



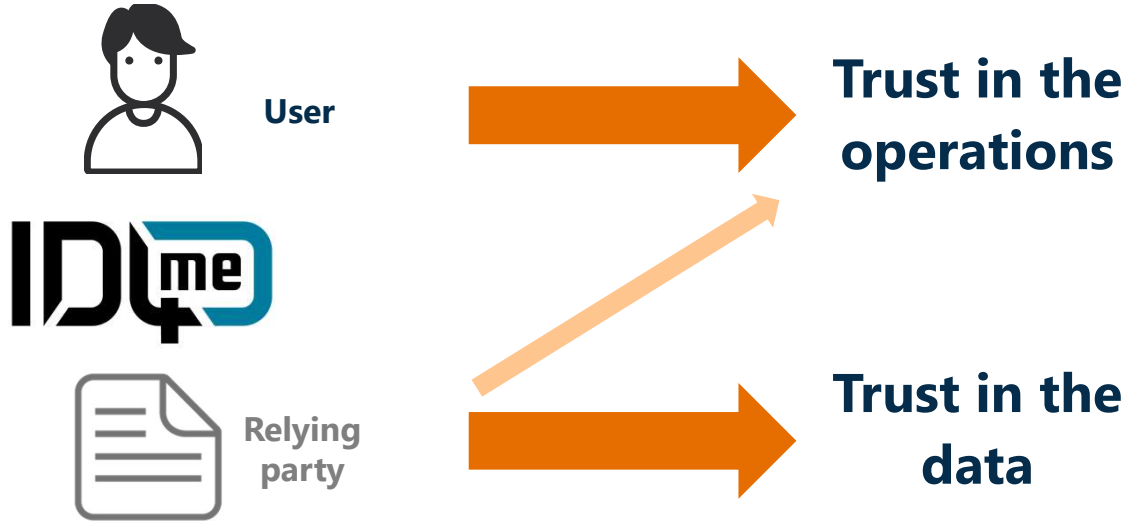
# ***THE TRUST MODEL FOR ID4me DATA***

ID4me Summit 2019, Madrid

Vittorio Bertola , Head of Policy & Innovation, Open-Xchange

10/18/2019

# Two different scopes of trust models



# Two trust models for identity data

## **Weak identity**

- The user enters whatever they want
- Data are not validated
- The relying parties trust what the user provides
- If they don't, verification is on themselves

## **Strong («verified») identity**

- The user enters data that must be real
- Some kind of authority validates the data
- The relying parties trust the authority, not the user
- The relying parties do not have to do verification

# Trust in the data = Level of assurance

Example levels of assurance, building on eIDAS LoAs:

Level of assurance	Features
None	No verification of claims at registration Password authentication
Low	Remote third-party proof of identity at registration Token with one-time password for authentication
Substantial	Verified remote third-party proof of identity at registration 2-factor authentication (e.g. PIN + smartphone app with token)
High	In-person validation + verified public proof of identity at registration 2-factor authentication with key on encrypted personal device

*Note: the above examples have just been made up for illustration – the actual levels and features are yet to be discussed among ID4me participants.*



**Verified identities**

# Trust frameworks

- Levels of assurance are defined within a **trust framework**
- The ID4me platform can support any trust framework (it is just one more parameter in the identity)
- We could create an ID4me trust framework, however that would require additional effort and do not provide interoperability
- To facilitate interoperability, **we recommend everyone to use the eIDAS definitions**
- However:
  - ID4me operators are not required to be valid eIDAS providers in their country
  - ID4me verified identities are not required to be valid eIDAS identities

# Strong identity model, the simple version



**Identity  
authority**

Validates credentials  
Validates  
identity agents



**Identity  
agent**

Provides the service to  
the user



**Data authority**

Validates claims  
(and asserts so)

