From Identity-Based Authorization to Capabilities: SciTokens, JWTs, and OAuth

Jim Basney
jbasney@ncsa.illinois.edu

HTCondor Workshop Autumn 2020
25 September 2020

Goals for an HTC Authorization System

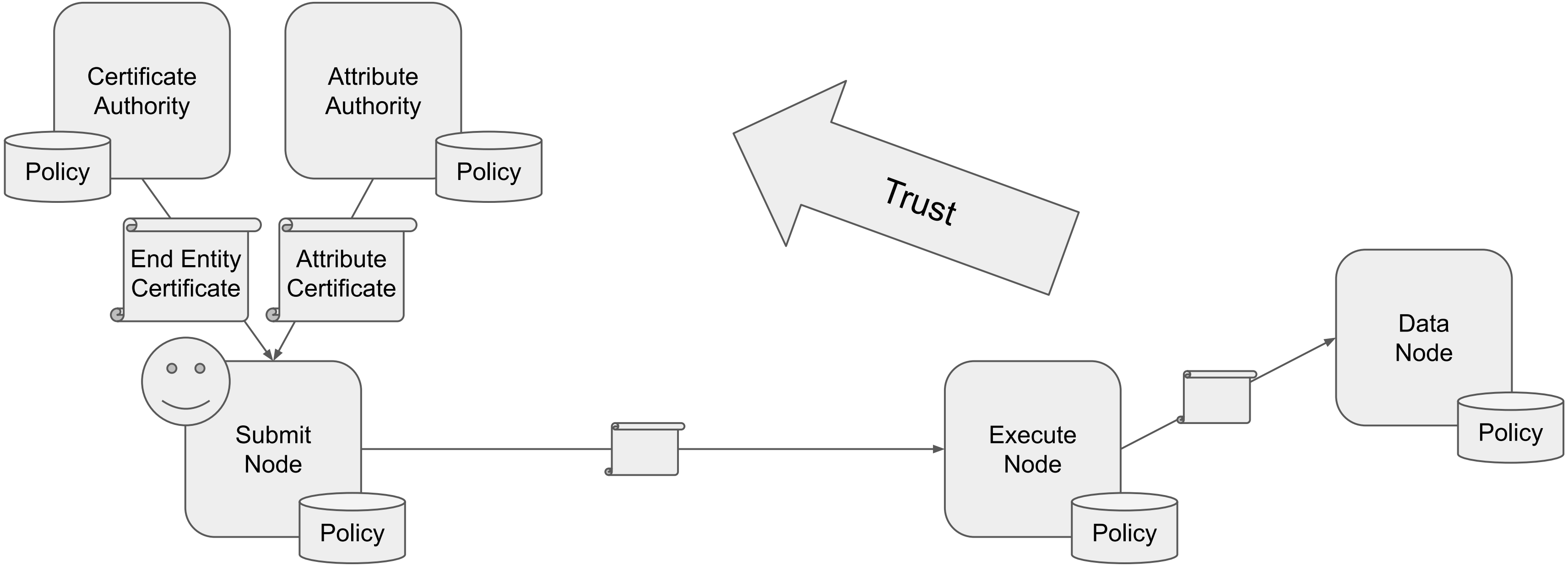
- Enable access to HTC!

- Implement appropriate resource/data access policies
- Ease of use
- Manageability
- Distributed/Decentralized

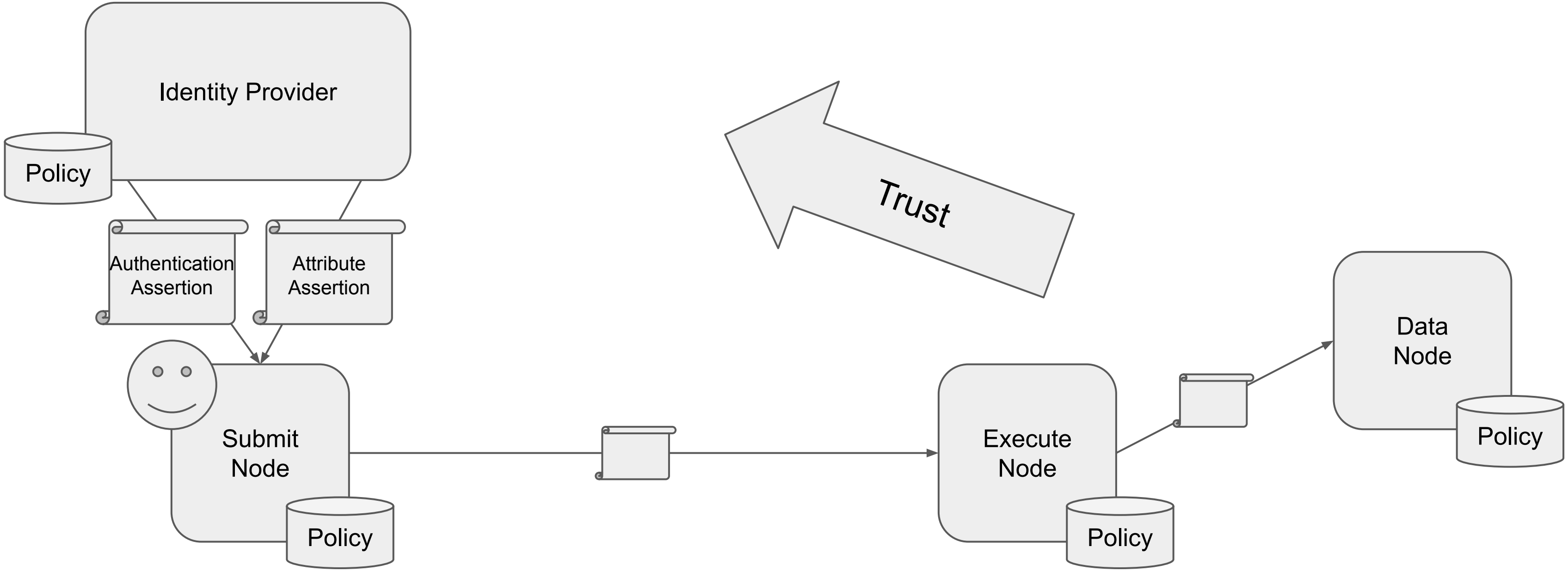
Authentication & Authorization Standards

