Impact certificate: Edelweiss protocol compiler

Description of work: Design and development of Edelweiss compiler MVP, featuring:

- Comprehensive type system for modeling decentralized and blockchain data, functions and services
- Code-generation of encoders and decoders for data schemas
- Code-generation of RPC clients and servers
- Support for user-defined target language backends (e.g. Go, Rust, JavaScript)
- Support for user-defined RPC networking stack backends (e.g. DAGJSON-over-HTTP, CBOR-over-libp2p)
- Support for user-defined serialization backends (e.g. IPLD, ProtocolBuffers)
- General framework for code-generation

Link to work: https://github.com/ipld/edelweiss

Time period of work: Dec 2021 to June 2022

Contributors and owners of certificate: Petar Maymounkov, Adin Schmahmann, Steve Loeppky

Plural Markets and the Creative Uncertainty of Open Source

Petar Maymounkov (Protocol Labs)

petar@protocol.ai
twitter.com/maymounkov

Overview

- There are no mechanisms to support the **day-to-day operational needs** of open-source and **embody community values**.

- We describe **plural market mechanisms** supporting open-source operations, as well as community **governance based on the same mechanisms**.

Motivation (1 of 3): Observation

There is a compelling similarity between **public companies** and **open-source projects**, despite their differences.

- While public companies produce **private goods**, whereas open-source projects produce **public goods**,
- Both **require investment**, build products **under uncertainty**, and **depend on moving targets**

Motivation (2 of 3): Financing, hedging and macro

- **Like public companies**, open-source projects have complex lives involving much **planning under uncertainty**.
  But flexible and accessible mechanisms for **financing** and **hedging risk** are missing.

- **Like the public sector**, the open-source ecosystem depends on **visibility** into **business metrics** and **macro trends**.
  But **transparent indicators** and **prediction mechanisms** for these are missing.

Motivation (3 of 3): Prior work and differentiation

We focus on facilitating **operational needs** of open-source projects (rather than inception needs).

Therefore we are **interested in mechanisms** that are:

- **endogenous:** driven by the project's needs and timelines
- **asynchronous:** performed on continuous basis, as needed
- **fine-grained:** pertaining to sub-product technical decisions
- **streamlined:** self-serve, no dependence on manual processes, governed my smart contracts

This is in **contrast** to mechanisms like **Quadratic Funding** (e.g. Gitcoin Grants):

- **exogenous:** driven by exogenous community needs
- **synchronous:** quarterly grant schedule
- **coarse-grained:** product-level
- **organized:** manual curation and socialization of bounty targets

Plan

- Meaningful **structure** abounds
- Financing using **plural equity tokens**
- Hedging risk from dependent and related projects using **binary options**
- **Visibility** and prediction of business indicators
- **Governance** itself seen as the operation of an open-source project

Meaningful structure abounds

Unlike the ecosystem of public companies,
The open-source ecosystem is very **transparent and structured**:

- Project **entities** could naturally be identified with **GitHub organizations**
- Project **dependencies** are **extractable programmatically**
- **Business metrics** (health, moat, popularity, impact, etc.) are **computable programmatically**
- **Business predicates** (e.g. *"there is an EVM implementation in Rust"*) are **verifiable programmatically**

We will use these to imagine streamlined, self-service financial mechanisms.

Financing (1 of 5): Plural equity tokens

The **entity** being financed is an open-source project, which
we model as a **GitHub organization**.

We propose **plural equity tokens** (an analog of company shares), which
can be **issued**, **sold** and **bought/back** asynchronously on an exchange, as needed.

Issuance must be self-serve and frictionless,
but **governed by a smart contract to protect investors**:

- Quantity of **issued tokens** is public
- **Anticipated** schedule of future issuance is public
- **Unanticipated** changes governed by a quadratic vote of shareholders (explained next)

Financing (2 of 5): Value of quadratic influence

The **value of an equity token** is **captured** in the GitHub organization — representing the organization's **URL namespace** and the **team** — not the code itself.

> Why? Project users **depend on the URL**, not on the code.
> *Consequently, for instance, forking a repo has no effect on equity tokens.*

The value of **private company equity** is quantified as present discounted **future dividends**.
How is the **value of an open-source equity token quantified**?

