OpenID Connect

Identity layer on top of OAuth 2.0 Protocol

- Certification program launched April 22, 2015
- Google, Microsoft, PayPal and others were first to self-certify conformance

Dipl. Math. Xenia Bogomolec
November 2016

OpenID Connect
Inhalt

- OpenID
  - Denotation
  - Basic Concept
  - Provider
  - Provider Choice
  - Login
  - Technical
  - Provider Authentication
- OpenID Connect
  - Improvements
  - Optional Features
- OAuth
  - Registration
  - Authorization
  - Signature

OpenID
Denotation

OpenID stands for

- Open standard
- Decentralized authentication protocol
- Identifier
- Foundation
- Trademark

OpenID
Basic Concept

Identifying yourself with only one username and password on all OpenID enabled websites.

Possibilities

- Associate information with your OpenID
- Choose how much information each website gets

OpenID
Provider…

… is the only one having your password
… proves you are who you say you are
… only unifies information you have already made public
… handles sign in and log in

… is not able to say what you are (e.g. robot)
… does not track your actions *

* But keeps your logins

OpenID
Provider Choice

Considerations

- Trust
- Longevity
- Extra services
- Security
- Who you want to be identified with

List of some providers at http://openidexplained.com/get

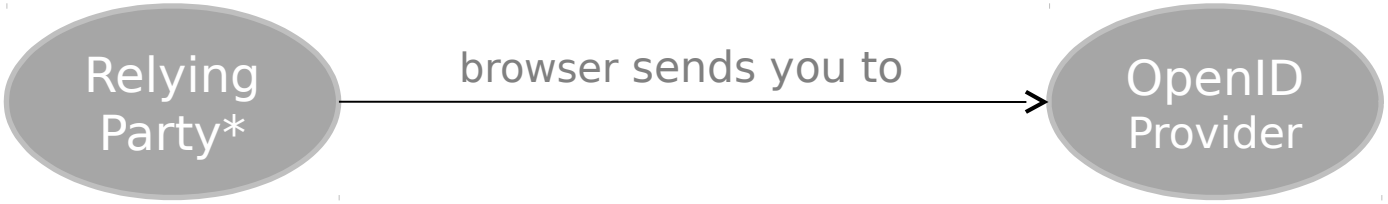
OpenID
Login

You

enter OpenID

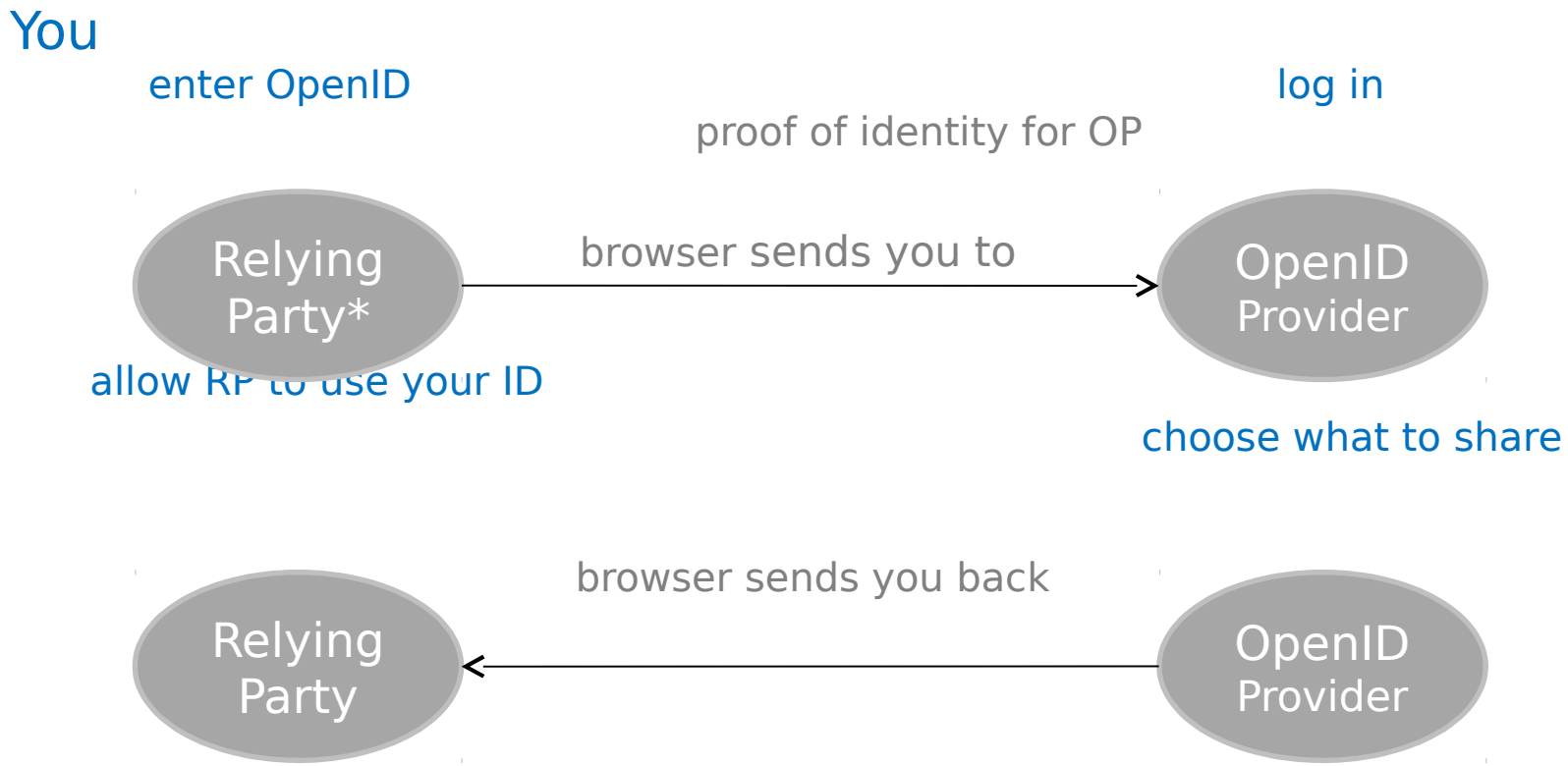
log in

proof of identity for OP



*OpenID enabled website

OpenID
Login



* OpenID enabled website

OpenID
Technical

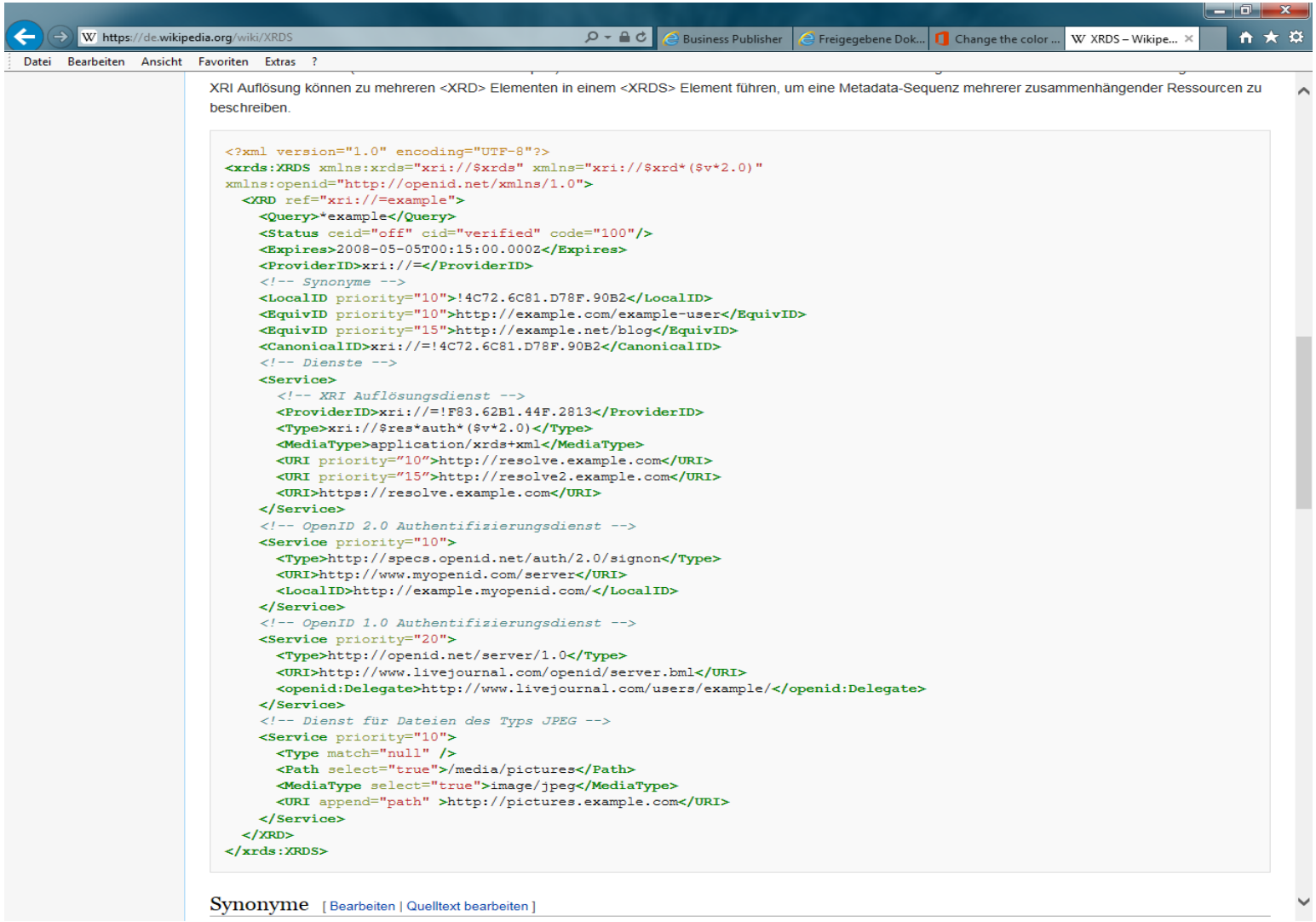
OpenID 1.0
- Requesting an HTML resource by URL derived from OpenID
- Read providers URL