# Strong identity model, the even simpler version



**Identity  
authority**

Validates credentials  
Validates  
identity agents



**Identity agent +  
Data authority**

Provides the service to  
the user and validates  
the claims

# Several advantages of the simpler model

- Does not require additional parties, additional protocols and additional system costs
- Does not require a distributed cryptographic infrastructure
  - Identity tokens are already signed by the agent, and the agent is the entity responsible for its content, so the relying party can just take whatever is in the token as valid
  - The agent can rely on third party data authorities to validate the identities, but this is completely transparent to the rest of the system
- The ID4me system already has some degree of validation of the agents
  - Agents have to sign a contract with authorities, so they are known and vetted by them
  - Relying parties can choose to trust a small list of authorities instead of a big list of agents
- It is fully distributed and technically and operationally scalable



# There is just a small problem

Especially when you are a control freak and many unknown new authorities start to appear



”

**I am a relying party and I really want to be sure that the claims in the identity are real.  
How do I know that I can trust the agent / data authority / identity authority?**

# Strong identity model, the complex version



**Identity  
authority**

Validates credentials  
Validates  
identity agents



**Identity  
agent**

Provides the service to  
the user



**Data authority**

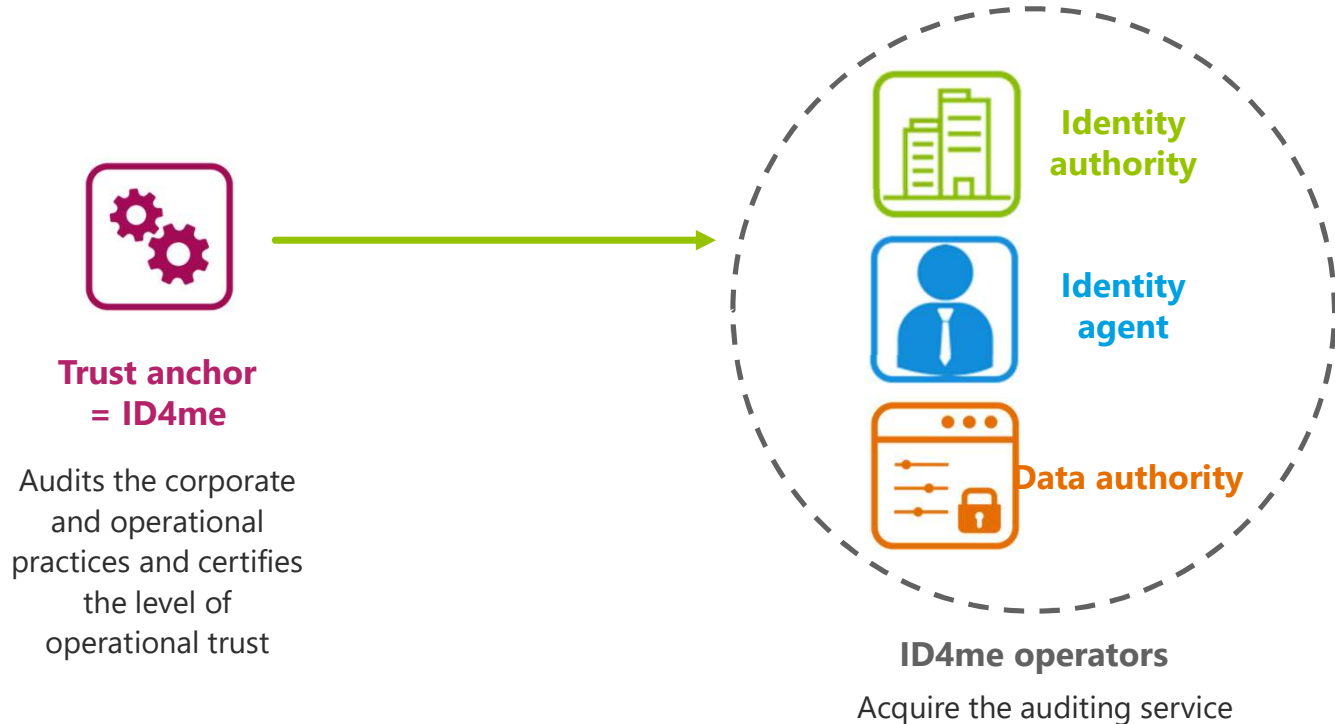
Validates claims



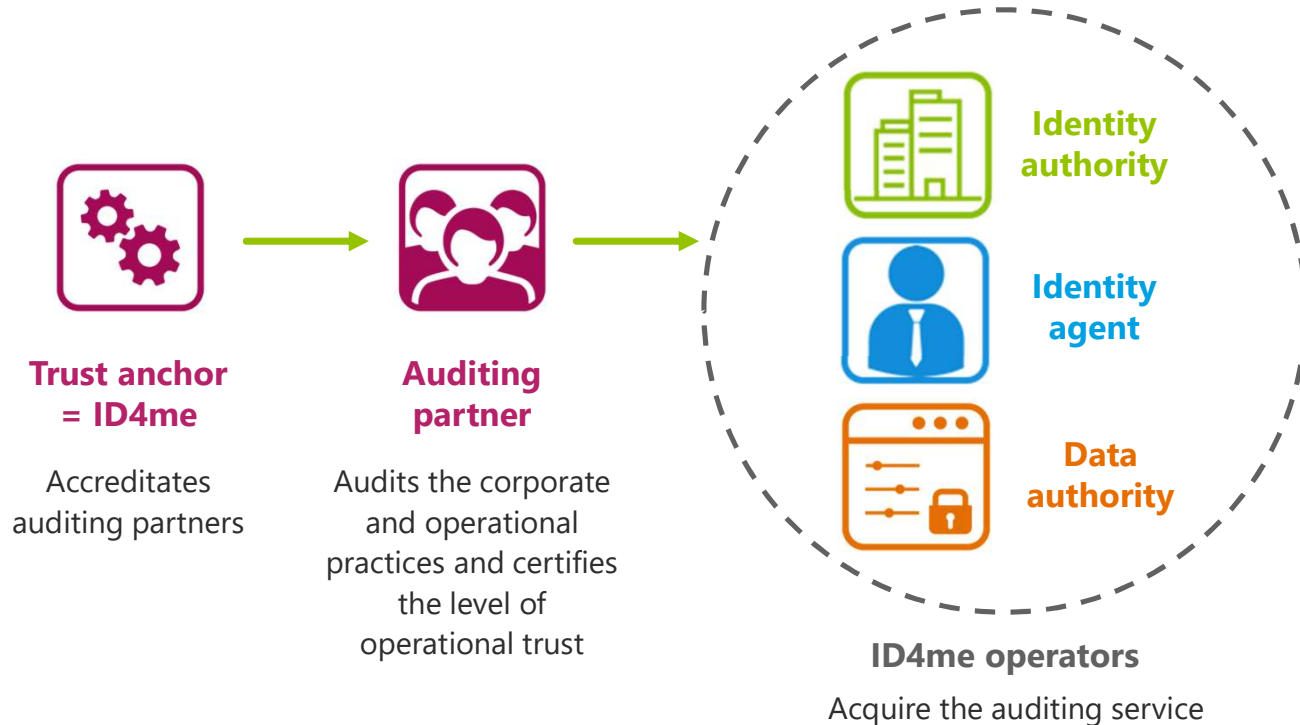
**Trust anchor  
/ Auditing  
partner**

Validates  
data authorities /  
identity agents /  
