

connect@id

CONNECTING APPS TO IDENTITIES



Your enterprise server for

**single sign-on
(SSO)**

**identity
provision**

**identity
federation**

**API access
management**

The four IdM pillars of the internet enterprise

Based on the latest standards

OpenID Connect



for ID tokens

OAuth 2.0



for access tokens

Modern token based security for web, mobile and native apps

Engineered for

**easy
integration**

**100%
uptime**

**scaling +
performance**

**agile
dev ops**

Move fast and with confidence

Used to provide identity services to



1%

of the world population *
and growing...

* 70 mio end-users as of sept 2015

Easy integration

UI / UX



User auth



Authz logic



Claims src



Admin



Monitoring



We want to liberate our customers. Enjoy simple web-based integration (REST + JSON), designed to give you lots of power and flexibility.

Sign-in experience

Login

User

Password

Consent

Allow Wonderland App access to your :

email

profile

[deny](#)

Design your own user experiences around login and consent

Sign-in experience

- A powerful web API lets you integrate a sign-in experience **branded** and **tailored** specifically for your enterprise or online service.
- Choose **any language** or **framework** for your UI and 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 your employees and another for your customers.

User authentication

- **Arbitrary** user authentication methods can be plugged in via simple **web API** to match your security needs.
- **MS-AD / LDAP** authentication is supported out-of-the box.
- You're free to integrate any other authentication method, such as **hardware tokens** or **biometrics**.
- The Connect2id server never has to deal with passwords directly, which is good for **security**.

User authentication

```
{  
  "sub"      : "alice",  
  "auth_time" : 12345678,  
  "acr"      : "c2id.loa.high",  
  "amr"      : [ "pwd", "otp"]  
}
```

