mytoken – secure, long-lived tokens for you everywhere!

Gabriel Zachmann, Morris Baumgarten-Egemole, Marcus Hardt

Jun 2023

Use Case

Long running compute

(Longer than lifetime of Access Token)

1. Load data at the beginning
2. Run computation, access other resources in between
3. Store data at the end

Problem:

- Access Tokens expire during the job
- Access Tokens cannot be revoked
- Access Tokens are difficult to be limited
- Emerging Antipattern:
  - Long-lived Access Tokens

Goals

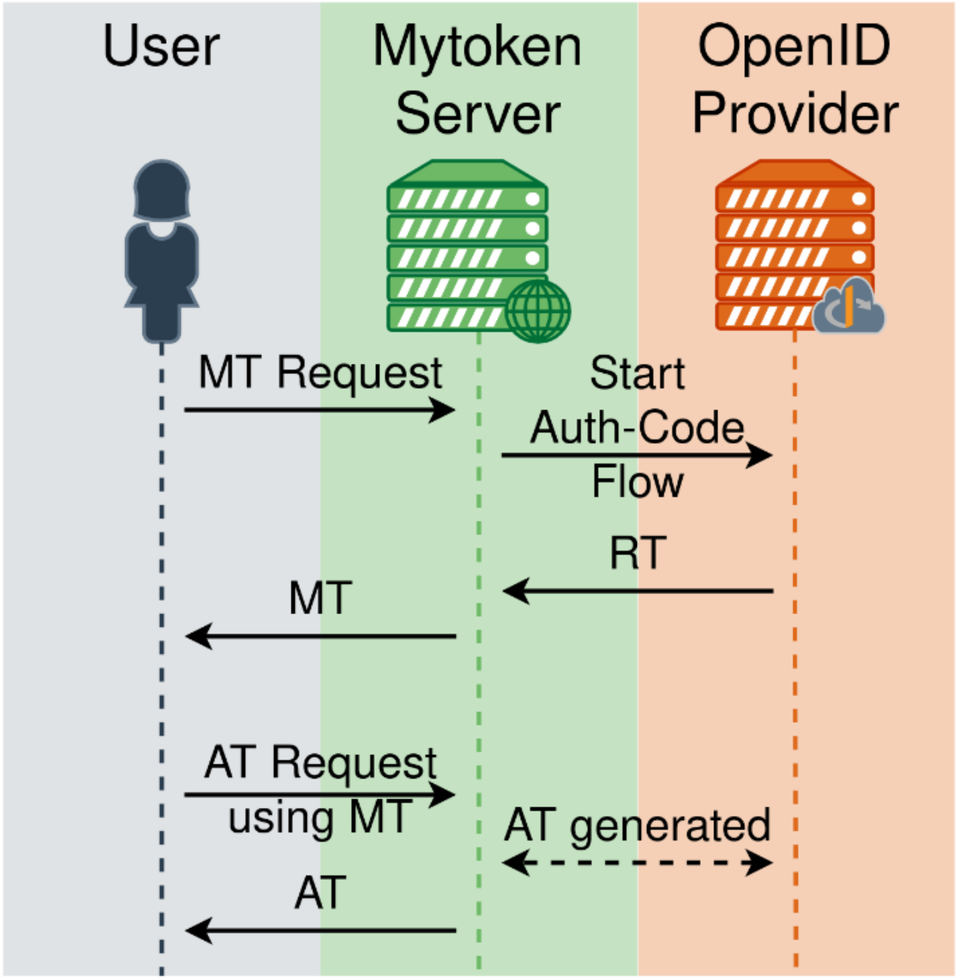
- Ensure availability of Access Tokens at any time
  - Allow short AT lifetime
  - No user interaction
- Address security adequately
- Add fine grained permissions to improve safety
- Support for multiple devices (Bearer Tokens)

Concept

Basics

Mytokens are a new class of tokens

- **Mytoken Server**
  - Proxy for Refresh Tokens (RT)
  - Implemented similarly to OIDC
- User flow:
  1. Create mytoken (MT)
  2. Use MT to obtain
     - Access Tokens (AT)
     - Other mytokens



Adequate security

Is this even possible?

