



The wonderful story of Web Authentication and Single Sign On



Clément OUDOT
@clementoudot
<http://sflx.ca/coudot>



- Founded in 1999
- >100 persons
- Montréal, Quebec City, Toronto, Paris
- ISO 9001:2004 / ISO 14001:2008
- contact@savoirfairelinux.com



What is Single Sing On?

Single Sign On Singing a Song

Imagine SSOng

Imagine there are no passwords

Or maybe just only one

A single secured form

To access our applications

Imagine all the users

Loving security

You may say

I'm a hacker

But I'm not the only one

I hope one day

You will log in

Using the Single Sign On

Imagine applications

No more storing passwords

Relying on a token

Even for authorizations

Imagine all developers

Loving security

Imagine some protocols

Made by clever people

CAS, OpenID or SAML

Even WS Federation

Imagine authentication

Interoperability



© John Lennon

Authentication on the web

The HTTP protocol

- 1989: first versions of HTTP, not documented
- 1991: HTTP/0.9
- 1996: HTTP/1.0
- 1997: HTTP/1.1 (updated in 1999)
- 2015: HTTP/2

Basic authentication



GET /index.html HTTP/1.1
Host: www.example.com

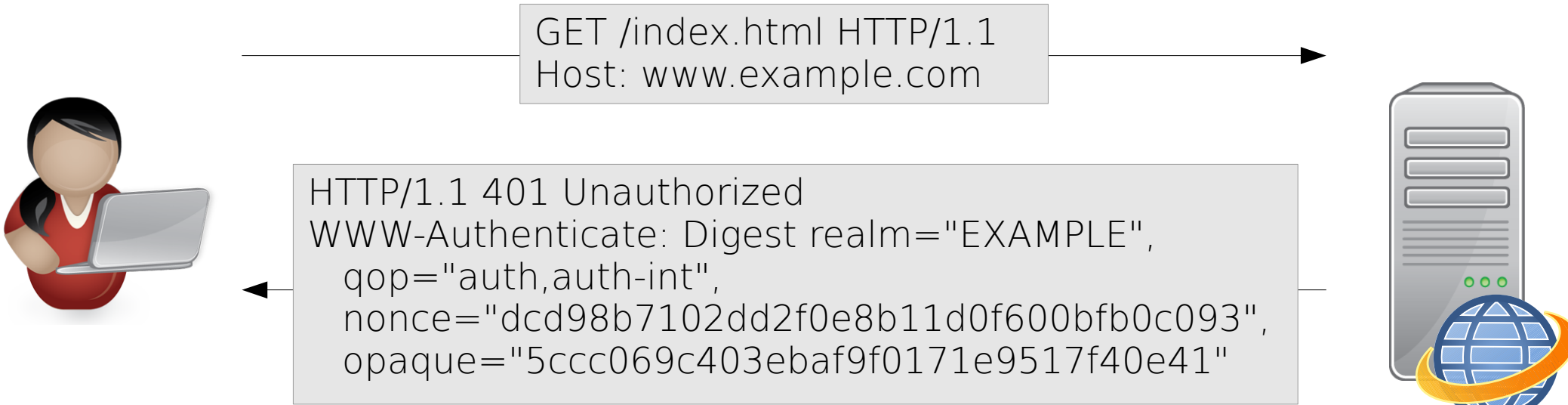
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Basic realm="EXAMPLE"

GET /index.html HTTP/1.1
Host: www.example.com
Authorization: Basic Y291ZG90OnNIY3JldA==

Base64(coudot:secret)



Digest authentication



```
GET /index.html HTTP/1.1  
Host: www.example.com
```

```
HTTP/1.1 401 Unauthorized  
WWW-Authenticate: Digest realm="EXAMPLE",  
qop="auth,auth-int",  
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",  
opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

Digest authentication

```
GET /index.html HTTP/1.1
Host: www.example.com
Authorization: Digest username="coudot",
  realm="EXAMPLE",
  nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
  uri="/index.html",
  qop=auth,
  nc=00000001,
  cnonce="0a4f113b",
  response="d2156c87fd5a1d75d23106edeecb232d",
  opaque="5ccc069c403ebaf9f0171e9517f40e41"
```



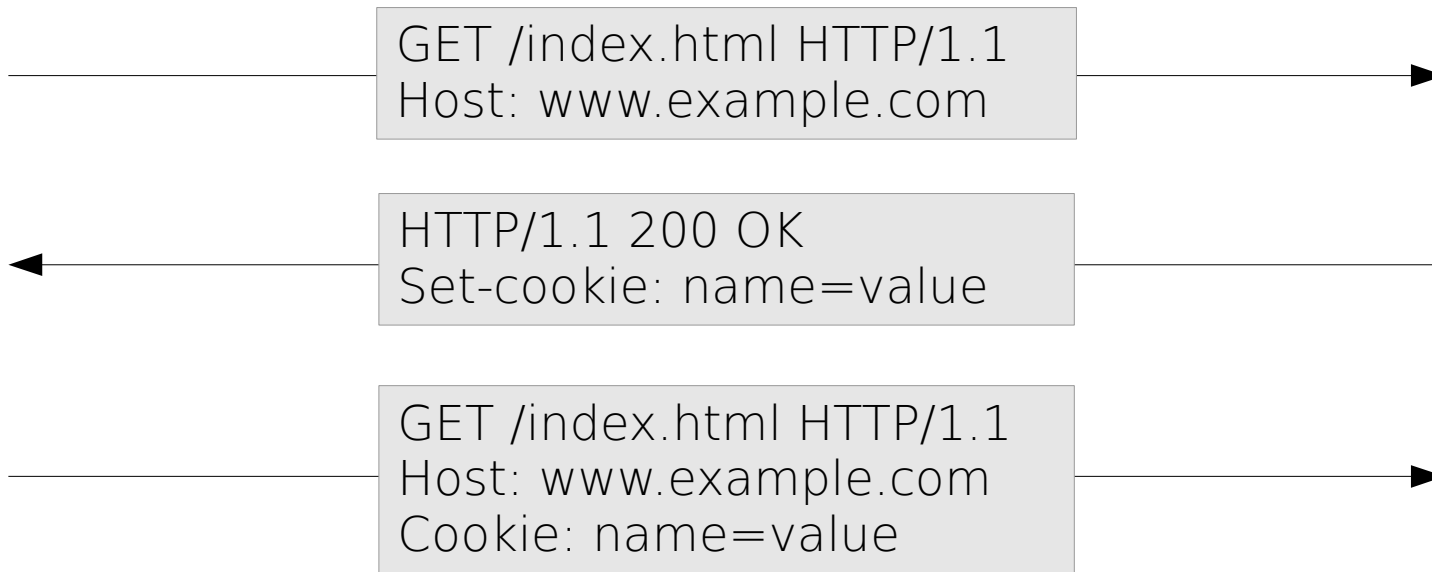
Digest authentication

