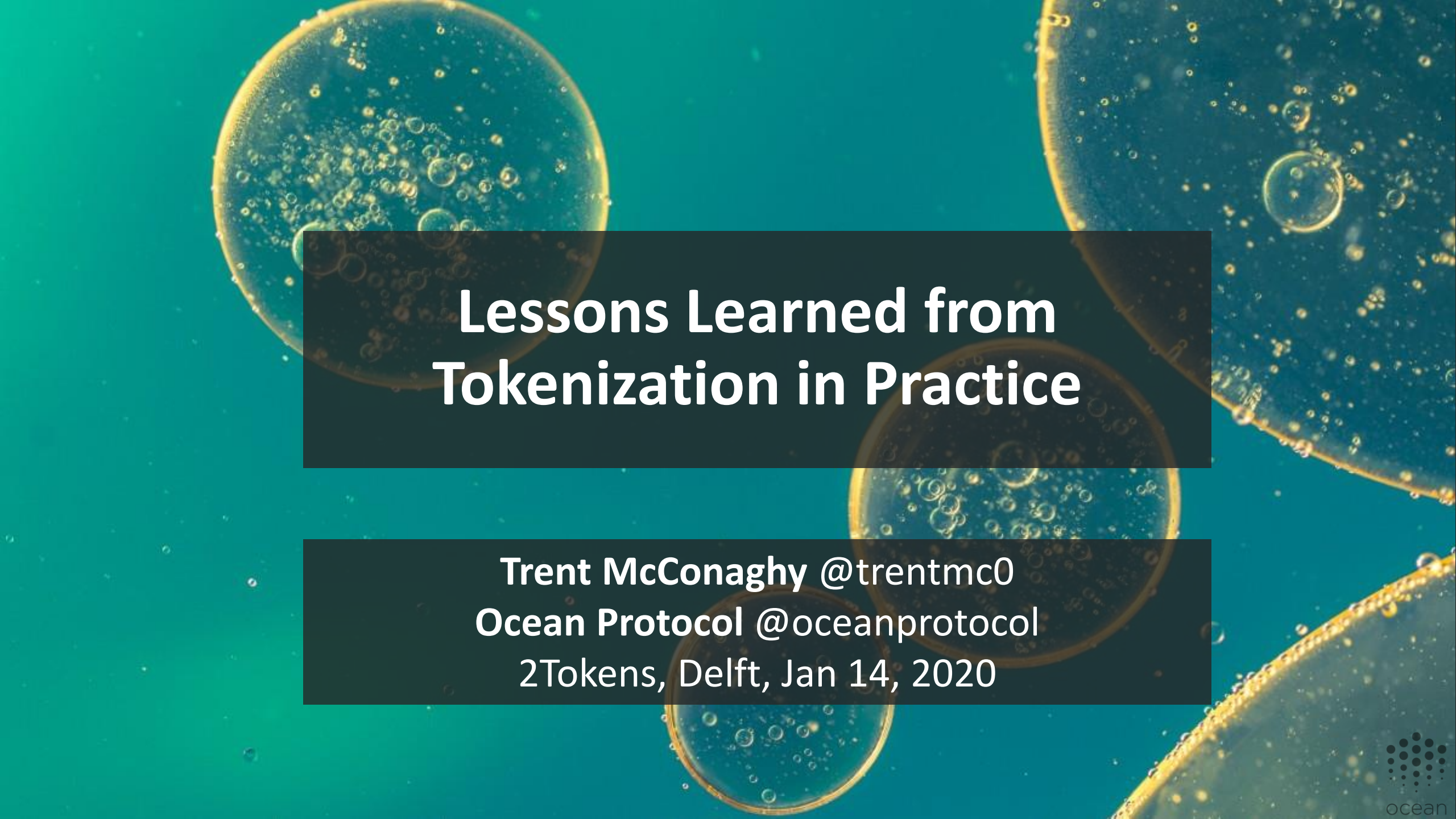
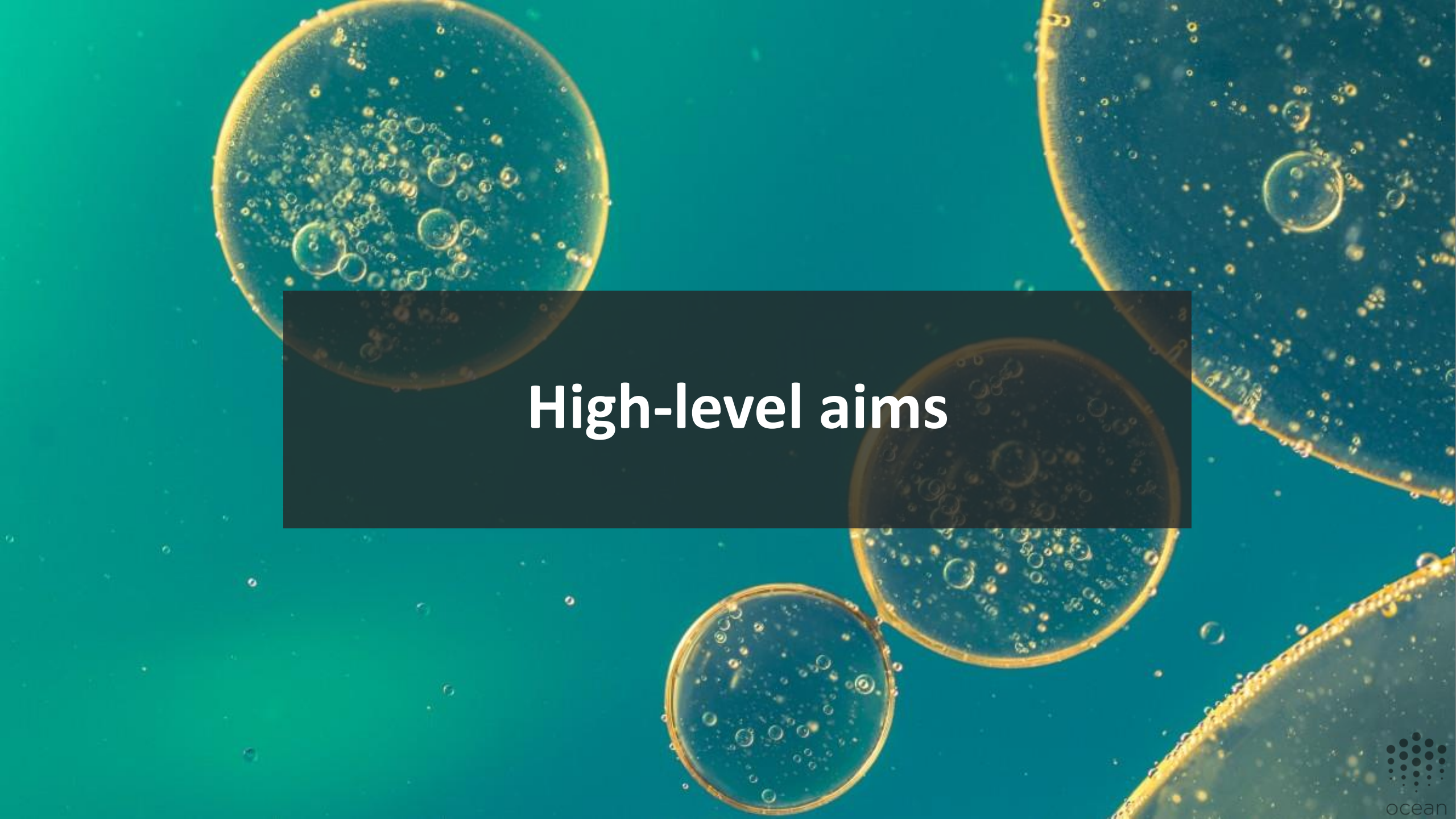

Lessons Learned from
Tokenization in Practice
Trent McConaghy @trentmc0
Ocean Protocol @oceanprotocol
2Tokens, Delft, Jan 14, 2020
ocean

Outline

- High level aims of my token-based project (Ocean Protocol)
- Early design iterations
- Learnings, towards an improved methodology
- Emergence of TE community
- Deployment / verification of design
- Learnings since initial design / deployment / verification

ocean



High-level aims

Silo mo' data

↓

Mo' accuracy

↓

Mo' $

Default incentive: hoard the data

ocean

“Show me the incentive
and I will show you the outcome.”

