DIRAC security infrastructure

DIRAC Project

Outline

- Basic principles
- DISET framework
  - Motivation and overview of DISET
  - Service certificate
  - DIPS protocol as part of the DISET framework
  - Authentication mechanism
  - Authorization algorithm
  - Properties
- Security section in the local configuration file
- Authorization related options in Configuration Service
  - Users
  - Groups
  - Hosts
  - VO
  - Services methods
- Typical administration tasks
  - Add/Remove a user
- Tutorial Exercises

Basic principles

- Trusted certification authorities (CA) are used for authentication
- Virtual Organizations are used for authorization
- Minimization of dependencies on the Globus toolkit
- DIRAC clients and agents use proxies to connect to DIRAC services
  - Proxy is using the X.509 PKI standard
- DIRAC servers has to be authenticated by client as well

DISET: Motivation and overview

- DIRAC goals by design:
  - Secure communication layer for accessing the resources
  - Framework to be used to build services and agents easily
- DIRAC Secure Transport - DISET:
  - Provides a set of tools used to create services or agents quickly
  - Uses a python wrapper around industry standard OpenSSL for secure transactions between services, agents and clients
  - Fine grained authorization rules
    - Per individual user or service using FQAN
    - Per service interface method

Service and host certificate

- Each and every DIRAC service have to be authenticated
- Service authentication can be done using service certificate
- Since service certificates are not issued or recognized by many authorities, DIRAC system uses host certificate as service certificate replacement by default
- Main difference between service based authentication and user authentication is that service based authentication is password-less
- Therefore service certificates should be protected as strongly as host certificate but the owner of the certificate should be dirac user
  - <DIRACRoot>/etc/grid-security/hostkey.pem with 400
  - <DIRACRoot>/etc/grid-security/hostcert.pem with 644

Service certificate. Properties

- Some services need or are allowed to have different kind of information from other services
- For quick service authorization DIRAC uses so-called properties
  - Example: JobAdministrator, FullDelegation or TrustedHost
  - Full list of properties and their explanation will be given later
- Since we are using host certificate for a service authorization, properties are set per host in special section in CS
  - Properties are "summing up" in case if a host have several services running on it
- Web service has to have "TrustedHost" property only and service itself need to run on dedicated machine
  - This limitation can be avoided with proper service certificate

DIP/DIPS

- DIP is a custom protocol that provides RPC and file transfer capabilities
  - Persistent connections will be available in new version
- DIPS is the DIP protocol with SSL authentication and DISET authorization
- Any service-to-service or client-to-service connection in DIRAC uses DIPS protocol
- Fine tuning of timeouts of DIPS protocol can be done by administrators
  - In most of the cases, the default timeouts are working just fine

