

GEANT 4-2 JRA3 T3 “TrustTech”: OpenID Connect Identity Federations

Henri Mikkonen

CSC – IT Center for Science Ltd.

Internet2 TechEx 2017 San Francisco

16. October 2017

- Introduction to the OpenID Connect Federation –draft
 - “OIDCfed” – by Roland Hedberg et al
 - (Roland will attend TechEx on Wednesday and Thursday)
- OIDC OP role implementation for Shibboleth IdP
- Next Steps / Schedule

- Relying Parties (RP) first need to discover OpenID Providers (OP)
 - *OpenID Connect Discovery* –spec
 - OPs publish **self-asserted** information about themselves
- RPs can register to OPs in two ways
 - *Static*: first developer registers to the provider, then he can register his RP (For instance social media providers)
 - *Dynamic*: from the OP's perspective, the RPs are anonymous, as the information is **self-asserted**
 - *OpenID Connect Dynamic Client Registration* –spec
- The OIDCfed spec only deals with **OP discovery and RP registration**

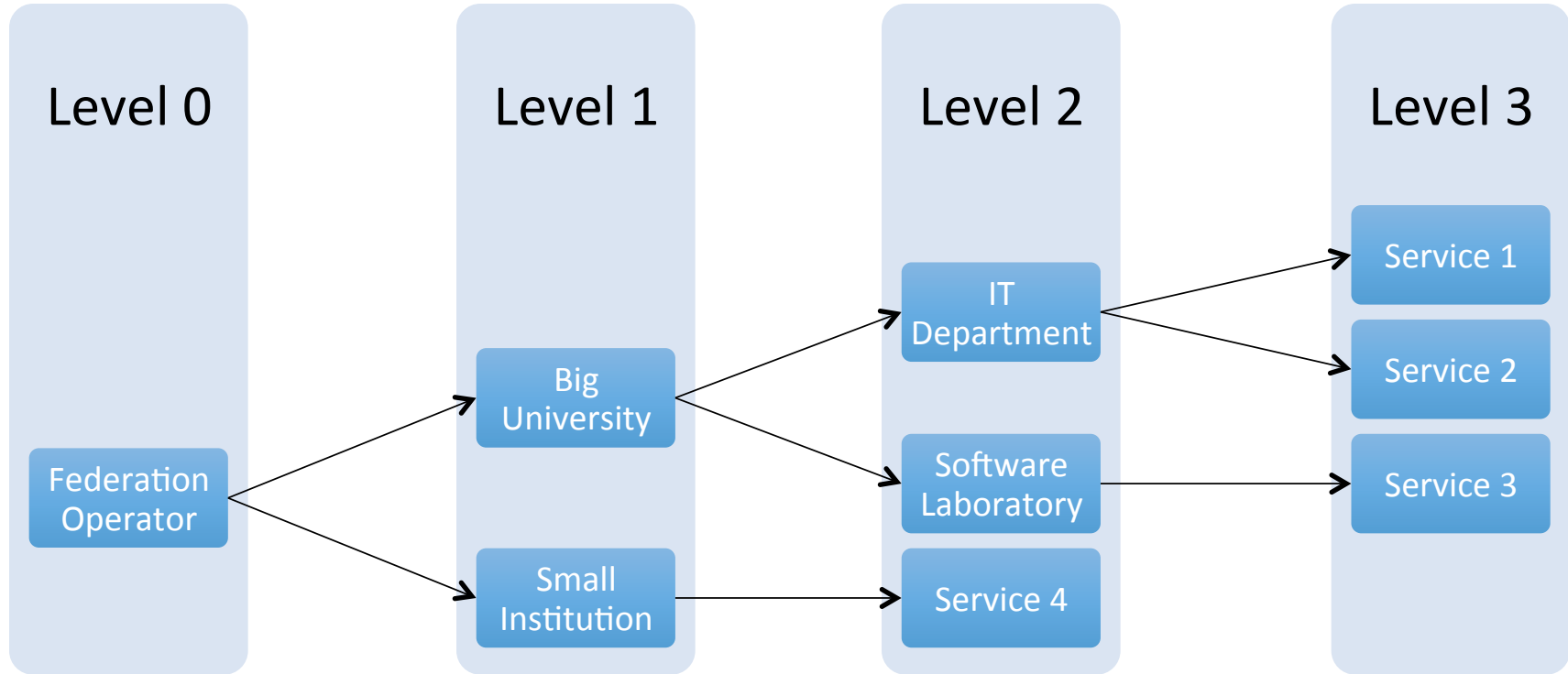
OIDCfed – Design principles

- Allow dynamic discovery and registration without losing trust
 - Mutual identification of OPs and RPs during the process
- Enforcement of federation and organization policies
 - Every entity and organization needs to behave in a certain way
- Allow delegation of entity registration
 - Not everything needs to go through operator: trusted organizations should be able to register their own entities by themselves
- Metadata transport and origin independent
- Self-contained metadata

