



Facility/Resource/Service Management

Michal Šťava, Michal Procházka,  
Slávek Licehammer

# Outline



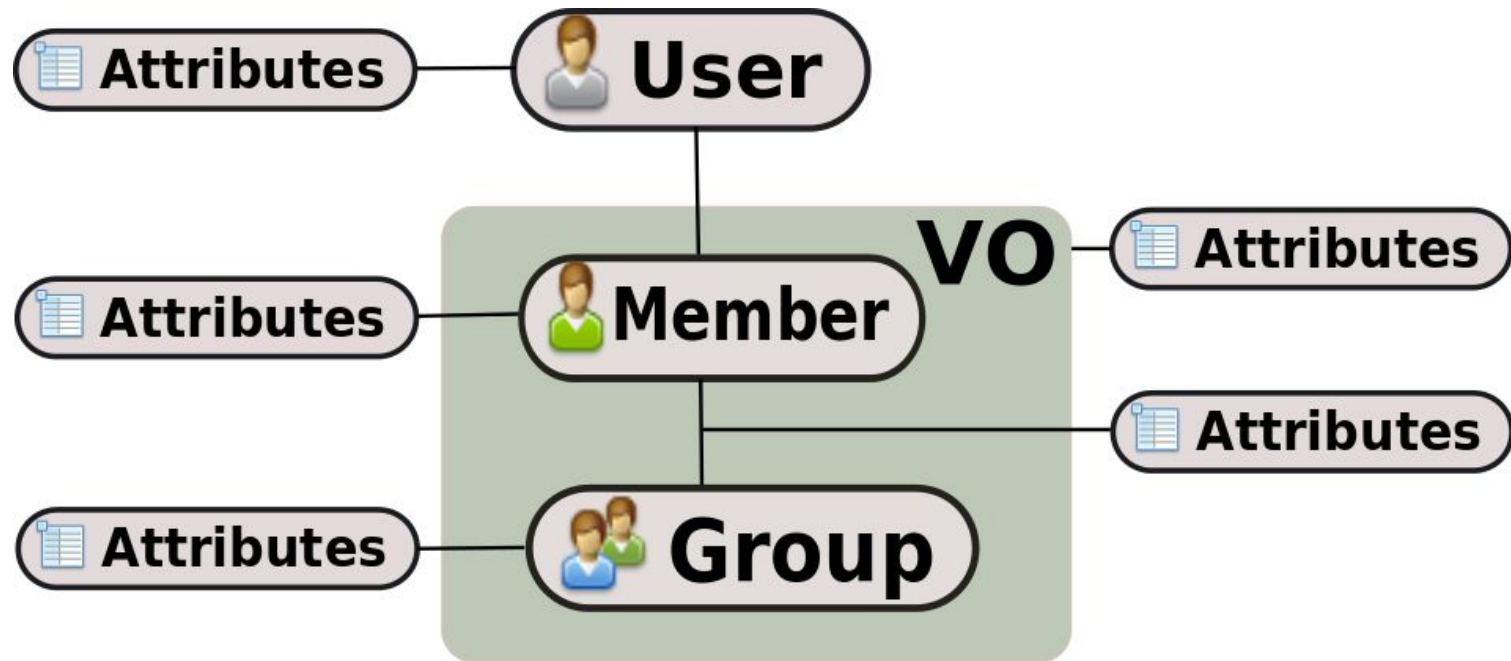
- Attributes
- Facility
- Resource
- Group
- Relationship between F/R/G
- Owners, Perun services and Destinations
- Provisioning (Push model vs LDAP)
- Examples

