Towards a Practice of Token Engineering

Trent McConaghy
@trentmc0

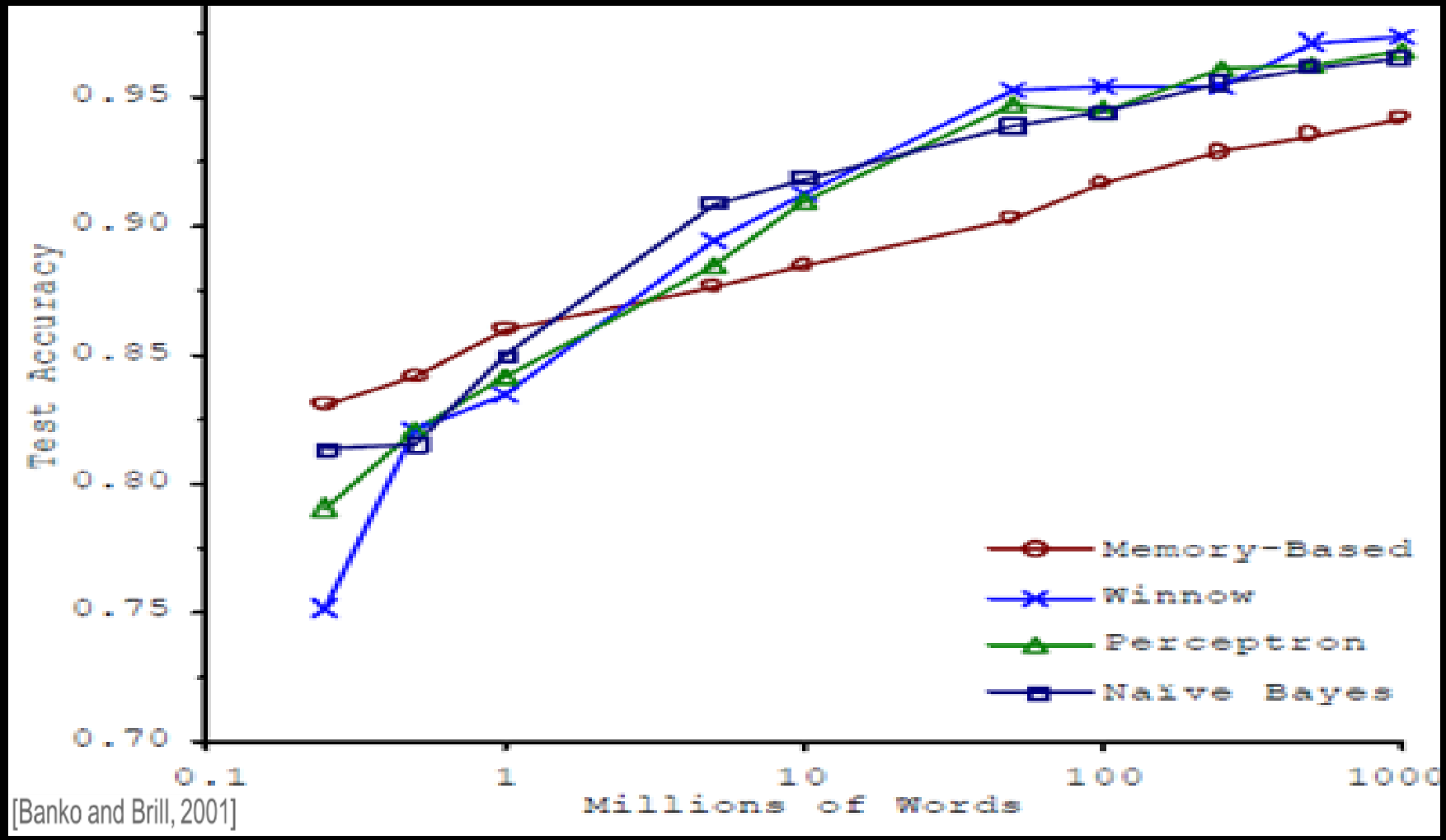
ocean

#Data
#Incentives

Audio radar

ocean

The Unreasonable Effectiveness of Data



1000% less error!

1000x *more* data

Silo mo' data

↓

Mo' accuracy

↓

Mo' $

Default incentive: hoard the data

“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!