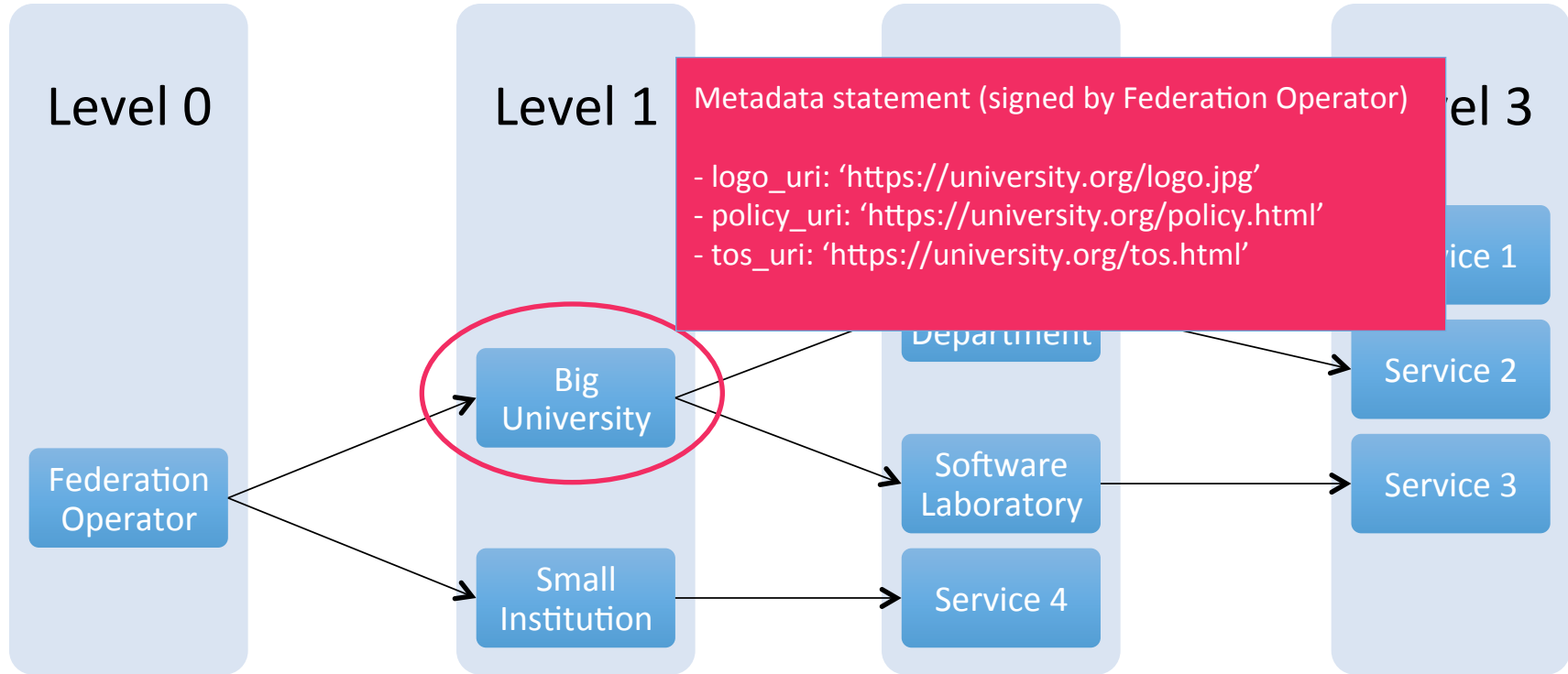
OIDCfed – Building blocks

- Trusted 3rd party
 - I.e. Federation operator (NREN, any small/big community)
- Chain of verifiable claims (metadata statements)
 - Federation operator provides and signs the root of the chain
 - Verification is done via JWT signatures
- Compounded metadata
 - Entity's metadata consists of the whole chain until the metadata statement signed by the federation operator
 - All the public keys except FO's exists in the chain