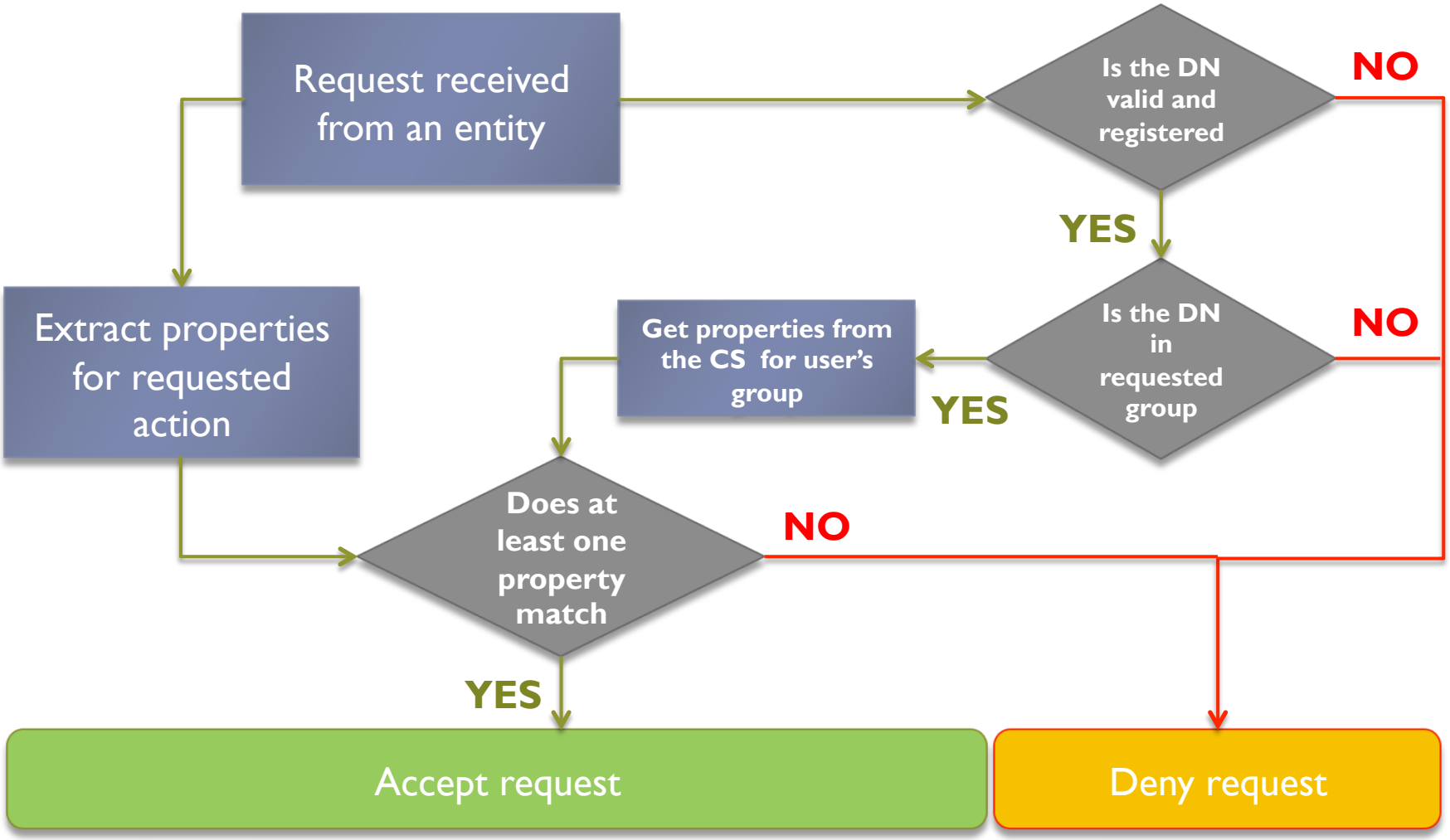
Authentication mechanism

- Based on X.509 Public Key Infrastructure standard
  - Uses standard grid certificates and certificate proxies
- Authentication is done by checking of received credentials against list of CAs
- Client can use both certificate and certificate proxy
- Service can use certificate only
- The main work is done by enhanced OpenSSL library
- One handshake per multiple calls
  - Session lifetime can be changed by administrators

Authorization algorithm

- On any action, the very first thing done by DIRAC is to check the validity of action request
  - DISET authenticates the request and extracts credentials from SSL handshake
  - Using extracted credentials DISET associates a set of properties to given requester:
    - In case of a host: Check that DN is registered in CS and get properties from Configuration Service
    - In case if a user: Get user's group from the request and associate group properties with the current user
  - If a single property of requester is matching to one of action's allowed properties then the action will be executed
  - If an action has no properties then any authenticated entity can execute it
    - There is a possibility to set a default list of properties for actions of a service

Authorization state machine



Authorization. Groups

- A user can belong to several DIRAC groups
  - User selects a group to work under at the time of proxy creation
  - Selected group is embedded as X.509 extension in the user's proxy and is signed by the user certificate
- DISET automatically extracts user's group, if the group is embedded in first level after the user's certificate in the proxy delegation chain
- On delegation, extra levels can be added to user's proxy but the first level with the user's group remains the same. It can not be re-written by any other entity.

List of properties

- *JobAdministrator*: Used to supervise **ALL** DIRAC jobs (analogue of a root user in WMS)
- *CSAdministrator*: Possibility to edit the Configuration Service
- *FileCatalogManagement:* Used for FC Management (analogue of a root user in FC)
- *AlarmsManagement:* Allow to set notifications and manage alarms
- *ProxyManagement:* Allow to manage the proxies (i.e. delete a proxy)
- *FullDelegation:* Allow to get full delegated proxies (i.e. when an operation have to be done using standard user proxy) normally used by agents
- *LimitedDelegation:* Allow to extract only limited proxies (i.e. for pilots). Such proxies can be used for data uploading but not for a job submission
- *PrivateLimitedDelegation:* Allow to get only limited proxies for one self
- *GenericPilot:* Generic pilot property used to extract any proxy from proxy storage
- *Pilot:* Private pilot allow to extract proxy for current DN only
- *NormalUser:* Normal user operations
- *JobSharing:* Job sharing among members of a group (defined in code)
- *ServiceAdministrator:* restart services
- *Operator:* Operator can monitor DIRAC services but can't restart them
- *SiteManager:* Used to display a site releated monitoring information
- *TrustedHost:* Host defined in the system to be trusted, used to work on behalf of a user

