



INDIGO - DataCloud

RIA-653549

WaTTS

Bas Wegh, Marcus Hardt

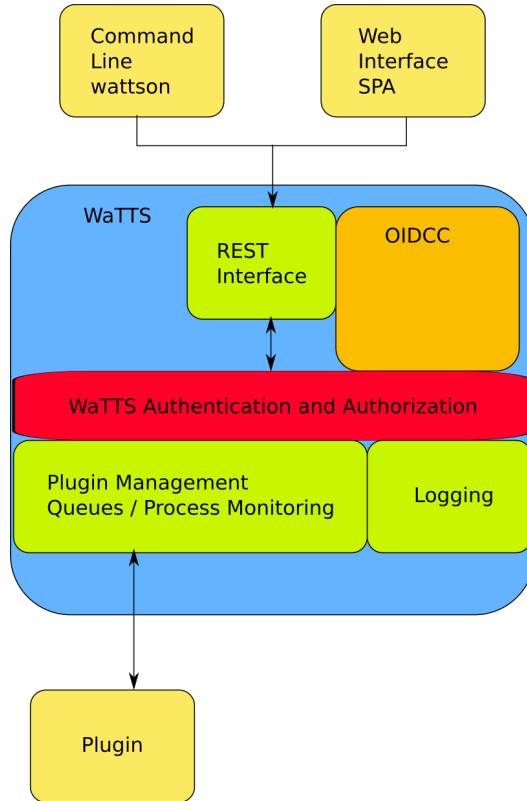
WaTTS Introduction

- Problem: Authentication to any service
 - End-Users live in different FIMs (e.g EduGain, Google, ...)
 - Services support various authentication schemes
 - A **simple, generic** solution is yet missing
- Following the AARC Blueprint Architectures (BPA) we designed a
 - Two-Level Credential Translation Scheme
 - 1. INDIGO IAM (or even checkin, b2access) → OpenID Connect
 - 2. WaTTS: OpenId Connect → Service specific scheme



INDIGO - DataCloud

WaTTS Architecture



■ External Clients

- Command line e.g. wattson or curl
- Web Interface JS SPA
- Using REST interface

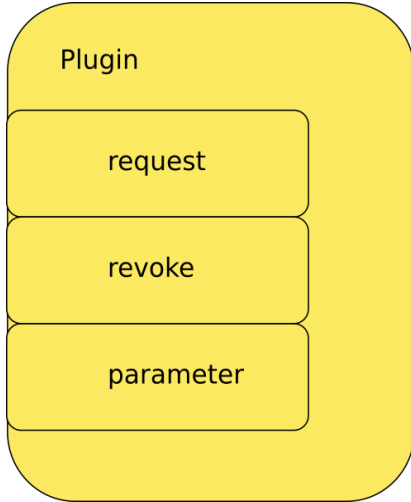
■ Authentication and Authorization layer

- All data must pass this layer
- Verification of data

■ Plugin Management

- Handle execution of plugins

Service scheme → Plugin



- Plugin
 - Highly customizable by the administrator
 - Any kind of executable (bash, python, binary)
 - Must implement three **operations**
- Operation 'request'
 - Actual translation step
 - OpenID Connect → Service specific scheme
 - Returns a 'state', the only information stored at WaTTS.
 - (Sometimes 'abused' for account-provisioning)
- Operation 'revoke'
 - Removal or revocation of the credential
 - Gets passed the 'state' from the request operation
- Operation 'parameter'
 - Configuration of the plugin at startup of WaTTS
 - Consists of two parameter sets
 - Configuration parameter
 - Per 'request' parameter



INDIGO - DataCloud

SSH Plugin

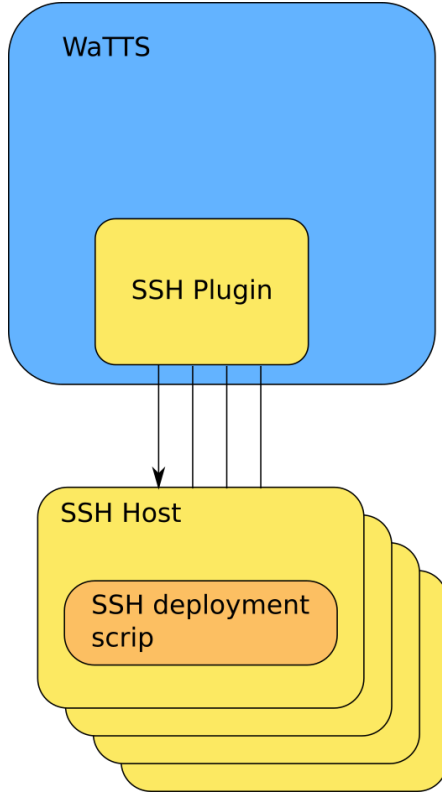
■ SSH Plugin

- SSH public key of user gets provisioned to a list of hosts
- The public key of the user ...
 - Is uploaded to WaTTS by the user (preferred)
 - Is generated (together with private key) by plugin and returned to user

