



worteks
make IT work, make IT free

LEMONLDAP::NG 2.0: MULTI-FACTOR AUTHENTICATION, IDENTITY FEDERATION, WEBSERVICE AND API PROTECTION

Clément OUDOT – Identity Solutions Manager
clement.oudot@worteks.com

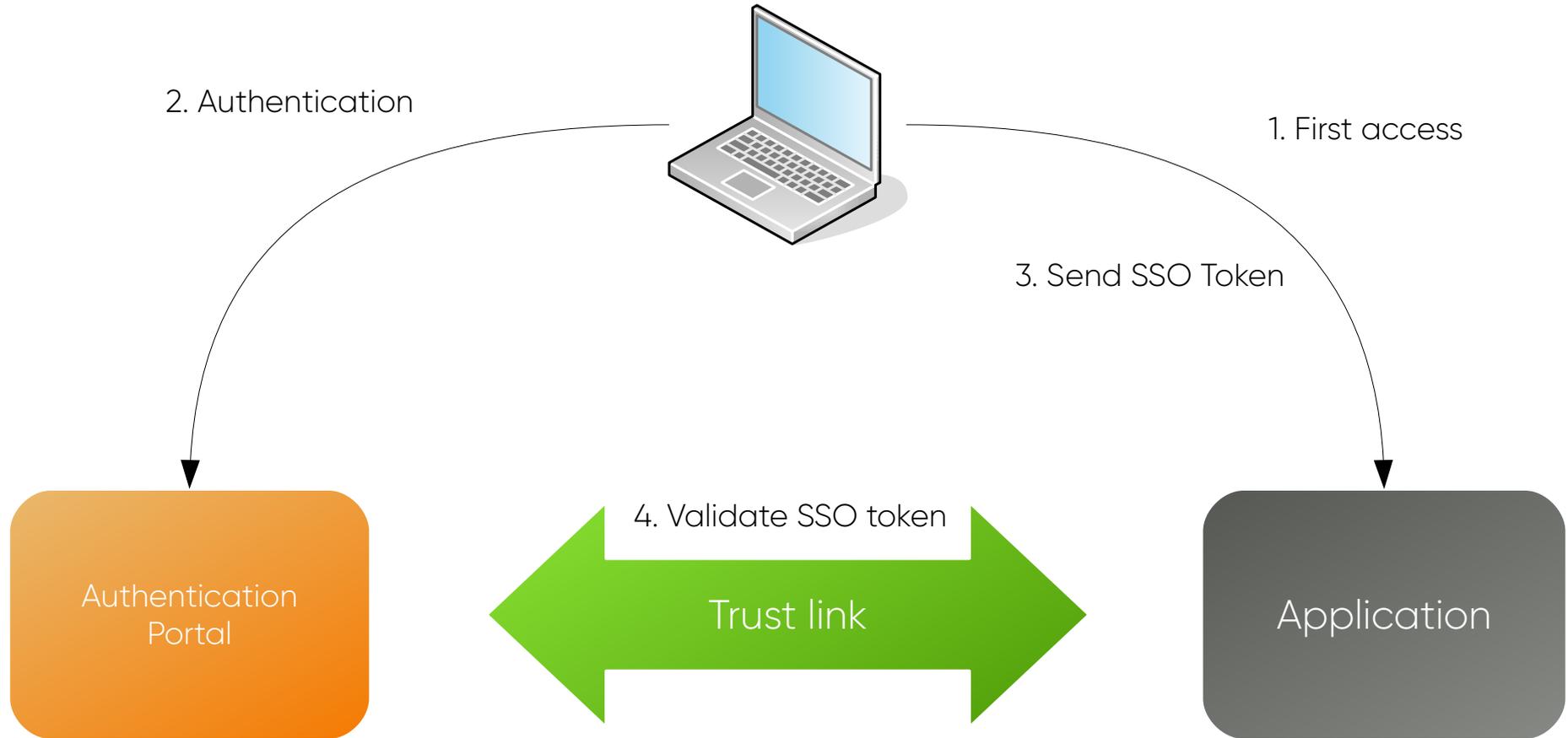


Single Sign On

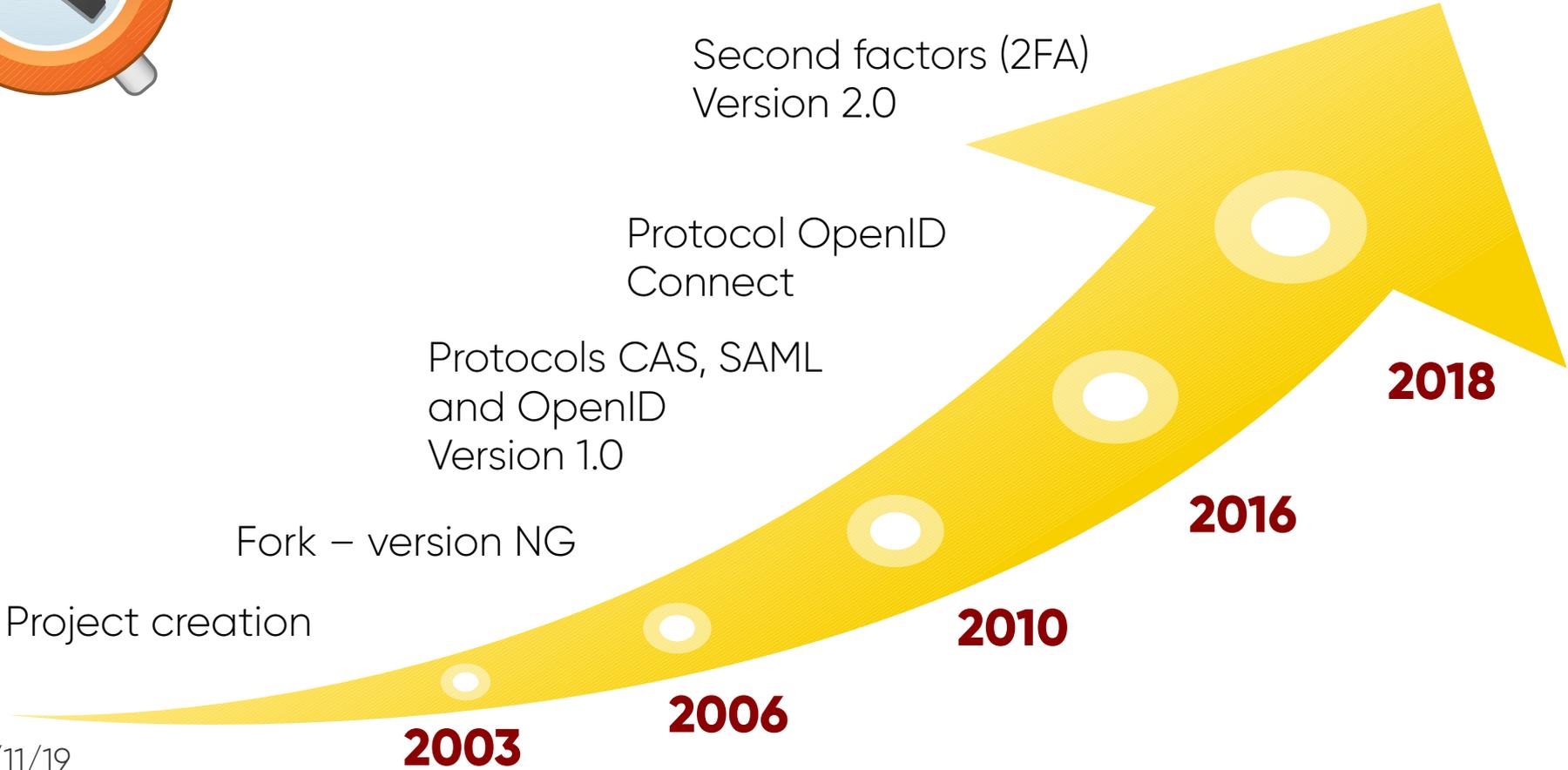
LDAPCon ❤️ Single Sign On

- LDAPCon 2007: The FederID Project
- LDAPCon 2011: The LemonLDAP::NG project
- LDAPCon 2015: The OpenID Connect Protocol
- LDAPCon 2017: Understanding main SSO protocols: CAS, SAML and OpenID Connect
- LDAPCon 2019: LemonLDAP::NG workshop and conference

SSO Workflow



LemonLDAP:NG Software



Main features



- Web Single Sign On
- Access control
- Applications portal
- Authentication modules choice and chain
- Password management, account creation
- Multi-factor authentication (MFA)
- Protection of Web applications and API/WebServices
- Graphical customisation
- Packages for Debian/Ubuntu/RHEL/CentOS

Login page



Authentication required

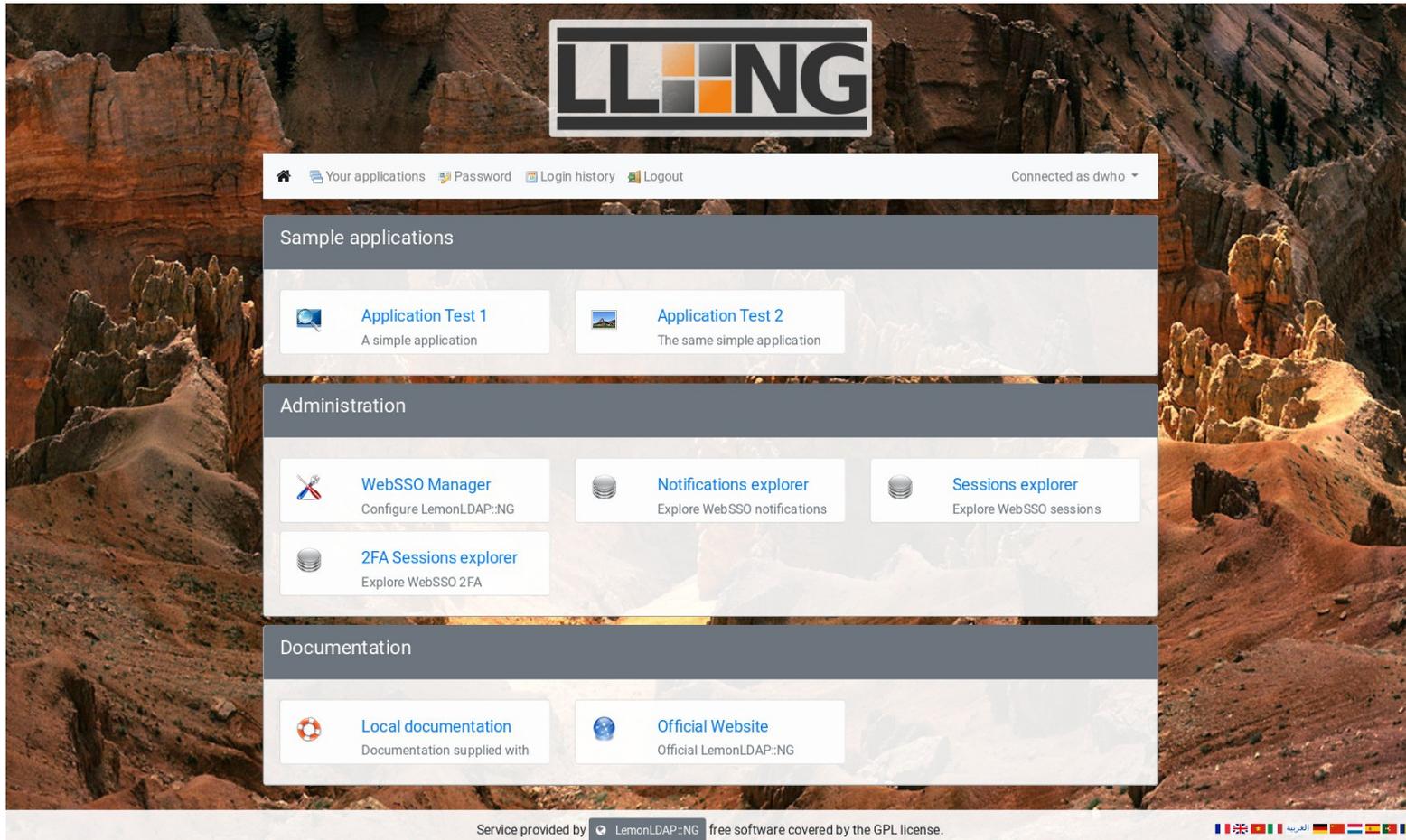
Check my last logins

[Connect](#)

[Reset my password](#)

[Create an account](#)

Portal with application menu



The screenshot displays the LLDAP:NG portal interface. At the top center is the LLDAP:NG logo. Below it is a navigation bar with links for home, 'Your applications', 'Password', 'Login history', and 'Logout', along with a user status indicator 'Connected as dwho'. The main content area is organized into four sections: 'Sample applications' with 'Application Test 1' and 'Application Test 2'; 'Administration' with 'WebSSO Manager', 'Notifications explorer', 'Sessions explorer', and '2FA Sessions explorer'; and 'Documentation' with 'Local documentation' and 'Official Website'. The footer contains the text 'Service provided by LemonLDAP:NG free software covered by the GPL license.' and a row of international flags.

LLDAP:NG

Home Your applications Password Login history Logout Connected as dwho

Sample applications

- Application Test 1**
A simple application
- Application Test 2**
The same simple application

Administration

- WebSSO Manager**
Configure LemonLDAP:NG
- Notifications explorer**
Explore WebSSO notifications
- Sessions explorer**
Explore WebSSO sessions
- 2FA Sessions explorer**
Explore WebSSO 2FA

Documentation

- Local documentation**
Documentation supplied with
- Official Website**
Official LemonLDAP:NG

Service provided by LemonLDAP:NG free software covered by the GPL license.