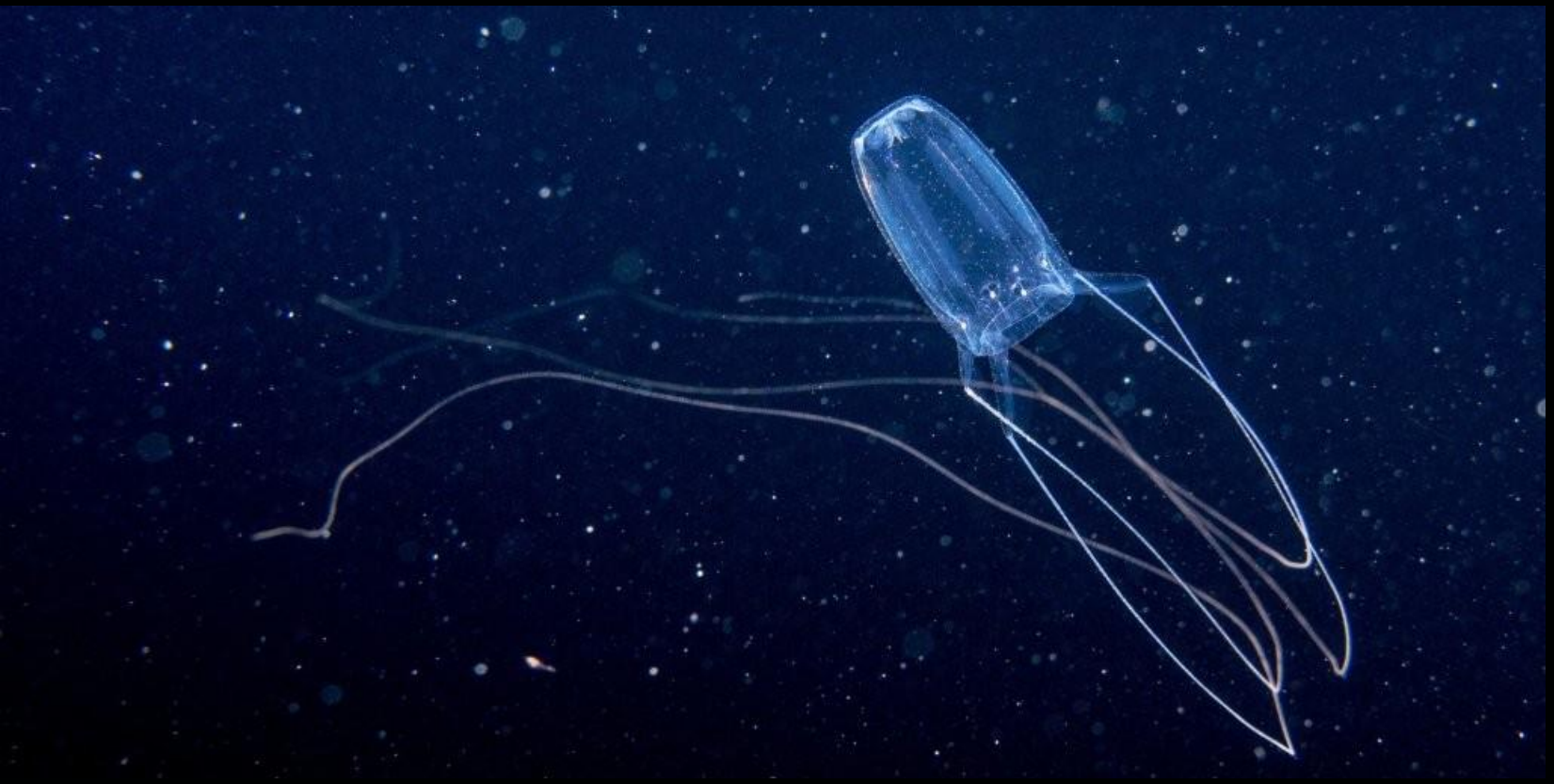
~~Silo~~ *Pool* mo' data

⬇

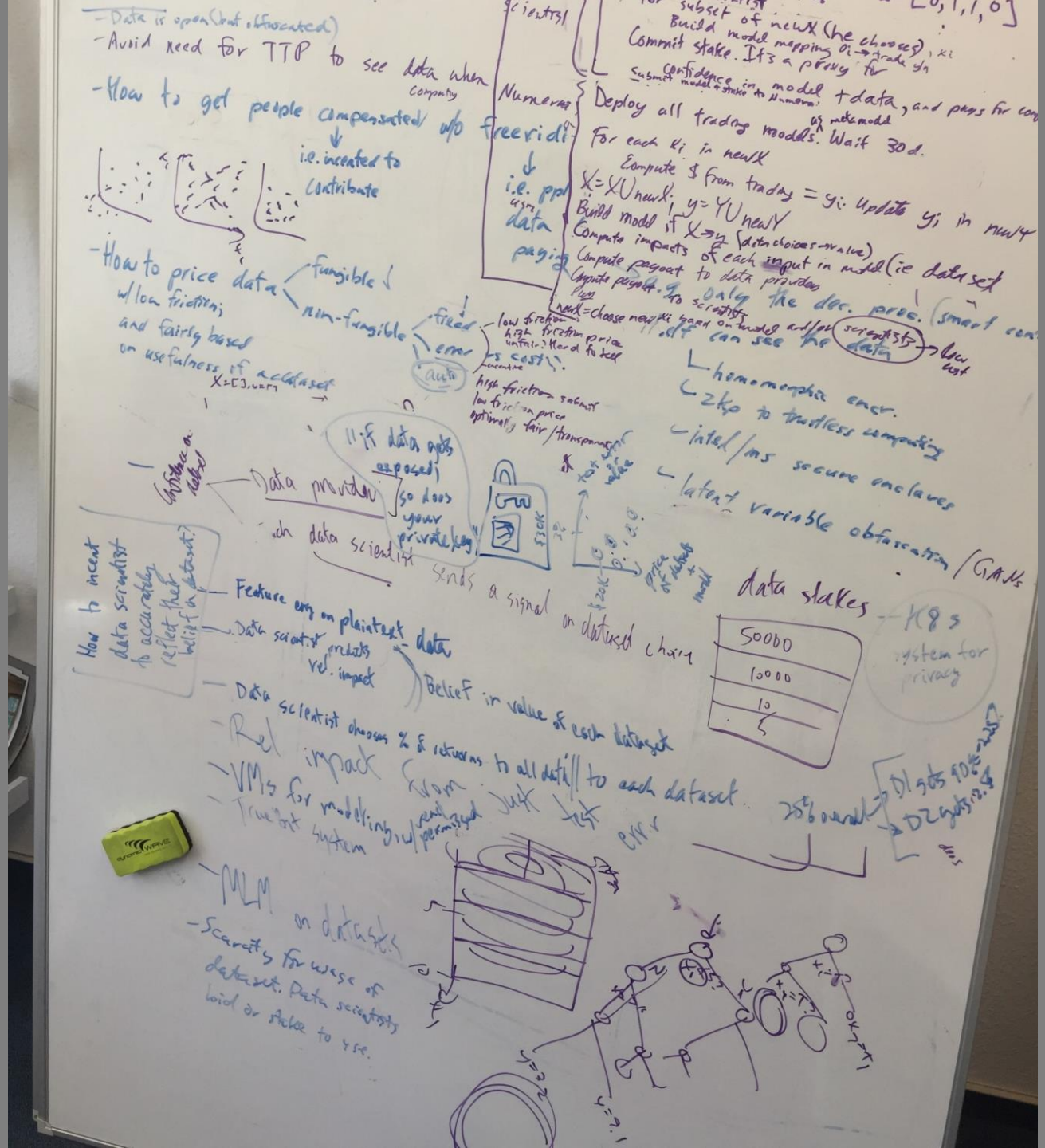
Mo' accuracy

⬇

Mo' $



Early iterations


- 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
Numerai
Deploy all trading models. Wait 30d.
For each ki in newX
Compute $ from trades = yi. Update yi in newY
Build model if X→y
Compute impacts of each input in model (ie dataset
Compute payout to data providers
Data providers
each data scientist
sends a signal
on distributed chain
Feature eng on plaintext data
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.
data stakes
50000
10000
10
K8s system for privacy
homomorphic encr.
zkp to trustless computing
intel/ms secure enclaves
latent variable obfuscation / GANs

Engagements / Incentives

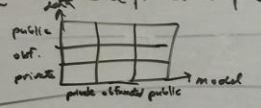
- fix price up-front
- " " "

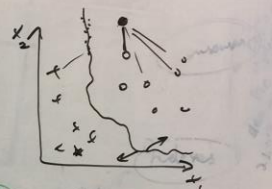
+ reputation (subjective)
  - on data / supplier / scientist

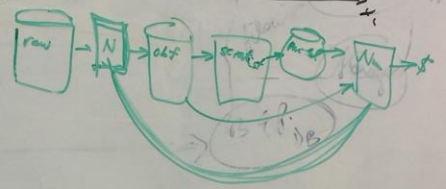
Stake

1. You have $STK staked in a wallet
   - 1/10,000 ownership
   - 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 $STK

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







co-owner of $

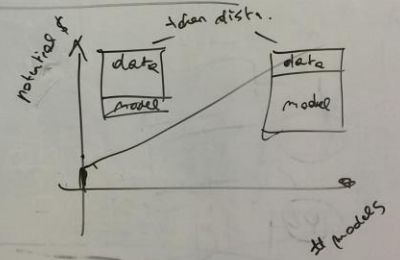
or Mene Markets?

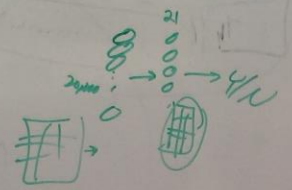
Tokenizing Access to Data Revenue, Fixed Supply

- 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.





markets

data → $ distance

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 use it
highest bid → $10K	now
2nd-highest bid → $6K	in 1 mo
$3K	in 2 mo
$1K	in 3 mo

"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"

Streamlit / MCM

- post content, get tokens
- post data, set tokens
- spend token to upvote
- if others upvote, your get tokens

1. Supplier says: total price $20K. Top bidders in pool get data now.
2. Rest get data later.

Pooling with ~~User~~ 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, ...)

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)