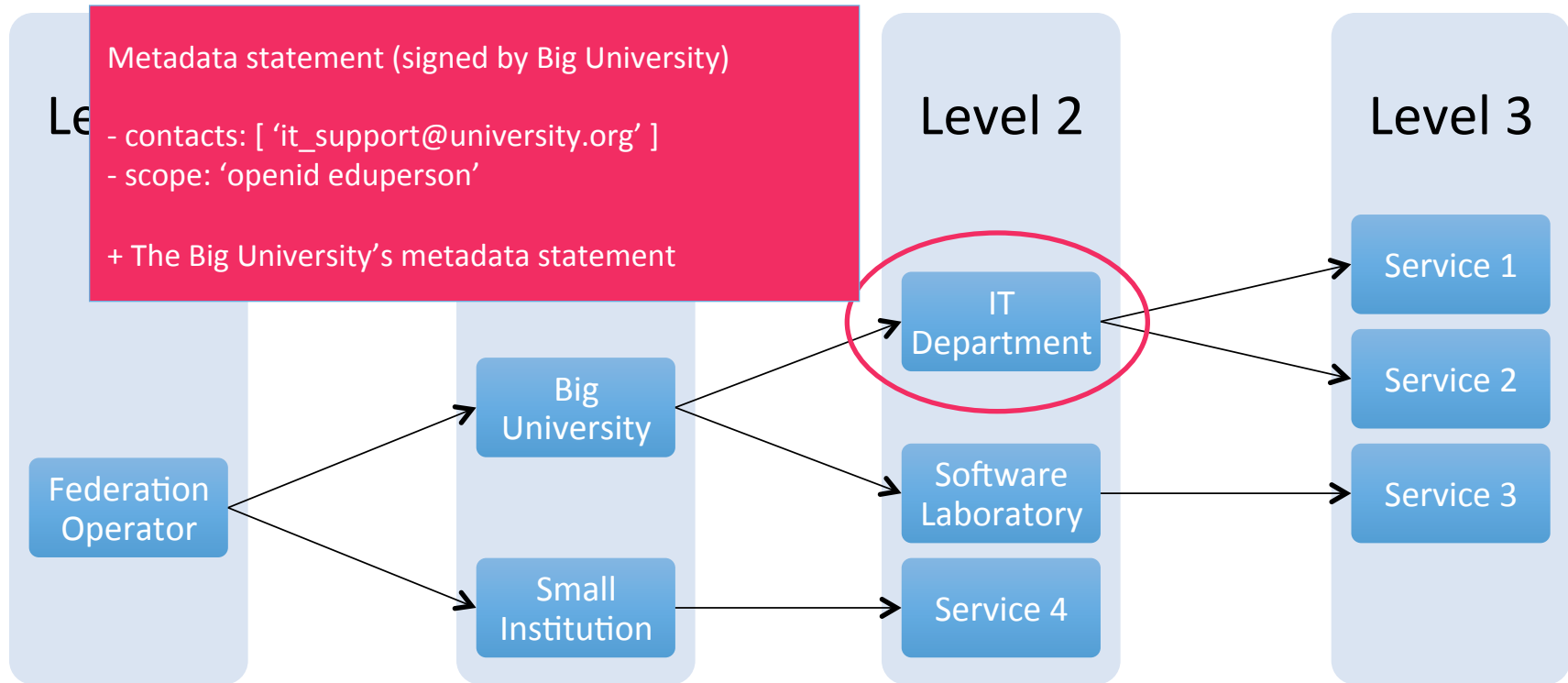
OIDCfed – Structure



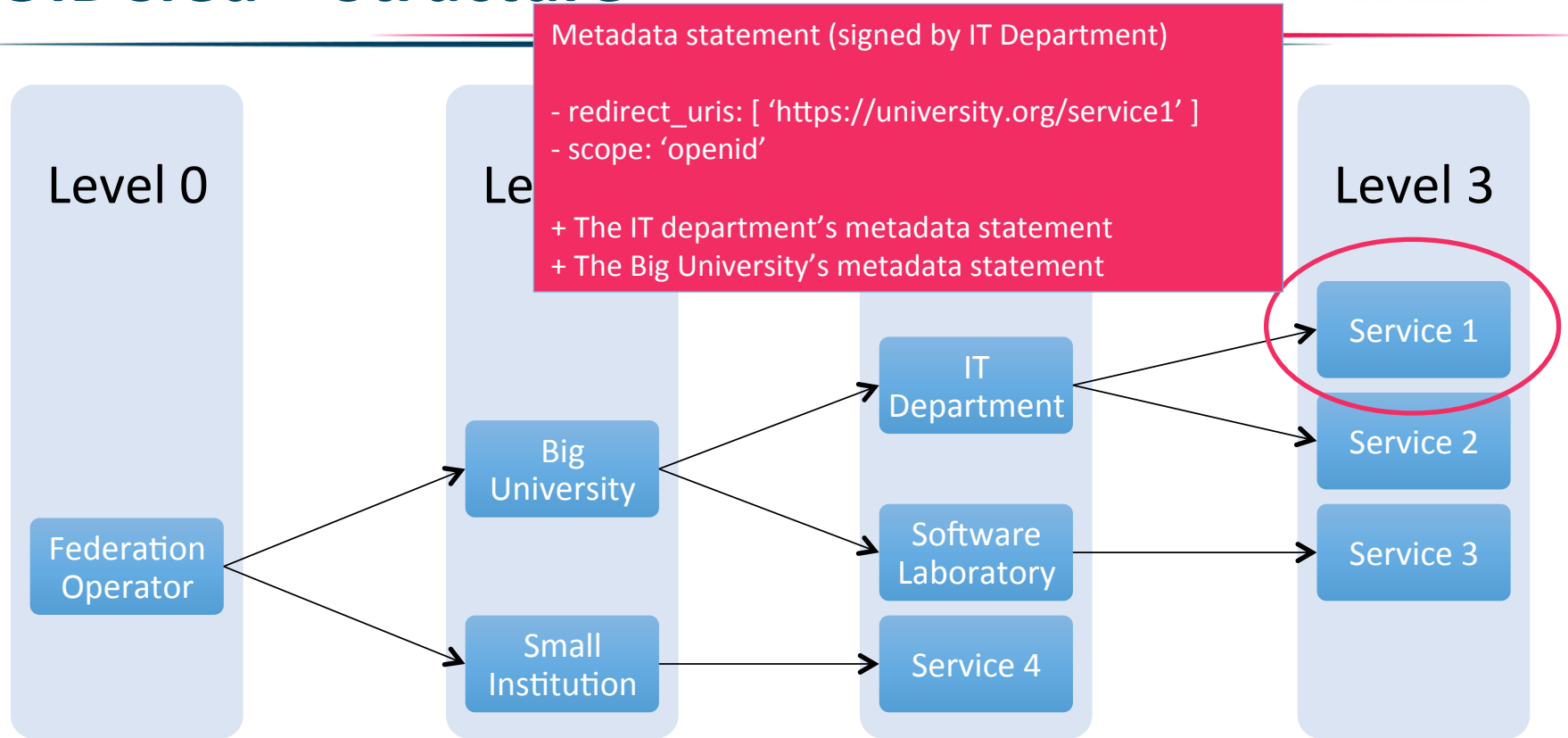
OIDCfed – Structure



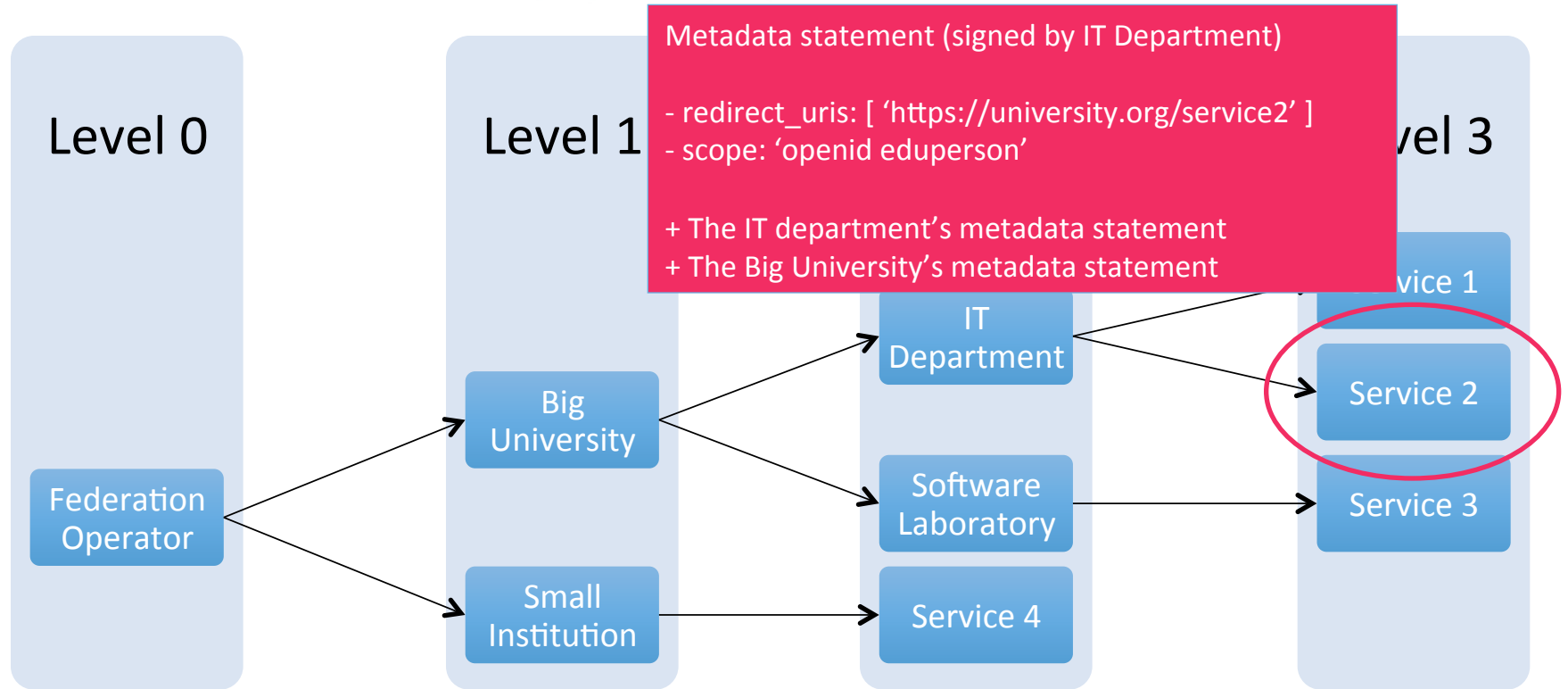