OpenID 2.0
- Requesting an XRDS** by an XRI* derived from OpenID
- Read providers URL

* Extensible Resource Identifier
(standard syntax and discovery format for abstract, structured identifiers)

** Extensible Resource Descriptor Sequence

OpenID
Technical



XRI Auflösung können zu mehreren <XRD> Elementen in einem <XRDS> Element führen, um eine Metadata-Sequenz mehrerer zusammenhängender Ressourcen zu beschreiben.

```
<?xml version="1.0" encoding="UTF-8"?>
<xrds:XRDS xmlns:xrds="xri://$xrds" xmlns="xri://$xrd*($v*2.0)"
xmlns:openid="http://openid.net/xmlns/1.0">
  <XRD ref="xri://=example">
    <Query>*example</Query>
    <Status ceid="off" cid="verified" code="100"/>
    <Expires>2008-05-05T00:15:00.000Z</Expires>
    <ProviderID>xri://=</ProviderID>
    <!-- Synonyme -->
    <LocalID priority="10">!4C72.6C81.D78F.90B2</LocalID>
    <EquivID priority="10">http://example.com/example-user</EquivID>
    <EquivID priority="15">http://example.net/blog</EquivID>
    <CanonicalID>xri://=!4C72.6C81.D78F.90B2</CanonicalID>
    <!-- Dienste -->
    <Service>
      <!-- XRI Auflösungsdienst -->
      <ProviderID>xri://=!F83.62B1.44F.2813</ProviderID>
      <Type>xri://$res*auth*($v*2.0)</Type>
      <MediaType>application/xrds+xml</MediaType>
      <URI priority="10">http://resolve.example.com</URI>
      <URI priority="15">http://resolve2.example.com</URI>
      <URI>https://resolve.example.com</URI>
    </Service>
    <!-- OpenID 2.0 Authentifizierungsdienst -->
    <Service priority="10">
      <Type>http://specs.openid.net/auth/2.0/signon</Type>
      <URI>http://www.myopenid.com/server</URI>
      <LocalID>http://example.myopenid.com/</LocalID>
    </Service>
    <!-- OpenID 1.0 Authentifizierungsdienst -->
    <Service priority="20">
      <Type>http://openid.net/server/1.0</Type>
      <URI>http://www.livejournal.com/openid/server.bml</URI>
      <openid:Delegate>http://www.livejournal.com/users/example/</openid:Delegate>
    </Service>
    <!-- Dienst für Dateien des Typs JPEG -->
    <Service priority="10">
      <Type match="null" />
      <Path select="true">/media/pictures</Path>
      <MediaType select="true">image/jpeg</MediaType>
      <URI append="path" >http://pictures.example.com</URI>
    </Service>
  </XRD>
</xrds:XRDS>
```

Synonyme [Bearbeiten | Quelltext bearbeiten]

OpenID
Technical

Special for OpenID 2.0:

XRI come in two forms, i-names and i-numbers, usually registered as synonyms.