🇬🇧 🇫🇷 🇮🇹 🇪🇸 🇩🇪 🇳🇱 🇧🇪 🇨🇪

Web Administration interface

[Configuration](#)[Sessions](#)[Notifications](#)[Second Factors](#)[Menu](#)[> General Parameters](#)[> Variables](#)[> Virtual Hosts](#)[> SAML2 Service](#)[> SAML Identity Providers](#)[> SAML Service Providers](#)[> OpenID Connect Service](#)[> OpenID Connect Providers](#)[> OpenID Connect Relying Parties](#)[> CAS Service](#)[> CAS Servers](#)[> CAS Applications](#)[Save](#)[Browse](#)[Show help](#)[Download it](#)[Restore](#)

Current configuration

Number	1
Author	The LemonLDAP::NG team
Author IP address	127.0.0.1
Date	04/04/2015 à 11:13:28
Configuration version	2.0.0
Resume	Default configuration provided by LemonLDAP::NG team

Command Line Interface

```
root@ader-worteks:~# /usr/share/lemonldap-ng/bin/lemonldap-ng-cli info
Num      : 88
Author   : clement
Author IP: localhost
Date     : Tue Dec 18 09:57:58 2018
Log      : Edited by lmConfigEditor
root@ader-worteks:~# /usr/share/lemonldap-ng/bin/lemonldap-ng-cli help
Usage: /usr/share/lemonldap-ng/bin/lemonldap-ng-cli <options> action <parameters>

Available actions:
- help           : print this
- info          : get currentconfiguration info
- update-cache  : force configuration cache to be updated
- get <keys>    : get values of parameters
- set <key> <value> : set parameter(s) value(s)
- addKey <key> <subkey> <value> : add or set a subkey in a parameter
- delKey <key> <subkey> : delete subkey of a parameter

See Lemonldap::NG::Common::Cli(3) or Lemonldap::NG::Manager::Cli(3) for more
root@ader-worteks:~# /usr/share/lemonldap-ng/bin/lemonldap-ng-cli set ldapServer 'ldap://ldap.example.com'█
```

Free Software



- License GPL
- OW2 project
- Forge: <https://gitlab.ow2.org/lemondap-ng/lemondap-ng>
- Site: <https://lemondap-ng.org>
- OW2 Community Award in 2014 and 2018
- SSO component of FusionIAM project: <https://fusioniam.org/>



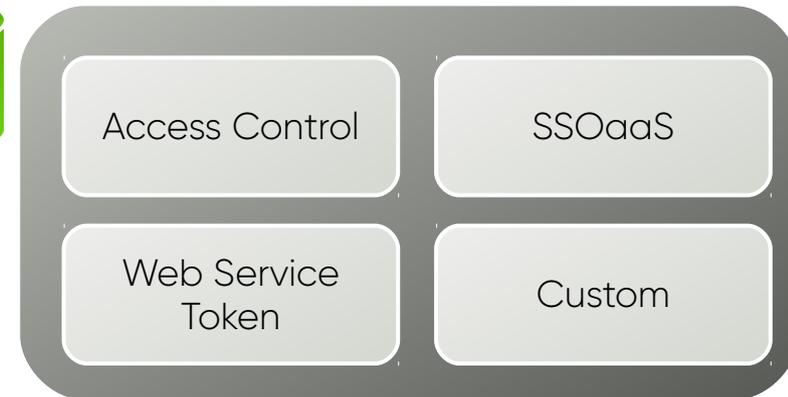
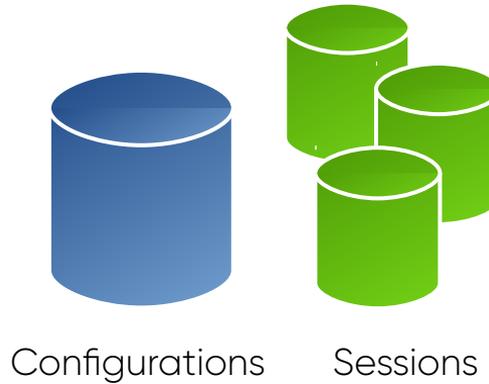
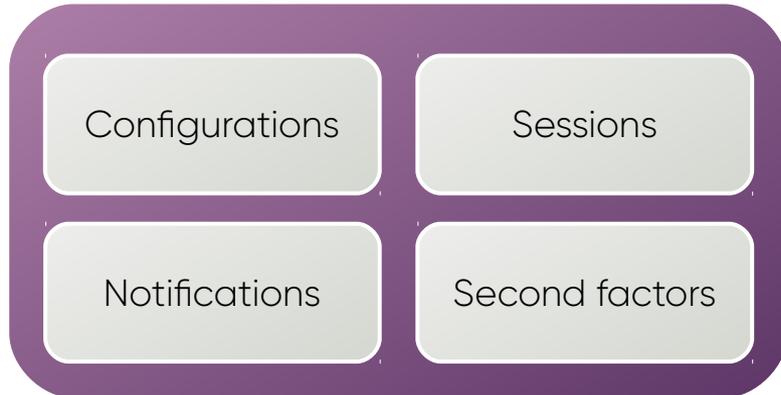
Component roles