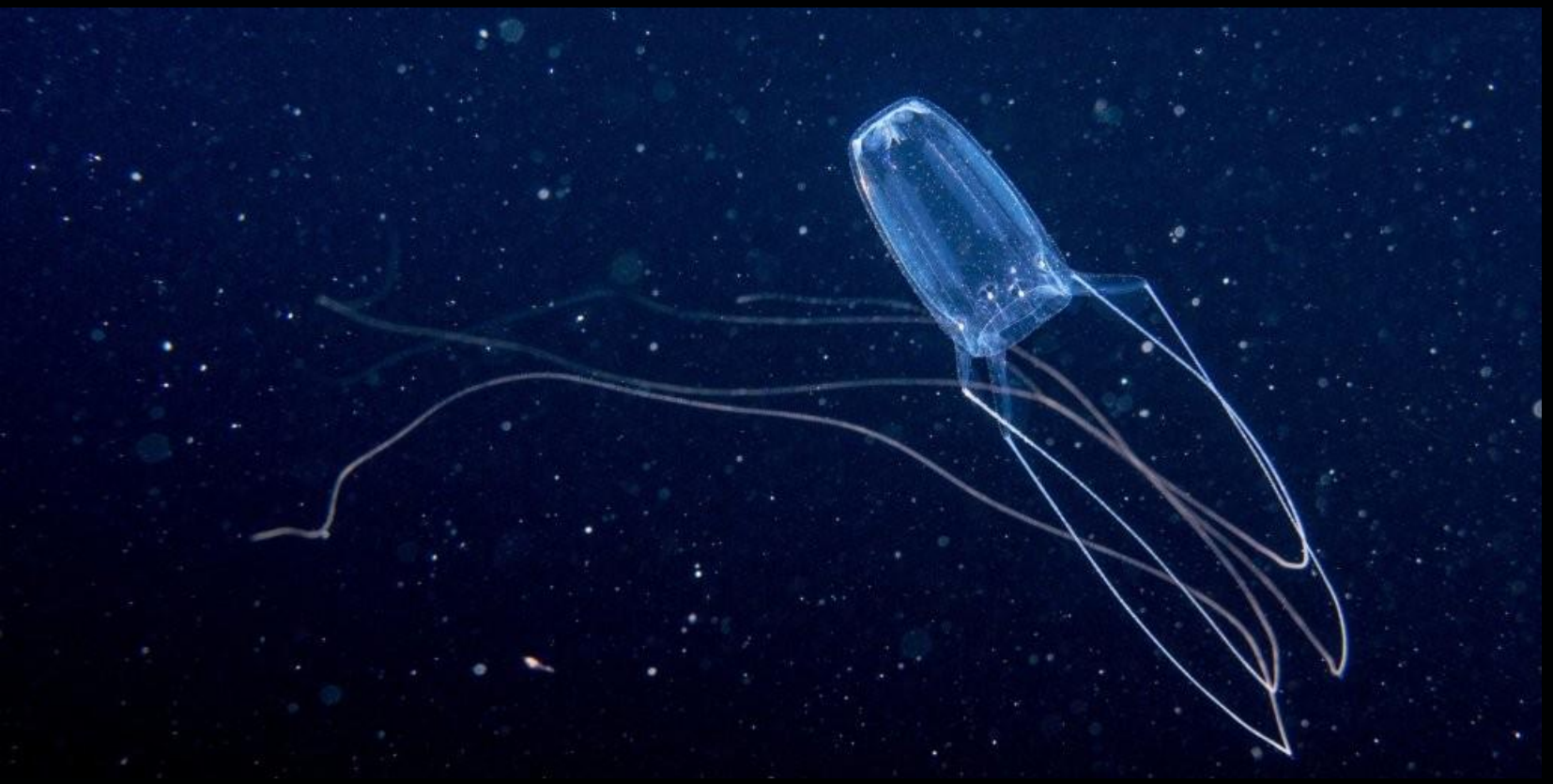
Signals

- data labeling service
- data obfuscation market

- 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 & medical; 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 gdpr [illegible] 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)
  (similar [illegible] GANs)



Early iterations: *Flailing*

Can we structure this better?

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

We can approach token design as optimization design.

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

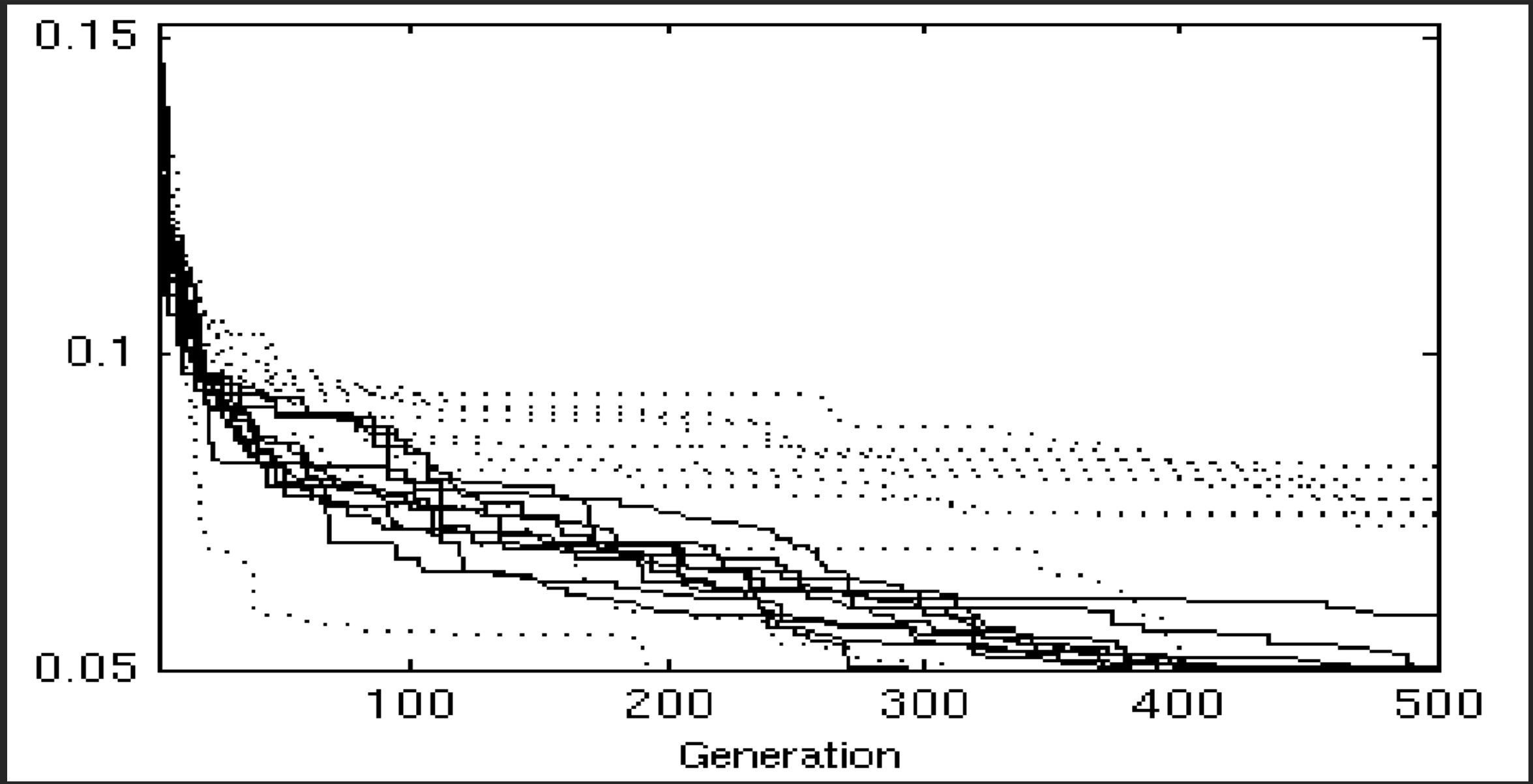
Objectives & constraints in a design space

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

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

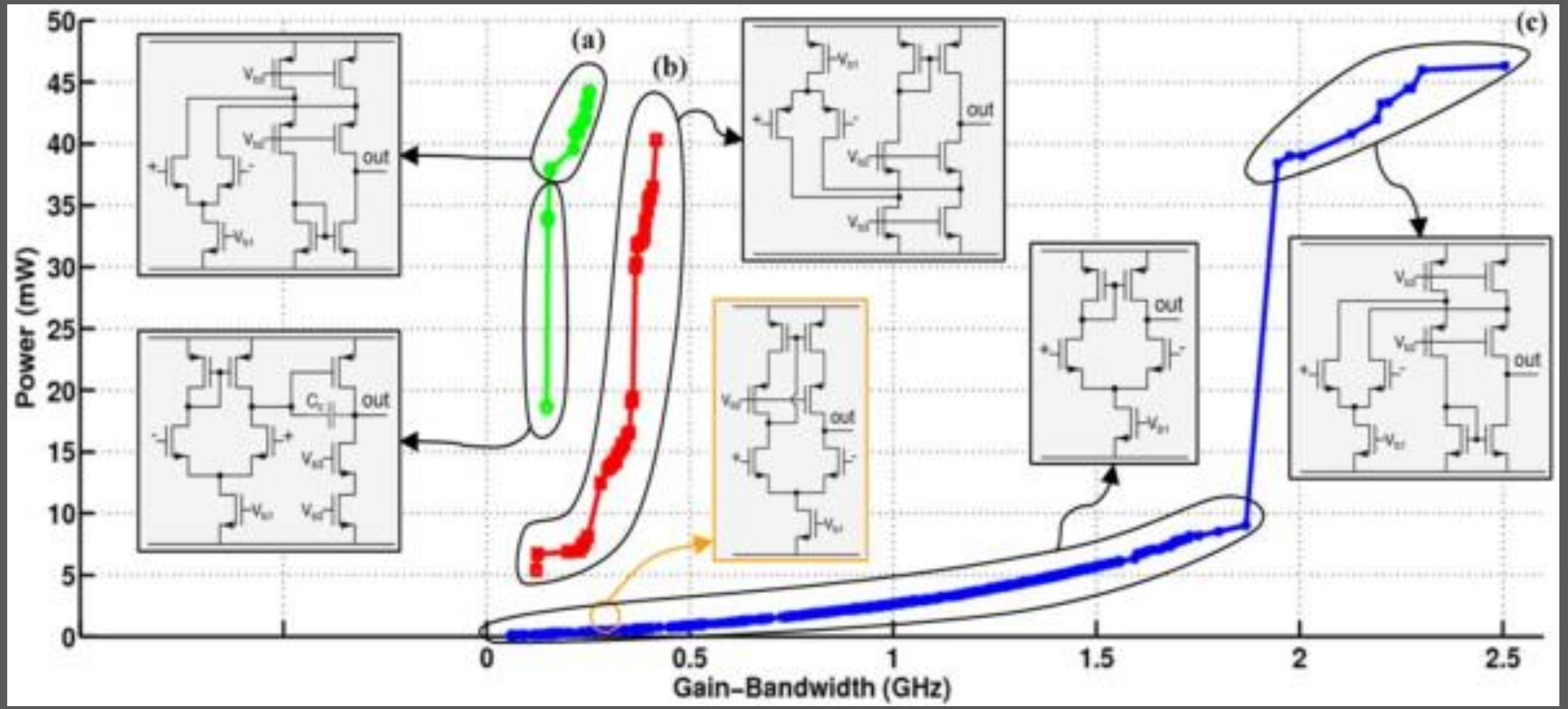
TABLE II
PROCEDURE SANGRIAOPTIMIZATION()