# Attributes

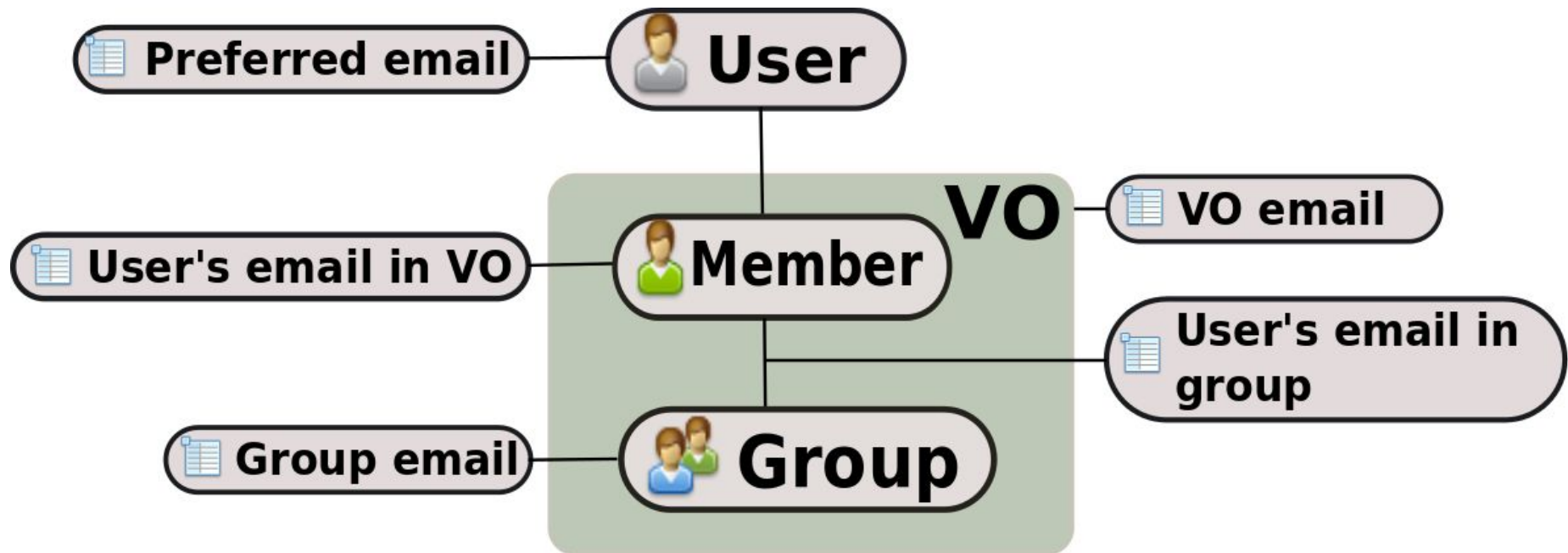


- Piece of information attached to the entities and relations among entities
- Various formats (String, Integer, List, Map)
- Attribute modules
  - Syntax checks
  - Auto-fill
  - Dependency checking
- All data in Perun can be obtained via Attributes

# Attributes



# Attributes



# Facility



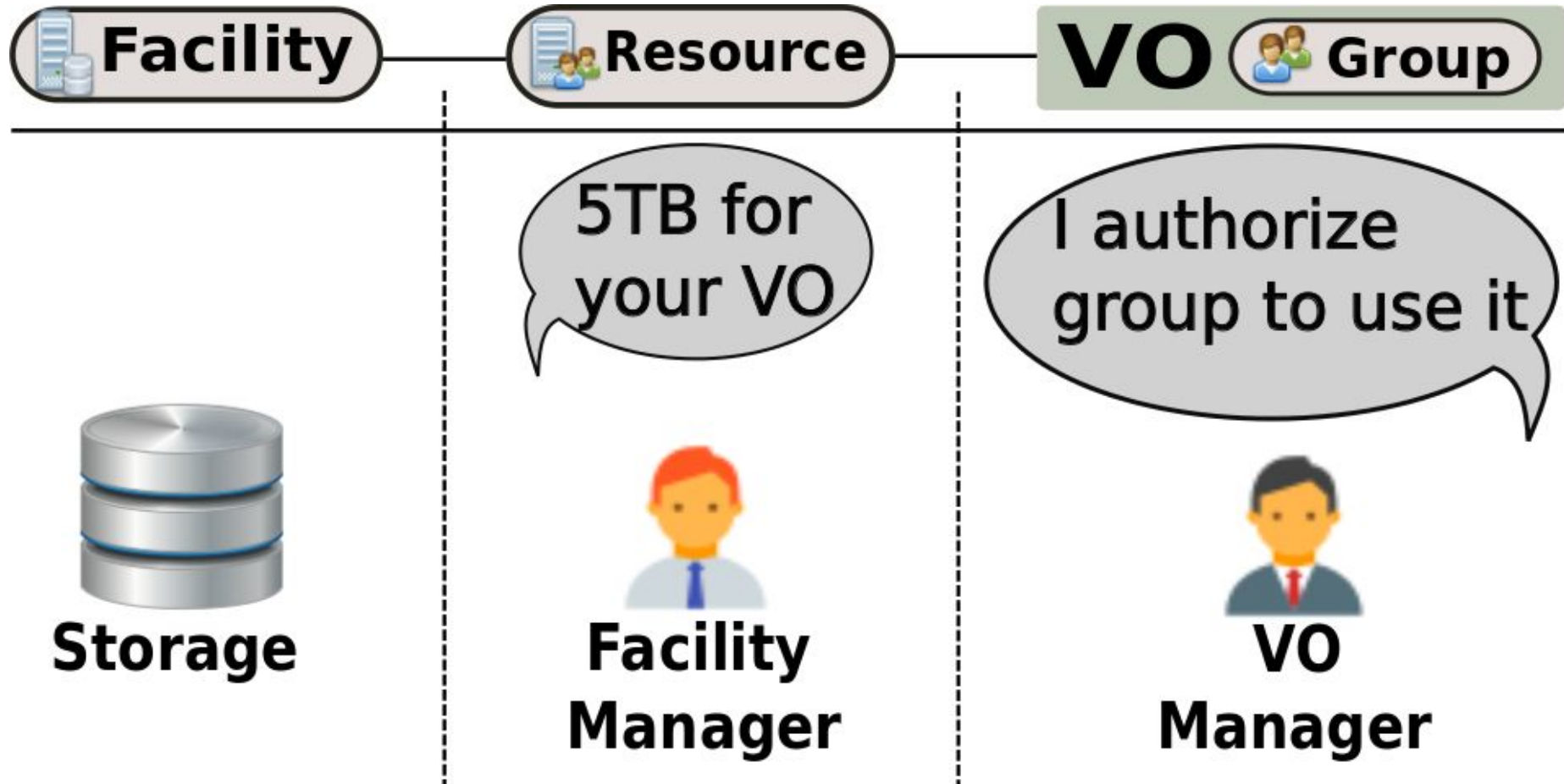
- Real world entity managed by Perun
  - physical or virtual (cluster vs mailing list)
  - homogenous configuration
- Has a set of specific attributes
- Can provide Resources to VOs
- Managed by Facility Manager