Portal

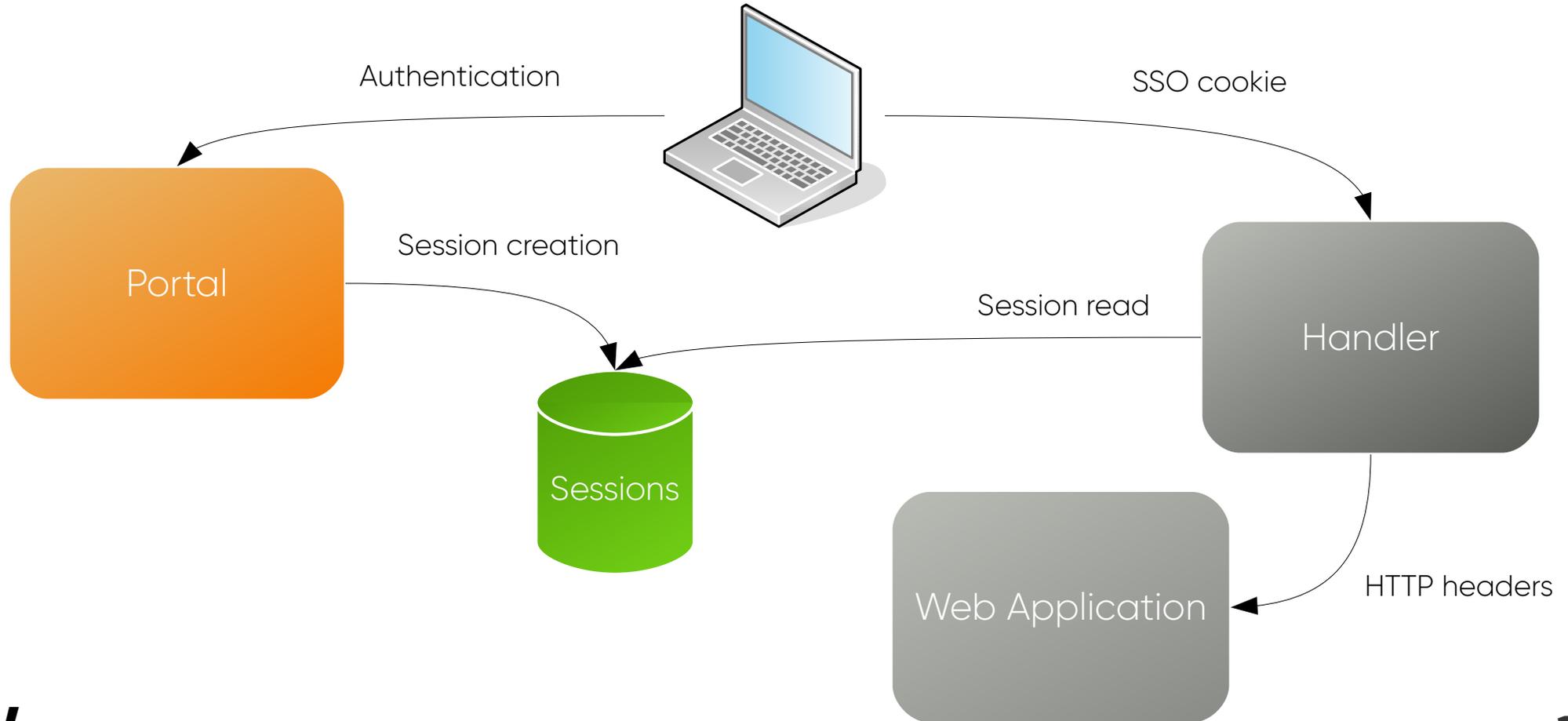


Manager

Handler



Web application protection with Handler



Multi Factor Authentication

Multi Factor Authentication

- Multi-factor authentication (MFA) is a method of confirming a user's claimed identity in which a user is granted access only after successfully presenting 2 or more pieces of evidence (or factors) to an authentication mechanism:
 - knowledge (something they and only they know)
 - possession (something they and only they have)
 - inherence (something they and only they are)

One-Time Password

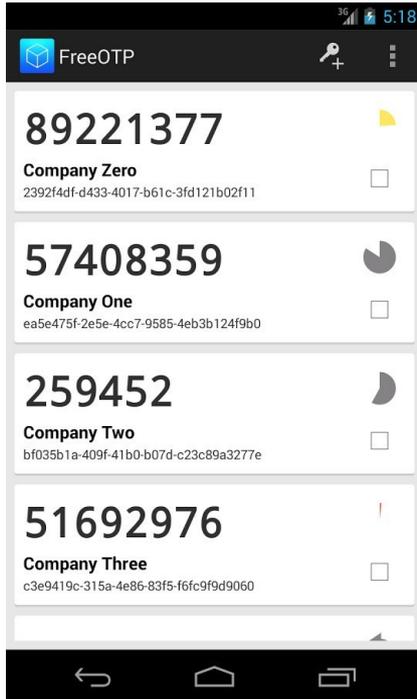
- One-Time Password (OTP) is a password that is valid for only one login session or transaction
- Two standards:
 - HOTP (RFC 4226): HMAC-Based One-Time Password
 - TOTP (RFC 6238): Time-Based One-Time Password
- Rely on a secret shared between user and server

TOTP

- Shared secret key K
- T_0 : start time
- T_I : time interval
- Time Counter $TC = \text{floor}((\text{unixtime}(\text{now}) - \text{unixtime}(T_0)) / T_I)$
- $\text{TOTP} = \text{Truncate}(\text{SHA1}(K \oplus 0x5c5c\dots \parallel \text{SHA1}(K \oplus 0x3636\dots \parallel TC)) \& 0x7FFFFFFF)$
- $\text{TOTP Value} = \text{TOTP} \bmod 10^d$, where d is the desired number of digits of the one-time password



Using a TOTP



- Registration on client: shared key can be registered manually or using a QR code
- Server associates shared secret to user
- At next authentication, TOTP value is computed by client and server



Universal Second Factor

- Universal 2nd Factor (U2F) is an open authentication standard that strengthens and simplifies two-factor authentication using specialized USB or NFC devices.
- Managed by FIDO Alliance <https://fidoalliance.org/>