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

Example of a Successful Outcome



Token Design as Optimization Design

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.

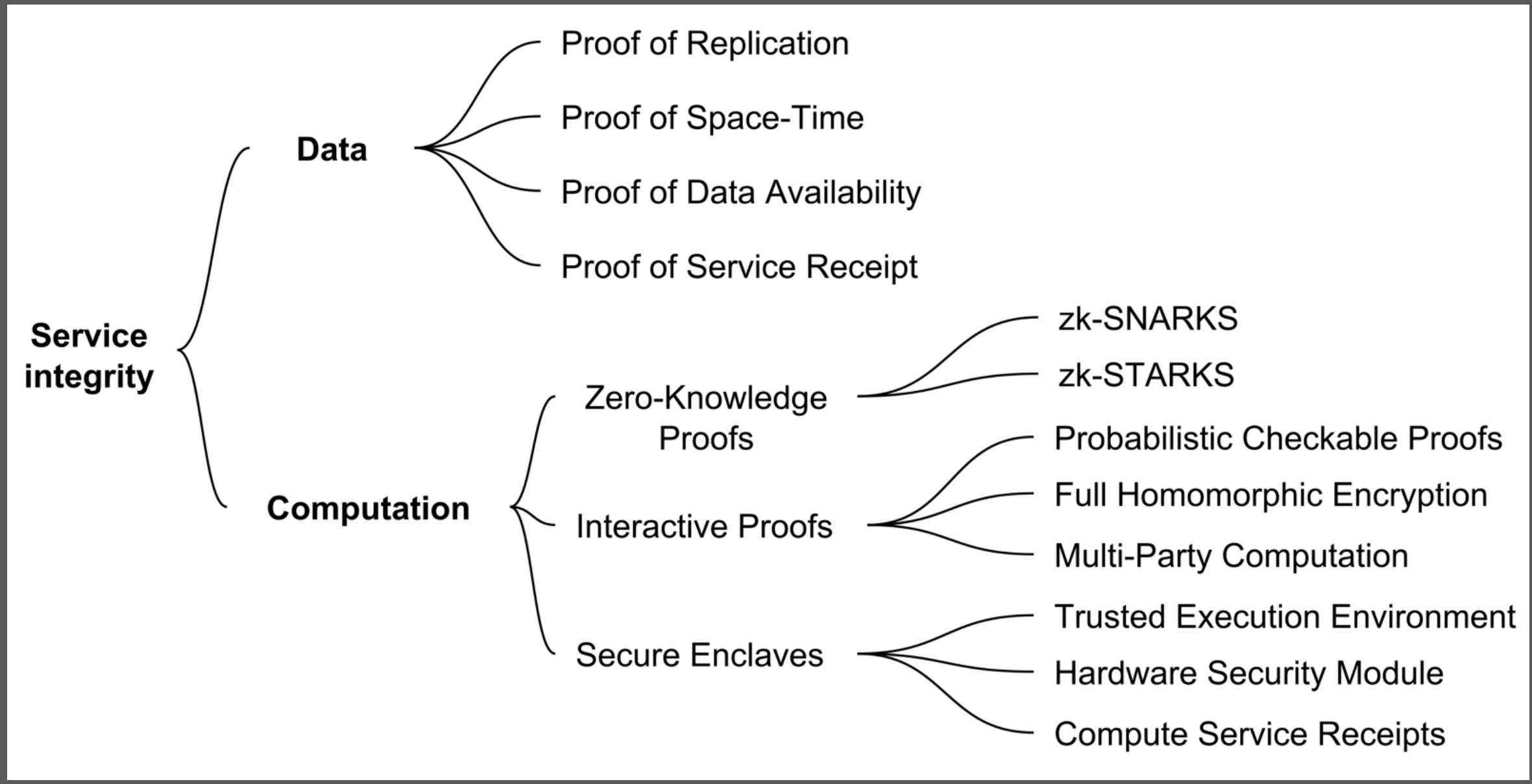
2. Try Existing Patterns

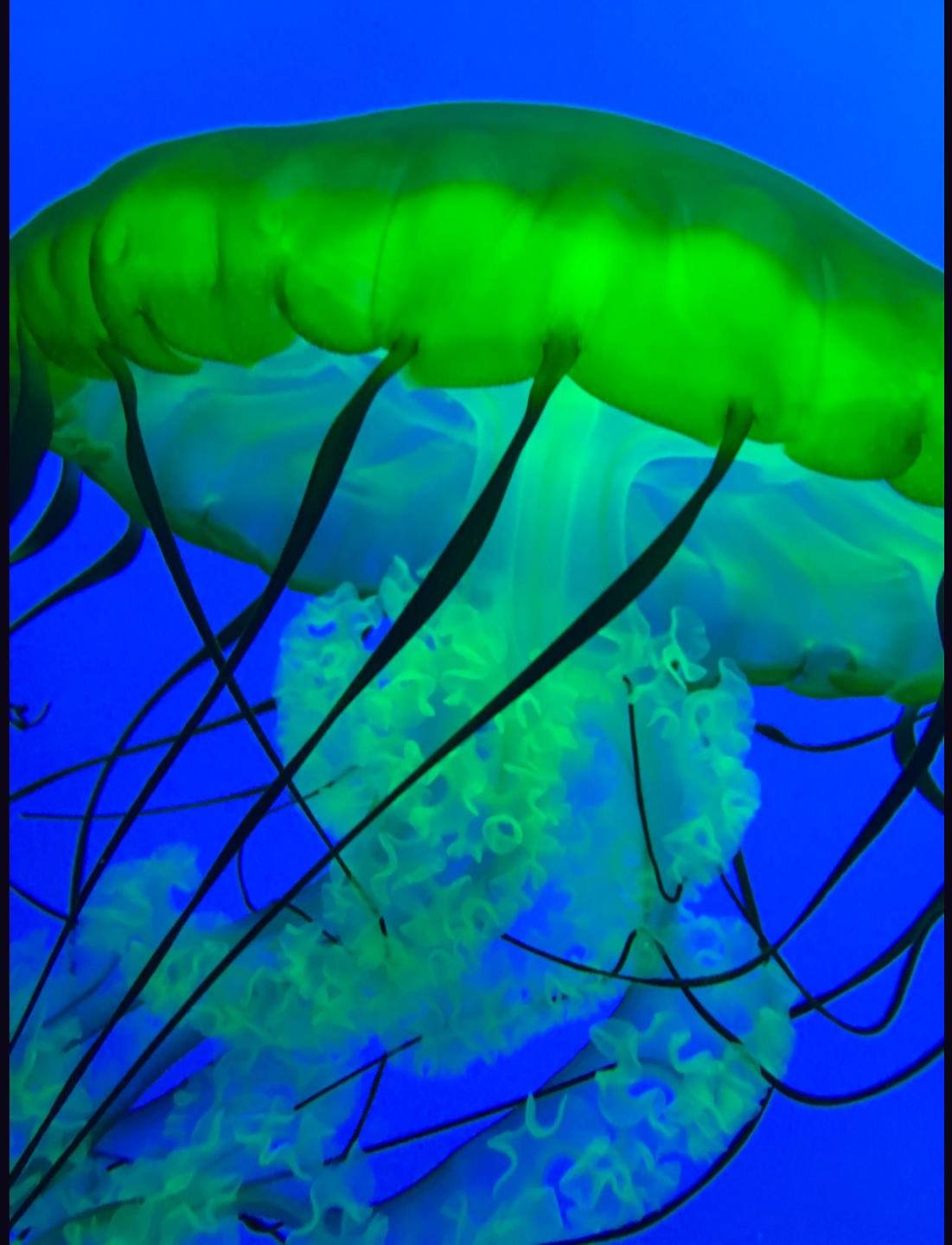
1. **Curation**
2. **Proofs of human or compute work**
3. Identity
4. Reputation
5. Governance / software updates
6. Third-party arbitration
7. …

