

Maintaining privacy and control

for IR in the context of Solid-like decentralized ecosystem

Rui Zhao

University of Oxford
rui.zhao@cs.ox.ac.uk



Online privacy is at risk, we know

For Usage:

- Platform-based storage
- Lack of user control
- Arbitrary downstream usage

For Understanding/Consent:

- “Biggest lie on the Internet”
- Lack of sensible transparency

Privacy: Secrecy, Control / Autonomy, Understanding...

Solution with Solid; IR on Solid

Solid itself provides

- User-based storage
- User-based (access) control
- Ephemeral Apps
- Permission request

ESPRESSO and others provide

- Solid-compatible search
- Access control-respected search
- Respect of decentralization
- ...

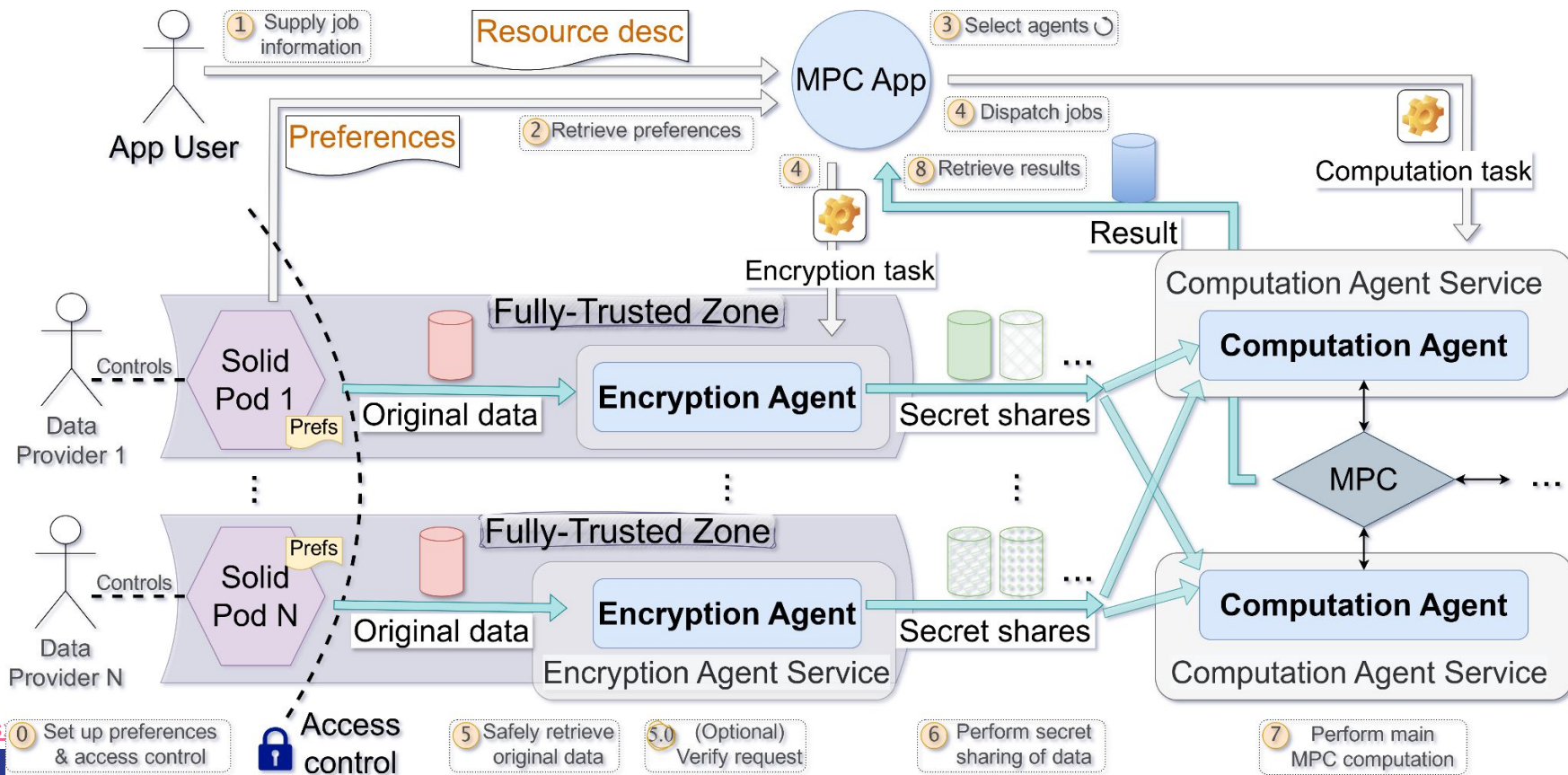
Great! But...