```
HA1 = MD5( "coudot:EXAMPLE:secret" )  
    = 8d1a89700d53c9383402f0ac3b24eff9
```

```
HA2 = MD5( "GET:/index.html" )  
    = 5f751b15eae8c79635edae8bf3b92354
```

```
Response = MD5( "8d1a89700d53c9383402f0ac3b24eff9:\  
                dcd98b7102dd2f0e8b11d0f600bfb0c093:\  
                00000001:0a4f113b:auth:\  
                5f751b15eae8c79635edae8bf3b92354" )  
    = d2156c87fd5a1d75d23106edeecb232d
```

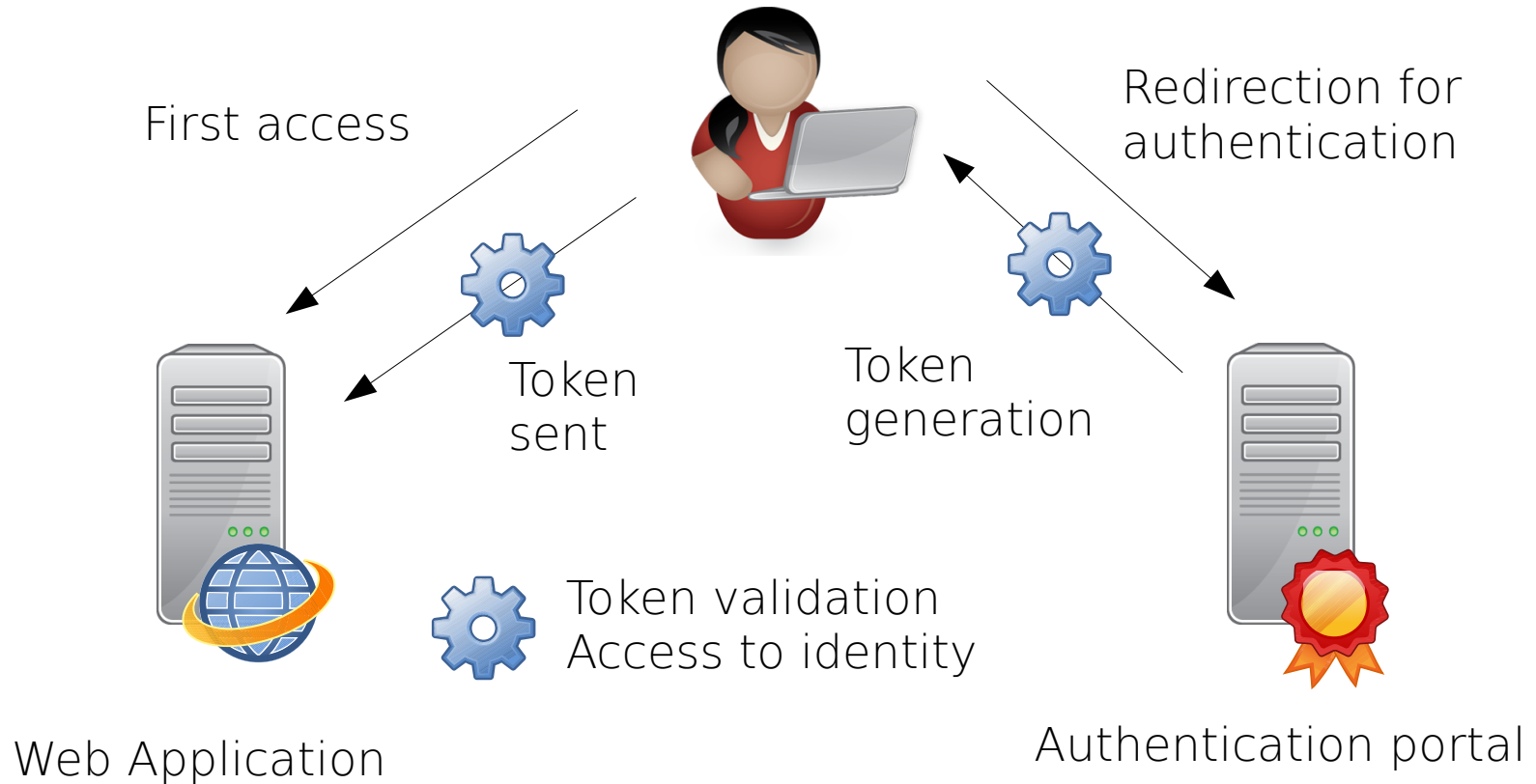
Cookies



Cookies

- Maximal size: 4 ko
- Expiration :
 - Session cookie: when browser is closed
 - Persistent cookie: at a specific date
