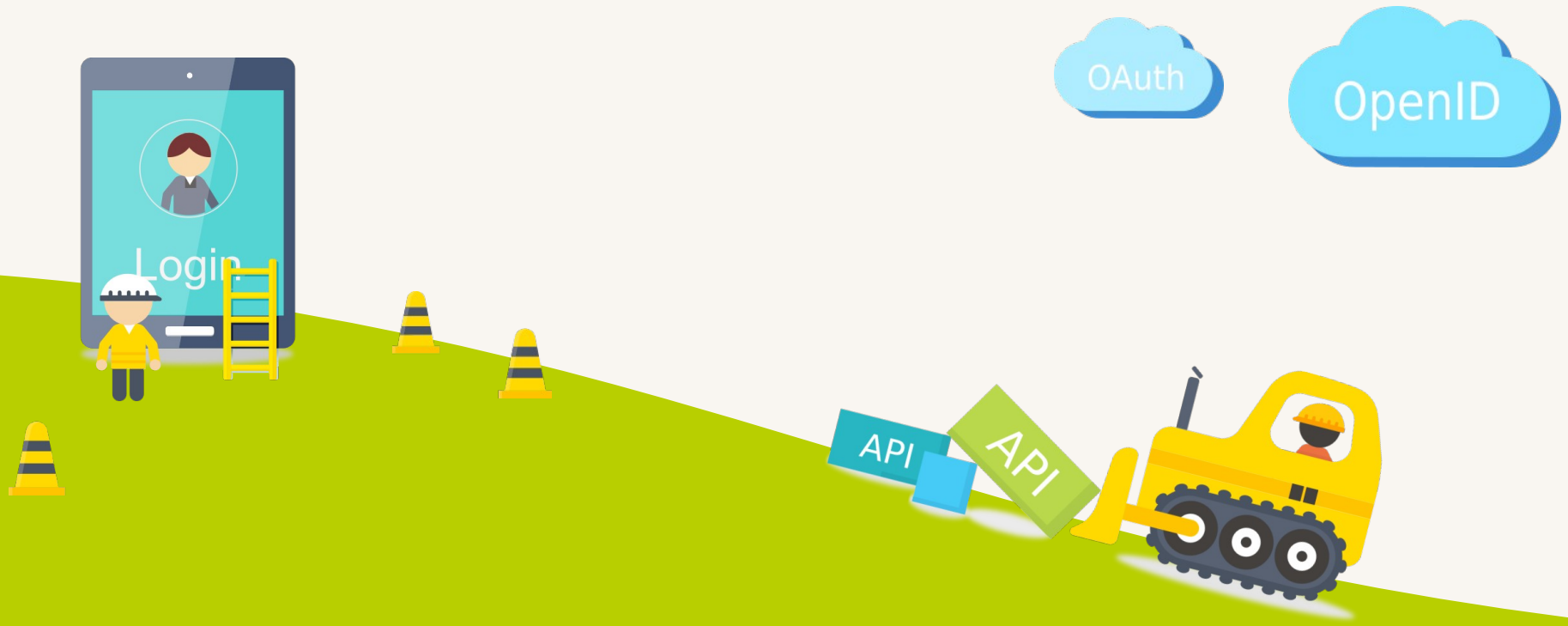


# connect@id

CONNECTING APPS TO IDENTITIES



**OpenID Connect & OAuth 2.0 Server  
for the Enterprise**

# Your enterprise server for

**single  
sign-on**

**identity  
provision**

**identity  
federation**

**API access  
management**

The four Connect2id server pillars

# Based on the latest standards

**OpenID Connect**



**for ID tokens**

**OAuth 2.0 / 2.1**



**for access tokens**

Modern token-based security for web, mobile and native applications

# Identity and security profiles

**FAPI**

financial-grade  
API security

**IdA / eKYC**

verified identities and data,  
AML compliance

**HEART**

electronic health record  
access and exchange

**iGov**

international government  
assurance profile

**Federation**

operate hierarchical and  
mesh federations at scale

**others to follow**

...