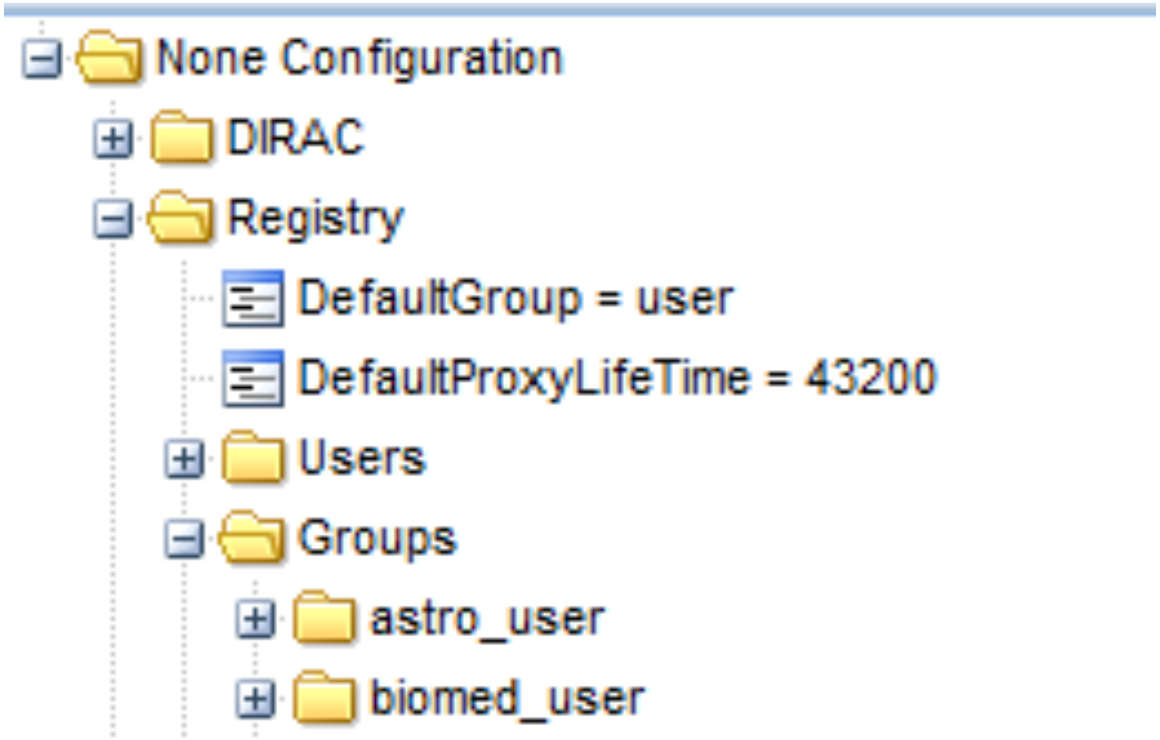
Local configuration

- Local configuration is stored in <DIRACRoot>/etc/dirac.cfg file and consists of different sections. One important section is /DIRAC/Security. It can contain options:
  - CertFile: File name
    - Permissions of file should be 644
    - Owner is DIRAC user
  - KeyFile: File name
    - Permissions of file have to be set to 400
    - Owner is DIRAC user
  - SkipCACheck: Boolean
    - Default is "No"
    - Use value "Yes" if:
      - No gLite UI installed on the machine
      - Certificates are not updated regularly
    - SkipCAChecks allow to use one way SSL handshake and should be used if a client is trusting the services. Services should use mutual authentication
  - UseServerCertificate: Boolean
    - Default is "No"
    - Use value "Yes" in case of:
      - Service installation
      - If no service certificates are used or installed

/Registry section

- */Registry* section contains among other things authorization rules for the users, groups and hosts. Also VO specific information is stored in this section
- System wide options are defined right in the root of *Registry* section:
  - *DefaultGroup*: Default user group to be used if the user for some reason did not specify a group. String
    - /Registry/DefaultGroup = user
  - *DefaultProxyTime*: Default proxy life time in seconds. Integer
    - /Registry/DefaultProxyTime = 4000