Using U2F

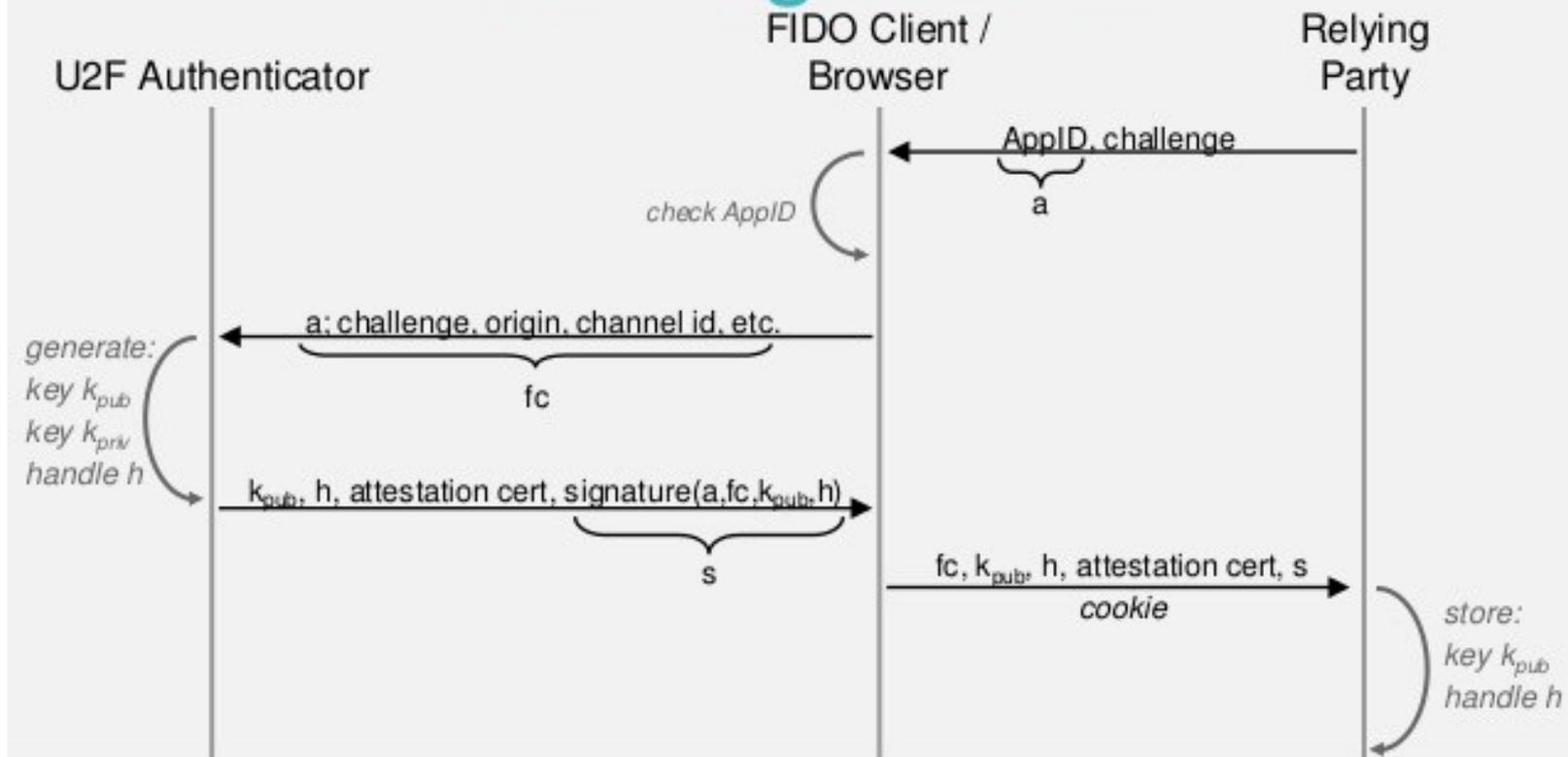


fido[™]
ALLIANCE

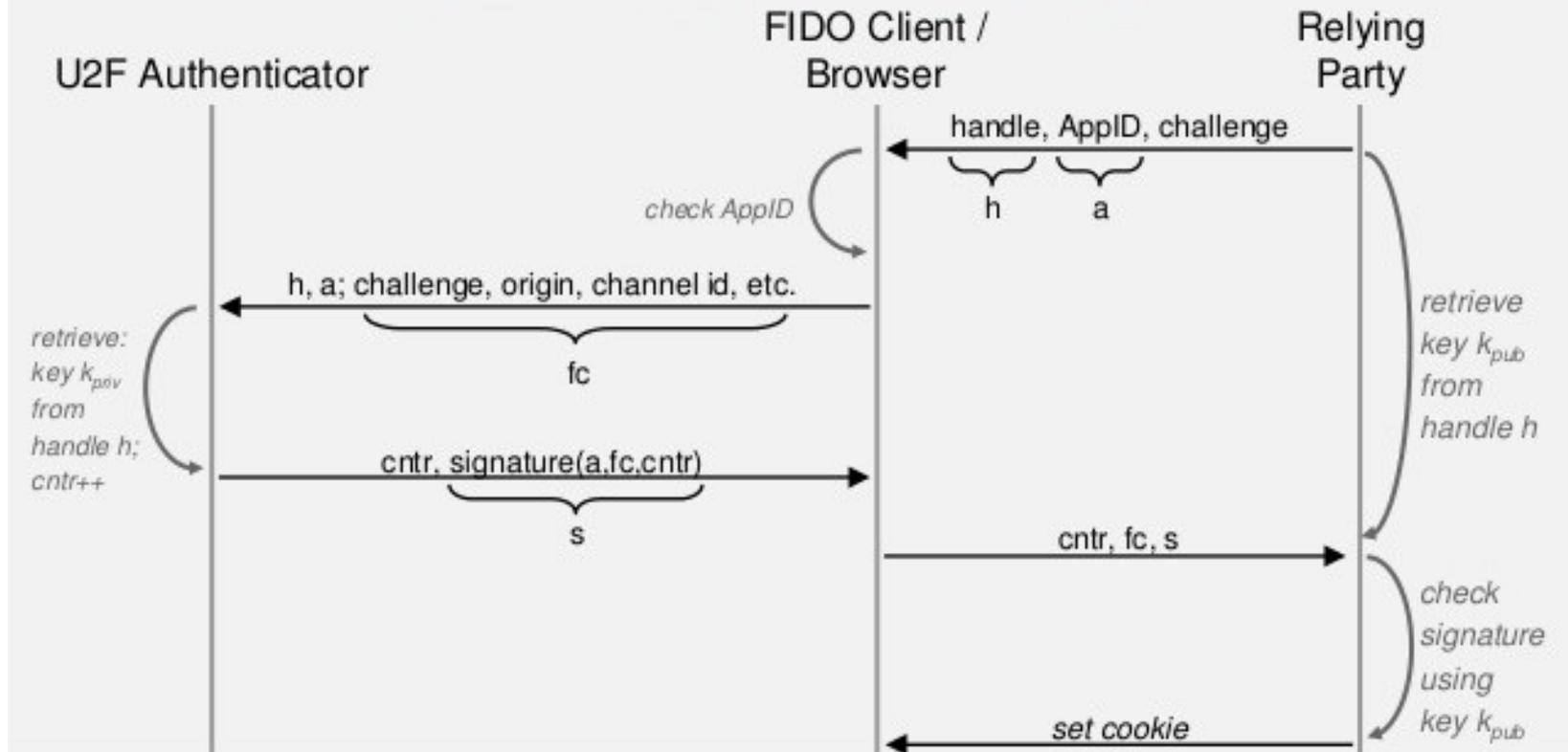
- Registration: Token generates private/public keys and a handle and send public key and handle to server
- The server associates the public key and the handle to user
- At next authentication, server sends the handle and a crypto challenge and the U2F token signs the challenge and sends it back



U2F Registration

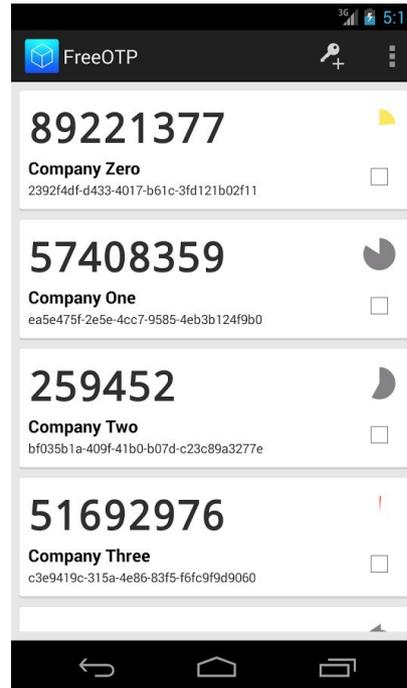


U2F Authentication



Support in LL::NG

- LemonLDAP::NG can use the following 2FA:
 - TOTP
 - U2F
 - TOTP or U2F
 - Mail
 - External
 - REST
 - Yubikey