■ SSH possible deployments

- Direct connection of plugin to machine via ssh and running of a local script, passing the key (easy to implement, first try, works)
- Use dedicated service to push keys to the hosts (e.g. ARGO pub/sub service)
- Utilize the ssh-CA concept (ongoing)

SSH Plugin






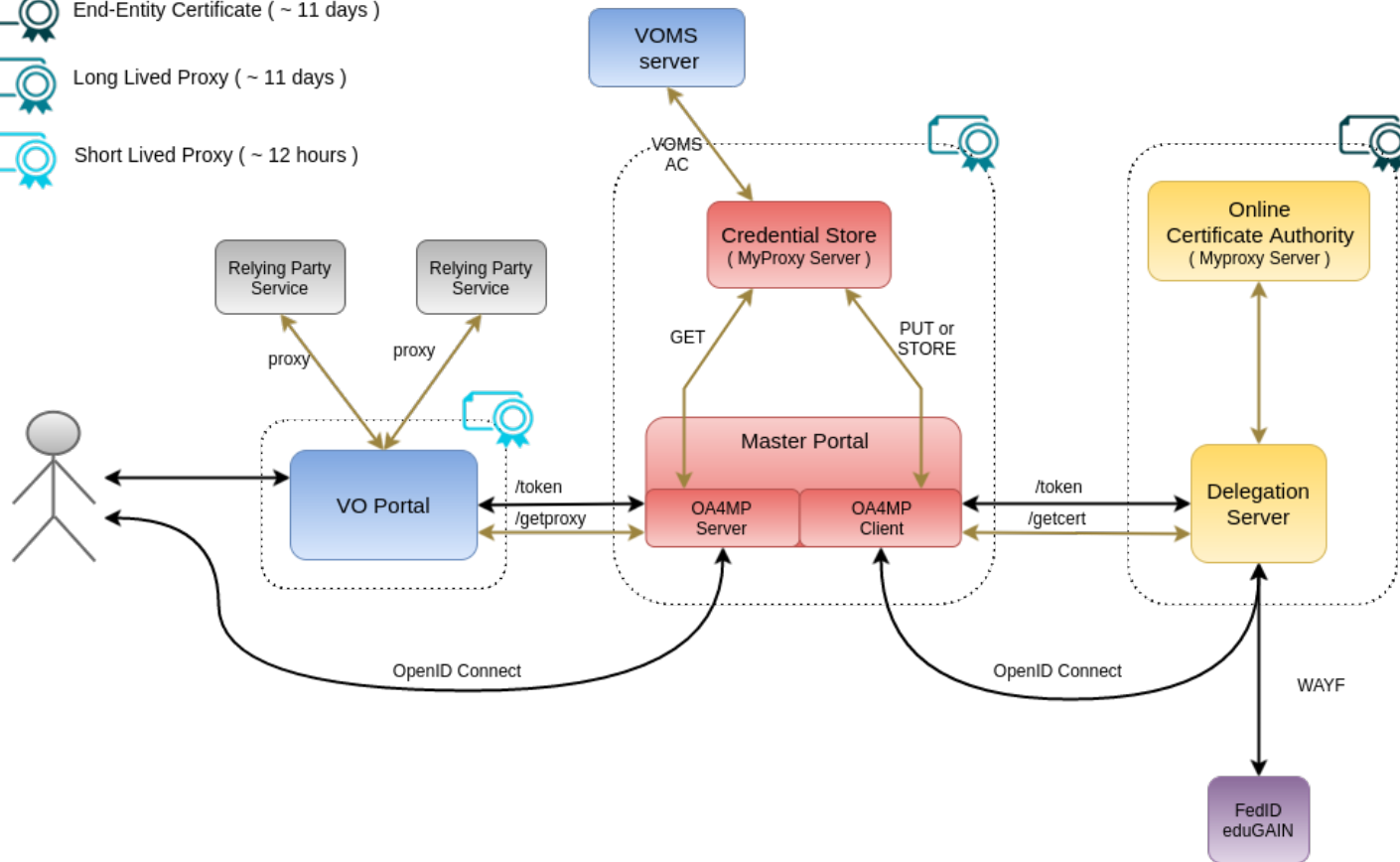
- SSH deployment
 - User logs in at WaTTS
 - User uploads public ssh-key
 - Plugin iterates through hosts
 - At each host the deployment script adds the public key to the `.authorized keys` file