- Path and domain: scope of the cookie
- Flags:
 - Secure: only sent over HTTPS
 - HttpOnly: not readable by Javascript

Authentication service



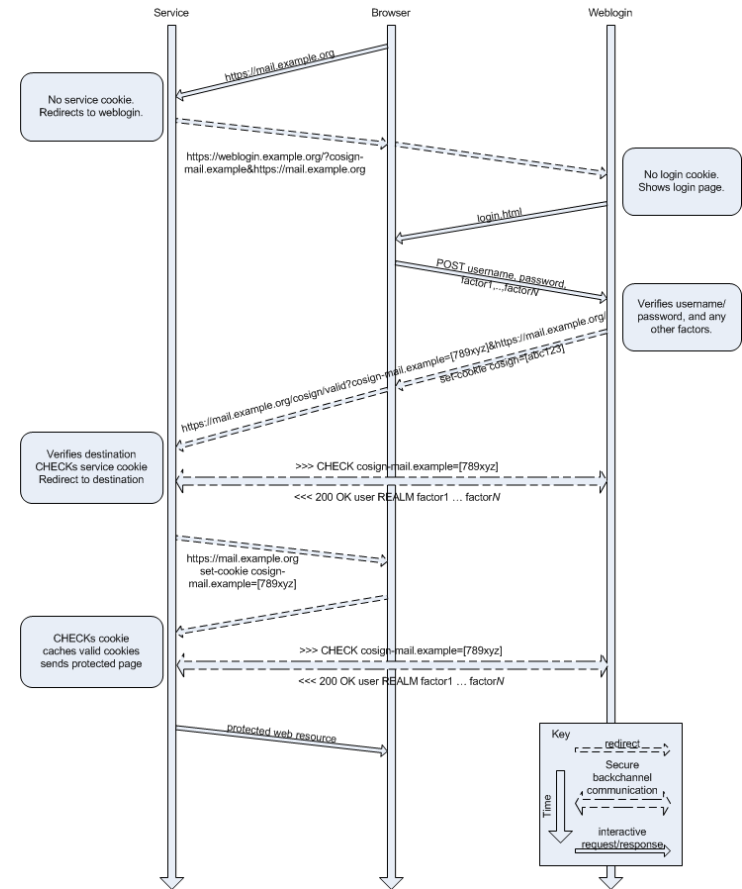
Single Sign On Protocols

CoSign

- Created by University of Michigan
- Relies on cookies with back-link verifications
- <http://weblogin.org/>

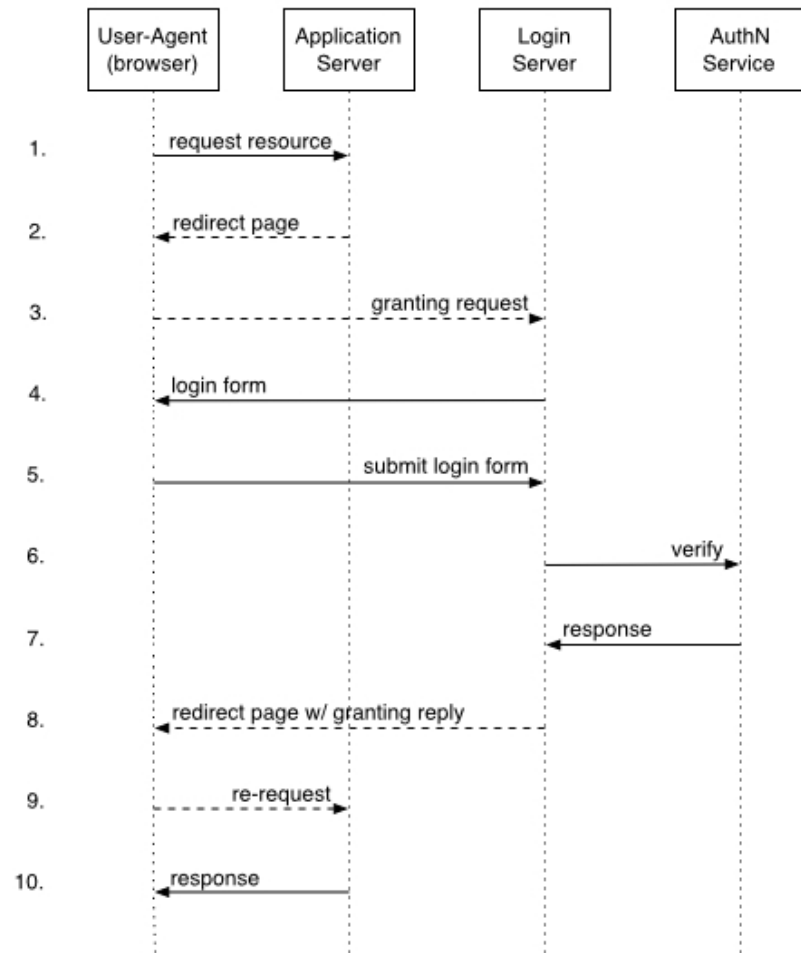


Cosign v3 Schema



Pubcookie

- Created by University of Washington
- Relies on cookies (granting cookie and login cookie) with signatures
- <http://www.pubcookie.org>

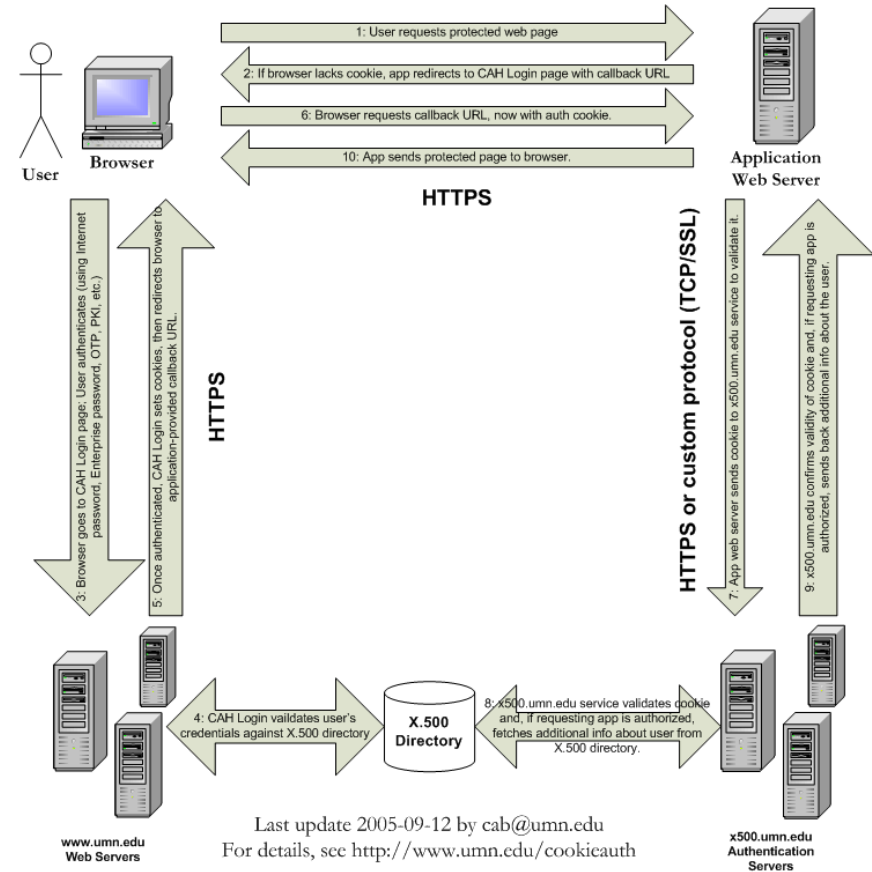


