

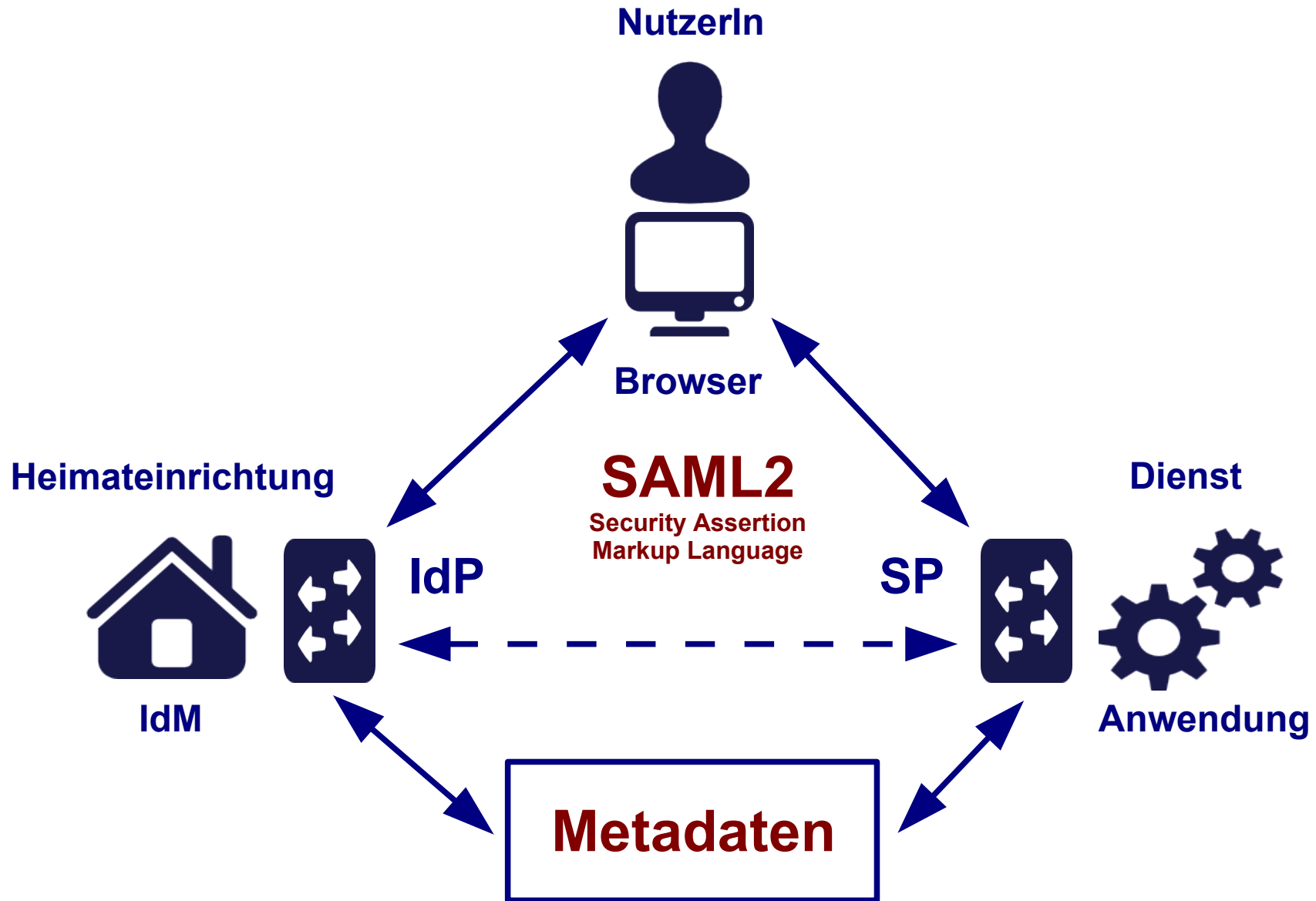
# OpenID Connect

## im Einsatz auf Föderationsebene

Wolfgang Pempe, DFN-Verein  
[pempe@dfn.de](mailto:pempe@dfn.de)

