
  - Flexibilty: accounts created based on need
  - No need to recycle, simplifies audit
  - Could potentially interfere with local account management systems

Dynamic Accounts Backend

- Creation: poolindex function
  - DN+attributes -> pool lease + groups
- Termination
  - Via explicit destroy call or TTL termination
  - Expires the lease
  - Callout to clean: kill processes, delete files, revert groups back to default
  - Default script, configurable by site administrator
- Quarantine
  - Puts the account in a “quarantine pool”
  - Mandatory quarantine: if the termination script exits with errors or checks fail
  - Optional quarantine: configurable by site administrator
  - Quarantine removal can be manual or automatic

Backend Adapters

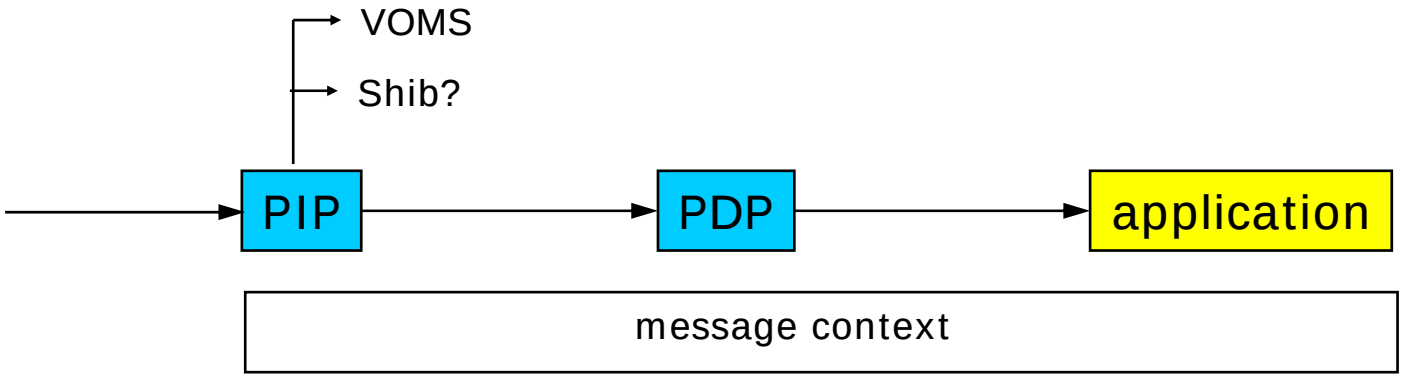
- LCMAPS
  - Developed at NIKHEF
  - Based on a gridmapdir patch
  - Maps credentials to pool accounts based on policy/algorithm by creating a hardlink
  - Allows for multiple pool leases
- Database (Derby)
  - A site administrator creates account pools and describes them in files (one file per pool), the database is populated from these files
  - The policies are managed in the database, uses db methods for persistence, transactions, etc.
- Truly dynamic accounts in prototype stage

Managing Accounts: Policy Aspect

- Account Access Policy: what DNs can map to this account?
  - “owner” access is implied
  - Optional: specify a list of DNs at creation
  - Enforcement depends on infrastructure
    - Plugins configure gridmapfile, lcmaps structures
- Account management policy
  - “owner” management is implied
  - Can I determine management policies for this account?
  - Management policies imply adding or limiting access to the account

Managing Accounts: Identity Mapping

- Account creation
  - Credential attributes are used for authorization
  - Credential attributes are mapped into attributes associated with the new credential, e.g.:
    - VOMS attributes -> UNIX groups,
    - An attribute -> account pool
  - Implied semantics attached to attributes within a system (TCB-specific policies)



Future Directions: CAS Interface

- CAS overview
  - WS Policy Management Interface
  - SAML-based OGSA-Authz authorization query
  - CAS is enhanced to accommodate dynamic, real-time policy management and enforcement
- CAS policy management as an alternative interface to the dynamic accounts service
  - Leverage CAS' full featured policy lifecycle management interface
  - Potentially also leverage CAS' more expressive policy language to write more sophisticated policies about accounts

Status

- Available as part of GT 4.0.2 distribution
  - Contribution, an incubator project
- Leverages GT4 features
  - GT4 logging for audit, persistence, security, etc.
- Integration with Globus services
  - GT4 GRAM patch available
  - Can be used with GT2 GRAM via a C client callout
- Documentation and download at http://workspace.globus.org/da

Workspaces and Dynamic Accounts

- Workspaces
  - Dynamically created and managed environment based on an authorized request
  - Associated with a resource allocation
  - Associated with an environment and its deployment capability
  - Associated with access and management policies
- Examples:
  - A physical machine configured to meet TeraGrid requirements
  - A cluster of virtual machines configured to meet OSG requirements
- Dynamic Accounts Service is part of the Workspace suite of tools
  - Also used to go by the name of WorkSpace Service (WSS)

Workspaces (cntd)

- Workspace creation:
  - Provision resources, provide/complete configuration, provide access
  - Dynamic accounts provide access
- Workspace Implementations:
  - Physical machines
  - Virtual Machines
- Virtual Workspace Service
  - Allows you to create an independently configured, isolated environment, manage its resource allocation on a fine-grained level
  - Used in OSG Edge Services
  - http://workspace.globus.org/vm

the globus alliance
www.globus.org

Summary

- Some current issues
  - Mapping into UNIX accounts: separating policy management and querying from resource management concerns
  - Agreement on interfaces for identity mapping and policy management
    - E.g., GUMS/Dynamic Accounts effort
  - Formalizing of attribute mapping between different domains
    - Right now typically defined by the implementation -- essentially requiring everybody to use the same implementation
    - More information at http://workspace.globus.org/da