- What if the App does malicious things **after** retrieving my data?
 - TEE or sandbox can be a solution for **single-user** Apps
 - What about **multi-user** (**multi-data-provider**) Apps
- Power of access control is limited... E.g.
 - **How** do I know what App to grant permissions and what not to?
 - How can the policy for data usage **persists across** derived data & downstream tasks?
- How about **AI (models)**?
 - Can we allow users to **remain in control** of data for **AI-enabled retrieval**?
 - What additional properties may AI-enabled information retrieval provide?

Libertas: privacy-preserving collective computation

- A solution to privacy (secrecy) guarantee through cryptographic
 - MPC: (Secure) Multi-Party Computation
 - No data is seen by the data consumer (App)
- Supports arbitrary computation
- Maintains user (data provider's) autonomy
- No changes to Solid protocol needed
 - Or minimal changes for extra functionality

Libertas: privacy-preserving collective computation

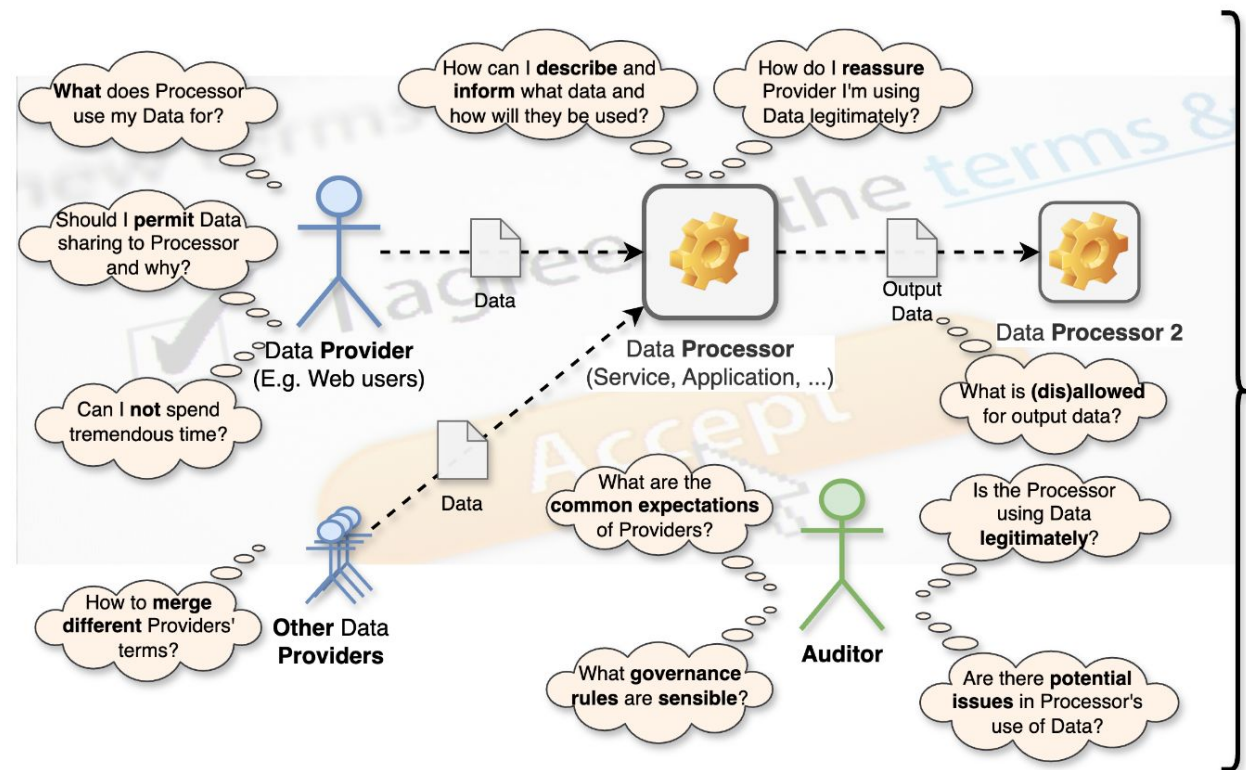


psDToU: perennial semantic Data Terms of Use

A (semi-)automated transparency and governance layer for decentralized Web

- Should I grant permission to an app?
 - a. Describe my preferences as formal policies
 - b. Obtain app's data usage policies
 - c. Check compliance
- How to keep this working for all derived data, and across apps?
 - a. Derive policies for output data
 - b. Support merge of policies
- How to minimize my efforts, as data provider, or developer?
 - a. Well-designed expressive formal language for this
 - b. Write-once and check-everywhere
 - c. Ontology for interoperability
 - Without requiring close cooperation

psDToU: perennial semantic Data Terms of Use



Data Terms of Use (DToU)