# Resource



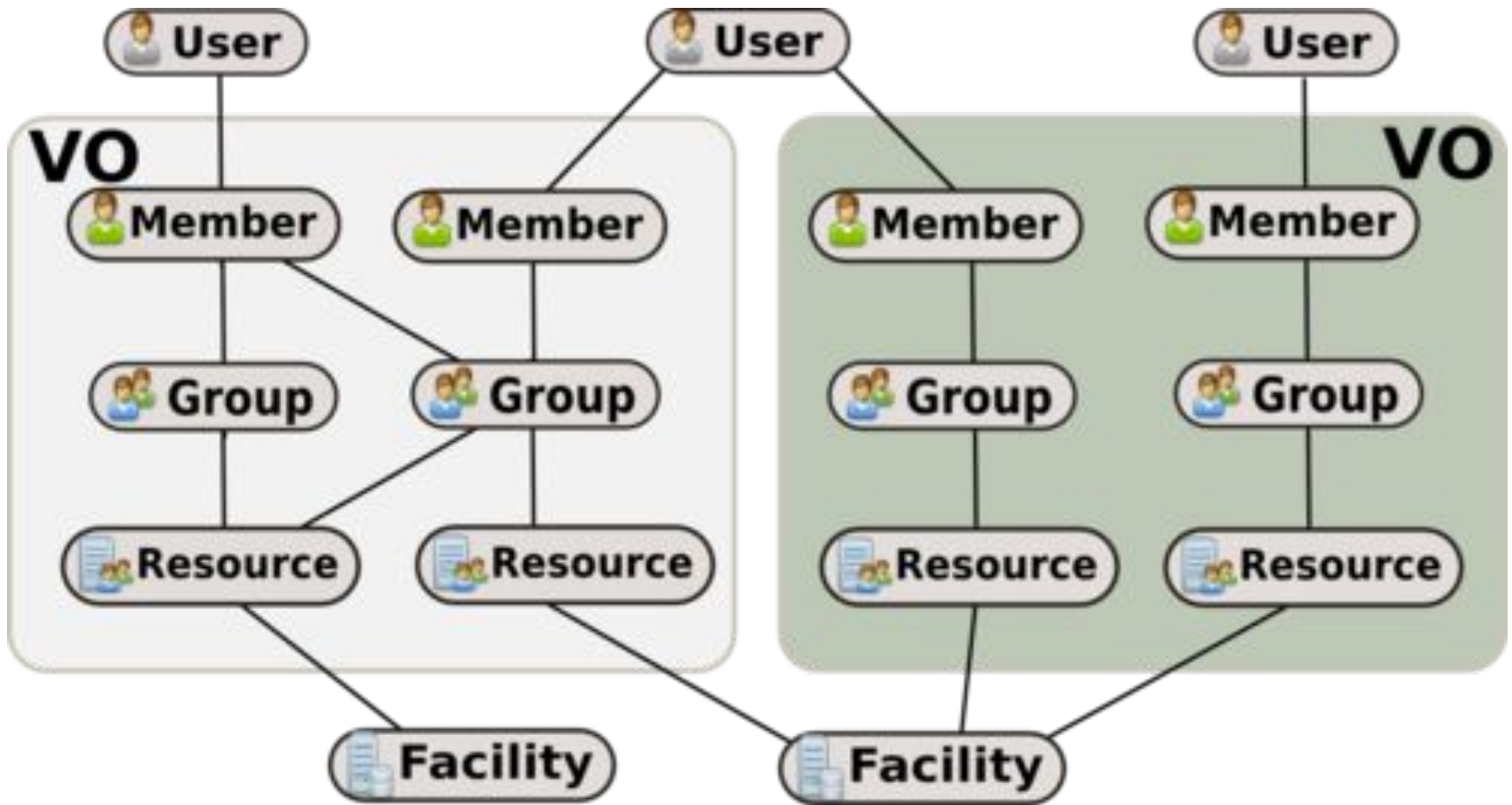
- Access to specified part of Facility for VO
- Resources are VO specific
- May restrict usage conditions (e.g.: disk quotas)
- Groups are assigned to Resources

# Facility and Resource





# Relationship



# Responsibilities



- Facility Manager:
  - creates and manages Facilities
  - creates Resources
  - assigns them to VOs
  - prepare services and other settings
- VO Manager:
  - chooses and assigns people on provided Resources
  - manages selected Resources Attributes

# Owners



- Owner is an administrative contact for Facility
- Contact to:
  - Person (not need to exists in Perun)
  - Organization
  - other
- Description or name with contact information (email, telephone number etc.)