Webauth

- Created by University of Stanford
- Relies on cookies/token
- Authentication portal is called WebKDC
- XML messages between applications and WebKDC
- <http://webauth.stanford.edu/>

CAH

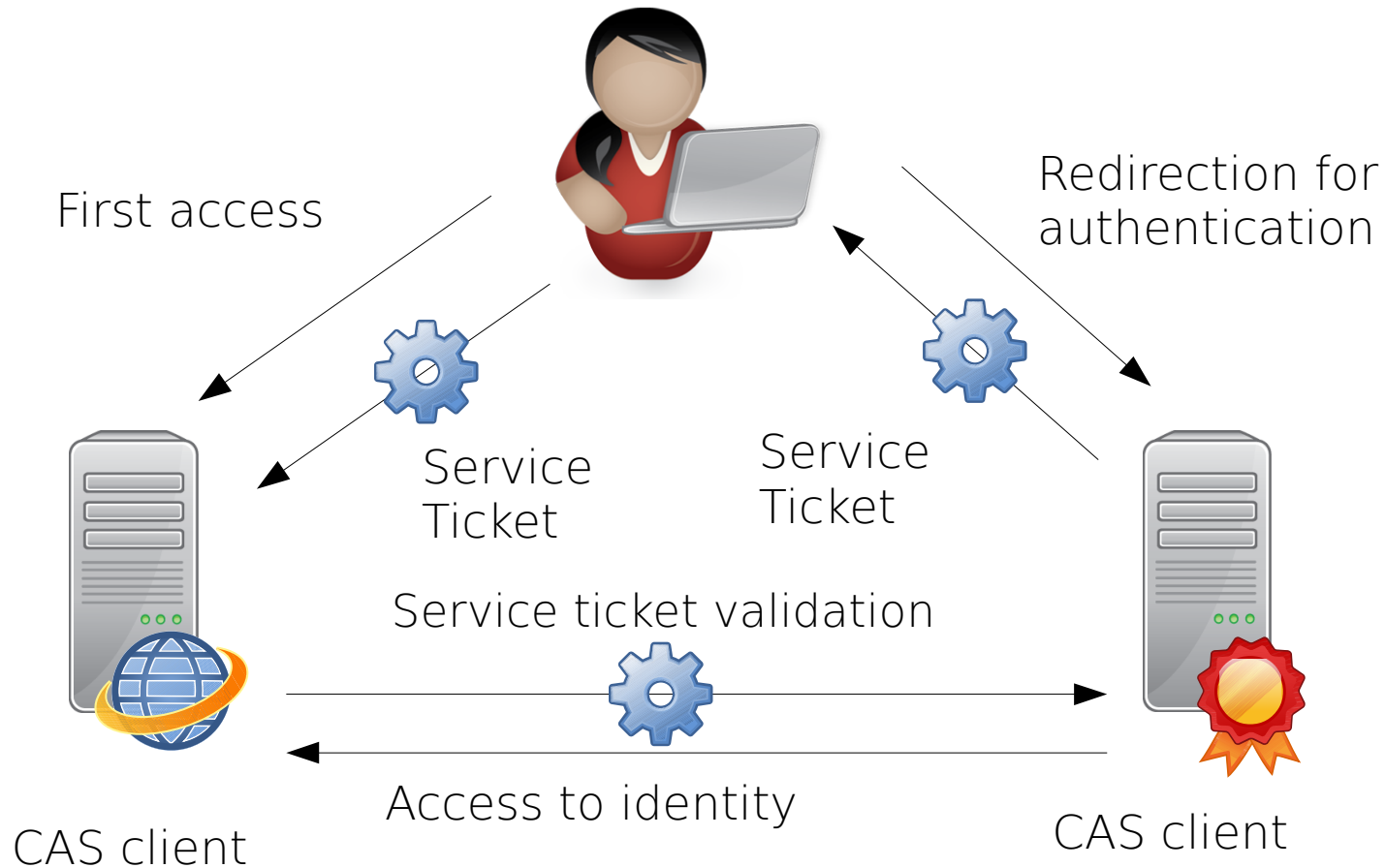
- Created by University of Minnesota in 1998
- Central Authentication Hub
- <https://it.umn.edu/about-central-authentication-hub>



CAS

- Created by University of Yale
- Central Authentication Service
- Proxy mode since v2.0
- Attributes sharing since v3.0
- <https://www.apereo.org/projects/cas>





First step

- From client to server, through browser:

```
https://auth.example.com/cas/login?  
service=http://appli.example.com/
```

- From server to client, through browser:

```
http://appli.example.com/?ticket=ST-  
6096f5d3ddb33df6fd79529e2d626a6d
```

Second step

- From client to server, direct link:

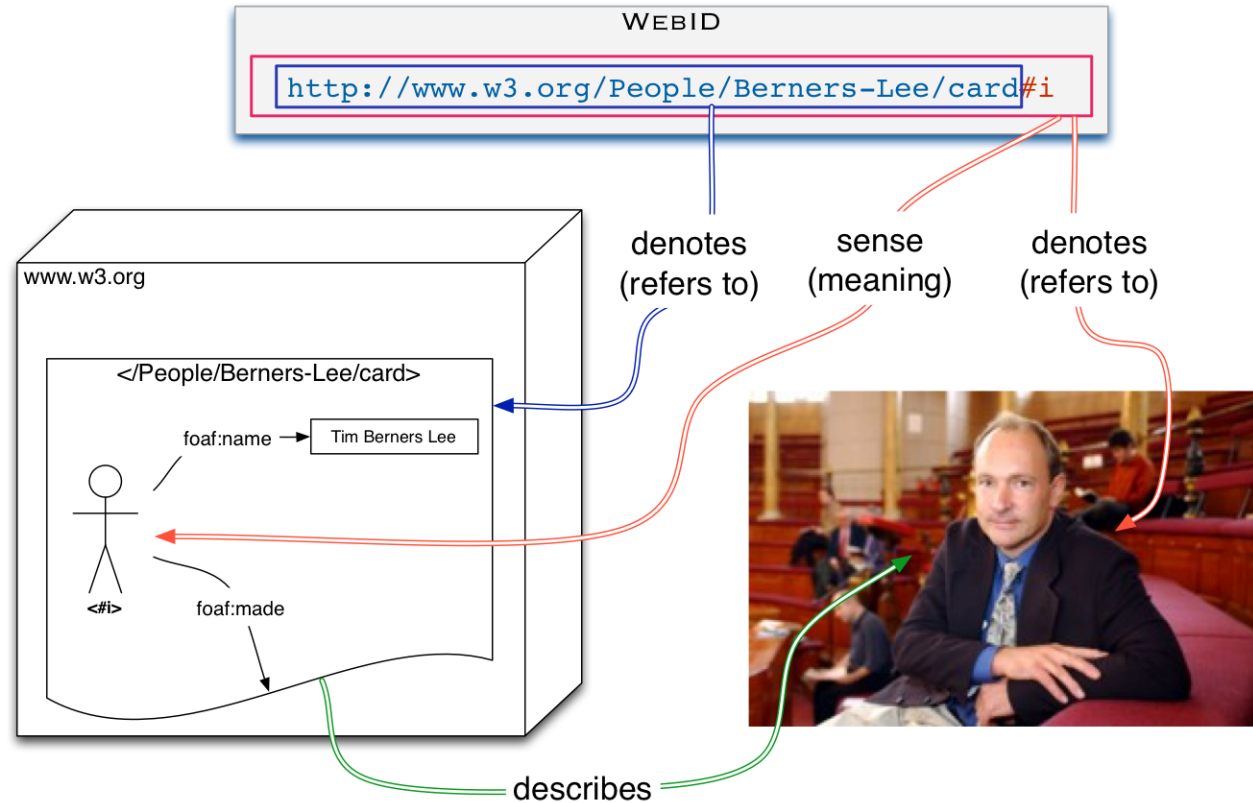
```
https://auth.example.com/cas/serviceValidate?  
service=http://appli.example.com/&ticket=ST-  
6096f5d3ddb33df6fd79529e2d626a6d
```

- From server to client, direct link:

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>  
  <cas:authenticationSuccess>  
    <cas:user>coudot</cas:user>  
  </cas:authenticationSuccess>  
</cas:serviceResponse>
```

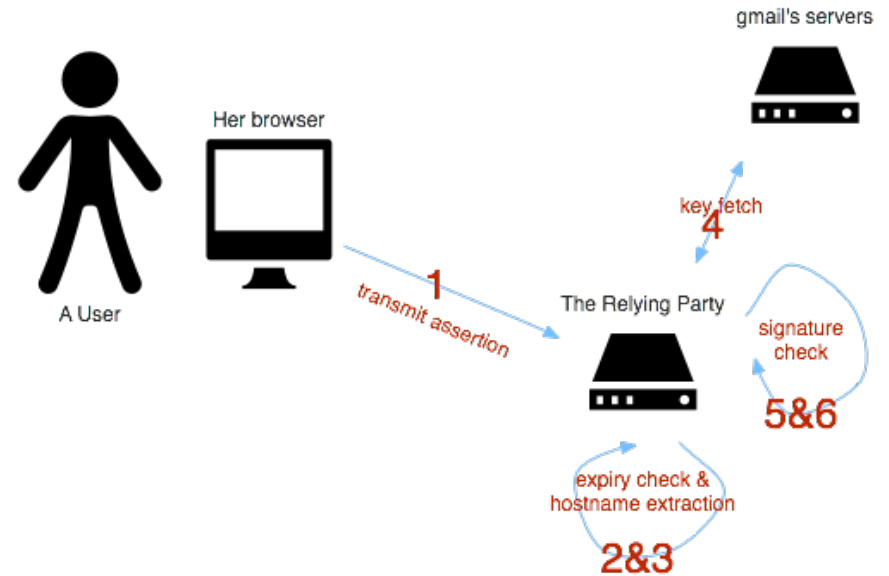
WebID

- FOAF + SSL
- Made by W3C
- Uses client certificates
- <https://www.w3.org/wiki/WebID>