> The value is the **influence** over the project, bestowed by token ownership:

- **voting power** on PRs (code, issuance, features, governance)
- **quadratic influence rule**
  - merge PR, if (votes for - votes against) > threshold
  - voting power = sqrt(value of tokens owned)

Financing (3 of 5): Partial common ownership

Partial Common Ownership of **land** *(Radical Markets, Glen Weyl, et al.)*:

Harberger Tax:

- Tax a plot of land, based on the owner-declared value of the plot
- Bind the owner to sell the plot at the self-declared value, at any time

Universal Basic Income:

- Distribute tax evenly to the community

Partial Common Ownership of **equity**:

Harberger Tax:

- Tax a token, based on the market value of the token
- All tokens must be on the limit order market book for sale, perpetually

Universal Basic Income:

- Distribute tax for governance costs and evenly to the community

Financing (4 of 5): Rationale

The Harberger Tax is the **marginal cost** of owning the token.
The influence over the project is the **marginal value** of owning the token.

Price **forces down**:

If the token price is too high,
the marginal cost of ownership (tax) is higher than the marginal value (influence).

Price **forces up**:

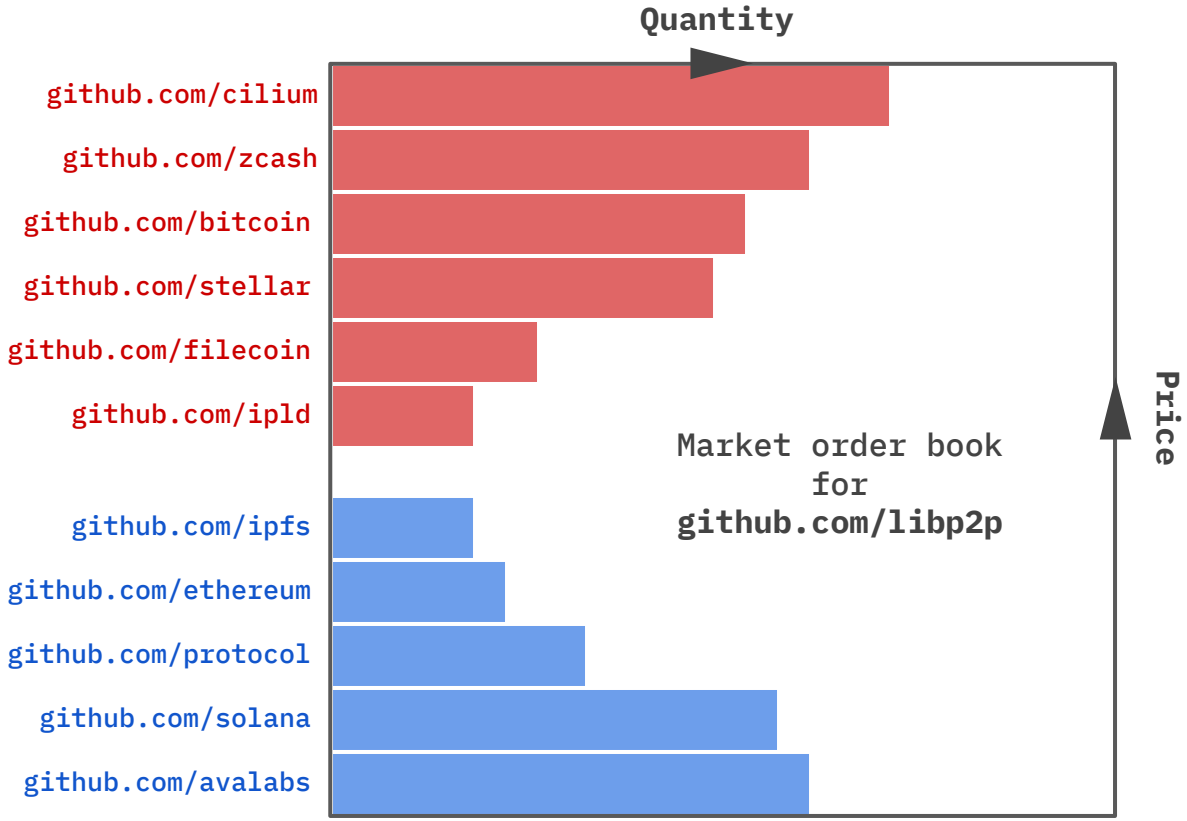
If the token price is too low,
competitors will take ownership (influence) at a bargain.