Supported industry profiles for Open Banking, government / eID, health care

# Engineered for

**easy  
integration**

**365/24/7  
uptime**

**scaling +  
performance**

**agile  
dev ops**

Move fast and with confidence

# Providing identity services to



every **100<sup>th</sup>** person\*  
on the planet,  
and growing...

\* 90 mio end-users as of July 2017

# Easy integration

UI / UX



User authN



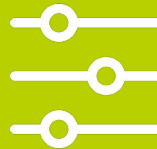
AuthZ logic



Claims



Admin



Monitoring



We want to liberate our customers. Smart web-based (REST + JSON) and native (Java SPI) integration for flexibility and performance.

# Sign-in experience

**Login**

User

Password

**Consent**

Allow Wonderland App access to your :

email

profile

deny

Design your own branded user experiences around login and consent



# Sign-in experience

- A powerful guided web API lets you integrate a sign-in experience **branded** and **tailored** specifically for your enterprise or SaaS.
- Choose **any language** and **framework** for your UI and authN / authZ logic. Save time and money, leverage your existing competence and resources.
- **Zero service downtime** for updates to the login page.
- You can even have **multiple dedicated login pages**, e.g. one for employees, another for contractors and a third for customers.

# User authentication

- All types of user authentication can be plugged in via the login [web API](#) to match your security needs.
- [Microsoft Active Directory](#) / [LDAP](#) authentication is supported out of the box.
- You're free to integrate any other authentication method, such as [one-time passwords](#) and [biometrics](#).
- The Connect2id server never has to deal with user credentials directly, which is good for [security](#).

## Submitting a user authentication

```
{  
  "sub"      : "alice",  
  "auth_time" : 1604392924,  
  "acr"      : "c2id.loa.high",  
  "amr"      : [ "pwd", "otp"]  
}
```

# Example authentication methods

LDAP \*

One-time  
password  
(OTP)

x.509  
certificate

SQL  
database

secure remote  
password  
(SRP-6a)