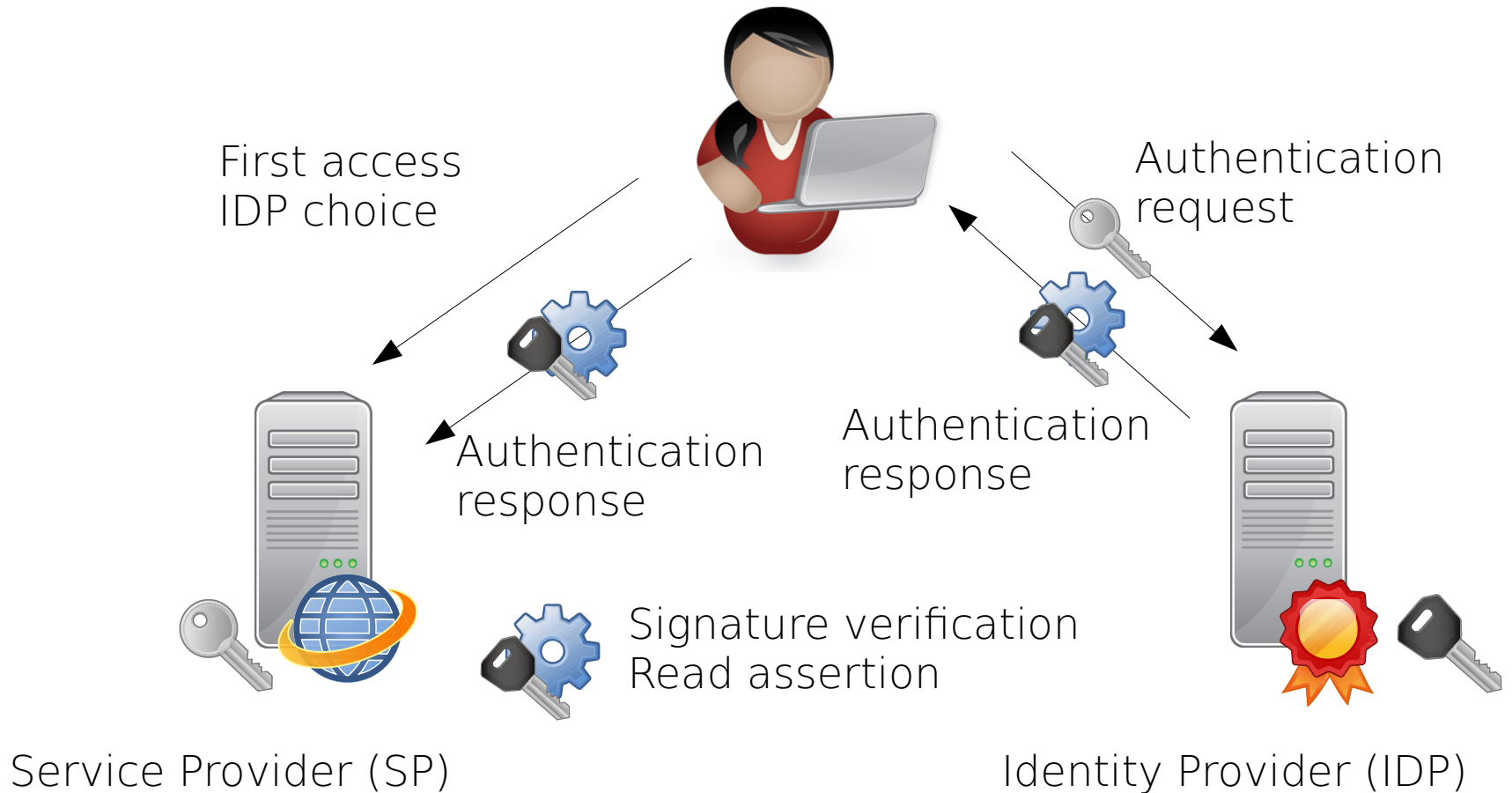
BrowserID

- Created by Mozilla in 2011
- Presented at RMLL in 2013
- Abandoned in 2014
- Uses email as principal identity
- <https://developer.mozilla.org/en-US/Persona>



SAML

- Created by OASIS organization
- Security Assertion Markup Language
- Version 1.0 in 2002
- Version 1.1 in 2003
- Version 2.0 in 2005 merging SAML, Shibboleth and ID-FF (Liberty Alliance)



SAML Authn Request

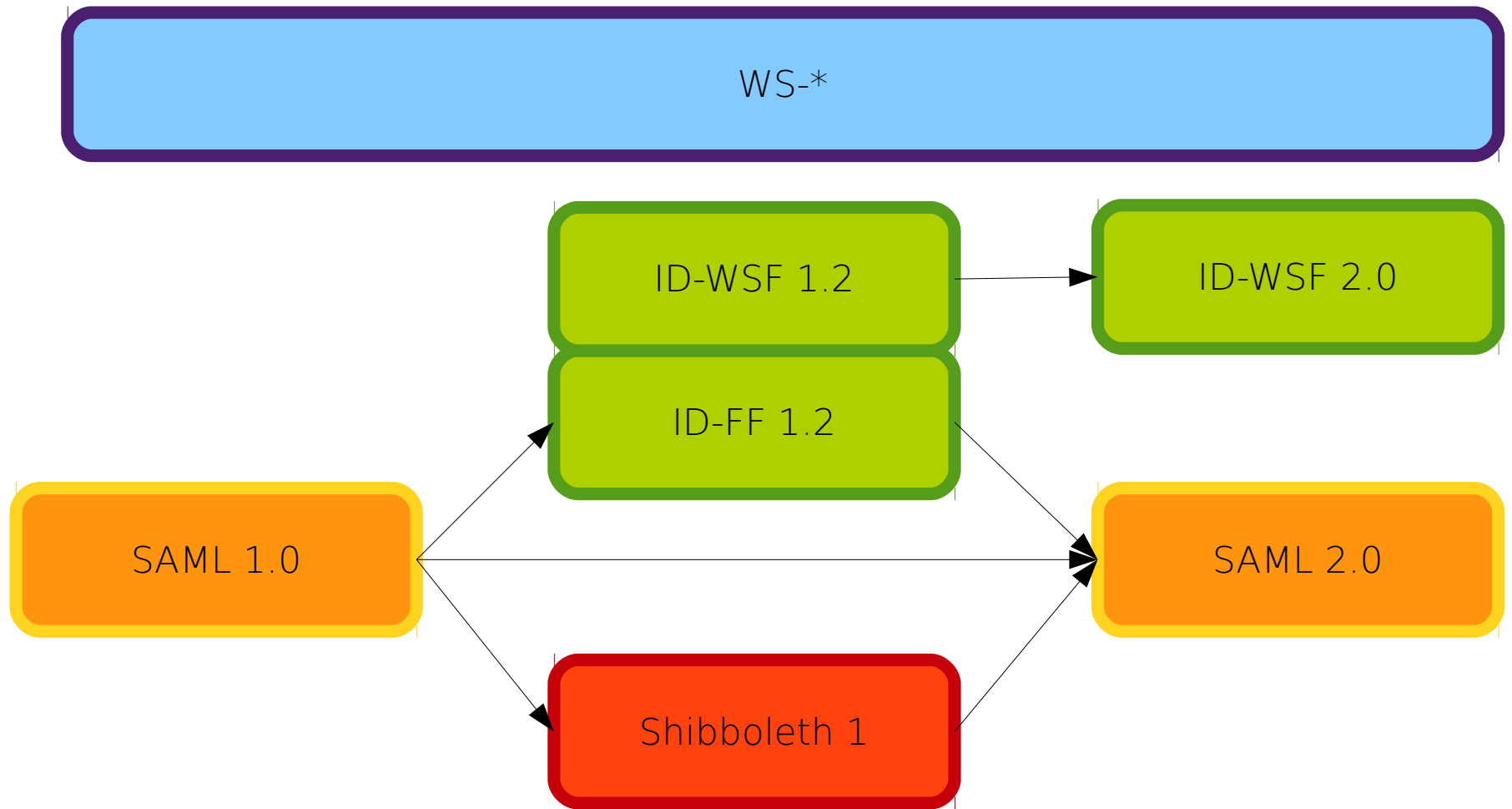
```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
ID="ONELOGIN_809707f0030a5d00620c9d9df97f627afe9dcc24" Version="2.0" ProviderName="SP test"
IssueInstant="2014-07-16T23:52:45Z" Destination="http://idp.example.com/SSOService.php"
ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
AssertionConsumerServiceURL="http://sp.example.com/demo1/index.php?acs">
  <saml:Issuer>http://sp.example.com/demo1/metadata.php</saml:Issuer>
  <samlp:NameIDPolicy Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress"
AllowCreate="true"/>
  <samlp:RequestedAuthnContext Comparison="exact">
<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</saml:Auth
nContextClassRef>
  </samlp:RequestedAuthnContext>
</samlp:AuthnRequest>
```

