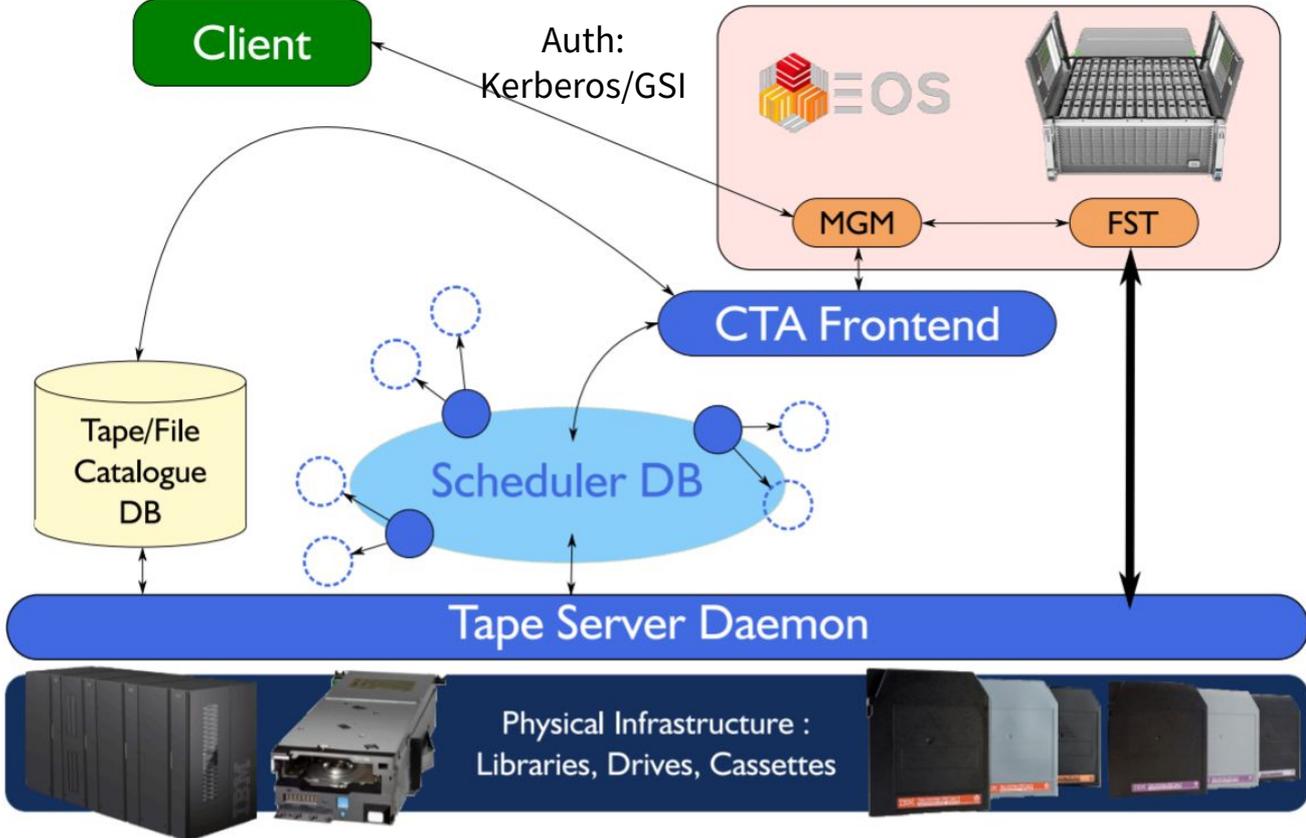


Configuring user access control in CTA

Volodymyr Yurchenko

CERN, IT Department, Storage Group

EOS+CTA architecture



Combination of 3 permission systems



UNIX permissions

Enhanced ACL

Mount rules

UNIX permissions

- **similar** to POSIX file-system permissions
- **read, write and browsing** defined by 'r'(4), 'w'(2), 'x'(1), e.g. 751 = 'rwxr-x--x'
- files **inherit** the permissions from the parent directory
 - if changed after creation, not automatically applied to the children
 - also the case for ACL

ACL

- directory or file level
 - extended attributes
sys.acl=<acllist>
- 3 types: **users, groups, e-groups**
- ‘!’ to invert
‘+’ to override ‘!’

r	grant read permission
w	grant write permission
x	grant browsing permission
c	grant change owner permission
m	grant change mode permission
d	allow deletion of files and directories
u	allow updates for files
p	prepare - for mounting tapes

ACL: note

- EOS relies on LDAP to check if an account is a member of an e-group
- Access model in EOS: most permissive combination of Unix and ACL

ACL example

```
# eos attr set sys.acl=  
u:124290:rwxp+d,  
g:1017:rxp,  
egroup:na61-cta:rxp,  
u:98119:rwxp,  
z:!u!d,  
u:0:+u  
"/eos/ctapublicdisk/archive/na61/"
```