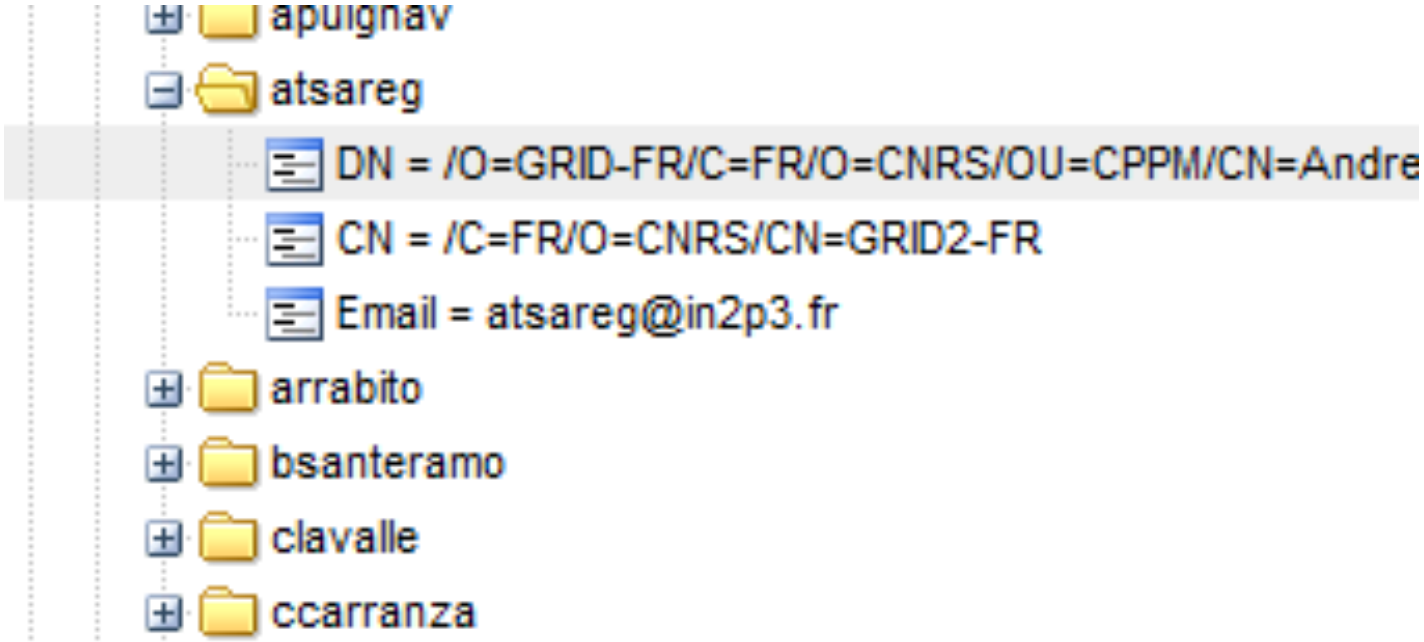
/Registry section



/Registry/Users

- /Registry/User section is used to store all users' related information. Each subsection is represents a user and called as DIRAC user name
  - /Registry/User/atsareg
- Subsection contains the attributes associated with the user. Options in red color are required:
  - DN: Distinguished name obtained from the user's certificate. String
    - /Registry/User/atsareg/DN = /O=GRID-FR/C=FR/O=CNRS/OU=CPPM/CN=Andrei Tsaregorodtsev
  - CN: Canonical name of certification authority who issued the user's certificate. String
    - /Registry/User/atsareg/CN = /C=FR/O=CNRS/CN=GRID2-FR
  - Email: User's email. Could be used for automatically sending of alerts or notification. String
    - /Registry/User/atsareg/Email = atsareg@in2p3.fr
  - Mobile: User's mobile. Could be used for sending sms with alerts. String
    - /Registry/User/atsareg/Mobile = +3362155555
  - Quota: Quota of disk space assigned to the user in MegaBytes. String
    - /Registry/User/atsareg/Quota = 300