WaTTS as master portal

INDIGO - DataCloud




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

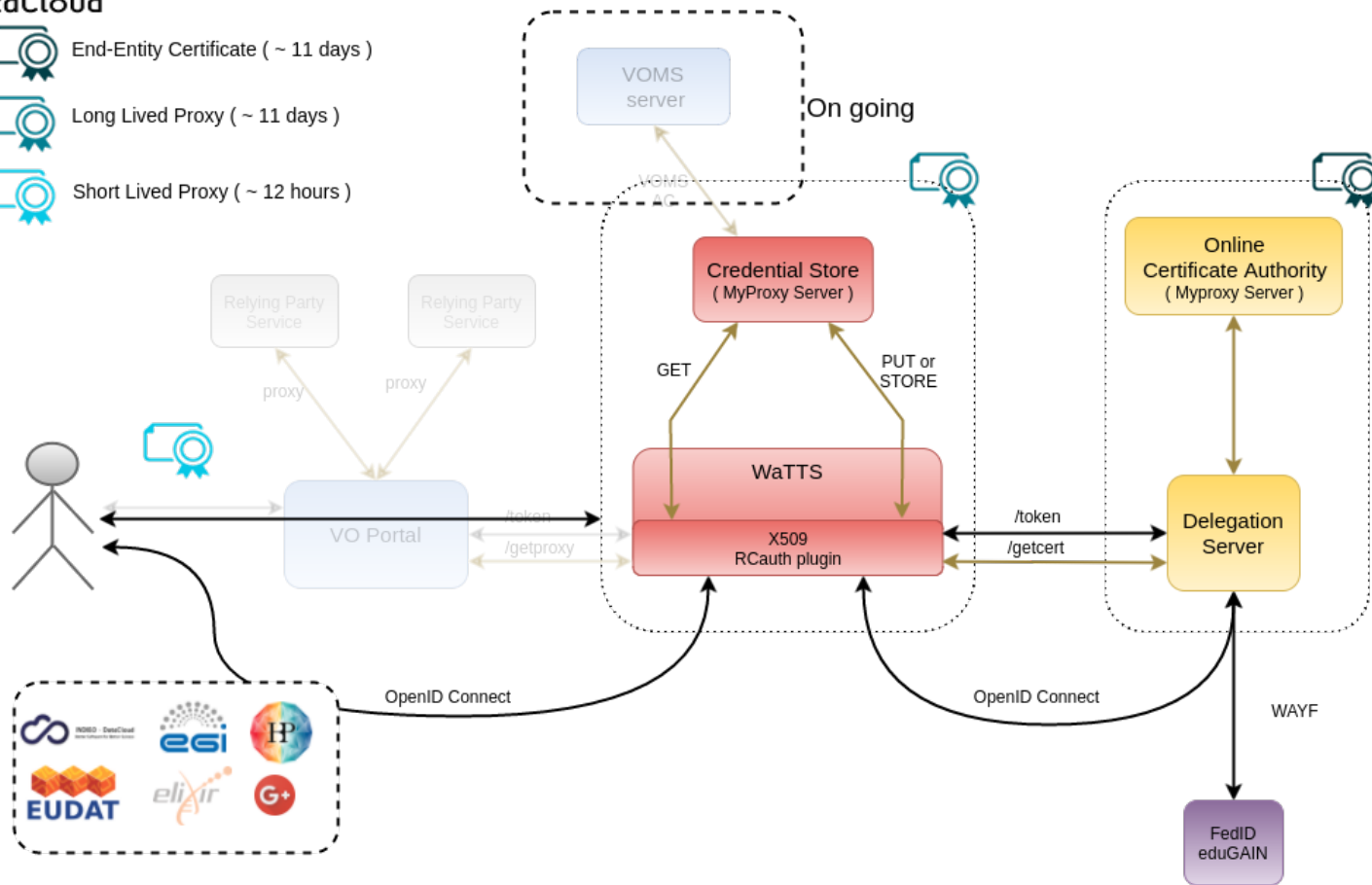




WaTTS as master portal

INDIGO - DataCloud

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





INDIGO - DataCloud

WaTTS live demo?

■ WaTTS

- <https://watts.data.kit.edu>
- Very simple ssh demo
 - Better demo at <https://watts-dev.data.kit.edu>
- Rc Auth pilot demo



Thank you

- Thank your attention.

- Questions?

- Links

- <https://github.com/indigo-dc/tts>
- https://github.com/indigo-dc/tts_plugin_info
- https://github.com/indigo-dc/tts_plugin_ssh
- https://github.com/indigo-dc/watts_plugin_tester