WLCG Token Transition Update

European HTCondor Workshop 2024

Brian Bockelman, 27 September 2024

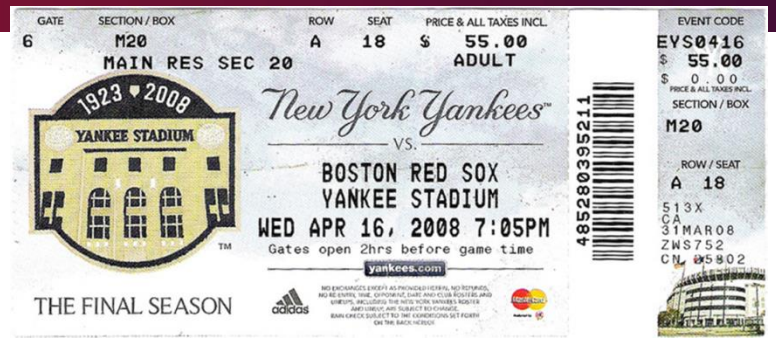
MORGRIDGE
INSTITUTE FOR RESEARCH

Authorization, authentication, and credentials: A recap

Three important terms for this presentation:

- **Authorization**: Deciding whether an entity is permitted to perform an action.
- **Authentication**: Mapping an entity to an identifier.
  - Note: *Authentication* is often part of an authorization scheme.
- **Credential**: knowledge that establishes a fact (e.g., identity).
  - Not too far off from the ‘credentials’ the university provides: a diploma establishes the bearer has particular knowledge.
  - Classic example: a username/password is used as a credential to perform authentication.

Moving from identity mapping to capabilities



- Authorization on WLCG was always based on identity mapping*:
  - A request was authenticated to a global identity.
  - The global identity was mapped to a local identity.
  - The request was authorized if the local identity was authorized to perform the action.

Scheme	Credentials	Authentication	Authorization
Gmail login	Password, 2FA	Username	Access to your inbox
Building access	ID card	Identity in HR database	Elevators
International Travel	Passport	Identity according to US Government	Enter Switzerland
Baseball Game	Ticket	**NONE!**	Sit in section 4, seat 34B
Workshop	Zoom URL	**NONE!**	Attend this wonderful talk!

Identity Mapping for Distributed Computing

- Each compute center had the concept of user accounts – a login to the local site that's unique for each user.
  - Suppose we had 10,000 physicists and 60 sites. To establish a login for each person at each site, we would need to manage 600,000 usernames and passwords. **Ouch!**
- Instead, the WLCG utilized the idea of **identity mapping**.
  - Each user established one **global identity**.
  - A mechanism would be developed to map the global identity to a local identity.
  - The user is authorized to perform the operations of the local identity.
- What credential do we use for establishing an identity? The X.509 certificate!
  - Since you don't want to hand out your identity to others, the X.509 "proxy" serves as your "power of attorney".