/Registry/Users



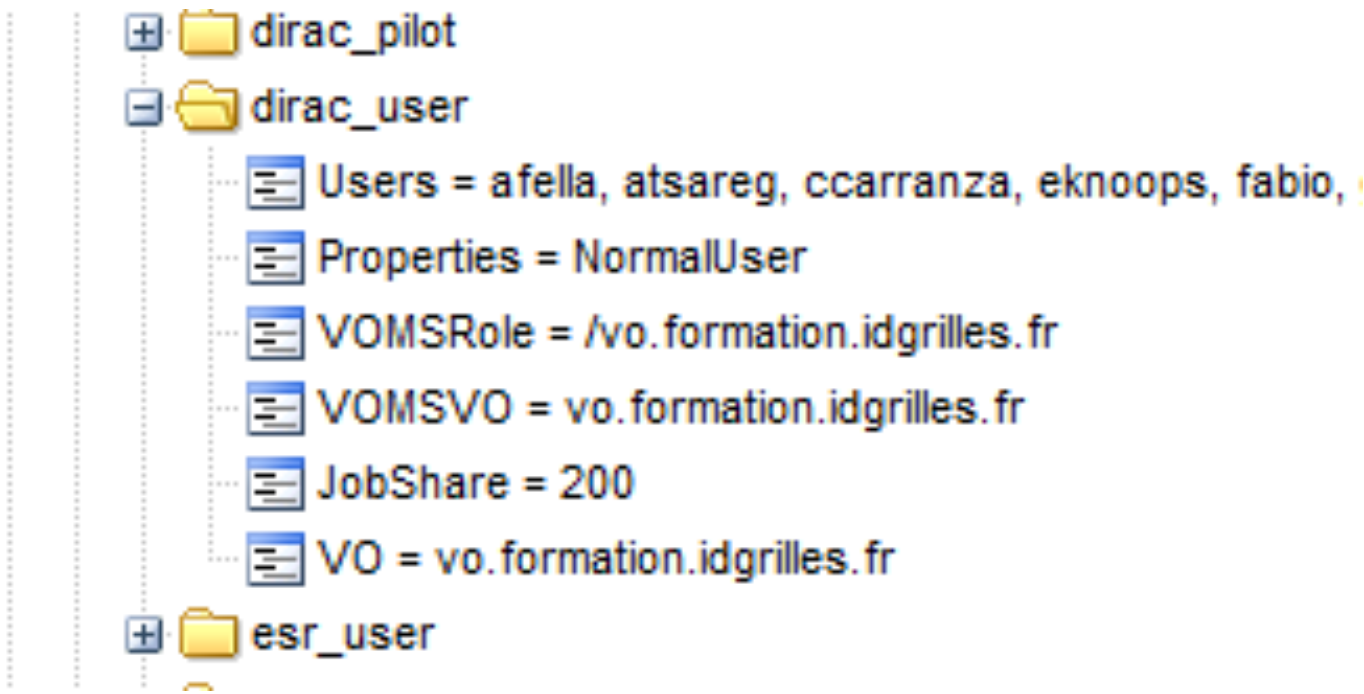
/Registry/Groups

- The main place to set authorization properties for the users
- Each subsection corresponds to a dirac group. Option in red is required
  - Users: DIRAC users logins than belongs to the group. List of strings
    - /Registry/Groups/dirac/Users = vhamar, atsareg, msapunov
  - Properties: List of properties of the group. This is the place where you are setting authorization rules. List of strings, e.g.
    - /Registry/Groups/dirac/Properties = NormalUser
  - *VOMSRole:* Role of the users belonging to the group in a VO. String
    - /Registry/Groups/dirac/VOMSRole = /lhcb/Role=production

/Registry/Groups

- VO: Nickname of a VO members of the group are belong to. Value should correspond to a VO described in /Registry/VO section. String
  - /Registry/Groups/dirac/VO = lhcb
- JobShare: Used if there is a high concurrency for the available resources among the users of different groups (production vs. users). Integer
  - /Registry/Groups/dirac/JobShare = 200
- AutoUploadProxy: Used to indicate dirac-proxy program to upload or not a user's proxy to the proxy store. Boolean
  - /Registry/Groups/dirac/AutoUploadProxy = True
- AutoAddVOMS: Indication of including VOMS attributes to the proxy extended by Proxy store. Boolean
  - /Registry/Groups/dirac/AutoAddVOMS = True

/Registry/Groups



Group defaults

- Some of the groups has a set of predefined properties which are used if no properties are set
- dirac_admin
  - AlarmsManagement
  - ServiceAdministrator
  - CSAdministrator
  - JobAdministrator
  - FullDelegation
  - ProxyManagement
  - Operator
- dirac_pilot
  - GenericPilot
  - LimitedDelegation
  - Pilot
- dirac_user
  - NormalUser