# Perun services



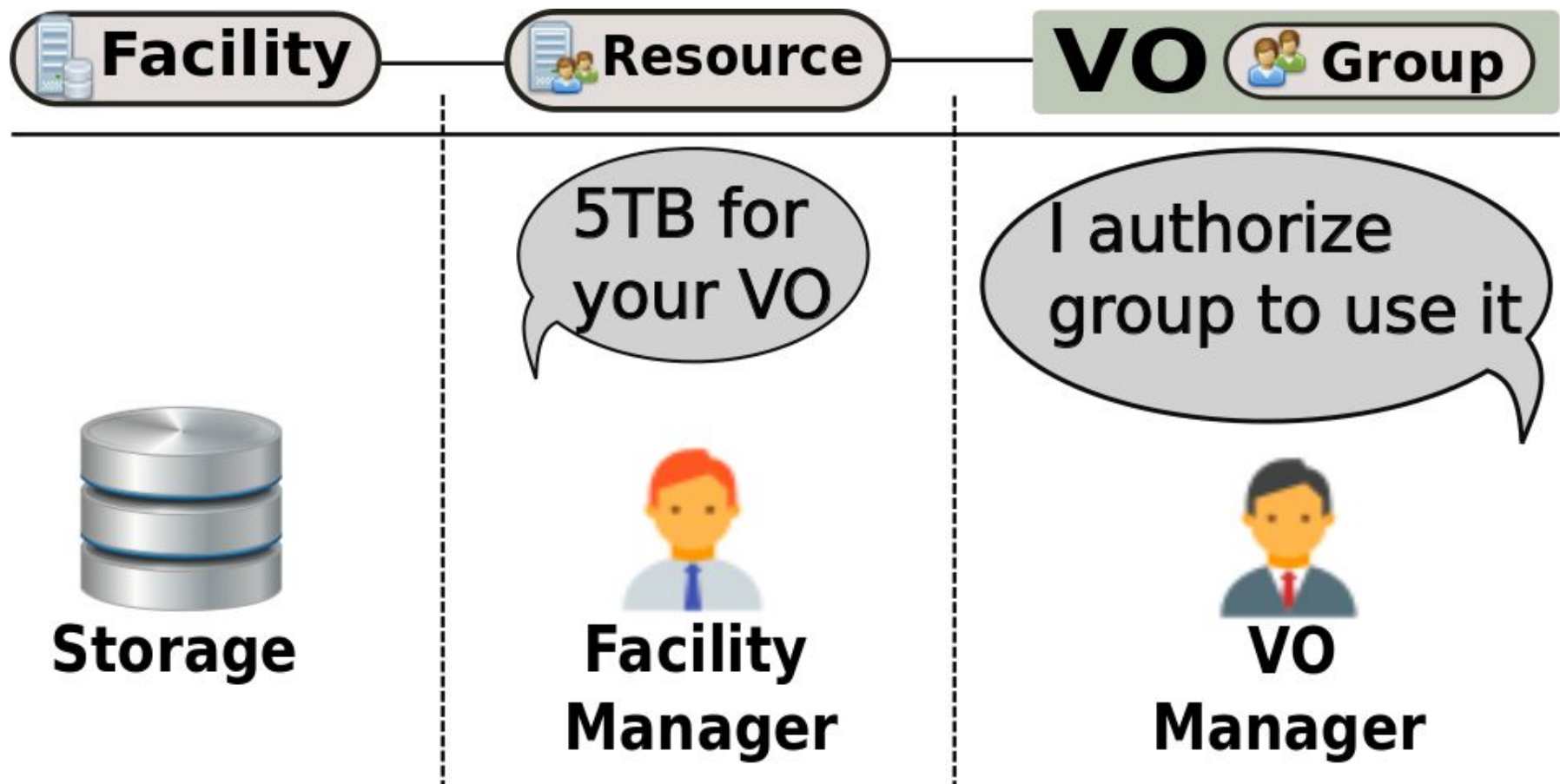
- Representation of a service on the Facility
- Script (or program) obtaining data about Perun entities assigned to the Resource
  - gen, send, slave scripts (explain later)
  - data for authorization decision support
  - saved in Attributes (user, facility etc.)
- Data for authorization are propagated to end services
  - e.g.: set unix account for all authorized users

