



DeepVerse

Train AI models collaboratively on Blockchain
and maintain data privacy

Data privacy:

- **GDPR** came into effect after a series of scandals: *Cambridge Analytica* had collected and exploited *Facebook* user data; vulnerabilities in *Google+* had exposed data from half a million users.
- The value of **collecting data** has **skyrocketed** ever since, like *GlaxoSmithKline* trying to get access to genetic data from *23andMe*. But they do not trust each other.
- Data are **essential** to advance science and technology for the good, e.g. fraud detection, drug design for COVID.

DeepVerse is here to protect data privacy, while explore the value within the data.



Problems:

- **High costs**

- Data collection is expensive, and face challenges from both legal, like GDPR and technological.

- **Low efficiency**

- Data utilisation is low, often the same data may be collected by different companies.

- **Hostile environment**

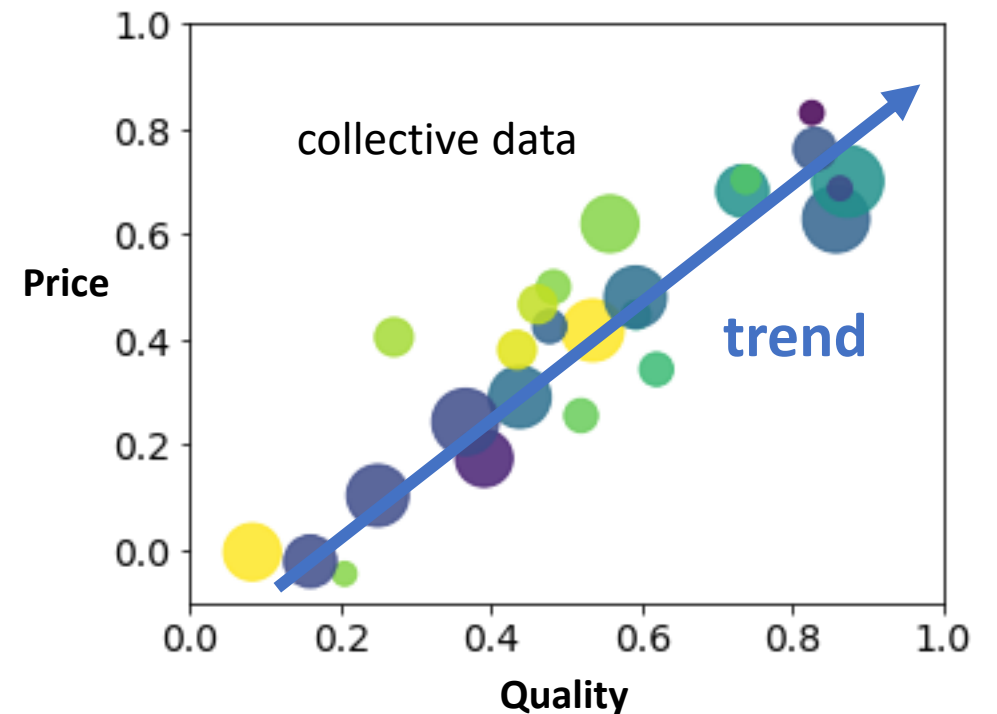
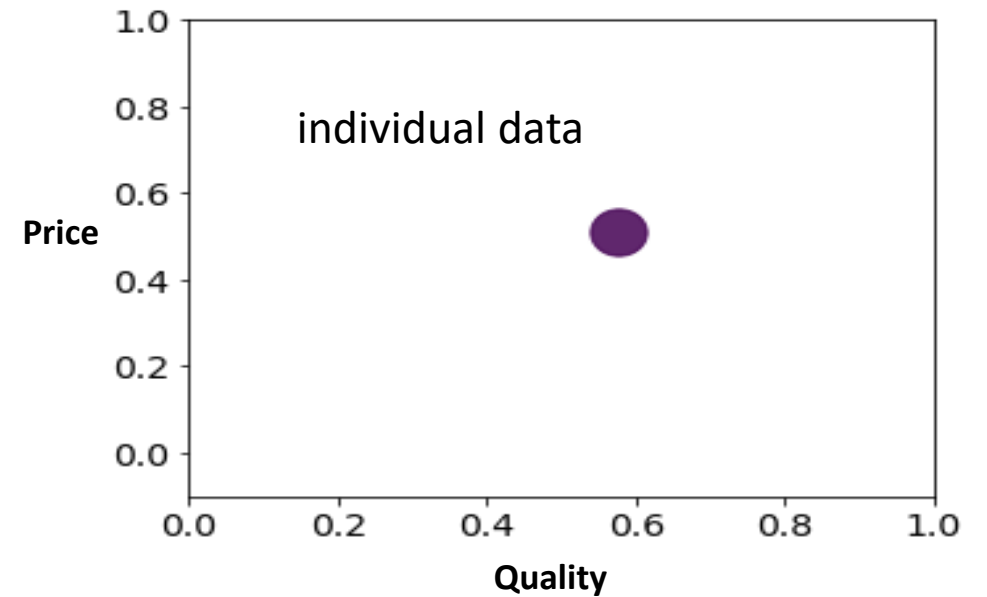
- Data sharing is rare, once you shared it, it's not 'yours' anymore!