/Registry/Hosts

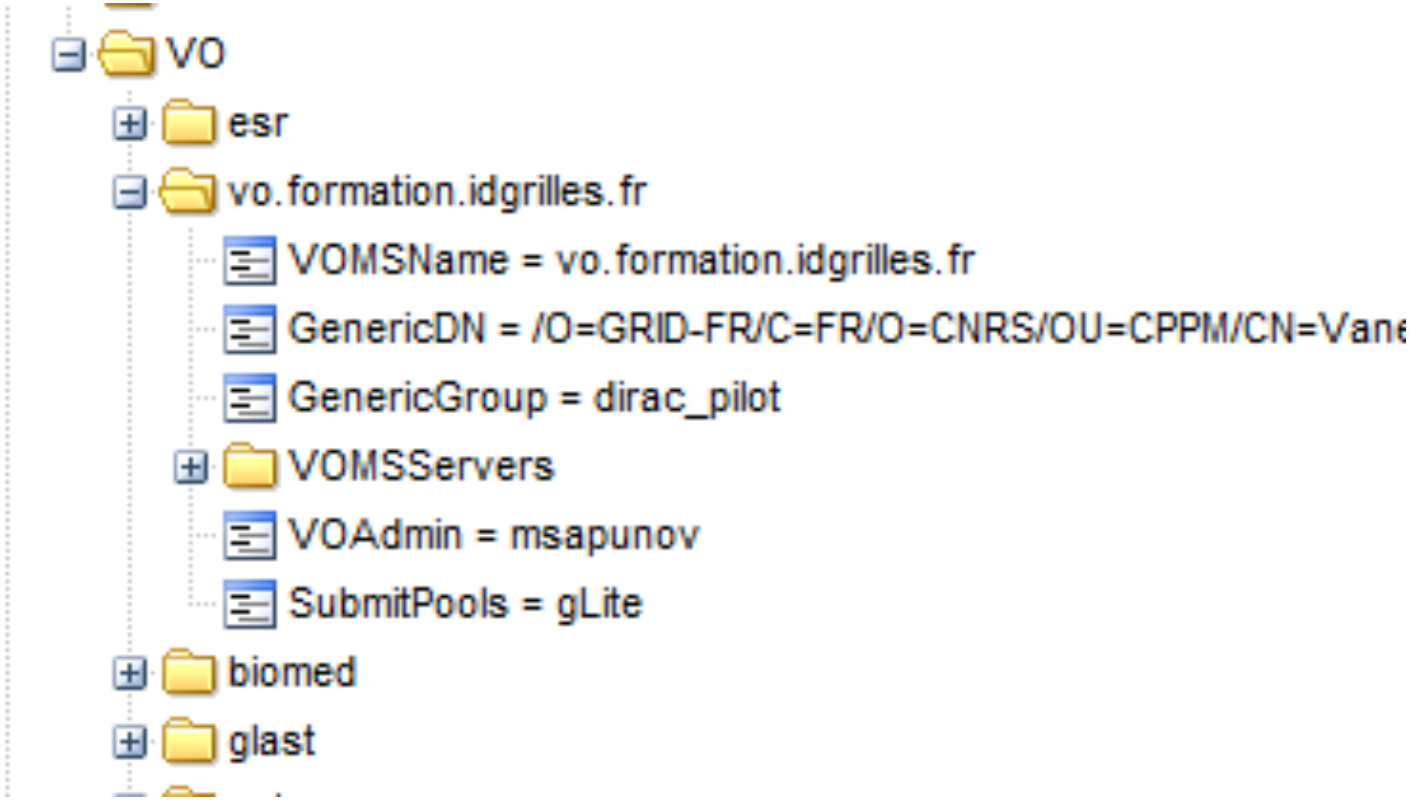
- This section is used to describe authorization rules for trusted hosts and services which are using hosts certificates
- Properties are set per host in the CS in /Registry/Hosts section
- Each subsection corresponds to one host. Naming convention is the following: "host-" + real host name
  - /Registry/Hosts/host-dirac.in2p3.fr
- Normally, there are two options in the host subsection but only one option indicated by red color is required:
  - DN: DN of the host's certificate used by DISET for automatic host authentication
    - /Registry/Hosts/host-dirac.in2p3.fr/DN = /O=GRID-FR/C=FR/O=CNRS/OU=CC-IN2P3/CN=dirac.in2p3.fr
  - Properties: List of properties corresponding to certificate's DN
    - /Registry/Hosts/host-dirac.in2p3.fr/Properties = JobAdministrator, Operator, FullDelegation

/Registry/Hosts

/Registry/VO

- Used to store VO specific information. No authorization rules are needed to set in this section
- Each subsection is a nickname and it should corresponds to the /Registry/Groups/<GroupName>/VO option in Groups section
  - VOMSName: Full name of the Virtual Organisation. String
    - /Registry/VO/<VONickname>/VOMSName = astro.vo.eu-egee.org
  - VOAdmin: DIRAC user name of the VO administrator. String
    - /Registry/VO/<VONickname>/VOAdmin = msapunov
  - GenericDN: Used as the DN by generic pilots and generic pilots payload would be displayed in Grid monitoring under this DN. String
    - /Registry/VO/<VONickname>/GenericDN = /O=GRID-FR/C=FR/O=CNRS/OU=CPPM/CN=Andrei Tsaregorodtsev
  - GenericGroup: Group to be used by generic pilot in order not to mix pilots payload with normal user payload. String
    - /Registry/VO/<VONickname>/GenericDN = dirac_pilot