# Destinations

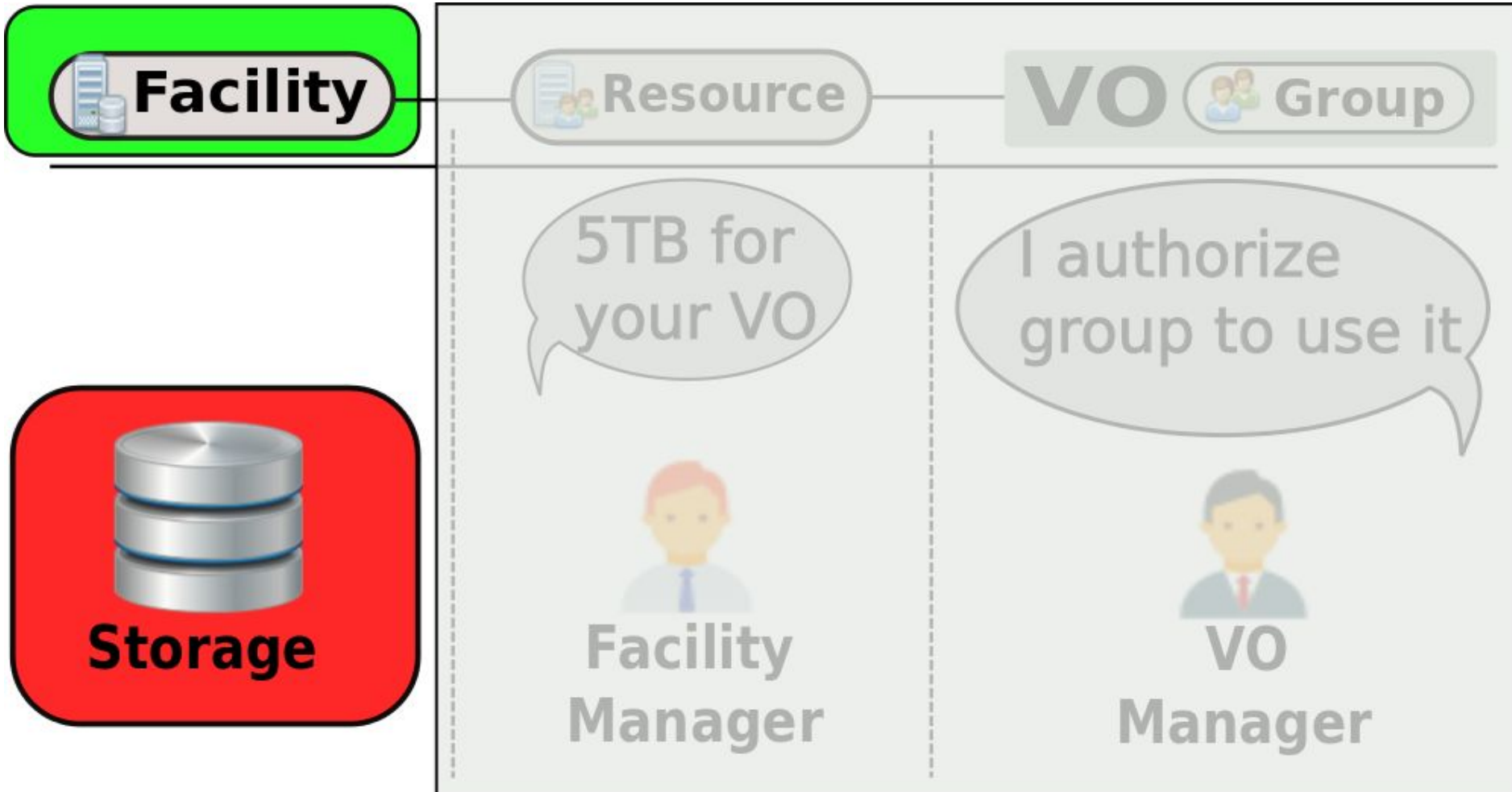


- Describe way to transfer configuration from Perun to real world entities
- Target of authorization data propagation
- Assigned to the Facility
- It is pair:
  - Perun Service
  - definition of a target propagation
- Managed by Facility Manager