X.509

```
-----BEGIN CERTIFICATE-----
MIIIzjCCBragAwIBAgITbgBUyVBjktiAr/TUCAAAAFTJUDANBgkqhkiG9w0BAQ0F
ADBWMRIwEAYKCZImiZPyLGQBGRYCY2gxFDASBgoJkiaJk/IsZAEZFgRjZXJuMSow
KAYDVQQDEyFDRVJOIEdyaWQgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkwHhcNMjAw
ODE4MjA1NDI5WhcNMjEwOTIyMjA1NDI5WjCBlTESMBAGCgmSJomT8ixkARkWAmNo
MRQwEgYKCZImiZPyLGQBGRYEY2VybjEWMBQGA1UECxMNT3JnYW5pYyBVbml0czEO
MAwGA1UECxMFVXNlcnMxETAPBgNVBAMTCGJib2NrZWxtMQ8wDQYDVQQDEwY2NTk4
NjkxHTAbBgNVBAMTFEJyaWFuIFBhdWwgQm9ja2VsbWFuMIIBIjANBgkqhkiG9w0B
AQEFAAOCAQ8AMIIBCgKCAQEAi53glUNf5tPI4fslPo6HfFenSplb07sIa1fI2ugO
M72LGOswDHj65P1QMKEoh8YVhj8XW3jdnb3h45NxWiRqjRO7pRW9LY5LsVkmJwt0
0EHEM3VHnwjrhCjNpgg5aacYGUaD1d0deD91IAlSlKmBEoR5ZW/Y2meOymUzDVeV
/esj0BqN5VthJghoxC3AfVJICVHY4BcoHA2wlA/sx3B4jADxSihy9vTMOOcH9mbQ
MQ+8CTszlholbxLs2gh8MB0BCK04wPY4f4uP9ivOHf4FaNiYUcHAkQO8ftT0dTNs
QIu+OqwCIPMb/qIviX98+mgv5P7MJXC/k0gljjl9hz9kxQIDAQABo4IEUzCCBE8w
HQYDVR0OBBYEFE8pv/HJDn6OVikdQlP1MMAjZccDMB8GA1UdIwQYMBaAFKWg/WZY
/bndeuGynZ+j0eVQGJTnMIIBOAYDVR0fBIIBLzCCASswggEnoIIBI6CCAR+GTmh0
dHA6Ly9jYWZpbGVzLmNlcm4uY2gvY2FmaWxlcy9jcmwvQ0VSTiUyMEdyaWQlMjBD
ZXJ0aWZpY2F0aW9uJTIwQXV0aG9yaXR5LmNybIaBzGxkYXA6Ly8vQ049Q0VSTiUy
MEdyaWQlMjBDZXJ0aWZpY2F0aW9uJTIwQXV0aG9yaXR5LENOPUNFUk5QS0kwNSxD
Tj1DRFAsQ049UHVibGljJTIwS2V5JTIwU2VydmljZXMsQ049U2VydmljZXMsQ049
Q29uZmlndXJhdGlvbixEQz1jZXJuLERDPWNoP2NlcnRpZmljYXRlUmV2b2NhdGlv
bkxpc3Q/YmFzZT9vYmplY3RDbGFzcz1jUkxEaXN0cmlidXRpb25Qb2ludDCCAWIG
CCsGAQUFBwEBBIIBVDCCAVAwYwYIKwYBBQUHMAKGV2h0dHA6Ly9jYWZpbGVzLmNl
cm4uY2gvY2FmaWxlcy9jZXJ0aWZpY2F0ZXMvQ0VSTiUyMEdyaWQlMjBDZXJ0aWZp
Y2F0aW9uJTIwQXV0aG9yaXR5LmNydDCBwgYIKwYBBQUHMAKGgbVsZGFwOi8vL0NO
PUNFUk4lMjBHcmlkJTIwQ2VydGlmaWNhdGlvbiUyMEF1dGhvcml0eSxDTj1BSUEs
Q049UHVibGljJTIwS2V5JTIwU2VydmljZXMsQ049U2VydmljZXMsQ049Q29uZmln
dXJhdGlvbixEQz1jZXJuLERDPWNoP2NBQ2VydGlmaWNhdGU/YmFzZT9vYmplY3RD
bGFzcz1jZXJ0aWZpY2F0aW9uQXV0aG9yaXR5MCQGCCsGAQUFBzABhhhodHRwOi8v
b2NzcC5jZXJuLmNoL2Fjc3AwDgYDVR0PAQH/BAQDAgWgMD0GCSsGAQQBgjcVBwQw
MC4GJisGAQQBgjcVCIO90AmC7Y0Nhu2LK4He9TeFgNBiHoWK40yBtaoEAgFkAgEQ
MCkGA1UdJQQiMCAGCisGAQQBgjcKAwQGCCsGAQUFBwMEBggrBgEFBQcDAjA1Bgkr
BgEEAYI3FQoEKDAmMAwGCisGAQQBgjcKAwQwCgYIKwYBBQUHAwQwCgYIKwYBBQUH
AwIwSwYDVR0RBEQwQqAnBgorBgEEAYI3FAIDoBkMF2JyaWFuLmJvY2tlbG1hbkBj
ZXJuLmNogRdicmlhbi5ib2NrZWxtYW5AY2Vybi5jaDBEBgkqhkiG9w0BCQ8ENzA1
MA4GCCqGSIb3DQMCAgIAgDAOBggqhkiG9w0DBAICAIAwBwYFKw4DAgcwCgYIKoZI
hvcNAwcwJwYDVR0gBCAwHjAOBgwrBgEEAWAKBAICAwEwDAYKKoZIhvdMBQICATAN
BgkqhkiG9w0BAQ0FAAOCAgEAh4ebvYh1wrz287MrA+aIMVYx4fL/FJZybtqwairi
2FtG/WFqbv6YLyUXBST4zudn3k+UlRbCWdo8uI5wIusI2CkWZNnsGIEsMg3Ghm/u
hbZUDxVD+3qrr09IJaY2ZVhuz7a0eVtB9KMQ5A4y+k9rBLxGpea2sMjEHAOVlM2N
5kRL5j8NLfNlm9T2U/Ay9jgQLB2MARn0bOn0mH8CDAG2ryHYXsxdqoAXBIeUG4Xp
rbcYcNXym1XE5r/3htMhC88UBCsuMeLcQTPjq9C30+IGcVcHgpSfioESftpAlf6S
9j24joZiLg48bWHyT9AkEfaZujZLM1xMZSV//PPgpqlM0trQNXD24S3tUS8D1xlf
xoNae/BKuUxgcTvo4pqUdqOGNACYh8PMXtAsGaq2KZUXN/ZF2X16b2HLmHRjEp2t
vJvEh6ruYBIKGBHY8pdye60gmNvtNHHxDdYKYYrdpAfMUzrSAf4+vtgW1kqPyOhO
tiiIcd3qzfa95pFI6Kj/MULOIfkL61S1du7X2OM9thf4z8rCIaK3YApo3z0XFZ//
v8ruPL+l0f0A/T8D3BDYIQkJwfhCBl+1kSYIQ8bWJ/KHkrxi+XuXnU0T1OG4Q4nQ
PyNlfbgJhLzT7YUkcAPvg6xxITtSXyeRnYnjYwviaf54DLFCUZz8BzFDqEkWTF9m
/Q0=
-----END CERTIFICATE-----
```