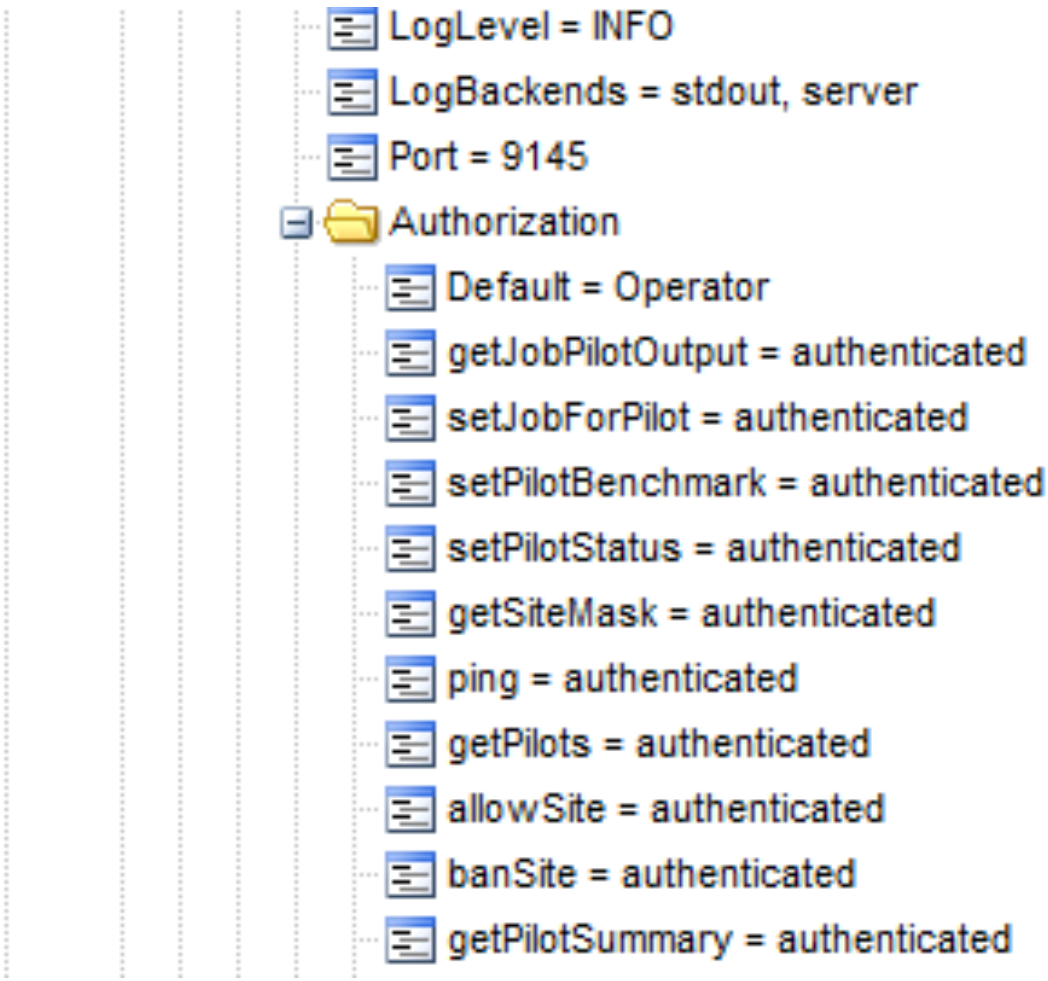
/Registry/VO


VO
esr
vo.formation.idgrilles.fr
VOMSName = vo.formation.idgrilles.fr
GenericDN = /O=GRID-FR/C=FR/O=CNRS/OU=CPPM/CN=Vane
GenericGroup = dirac_pilot
VOMSServers
VOAdmin = msapunov
SubmitPools = gLite
biomed
glast

Service methods

- For fine tuning of authorization capabilities it's possible to set certain rules per service methods
- Rules can be defined per service and per setup
- The subsection which contains the authorization rules is following the certain convention /Systems/<SystemName>/<Setup>/Services/<ServiceName>/Authorization
  - /Systems/WorkloadManagement/Production/Services/WMSAdministrator/Authorization
- Each option is the name of the service method apart from special option which sets the default property
  - /Systems/WorkloadManagement/Production/Services/WMSAdministrator/Authorization/Default = Operator
  - /Systems/WorkloadManagement/Production/Services/WMSAdministrator/Authorization/getSiteMask = authenticated
  - /Systems/ResourceStatus/Production/Services/ResourceStatus/Authorization/ping = All
- Also authorization subsection could contain a subsection which handles file transfer rules:
  - /Systems/WorkloadManagement/Production/Services/SandboxStore/Authorization/FileTransfer/Default = authenticated