An XRI i-name used as OpenID is resolved to the synonymous i-number.

i-numbers are never reassigned.

PROTECTION OF BEING TAKEN OVER BY ANOTHER PARTY!

OpenID
Technical

Two communication modes

checkid_immediate
Relying party requests no interaction of OpenID provider with end-user

checkid_setup
End-user communicates with OpenID provider

OpenID
Provider Authentication

After successful login and confirmed sharing of credentials of the end-user, the relying party must confirm that those credentials really came from the OpenID provider (JSON Web Token).

stateful
Relying party stores secret between sessions

stateless/dumb
One more background request to ensure origin of data

OpenID Connect

Improvements

- Authentication layer that sits on top of OAuth 2.0
- Integration of OAuth 2.0 capabilities with the protocol itself
- More API-friendly way
- Better usability by native and mobile apps

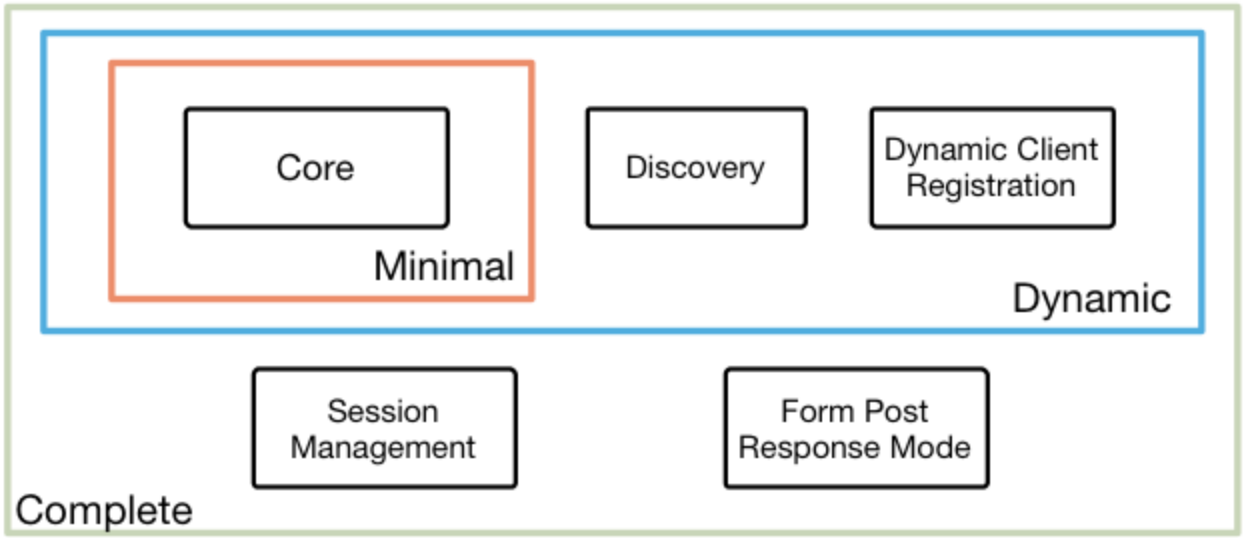
OpenID Connect
Optional Features

E.g.

- Encryption of identity data
- Robust signing
- Discovery of OpenID providers
- Session management