OIDCfed – Structure



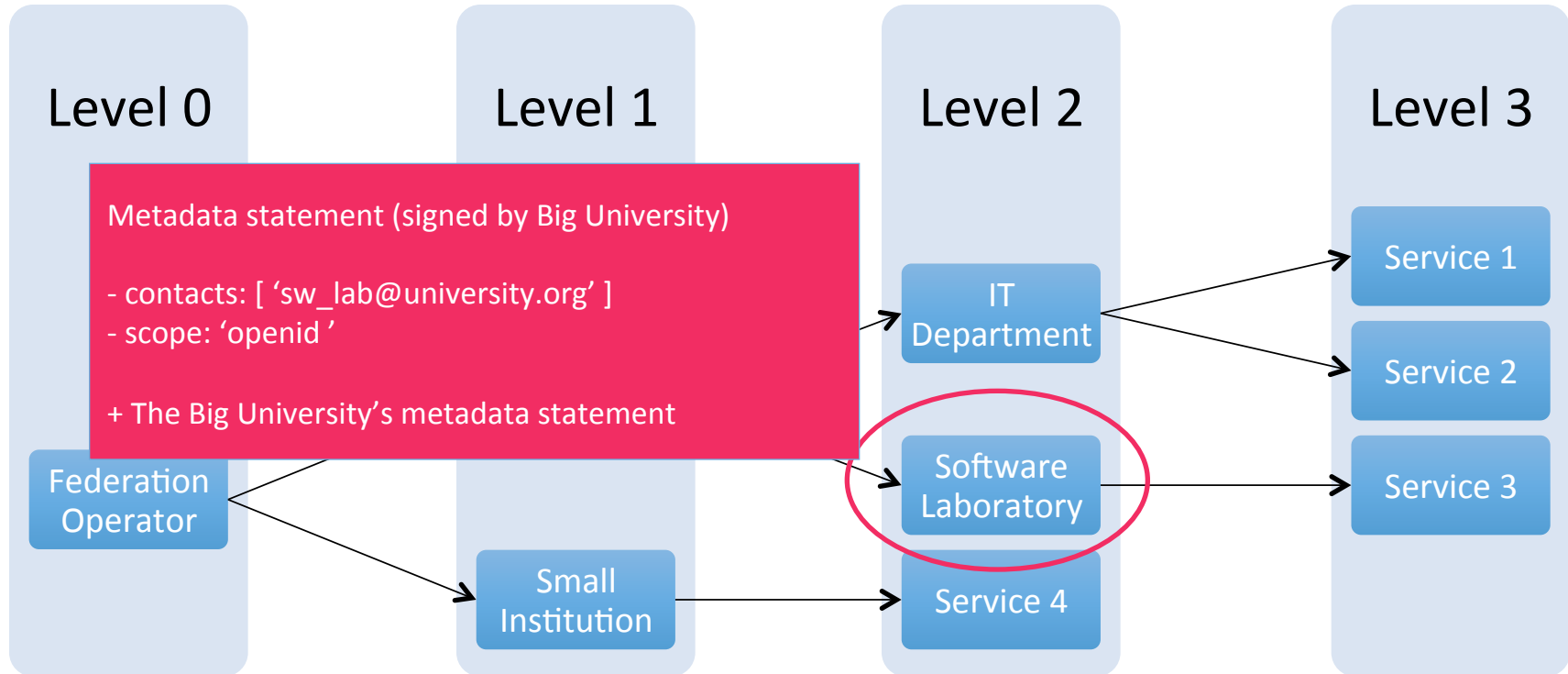
OIDCfed – Structure



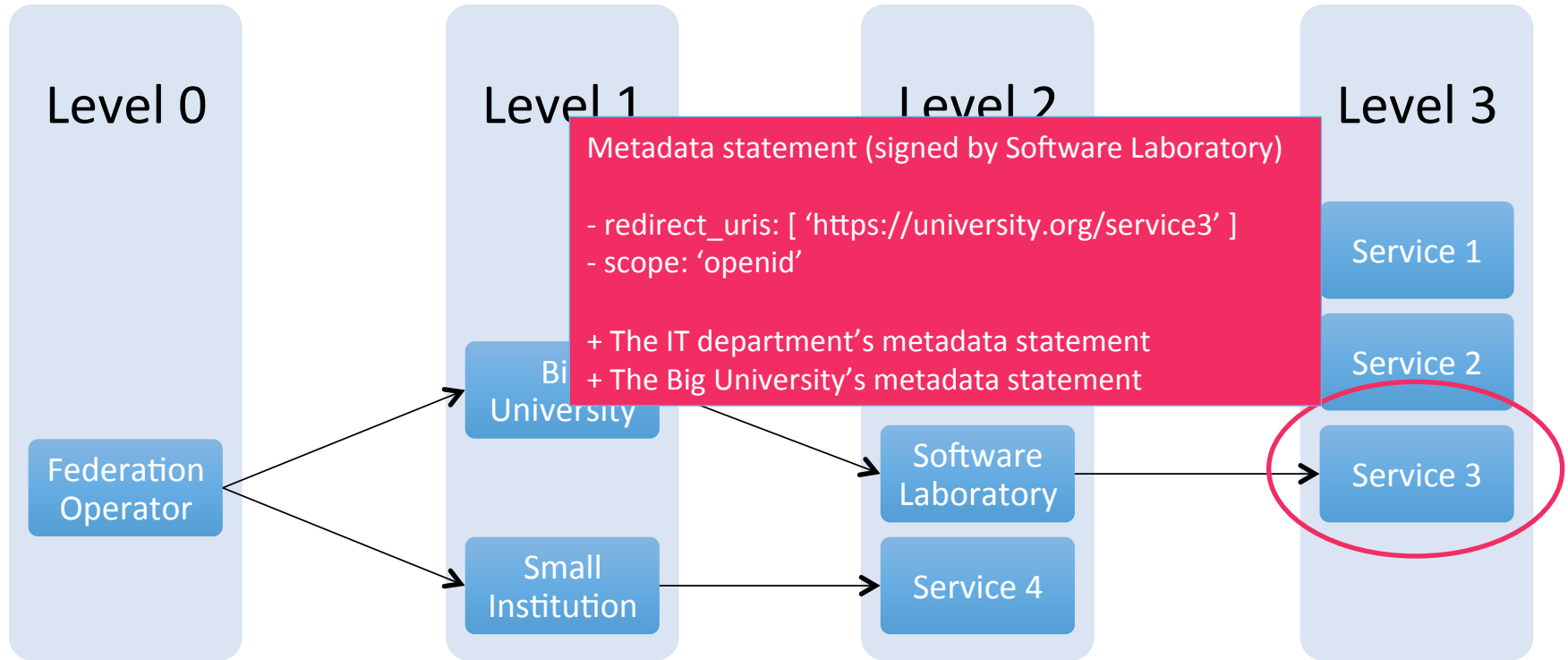
OIDCfed – Structure



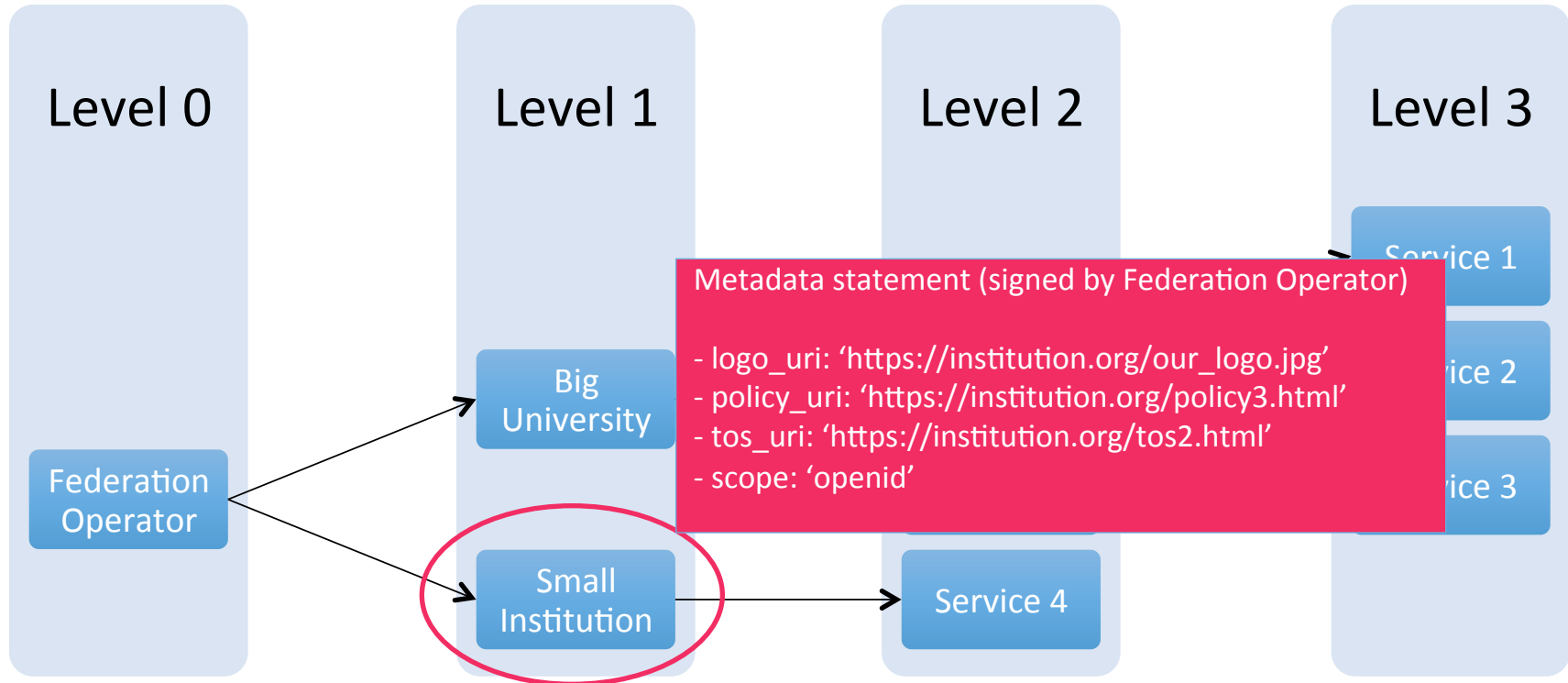