Example authentication methods

LDAP *

hardware
tokens

x.509
certificate

SQL DB

secure remote
password

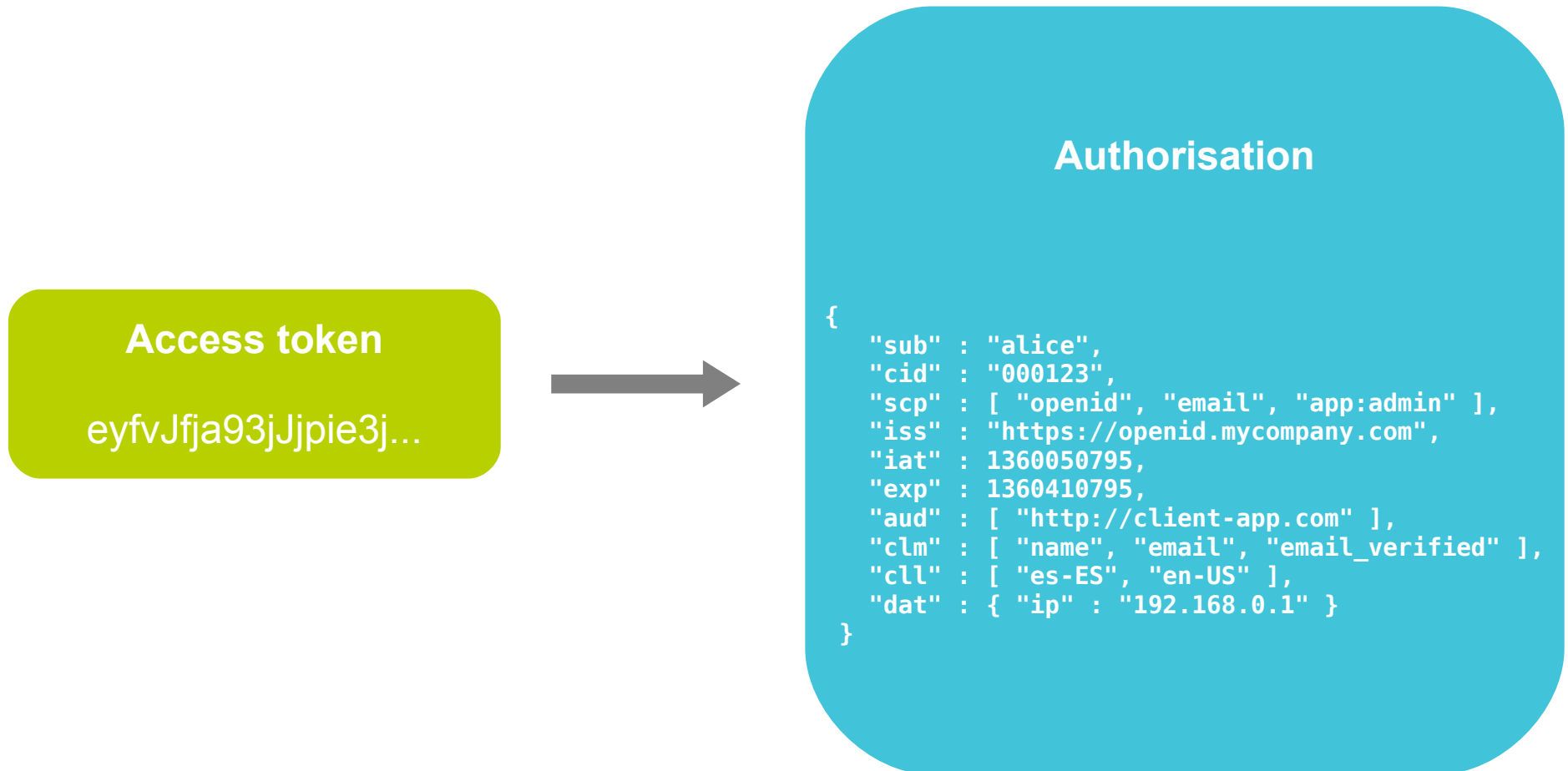
biometrics

* 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.
- Additional grants, such as **SAML** and **JWT Bearer** may be accepted via special endpoint.
- Can generate **self-contained** (JWT) as well as **identifier**-based bearer access tokens. JWT access tokens are ideal for **distributed applications**.
- You can plug in **arbitrary logic** to drive consent (explicit / implicit) and customise tokens.

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 dedicated web API.