- X.509: Certificates
  - Grid Security Infrastructure (GSI)
  - Virtual Organization Membership Service (VOMS)
- SAML: Security Assertion Markup Language
  - Using XML
  - Single Sign-on for Higher Education: eduGAIN / InCommon / Shibboleth
- JWT: JSON Web Tokens
  - Using JavaScript Object Notation (JSON)
  - Pronounced "jot"
  - Digitally signed, self-describing security tokens
- ★ OAuth: Authorization Framework
  - Optionally using JWTs
  - Tokens for limited access to resources
- OIDC: OpenID Connect
  - An identity layer on top of OAuth
  - Using JWTs

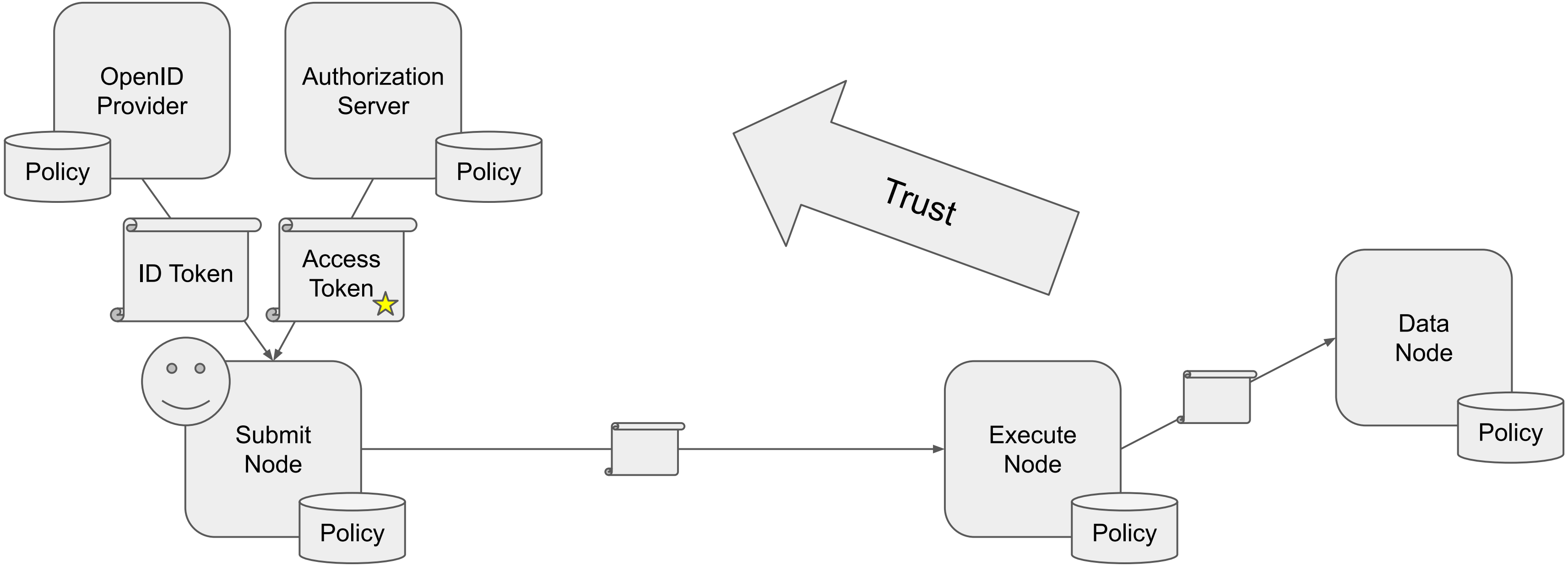
X.509


Certificate Authority
Policy
Attribute Authority
Policy
End Entity Certificate
Attribute Certificate
Trust
Submit Node
Policy
Execute Node
Policy
Data Node
Policy

SAML


Identity Provider
Policy
Authentication Assertion
Attribute Assertion
Trust
Submit Node
Policy
Execute Node
Policy
Data Node
Policy