- Mytoken allows any bearer of a MT to use a stored Refresh Token
- This is a breach in security
- Unless...
  1. We limit the power of mytokens => Introduce Capabilities and Restrictions
  2. We employ additional security m => Rotation, counting, serverside voodoo
  3. We apply proper crypto => Encrypt refresh token set with mytoken
- Certainly more secure than giving t Only the user has it!

(e.g. https://aai.egi.eu/token)

NEW: Profiles

- Make it simpler to do it right

Future: pass online-exam for advanced profiles

(read: longer token lifetimes)

Capabilities

Define the actions allowed for a mytoken

- Examples:
  - Obtain Access Token
  - Create new mytoken
  - List / introspect mytokens

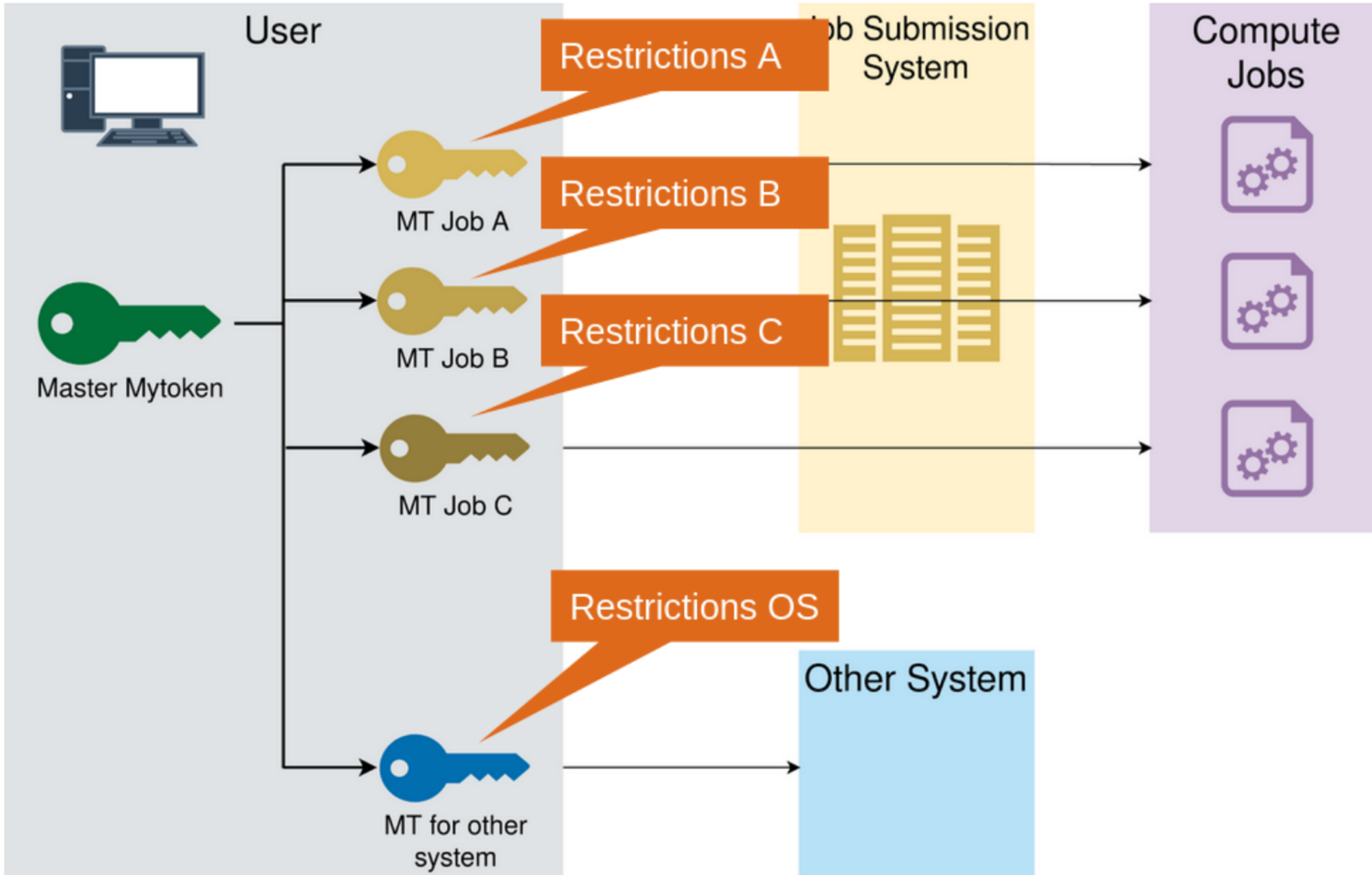
Restrictions

Reduce damage in case of loss

- Restrictions are flexible
  - Extensible restriction dimensions
  - Multiple restictions per token
- Typical restrictions:
  - Time, Location (IP, geoIP)
  - OIDC: `scope`, `audience`
  - Number of Usages

Hierarchies

- Mytokens can be used to create
  - Mytokens (with less privileges)
  - Access Tokens
- Only the mytoken is needed for this
- Typical scenario:
  - One “Master” mytoken on personal device
  - Subtokens for sending out to the infrastructure


User
Restrictions A
Job Submission System
Compute Jobs
MT Job A
Restrictions B
Master Mytoken
MT Job B
Restrictions C
MT Job C
Restrictions OS
Other System
MT for other system

Example case

Long running job

1. At Job start:
   - Only from Belgium Obtain AT that can
     - Create compute jobs
     - On Federated Cloud
   - Obtain an AT that can
     - Read storage
     - from CERN
2. No action allowed for some time
3. At Job end:
   - Obtain AT than can
     - Store the output
     - at DESY


```
[{"exp"         :1634300000,
  "nbf"         :1634400000,
  "geoip_allow":["BE"],
  "scope"       :"compute.create",
  "audience"    :["fedcloud"]
},{
  "exp"         :1634300000,
  "nbf"         :1634400000,
  "geoip_allow":["BE"],
  "scope"       :"storage.read",
  "audience"    :["se.cern.ch"]
},{
  "exp"         :1635300000,
  "nbf"         :1635400000,
  "scope"       :"storage.write",
  "audience"    :["dcache.desy.de"]}]
```