identity authorities

# Direct verification of operator trustability



# Delegated verification of operator trustability



# Trust in the operations = Level of operational trust

Example levels of operational trust:

Level of operational trust	Features
Unverified	No verification at all
Low	An auditing partner has verified the name and place of business of the operator through documents supplied by them electronically
Substantial	An auditing partner has verified the name and place of business of the operator independently from it The operator has subscribed to a standard code of conduct related to the security and availability of its services
High	Like the previous one, plus the operator's procedures have been audited by the certification authority

*Note: the above examples have just been made up for illustration – the actual levels and features are yet to be discussed among ID4me participants.*

# Evaluation of the complex model

## Advantages

- It supports any number of authorities, including small and independent ones
- It is resilient against bad authorities
- It is fully flexible and supports many different use cases

## Disadvantages

- Potentially hard and expensive to scale
- Complex to understand (significant number of players with different roles)
- Requires some direct liability by centralized trust anchors (e.g. ID4me aisbl)
- Requires additional technical work (protocols between various roles, key distribution infrastructure, etc.)
- It is overkill for simple use cases



# Login with ID4me

Examples of use cases  
that can be supported  
in the complex (4-role) model

monodirectional trust



# Use case 0: No guaranteed trust



There is no trust guaranteed by the system. The relying party receives information from the user and cannot know if it is «true».