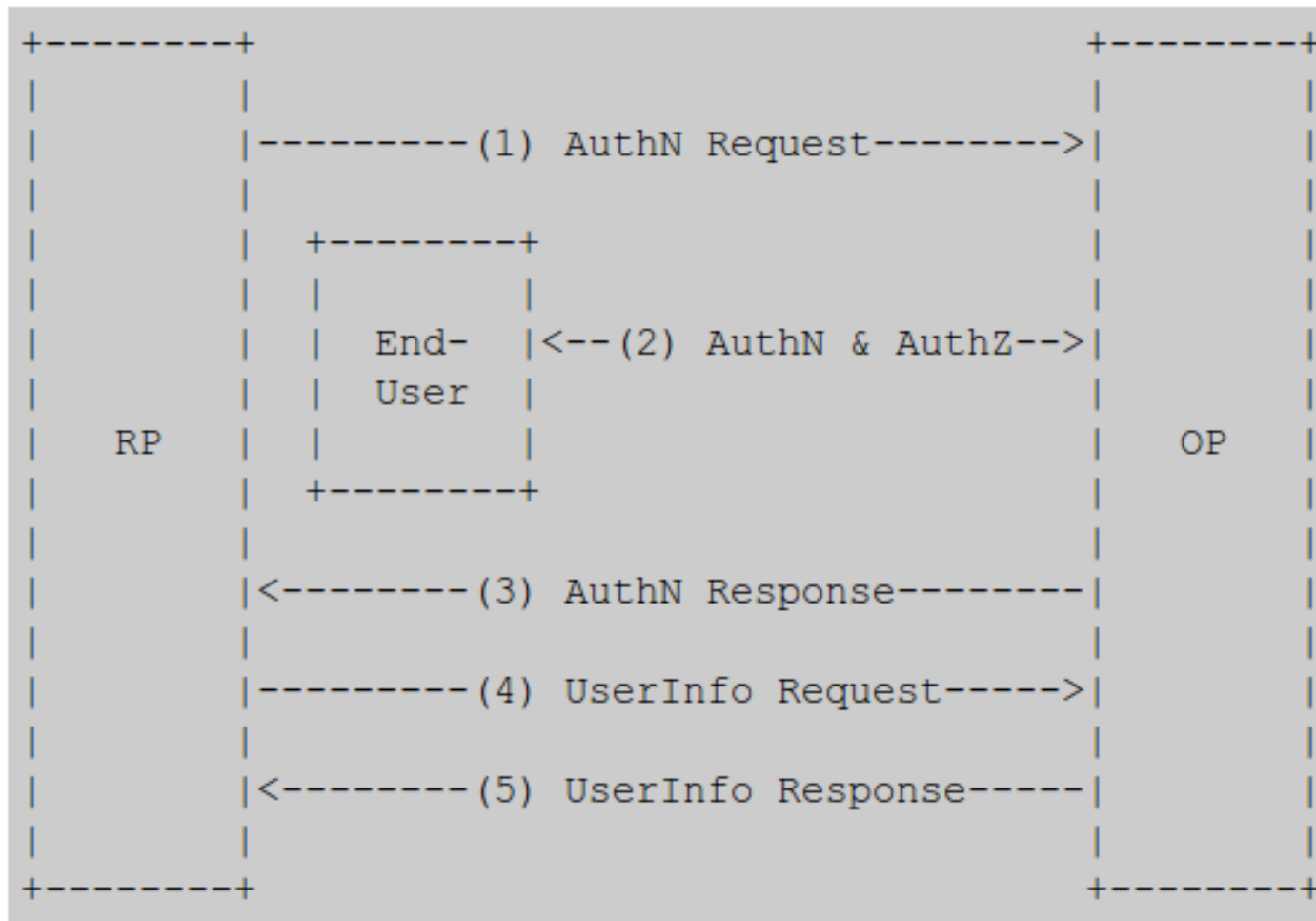
ZKI AK Verzeichnisdienste,  
Herbsttreffen 12. September 2016, Ulm

- "A simple identity layer on top of the OAuth 2.0 protocol"
- Standard wurde im Februar 2014 verabschiedet
- Protokoll basiert auf REST/JSON + JWT ([JSON Web Token](#)), funktioniert auch ohne Web Browser (→ mobile Endgeräte, Apps)
- Entwicklung wurde und wird von diversen Internet-Konzernen getrieben: Google, Facebook, Microsoft, Deutsche Telekom, PayPal, Yahoo! u.a.m.
- Kommt u.a. bei Google+ zum Einsatz
- Wird von manchen als Konkurrenz zu SAML gesehen
- Infos, Spezifikationen, Software etc. unter <http://openid.net/connect/>



Siehe auch: <https://wiki.shibboleth.net/confluence/display/CONCEPT/Home>

ähnlich wie SAML ...