JWT / OIDC / OAuth



Credentials for Authentication / Authorization

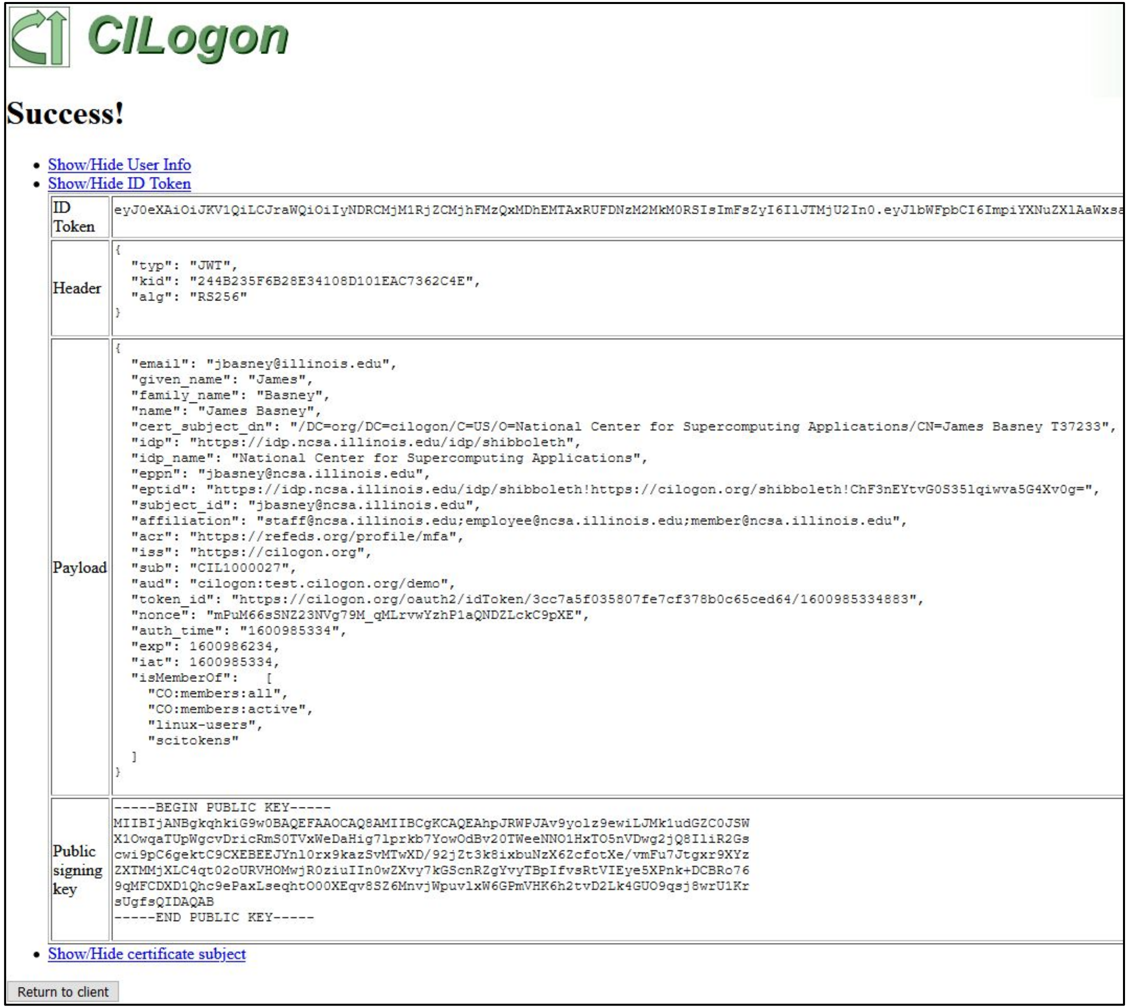
	X.509	SAML	OIDC	OAuth / JWT
Credential Issuer	Certificate Authority	Identity Provider	OpenID Provider	Authorization Server
Credential Verifier	Relying Party	Service Provider	Relying Party	Resource Server
Credential	Certificate	Assertion	ID Token	Access Token
Language	ASN.1	XML	JSON	JSON
Credential Contents	Distinguished Names / Fully Qualified Attribute Names	Attributes	Claims	Claims
User Identifier	Subject DN	NameID / eduPersonPrincipalName	Subject Identifier (sub) Claim	Subject (sub) Claim
Managing Trust	CA Certificate Bundle	SAML Metadata	OpenID Provider Metadata	Authorization Server Metadata

Authorization / Access Control

		Supported By			
		X.509	SAML	OIDC	OAuth
Identity-based	User identifiers and access control lists	YES	YES	YES	
Attribute-based	Access policies based on user attributes	YES	YES	YES	
Role-based	Access controls based on group memberships and roles	YES	YES	YES	
Capability-based	Tokens allow actions on resources				YES

OIDC JWT Demo

Log on to https://demo.cilogon.org/ with your campus identity provider or use your GitHub, Google, or ORCID account.



Success!

- Show/Hide User Info
- Show/Hide ID Token

ID Token

```
eyJ0eXAiOiJKV1QiLCJraWQiOiIyNDRCMjM1RjZCMjhFMzQxMDhEMTAxRUFDNzM2MkM0RSIsImFsZyI6IlJTMjU2In0.eyJlbWFpbCI6ImpiYXNuZXlAaWxsa
```

Header

```
{
  "typ": "JWT",
  "kid": "244B235F6B28E34108D101EAC7362C4E",
  "alg": "RS256"
}
```