## Use cases:

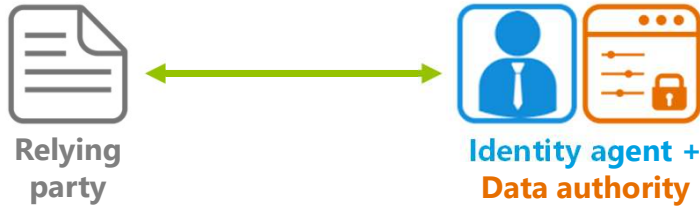
- Weak identity for low value logins

The original ID4me concept.

Very easy and immediate, and relying parties can still do direct out-of-band verification if necessary. Not suitable if the relying party needs trusted data.



# Use case 1: Closed identity system, centralized



## Use cases:

- Corporate identity systems, with identities managed internally
- Closed silo over ID4me

Only works on a very local scale; relying parties can just hardcode the only agent they accept. The agent is likely to also be the identity authority.

There is a single agent + data authority in the system. There is no certification of actors, but the agent could reject relying parties. The relying party already knows the agent + data authority and trusts it.

## Use case 2: Closed identity system, overlaid



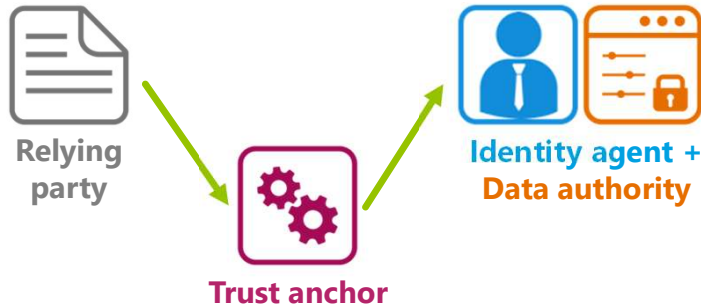
### Use cases:

- Corporate identity systems, federated
- Local public identity systems, where the local public administration runs data verification directly

Any ID4me identity can be used within these systems; the trust is ensured by a signed assertion.

There is a single data authority in the system. There is no certification of actors and the trust is one-way: the relying party already knows the data authority and trusts it, but the data authority accepts any agent and relying party.

## Use case 3: Local trusted identity, federated



### Use cases:

- Local/national public identity systems, where the local public administration accredits a number of local data authorities

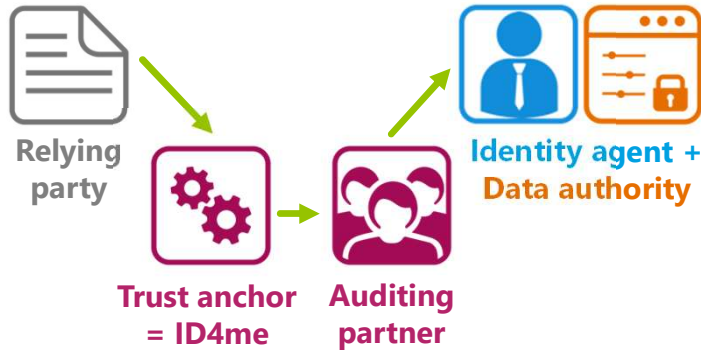
Only works on a relatively local scale, where the trust anchor can validate all agents and relying parties trust it by design. Needs assertions by the trust anchor.

There are multiple agents + data authorities in the system – they verify the claims.

There is one trust anchor that accredits all agents + data authorities.

The relying party already knows the trust anchor and trusts it.