- Authorisations can be persisted so that the user is **not asked again** for **previously consented** scope values and claims.
- You can build an **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

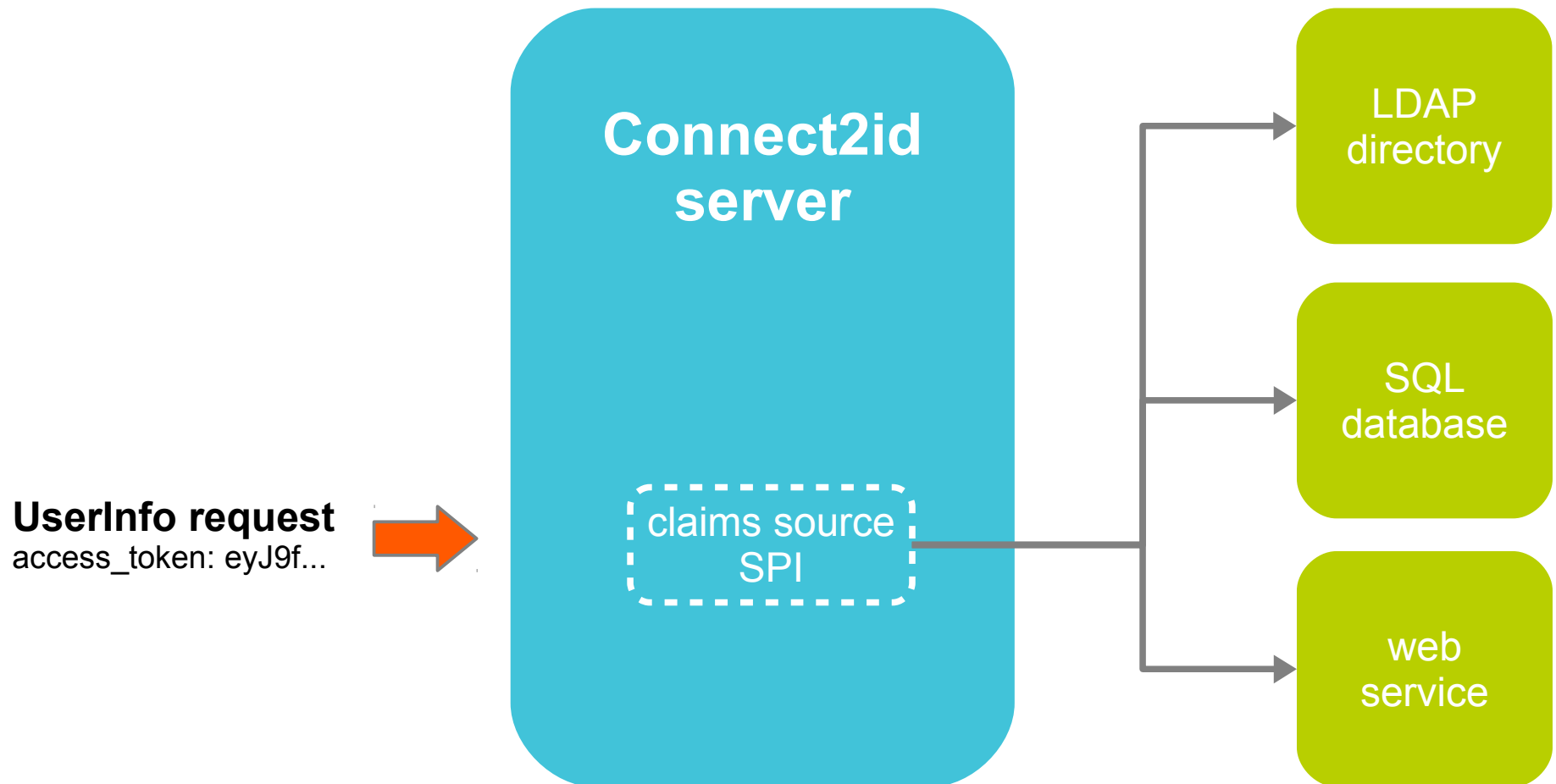
```
{  
  "sub"           : "alice",  
  "name"          : "Alice Adams",  
  "given_name"    : "Alice",  
  "family_name"   : "Adams",  
  "email"         : "alice@wonderland.net",  
  "email_verified" : true,  
  "phone_number"  : "+359 (99) 100200305",  
  "profile"       : "https://c2id.com/users/alice",  
  "ldap_groups"   : [ "audit", "admin" ]  
}
```