OpenID Connect

Protocol Suite



Underpinnings

OAuth 2.0

Registration

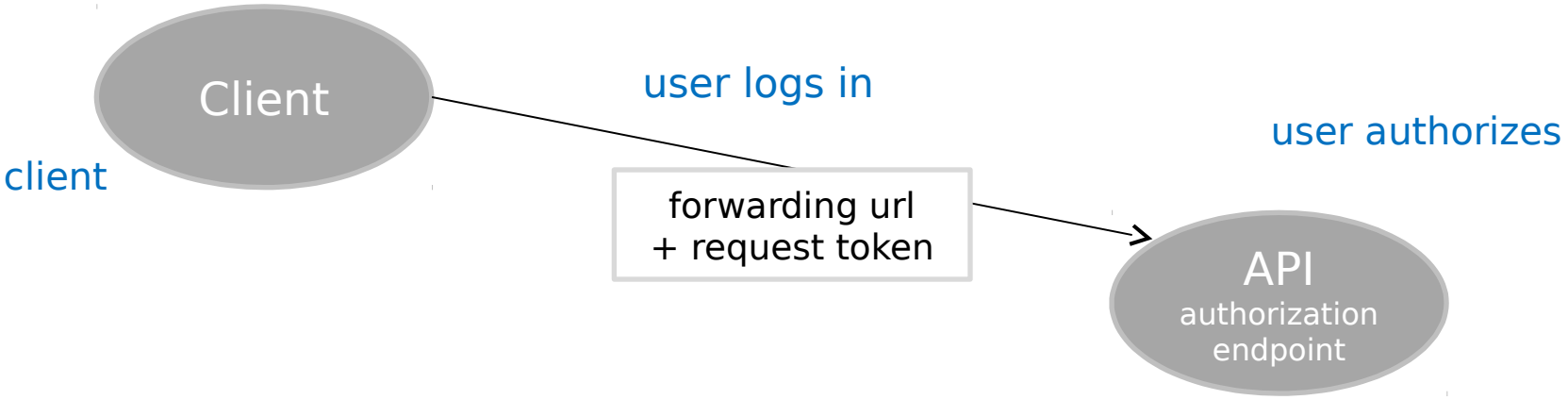
Protocol for client authorization at APIs