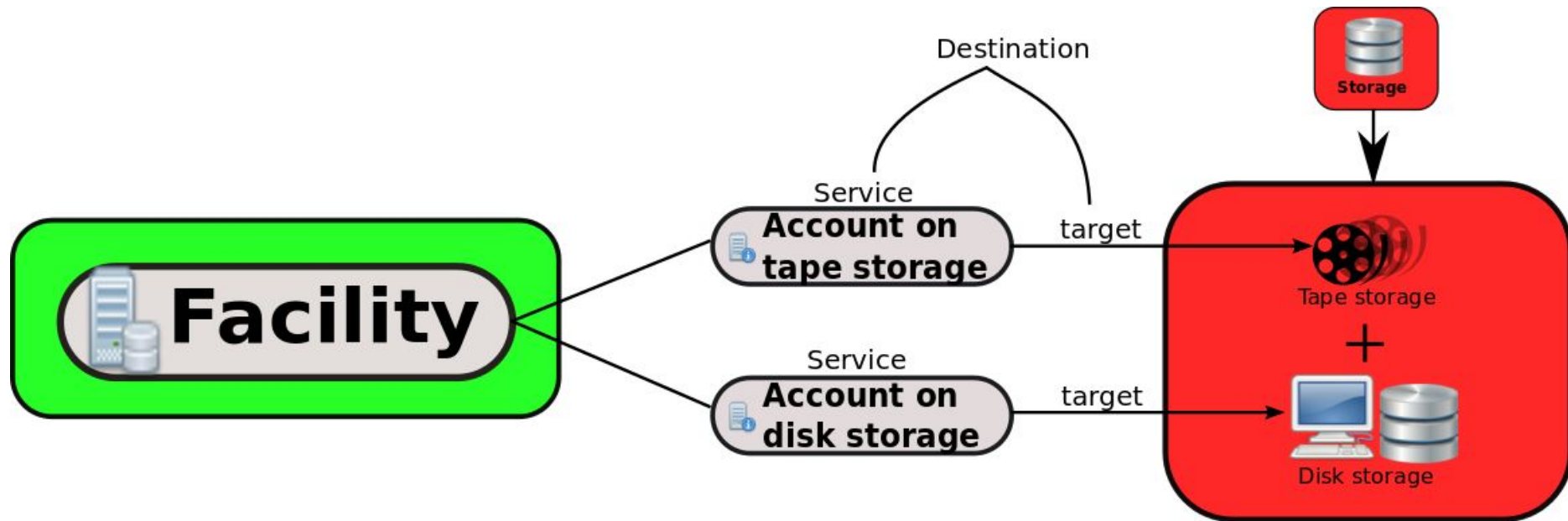
# Perun to real world mapping



# Perun to real world mapping



# Perun to real world mapping



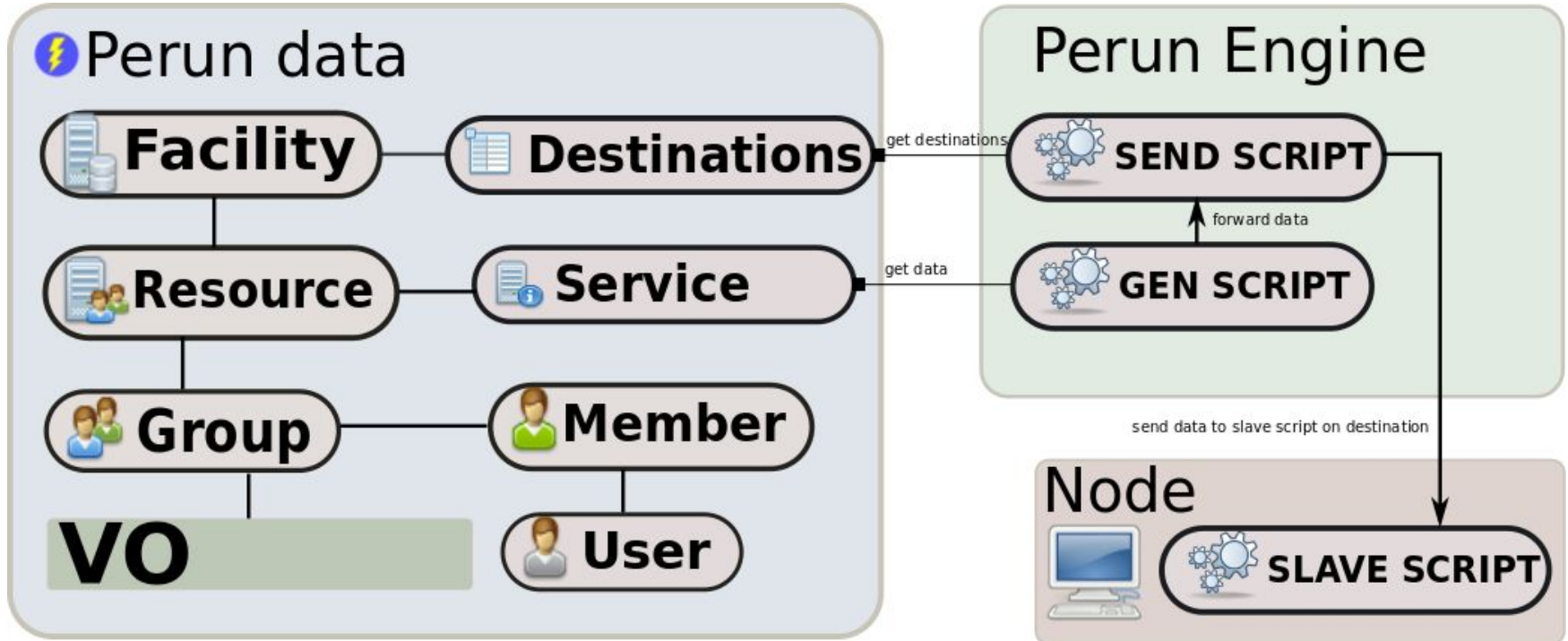


# Push model provisioning



- Most frequent propagation process
- There are 3 types of script:
  - GENERATE: generates authorization data (about Users, Groups etc.)
  - SEND: send already generated data to destinations
  - SLAVE: sits on destination, receives data from send script and process them (update web ACL, restart service etc.)
- It takes some time