Creating mytokens

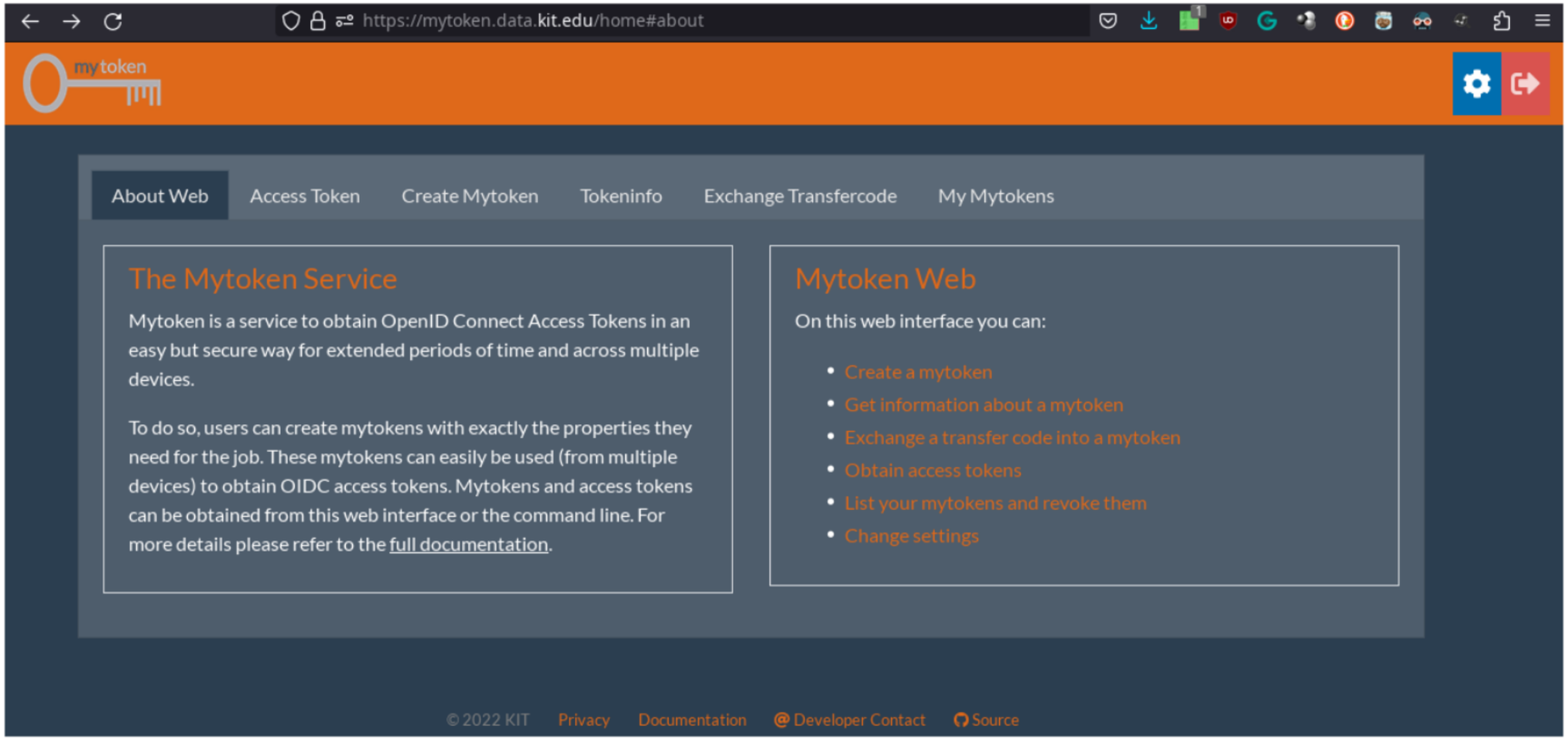
Options

1. Command line
2. Web Interface https://mytoken.data.kit.edu
3. **NEW: Mytoken APP**
4. Existing Mytoken
    - Create sub-token from mytoken
        - Specific capabilities
        - Individual restrictions

- Btw: Transfer-Codes allow transferring token to different device
- Btw: `oidc-agent` also has support for mytoken

Web Interface

https://mytoken.data.kit.edu



Mytoken App

- https://git.scc.kit.edu/m-team/oidc/mytoken-app
- APK: http://repo.data.kit.edu/devel/android/

The Future

Mytoken developments

- Host mytoken server in the “credential store” Environment
  - Secured environment
  - Based on well defined policy
- Fault tolerance
  - Looking at RCauth as an example
- Smart ways of giving powerful tokens to the right people only
- Support for OIDC-FED

Summary

`mytoken` - Summary

- Mytokens offer one way of securing the infrastructure
  1. By allowing short-lived Access Tokens
  2. By being revocable
  3. With rich set of **capabilities** and **restrictions**
- Capabilities and Restrictions
  - Offer fine-grained control
  - Adequate balance for security vs usability
    - Potential security course for extended profiles
- https://mytoken.data.kit.edu

Links

Demo instance

https://mytoken.data.kit.edu

Documentation

https://mytoken-docs.data.kit.edu

Github

https://github.com/oidc-mytoken

Contact

m-contact@lists.kit.edu