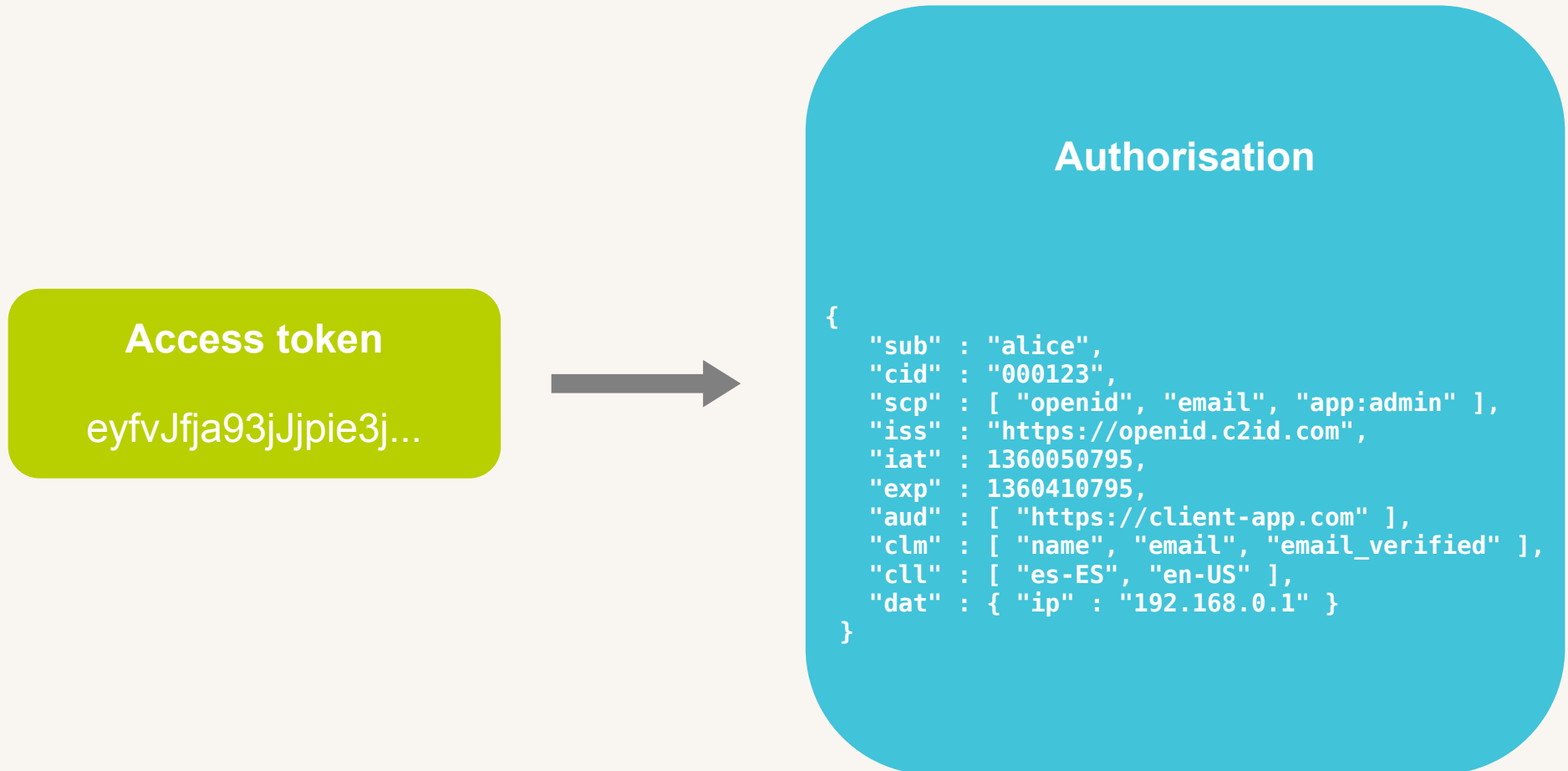
biometrics

\* Supported out of the box

# Your OAuth 2.0 authorisation server

- The Connect2id server can act as an OAuth 2.0 **authorisation server** to issue access tokens to clients.
- Supports all core OAuth 2.0 **grants**: code, implicit, password, client credentials. **SAML 2.0** and **JWT Bearer** assertion grants are also accepted.
- Can generate **self-contained** (JWT) as well as **identifier**-based bearer access tokens. JWT-encoded access tokens are ideal for **distributed applications**.
- The issued tokens can be client x.509 certificate (**mTLS**) bound for extra security in financial (**FAPI**) and other applications.
- You can plug in **arbitrary logic** for consent (explicit / implicit), to customise tokens and their introspection.

# Access token attributes



Access tokens can be decoded and verified on the spot (JWT)  
or inspected at a Connect2id server endpoint

# Managing existing authorisations

- You can **query** and **manage** the **authorisations** for each user and client application via a dedicated web API.
- Authorisations can be persisted so that users are **not asked again** for **previously consented** scope values and claims.
- You can build a **UI** or a **risk management agent** to **revoke tokens** for a user, client or combination thereof.