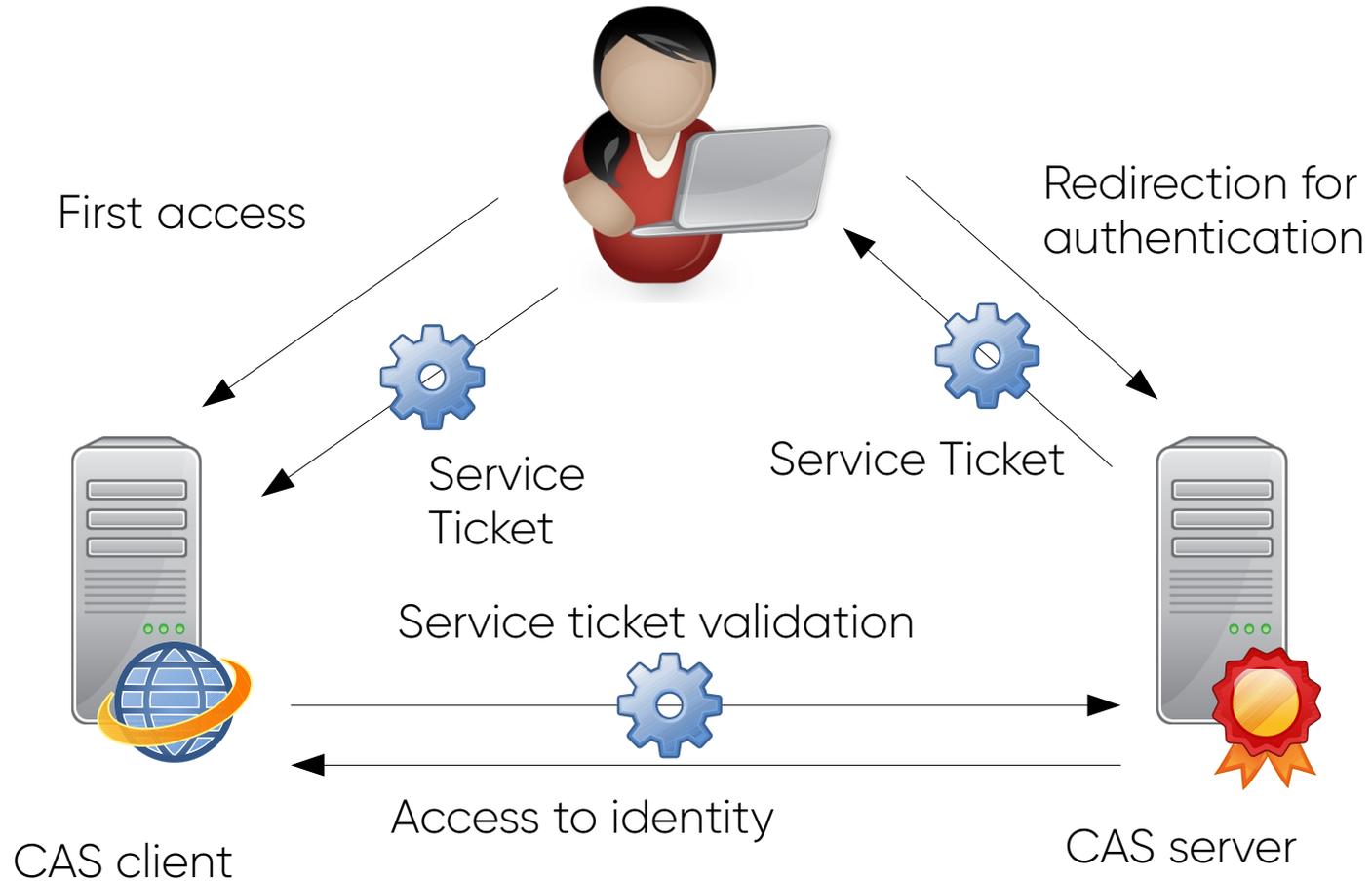
fido
ALLIANCE

Identity federation

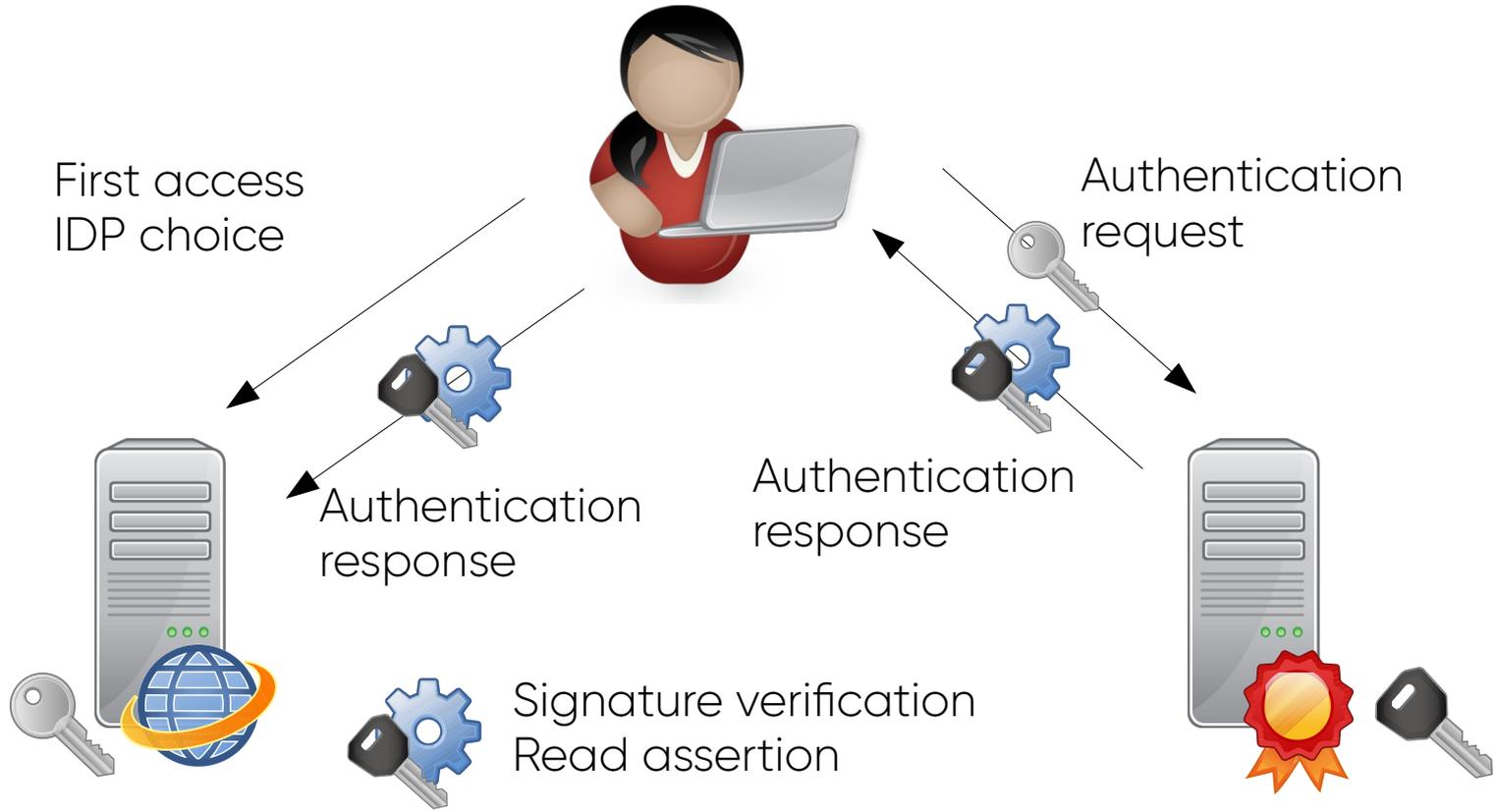
Main features

- LL::NG can act as client and as server
- Attributes sharing
- Manage authentication contexts and levels
- Autogeneration of public/private keys
- Access control per services
- Publication of configuration data (metadata)
- Multi-protocols gateway
- Single logout