Formal Policy Language

Data Policy

App Policy

- Unambiguous
- Universal & Interoperable
- Expressive & Extensible
- Specify-once-use-everywhere
- Specified separately, but interacting between:
 - Data Providers
 - Data Processors

Logic-based Reasoning

Conflict Check

Obligation Check

Policy Derivation

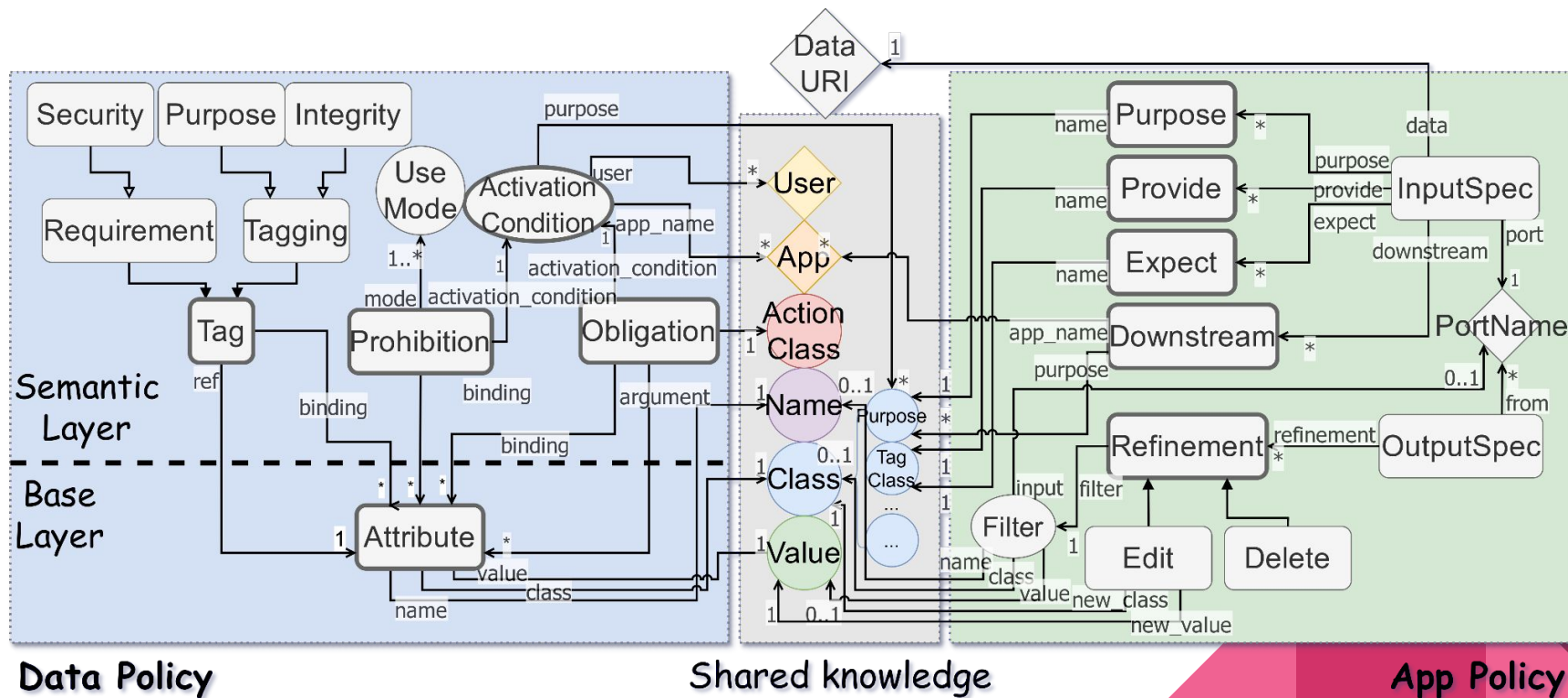
- Automated
- Deterministic
- Accountable