OIDCfed – Structure



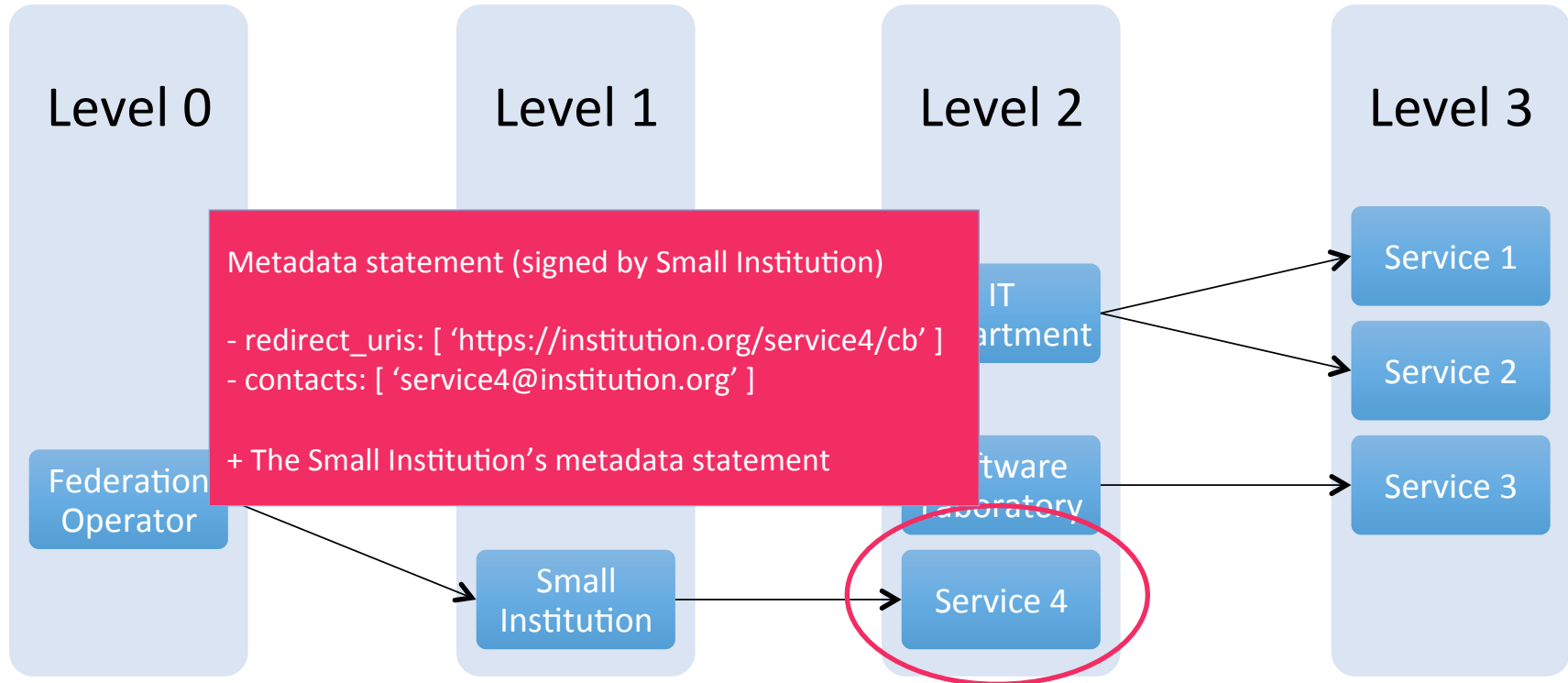
OIDCfed – Structure



OIDCfed – Structure



OIDCfed – Structure



OIDCfed – Big changes to existing feds

- New entity registration is outsourced to the members
 - Federation Operator doesn't necessarily know the exact amount of entities in the federation
- Existing business models may not work
 - For instance, if registration costs money
- However, both styles (and protocols) need to be supported for (quite) some time