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            6e:00:54:c9:50:63:92:d8:80:af:f4:d4:08:00:00:00:00:54:c9:50
    Signature Algorithm: sha512WithRSAEncryption
        Issuer: DC=ch, DC=cern, CN=CERN Grid Certification Authority
        Validity
            Not Before: Aug 18 20:54:29 2020 GMT
            Not After : Sep 22 20:54:29 2021 GMT
        Subject: DC=ch, DC=cern, OU=Organic Units, OU=Users, CN=bbockelm, CN=659869, CN=Brian Paul Bockelman
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:8b:9d:e0:95:43:5f:e6:d3:c8:e1:fb:25:3e:8e:
                    87:7c:57:a7:4a:99:5b:d3:bb:08:6b:57:c8:da:e8:
                    0e:33:bd:8b:18:eb:30:0c:78:fa:e4:fd:50:30:a1:
                    28:87:c6:15:86:3f:17:5b:78:dd:9d:bd:e1:e3:93:
                    71:5a:24:6a:8d:13:bb:a5:15:bd:2d:8e:4b:b1:59:
                    26:27:0b:74:d0:41:c4:33:75:47:9f:08:eb:84:28:
                    cd:a6:08:39:69:a7:18:19:46:83:d5:dd:1d:78:3f:
                    75:20:09:52:94:a9:81:12:84:79:65:6f:d8:da:67:
                    8e:ca:65:33:0d:57:95:fd:eb:23:d0:1a:8d:e5:5b:
                    61:26:08:68:c4:2d:c0:7d:52:48:09:51:d8:e0:17:
                    28:1c:0d:b0:94:0f:ec:c7:70:78:8c:00:f1:4a:28:
                    72:f6:f4:cc:38:e7:07:f6:66:d0:31:0f:bc:09:3b:
                    33:96:1a:25:6f:12:ec:da:08:7c:30:1d:01:08:ad:
                    38:c0:f6:38:7f:8b:8f:f6:2b:ce:1d:fe:05:68:d8:
                    98:51:c1:c0:91:03:bc:7e:d4:f4:75:33:6c:40:8b:
                    be:3a:ac:02:20:f3:1b:fe:a2:2f:89:7f:7c:fa:68:
                    2f:e4:fe:cc:25:70:bf:93:48:25:8e:39:7d:87:3f:
                    64:c5
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                4F:29:BF:F1:C9:0E:7E:8E:56:29:1D:42:53:F5:30:C0:23:65:C7:03
            X509v3 Authority Key Identifier:
                keyid:A5:A0:FD:66:58:FD:B9:DD:7A:E1:B2:9D:9F:A3:D1:E5:50:18:94:E7

            X509v3 CRL Distribution Points:

                Full Name:
                  URI:http://cafiles.cern.ch/cafiles/crl/CERN%20Grid%20Certification%20Authority.crl
                  URI:ldap:///CN=CERN%20Grid%20Certification%20Authority,CN=CERNPKI05,CN=CDP,CN=Public%20Key%

