Tokens are the new Proxies

CERN

EOS

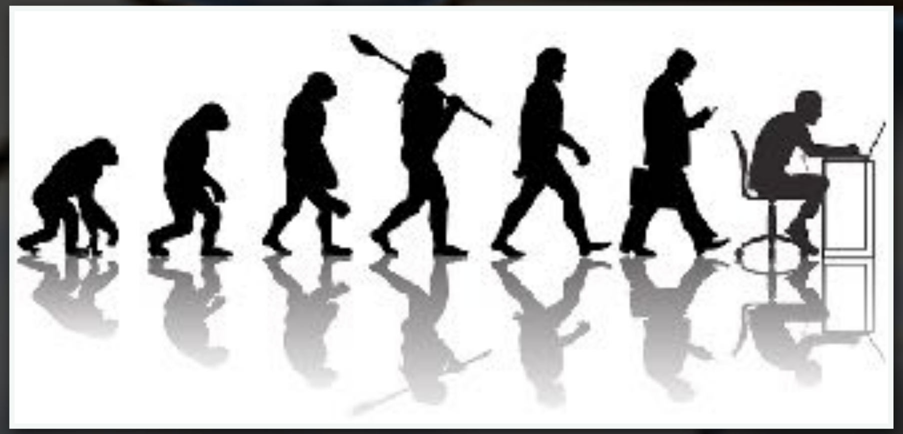
WORKSHOP

Andreas-Joachim Peters
CERN IT Storage Group

Overview

- **The (ab-)uses of tokens and the death of the proxy - not yet!**
- **EOS token support**

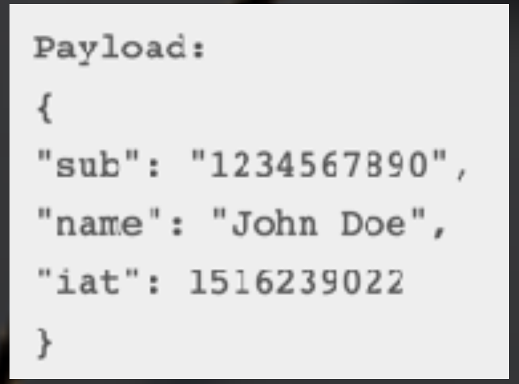
Authorization Evolution



- the old way: a user authenticates with kerberos or a certificate/proxy and the storage system checks if the user has access according to the ACLs defined and the authentication provided
- the new way: a user authenticates with an authorisation service and retrieves a token. The token tells the storage service if the user has access and where and in an obfuscated way … also who the person is …
  - for the moment the token model is envisaged only for file transfers via HTTPS protocol
- Why? Simplifies the problem of delegating credentials between services!

Token & Transport

- a token carries **authorisation information** for a target service
  - you might have heard about OAUTH2, OIDC, JWT, Sci & WLCG-Token, right?
    - in short: this is a signed JSON file carrying authorisation KV pairs



```
Payload:
{
"sub": "1234567890",
"name": "John Doe",
"iat": 1516239022
}
```

  - the token can be 'inspected' easily, the transport has to be secure, otherwise it is too easy to abuse it - HTTPS is a secure transport - XRootD4 protocol is not because path names, CGI etc. are not encrypted on the wire

Token & Transport

- in HTTPS protocol a token is transferred as an HTTP auth header or as a CGI KV <url>?auth=<>
- in XRootD protocol a token is transferred as CGI KV pair <file>?auth=<>
  - we need XRootD5 with request encryption to use tokens safely
- in WLCG the initial usage will just be one token for the whole accessible namespace e.g. for every request originating from one client the same token will be sent, although it is always the same
- **EOS** will use the *official* token plug-in of WLCG, no need to implement anything - see next presentation
- Nevertheless token are neat in particular if you extend their scope/concept slightly - like S3 signed URLs