- Merge & Split
- Changes due to processing

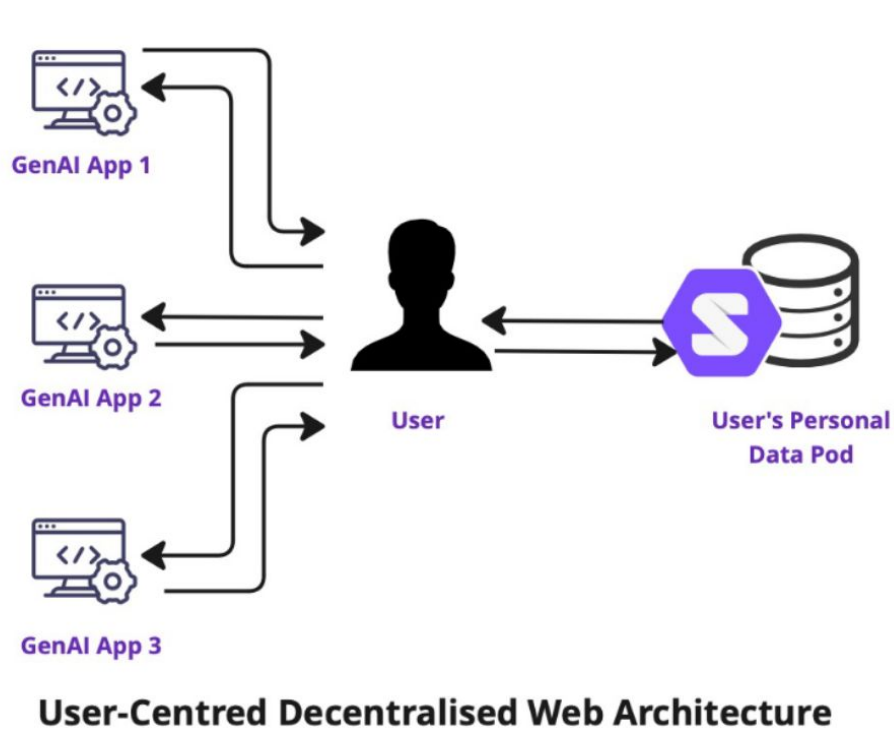
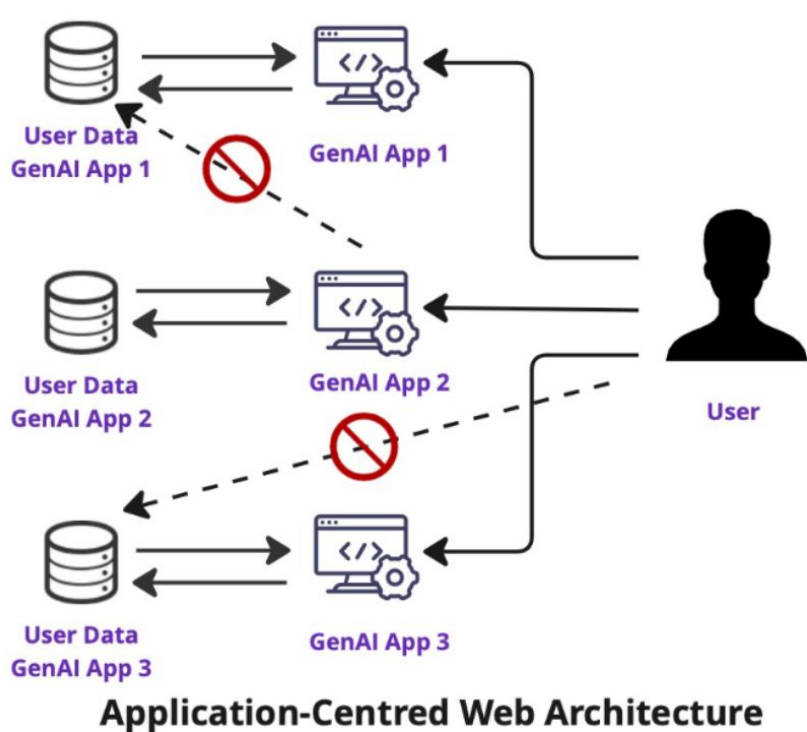
Integration with Solid

- Proof of concept
- Assisting decision making of authorisation
- Governing all downstream data automatically