Quelle: [http://openid.net/specs/openid-connect-core-1\\_0.html](http://openid.net/specs/openid-connect-core-1_0.html)

<b>OpenID Connect (OIDC)</b>	<b>SAML</b>
OP (OpenID Provider)	IdP (Identity Provider)
RP (Relying Party), Client	SP (Service Provider)
Claim (Information über eine Entity, z.B. User)	Bestandteil einer Assertion (Attribut, NameID)

Details unter

[http://openid.net/specs/openid-connect-core-1\\_0.html#Terminology](http://openid.net/specs/openid-connect-core-1_0.html#Terminology)

- Ablauf (ähnlich SAML):

0. (Discovery)

1. The RP (Client) sends a request to the OpenID Provider (OP)

2. The OP authenticates the End-User and obtains authorization

3. The OP responds with an ID Token and usually an Access Token

4. The RP can send a request with the Access Token to the UserInfo Endpoint

5. The UserInfo Endpoint returns Claims about the End-User

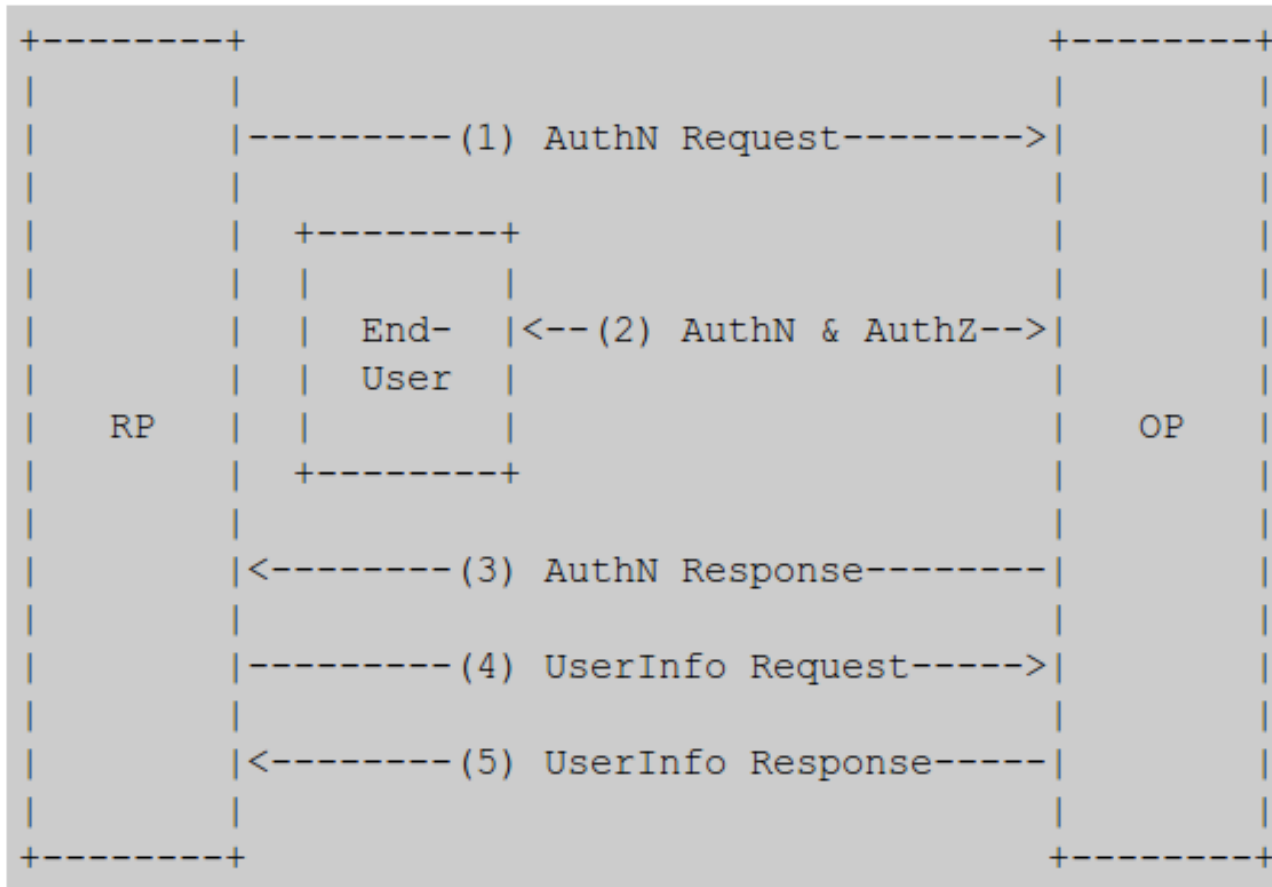
→ Siehe [http://openid.net/specs/openid-connect-core-1\\_0.html](http://openid.net/specs/openid-connect-core-1_0.html)