-Charlie Munger

You can get people to do stuff

by rewarding them with tokens.

This is a superpower.

Change the incentives!

~~Silo~~ *Pool* mo’ data

⬇

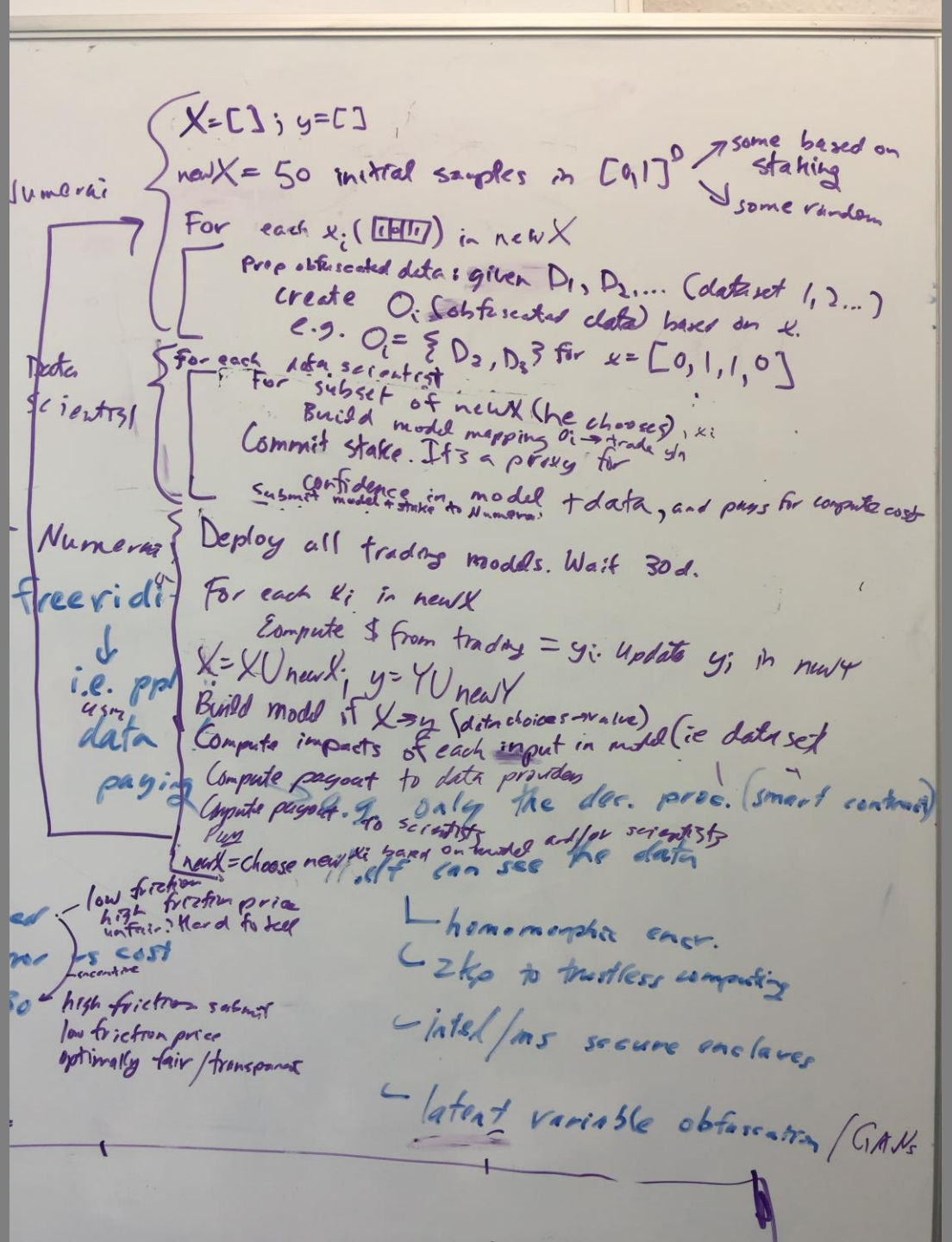
Mo’ accuracy

⬇

Mo’ $

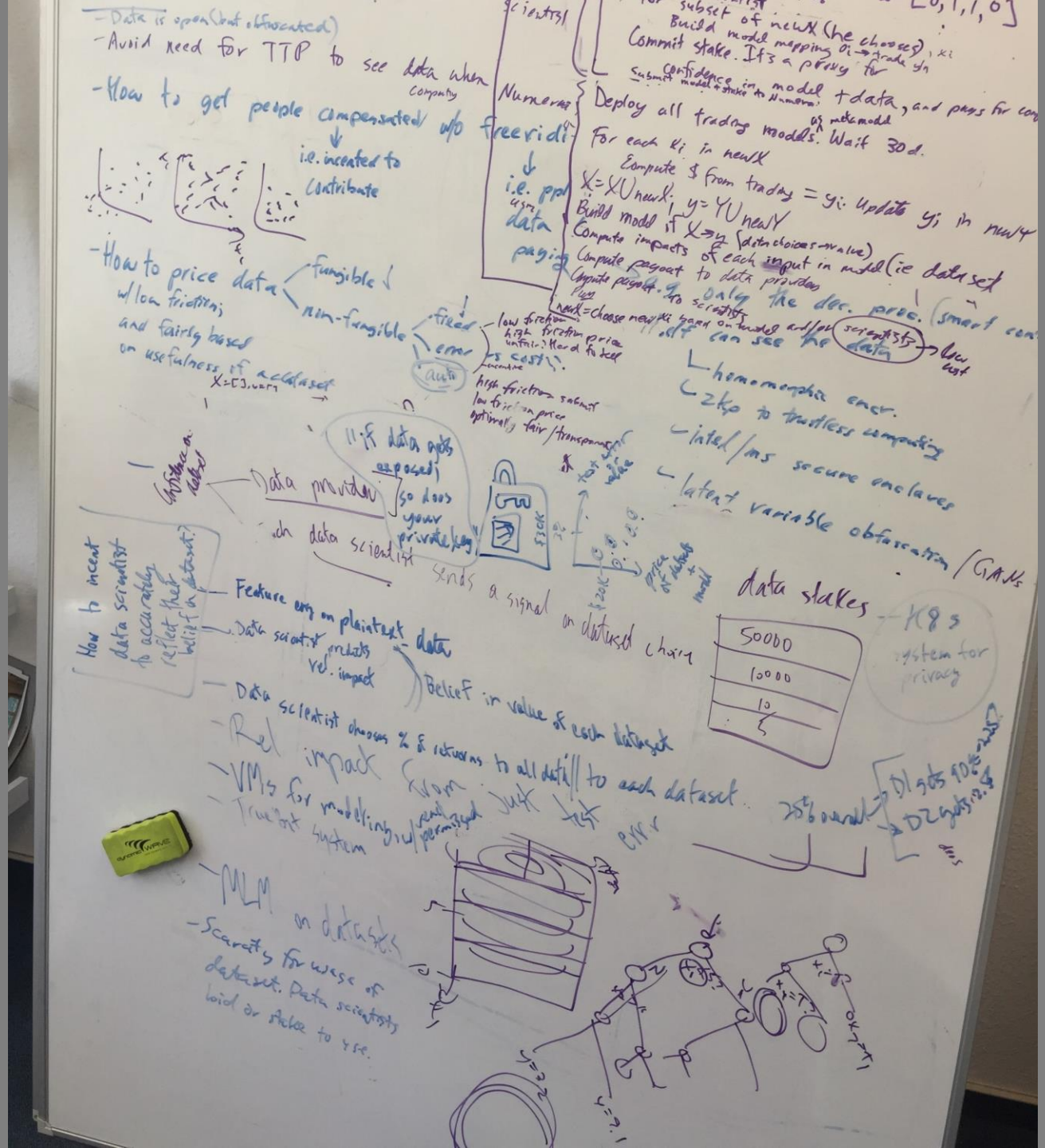
ocean

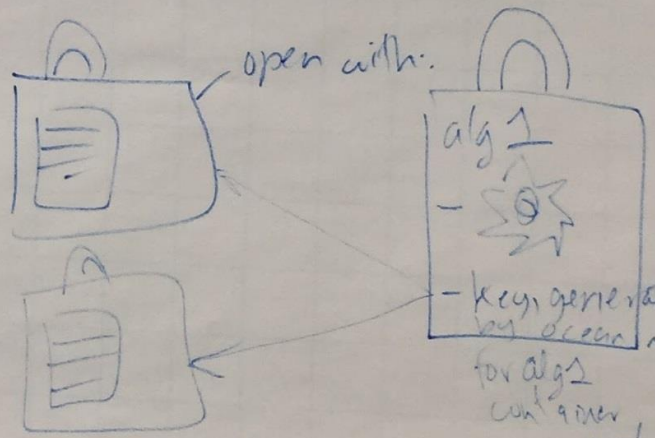

Early design iterations


X=[]; y=[]
newX = 50 initial samples in [0,1]^D
some based on staking
some random
Numerai
For each x_i ( ) in newX
Prep obfuscated data: given D1, D2, ... (dataset 1, 2...)
create O_i (obfuscated data) based on x_i
e.g. O_i = {D2, D3} for x = [0, 1, 1, 0]
Data scientist
For each data scientist
For subset of newX (he chooses)
Build model mapping O_i → trade y_i
Commit stake. It's a proxy for confidence in model + data, and pays for compute cost
Submit model + stake to Numerai
Numerai
Deploy all trading models. Wait 30d.
For each x_i in newX
Compute $ from trading = y_i. Update y_i in newY
X = X ∪ newX, y = Y ∪ newY
Build model if X → y (data choices → value)
Compute impacts of each input in model (ie dataset)
Compute payout to data providers
Compute payout. e.g. to scientists
Plan newX = choose new x_i based on model and/or scientists
freeridi
i.e. ppl using data paying
only the dec. proc. (smart contract) can see the data
low friction high friction price unfair: Hard to set
high friction submit
low friction price
optimally fair/transparent
homomorphic encr.
zkp to trustless computing
intel/ms secure enclaves
latent variable obfuscation / GANs

Goals:

1. Healthy ecosystem (initially) / long term
2. Maintain ethics/values → governance


- Data is open (but obfuscated)
- Avoid need for TTP to see data when
- How to get people compensated w/o freeridi
i.e. incented to contribute
- How to price data
fungible
non-fungible
w/low friction
and fairly based on usefulness of a dataset
fixed
error
auto
low friction
high friction price
high friction segment
low friction price
optimally fair / transparent
Data providers
each data scientist
sends a signal on distributed chain
your private key
stick
How to incent data scientist to accurately reflect their belief in dataset?
Feature engineering on plaintext data
Data scientist creates rel. impact
Belief in value of each dataset
Data scientist chooses % of returns to all data to each dataset
Rel impact
VMs for modeling
MLM on datasets
Scarcity for reuse of dataset. Data scientists paid or stake to use.
Numerai
Deploy all trading models. Wait 30d.
For each Xi in newX
Compute $ from trades = yi. Update yi in newY
X=X∪newX; y=Y∪newY
Build model if X→y
Compute impacts of each input in model (ie dataset
Compute payout to data providers
Compute payout to scientists
Only the doc. proc.
scientists
data
homomorphic encr.
zkp to trustless computing
intel/ms secure enclaves
latent variable obfuscation / GANs
data stakes
50000
10000
10
K8s system for privacy
25% overall

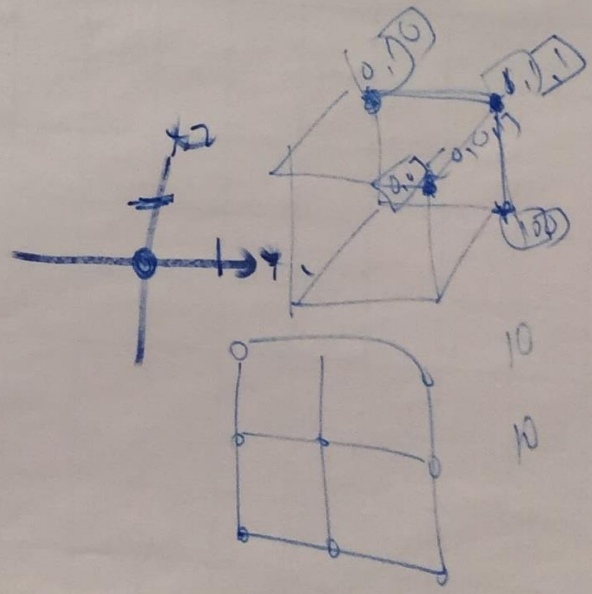


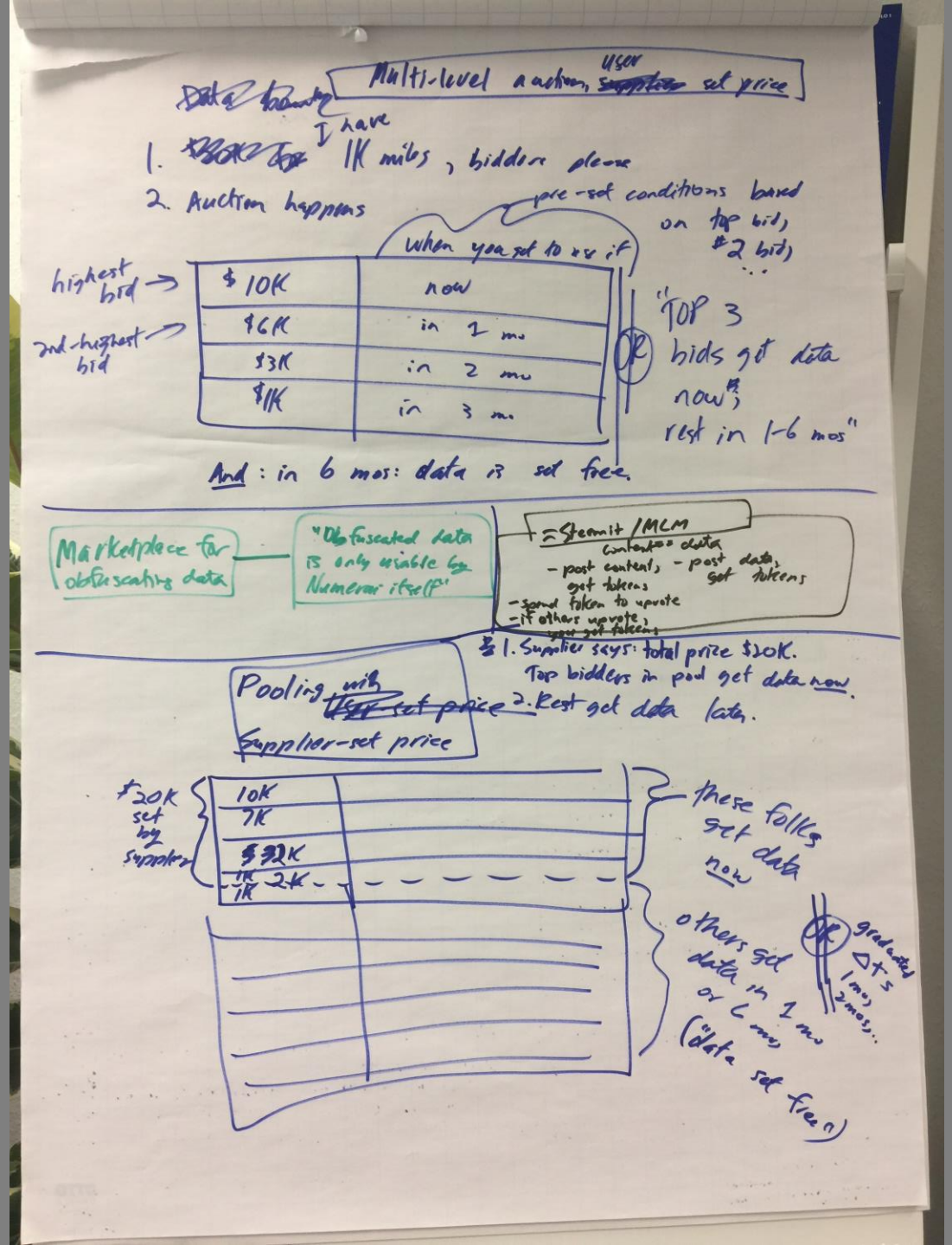
Challenge
- w/o 1.1 : how to ~~pay~~ per dataset
- w/ 1.1 : feels wrong.