Payload

```
{
  "email": "jbasney@illinois.edu",
  "given_name": "James",
  "family_name": "Basney",
  "name": "James Basney",
  "cert_subject_dn": "/DC=org/DC=cilogon/C=US/O=National Center for Supercomputing Applications/CN=James Basney T37233",
  "idp": "https://idp.ncsa.illinois.edu/idp/shibboleth",
  "idp_name": "National Center for Supercomputing Applications",
  "eppn": "jbasney@ncsa.illinois.edu",
  "eptid": "https://idp.ncsa.illinois.edu/idp/shibboleth!https://cilogon.org/shibboleth!ChF3nEYtvG0S35lqiwva5G4Xv0g=",
  "subject_id": "jbasney@ncsa.illinois.edu",
  "affiliation": "staff@ncsa.illinois.edu;employee@ncsa.illinois.edu;member@ncsa.illinois.edu",
  "acr": "https://refeds.org/profile/mfa",
  "iss": "https://cilogon.org",
  "sub": "CIL1000027",
  "aud": "cilogon:test.cilogon.org/demo",
  "token_id": "https://cilogon.org/oauth2/idToken/3cc7a5f035807fe7cf378b0c65ced64/1600985334883",
  "nonce": "mPuM66sSNZ23NVg79M_qMLrvwYzhP1aQNDZLckC9pXE",
  "auth_time": "1600985334",
  "exp": 1600986234,
  "iat": 1600985334,
  "isMemberOf":   [
    "CO:members:all",
    "CO:members:active",
    "linux-users",
    "scitokens"
  ]
}
```

Public signing key

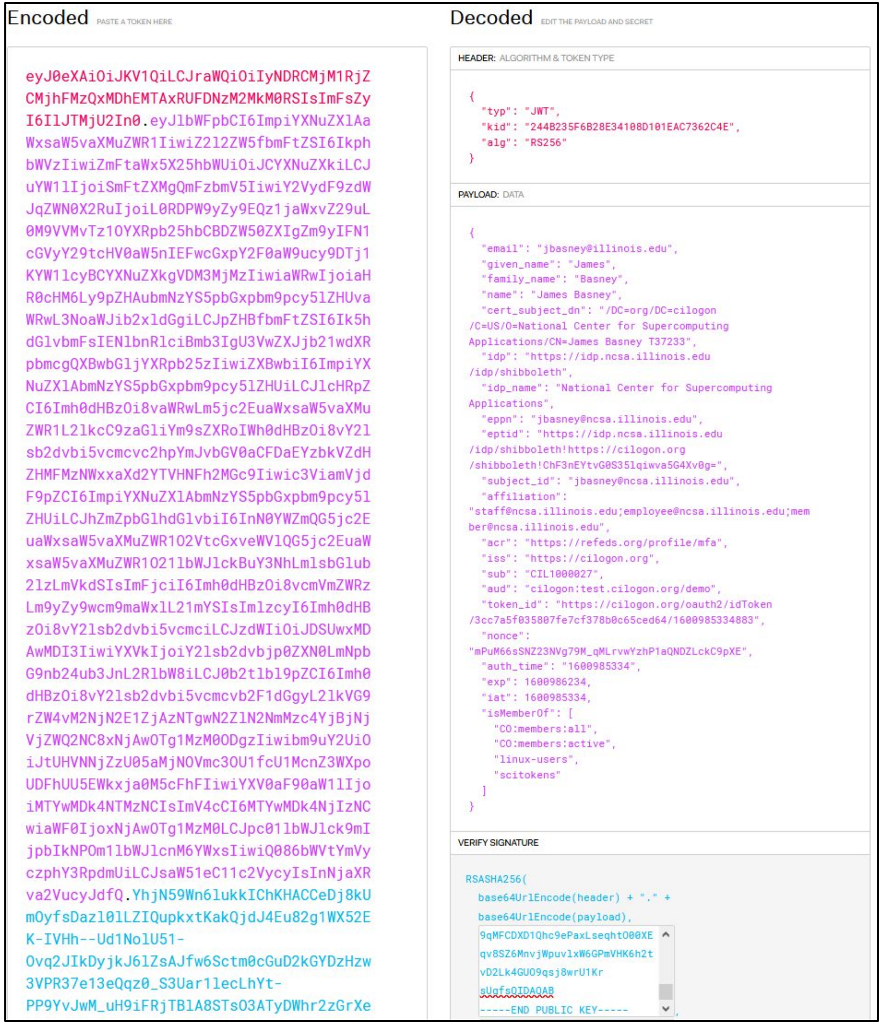
```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAhpJRWPJAv9yolz9ewiLJMk1udGZC0JSW
X1OwqaTUpWgcvDricRmS0TVxWeDaHig7lprkb7YowOdBv20TWeeNNO1HxTO5nVDwg2jQ8IliR2Gs
cwi9pC6gektC9CXEBEEJYnl0rx9kazSvMTwXD/92jZt3k8ixbuNzX6ZcfotXe/vmFu7Jtgxr9XYz
ZXTMMjXLC4qt02oURVHOMwjR0ziuIIn0wZXvy7kGScnRZgYvyTBpIfvsRtVIEye5XPnk+DCBRo76
9qMFCDXD1Qhc9ePaxLseqhtO00XEqv8SZ6MnvjWpuvlxW6GPmVHK6h2tvD2Lk4GUO9qsj8wrU1Kr
sUgfsQIDAQAB
-----END PUBLIC KEY-----
```

- Show/Hide certificate subject

Return to client

OIDC JWT Demo

Paste the ID Token and Public Key into https://jwt.io/ to verify it.