# Push model provisioning

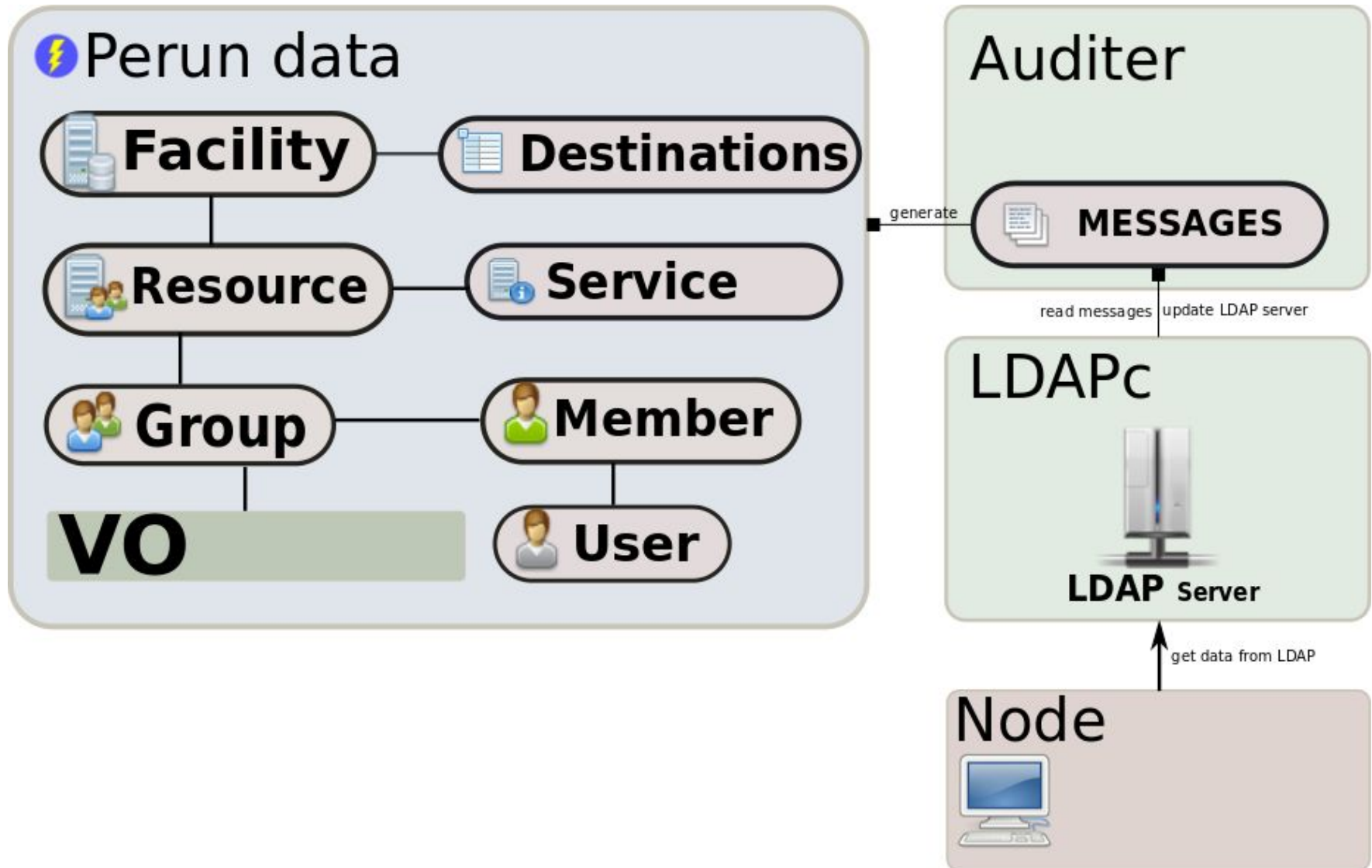


# Perun LDAP



- Another interface to Perun data
  - changes are processed in a real time
- Consumers get rights to read specific information from the Perun LDAP
- Real time access to data in LDAP

