WaTTS service to provide proxy certificates

Demo

Marcus Hardt, Bas Wegh, Uros Stevanovic

Athens
21 March 2017
WaTTS

- OIDC TTS service
- KIT in-house development, Erlang
- Plugin based, easily extendable
- Enables functionality according to attributes and attributes’ LOA
- Already provides many services
  - SSH
  - S3
  - X.509
- Many Python plugins already available
- Open source, Apache 2.0
- Last demo: WaTTS ssh plugin to deploy keys on OKEANOS
WaTTS

- OIDC TTS service
- KIT in-house development, Erlang
- Plugin based, easily extendable
- Enables functionality according to attributes and attributes’ LOA
- Already provides many services
  - SSH
  - S3
  - X.509
- Many Python plugins already available
- Open source, Apache 2.0
- Last demo: WaTTS ssh plugin to deploy keys on OKEANOS
- New development → x.509 IOTA certificates using RCauth
Goals

- OIDC flow, with OIDC providers (EGI, HBP, B2ACCESS...)
- “User uses its OIDC token to get a IOTA proxy certificate”
- New certificate is obtained from RCauth only when needed (or has expired)
- Simple interface (with web frontend) to receive (VOMS) proxy certificate
- CLI access to get a proxy certificate (with OIDC access token)
- Using initial OIDC authentication to get a proxy, not the one used for RCauth
  - Although, if supported by RCauth, one can use the same
Current implementation

- End-Entity Certificate (~ 11 days)
- Long Lived Proxy (~ 11 days)
- Short Lived Proxy (~ 12 hours)

Diagram showing the flow of communication and authentication processes within the AARC project, including interactions with the VO Portal, VOMS server, Credential Store, and Delegation Server.
WaTTS implementation

- End-Entity Certificate (~11 days)
- Long Lived Proxy (~11 days)
- Short Lived Proxy (~12 hours)

On going:

- VOMS Server
- Credential Store (MyProxy Server)
- WaTTS
- X509 RCauth plugin

Delegation Server

- Online Certificate Authority (Myproxy Server)

OpenID Connect

WAYF

FedID eduGAIN
Key differences (vs. Master Portal)

• Differences (worth mentioning)
  • WaTTS is not an OIDC provider, but a service
    => WaTTS support many OIDC providers
  • WaTTS uses a new flow for proxy certificate distribution
    • Web client
    • CLI interface supported
  • WaTTS is a “one stop shop” for token translation
  • SSH, S3, x509..
  • https://watts-dev.data.kit.edu
  • https://github.com/urost/cli-get-proxy

• Future and on-going development
  • Integration with B2Access and EGI-CheckIn
    • Possibly within EOSC-Hub?