Encoded PASTE A TOKEN HERE

```
eyJ0eXAiOiJKV1QiLCJraWQiOiIyNDRCMjM1RjZ
CMjhFMzQxMDhEMTAxRUFDNzM2MkM0RSIsImFsZy
I6IlJTMjU2In0.eyJlbWFpbCI6ImpiYXNuZXlAa
WxsaW5vaXMuZWR1IiwiZ2l2ZW5fbmFtZSI6Ikph
bWVzIiwiZmFtaWx5X25hbWUiOiJCYXNuZXkiLCJ
uYW1lIjoiSmFtZXMgQmFzbmV5IiwiY2VydF9zdW
JqZWN0X2RuIjoiL0RDPW9yZy9EQz1jaWxvZ29uL
0M9VVMvTz1OYXRpb25hbCBDZW50ZXIgZm9yIFN1
cGVyY29tcHV0aW5nIEFwcGxpY2F0aW9ucy9DTj1
KYW1lcyBCYXNuZXkgVDM3MjMzIiwiaWRwIjoiaH
R0cHM6Ly9pZHAubmNzYS5pbGxpbm9pcy5lZHUva
WRwL3NoaWJib2xldGgiLCJpZHBfbmFtZSI6Ik5h
dGlvbmFsIENlbnRlciBmb3IgU3VwZXJjb21wdXR
pbmcgQXBwbGljYXRpb25zIiwiZXBwbiI6ImpiYX
NuZXlAbmNzYS5pbGxpbm9pcy5lZHUiLCJlcHRpZ
CI6Imh0dHBzOi8vaWRwLm5jc2EuaWxsaW5vaXMu
ZWR1L2lkcC9zaGliYm9sZXRoIWh0dHBzOi8vY2l
sb2dvbi5vcmcvc2hpYmJvbGV0aCFDaEYzbkVZdH
ZHMFMzNWxxaXd2YTVHNFh2MGc9Iiwic3ViamVjd
F9pZCI6ImpiYXNuZXlAbmNzYS5pbGxpbm9pcy5l
ZHUiLCJhZmZpbGlhdGlvbiI6InN0YWZmQG5jc2E
uaWxsaW5vaXMuZWR1O2VtcGxveWVlQG5jc2EuaW
xsaW5vaXMuZWR1O21lbWJlckBuY3NhLmlsbGlub
2lzLmVkdSIsImFjciI6Imh0dHBzOi8vcmVmZWRz
Lm9yZy9wcm9maWxlL21mYSIsImlzcyI6Imh0dHB
zOi8vY2lsb2dvbi5vcmciLCJzdWIiOiJDSUwxMD
AwMDI3IiwiYXVkIjoiY2lsb2dvbjp0ZXN0LmNpb
G9nb24ub3JnL2RlbW8iLCJ0b2tlbl9pZCI6Imh0
dHBzOi8vY2lsb2dvbi5vcmcvb2F1dGgyL2lkVG9
rZW4vM2NjN2E1ZjAzNTgwN2ZlN2NmMzc4YjBjNj
VjZWQ2NC8xNjAwOTg1MzM0ODgzIiwibm9uY2UiO
iJtUHVNNjZzU05aMjNOVmc3OU1fcU1McnZ3WXpo
UDFhUU5EWkxja0M5cFhFIiwiYXV0aF90aW1lIjo
iMTYwMDk4NTMzNCIsImV4cCI6MTYwMDk4NjIzNC
wiaWF0IjoxNjAwOTg1MzM0LCJpc01lbWJlck9mI
jpbIkNPOm1lbWJlcnM6YWxsIiwiQ086bWVtYmVy
czphY3RpdmUiLCJsaW51eC11c2VycyIsInNjaXR
va2VucyJdfQ.YhjN59Wn6lukkIChKHACCeDj8kU
mOyfsDazl0lLZIQupkxtKakQjdJ4Eu82g1WX52E
K-IVHh--Ud1NolU51-
Ovq2JIkDyjkJ6lZsAJfw6Sctm0cGuD2kGYDzHzw
3VPR37e13eQqz0_S3Uar1lecLhYt-
PP9YvJwM_uH9iFRjTBlA8STsO3ATyDWhr2zGrXe
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "typ": "JWT",
  "kid": "244B235F6B28E34108D101EAC7362C4E",
  "alg": "RS256"
}
```

PAYLOAD: DATA

```
{
  "email": "jbasney@illinois.edu",
  "given_name": "James",
  "family_name": "Basney",
  "name": "James Basney",
  "cert_subject_dn": "/DC=org/DC=cilogon
/C=US/O=National Center for Supercomputing
Applications/CN=James Basney T37233",
  "idp": "https://idp.ncsa.illinois.edu
/idp/shibboleth",
  "idp_name": "National Center for Supercomputing
Applications",
  "eppn": "jbasney@ncsa.illinois.edu",
  "eptid": "https://idp.ncsa.illinois.edu
/idp/shibboleth!https://cilogon.org
/shibboleth!ChF3nEYtvG0S35lqiwva5G4Xv0g=",
  "subject_id": "jbasney@ncsa.illinois.edu",
  "affiliation":
"staff@ncsa.illinois.edu;employee@ncsa.illinois.edu;mem
ber@ncsa.illinois.edu",
  "acr": "https://refeds.org/profile/mfa",
  "iss": "https://cilogon.org",
  "sub": "CIL1000027",
  "aud": "cilogon:test.cilogon.org/demo",
  "token_id": "https://cilogon.org/oauth2/idToken
/3cc7a5f035807fe7cf378b0c65ced64/1600985334883",
  "nonce":
"mPuM66sSNZ23NVg79M_qMLrvwYzhP1aQNDZLckC9pXE",
  "auth_time": "1600985334",
  "exp": 1600986234,
  "iat": 1600985334,
  "isMemberOf": [
    "CO:members:all",
    "CO:members:active",
    "linux-users",
    "scitokens"
  ]
}
```