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

2.2 Patterns for Proofs of Compute Work





Case Study: Analysis of Bitcoin

Bitcoin objective function

Objective: Maximize security of network

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

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

Result of Bitcoin's objective function:

People are maximizing security! = Maximizing electricity

More power than USA by mid 2019

Case Study: Design of 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

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

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?

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

Towards Good Acting via Staking, Id, Reputation

Good acting general

- **Key goal: Is there a means to get high-quality metadata? Eg How do we prevent non-owners of the data from submitting that data?** (Fraud).

- **Key goal: Are we incentivizing skin-in-the-game? E.g. Does return on capital increase as stake increases?**
- Do big providers of data need to stake a lot? Consumers?
- If I have high stake but low reputation, can I make $? If I have low stake but high reputation, can I make $? If I have high stake and high reputation, can I make $$$?
- Do keepers (at least keepers with a higher level of reward or privilege) need to stake a lot?
- Is there a good threshold of individual / org identity - are they are who they say they are? At the very least, to prevent Sybil attacks. But potentially more, to adhere to data privacy regulations.
- Is there a good measure of individual / org reputation - are they a good actor in the ecosystem? (In buying, selling, keeping, etc?)
- Is there a good threshold of data identity - is the data what they say it is?
- Is there a good measure of data reputation - is the data useful?
- Does remuneration favor data freshness?

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”**

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?	✓	✓	≈	≈	✓	✓

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

From AI data to AI *services*

ocean

Motivations:

- **Privacy, so compute on-premise or decentralized**
- **Data is heavy, so compute on-premise**
- **Link in emerging decentralized AI compute**

Objective function: Maximize supply of relevant *services*

=reward curating *services* + proving that it was delivered

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

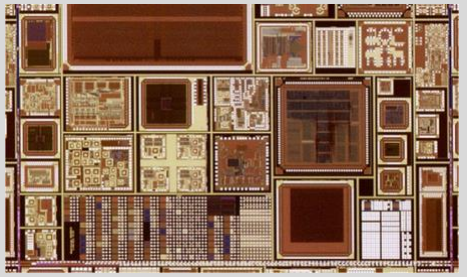
predicted popularity of *service*

proofed popularity of *service*

On Tools

Tools

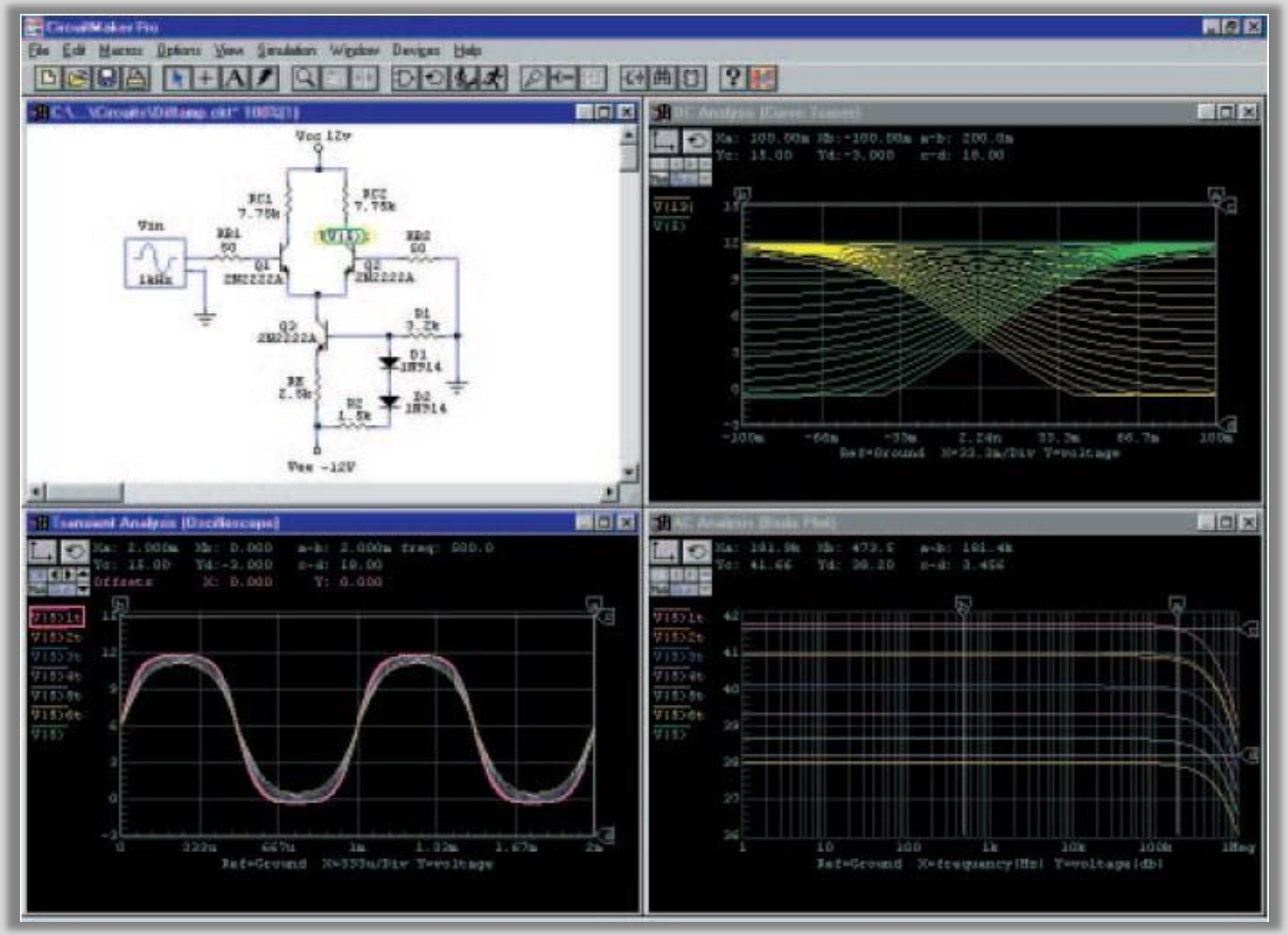
Q: How do we design circuits?
($50M+ at stake)