0 staked) datasets
models

datasets stake $ to choose which dataset to use

1
2
3
4
5
6
7
8
9




Multi-level auction, user set price
I have
1. 1K mibs, bidders please
2. Auction happens
pre-set conditions based on top bid, #2 bid, ...
when you get to see if
highest bid →
2nd-highest bid →
$10K
$6K
$3K
$1K
now
in 1 mo
in 2 mo
in 3 mo
OR
"Top 3 bids get data now, rest in 1-6 mos"
And: in 6 mos: data is set free.
Marketplace for obfuscating data
"Obfuscated data is only usable by Numerai itself"
Streamit / MCM
- post content, get tokens
- post data, set tokens
- spend token to upvote
- if others upvote, your set tokens
1. Supplier says: total price $20K.
Top bidders in pool get data now.
2. Rest get data later.
Pooling with set price
Supplier-set price
$20K set by supplier
10K
7K
$2K
1K
1K
these folks get data now
others get data in 1 mo or 6 mos ("data set free")
graduated Δt's 1 mo, 2 mos, ...

Engagements / Incentives

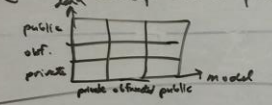
- fix price up-front
- " " "

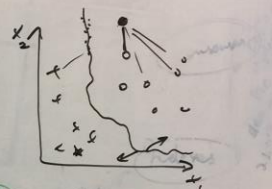
+ reputation (subjective)
  - on data
  - supplier
  - scientist

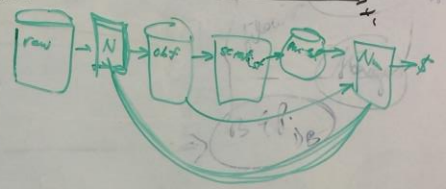
1. You have $OCK staked in a wallet
   - 1/10,000 mining wallet
   - each key is a perceptual hash of a row of data. (HD wallet)
2. Alg to compute a key from data is public
3. ∴ if any data is made public, then anyone can open any wallet kept $OCK.

Stake

model params visible for data supplier
data visible for scientist
data private – model params private







co-owner of $

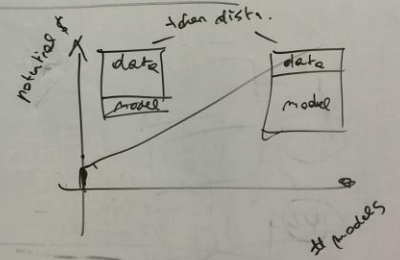
or Mene Markets?

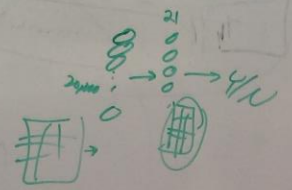
Tokenizing Access to Data Revenue, Fixed Supply

- Each dataset has its own tokens. Fixed supply
- "ICO a dataset"
- When data is purchased for usage, $ is split according to token ownership

Tokenizing Access to Data Itself

- e.g. 100 tokens. You can access the data if you own the token.





Challenges

0. Democratize data.

1. How to ensure supplier gets paid w/o losing ability to get paid in future. "Free riding"
   "Privacy"
   "Copy vs title"

2. Friction in pricing (How to price)
   - overall price
   - relative impact per dataset

Static ⟷ Dynamic dataset (goes stale) | Fungible ⟷ Non fungible (promote via liquidity/size ↓ or Bancor pooling)

- data labeling service
- data obfuscation market

Signals

- stake in belief of value of dataset
  - supplier
  - .... by data scientist
- price asked by supplier
- price bid by scientist
- reputation of dataset
- .... supplier
- .... scientist
- bounties for free riding detection
- $ value gained from dataset(s) in RL (eg Numerai & models; insurance savings)
- total value of network ⟷ value per dataset
- prediction market belief in value of dataset
- .... novelty of a dataset
- take a price of dataset
- scarcity
- demand

Tools to address free riding

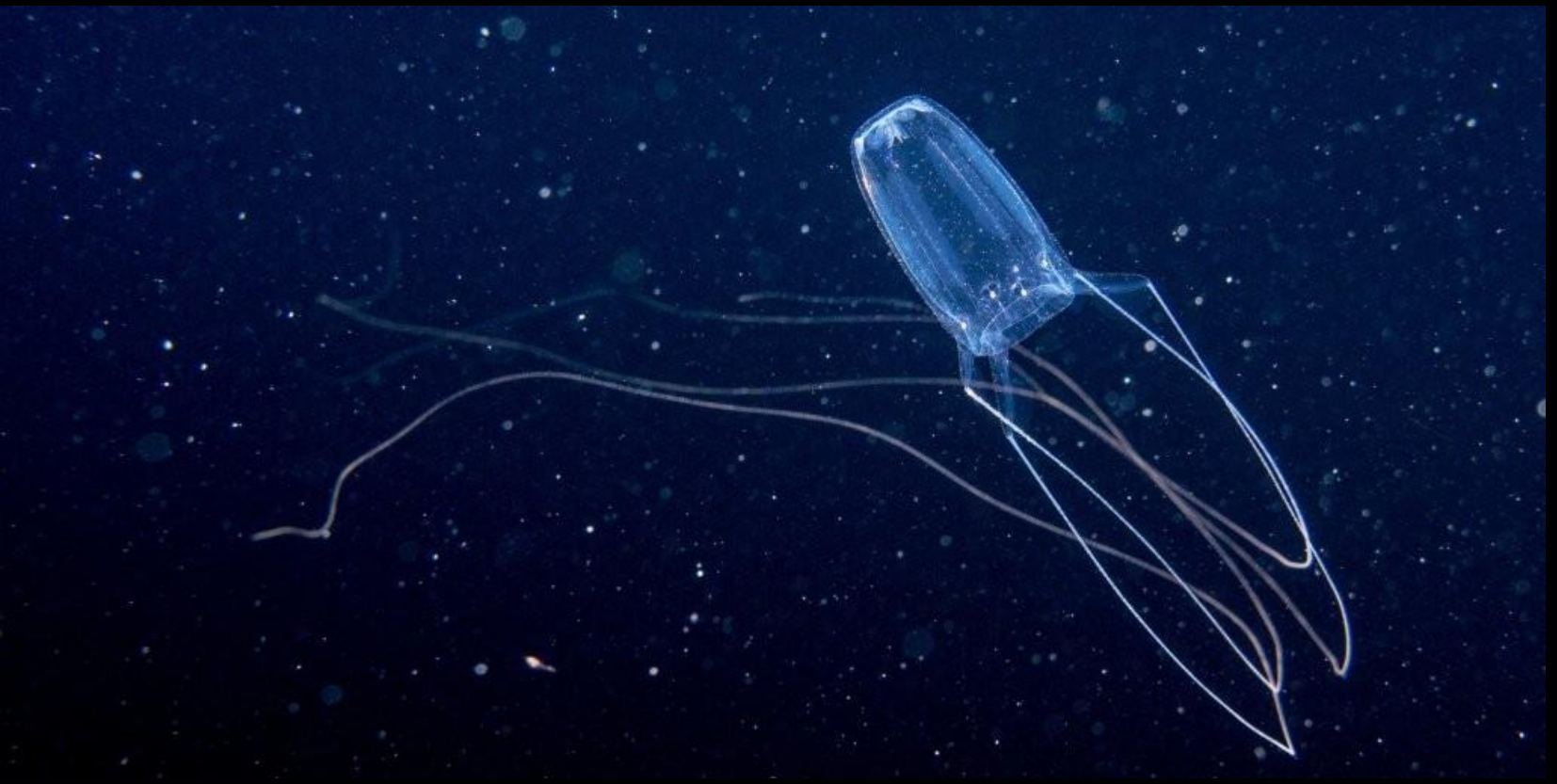
- "Set the free" after Δt ⟹ stream to own
- Data provider doesn't care
- Licensing
- Prove opt-in GDPR
- watermarking
- reputation
- provenance
- risk of litigation