CAS



SAML



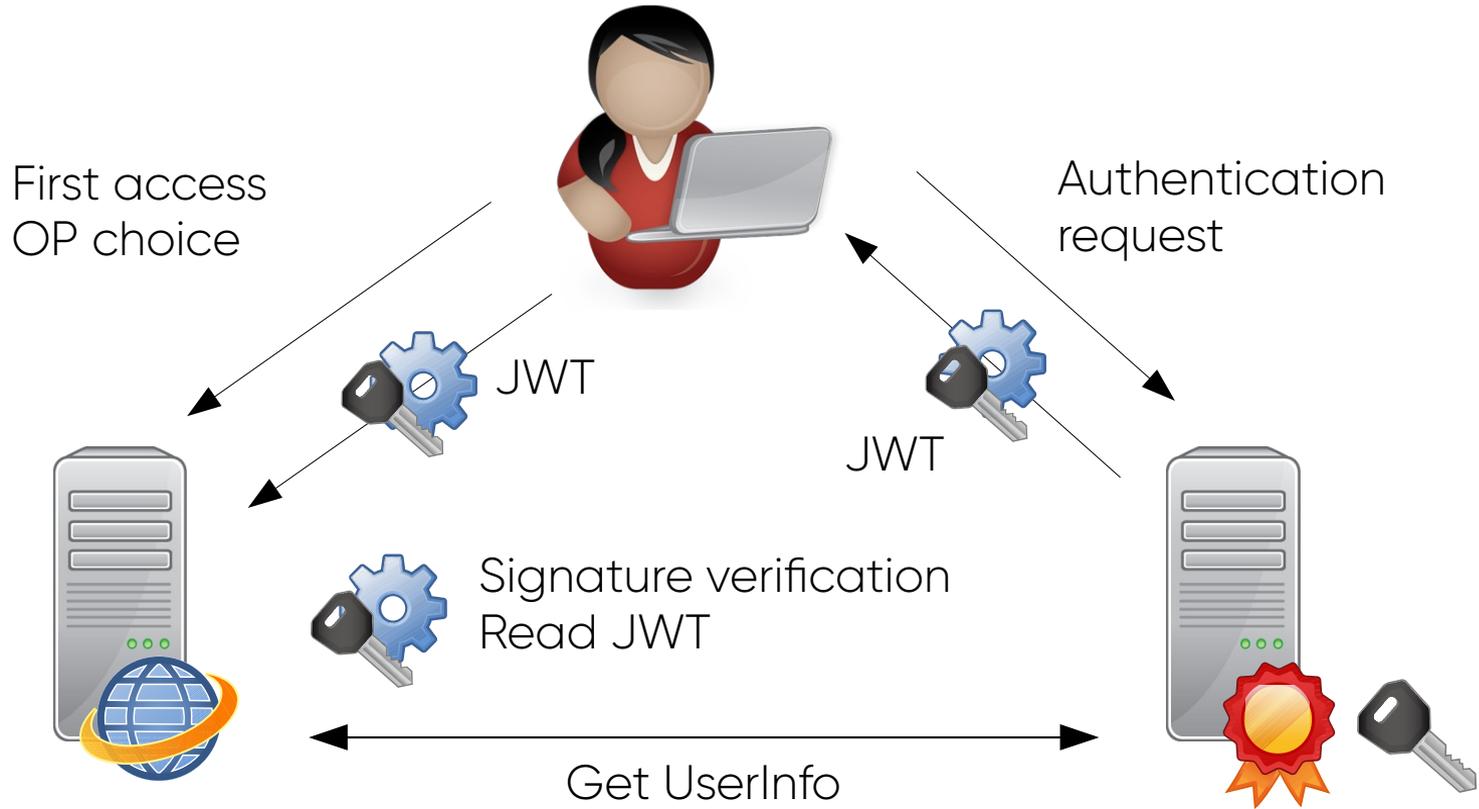
Service Provider (SP)

Identity Provider (IDP)



05/11/19

OpenID Connect



Relying Party (RP)

OpenID Provider (OP)