Financing (5 of 5): Transparent market order book

The entire equity supply is perpetually on sale: No market making is necessary

Ownership of equities is public?

Buyers/owners can only be other GitHub organizations



Hedging risk (1 of 3): Binary options

A **binary option** is a contract paying $1 in the event a condition is met.

Examples:

"The Go programming language supports generics, by 2022."
"IPFS is included in the Homebrew package manager."
"The Protocol Labs libp2p library is a transitive dependency of the Brave Browser."
"Julia beats C++ in a given speed benchmark."
"The Ethereum Virtual Machine (EVM) is implemented in Rust."

Option conditions must be **verifiable statements**.
They can be verified by groups of community members (GitHub organizations) for a fee.
They can be implemented using a zero-supply market.

Hedging risk (2 of 3): Market sides

	Buy side	**Sell side**
Hedging	*Projects **insuring** themselves against the condition happening*	*Projects **insuring** themselves against the condition NOT happening*
Speculation	***Analysts** who think the condition is likely to happen*	***Analysts** who think the condition is unlikely or infeasible*
Representing agency	***Developers** who plan to make the condition happen*	***Attackers** with agency to prevent the condition from happening*

Hedging risk (3 of 3): Insurance for uncertainty

Buy side:

The NewBlockchain project is debating Rust vs. Go.
Decides to use Go, because there is no EVM in Rust now, even though Rust is faster.
Buys *"EVM is implemented in Rust"* option to finance pivoting to Rust,
if the decision assumption is violated.

Sell side:

The NewBlockchain project is debating Rust vs. Go.
Decides to use Rust, because Go is slower, expecting there will be an EVM in Rust.
Sells *"EVM is implemented in Rust"* option to finance building a Rust EVM,
if the decision expectation is not met.

Visibility (1 of 1): Project indicators and trends

Visibility is key for a healthy community. Indicators can be hedged/predicted using options.

Open-source (GitHub) visibility is in the **Alta-Vista age**. It needs to be in the **Google age**.

"Alta-Vista" visibility captures **local (per-project) indicators**, not meaningful on their own:
- stars, forks, developers, contributions, pulse, etc.

"Google" visibility captures meaningful **global (across projects) indicators**, defined **recursively**:
- project impact on the rest of the ecosystem:

$$\text{impact}(\text{project}) = \sum_{\substack{\text{dependant} \\ \text{project}}} \text{impact}\begin{pmatrix}\text{dependant} \\ \text{project}\end{pmatrix}$$

- project quality based on developer experience, and vice versa:

$$\text{quality}(\text{project}) = \sum_{\text{developer}} \text{experience}\,(\text{developer})$$
$$\text{exprience}(\text{developer}) = \sum_{\text{project}} \text{quality}(\text{project})$$

"Google" **indicators are costly**, but **can be funded by Harberger's tax**:

access, network, storage, compute

Governance (1 of 2): Revenue and responsibilities

Government is a DAO = smart contract = open-source project.

Revenue from Harberger's tax, used to pay for:
- welfare (Universal Basic Income)
- operational costs (gas, infra, quadratic votes, etc.)

Responsibilities are:
- **operational:** run markets and tax collection (gas and infra)
- **observational:** collect and publicize business metrics (infra, service fees)
- **evolutionary:** self-improvement via a **plural policy mechanism**

Governance (2 of 2): Policy

Governance is an open-source program (e.g. an Ethereum smart contract).
Therefore, **a policy proposal is a pull request (PR).**

Government needs **plural mechanisms** for considering PRs both:

Synchronously: Using **quadratic voting**, and

Asynchronously: Using the **plural equity mechanism**

- Review policy PRs with the voting power of your equity holdings
- Voting power = sqrt(fraction of overall market capitalization owned)

Thank you!

Petar Maymounkov (Protocol Labs)

petar@protocol.ai
twitter.com/maymounkov