# METAMORPHIC DAO

# The Current State of DAOs

- Most DAO implementations are limited to basic voting platforms.
- Execution logic is rigid and challenging to change once established.
- Current DAOs operate within the limitations of the virtual machines of the blockchain platform and it is very difficult or expensive to implement complex execution logic

# Philosophy of Continuous Evolution and automated governance

- DAO execution logic is similar to the constitution of a country or rulebook of an organisation, and just like the constitution or rulebook needs continuous change through amendments and legislative processes, so should be the case for DAOs

- Next generation of DAOs also demand Artificial Intelligence and Machine Learning assisted governance which required even more intensive execution capabilities.

# A New Era of DAOs with Cartesi

- Cartesi introduces app-specific Layer 2 rollup chain which allows to build decentralized apps directly on the Risk V machine with Linux backend

- This allows to build decentralised applications of any complexity or requirements using tools and packages used for building traditional applications

# INTRODUCING METAMORPHIC DAO

- Metamorphic DAO is built on top of Cartesi, allowing it to handle complex runtime logic and provides malleable execution logic through a dynamic governance process.

- Anyone can propose code changes to the execution logic of the DAO, and if approved through consensus protocol, it will become part of the runtime environment that governs the DAO

- Powerful Risk-V architecture of Cartesi machines allows to write runtime logic of any complexity using traditional machine learning tools and packages like C, Rust, Python and packages like Scikit-Learn and TensorFlow

- Access to these powerful tools allows to write complex execution logic code that can help in ML assisted governance.

# HOW IT WORKS?

*In the beginning, there are 2 data stores and 2 execution rules, and the whole DAO is bootstrapped using them by manipulating them…*

# DATA STORES & EXECUTION RULES AT GENESIS

- **Amendment Proposal Dataset:** It is empty at genesis

| name | Execution logic | description |
|---|---|---|
| | | |
| | | |

- **Rules Dataset:** It contains two rules at genesis

| name | Execution logic | description |
|---|---|---|
| Create Amendment | 0x12a77daf76ad6ada8 | Code to add new amendment to amendment proposal table |
| Amendment Consensus | 0x12a77daf76ad6ada8 | Code and conditions to move amendments proposals to rules table |

The consensus rule at the genesis of DAO says that if there is any proposed amendment, make it a rule..

# Bootstrapping process

1. Propose the amendment rules for DAO membership and voting using “create amendment code” and make them a rule using the “genesis consensus code”
2. Propose a new consensus rule using “create amendment code” and replace the genesis consensus  rule using “genesis consensus code”
3. Now the DAO is like any normal DAO and anything new feature can be proposed and passed using consensus rules