# Revocation UI

Alice : Your authorised apps

- **Wonderland App** [ edit ] [ revoke ]
- **Weather App** [ edit ] [ revoke ]
- **Bookstore App** [ edit ] [ revoke ]

Design your own UIs and tools for managing authorisations

# UserInfo

```
{  
  "sub"           : "alice",  
  "name"          : "Alice Adams",  
  "given_name"   : "Alice",  
  "family_name"  : "Adams",  
  "email"         : "alice@wonderland.net",  
  "email_verified" : true,  
  "phone_number" : "+359 (88) 200305",  
  "profile"       : "https://c2id.com/users/alice",  
  "ldap_groups"  : [ "audit", "admin" ]  
}
```

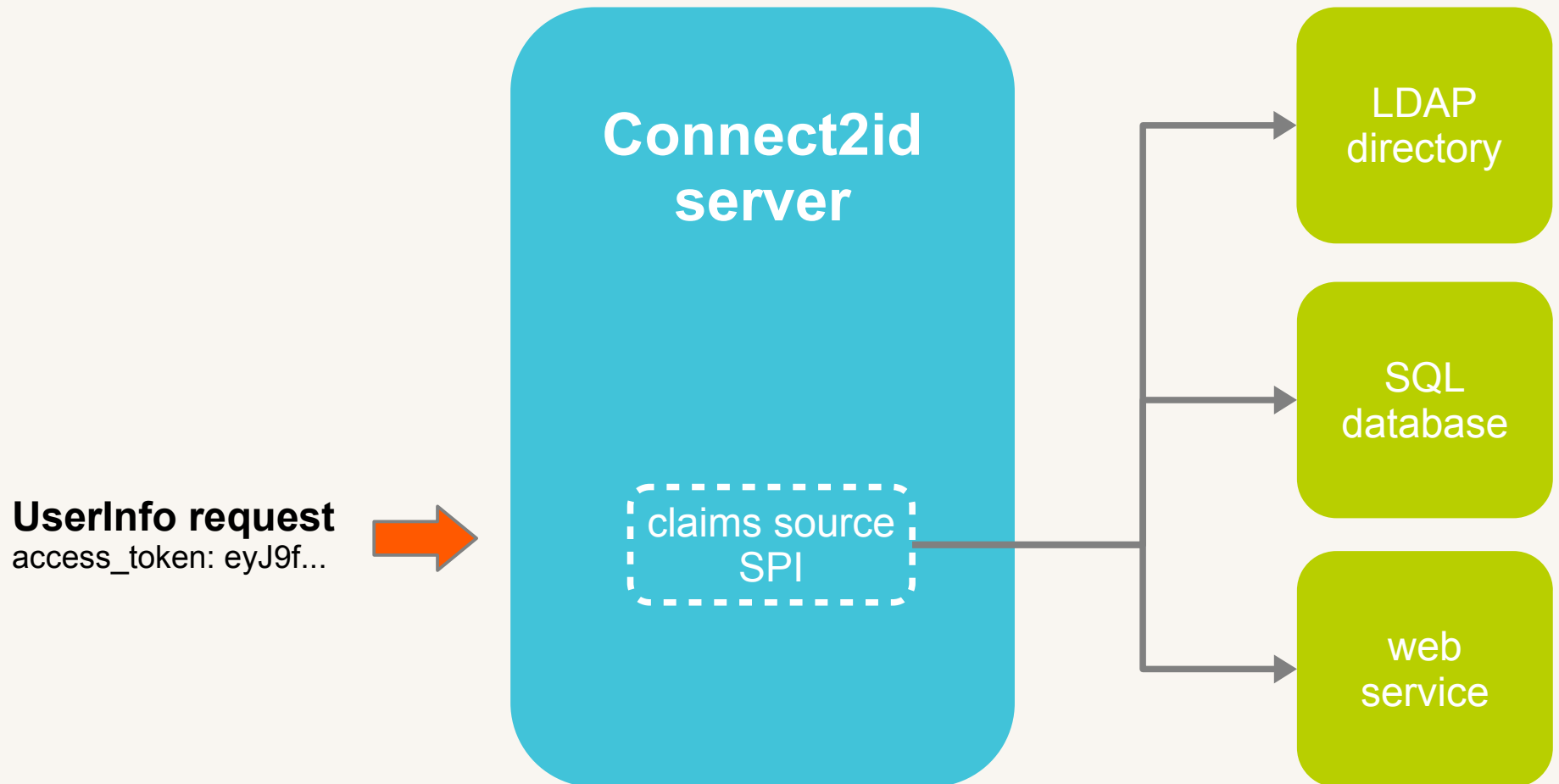
OpenID Connect defines an extensible JSON schema for releasing consented user details (OpenID claims) to client applications