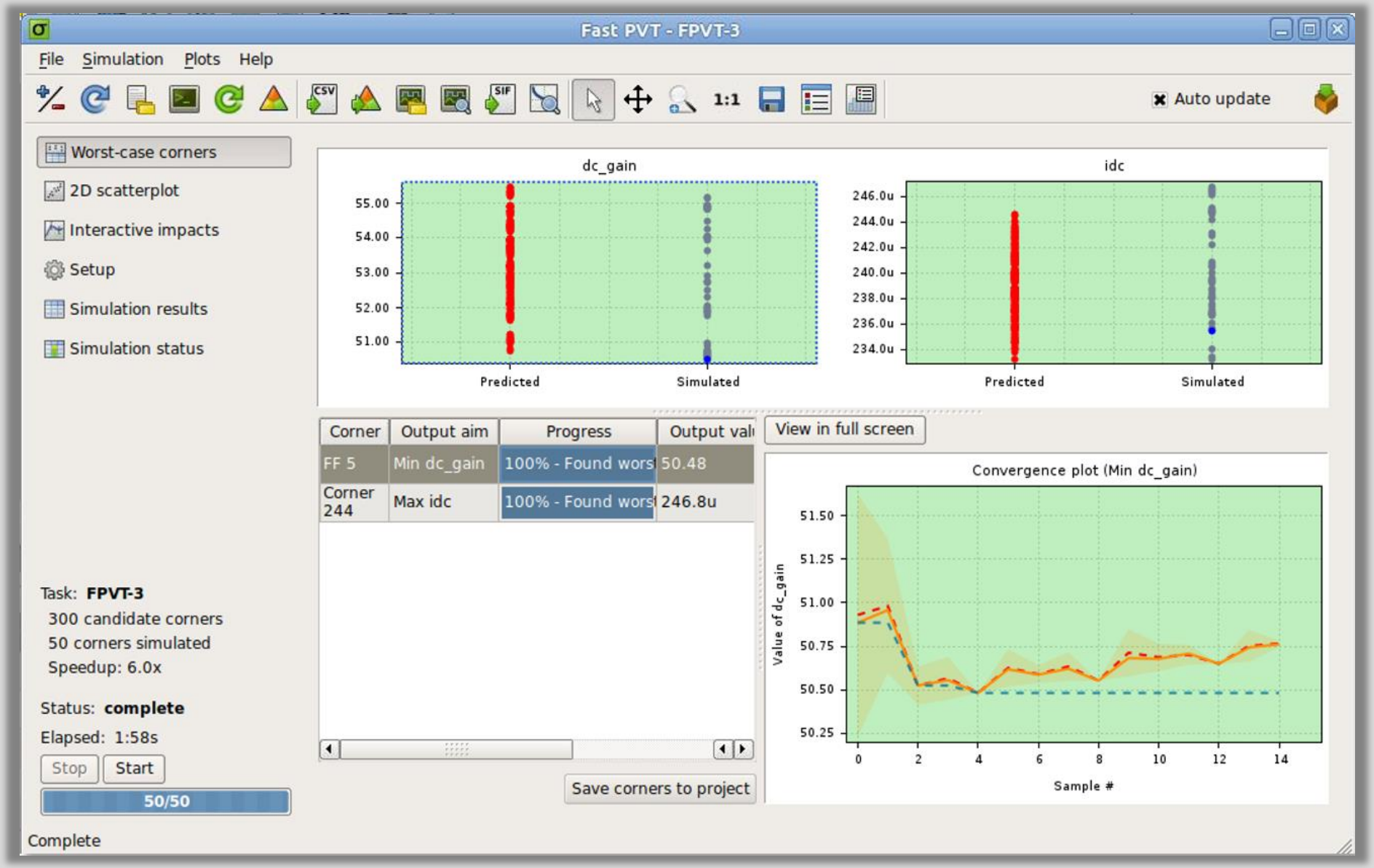
A: **CAD Tools** for

- Simulation,
- Verification, and
- Design

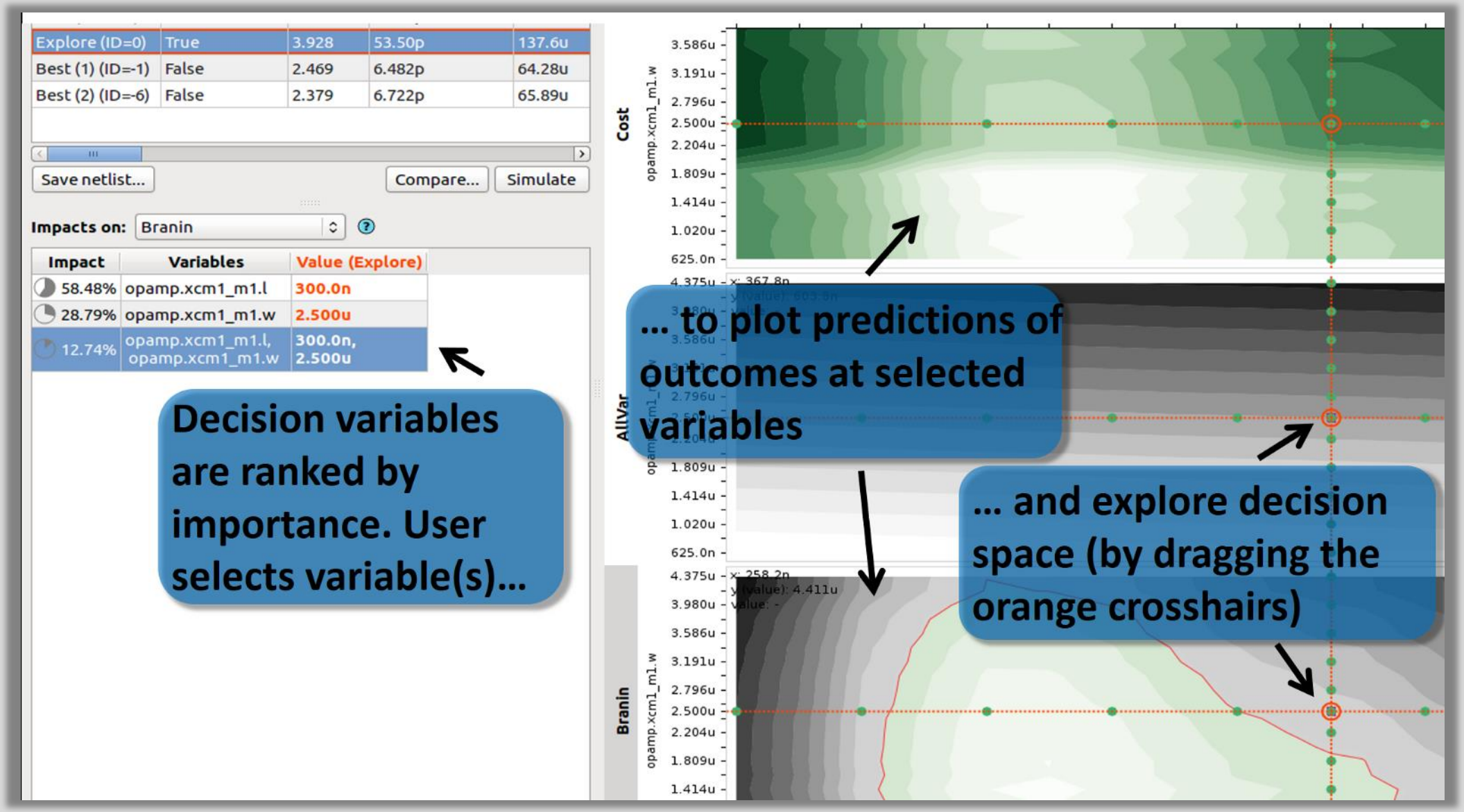
Simulation of Circuit Dynamics



Verification across Worst-Case Conditions



Interactive Design / Exploration



Tools for Tokenized Ecosystems?

- **Simulation? _____**
- **Verification? _____**
- **Design? ___**

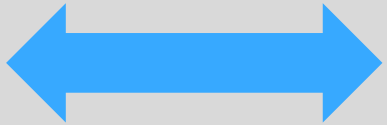
- **We have tokenized ecosystems to design, $1B at stake**
- **are designing tokenized ecosystems…**
- **Without tools**
- **Which means we might be getting it all wrong!** 😱

Towards Token *Engineering*

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

Analysis:

Game theory

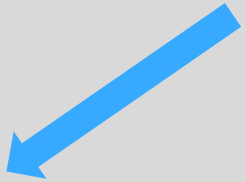


Synthesis:

Mechanism Design

Practical constraints

Optimization Design



Engineering theory, practice and tools + responsibility

Token Engineering for Analysis & Synthesis

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.

Engineering Responsibility

Engineering has

Theory,
Practice,
Tools,
Responsibility

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**.