SAML Authn Response

```
<saml:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="_8e8dc5f69a98cc4c1ff3427e5ce34606fd672f91e6" Version="2.0" IssueInstant="2014-07-17T01:01:48Z" Destination="http://sp.example.com/demo1/index.php?acs" InResponseTo="ONELOGIN_4fee3b046395c4e75101e97f8900b5273d56685">
  <saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>
  <samlp:Status>
    <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </samlp:Status>
  <saml:Assertion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" ID="_d71a3a8e9fcc45c9e9d248ef7049393fc8f04e5f75" Version="2.0" IssueInstant="2014-07-17T01:01:48Z">
    <saml:Issuer>http://idp.example.com/metadata.php</saml:Issuer>
    <saml:Subject>
      <saml:NameID SPNameQualifier="http://sp.example.com/demo1/metadata.php" Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">_ce3d2948b4cf20146dee0a0b3dd6f69b6cf86f62d7</saml:NameID>
      <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
        <saml:SubjectConfirmationData NotOnOrAfter="2024-01-18T06:21:48Z" Recipient="http://sp.example.com/demo1/index.php?acs" InResponseTo="ONELOGIN_4fee3b046395c4e75101e97f8900b5273d56685"/>
      </saml:SubjectConfirmation>
    </saml:Subject>
    <saml:Conditions NotBefore="2014-07-17T01:01:18Z" NotOnOrAfter="2024-01-18T06:21:48Z">
      <saml:AudienceRestriction>
        <saml:Audience>http://sp.example.com/demo1/metadata.php</saml:Audience>
      </saml:AudienceRestriction>
    </saml:Conditions>
    <saml:AuthnStatement AuthnInstant="2014-07-17T01:01:48Z" SessionNotOnOrAfter="2024-07-17T09:01:48Z" SessionIndex="_be9967abd904ddcae3c0eb4189adbe3f71e327cf93">
      <saml:AuthnContext>
        <saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Password</saml:AuthnContextClassRef>
      </saml:AuthnContext>
    </saml:AuthnStatement>
    <saml:AttributeStatement>
      <saml:Attribute Name="uid" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml:AttributeValue xsi:type="xs:string">test</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute Name="mail" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml:AttributeValue xsi:type="xs:string">test@example.com</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute Name="eduPersonAffiliation" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml:AttributeValue xsi:type="xs:string">users</saml:AttributeValue>
        <saml:AttributeValue xsi:type="xs:string">examplerole1</saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
  </saml:Assertion>
</samlp:Response>
```

WS-*

- WS-Security: extension to SOAP to secure Web Services, published by OASIS
- WS-Trust/WS-Federation: developed by companies (including IBM, BEA, Microsoft) and standardized by OASIS in 2007
- STS: Secure Token Service





- Created in may 2005
- First known as Yadis (Yet another distributed identity system)
- Version 2.0 in 2006 introducing attributes exchange
- Uses URL as identifier
- Abandoned in 2014



