mytoken

mytoken – secure, long-lived tokens for you

everywhere!

Gabriel Zachmann, Xavier Mol, Marcus Hardt

Jun 2023

`mytoken`

Recap `mytoken`

- Goal: Ensure availability of Access Tokens **at any time**
  - Allow short AT lifetime
  - No user interaction
  - On any remote machine (Bearer Tokens)
  - But **secure**
- Just like **`oidc-agent`**, but mostly serverside
  - (i.e. Refresh Token stored serverside)
  - Access protected via **`mytoken`** token (which is given to the user)
- **`mytoken`** tokens are essentially “full of policies”
  - Capabilities
  - Restrictions

Basics

Skipped

For concept and security of the tokens itself

- See slides and documentation
- Important Updates:
  - **Profiles**
    - Make it simpler to do it right
  - **OP may pass attributes for advanced profiles**
    - read: OP controls which users may obtain longer token lifetimes
  - **Notifications**
    - Receive emails or subscribe to calendar feeds

Security (of the hosted mytok.eu service)

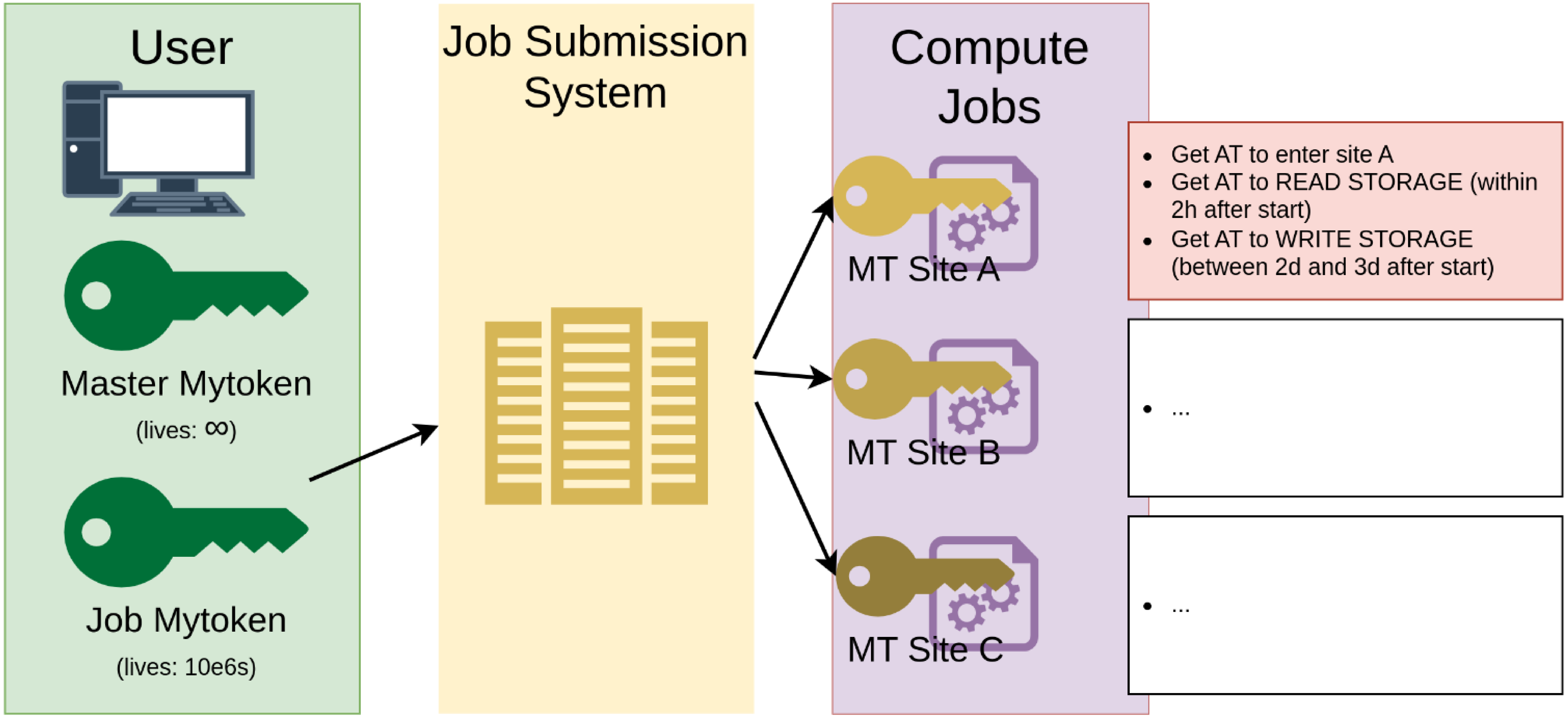
- https://mytok.eu is operated inside a **Credential Store**
    - Definition by eugridpma
    - Secured environment (locked room, logged access, secured cabinet, ...)
    - Pro active logging, system updates, ...
    - Incident response, audits, traceability, ...
- Equivalent to requirements of RCAuth-**Masterportal**
- Exceeds security requirements of some IdPs

“Eternal Life”

- **`Refresh Tokens`** often have a limited lifetime
- Goal: Reduce human interaction to “once” (create, and maybe: revoke)
- Suggested technical workaround:
  - Use **short lived** (1 month) `Refresh Tokens`
  - Enable **“Token Rotation”** for `Refresh Tokens`
    - New RT on every use
    - Legitimate user will notice, if token was used elsewhere
      - RT can timely be revoked
  - **Renew lifetime** for every new RT
  - Unused tokens **expire automatically** in this scenario
- Addresses use cases such as “monitoring” or “ci/cd”

Hierarchies

- mytokens can be used to create
  - Other 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



- Get AT to enter site A
- Get AT to READ STORAGE (within 2h after start)
- Get AT to WRITE STORAGE (between 2d and 3d after start)

- ...

- ...

Creating mytokens

1. Command line
2. Web Interface https://mytok.eu
3. Mytoken APP (Android + Apple OS)
4. Existing Mytoken
   - Create sub-token from mytoken
     - Specific capabilities
     - Individual restrictions

- Btw: Transfer-Codes allow transferring tokens between device
- Btw: `oidc-agent` also has support for mytoken

Using mytokens

1. Commandline: `mytoken` client command
2. Commandline: fancy `curl` command
3. Web

Links

- Web Interface https://mytok.eu
- App: http://repo.data.kit.edu/devel/android/

Use Cases

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

Monitoring

Check health of token-based infrastructure services. Also: CI/CD style cases.

1. Obtain long living `mytoken`
2. Regularly use `mytoken` to obtain `Access Token` to test services

=> Boils down to requirement “eternal life”

Links

Demo instance

https://mytoken.data.kit

Documentation

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

Github

https://github.com/oidc-mytoken

Contact

m-contact@lists.kit.ed