Towards a Token Engineering 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!**

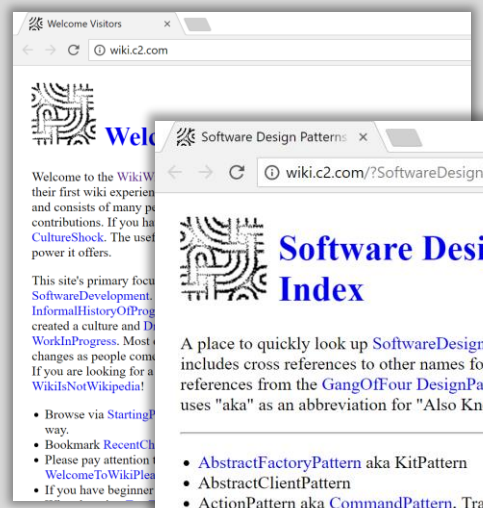
Mission of the TE Community

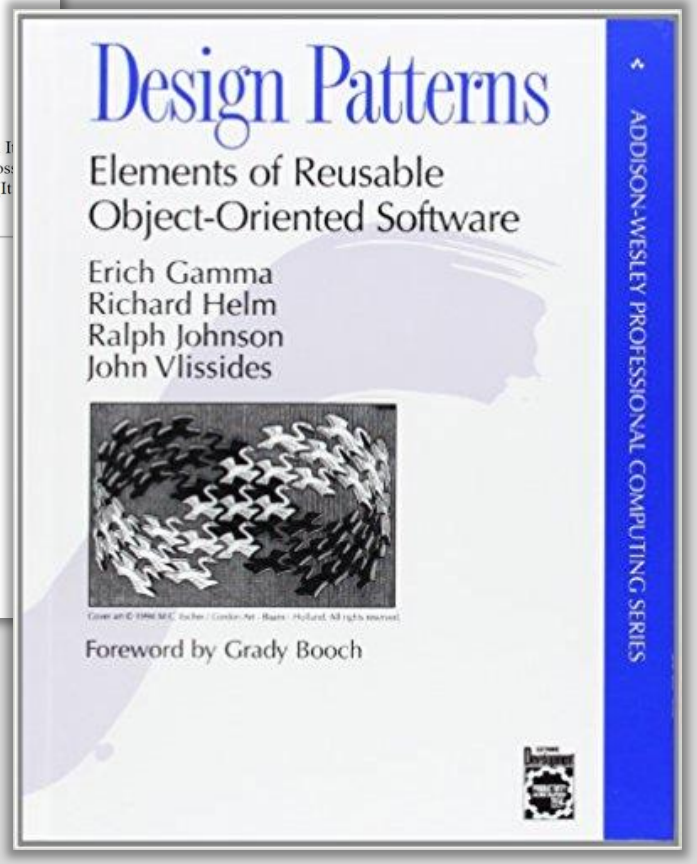
To grow TE into an **engineering discipline**

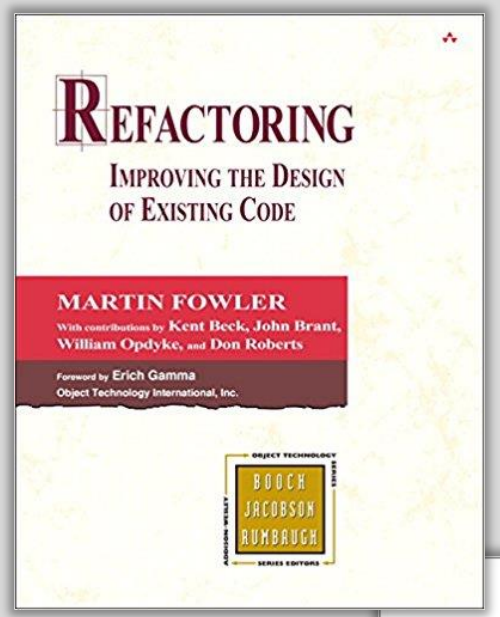
collectively as a community

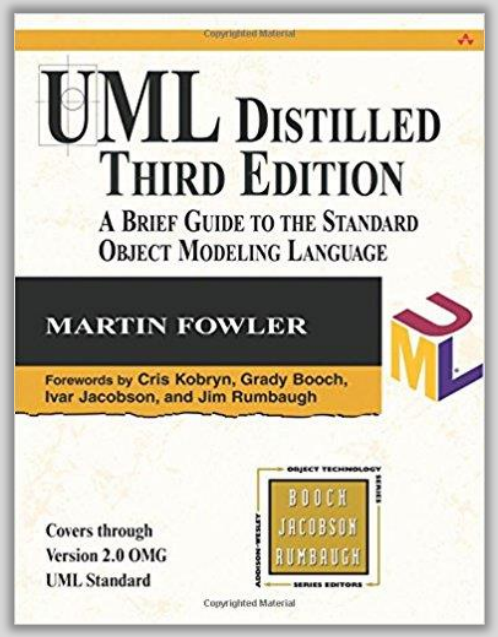
in a decentralized, permissionless, open-source fashion that all can contribute to and all can use.

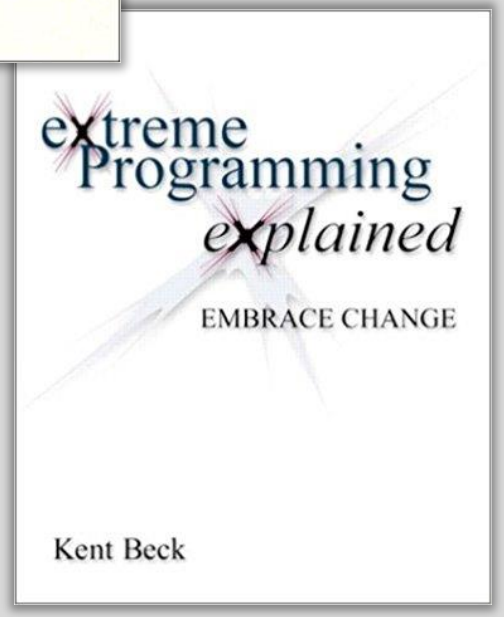
An inspiration: evolution of software engineering. C2 wiki → 💥



