PDS$^2$: Privacy-Preserving Decentralized Data Sharing System

Presentation by Lodovico Giaretta

PDS$^2$ is a project by Lodovico Giaretta, Ioannis Savvidis and Thomas Marchioro
Motivation
The Problems of Data Collection

Data Analysis and Machine Learning drive value generation in many sectors. Thus, data collection and exploitation are fundamental for business success.

For the user:

- No control over the data
  - Can’t control when, how or by who it is used
- No privacy guarantees
- No reward for the value generated

For organizations:

- High barriers to entry
  - Small orgs cannot compete without data
- Legal burdens due to sensitive data
- Infrastructural costs for data analysis
Existing Data Marketplaces (mostly for IoT)

For the user:

❌ No control over the data
  ○ Can’t control when, how or by who it is used

❌ No privacy guarantees

✅ Rewards for the value generated

❌ Often no user-centered design
  ○ Designed for SMEs as data producers

For organizations:

✅ Lower barriers to entry
  ○ Can more easily access any available data

❌ Legal burdens due to sensitive data

❌ Infrastructural costs for data analysis
PDS$^2$ Properties

For the user:

- **✅ Full control** over the data
  - Need explicit permission for each task
- **✅ Strong privacy guarantees**
  - Organizations do not directly see the raw data
- **✅ Rewards** for the value generated
- **✅ User-centered design**
  - Designed with individual users in mind

For organizations:

- **✅ Lower barriers to entry**
  - Can more easily access any available data
- **✅ No legal burdens** (no direct data access)
- **✅ Lower infrastructural costs**
  - Tasks run remotely in the marketplace
- **✅ Strong intellectual properties protections**
  - Tasks and results invisible to other players
- **✅ a share of the rewards**

For the infrastructure maintainers:

- **✅ a share of the rewards**
PDS$^2$ Architecture
General Architecture
Task Workflow

GOVERNANCE

1. TASK
2. TASK NOTIFICATION
3. DATA
4. TASK NOTIF.
5. DATA
6. ACCEPT
7. START
8. FINISH
9. RESULTS

EXECUTORS

STORAGE

CONSUMER

PROVIDER

ACCEPT

ACCEPT
Task Workflow

Fig. 2. Sequence diagram of the high-level interactions during the lifetime of a workload in PDS².
Modular Architecture

Configurable Aggregation Strategy (**Gossip**, Federated, Hierarchical, ...)

Configurable Storage Solution (Local, Cloud, Decentralized, ...)

Data-independent

Supports any data aggregation task (with caveats)

Configurable Privacy-Preserving Execution (**TEEs**, Homomorphic Encryption, ...)

Configurable Governance System (**Ethereum**, other blockchains, Trusted Entity, ...)

Figure 1: High-level architecture of PDS²
User-Centered Flexibility

Figure 1: High-level architecture of PDS²
Building Blocks
Privacy-Preserving Data Processing

Two types of **private information**:  
- Providers' data  
- Consumers' intellectual properties (e.g. code)

Must be **inaccessible to anyone else**  
- Including the providers’ own storage layer  
- Including the **executors** that run the code

Solution: use **encryption**!

Problem: how can the executors **perform the task, without seeing the code nor the data**?

Solution: **privacy-preserving data processing**!
Trusted Execution Environments

Isolated, tamper-proof hardware black boxes

- Impossible to see what is inside them
  - Even for the owner
- All outside communications are encrypted
- Possible to verify that the correct code is being run
- Just need to trust that the TEE is secure
- Widely available in Intel CPUs (Intel SGX)

"TEEs are the most suitable privacy-preserving data computation technique for PDS"
Decentralized Aggregation

Each executor can only compute **partial results**.

Problem: how do we merge them?

Solution 1: let the consumer do it! (e.g. Federated Learning)
- Scalability issues
- Fairness, transparency, auditability issues
- Privacy issues

Solution 2: **decentralized aggregation**! (e.g. Gossip Learning)
- Peer-to-peer protocols based on gossip communications
- Efficient usage of all available resources
- Runs on the executors (privacy-preserving data processing)

**Gossip-based aggregation is the most suitable technique for PDS**

---

Privacy risk?
Blockchain Technology

Natural solution for decentralized governance

PDS$^2$ requirements:

- Complex **smart contracts**
  - Manage the workflow of each task

- **Non-fungible assets** management
  - Unique, indivisible assets
  - E.g. data chunks, code

- **Fungible assets** management
  - Divisible, indistinguishable assets
  - E.g. currencies, reward tokens

**Ethereum** provides all of this, along with a vast, mature ecosystem

**Ethereum is the most suitable blockchain for PDS$^2$**
Open Challenges (1)

Rewarding Schemes

- Same reward for all participants? Reward based on amount of data?
  - Is it fair? Is all data worth the same?
- Reward based on the “added value” of each provider?
  - Computationally expensive; reward not known until the task is finished

Data Authenticity

- Prevent providers from forging fake data (useful for extra rewards!)
  - Possible with cryptographic signatures?
- Prevent users from replicating their data
  - I.e. send multiple copies of the data to different executors, to increase their rewards
  - Preventable with blockchain validation of non-fungible assets?
Open Challenges (2)

Indirect Privacy Leaks

- Certain consumer tasks might leak too much user information (maybe even on purpose!)
  - Static / dynamic task analysis to detect this?
  - Indiscriminately inject noise in the results (i.e. differential privacy)?

Data Discovery and Filtering

- Storage subsystem uses metadata to identify eligible data for each task
- “I want Fitbit data of people running when ambient temperature was less than 5°C”
  - Fine-grained metadata implies privacy leaks
  - Even participation in the task implies privacy leaks!
- Let the executors do the filtering?
  - Computationally expensive; eligibility and rewards not known in advance
Conclusions
PDS$^2$ in a Nutshell

*A user-centered decentralized data marketplace for privacy-preserving data processing*

**Not reinventing the wheel**: built on existing technologies, bringing together different research areas

**Driven by user requirements**: evolved from a simple sketch, growing to accommodate all needs

**Modular, flexible and extensible**: because technologies and needs constantly evolve
Project Status

Current Status:

- High-level architecture and interactions fully defined
- Most suitable technological solutions identified
- Vision paper drafted, to be submitted for peer-review on Jan 25

Future Directions:

- Proof-of-concept implementation
  - Test overall feasibility of the architecture
  - Evaluate different technologies for each component

- Follow-up work on each separate component
  - Modular design allows parallel work on different aspects
  - Each of us will work on a specific component, based on personal expertise and interest
  - Anyone can design additional components or different implementations!
Any Questions?