# OpenID claims sources

- OpenID Connect defines a simple extensible **JSON** schema for releasing consented **user information** (claims), such as name, profile and contact details, to client applications.
- The **claims** can be included in the **ID token**, returned at the **UserInfo endpoint**, or even piped into **access tokens** for resource server consumption.
- Support for **verified claims** and data (eKYC).
- The Connect2id server supports **aggregation** of UserInfo claims from one or more **data sources** (LDAP directory, HR database, etc).
- Claims sources can be integrated via a **Java SPI** or a **web hook**.
- **Microsoft Active Directory** / **LDAP** supported out of the box.

# Claims source aggregation



OpenID claims aggregation from multiple data sources

# Managing user sessions

- User sessions can be **queried**, **monitored** and **managed** via a dedicated **web API** (e.g. who is online?)
- The **login page** may store arbitrary **attributes** in the user session, to personalise the UI or for other purposes.
- Client applications can initiate standard **logout requests**.
- Clients can also receive standard front and back-channel **logout notifications**.



# User session object

```
{
  "sub"           : "alice",
  "auth_time"    : 1604392924,
  "acr"          : "c2id.loa.high",
  "amr"          : [ "pwd", "otp" ]
  "creation_time" : 1604392924,
  "max_life"     : 20160,
  "auth_life"    : 1440,
  "max_idle"     : 15,
  "data"         : { "name" : "Alice Adams",
                    "email" : "alice@wonderland.net" }
}
```