*However, mutual benefits can be achieved if we collaborate together. We are building a platform to minimize the level of trust required, **aggregate** all source of information, **reduce** costs, and **accelerate** discoveries.*

Value of the data:

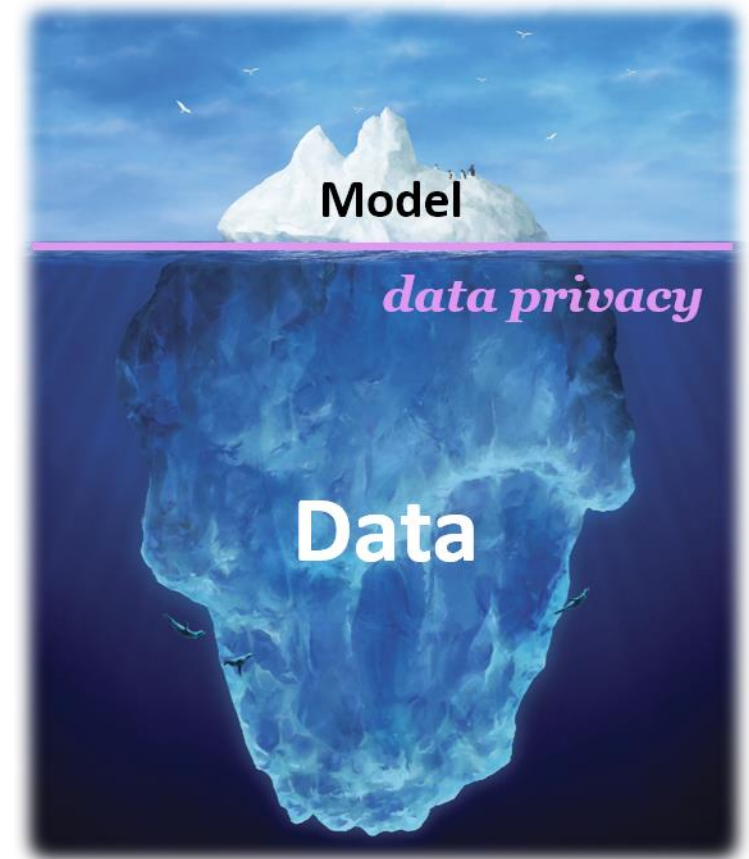
- The individual data point is not interesting, it's the **collective phenomenon** emerged from many data that matters.
- **AI** can identify the **trend** or **model** that gives us descriptive or predictive powers.

Instead of sharing data, we can share models and protect the data privacy.



Solution:

- We are the first to train **machine learning** models using private database, and store the models on the blockchain for others to validate and improve.
- **Share the model** instead of the data, **maximize** the value within the data using **AI**, and the data never leave you.
- With the model, you can easily verify the accuracy, but you can't see the data or reverse-engineer the data from the model. This is similar to the asymmetric encryption used in blockchain.



Blockchain:

For a multi-party who don't want to share their data, but would like to tackle a problem together.