- **The RP (Client) sends a request to the OpenID Provider (OP)**
- **Drei vorgesehene Profile bzw. Flows:**
  - **Authorization Code Flow** (`response_type=code`)
  - **Implicit Flow** (`response_type=id_token token, response_type=id_token`)
  - **Hybrid Flow** – Kombination aus den beiden o.g. Flows, gesteuert über `response_type`
- **Abhängig vom ausgehandelten Flow erfolgt dann die weitere Kommunikation**

Flow property	Code	Implicit	Hybrid
Browser redirection step	✓	✓	✓
Backend request step	✓	×	✓
Tokens revealed to browser	×	✓	✓
Client can be authenticated	✓	×	✓

Quelle: <http://connect2id.com/learn/openid-connect>





## OP-seitige Endpunkte:

- authorization
- token
- userinfo

## optional:

- WebFinger
- Provider metadata
- Provider JWK set
- Client registration
- Session management

```
GET /rp?uid=babs%40op-test.aai.dfn.de HTTP/1.1
```

```
HTTP/1.1 302 Found
```

```
Location: https://op-test.aai.dfn.de:8092/authorization?
```

```
acr_values=PASSWORD&state=urn%3Auuid%3A67069088-eff1-4bd5-8032-f87e8f81bd70&redirect_uri=https%3A%2F%2Frp-test.aai.dfn.de%3A8666%2F919D3F697FDAAF138124B83E09ECB0B7&response_type=code&client_id=FFYUG1YPlSrE&scope=openid+profile+email+address+phone
```

```
GET /authorization?acr_values=PASSWORD&state=urn%3Auuid%3A67069088-eff1-4bd5-8032-f87e8f81bd70&redirect_uri=https%3A%2F%2Frp-test.aai.dfn.de%3A8666%2F919D3F697FDAAF138124B83E09ECB0B7&response_type=code&client_id=FFYUG1YPlSrE&scope=openid+profile+email+address+phone HTTP/1.1
```

```
HTTP/1.1 302 Found
```

```
Location: https://rp-
```

```
test.aai.dfn.de:8666/919D3F697FDAAF138124B83E09ECB0B7?
```

```
scope=openid+profile+email+address+phone&state=urn%3Auuid%3A67069088-eff1-4bd5-8032-f87e8f81bd70&code=2wma2h7BXRv1PBTmwDl sq2UDnJoR8TGJWH2dz7KnOo7y%2B%2F1DpdMoF9frcVu2OMbGG%2FfOPU%2BKbvIk17CiLj0RwDKh5769X7BFx8k9HjNMtsI%3D
```

```
{
  'access_token': '2wma2h7BXRv1PBTmwd1sq2UDnJoR8TGJWH2dz7KnOo7y+/1DpdMoF9
    frcVu2OMbGG/fOPU+KBvIk17CiLj0RwDKh5769X7BFx8k9HjNMtsI=',
  'id_token': 'eyJhbGciOiJSUzI1NiIsImtpZCI6Im9wMSJ9.eyJzdWIiOiAiaWNGMwNT
    cyM2Q0ODhjOWIzODk1ZjQzNzgxODZjYzRlMzc1NjFjZTZjNDAYZjhkYzZlMT
    BjOGJhYmZiZTFmMmQ2NSIsICJpc3MiOiAiaHR0cHM6Ly9vcC10ZXN0LmFhaS
    5kZm4uZGU6ODA5Mi8iLCAiYWNYIjogIlBBU1NXT1JJEIiwgImV4cCI6IDE0ND
    U3MDk2MTIsICJhdXRoX3RpbWUiOiAxAxNDQ1NjIzMjExLCAiaWF0IjogMTQ0NT
    YxOTYxMiwgImF1ZCI6IFsideEQ1RG5IZ01OVnJ0Il19.s0Zn8Bqa4i2oe1ToS
    DA2qOFiJWCSdvpei_r44SzgFRCEydo4a3K5yacUOeCCe_G_Nue2n2iLxtNtn
    S62ZzrE77o-Jsf8k4aa57iw65aleGXzpXM7MKxPrdTGNeK5AwOoKG-8cZeKL
    GbtTFBFsma5fsQXv4GvpF_9pW38pKStjWb2Rhs8FchKcNvmz4y0S1dLCmDIP
    VksM33G392fY1Z6anFsGSZ_S6skalvEIVgJtWESBQJN-F9JB7cTUUVDWU5NU
    Hcyhls0xKWEIzRS60yujyYoYQ5Dod70aNz9nF92YRtxvke6b5v0ibQSk6K3T
    Vvx9qcguRMdi6PxGwpPQ-6Q',
  'expires_in': 3600,
  'token_type': 'Bearer',
  'state': 'urn:uuid:67069088-eff1-4bd5-8032-f87e8f81bd70',
  'scope': 'openid profile email address phone',
  'refresh_token': 'XzofScR3I0ugNe3xOfhNsqNEZ5eAW208DQYH/zJSQsn6sxs8Aw
    Pxen6iQ7ltR6OGn+SMX3Dsa2kjSPHy97WqRox4OF0SeHEPfy7VIwvA4D0='
}
```