(combine)

- only the smart contract can see the data. Eg docker + locks, MPC, ZKP ⇕
- If data set free, you private key gets exposed w/ stake of $$$
- Data obfuscation, eg latent variables on NN (like Numerai) ([illegible] eg GANs)

ocean


Lessons from early iterations, towards an improved method



Early iterations: *Flailing*

Can we structure this better?

ocean

Realization:
Tokenized Ecosystems
Are a Lot Like Evolutionary Algorithms!

What		Tokenized ecosystem		Evolutionary Algorithm
Goals		Block reward function E.g. “Maximize hash rate”		Objective function E.g. “Minimize error”
Measurement & test		Proof E.g. “Proof of Work”		Evaluate fitness E.g. “Simulate circuit”
System agents		Miners & token holders (humans) In a network		Individuals (computer agents) In a population
System clock		Block reward interval		Generation
Incentives & Disincentives		You can’t control human, Just reward: give tokens And punish: slash stake		You can’t control individual, Just reward: reproduce And punish: kill

ocean

We can approach token design as optimization design.

How do you do Optimization Design??

Steps in Optimization Design

1. **Formulate the problem.** Objectives, constraints, design space.
2. **Try an existing solver.** If needed, try different problem formulations or solvers.
3. **Design new solver?**

1. Formulation of an optimization problem

Objectives & constraints in a design space

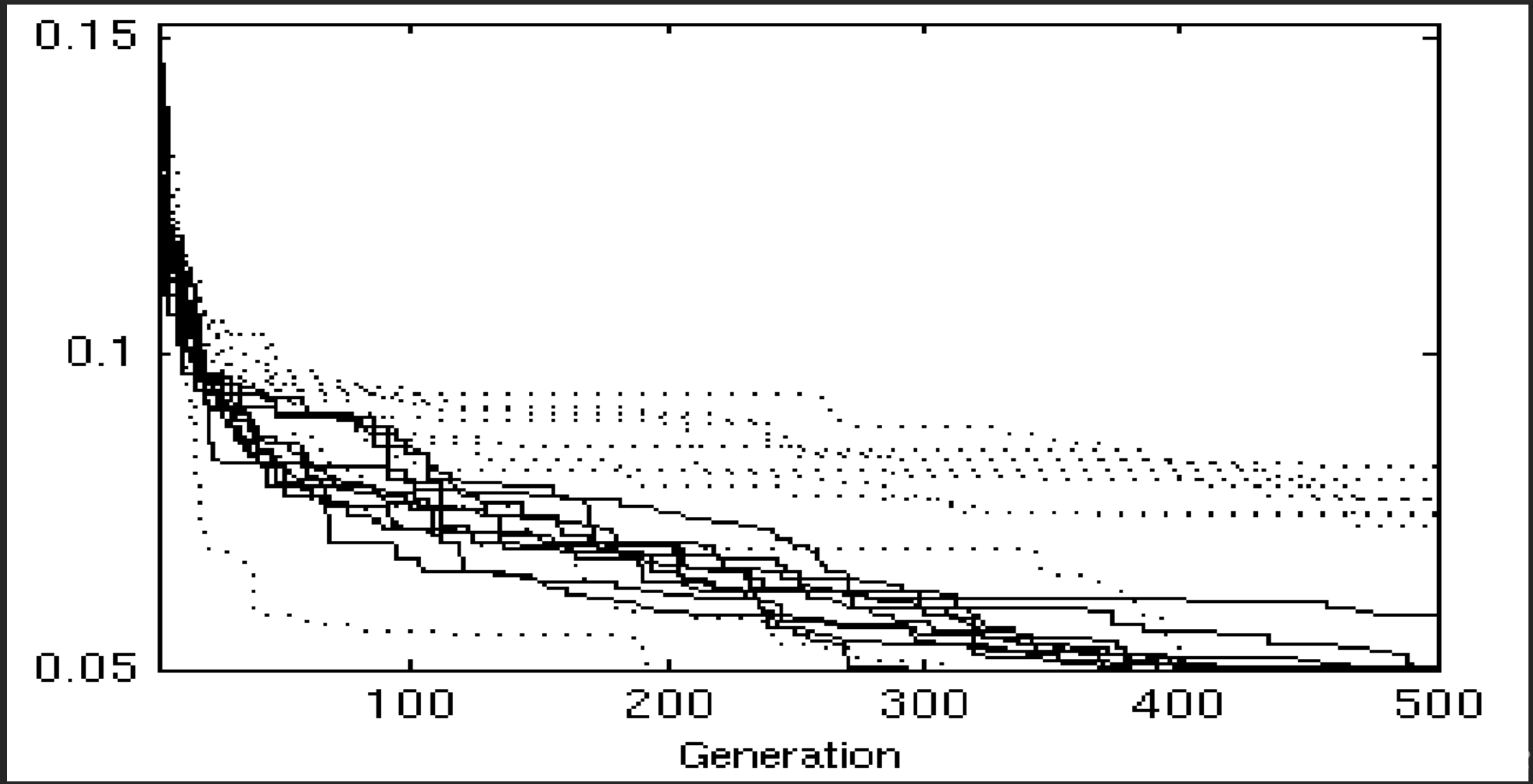
The algorithm's aim is formulated as a constrained multi-objective optimization problem

$$\begin{aligned} \text{minimize} \quad & f_i(\phi) && i = 1 \ldots N_f \\ \text{s.t.} \quad & g_j(\phi) \leq 0 && j = 1 \ldots N_g \\ & h_k(\phi) = 0 && k = 1 \ldots N_h \\ & \phi \in \Phi \end{aligned} \tag{1}$$

where Φ is the "general" space of possible topologies and sizings. The algorithm traverses Φ to return a Pareto-optimal

ocean

2. Try an existing solver. Does it converge?



3. Design new solver

e homo-
motopy
coarsely
ructural
y. Tradi-
ro path,
the zero
several

mulated
nalyses,
oint $\boldsymbol{\theta}$}.
nt/other
onnom-
corners
rated in
on (with

TABLE II
PROCEDURE SANGRIAOPTIMIZATION()