# Use case 4: Global trusted identity, federated



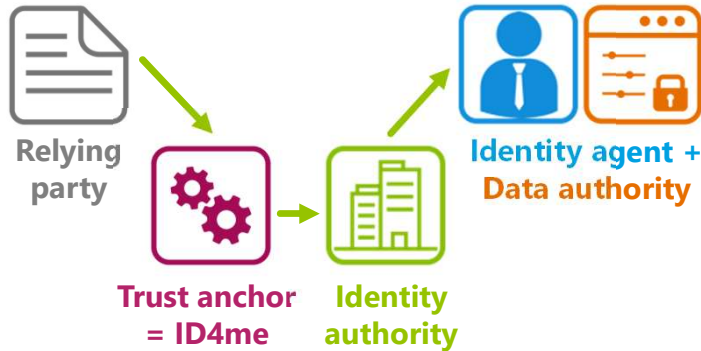
## Use cases:

- The “global ID card” for the Internet

It is theoretically possible, but complex and expensive to scale and keep secure, especially if the level of assertion over the claims is high. Still needs assertions that relying parties can verify independently.

Like the previous case, but with ID4me as the centralized trust anchor that, through auditing partners, accredits each and every «trusted» agent in the system – so it would apply by default to any ID4me identity. Similar to the CA system.

## Use case 5: Global trusted identity, federated<sup>2</sup>



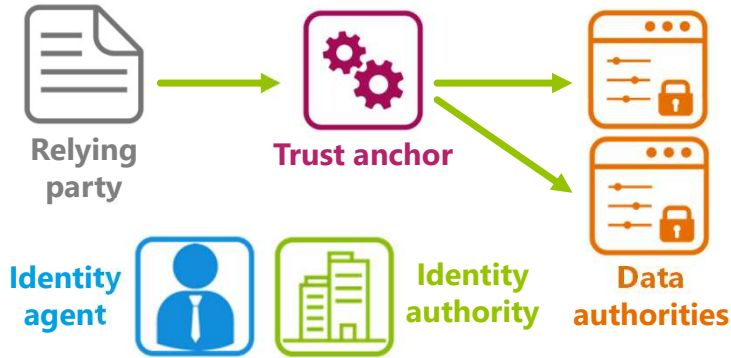
### Use cases:

- The “global ID card” for the Internet

This is more scalable, but still pretty hard to keep secure in practice. Needs nested assertions that relying parties can verify independently.

Like the previous case, but there is a chain of trust in which ID4me accredits authorities and each authority accredits the agents. Similar to the CA system with intermediate certificates.