## id\_token

```
{  
  'sub': '4c05723d488c9b3895f4378186cc4e37561ce6c402f8[...]',  
  'iss': 'https://op-test.aai.dfn.de:8092/',  
  'acr': 'PASSWORD',  
  'exp': 1445709612,  
  'auth_time': 1445623211,  
  'iat': 1445619612,  
  'aud': ['FFYUG1YPlSrE']  
}
```

- **token\_type = 'Bearer':**  
Vielfältige Einsatzmöglichkeiten, z.B. Weiterreichen an andere (non-Web) Applikationen – man/frau beachte, dass 'aud' mehrere client\_ids beinhalten kann
- **'sub' = 'Subject',** eindeutiger User-Identifizierer. Zwei Varianten: 'public' = unveränderbar, Unique ID; 'pairwise' = Client-spezifisch, entspr. Targeted ID

```
POST /userinfo (mittels access_token)
HTTP/1.1
```

```
HTTP/1.1 200 OK
```

```
{ "family_name": "Jensen",
  "sub": "4c05723d488c9b389[...]",
  "email_verified": true,
  "given_name": "Barbara",
  "address": { "country": "USA",
               "region": "CA",
               "postal_code": "91608",
               "street_address": "100 Universal City Plaza",
               "locality": "Hollywood" },
  "nickname": "babs",
  "email": "babs@example.com",
  "name": "Barbara J Jensen" }
```

Wir erinnern uns an den Authentication Request:

```
GET /authorization?...&scope=openid+profile+email+address+phone
```