OpenID 1.0



OpenID 2.0

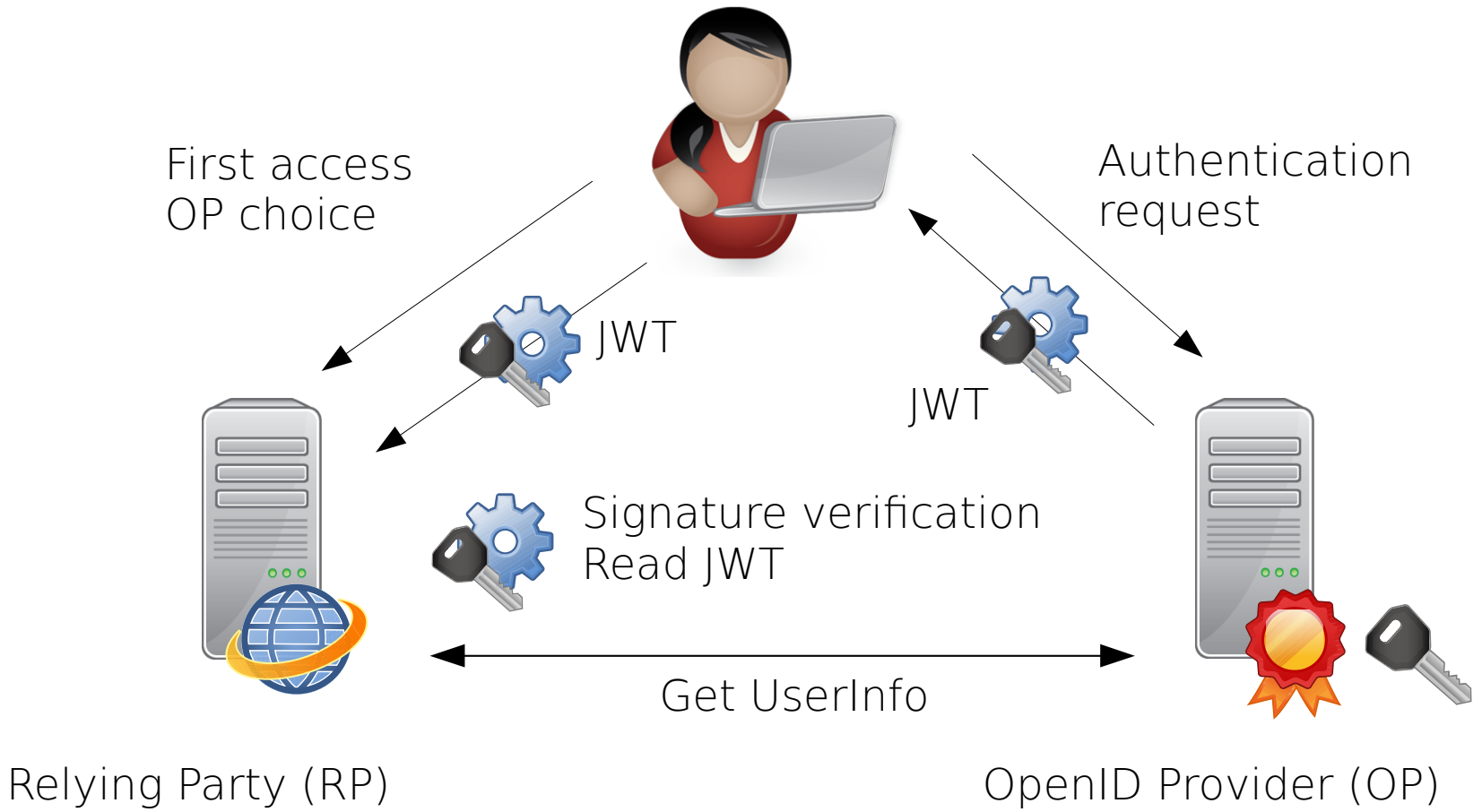


OpenID Connect

The logo for OpenID Connect, featuring a stylized 'O' with a vertical bar and an arrow pointing right.

OpenID Connect

- Created in 2014
- Presented at RMLL in 2015
- Based on OAuth 2.0, REST, JSON, JWT, JOSE
- Adapted to web browser and native mobile applications
- Attributes sharing through UserInfo endpoint





```
http://auth.example.com/oauth2.pl?  
openidconnectcallback=1;  
code=f6267efe92d0fc39bf2761c29de44286;  
state=ABCDEFGHIJKLMNOPQRSTUVWXYZ
```



```
POST /oauth2/token HTTP/1.1
Host: auth.example.com
Authorization: Basic xxxx
Content-Type: application/x-www-form-urlencoded
```

```
grant_type=authorization_code
&code=f6267efe92d0fc39bf2761c29de44286
&redirect_uri=http%3A%2F%2Fauth.example.com
%2Foauth2.pl%3Fopenidconnectcallback%3D1
```





```
{ "id_token" : "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY3liOijsb2EtMlslmF1dGhfdGltZSI6MTQzMjExMzU5MywiaWF0IjoxNDMyMTEzOTY2LCJhdF9oYXNoIjoiOWF4enNOaTlwTkRrNXpXZWZLc002QSIsImlzcyI6Imh0dHA6Ly9hdXRoLmV4YW1wbGUuY29tLylsImV4cCI6IjM2MDAiLCJhenAiOijsZW1vbmxkYXAiLCJub25jZSI6IjEyMzQ1Njc4OTAiLCJzdWIiOiJjb3Vkb3RAbGluYWdvcmluY29tliwiYXVkljpbImxlbW9ubGRhcCJdfQ==.daYGzlr37dC1R0bilwQvQLM1LICMsBFFcEufeMZtXsZvCiiAm-1LFJwJJJDHFOhd-WQnc9_GvtP3gTabXB8U4gQ2IW-bPNLUstT24njmBPYunHy8YTQ5PV-QnQI5EK5WrrTS04AF86U5Qu6m3b27yWKFXkluGI7EUvvByv8L1Anh1gPG3il5cEOnMFHIUzAaC6Pkjiy1sjSBM53nLRAF9NQ6eux4iCVBIRwl26CCgmRTsTRy-iTxB3bf0LrILohUIAR_-HPWGsealAMvqUpGeaovgGDpt4Zip9KERo7368ykgQc09VFILvZlwyMTWQdVBIYdW0oY6el9ZHjofn0mg", "expires_in" : "3600", "access_token" : "512cdb7b97e073d0656ac9684cc715fe", "token_type" : "Bearer" }
```



ID Token payload

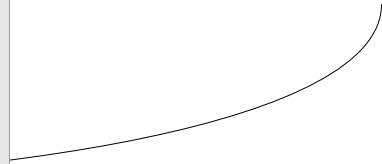
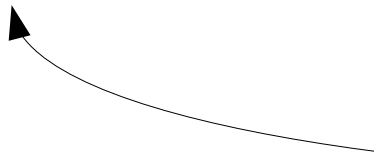
```
{
  "acr": "loa-2",
  "auth_time": 1432113593,
  "iat": 1432113966,
  "at_hash": "9axzsNi9pNDk5zWefKsM6A",
  "iss": "http://auth.example.com/",
  "exp": "3600",
  "azp": "lemonldap",
  "nonce": "1234567890",
  "sub": "clement@oudot.me",
  "aud": [
    "lemonldap"
  ]
}
```

```
POST /oauth2/userinfo HTTP/1.1
Host: auth.example.com
Authorization: Bearer 512cdb7b97e073d0656ac9684cc715fe
Content-Type: application/x-www-form-urlencoded
```





```
{  
  "name": "Clément OUDOT",  
  "email": "clement@oudot.me",  
  "sub": "clement@oudot.me"  
}
```



Thanks for your attention

Blog : <http://sifax.ca/coudot>

Twitter : @clementoudot