EOS Tokens

Bearer Token Support

Proprietary format

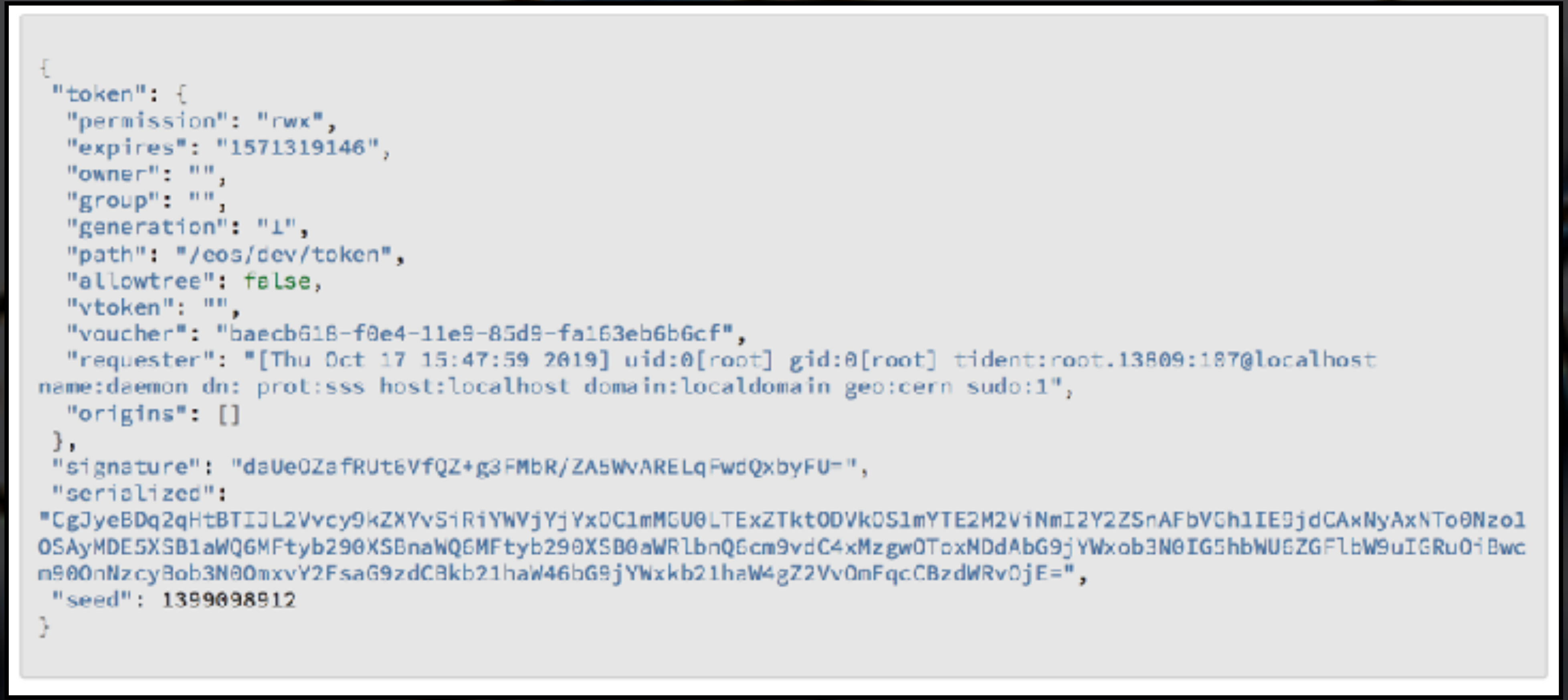
- Serialized **PROTOBUF** structure + **ZLIB** Compression + **Base64URL** encoding

Token carries

- a namespace scope file, directory or tree
- an ACL entry replacing locally stored ACLs - no need to invent new syntax like UPLOAD, DOWNLOAD…
- an optional role e.g. the owner when creating a file
- an optional set of origin restrictions - which clients can use this token and how do they have to be authenticated
  - we can enforce additional strong authentication if a bearer wants to use a token
- a generation value allows immediate token revocation of a given generation
- an expiration time

EOS Tokens

JSON representation



```
{
 "token": {
  "permission": "rwx",
  "expires": "1571319146",
  "owner": "",
  "group": "",
  "generation": "1",
  "path": "/eos/dev/token",
  "allowtree": false,
  "vtoken": "",
  "voucher": "baecb618-f0e4-11e9-85d9-fa163eb6b6cf",
  "requester": "[Thu Oct 17 15:47:59 2019] uid:0[root] gid:0[root] tident:root.13809:187@localhost
name:daemon dn: prot:sss host:localhost domain:localdomain geo:cern sudo:1",
  "origins": []
 },
 "signature": "daUeOZafRUt6VfQZ+g3FMbR/ZA5WvARELqFwdQxbyFU=",
 "serialized":
"CgJyeBDq2qHtBTIJL2Vvcy9kZXYvS1RiYWVjYjYxOC1mMGU0LTExZTktODVkOS1mYTE2M2ViNmI2Y2ZSnAFbVGh1IE9jdCAxNyAxNTo0Nzo1
OSAyMDE5XSB1aWQ6MFtyb290XSBnaWQ6MFtyb290XSB0aWRlbnQ6cm9vdC4xMzgwOToxNDdAbG9jYWxob3N0IG5hbWU6ZGFlbW9uIGRuOiBwc
m90OnNzcyBob3N0OmxvY2FsaG9zdCBkb21haW46bG9jYWxkb21haW4gZ2VvOmFqcCBzdWRvOjE=",
 "seed": 1399098912
}
```

Usage

token **as filename** or **CGI** authz=<token> usable with **XRootD, HTTP, GRPC, eos**xd **(fuse)**

```
# as a filename
xrdcp root://myeos//zteos64:MDAwMDAwNzR4nONS4WIuKq8Q-Dlz-ltWI3H91Pxi_cSsAv2S_OzUPP2SeAgtpMAY7f1e31Ts-od-
rgcLZ_a2_bhwcZO9cracy /tmp/

# via CGI
xrdcp "root://myeos//eos/myfile?authz=zteos64:MDAwMDAwNzR4nONS4WIuKq8Q-Dlz-
ltWI3H91Pxi_cSsAv2S_OzUPP2SeAgtpMAY7f1e31Ts-od+rgcLZ_a2_bhwcZO9cracy" /tmp/
```

Creation

```
eos token --path /eos/myfile --expires $LATER
zteos64:MDAwMDAwNzR4nONS4WIuKq8Q-Dlz-ltWI3H91Pxi~cSsAv2S~OzUPP2SeAgtpMAY7f1e31Ts-od-
rgcLZ~a2~bhwcZO9cracyhm1b3c6jpRIEWWOws71Ox6xAABeTC8I
```

EOS Tokens

How can they be used?

- usable by applications for restricted ***on-behalf* access**
  via any supported access method - even fuse mounts
- can be used by CERNBOX services to provide shares and delegate permissions
- as internal format for **external tokens** WLCG/ALICE tokens (probably obsolete)
- as **single file token** like signed S3 URLs are used

http://eos-docs.web.cern.ch/eos-docs/using/tokens.html