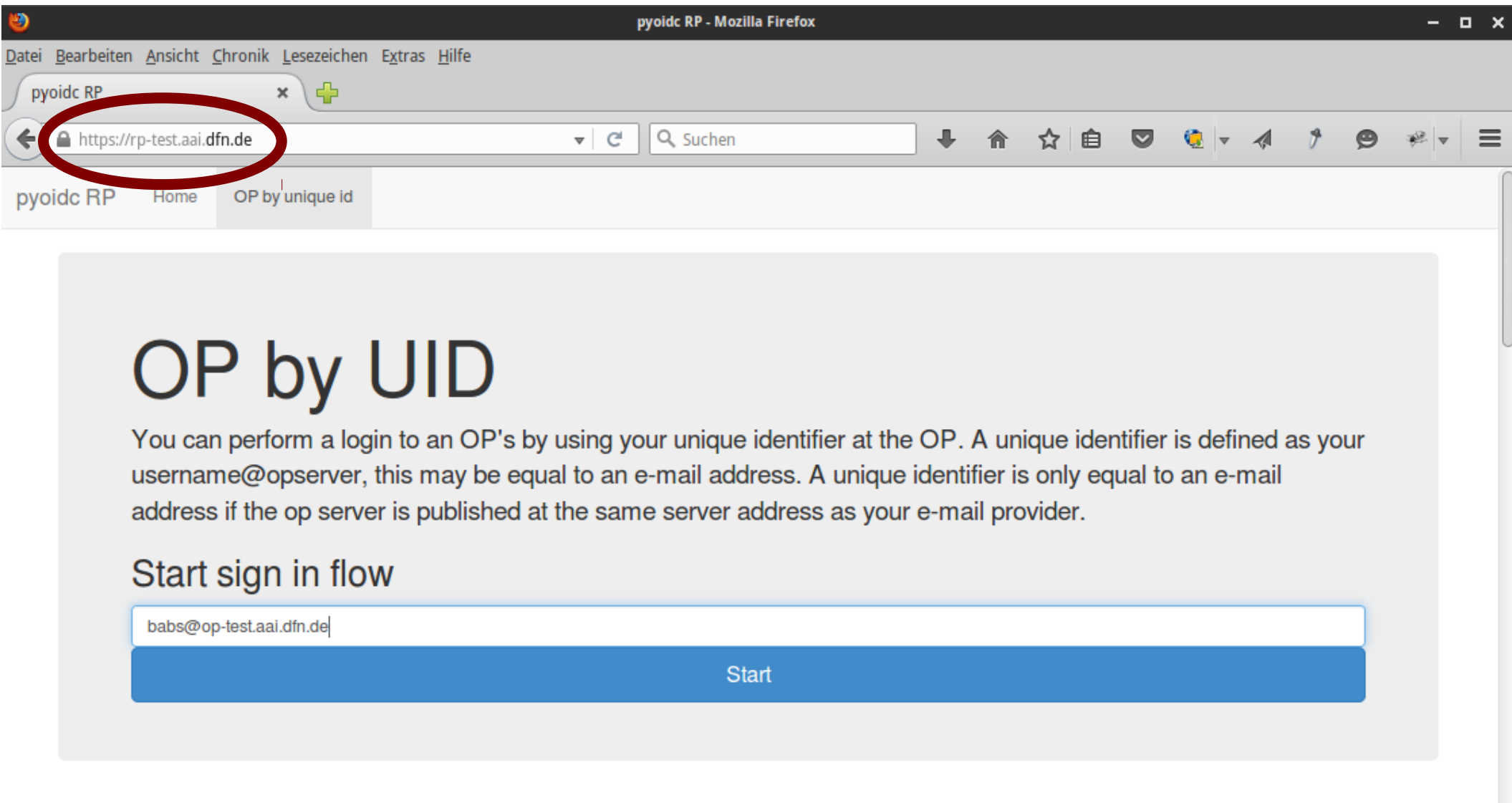
Scope value	Associated claims
email	email, email_verified
phone	phone_number, phone_number_verified
profile	name, family_name, given_name, middle_name, nickname, preferred_username, profile, picture, website, gender, birthdate, zoneinfo, locale, updated_at
address	address

Quelle: <http://connect2id.com/learn/openid-connect>

- Spezifikation aller angedachten Funktionen noch nicht abgeschlossen (z.B. Logout)
- Bislang auf Einzelanwendungen und abgeschlossene (Projekt- und ähnliche) Kontexte beschränkt, in denen Vertrauen über den Kontext (Verträge, gleiche Organisation) hergestellt wird
- **Einsatz im Föderationskontext?**
  - Discovery?
  - Wie wird Vertrauen hergestellt?
  - Zentral verwaltete und/oder signierte Metadaten?
  - Datenschutz? (Attributfreigabe, User ID in Webfinger, etc.)
  - Unterstützung bestimmter Attribut-Schemata, Custom Attributes?

