



Lessons learnt From implementing a federated data space

Victor Mireles

Researcher, Semantic Web Company

EBDVF Prague

November 22, 2022



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871481




TRUSTS






- Federated platform
- Central Data Catalogue in the form of a Knowledge Graph
- Datasets, applications and services
 - Privacy preserving computation
 - Federated learning, de-anonymization risks, PSI
- Access control and machine-readable policies
- Recommender system
- Prototypes of
 - distributed immutable ledger
 - harvesting and policy translation from 3rd parties

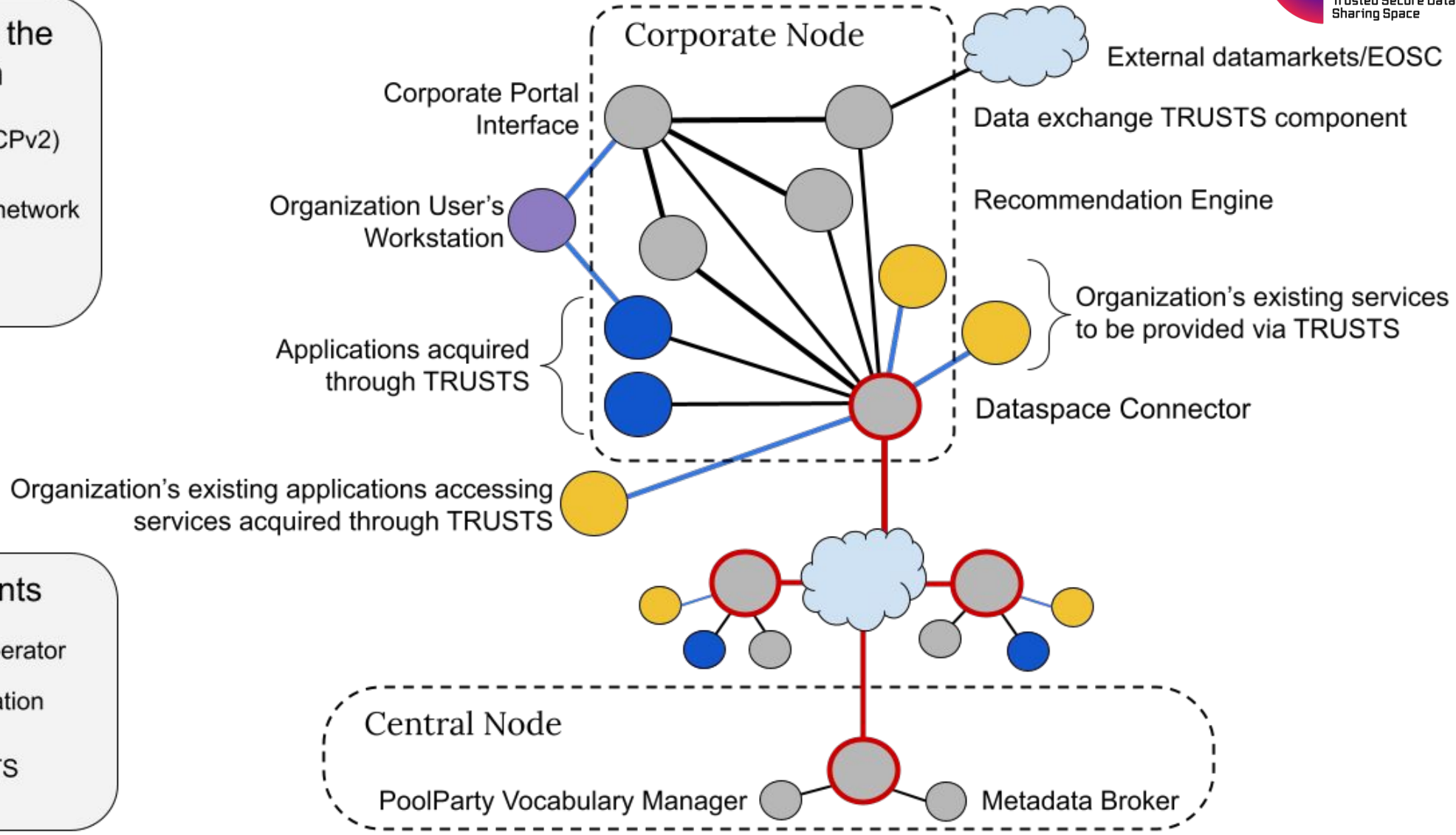
TRUSTS

Networks involved in the TRUSTS platform

-  Secure connection over the Internet (IDSCPv2)
-  Connection within the Organization's private network
-  Connection within the TRUSTS node

Origin of Components

-  Provided by TRUSTS Operator
-  Provided by the Organization operating this node
-  Acquired through TRUSTS platform



Join us tomorrow at 10:00
Bravo Room

Specify, specify, specify!

Tell apart actual & potential users from fellow idealists

- Requirement workshops should start from existing processes, not wishful thinking.
- Do not make a product for managers to buy and users to hate (or not use).

Refine specifications until they are workable

- Buzzwords are not specifications.
- Tickets should entail code, tests and documentation. Not narrative text.

Think about problems, not tools

- Beware of tools looking for problems.
- Consider the existing solutions before adopting to the newest fad.

Reuse, but with care

There is a lot of open source software that can help you

- Prioritize tools that have been around for a while and have good maintenance.
- Keep your stack up to date, also during development!

Prepare for the data model confusion

- Every project has their concept of asset, dataset, service, etc.
- Be careful when mapping between them, use mapping tools when possible.

Do not fork, contribute

- If you need to make small changes, push them upstream.
- Beware that maintainers and other contributors have their own paces and agendas.

Take barriers to adoption seriously



There are difficult problems, do not sell them as easy

- Crawling arbitrary stores, converting thousands of formats, distributing certs: difficult.
- Distributed, federated, decentralized, are all good, but difficult. Also to your expert devs.

Different industries have different preferred tools

- Potential users will not always retain a sysops team.
- Be prepared to port to that OS you thought was dead.

When making user projections, be honest about the above

- Will a potential user invest X€ in setup, just to trade 10 assets a month?
- How much will it cost to cater to those extra 10 SMEs?

Open Challenges



It is not clear what a data space user experience should be



There are still no *network effects* to drive adoption



There is a gap between reference implementations and usable tooling



Contracts are hard to specify in machine-usable ways



Metadata about structure and field-level meaning is not yet ready

<https://www.trusts-data.eu/>

Join us tomorrow at 10:00
Bravo Room