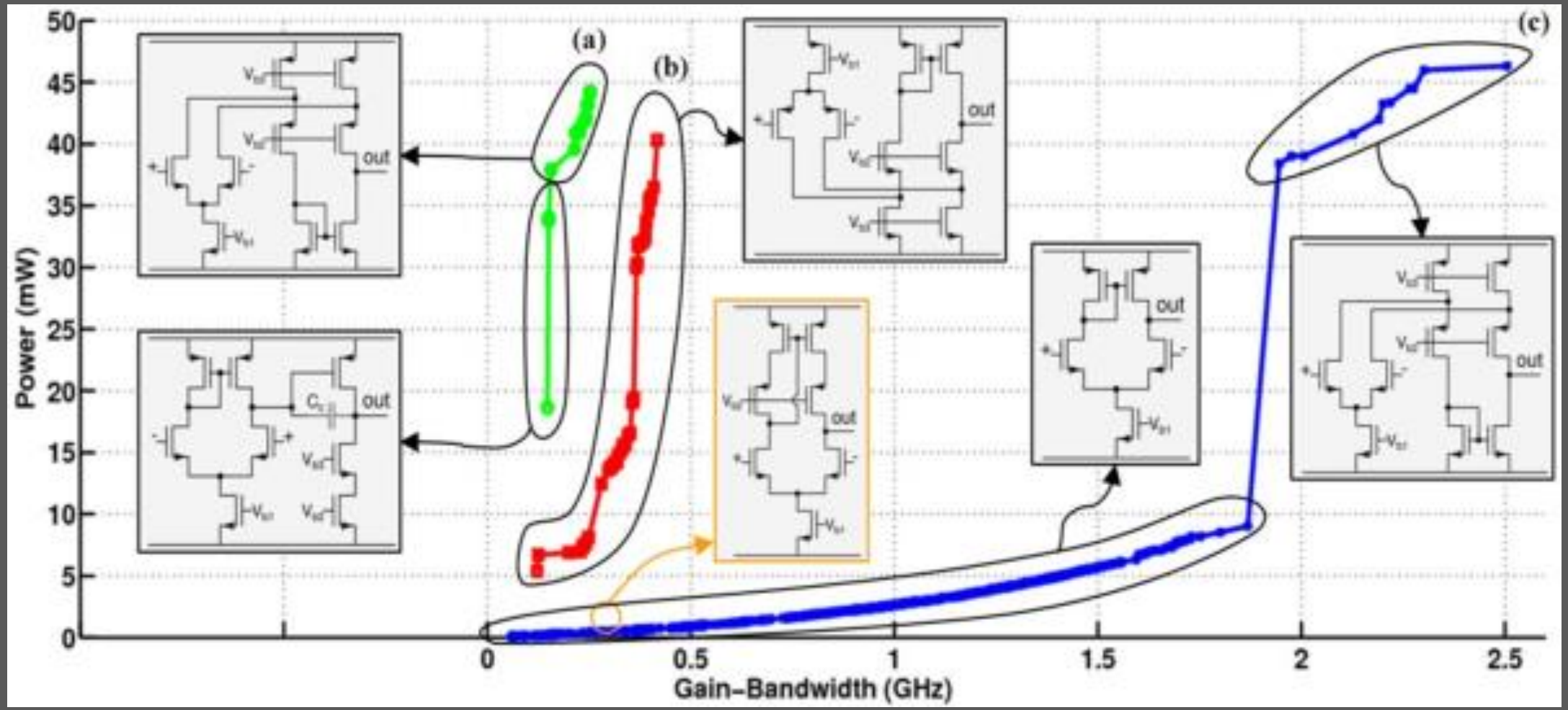
Inputs: D, N_a, K, $N_L(k)$
Outputs: $\boldsymbol{d}^*$

1. $N_{gen} = 0$; $P = \emptyset, P_{all} = \emptyset$
2. while stop() $\neq True$:
3. if $(N_{gen} \% N_a) = 0$:
4. if $|P| < K$:
5. $P_{|P|+1} = \emptyset$
6. P_0 = SpaceFillIndividuals$(N_L(k), N_D, D)$
7. for $k = 1$ to $|P|$:
8. P_k = SelectParents$(P_k, P_{k-1}, N_L(k))$
9. $P_{k,j}$ = UpdateLocalOptState$(P_{k,j}, k)$, $j = 1$ to $|P_k|$
10. P_{all} = unique$(P_{all} \cup P)$
11. $P_{|P|} = P_{|P|} \cup$ InnerOptimize(P_{all}, D, k)
12. $\boldsymbol{d}^* = \boldsymbol{d_i}$ in P_{all} with highest Y or Cpk
13. $N_{gen} = N_{gen} + 1$
14. return $\boldsymbol{d}^*$

and all individuals encountered so far in the search, P_{all}.
Lines 2–13 are the generational loop, which repeats until stop

ocean

Example of a Successful Outcome



ocean

Token Design as Optimization Design

ocean

Steps in *Token* Design

1. **Formulate the problem.** Objectives, constraints, design space.
2. **Try an existing pattern.** If needed, try different formulations or solvers.
3. **Design new pattern?**

1. Formulate the Problem

(a) Ask

- Who are my potential **stakeholders**?
- And what do each of them **want**?
- What are possible **attack vectors**?

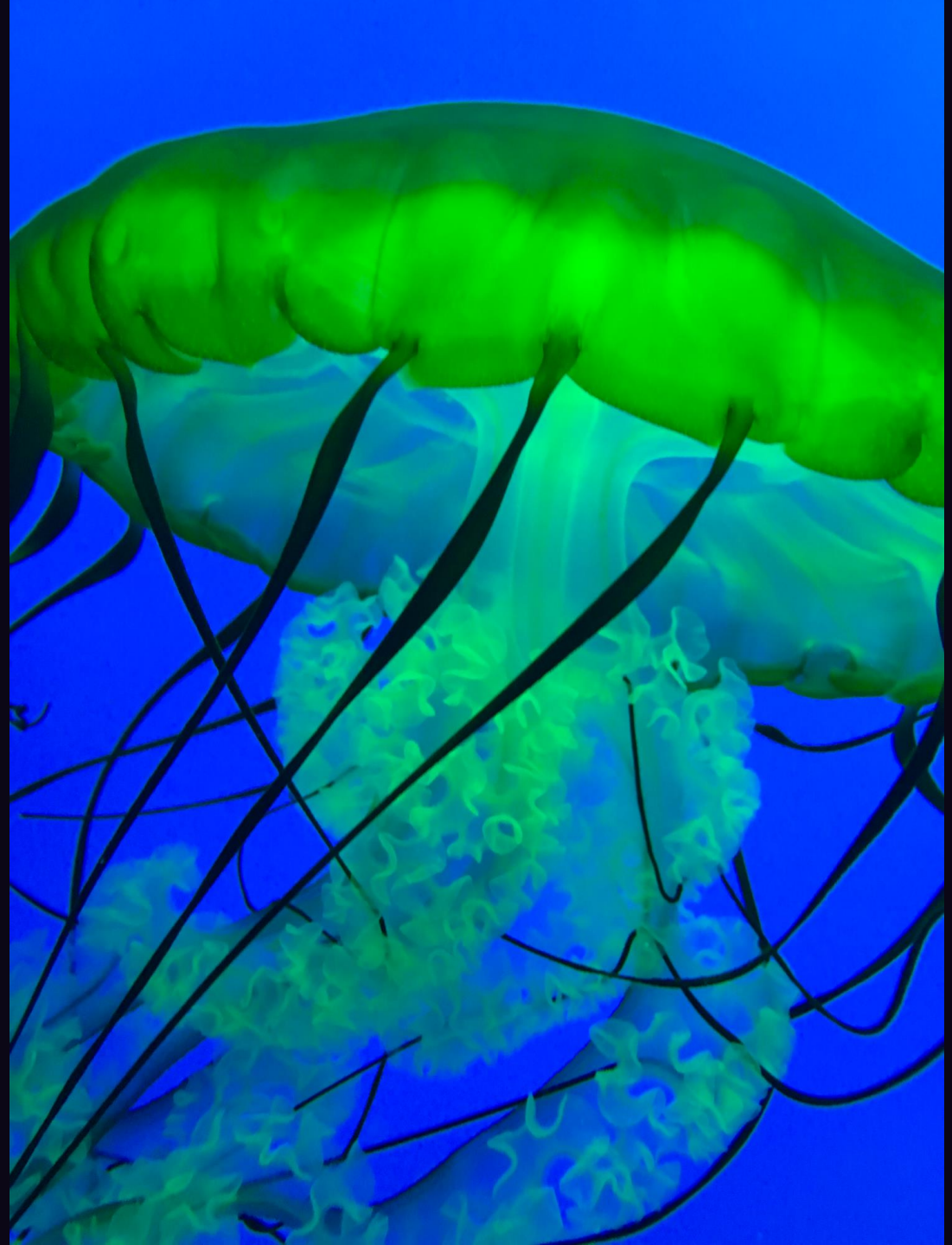
(b) Translate those into objectives and constraints.