VERIFY SIGNATURE

```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  9qMFCDXD1Qhc9ePaxLseqhtO00XE
  qv8SZ6MnvjWpuvlxW6GPmVHK6h2t
  vD2Lk4GUO9qsj8wrU1Kr
  sUgfsQIDAQAB
  -----END PUBLIC KEY-----,
```

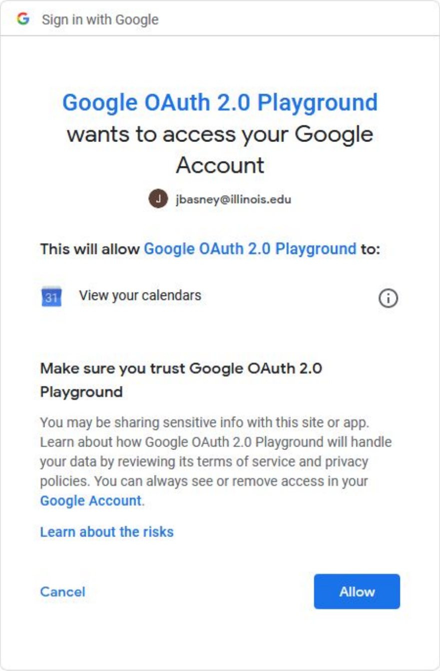
Least Privilege Authorization

- Good security practice: grant only those privileges that are required
  - for only as long as they are required

- Identity-based authorization
  - Limit the privileges granted to an identity
- Attribute-based authorization
  - Use attributes to determine appropriate privileges at this time
- Role-based authorization
  - Assign privileges to roles, and activate roles only when needed
- Capability-based authorization
  - Issue tokens granting only those privileges that are required, for the required lifetime

OAuth and Least Privilege

- OAuth Access Token "scope" identifies specific actions that are authorized on resources in the token "aud" (audience)
- OAuth obtains consent from the resource owner prior to token issuance
- OAuth clients <u>should</u> request only those "scope" values that are required



Sign in with Google

Google OAuth 2.0 Playground wants to access your Google Account

jbasney@illinois.edu

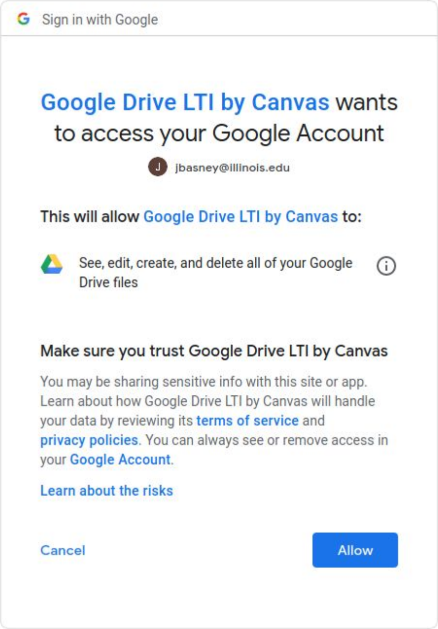
This will allow Google OAuth 2.0 Playground to:

View your calendars

Make sure you trust Google OAuth 2.0 Playground

You may be sharing sensitive info with this site or app. Learn about how Google OAuth 2.0 Playground will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your Google Account.

Learn about the risks



Sign in with Google

Google Drive LTI by Canvas wants to access your Google Account

jbasney@illinois.edu

This will allow Google Drive LTI by Canvas to:

See, edit, create, and delete all of your Google Drive files

Make sure you trust Google Drive LTI by Canvas

You may be sharing sensitive info with this site or app. Learn about how Google Drive LTI by Canvas will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your Google Account.