With the registration, the client gets

1. Consumer key
2. Consumer secret

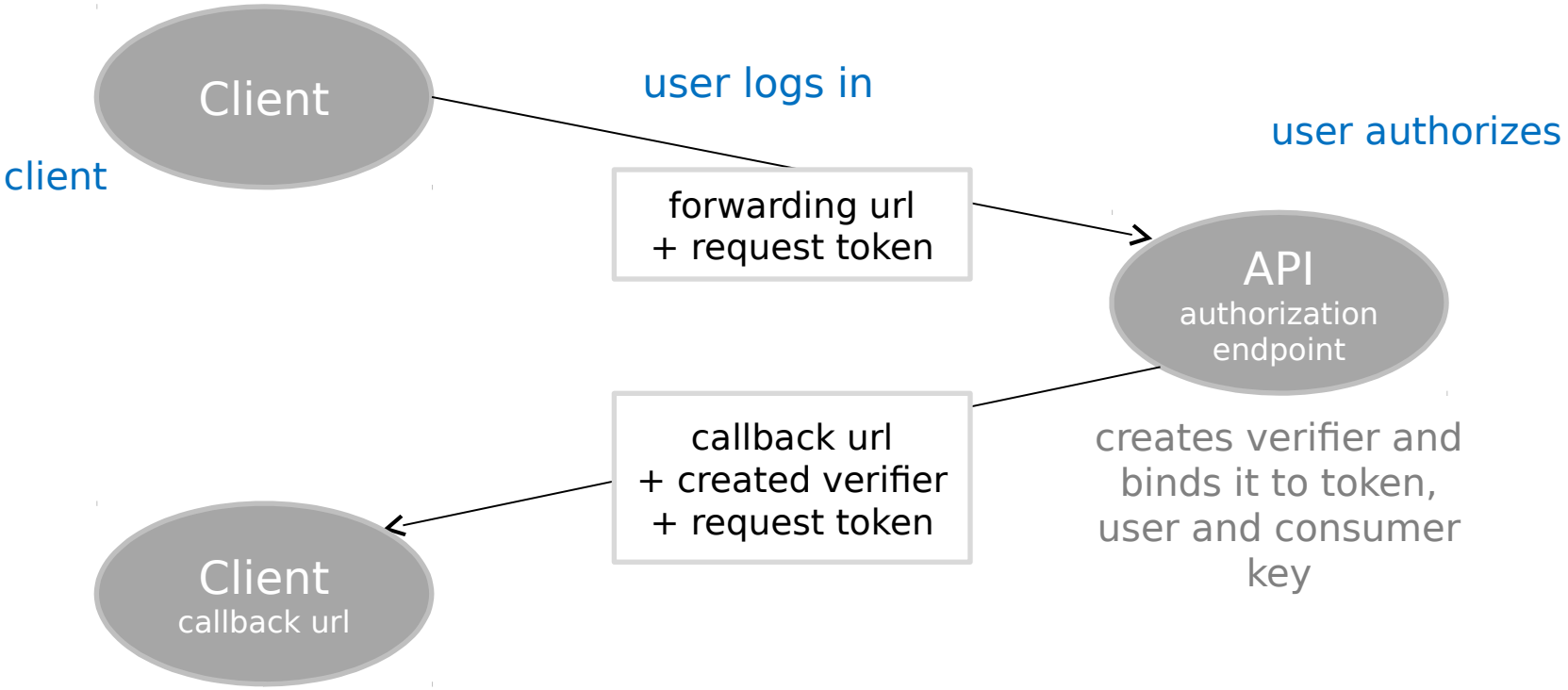
OAuth 2.0
Authentication

Step 1: User is redirected



OAuth 2.0
Authentication

Step 1: User is redirected



OAuth 2.0

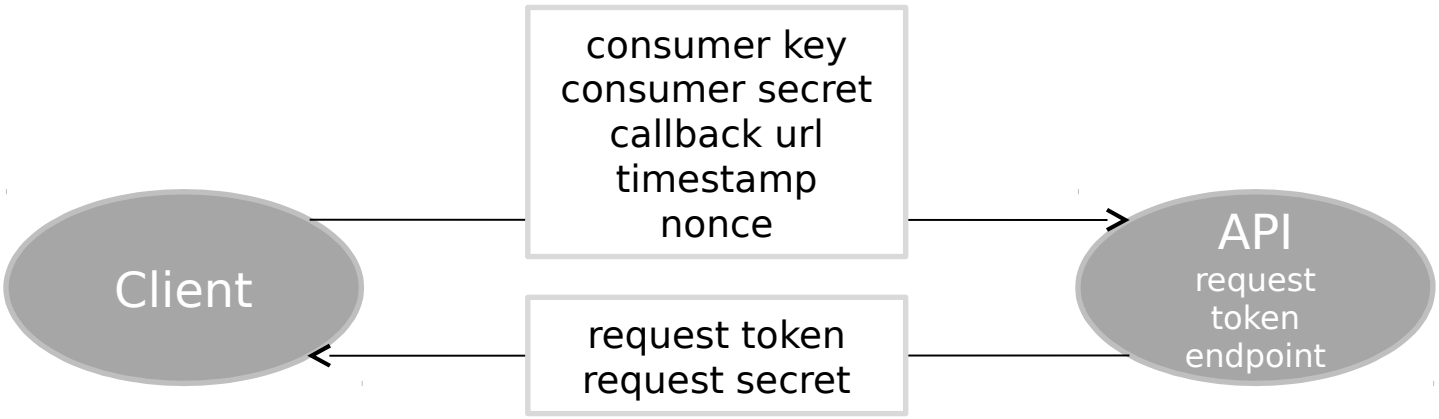
Authorization

Step 2

OAuth 2.0

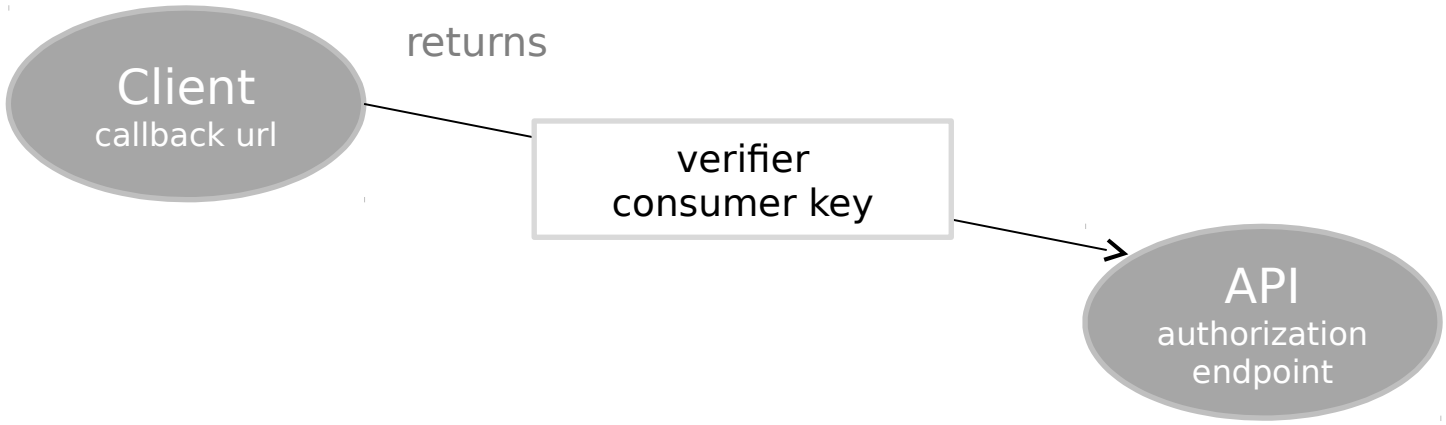
Authorization

Step 2



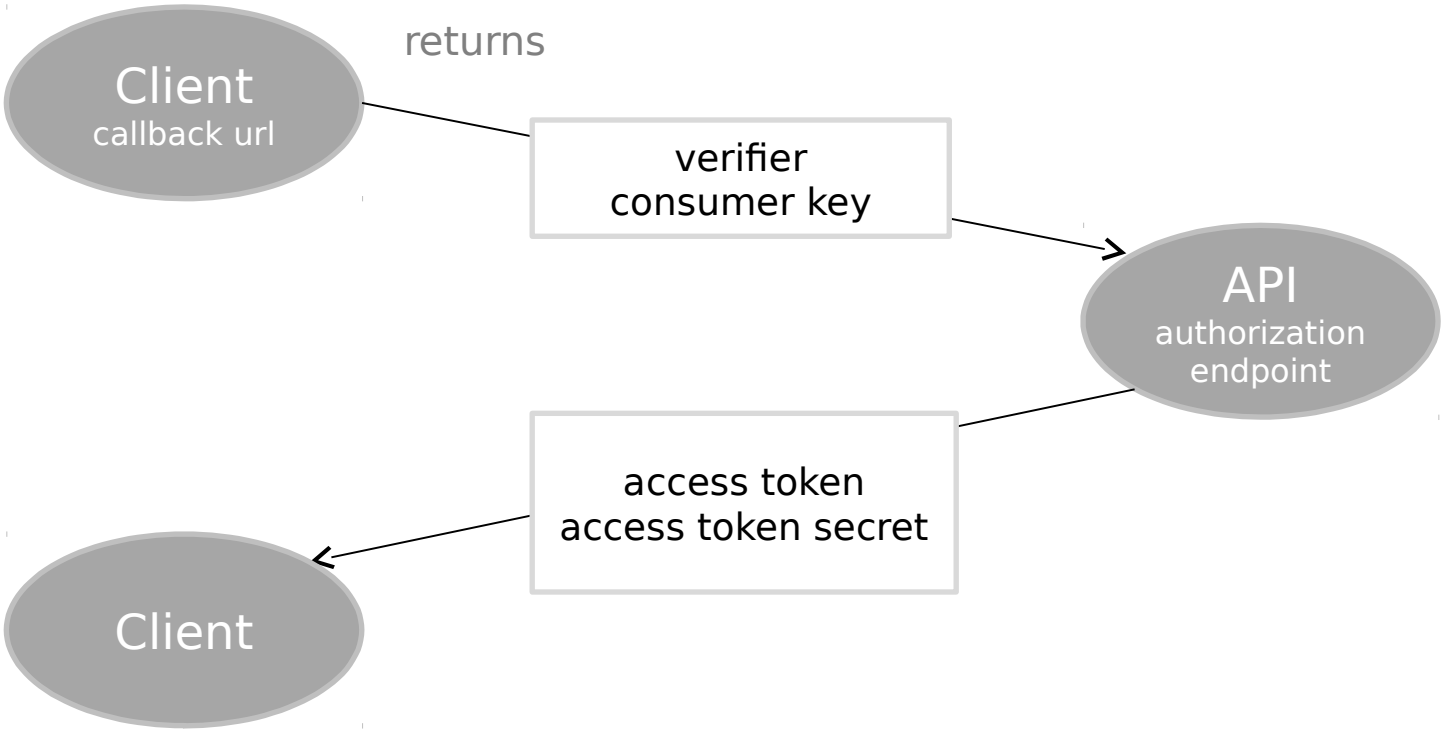
OAuth 2.0
Authorization

Step 2:



OAuth 2.0
Authorization

Step 3:



OAuth 2.0

Signature

All server to server requests are signed with an OAuth signature, which is based on the whole normalized request. It includes

1. HTTP method
2. URL incl. sorted query parameters
3. Hash of request body

Then the normalized request concatenated with the consumer secret gets hashed.

Various hash algorithms are available.

In OpenID Connect, the id_token includes a verification of the OpenID Provider by a symmetric or an asymmetric signature.