## Use case 6: Global trusted identity, overlaid



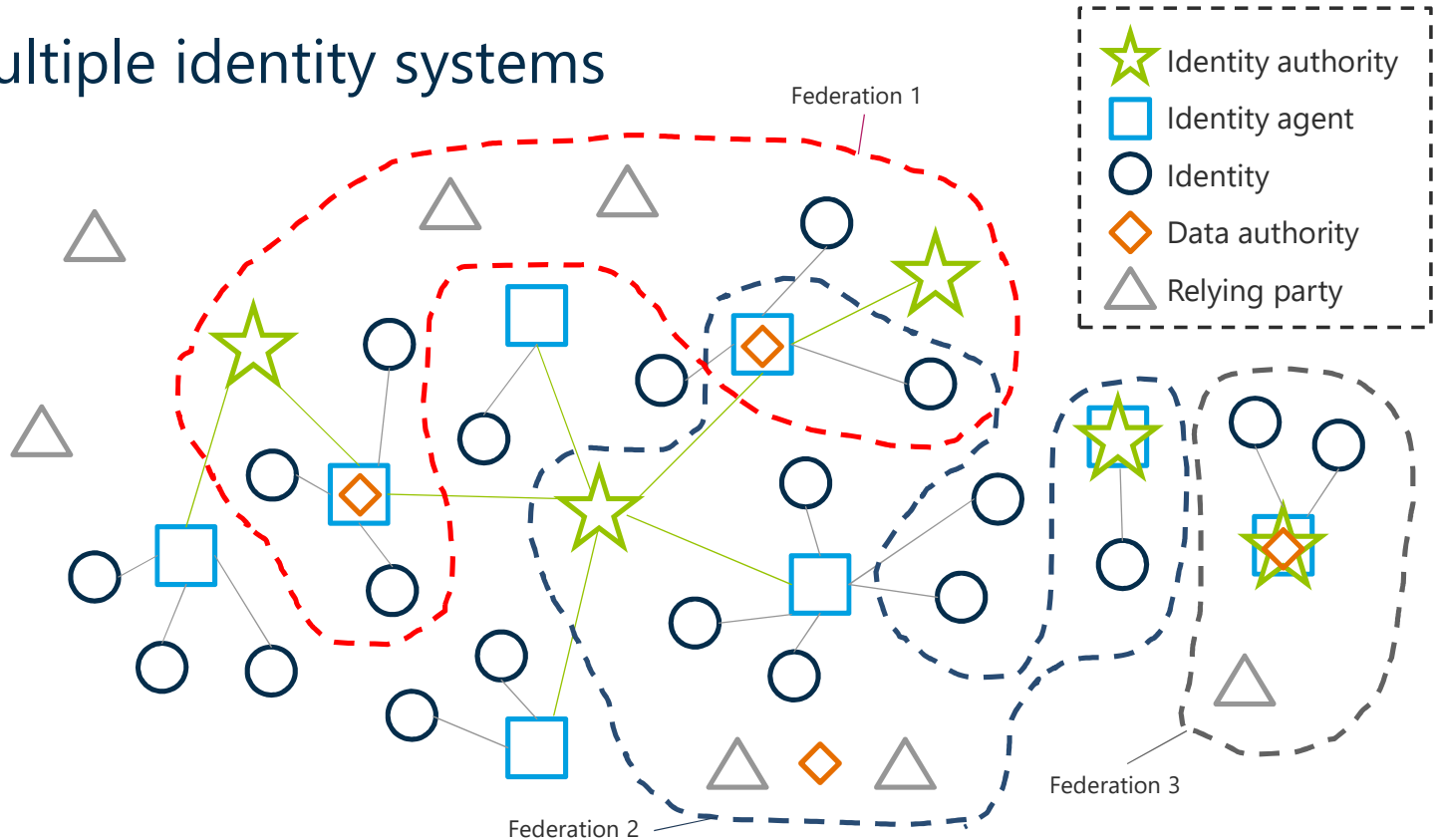
There are multiple data authorities that can be responsible for different claims / countries / user groups. For example, government X could run a data authority for (name, birthdate, address) for citizens of that country; airline Y for its frequent flyer numbers; etc.

### Use cases:

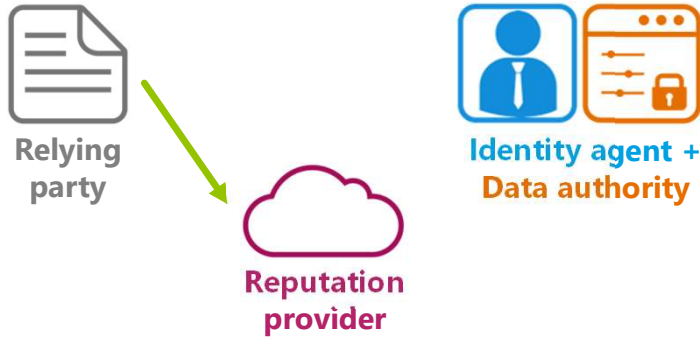
- A global, certified information distribution infrastructure

Requires a lot of work both organizationally and technically. Needs validated assertions plus a public ontology describing which data authority is responsible for each (claim, identity) couple.

# Multiple identity systems



# Use case NaN: Statistically trusted identity



## Use cases:

- More trustable weak identities

This is much easier to implement, but only works “statistically”, so there will always be failures.

(If you replace “agent + authority” with “ESP/ISP”, this is how anti-spam works)


There are multiple data authorities in the system. There are no trust anchors, but just one or more reputation providers.

The relying party trusts the reputation provider, which suggests if the data authority is reliable, according to past behaviour or accreditation policies.



# Next steps

- Some technical specification work is likely to be necessary (depending on the use case)
  - OpenID Connect recent extensions (Identity Assurance, Federation)
  - W3C standard on Verifiable Claims
- So, which of these use cases should be supported first?
  - We don't know – it depends on you
  - Looking forward to partners willing to try some of these cases
  - We will support and refine the specifications together in the Competence Groups



*Vittorio Bertola*  
*[vittorio.bertola@open-xchange.com](mailto:vittorio.bertola@open-xchange.com)*