2. Try Existing Patterns

1. **Curation**
2. **Proofs of human or compute work**
3. Identity
4. Reputation
5. Governance / software updates
6. Third-party arbitration
7. …

2.1 Patterns for Curation

- **Binary** membership: Token Curated Registry (TCR)
- **Discrete-valued** membership: Stake Machines
- **Continuous-valued** membership: Curation Markets characterized by bonding curve
- **Hierarchical** membership: each label gets a TCR
- **Work** tied to membership: Proofed Curation Market
- **Non-fungible** tokens: Re-Fungible Tokens



Case Study: Analysis of Bitcoin

ocean

Bitcoin objective function

Objective: Maximize security of network

- **Where “security” = compute power**
- **Therefore, super expensive to roll back changes to the transaction log**

ocean

Bitcoin objective function

Objective: Maximize security of network

- **Where "security" = compute power**
- **Therefore, super expensive to roll back changes to the transaction log**

$$E(R_i) \ \alpha \ H_i * T$$

E() = expected value

block rewards

hash power of actor = contribution to "security"

 # tokens (BTC) dispensed each block

ocean

Result of Bitcoin's objective function:

People are maximizing security! = Maximizing electricity

More power than USA by mid 2019

Case Study: Design of Ocean

ocean

1. Formulate the Problem:
(a) Who are stakeholders? What do they want?

Key stakeholders in Ocean ecosystem

Stakeholder	What value they can provide	What they might get in return
Data/service provider, data custodian, data owner	Data/service (market's supply)	Tokens for making available / providing service
Data/service referrers, curators. Includes exchanges and other application-layer providers.	Data/service (via a provider etc), curation	Tokens for curating
Data/service verifier. Includes resolution of linked proofs on other chains	Data/service (via a provider etc), verification	Tokens for verification
Data/service consumer	Tokens	Data/service (market's demand)
Keepers	Correctly run nodes in network	Tokens for chainkeeping

ocean

1. Formulate the problem:
(b) Translate into objectives and constraints

Objective function: maximize supply of relevant data

Token rewards if: supply relevant data
Token rewards if: supply data, and curate it

ocean

1. Formulate the problem:
(b) Translate into objectives & constraints

Constraints = checklist:

- For priced data, is there incentive for supplying more? Referring?
- For priced data, good spam prevention?
- For free data, is there incentive for supplying more? Referring?
- For free data, good spam prevention?
- Does the token give higher marginal value to users of the network versus external investors? Eg Does return on capital increase as stake increases?
- Are people incentivized to run keepers?
- Is it simple? Is onboarding low-friction?

2. Try Existing Patterns
Some patterns:

1. Actor registry
2. Data registry
3. Actor registry + data registry
4. Data registry + free-as-in-beer data curation market.
   Curation: Pay tokens to listen.

2. Try existing patterns: evaluate on objectives & constraints. None passed…

Key Question	1	2	3	4
For priced data: incentive for supplying more? Referring?	✗	≈	✓	≈
For priced data: good spam prevention?	≈	✓	✓	✓
For free data: incentive for supplying more? Referring?	✗	≈	✗	✓
For free data: good spam prevention?	≈	✓	≈	✓
Does token give higher marginal value to users of the network, vs external investors? Eg Does return on capital increase as stake increases?	✓	✓	✓	✓
Are people incentivized to run keepers?	≈	≈	✓	✓
It simple? Is onboarding low-friction? Where possible, do we use incentives/crypto rather than legal recourse?	✓	✓	≈	≈

3. Try New Patterns
Some patterns:

1. Actor registry
2. Data registry
3. Actor registry + data registry
4. Data registry + free-as-in-beer data curation market. Curation: Pay tokens to listen.
5. **Data registry + free data curation market. Curation: Stake tokens as belief in reputation. Auto CDN.**
6. **Actor registry + free&priced data curation market. Curation: Stake tokens as belief in reputation. Auto CDN. "Proofed Curation Market"**

ocean

3. Try new patterns: evaluate on objectives & constraints

Key Question	1	2	3	4	5	6
For priced data: incentive for supplying more? Referring?	✗	≈	✓	≈	≈	✓
For priced data: good spam prevention?	≈	✓	✓	✓	✓	✓
For free data: incentive for supplying more? Referring?	✗	≈	✗	✓	✓	✓
For free data: good spam prevention?	≈	✓	≈	✓	≈	✓
Does token give higher marginal value to users of the network, vs external investors? Eg Does return on capital increase as stake increases?	✓	✓	✓	✓	✓	✓
Are people incentivized to run keepers?	≈	≈	✓	✓	✓	✓
It simple? Is onboarding low-friction? Where possible, do we use incentives/crypto rather than legal recourse?	✓	✓	≈	≈	✓	✓

ocean

Objective: maximize supply of relevant data

ocean

- **Reward curating data (staking on it) + making it available**
- **New pattern: Proofed Curation Market**

$$E(R_{ij}) \alpha \ \log10(S_{ij}) * \log10(D_j) * T * R_i$$

Expected reward for user *i* on dataset *j*

S_{ij} = predicted popularity = user's curation market stake in dataset *j*

D_j = proofed popularity = # times made dataset available

 # tokens during interval

ocean


Token Engineering

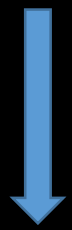
Design of Tokenized Ecosystems
From Mechanism Design to *Token Engineering*

Analysis:

Game theory

Synthesis:

Mechanism Design



Practical constraints

Optimization Design

ocean

Design of Tokenized Ecosystems
From Mechanism Design to *Token Engineering*

Analysis:

Game theory

Synthesis:

Mechanism Design

Practical constraints

Optimization Design

Engineering theory, practice and tools + responsibility

Token Engineering for Analysis & Synthesis

ocean

Engineering 

is the creative application of science, mathematical methods, and empirical evidence

to the innovation, design, construction, operation and maintenance

of structures, machines, materials, devices, systems, processes, and organizations.

ocean

Engineering Responsibility

ocean

Engineering has

Theory,
Practice,
Tools,
Responsibility

- Electrons : Electrical Engineering
- Tokens : Token Engineering

ocean

Science ←→ Engineering

- **Engineering** is about building things that work.
- **Science** is about contributing new knowledge.
- They're complementary.