Rich session attributes with support for arbitrary data

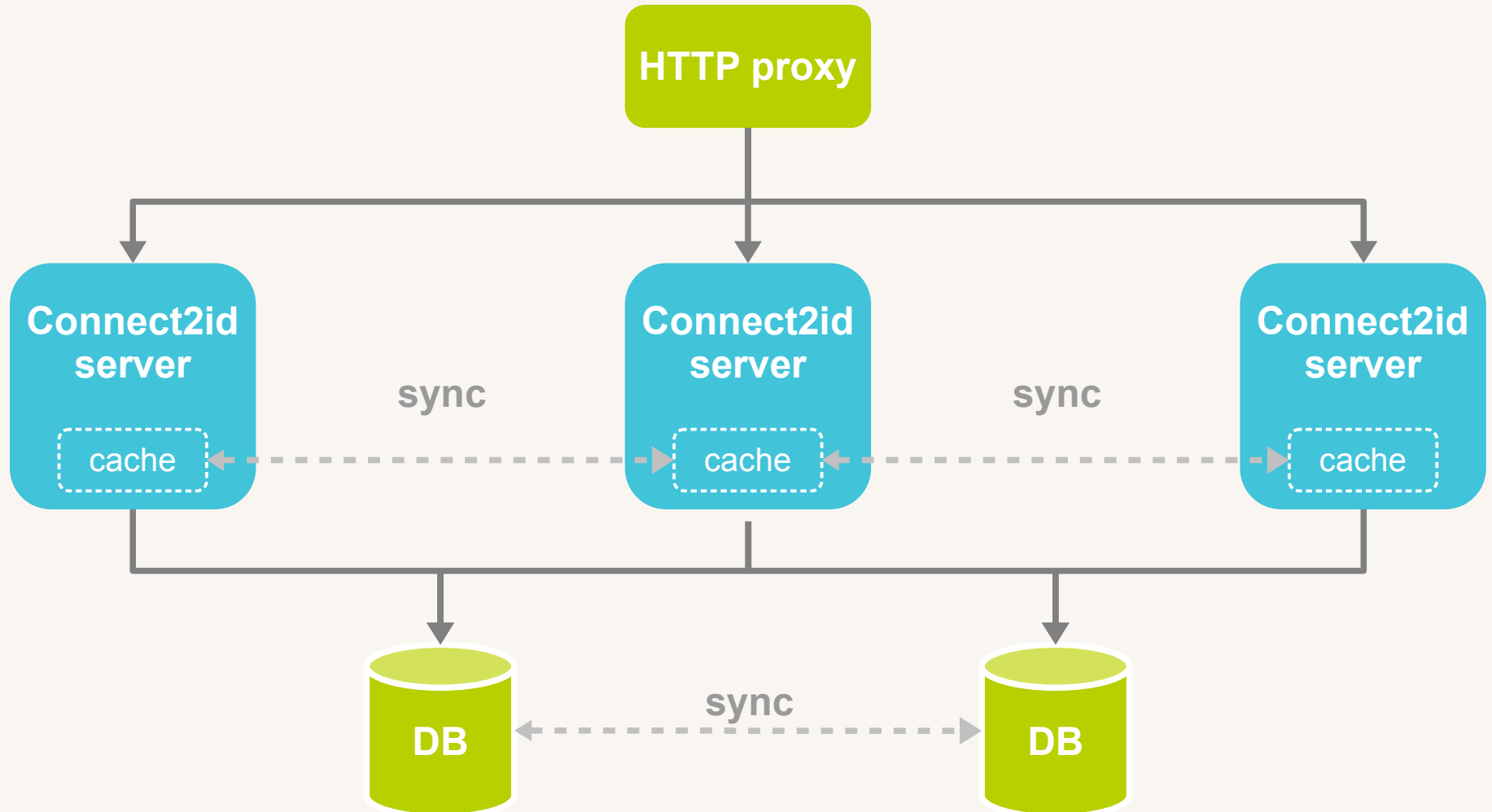
# Engineered for 365/24/7 uptime

Identity services can be critical to relying applications.

The Connect2id server is designed from the ground up for **continuous availability**:

- Avoiding **single points** of failure: the web service layer and the underlying database can be **clustered** for high-availability.
- Seamless **scaling**: server and database nodes can be added or removed to / from the cluster when required.
- Seamless **upgrades**: the software is designed for upgrades with zero disruption to service. Front-ends, OAuth 2.0 grant handlers and claims sources are decoupled from the main service.

# Connect2id server cluster



Choice between stateless (with optional Redis cache) and replication clustering

# Scaling + performance

- For **small and medium organisations** (~ thousands of users) the Connect2id server can be run in a VM with 1 CPU core and 2 GB RAM.
- **Large user bases** can benefit from a Connect2id **cluster** where the OpenID Connect / OAuth 2.0 requests are load-balanced over multiple nodes.
- Selected **asynchronous** operations for improved **responsiveness**.
- Connect2id server nodes can be dynamically **added** or **removed** to / from the cluster to **match demand**.
- **Redis** can be optionally deployed as primary cache.

# Server monitoring

- Backend database **health checks**
- Monitoring endpoint with **120+ metrics**:
  - sign-in activity
  - detailed endpoint stats
  - OAuth 2.0 grant handler stats
  - claims sources latency and performance
  - database latency and performance
- **Token issue events** for audit and accounting purposes





# DevOps friendly

Key DevOps jobs can be done **safely** and without impacting the **uptime** of a running Connect2id server / cluster:

- Updating the OpenID Connect **login UI** or testing new ones;
- Upgrading the **authentication** method or incorporating new second factors (e.g. **FIDO** OTP or biometrics);
- Updating the **user** and **administrative interfaces** for the service or introducing new ones;
- Updating UserInfo **claims sources** (for web-based ones).

To find out more about the  
Connect2id server

<https://connect2id.com/server>