ACL example

```
# eos attr set sys.acl=  
u:124290:rwxp+d,  
g:1017:rxp,  
egroup:na61-cta:rxp,  
u:98119:rwxp,  
z:!u!d,  
u:0:+u  
"/eos/ctapublicdisk/archive/na61/"
```

z:<acl> - rule applied to everyone

ACL example

```
# eos attr set sys.acl=  
u:124290:rwxp+d,  
g:1017:rxp,  
egroup:na61-cta:rxp,  
u:98119:rwxp,  
z:!u!d,  
u:0:+u  
"/eos/ctapublicdisk/archive/na61/"
```

user na61tdaq can delete files

ACL example

```
# eos attr set sys.acl=  
u:124290:rwxp+d,  
g:1017:rwp,  
egroup:na61-cta:rwp,  
u:98119:rwxp,  
z:!u!d,  
u:0:+u  
"/eos/ctapublicdisk/archive/na61/"
```

wj group and na61-cta e-group
can stage files

ACL management script

- Easily **change ACL** permissions and **add new users**
- eos-ns-inspect to get namespace dump
- ACL description in json files
- eos attr set
sys.acl=<ACL> <path>

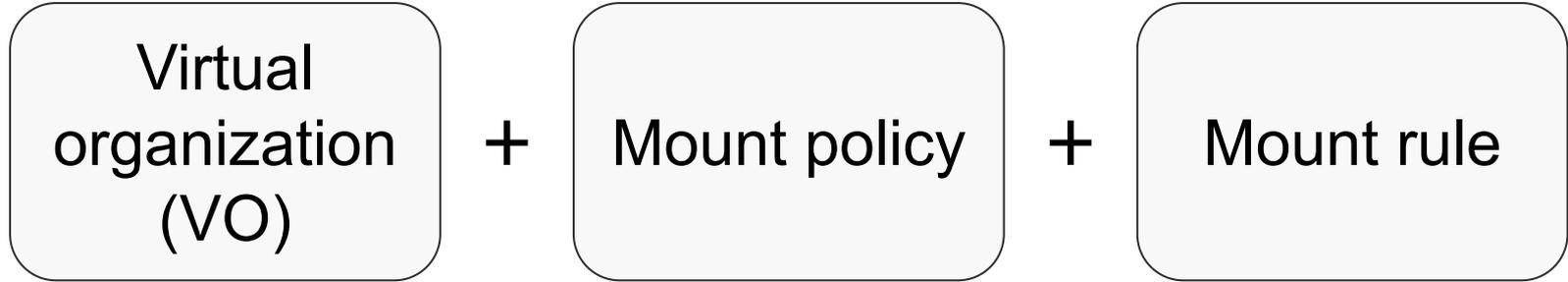
```
{
  "name_of_experiment": [
    {
      "directory": "path",
      "users": [ "service_account1", "service_account2" ],
      "unix_groups": [ ],
      "egroups": [ "egroup1" ],
      "write": [ "service_account1" ],
      "read": [ "service_account1", "service_account2", "egroup1" ],
      "prepare": [ "service_account1", "egroup1" ],
      "delete": [ "service_account1" ],
      "chmod": [ ],
      "chown": [ ]
    }
  ]
}
```

Invoking the script

```
# bash +x instance_set_acl.sh
cat 1646781764_DIRS_ACL.cmd | pv -l -s $(wc -l 1646781764_DIRS_ACL.cmd) |
xargs -d$'\n' -P50 -itoto bash -c 'toto'
```

```
# head 1646781764_DIRS_ACL.cmd
eos attr set sys.acl=u:42703:wxp+d,u:30065:rxp,u:98119:rxp,z:!u!d,u:0:+u
"/eos/ctapublic/archive/na62/user/lkrtpc/"
eos attr set sys.acl=u:42703:wxp+d,u:30065:rxp,u:98119:rxp,z:!u!d,u:0:+u
"/eos/ctapublic/archive/na62/user/lkrtpc/raw_data/"
```

Access to the tape resources



- maximum read drives
- maximum write drives

- minimum request age
- archive/retrieve priority

- user accounts

CTA mount rules

Requester
mount rule

Group
mount rule

OR

Instance default mount rule

Problems and solutions

File is readable when it shouldn't be. Can be caused by Unix permissions:

`rw-r--r--`

Solution. Add 'm' ACL flag to allow user set correct permissions

File owner can always delete it (undesirable in archive systems). This is granted by 'w' Unix permission: rw-----

Solution 1. Add 'm' ACL flag to allow user remove 'w' permission.

Solution 2. Change the owner after writing a file with 'c' ACL flag.

Summary

CTA access control:

- combination of EOS and CTA components
- interaction between them may be not obvious

Script and description of ACLs are in [operations repo](#). Feel free to contact us to get access to it:

cta-support@cern.ch

cta-community.web.cern.ch