Blockchain provides:

Static registry or Identity for who own the model,

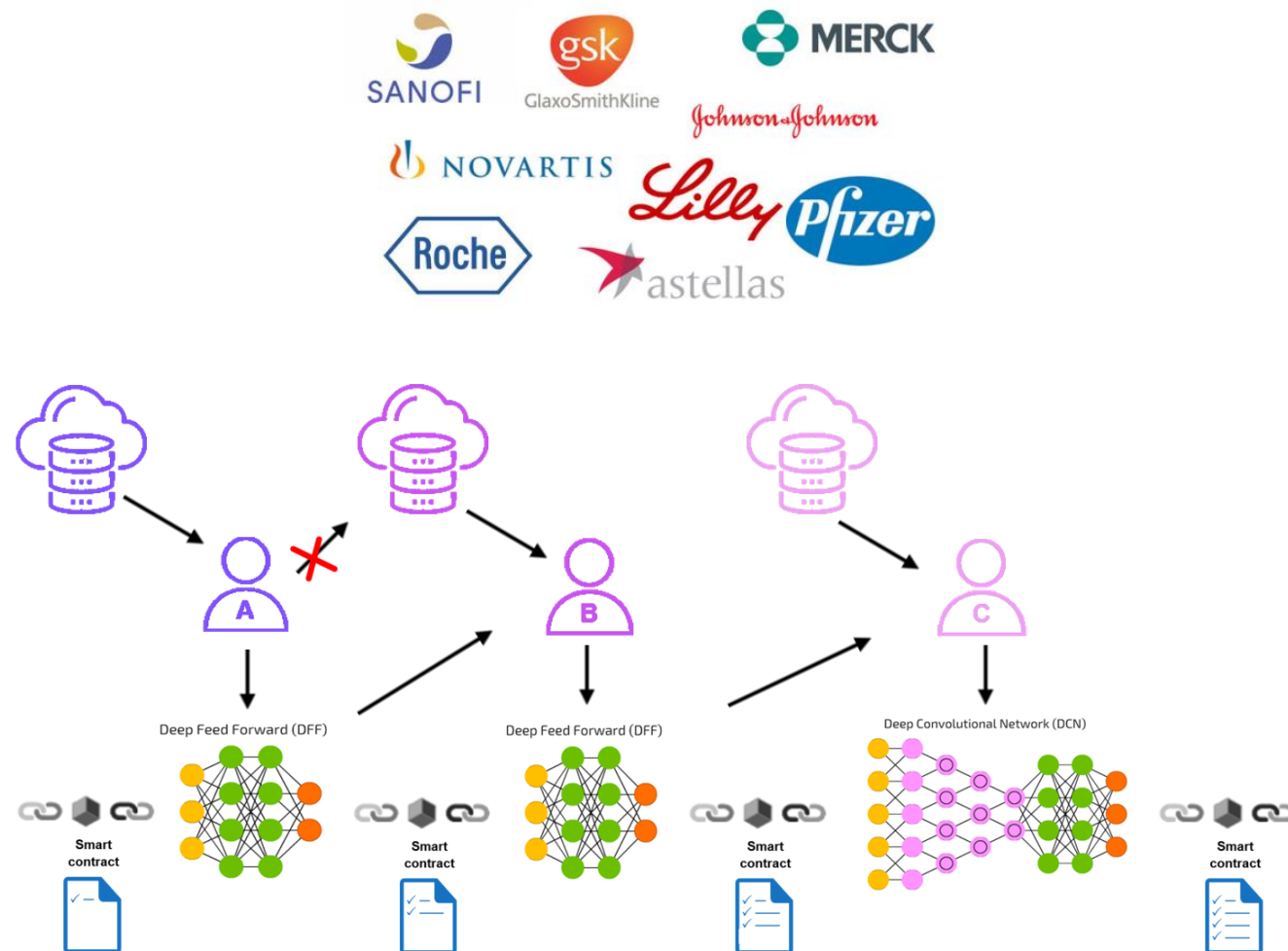
Smart contracts to trigger actions automatically once the pre-defined conditions are met,

Payment registry updates the rewards for the contributing parties.

Illustration:

Alice, Bob, and Charlie, who are *pharmaceutical companies* and each has a private database of potential drug molecules with their efficacy and assay information. They all want to build a neural network (NN) model for new drug discovery, e.g. vaccines for COVID. They can either share the database on the blockchain directly with access control or keep their database private and train the model together.

With DeepVerse, Alice can propose and upload a NN model to the blockchain. Bob can continue on Alice's model and train with his data and upload the improved model to the chain. (Note, Alice can't access others' data). Charlie can evaluate both models, and make a more creative model which outperforms previous ones. With an active community with data & model, the performance would boost under this iterative process. We developed smart contract to reward contributions based on pre-defined conditions, such as the accuracy.