# Perun LDAP

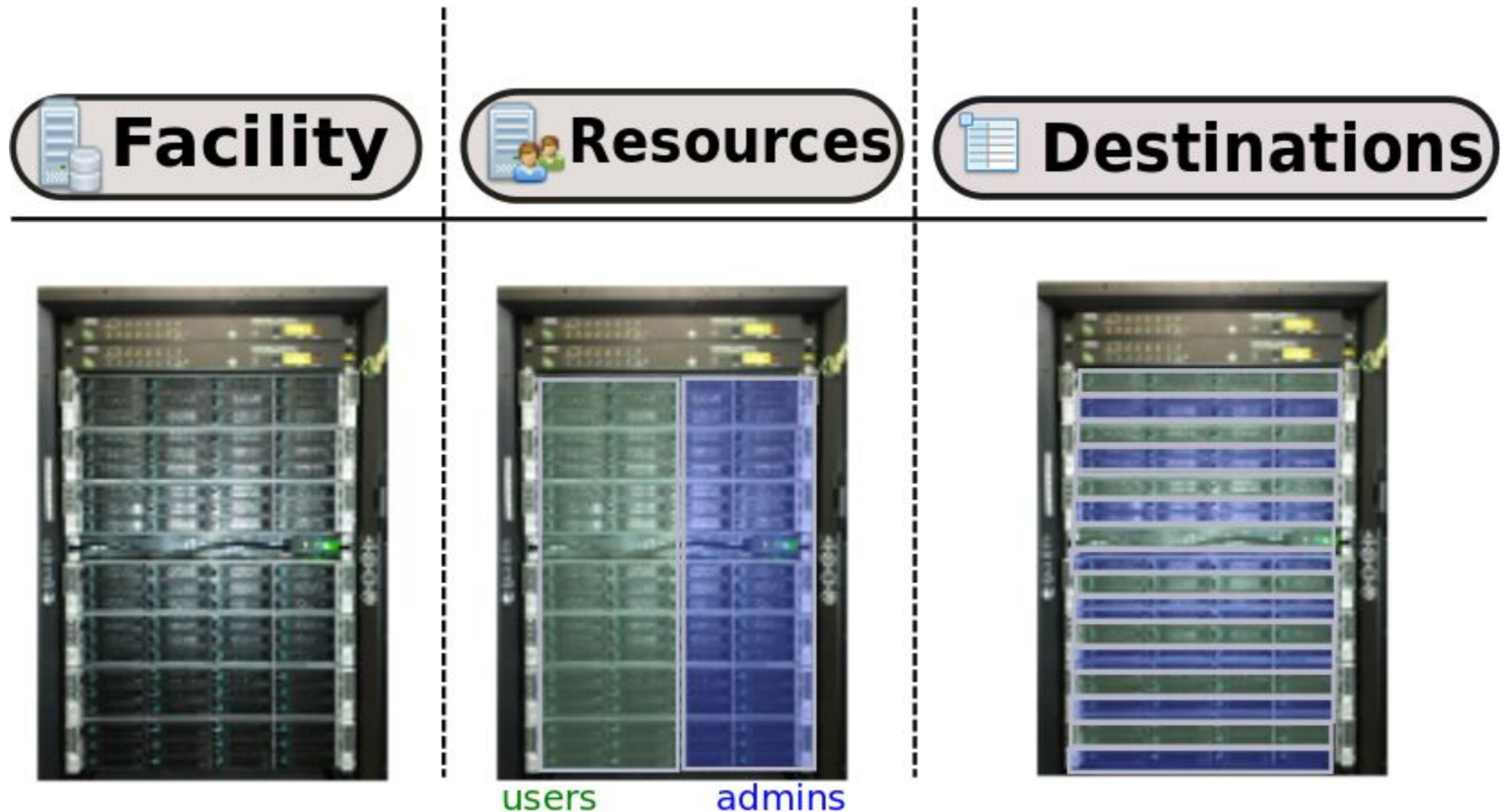


# Example 1 - Cluster management



- Facility = whole cluster (homogenous)
- Resources: (2 per VO)
  - 1 for users (unix account)
  - 1 for admins (k5login\_root)
- Destinations = all nodes of cluster
- Default options and limitations defined by Facility Manager
- Preferred options defined by Users itself
  - Using gen/send/slave mechanism

# Example 1 - Cluster management

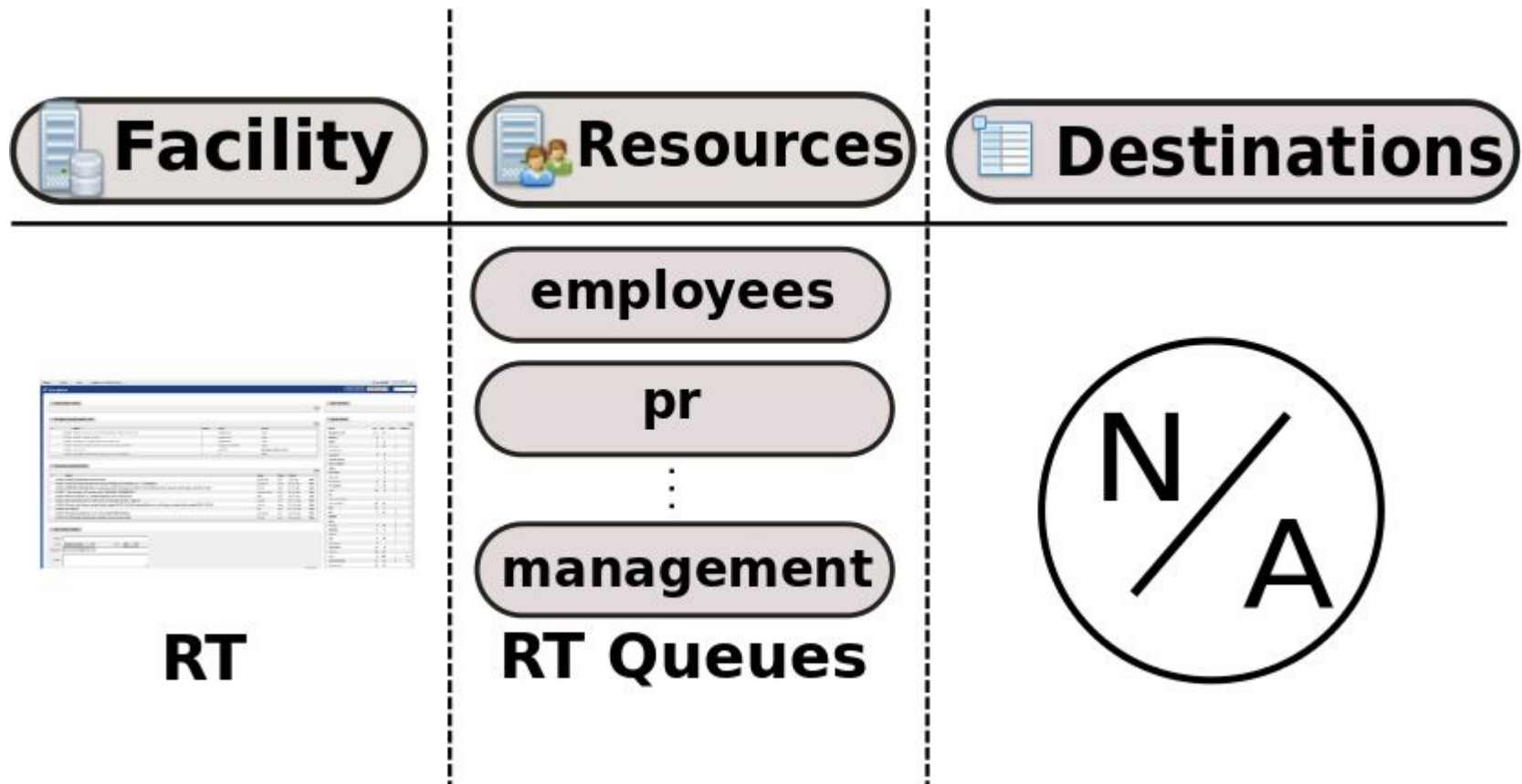


# Example 2 - RT management



- RT - Request Tracker (issue tracking system)
- Facility = RT
- Resources = Queues
- Facility Manager provides queues to chosen VO
- VO Manager can assign Groups only to provided queues
- Members from Perun are strictly synchronized with RT system
  - Using Perun LDAP mechanism

# Example 2 - RT management





# Example 3 - Mailing lists



- Facility = Mailman or Sympa
- Resources = mailing lists
- Destinations = host with Mailman or Sympa
- Attributes = email of mailing list manager
  - Using gen/send/slave mechanism

# Example 3 - Mailing lists