Therefore **token engineering** is complementary to the science of cryptoeconomics / **token economics**.

ocean


Emergence of TE Community

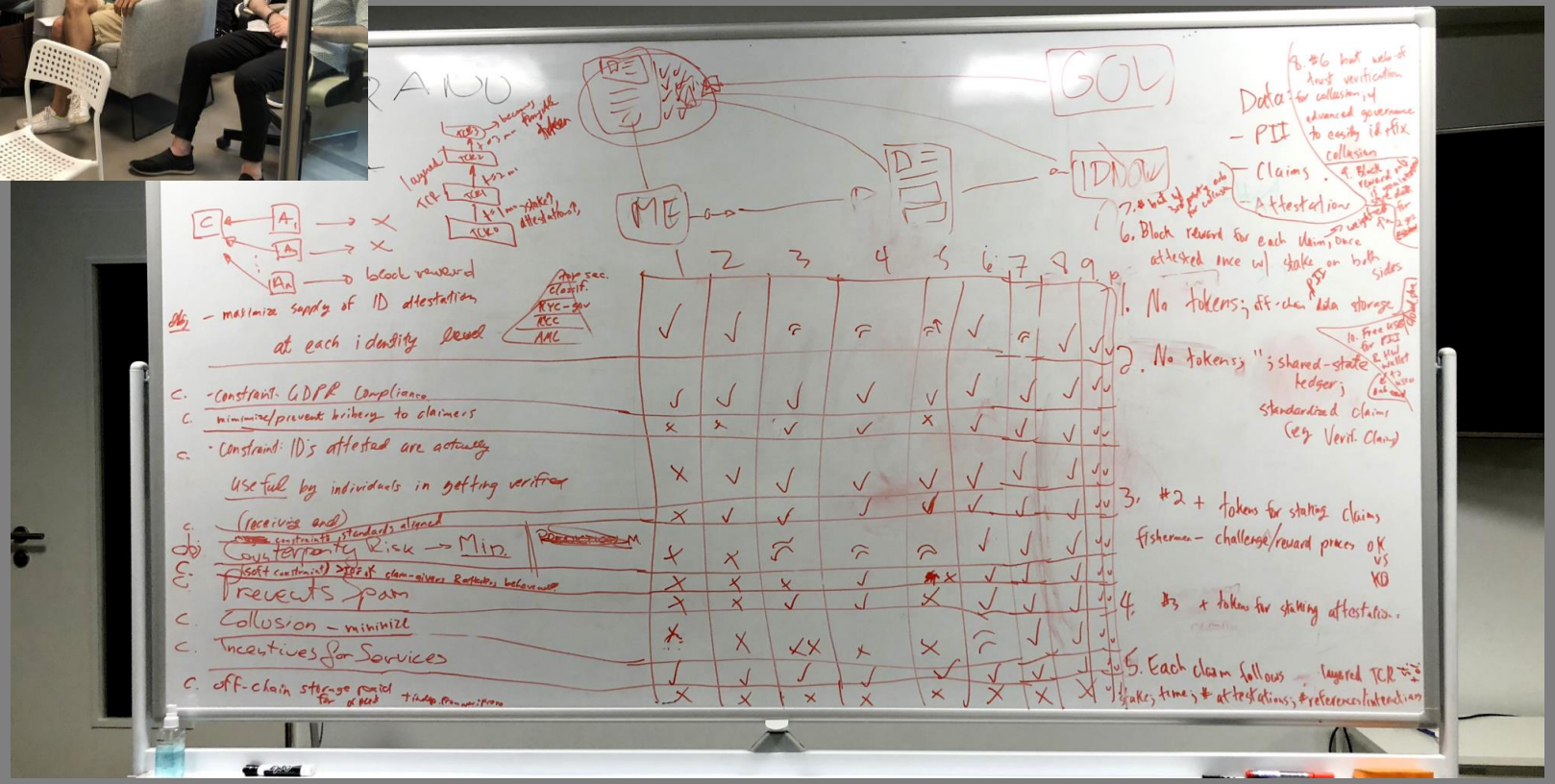
TE → TE Community

- A pleasant surprise to me: “Token Engineering” resonated with a *lot* of people
- And many new connections for me.
- Many amazing conversations.
- **A collective realization: we need to share knowledge, to learn from each other!**

1st TE event
Berlin, Apr 2, 2018

ocean

More TE Meetups!



TE Local Meetup Groups

(The actual meetup.com pages will ty

- TE Amsterdam
- TE Berlin
- TE Budapest
- TE Hong Kong
- TE London
- TE Munich
- TE NYC
- TE Stockholm
- TE San Francisco
- TE St Petersburg
- TE Sydney
- TE Tokyo
- TE Toronto
- TE Vancouver
- TE Vienna
- TE Zurich/Zug

ocean

Mission of the TE Community

To grow TE into an **engineering discipline**

collectively as a community

in a decentralized, permissionless, open-source fashion that all can contribute to and all can use.


Deployment / Verification
of Design

Token Verification

1. **Human-based**
   - Share docs / whitepaper, get feedback
   - Role-playing in a group game. E.g. Cadence / Joe Costello
   - Board games. Think Magic the Gathering
2. **Software-based**
   - Simulation. E.g. CadCad
   - Verification. E.g. formal verification tool in Remix
3. **Economic**
   - Release the software / network with ever-increasing skin-in-the-game. E.g. Ocean ramp-up network rewards.
   - Bounties++ over time. Eg Cosmos, DutchX
   - Testnets with value. Eg Polkadot
   - Blockchains as built-in bug-bounty systems. E.g. Bitcoin

ocean

Token Re-Engineering is OK!

Example: Synthetix

1. Initial Design : no transaction
2. New design: #2 DeFi app within 6 mos.


Learnings since initial design / deployment / verification

Company Startups vs. Ecosystem Startups

- 50% of the lessons building a startup apply to tokenized ecosystems.

Company Startups vs. Ecosystem Startups

- 50% of the lessons building a startup apply to tokenized ecosystems.
- But we don’t know which 50% (!)

Company Startups vs. Ecosystem Startups

- 50% of the lessons building a startup apply to tokenized ecosystems.
- But we don’t know which 50% (!)
- Actually, not quite true, we’re starting to build a theory of it…

Startup Knowledge Sources

- Incubators like YES! Delft
- Literature:
  - Paul Graham essays
  - The Lean Startup
  - Zero to One
  - …
- … and a million more things now!

Guideline: Order of Operations

Advice from Reid Hoffman (LinkedIn) in BlitzScaling:

Order of operations in bringing a product to scale

1. Prototype (incl token design)
2. Product
3. Distribution (incl virality)
4. Monetization
5. Scale-up

Advice from CZ (Binance):

Only put $ into distribution / marketing once you have product-market fit. (Summary of above!)

ocean

Guideline: Use Platform Startup Tricks

- **Constraint: ecosystem = multi-sided platform. So use startup guidelines for those!**
- It's a chicken and egg (empty network) problem.
- How to solve? Various tricks
  - Trick 1: leverage a pre-loaded ecosystem
  - Trick 2: one sub-network at a time, launched aggressively. E.g. FB per university, Uber per city
  - Trick 3: something that's valuable with just 1 actor. E.g. Instagram filters, LinkedIn resumes.

Guideline: Biz Models

- **Constraint: your system must be open, with $$ going to community.**
  - **Can't rely on data moats!**
  - **Yet you still need to eat**
- **Solution: Use an emerging "Web3 Business model". Funding DAOs** are a more extreme-sounding version, which many are trending towards:
  - Grants $ from network rewards, investment $ from investors
  - Community members propose projects
  - Community curates which projects receive $
  - Coordinated in a decentralized community that manages $ (i.e. a DAO).
  - The entity has a legal basis (a LAO!)
- Examples:
  - Grants: DASH, Decred, MolochDAO
  - Venture: Metacartel Ventures, The DAO 2.0
- **A good ol' Web2 biz model may be fine too! E.g. tx fees** like CryptoKitties.

ocean

Guideline: Pet Idea Syndrome

- **Shorten the time to idea, lest you become too attached to it.**
  - This flies in the face of “whitepapers”, which were needed for ICOs
  - But we’re past the ICO era!
  - The new / better reality: create value, *then* (maybe) ask for $
- Carmack’s extreme-but-cool way: try your idea the day you think of it
  - (Which means set up your environment such that you can)

Conclusion

Conclusion / Summary of Learnings

- It’s not just “design”. It’s also: build, verify, deploy, maintain, improve (re-engineer).
- Don’t invent something new if you don’t need to
- Exploit knowledge & processes from other fields
  - Optimization, SW engineering
  - Startups, go-to-market

Trent McConaghy @trentmc0
Ocean Protocol @oceanprotocol

ocean