            Authority Information Access:
                CA Issuers - URI:http://cafiles.cern.ch/cafiles/certificates/CERN%20Grid%20Certification%20Au
                CA Issuers - URI:ldap:///CN=CERN%20Grid%20Certification%20Authority,CN=AIA,CN=Public%20Key%20
                OCSP - URI:http://ocsp.cern.ch/ocsp

            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
```

- X.509 was born out of the International Telecommunication Union's attempt to build a distributed global directory information tree in the 1980's.
  - The project never came to fruition.
  - However, it was a way to distributed to establish identity and associate cryptographic keys.
    - Which fit the needs for a company called Netscape when they wanted to secure ecommerce transactions in the browser.
    - And thus became a component of 'HTTPS', which is used everywhere today!

A critical piece of today's Global Economy relies on a failed attempt to build a phone book.

(with only a little exaggeration)

Group-based Mapping

- A conceptual leap occurred when sites realized they cared little about authorizing users but rather groups:
  - No need to know all 10,000 users and authorize them individually; rather, a site only cared that an identity was one of 5,000 members of a given experiment they supported
    - Example: every individual in the CMS experiment can store 1TB of data at my site.
  - Identities are only useful in determining whether two remote entities were the same individual.
    - As long as we know they belong to CMS, it is irrelevant if the username was “brian.bockelman” or “cms018”.
    - Sites still had a traceability requirement: need to determine all the actions “cms018” performed.

A_N(global ID) -> group

A_Z(group) -> P_1, P_2, …, P_N

Implementation: X.509 + VOMS

- Group mapping is implemented with the “Virtual Organization Management Services” (VOMS)
  - The VOMS server would sign a cryptographic extension to the user’s X.509 certificate or delegation.
    - Signing technology builds on X.509 attribute certificates.
  - This extension would assert group membership or roles.
    - A local mapfile would map the group / roles to a local identity. The rest proceeds as in identity mapping.
  - Note there’s no use of the user’s identity in this system!

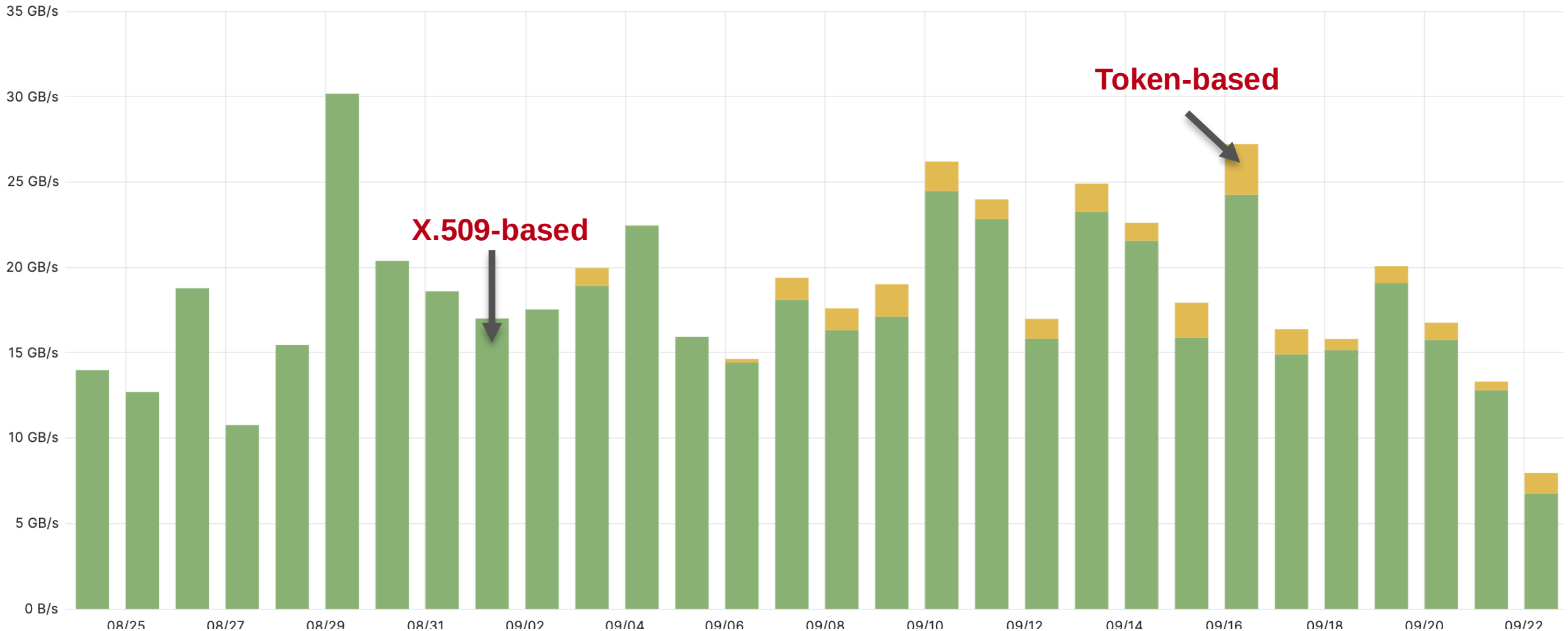
```
subject   : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=bbockelm/CN=659869/CN=Brian Paul Bockelman/CN=1761602861
issuer    : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=bbockelm/CN=659869/CN=Brian Paul Bockelman
identity  : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=bbockelm/CN=659869/CN=Brian Paul Bockelman
type      : RFC3820 compliant impersonation proxy
strength  : 2048
path      : /tmp/x509up_u1221
timeleft  : 11:59:54
key usage : Digital Signature, Key Encipherment
=== VO cms extension information ===
VO        : cms
subject   : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=bbockelm/CN=659869/CN=Brian Paul Bockelman
issuer    : /DC=ch/DC=cern/OU=computers/CN=voms2.cern.ch
attribute : /cms/Role=NULL/Capability=NULL
attribute : /cms/integration/Role=NULL/Capability=NULL
attribute : /cms/uscms/Role=NULL/Capability=NULL
timeleft  : 11:59:54
uri       : voms2.cern.ch:15002
```

Human-friendly printout of the VOMS extension

Overhauling Authorization for the WLCG

- Identities don't belong in the distributed authorization decisions (though needed for traceability).
- We want authorization to be based on the experiment's assertion, not the user identifier.
- We want to attenuate / reduce the credential's authorization as much as possible!