API / Webservice protection

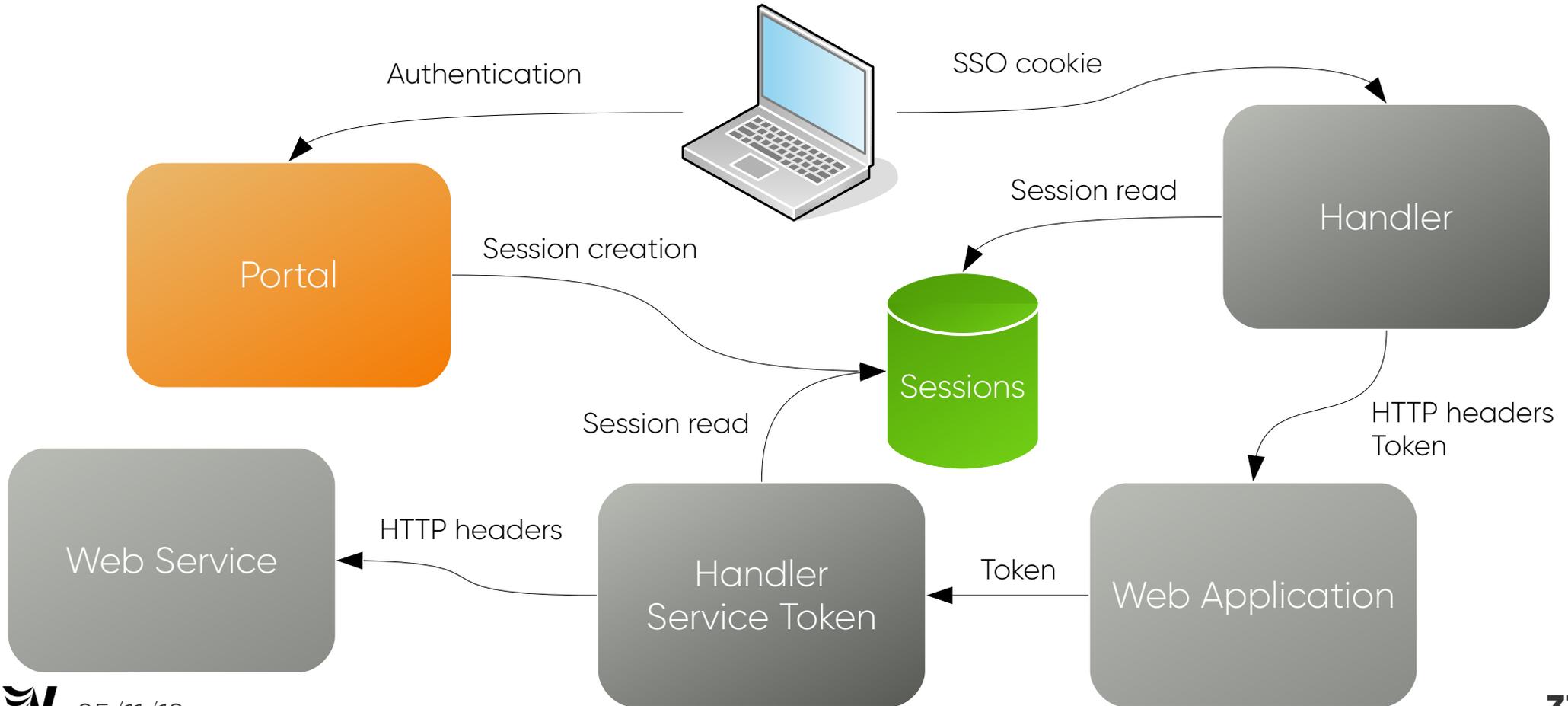
How to protect a Webservice

- Global authentication:
 - HTTP Basic
 - SSL client certificate
- User oriented authentication?

LL::NG ServiceToken Handler

- New Handler "Service Token" installed between application and WebService
- Main Handler generates a token based on time session_id and virtual hosts: `cipher(time, session_id, vhost_list)`
- The token is sent by application to WebService
- The Handler "Service Token" intercepts the token, validates it and apply access rules, and sent HTTP headers to WebService

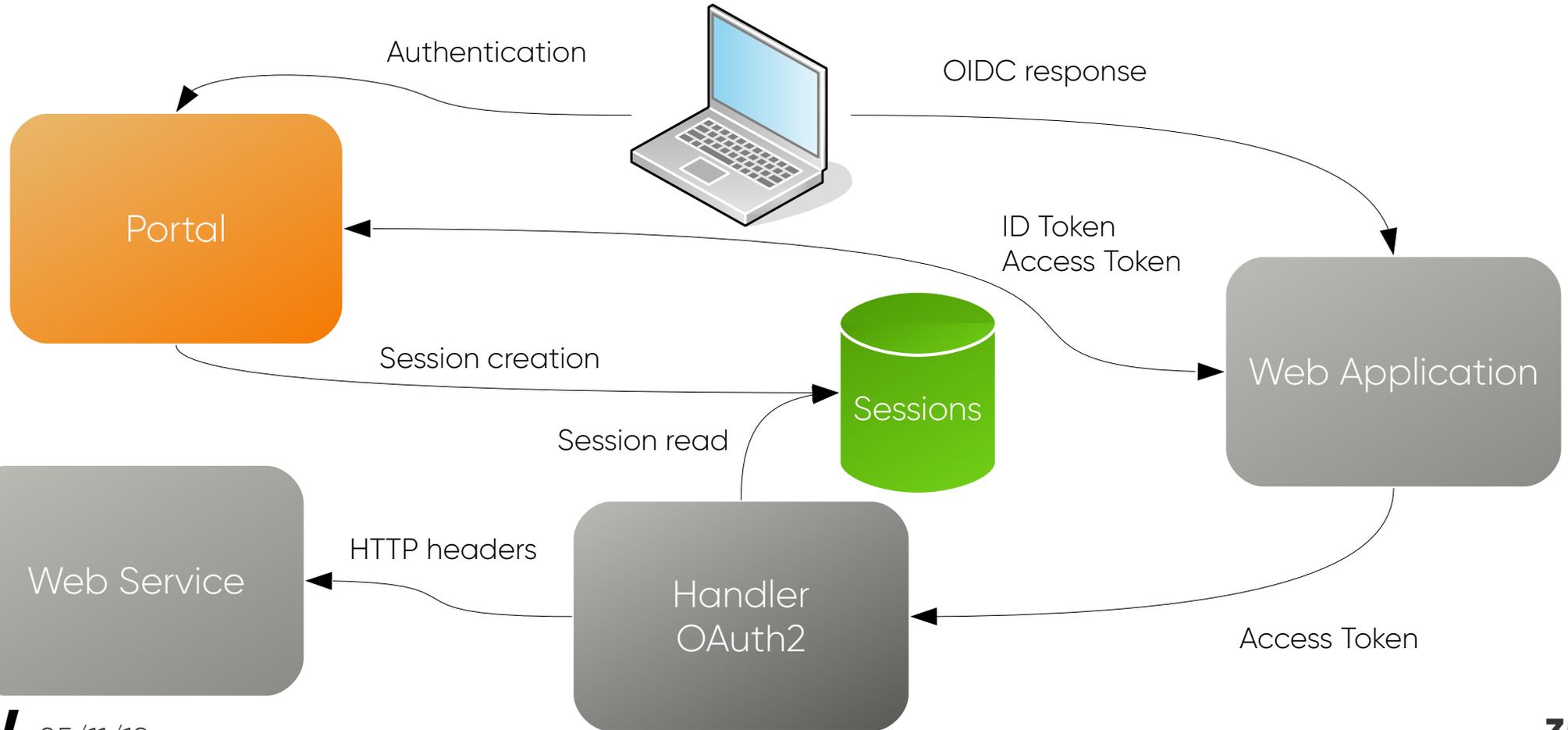
LL::NG ServiceToken Handler



Using OAuth2

- When LL::NG acts as OIDC provider, it delivers an OAuth2 access token
- This access token can be validated with different operations:
 - Call `/oauth2/userinfo`, which will return user attributes
 - Call `/oauth2/introspect`, which will return token information (including the token owner) – see RFC 7662
 - Use LL::NG OAuth2 Handler

LL:NG OAuth2 Handler



Example – UserInfo Endpoint

```
$ curl -k \  
-H "Authorization: Bearer a74d504ec9e784785e70a1da2b95d1d2" \  
https://auth.openid.club/oauth2/userinfo | json_pp  
  
{  
  "family_name" : "OUDOT",  
  "name" : "Clément OUDOT",  
  "email" : "clement@oodo.net",  
  "sub" : "coudot"  
}
```

Example – Intropsection Endpoint

```
$ curl -k \  
-H "Authorization: Basic bGVtb25sZGFwOnNlY3JldA==" \  
-X POST -d "token=a74d504ec9e784785e70a1da2b95d1d2" \  
https://auth.openid.club/oauth2/introspect | json_pp  
  
{  
  "client_id" : "lemonldap",  
  "sub" : "coudot",  
  "exp" : 1572446485,  
  "active" : true,  
  "scope" : "openid profile address email phone"  
}
```

Example – OAuth2 Handler

```
$ curl -k \  
-H "Authorization: Bearer a74d504ec9e784785e70a1da2b95d1d2" \  
https://oauth2.openid.club/api.pl  
  
{  
  "check" : "true",  
  "user" : "coudot"  
}
```



**THANKS FOR YOUR
ATTENTION**

More informations:

 info@worteks.com

 [@worteks_com](https://twitter.com/worteks_com)

 [linkedin.com/company/worteks](https://www.linkedin.com/company/worteks)