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

UserInfo claims sources

- OpenID Connect defines a simple **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** or returned at the **UserInfo endpoint** (requires an access token).
- 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 **web interface**.
- **MS-AD / LDAP** directories are supported out of the box.

Claims source aggregation



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.
- Custom **logout** callbacks / agents may be implemented (on the roadmap).



User session object

```
{  
  "sub"           : "alice",  
  "auth_time"    : 12345678,  
  "acr"          : "c2id.loa.high",  
  "amr"          : [ "pwd", "otp" ],  
  "creation_time" : 1234567,  
  "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

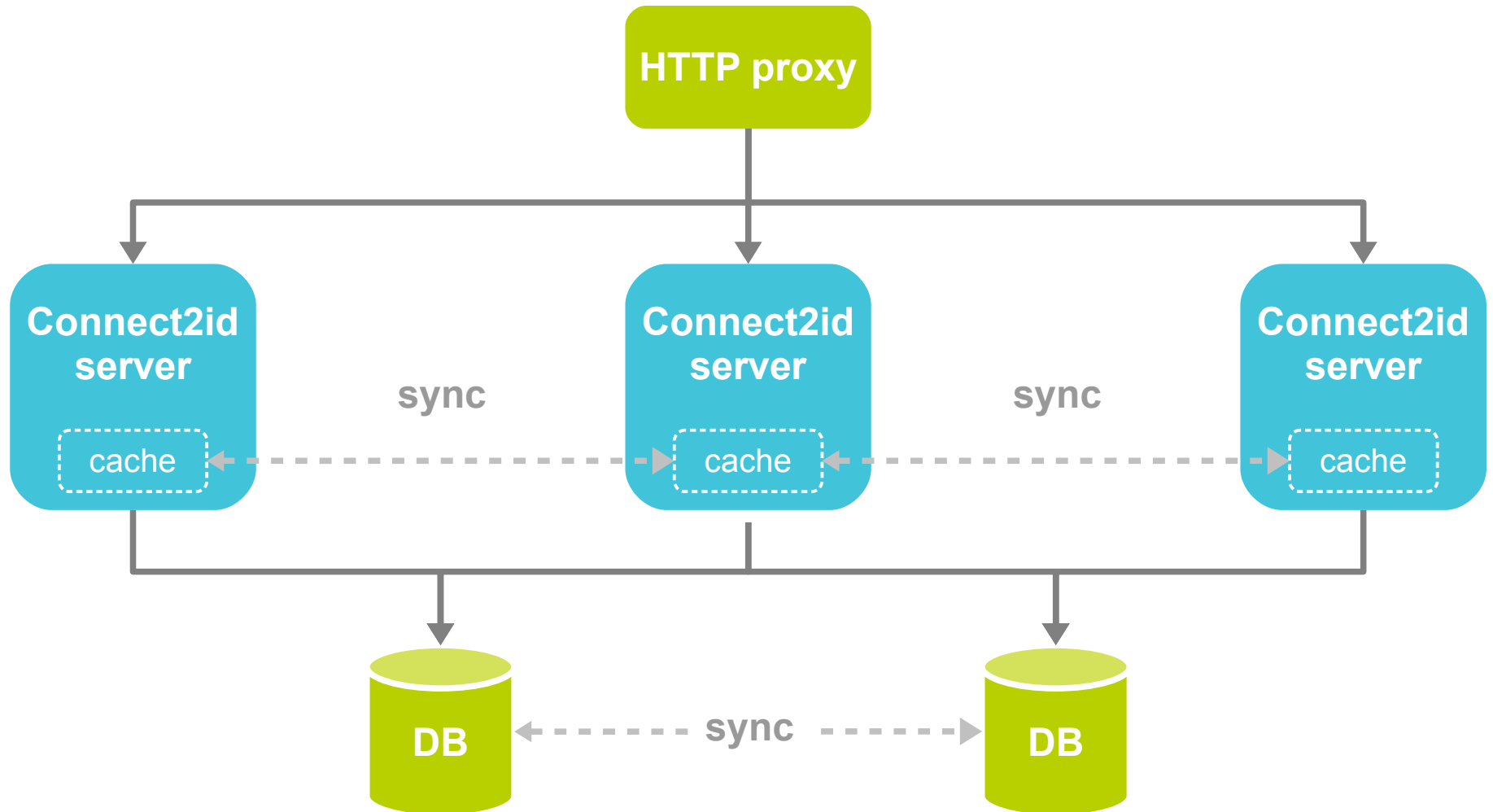
Engineered for 100% uptime

Identity services can be critical to relying applications.

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

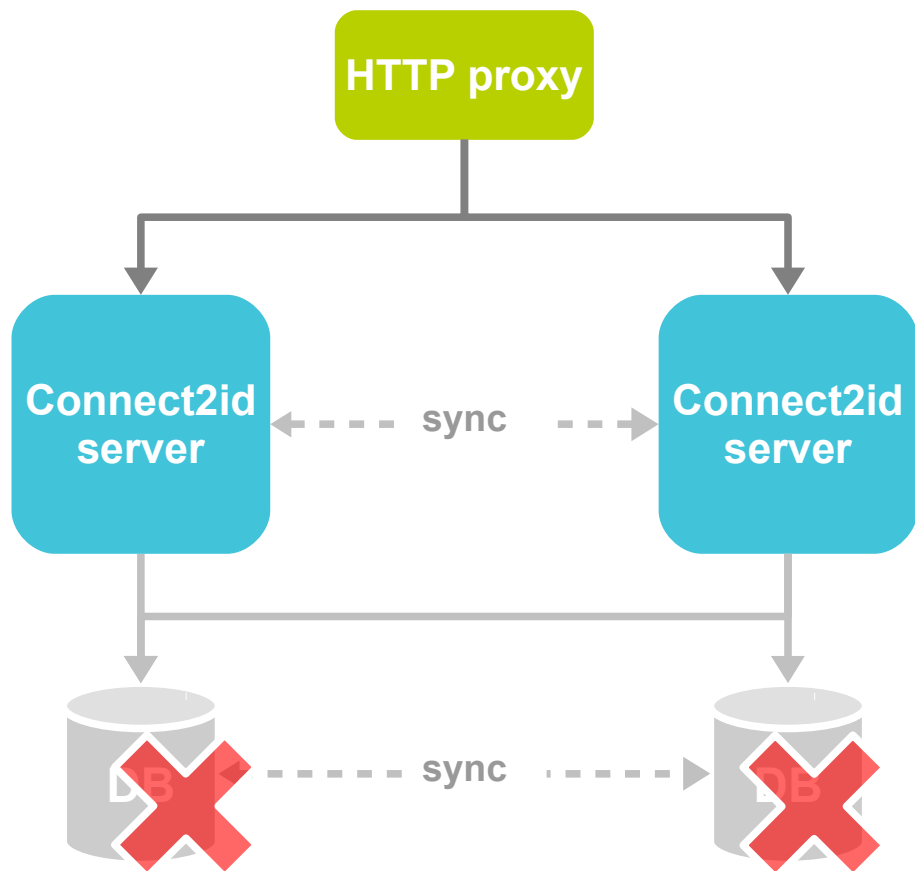
- Avoiding **single points** of failure: the web service layer and the underlying database can be **clustered** for high-availability (HA).
- Seamless **scaling**: server and database nodes can be added or removed to / from the cluster when required.
- Seamless **upgrades**: the software is designed for upgrades with zero disruption to service.

Connect2id server cluster



The Connect2id server supports clustering at the web service and DB tiers

For your peace of mind



- In case of a database or storage layer **crash** the Connect2id server can **maintain indefinitely** key OpenID Connect / OAuth 2.0 services.
- The server nodes **cache** all important client registration and authorisation data. This not only makes the service more **responsive**, but also protects it against **DB outages**.

Scaling + performance

- For **small organisations** (up to a few thousand users) the Connect2id server can run on a virtual host with 1 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.
- The nodes communicate **asynchronously** which greatly improves **responsiveness**.
- Connect2id server nodes can be dynamically **added** or **removed** to / from the cluster to **match demand**.

Server monitoring

- Database backend **health checks**
- Monitoring endpoint providing over **100 metrics**:
 - sign-in activity
 - detailed endpoint stats
 - OAuth 2.0 grant handler stats
 - claims sources latency and performance



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 a new one (e.g. hardware tokens);
- 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

<http://connect2id.com/server>