Service methods



Administration tasks

- Add a user
  - Checks if a user is registered in a VO
  - Add dirac username (might be taken from the VO) DN,CN and email to CS
    - /Registry/Users
  - Add dirac username to a dirac group
    - /Registry/Groups
  - Add dirac username to the default dirac group
    - /Registry/Groups/<DefaultGroup>/Users
  - Create an entry in DIRAC FileCatalogue according to the convention rules
    - /<vo>/user/X/Xuser
  - Set disk quotas
    - /Registry/Users/Username

Administration tasks

- Add a group
  - Create a group
    - /Registry/Groups
  - Set properties. For most of the cases “NormalUser” property is enough
    - /Registry/Groups/<GroupName>/Properties
  - Add users to the group
    - /Registry/Groups/<GroupName>/Users
  - Add VO nickname if the users are belongs to a VO
- Add a host
  - Create host subsection
    - /Registry/Hosts/
  - Add DN
    - /Registry/Hosts/host-<HostName>/DN
  - Add authorization rules if required
    - /Registry/Hosts/host-<HostName>/Properties

Administration tasks

- Add a VO
  - Create a subsection with VO nickname
    - /Registry/VO/<VONickname>
  - Set the full VO name in VOMSName section
    - /Registry/VO/<VONickname>/VOMSName
- Ban a user
  - Remove dirac username from dirac group
    - /Registry/Groups/dirac/Users
- Remove a user
  - Remove jobs
  - Remove proxy from proxy store
  - Remove data from SE (the most difficult task)
  - Remove entries from FileCatalogue (easy in case of DIRAC FC, tricky in case of LFC)
  - Remove entries from /Registry/ Users and /Registry/Groups

Administration tasks

- There are some administration task which are quite complicated with no simple recipe
  - Delete user's data from the SE could be done if
    - User's credentials are exists and "alive"
    - User is still a member of the VO
  - Move a user from one VO to another
    - From national VO to a professional VO
    - The main question is data
    - The only possible solution is to organize data replication proxy which would physically move and register data from one SE to another

Exercises

- Add user using command line tools and CS editor
- Add group also using both ways
- Try to upload proxy with and without AutoUploadproxy flag
- Ban a user
- Set specific authorization rules for a method of a service
- Remove a user and a group

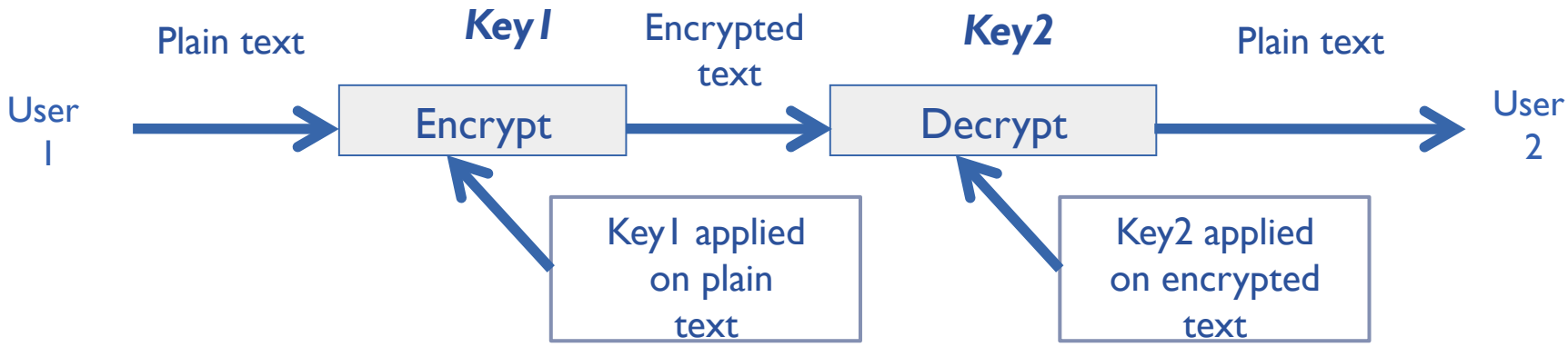
Last slide

- Question?

Backups

Key concept. Encryption

- Encryption is the process of transforming of information using an algorithm to make it unreadable to anyone except one with a key



- Algorithms
  - Symmetric: Key1 = Key2
  - Asymmetric: Key1 ≠ Key2

Asymmetric encryption

- Solving problem of key distribution in not-safe environment
- Each entity (machine, user) has two keys:
  - Public key
  - Private key
- Grid authentication is based on X.509 PKI (Public Key Infrastructure) standard
  - DIRAC uses X.509 PKI as well
- Advantages:
  - Public keys are safe to publish anywhere
  - Validity of a user can be easily proven. Could be used as digital signature
- Disadvantages:
  - Speed (Symmetric algorithms can be 10.000 times faster)