  - Principle of least authorization!
- Technology-wise, we wanted to get out of the Grid Security Infrastructure (GSI) business.
  - So did the Globus folks who wanted to push this in the first place!
- **Idea**:
  - Replace the X.509-based credential technology with more commonly-used JWTs (bearer tokens).
  - Replace identity-/group-mapping infrastructure with **capability-based**: credential encodes what you are allowed to do, not who you are!
- We took the idea and ran with it. For CMS:
  - Started developing the technology in ~2018 with the SciTokens project.
  - The WLCG profile (describing the authorization infrastructure) was published in 2020.
  - Converted the CEs to use tokens in 2021.
  - Majority of storage systems added support in 2023.
  - Peaked at ~50% data movement by volume used tokens for authorization in DC24.

CMS Production Transfers – last month



Tokens – more ways to reduce power!

- X.509 proxies were either “full power” or “limited” (with no clear definition of what “limited” met).
- Tokens have several dimensions to restrict the authorization:
  - **Time**: nbf (“**n**ot **b**e**f**ore”) / exp (“**exp**iration”): Restricts when the token’s validity begins and ends.
  - **Scopes**: Positive statements of “what can be done” with the token.
    - Compute-related: compute.read, compute.{modify,create,cancel}. Authorizes interaction with CE.
    - Storage-related: storage.read, storage.{create,modify}, storage.stage. Authorizes interaction with storage endpoint. Can be further restricted by prefix (storage.read:/simulations/A/2024/09/26)
  - **Audience**: Who is the token for?
    - Prevents a stolen token from host A to be used to send jobs to host B.

- Design tradeoffs: which of these mechanisms to use?
- Finer grained = more secure
  - … = more complexity = more load = more difficulty.

No free lunch!

What are the *downsides* of tokens?

- **Tyranny of choice**: multi-dimensional optimization problem versus one-bit. "Perfect is the enemy of the good".
- Switch from proof of possession to **bearer**: Full token is sent over the wire, meaning receiver has full power of token.
  - This happens with GSI for GridFTP and the Globus CE (automatic delegation).
  - The problem is sloppy software: headers and signed URLs were never redacted in our community. Lots of logging of secrets / credentials to files.
    - **It can be done: need to catch up with the rest of the world.**
- More **reliance on token issuer**:
  - Tokens focus on much shorter lifetimes.
  - Finer grained = Larger cardinality of tokens.
  - Token issuers tend to keep created tokens in a database.
- **No standardized CLI client / client library**.
  - Would prefer we *don't* have a standalone "token-init"-style client but each CLI client (rucio, CRAB) use a common library to get the token it needs.

Who is your token issuer?

- In X.509 / VOMS world, there was approximately one single VOMS server implementation.
  - VOMS-Admin was never ported to RHEL8+.
  - VOMS C++ implementation went from April 2021 to July 2024 without a production release.
- For tokens, there's a (relative) plethora of options:
  - **Indigo IAM**: Developed by INFN, used by the LHC experiments and at RAL. All-in-one solution, including authentication, group-management, and a **VOMS attribute authority**.
  - **OA4MP**: Developed by NCSA, deployed by CILogon, used by OSG services and FNAL/JLab experiments. Scriptable interface, *only* issues tokens (integrates with existing identity and group management solutions).
  - **EGI-CheckIn**: Implements AARC profile, meant to be single-instance issuer.