Learn about the risks

SCI TOKENS

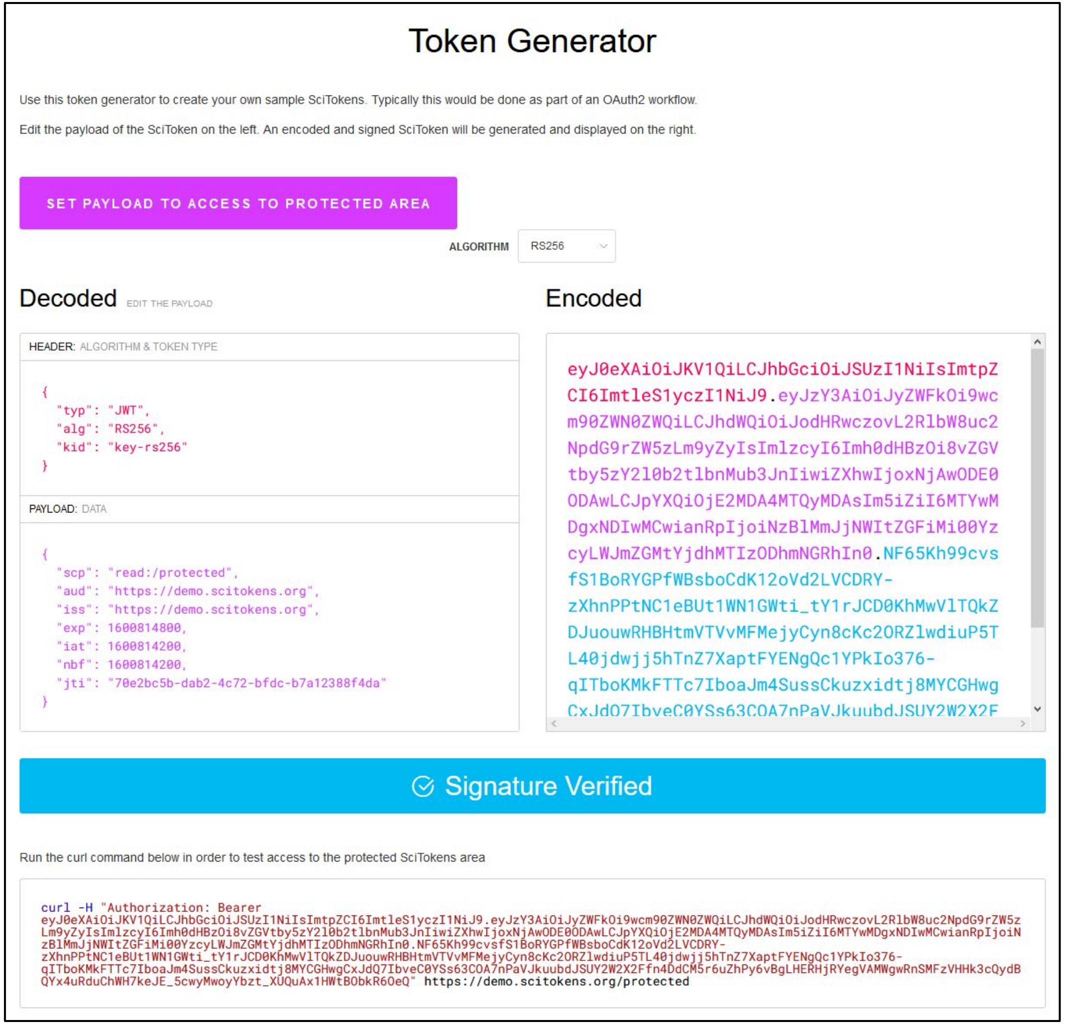
- Developing a capabilities-based authorization infrastructure for distributed scientific computing
- Using the OAuth and JWT standards for distributed authorization
- Implementing the Principle of Least Privilege
- Visit https://www.scitokens.org/ for specifications, publications
- Visit https://github.com/scitokens for open source implementations

SciTokens JWT Demo

Visit https://demo.scitokens.org/ and click the "Set Payload" button.

Try the curl command.

* Note: This demo implements an early draft specification that used "scp" instead of "scope".



Token Generator

Use this token generator to create your own sample SciTokens. Typically this would be done as part of an OAuth2 workflow.

Edit the payload of the SciToken on the left. An encoded and signed SciToken will be generated and displayed on the right.

ALGORITHM: RS256

Decoded

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "typ": "JWT",
  "alg": "RS256",
  "kid": "key-rs256"
}
```

PAYLOAD: DATA

```
{
  "scp": "read:/protected",
  "aud": "https://demo.scitokens.org",
  "iss": "https://demo.scitokens.org",
  "exp": 1600814800,
  "iat": 1600814200,
  "nbf": 1600814200,
  "jti": "70e2bc5b-dab2-4c72-bfdc-b7a12388f4da"
}
```

Encoded

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZ
CI6ImtleS1yczI1NiJ9.eyJzY3AiOiJyZWFkOi9wc
m90ZWN0ZWQiLCJhdWQiOiJodHRwczovL2RlbW8uc2
NpdG9rZW5zLm9yZyIsImlzcyI6Imh0dHBzOi8vZGV
tby5zY2l0b2tlbnMub3JnIiwiZXhwIjoxNjAwODE0
ODAwLCJpYXQiOjE2MDA4MTQyMDAsIm5iZiI6MTYwM
DgxNDIwMCwianRpIjoiNzBlMmJjNWItZGFiMi00Yz
cyLWJmZGMtYjdhMTIzODhmNGRhIn0.NF65Kh99cvs
fS1BoRYGPfWBsboCdK12oVd2LVCDRY-
zXhnPPtNC1eBUt1WN1GWti_tY1rJCD0KhMwVlTQkZ
DJuouwRHBHtmVTVvMFMejyCyn8cKc2ORZlwdiuP5T
L40jdwjj5hTnZ7XaptFYENgQc1YPkIo376-
qITboKMkFTTc7IboaJm4SussCkuzxidtj8MYCGHwg
CxJdQ7IbveC0YSs63COA7nPaVJkuubdJSUY2W2X2F
```

Run the curl command below in order to test access to the protected SciTokens area

```
curl -H "Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImtleS1yczI1NiJ9.eyJzY3AiOiJyZWFkOi9wcm90ZWN0ZWQiLCJhdWQiOiJodHRwczovL2RlbW8uc2NpdG9rZW5z
Lm9yZyIsImlzcyI6Imh0dHBzOi8vZGVtby5zY2l0b2tlbnMub3JnIiwiZXhwIjoxNjAwODE0ODAwLCJpYXQiOjE2MDA4MTQyMDAsIm5iZiI6MTYwMDgxNDIwMCwianRpIjoiN
zBlMmJjNWItZGFiMi00YzcyLWJmZGMtYjdhMTIzODhmNGRhIn0.NF65Kh99cvsfS1BoRYGPfWBsboCdK12oVd2LVCDRY-
zXhnPPtNC1eBUt1WN1GWti_tY1rJCD0KhMwVlTQkZDJuouwRHBHtmVTVvMFMejyCyn8cKc2ORZlwdiuP5TL40jdwjj5hTnZ7XaptFYENgQc1YPkIo376-
qITboKMkFTTc7IboaJm4SussCkuzxidtj8MYCGHwgCxJdQ7IbveC0YSs63COA7nPaVJkuubdJSUY2W2X2Ffn4DdCM5r6uZhPy6vBgLHERHjRYegVAMWgwRnSMFzVHHk3cQydB
QYx4uRduChWH7keJE_5cwyMwoyYbzt_XUQuAx1HWtBObkR6OeQ" https://demo.scitokens.org/protected
```