- **The RP (Client) sends a request to the OpenID Provider (OP) ...**
- Wohin muss der Request gesendet werden?
- Bei 1:1 Verbindungen fällt die Wahl (hart codiert) nicht schwer ...
- Andernfalls: OpenID Connect Discovery  
[http://openid.net/specs/openid-connect-discovery-1\\_0.html](http://openid.net/specs/openid-connect-discovery-1_0.html)
  - OIDC verwendet WebFinger [RFC7033]  
<http://tools.ietf.org/html/rfc7033>
  - Außerdem: OpenID Connect Dynamic Client Registration (RP-Metadaten → OP)  
[http://openid.net/specs/openid-connect-registration-1\\_0.html](http://openid.net/specs/openid-connect-registration-1_0.html)





The screenshot shows a Mozilla Firefox browser window with the title "pyoidc RP - Mozilla Firefox". The address bar contains the URL "https://rp-test.aai.dfn.de", which is circled in red. The browser's menu bar includes "Datei", "Bearbeiten", "Ansicht", "Chronik", "Lesezeichen", "Extras", and "Hilfe". The browser's toolbar includes a search bar with the text "Suchen" and various navigation icons. The page content features a large heading "OP by UID" and a paragraph explaining the login process: "You can perform a login to an OP's by using your unique identifier at the OP. A unique identifier is defined as your username@opserver, this may be equal to an e-mail address. A unique identifier is only equal to an e-mail address if the op server is published at the same server address as your e-mail provider." Below this text is a section titled "Start sign in flow" with a text input field containing "babs@op-test.aai.dfn.de" and a blue "Start" button.

Client-/RP-seitige Eingabe einer Nutzerkennung (ähnlich kritisch wie eduPersonPrincipalName?)

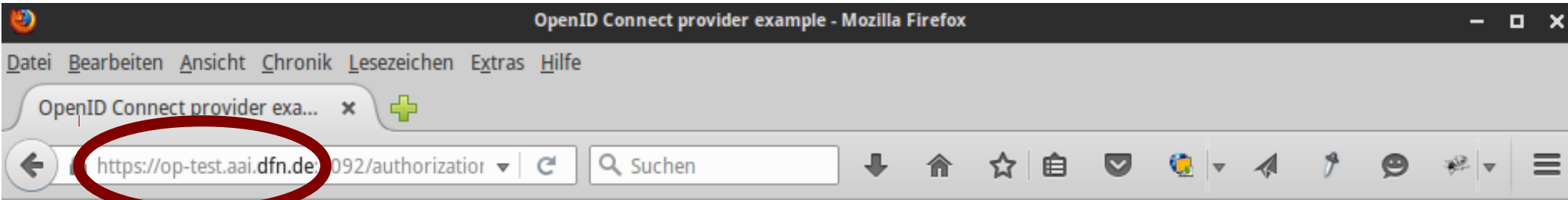
Alternativ ist auch die Eingabe eines URLs vorgesehen, hier: <https://op-test.aai.dfn.de/babs>

# OP-Discovery via WebFinger (2)

```
GET /.well-known/webfinger?resource=acct%3Ababs%40op-test.aai.dfn.de
    &rel=http%3A%2F%2Fopenid.net%2Fspecs%2Fconnect%2F1.0%2Fissuer
HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/jrd+json
```

```
{
  "subject": "acct:babs@op-test.aai.dfn.de",
  "links":
  [
    {
      "rel": "http://openid.net/specs/connect/1.0/issuer",
      "href": "https://op-test.aai.dfn.de"
    }
  ]
}
```



## User log in

Username

Password

© Copyright 2014 Umeå Universitet

Die weitere, Token-basierte Kommunikation erfolgt dann wie eingangs beschrieben ...

## OpenID Provider Configuration Request

```
GET /.well-known/openid-configuration HTTP/1.1
```

```
Host: example.com
```

([http://openid.net/specs/openid-connect-discovery-1\\_0.html#ProviderMetadata](http://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata))

## OpenID Provider Configuration Response

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{  
  "issuer":  
    "https://server.example.com",  
  "authorization_endpoint":  
    "https://server.example.com/connect/authorize",  
  "token_endpoint":  
    "https://server.example.com/connect/token",  
  "token_endpoint_auth_methods_supported":  
    ["client_secret_basic", "private_key_jwt"],  
  ... (usw.) ...  
}
```

## ”OpenID Connect Dynamic Client Registration”

### *Client Registration Request an den Client Registration Endpoint des OP*