  - **KeyCloak**: Community project (CNCF); supported by RedHat. Powers CERN SSO.
    - With modest tweaks, could generate tokens following WLCG profile.

Where are things going? CMS

- I'm not qualified to give an overview of every LHC experiment.
- Within my wheelhouse: CMS
  - SAM tests thoroughly probe site services (CE, storage) for token support.
    - Missing support: STORM ☹
  - Will continuously expand list of sites using tokens with Rucio.
  - Web services now can use WLCG tokens
  - Job submission:
    - (Fall) Using "htgettoken"/Vault to distribute tokens to laptops / lxplus-type environments.
    - (Fall) HTCondor credmon to ensure valid token is always available in the job.
  - (Winter) Adapt workflow management tools to manage & use tokens.
  - (Fall/winter) "rucio upload", "rucio download" with tokens.

Let HTCondor manage your tokens

- You want to use tokens in your job, right?
  - **Don't** put an access token in transfer_input_files. The token will expire!
  - **Don't** put a refresh token in your job and create a new access token from the EP. Refresh token is too powerful!
  - **Do** let the AP manage your credentials!
    - The job will always have fresh, valid tokens, periodically updated.
- There are three credmons:
  - OAuth2 authorization code flow: Traditional OAuth2 flow, requires running a web server at the AP.
  - Vault: Receives tokens from a Vault host; eases pain of OAuth2 flows + multiple APs.
  - Local issuer: Requires signing key locally, creates token on the fly.
- <u>These are all mutually exclusive</u>!
- Up next for credmon:
  - Merge all code bases together – have as many instances as you would like!
  - Add support for client credential flow: skips need for user interaction if AP is a trusted OAuth2 client.
- Still missing: device code flow ☹

X.509, Redux

- X.509 is a credential type. How do we validate credentials?
  - **GSI** (Grid Security Infrastructure): A set of rules for validating credentials specific to the grid, including proxies, namespace verification. Implemented by the Grid Community Toolkit / Globus Toolkit.
  - TLS / **SSL** / Internet PKI: The more common, RFC-standardized set of rules for X.509 clients. No namespace verification, no custom transport layer. Implemented by OpenSSL.
- In HTCondor 23, we support TLS / SSL / Internet PKI.
  - Different protocol than prior “GSI” protocol
  - If VOMS support is enabled (23.5.x or later), then VOMS C++ library will be used to validate the client credentials. **Allows VOMS proxies**.
    - If you trust a library with fairly inactive development for your core security.

Want to learn how to enable SSL/VOMS authentication? How to authorize based on VOMS proxies?
See https://htcondor-wiki.cs.wisc.edu/index.cgi/wiki?p=HowToUseProxiesWithSsl

Outlook ahead

- The token transition is well on its way!
  - Not all experiments proceeding in lockstep but forward progress year-over-year.
  - Big use cases in production for big experiments (e.g., CMS data transfers).
- Services largely have support in place.
- “Scaling down” to smaller organizations is the current challenge, esp. with token issuers:
  - IAM is a good “all in one” choice if you were happy with VOMS-Admin.
  - OA4MP is a good option if you want to provide your own LDAP instead.
  - KeyCloak has lots of promise; looking for partners.
- Clients and client libraries are likely the next focus!

Questions?

This project is supported by the National Science Foundation under Cooperative Agreements OAC-2030508. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

FEARLESS SCIENCE

MORGRIDGE INSTITUTE FOR RESEARCH