Implementing Standards

- RFC 6749: OAuth 2.0 Authorization Framework
  - token request, consent, refresh
- RFC 7519: JSON Web Token (JWT)
  - self-describing tokens, distributed validation
- RFC 8414: OAuth 2.0 Authorization Server Metadata
  - token signing keys, policies, endpoint URLs
- RFC 8693: OAuth 2.0 Token Exchange
  - token delegation, drop privileges (reduce "scope")
- draft-ietf-oauth-access-token-jwt: JWT Profile for OAuth 2.0 Access Tokens
  - authorization claims using JWT "scope" and "aud"

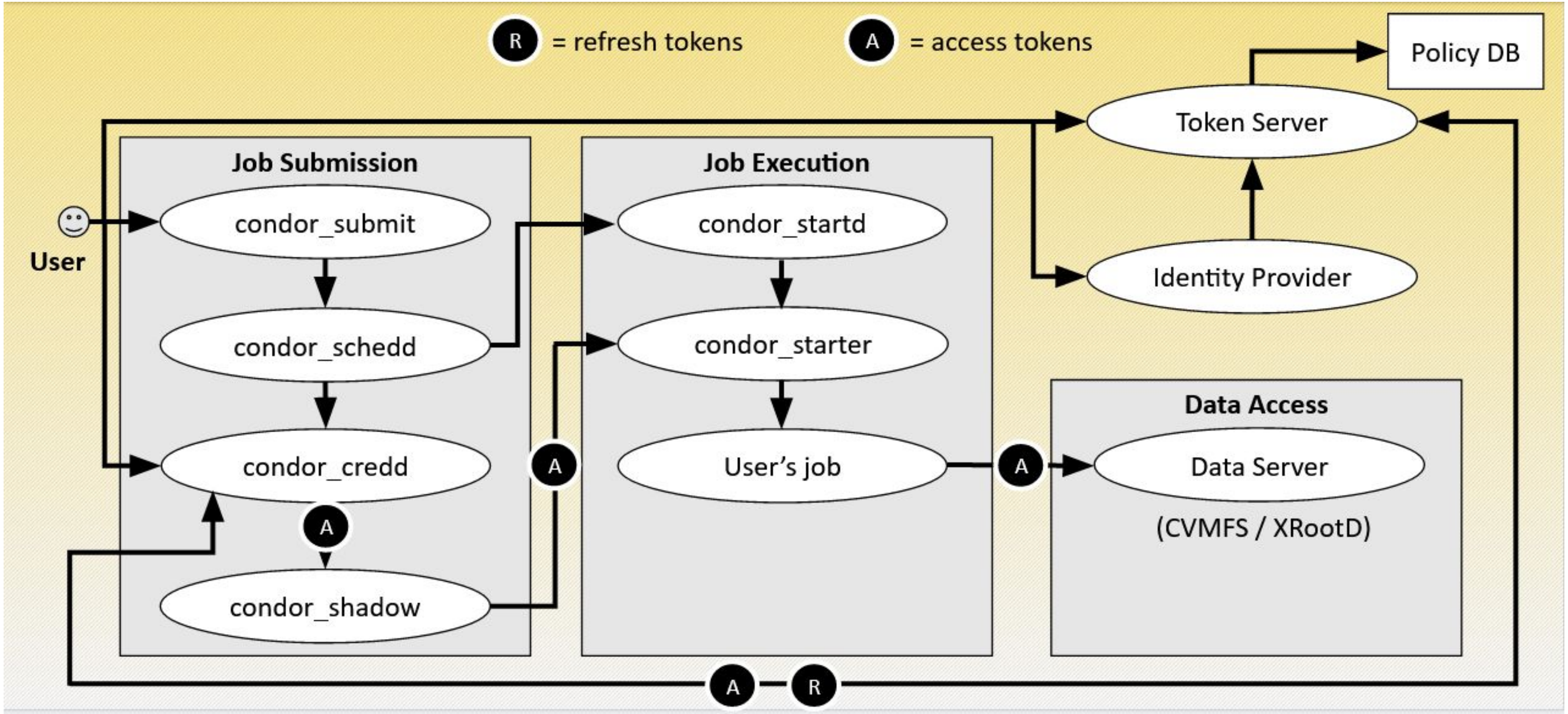
Implementing WLCG Common JWT Profiles

- Defines profiles for Group Based Authorization (wlcg.groups) and Capability Based Authorization (scope)
- Use cases:
  a. Identity Token with Groups
  b. Access Token with Groups
  c. Access Token with Authorization Scopes
- SciTokens supports and helped define use case (c)

https://doi.org/10.5281/zenodo.3460257

https://github.com/WLCG-AuthZ-WG

SciTokens & HTCondor



OAuth support in HTCondor is not just for SciTokens…

See next talk for details...

Thanks!

Questions?

Contact: jbasney@ncsa.illinois.edu

SciTokens Project Team:
Alex Withers, Brian Bockelman, Derek Weitzel, Duncan Brown, Jason Patton, Jeff Gaynor, Jim Basney, Todd Tannenbaum, You Alex Gao, and Zach Miller

This material is based upon work supported by the National Science Foundation under Grant No. 1738962. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.