```
POST /connect/register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: server.example.com
Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJ ...
{
  "application_type": "web",
  "redirect_uris":
    ["https://client.example.org/callback",
     "https://client.example.org/callback2"],
  "client_name": "My Example",
  "client_name#ja-Jpan-JP":
    "クライアント名",
  "logo_uri": "https://client.example.org/logo.png",
  "subject_type": "pairwise",
  "sector_identifier_uri":
    "https://other.example.net/file_of_redirect_uris.json",
  "token_endpoint_auth_method": "client_secret_basic",
  "jwks_uri": "https://client.example.org/my_public_keys.jwks",
  "userinfo_encrypted_response_alg": "RSA1_5",
  "userinfo_encrypted_response_enc": "A128CBC-HS256",
  "contacts": ["ve7jtb@example.org", "mary@example.org"],
  "request_uris":
    ["https://client.example.org/rf.txt
     #qpXaRLh_n93TTR9F252ValdatUQvQiJi5BDub2BeznA"]
}
```

- Metadaten werden nach dem eben beschriebenen Modell dynamisch ausgetauscht
- Metadaten sind *self-asserted* und nicht signiert
- Keine *Trusted Third Party* involviert, die für ein Vertrauensverhältnis sorgt
- Ergänzende Spezifikation:  
**OpenID Connect Federation (Draft)**
- Neueste Version unter [https://github.com/rohe/pyoidc/blob/master/oidc\\_fed/oidcfed.txt](https://github.com/rohe/pyoidc/blob/master/oidc_fed/oidcfed.txt)
- Hierarchisches Modell, das mit sogenannten *Metadata Statements* arbeitet, bei denen mindestens die oberste Ebene vom Föderationsbetreiber signiert werden muss.
- Verwendung von **JSON Web Keys (JWK)**

- *“Metadata statements and [public] signing keys can be transferred in two different ways: either by including the information in the statement, or by providing a URI that points to the information.”*
- Metadata Statements können also bei Bedarf zentral gepflegt werden
- Metadata Statement *“MUST be a signed JWT”*
- Hierarchische Struktur
  - Provider-unabhängige Informationen: Signing Keys, Kontakte, (UI-)Infos zum RP- und/oder OP-Betreiber
    - RP- und OP-spezifische Angaben

- Signing Requests werden seitens der von den jeweiligen Föderations-Teilnehmern bestimmten Repräsentanten eingereicht (wie gehabt)
- Neu: Hierarchisches Modell, bei dem der Requester der jeweils höheren Ebene für die Signierung zuständig ist
- Signaturen müssen ein Verfallsdatum haben
- Die Signaturkette und die Gültigkeit der Signaturen muss seitens OP und RP validiert werden
- **NB:** Für JWK können auch X.509 Zertifikate verwendet werden, das Format ist lediglich ein anderes.



- **WebFinger und Datenschutz**  
Anonymer Identifier, der ausschließlich zur Identifizierung des OP verwendet wird?
- **Attribut-Schemata, Mapping auf OIDC Claims**  
REFEDS Working Group
- **DFN-AAI**
  - Proof of Concept Implementierung der *OpenID Connect Federation* Spezifikation für die DFN-AAI
  - Testbed für Shibboleth IdP OIDC-Implementierung
  - Überlegungen zum Einsatz von Bridging Elementen  
SAML2 ↔ OIDC innerhalb der DFN-AAI

- Allgemeiner Überblick  
<http://openid.net/connect/>
- Verfügbare Implementierungen  
<http://openid.net/developers/libraries/>
- Shibboleth IdP  
<https://github.com/uchicago/shibboleth-oidc>
- Diverse Präsentationen zum Thema  
[https://github.com/rohe/ojou\\_course/tree/master/presentation](https://github.com/rohe/ojou_course/tree/master/presentation)
- OpenID Connect Federation (Draft)  
[http://openid.net/specs/openid-connect-federation-1\\_0.html](http://openid.net/specs/openid-connect-federation-1_0.html)
- JSON Web Key (JWK)  
<https://tools.ietf.org/html/rfc7517>
- JSON Web Token (JWT)  
<https://tools.ietf.org/html/rfc7519>
- OIDCref REFEDS Working Group  
<https://wiki.refeds.org/display/GROUPS/OIDCref>

# Vielen Dank für Ihre Aufmerksamkeit!

## Fragen? Anmerkungen?

### Kontakt

www: <https://www.aai.dfn.de>

eMail: [aai@dfn.de](mailto:aai@dfn.de)

Tel.: +49 30 884299 9124