Advantages:

Common Collaboration

Not “your” data anymore

Credit unknown

Reward system unknown

Execution not guaranteed

Closed community

DeepVerse

Data privacy

Contribution quantified

Fair reward system

Automated execution

Open community



Alpha version

website: <https://deepverse.co.uk>

video demo: <https://lnkd.in/dhSEXQX>

The screenshot shows the DeepVerse website interface. At the top, there's a navigation bar with 'HOME', 'PILOT', 'TEAM', and a purple 'MANUAL' button. Below the navigation, a header reads 'EXPLORE DEEPVERSE STEP BY STEP'. The main heading is 'Step 1: Build Your Model'. A sub-heading says 'Here is just a demo of Tensorflow, in practice we recommend that you do this on' followed by a blue 'Google CoLab' button. Below this, there's a control panel with a play button, a refresh button, and sliders for 'Epoch' (set to 000,000), 'Learning rate' (0.03), 'Activation' (Tanh), 'Regularization' (None), and 'Regularization rate' (0). The main area is a neural network diagram with '2 HIDDEN LAYERS'. The first hidden layer has 4 neurons, and the second has 2 neurons. The input layer has 2 neurons labeled X_1 and X_2 . The output layer has 2 neurons. A scatter plot on the right shows the output data. A 'REGENERATE' button is at the bottom left.

The screenshot shows the 'Step 2: Store Models On Blockchain' section. A red box highlights the word 'Store'. Below it, text says 'Arweave and IPFS supported currently, more to come!'. There are two buttons: 'ARWEAVE' and 'IPFS', with 'IPFS' being highlighted in purple.

IPFS

Upload or download model with IPFS!

IPFS Node status: online
Connect to local ipfs node via websocket

address

Local Host Status: Not connected

Upload a file to IPFS

No file chosen

Search for a file hash on IPFS

hash

Step 3: Race Your Neural Network And Get Rewards!

Tezos! Till Everyone Zap One Score!

Please use cautiously and at your own risk.

Game	Loss	Model

Interact with the contract Direct Access to Test Networks With the Tezos Client

1. Private Key and Account

Warning: this is for test networks such as Babylonnet and Zeronet, do not input Mainnet
A private key is needed - you can use the [Tezos Faucet Importer](#) to retrieve one.

Private Key:

With an private key, you can

Account public key hash:

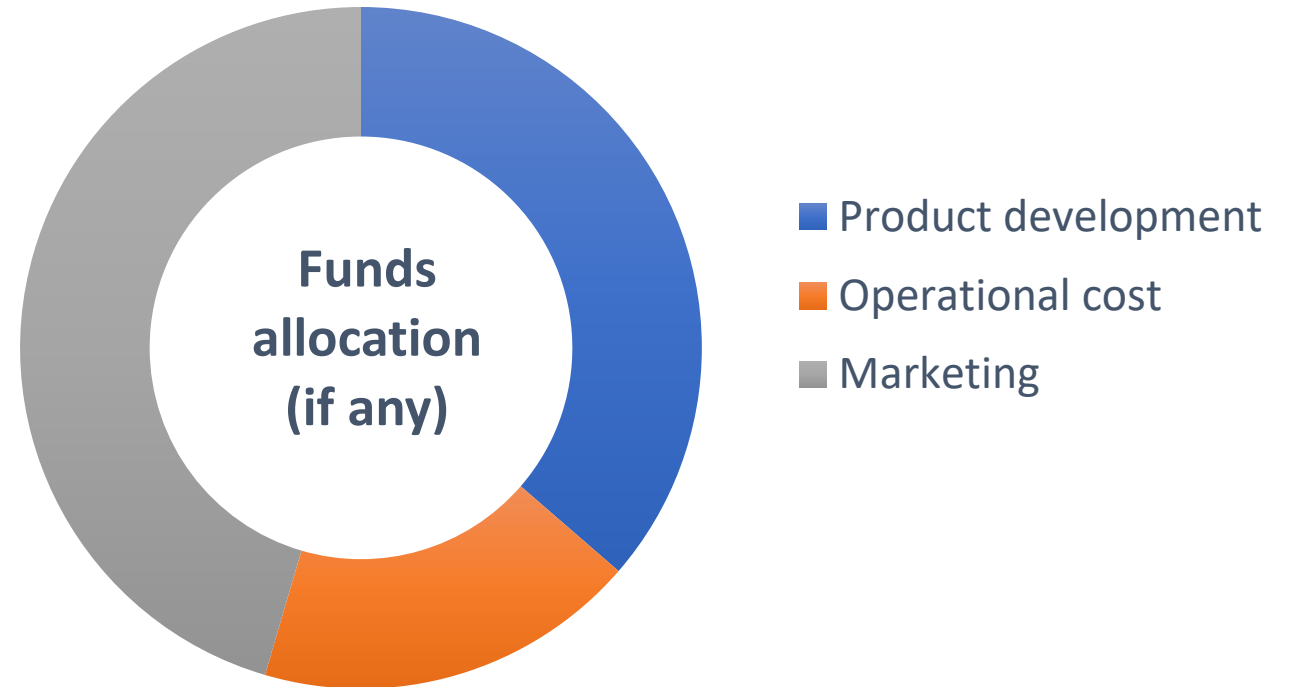
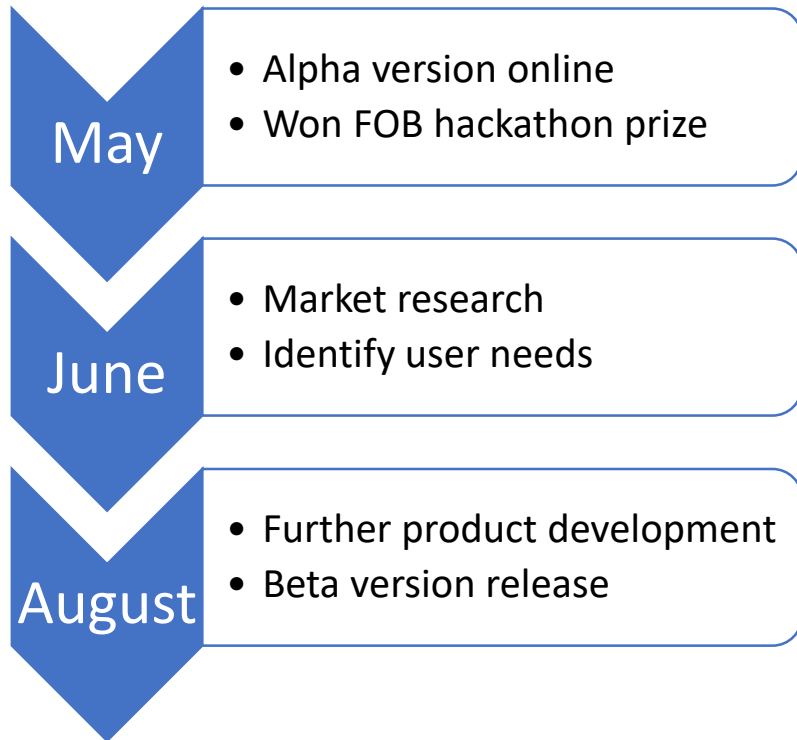
2. Transaction Parameters

Amount:

Fee:

Storage Limit:

Timeline:



We have not raised any funding so far.

We won the Future of Blockchain 2 hackathon with £5,000 prizes.

Team:

3 Cambridge PhDs in theoretical physics and computing each with 6 years+ programming experience

- **Fredrik Liu**, Blockchain and AI (full stack), Fortran, Python. developed decentralized in-game items trading platform, novel AI algorithms for materials and drug discovery. won G-research algorithm trading competition.
- **Lupeng Yang**, Blockchain (back-end), Fortran, Python, Javascript. won G-research algorithm trading competition.
- **Siyu Chen**, Machine learning and scientific computing (back-end), Python, C++. Lead programmer with a trach record of publications.



We sail together.

Lupeng(first mate), Fredrik(skipper), Siyu(crew)
Fredrik & Siyu

AI advantages:

- We have AI algorithms to exploits **all available** information
- Apply deep learning to high-value **fragmented** data
- Understand and exploit **probability distribution** to focus on most confident results
- Developed algorithms to deliver insights from machine learning models spanning **different private datasets**
- Designed and experimentally **verified** stem cell therapies using AI for *Bioprocessing Technology Institute*.