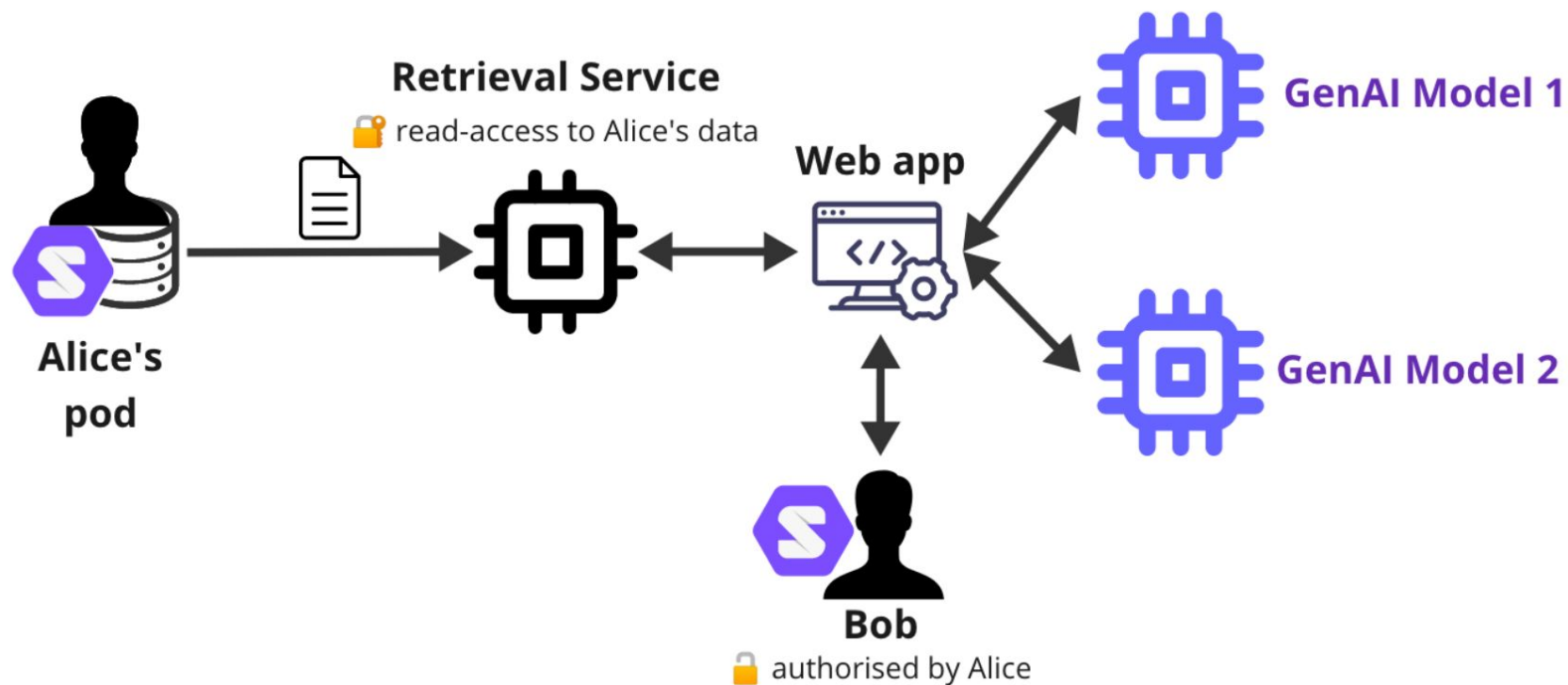
psDToU: perennial semantic Data Terms of Use



SocialGenPod: linking AI and Solid



SocialGenPod: a AI-enabled retrieval model



SocialGenPod: summary of opportunities

- Privacy, of course
- One identity & one authentication mechanism
- Multiple models, at users' will
- Social sharing of agents, and services
 - Potentially also for businesses
- Fuzzy data sharing, rather than exact data sharing
- Indirect sharing, rather than direct sharing
- Changing of models in-between conversation

Then what?

- Continuing the research above, sure
- Tools for other aspects
 - Provenance / Data lineage?
 - Any / Something from blockchain for undeniable records?
 - ...?
- Enforcement mechanisms
- Adoption and integration
- Standards / Agreements

<https://arxiv.org/abs/2309.16365>

<https://arxiv.org/abs/2403.07587>

<https://arxiv.org/abs/2403.10408>