OIDC(fed) and Shibboleth IdP?

- Shibboleth IdP is widely used by federations worldwide
 - In some countries nearly all IdPs use Shibboleth (Finland, Switzerland, ..)
- Organizations are not keen to run multiple services for “one” purpose
 - The existing deployments are quite complex
 - Not just IdM/LDAP integration: 2FA, clustering, etc
 - Lots of overhead to configure same logic for two pieces of software
 - Admins are familiar with the Shibboleth configuration logic
- Shibboleth IdP v3 has very modular architecture
 - CAS is natively supported – why not OIDC, too?

OIDC + Shibboleth IdP implementation

- GÉANT task implements OIDC support as plugins
 - Already started with the standard OIDC profiles/flows/features
 - Implicit flow first (many similarities with saml2int) + dynamic registration
 - Continue with authorization code & hybrid flows + OIDCfed
- Aim at being as orthodox IdP plugin as possible
 - Exploit the protocol-independent features
 - Authn (incl. MFA), Attribute resolution, Session mgmt, RP config, Consents, etc
- Collaborate actively with the Shibboleth team
 - Some VCs already taken place and has been very valuable

Not just Shibboleth

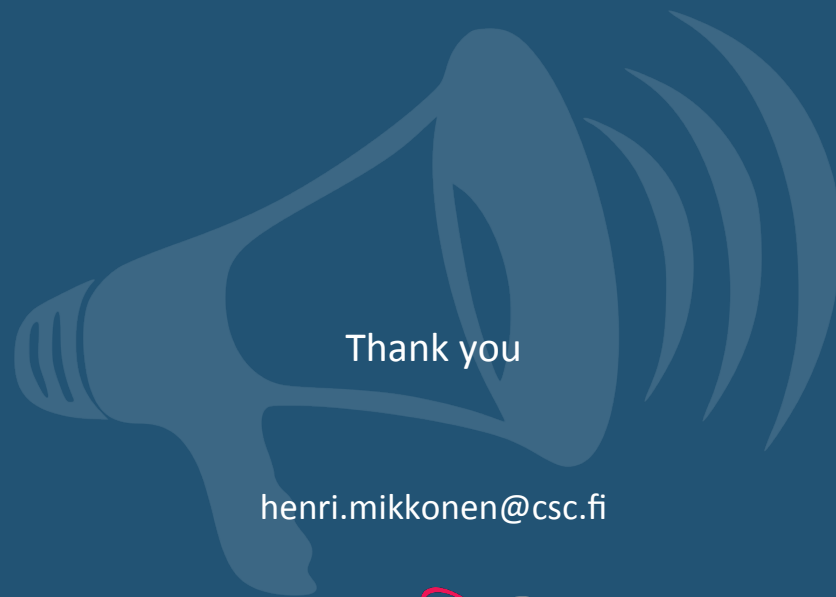
- OIDCfed reference Python implementation exists
 - OP + RP
- GÉANT task implements OIDCfed-aware RP libraries
 - PHP
 - Android (using AppAuth)
 - iOS (using AppAuth)
- Test tool for compliance of the fed-aware RP libraries and OPs

Next Steps

- Q4 2017
 - Call for pilots
 - Standalone fed-aware Python implementation
 - Finish and document RP libraries
- Q1 2018
 - Limited fed-aware Shibboleth IdP –based OP
- Q1-Q3 2018
 - Pilots with interested parties
- Q4 2018
 - Finalization of the draft and road to standardization

Further reading

- The OIDCfed specification
 - https://openid.net/specs/openid-connect-federation-1_0.html
- Reference OIDCfed implementation (Python)
 - <https://github.com/OpenIDC/fedoidc>
- Shibboleth IdP + OIDC work (code + wiki)
 - <https://github.com/CSCfi/shibboleth-idp-oidc-extension>
- The GEANT 4-2 JRA3T3 wiki
 - <https://wiki.geant.org/display/gn42jra3/T3.1A+OpenID+Connect+Federation>



Thank you

henri.mikkonen@csc.fi



Networks · Services · People
www.geant.org



This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 691567 (GN4-2).