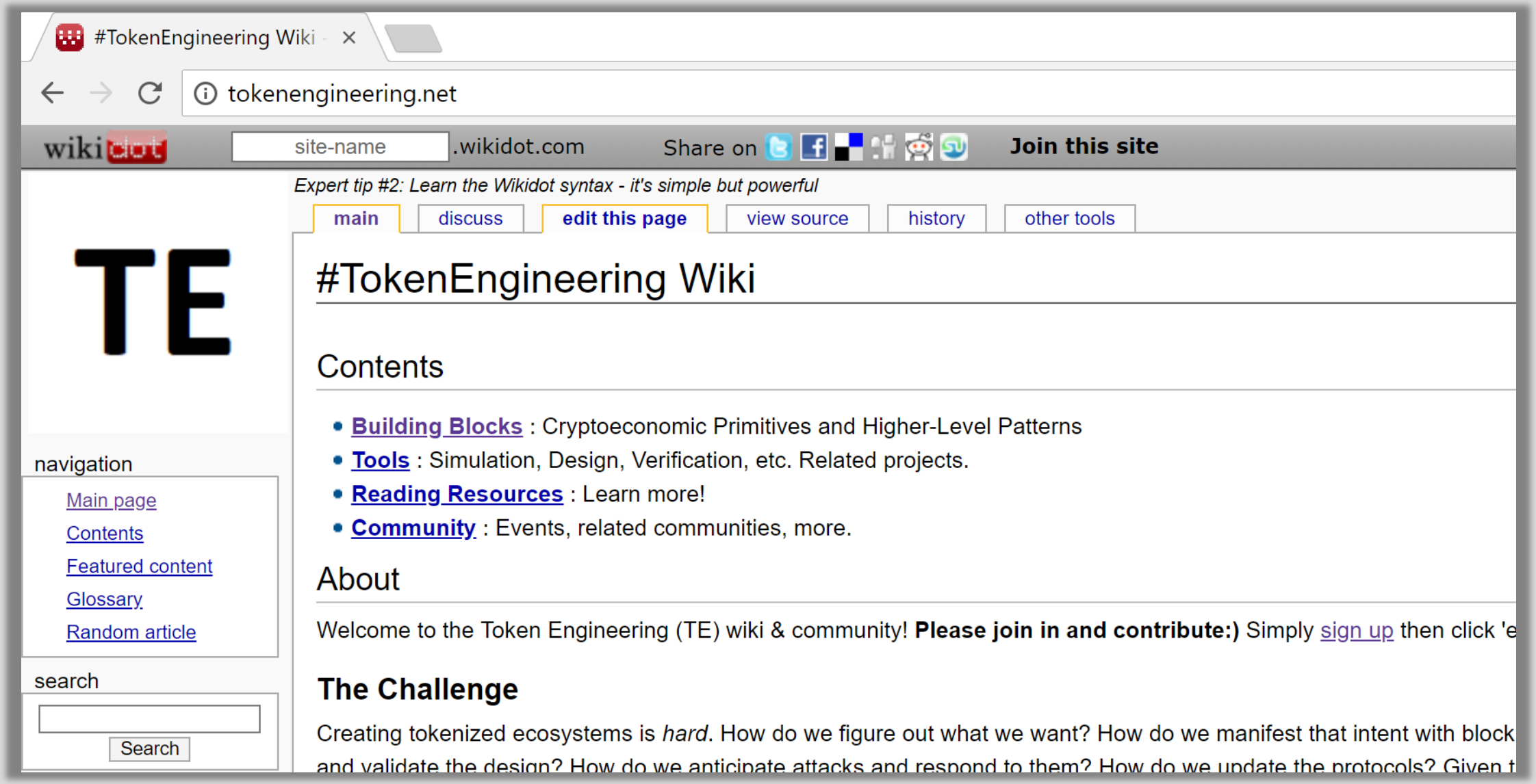








A Wiki for TE: tokenengineering.net



TE

navigation
Main page
Contents
Featured content
Glossary
Random article
search
Search
About this site
Recent changes
Contact
Donate
Legal
Help
toolbox
Printable version

Building Blocks

Building Blocks : Cryptoeconomic Primitives and Higher-Level Patterns

Types of blocks (highest level) (one framing)

- Curation
- Proofs of human or machine work
- Identity
- Markets: Prediction Markets, Trading, Bonding Curves, etc
- Micro-Economical: DAOs, Stablecoins, etc
- Consensus: Voting, staking, etc
- Reputation
- Governance / software updates
- Third-party arbitration
- Inter-operability
- «add to me! revise me! :)»

Token Engineering In Practice

Other Framings

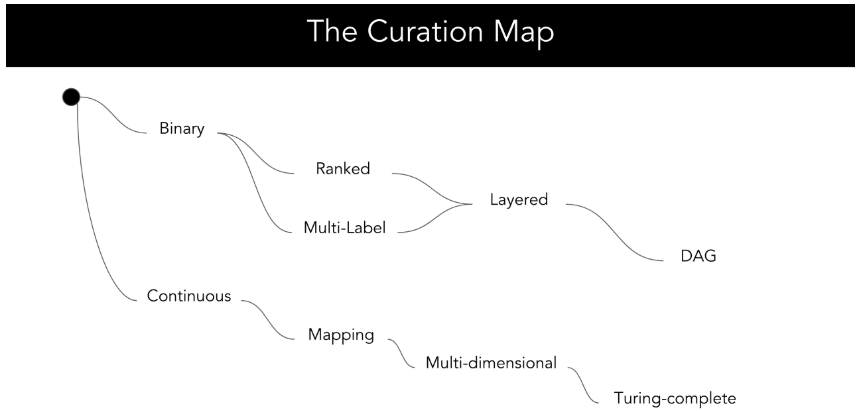
Other ways to frame or group building blocks include: «FIXME: add links to these. The links are here »

- How tokens are distributed. This includes releasing coins for “work”, according to a controlled supply schedule; 100% pre-mining; burn-and-mint; bounty ICOs; and

TE

navigation
Main page
Contents
Featured content
Glossary
Random article
search
Search
About this site
Recent changes
Contact
Donate
Legal
Help
toolbox
Printable version

Curation



(Overview slides)

Building blocks for discrete-valued membership:

- Binary membership: TCR (Token Curated Registry)
- Ranked membership: Graded TCRs (gTCRs)
- Multi-label membership: see slide
- Layered membership: Layered TCR (LTCR)
- DAG membership: Stake Machines

TE

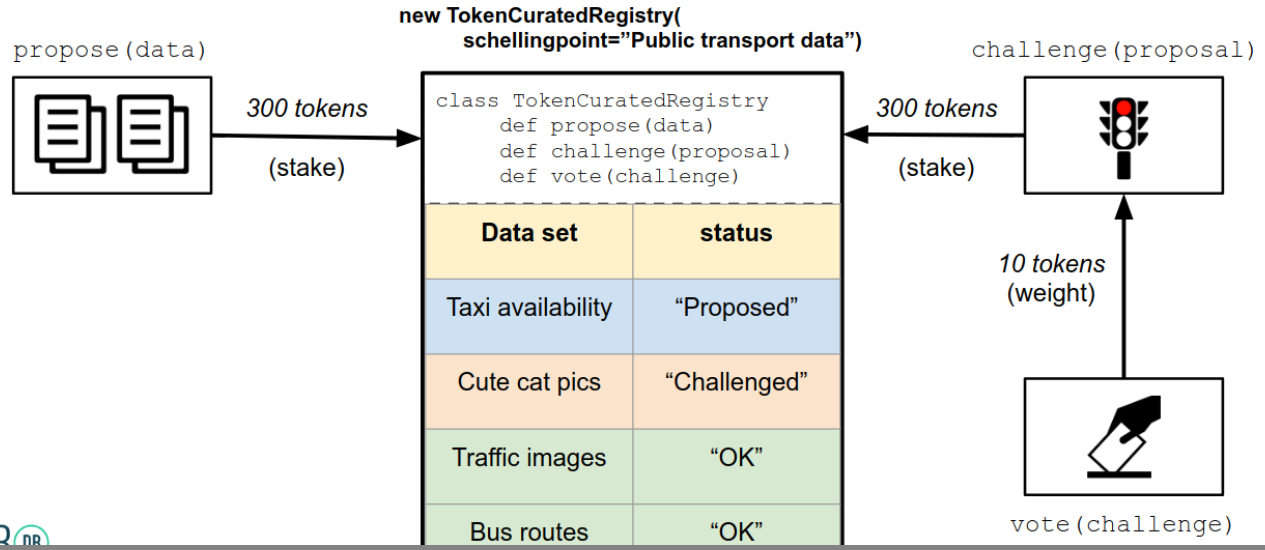
navigation
Main page
Contents
Featured content
Glossary
Random article
search
Search
About this site
Recent changes
Contact
Donate
Legal
Help
toolbox
Printable version

TCR (Token Curated Registry)

Basic Idea

TCRs are useful for whitelists, blacklists, and more. Items are either in a set, or not (binary membership).

Conceptually, a TCR is a registry of listings generated by token holders. Listings can be anything from names to hashes of full blown records stored in IPFS. A holder stakes a portion of tokens to perform an action, i.e. adding a listing. Token holders can vote on whether to accept or reject the action using their tokens. A majority wins the vote and the action is accepted or rejected. The list is updated and should become more valuable given the improved listings and staked tokens are now out of circulation. Perhaps it is a finite list of exclusive members who have access to VIP services? Ref



TE

navigation
Main page
Contents
Featured content
Glossary
Random article
search
Search
About this site
Recent changes
Contact
Donate
Legal
Help
toolbox
Printable version

Tools

Tools : Simulation, Design, Verification, etc. Related projects

Tools for Simulation & Verification

- Incentivai – test smart contract mechanism design via simulation using AI agents
- Related: there's lots of work on formal verification of smart contracts. That covers straightforward logic but doesn't measure dynamics of tokenized systems, which is what a token simulator could do.
- Related: circuit simulators like SPICE which simulate dynamical systems
- Related: dynamical systems simulators in other fields
- Related: agent-based simulators in other fields
- «add to me»

Simulation cases

- Partial Slashing Simulation - simple agent-based simulation for partial slashing of Ethereum Casper FFG

Tools for Design

- Related: computer-aided design (CAD) tools in other fields. E.g. in the Electronic Design Automation (EDA) industry for chip design.
- «add to me»

Other Related Projects

- BlockScience – a technology research and analytics firm specializing in the design and evaluation of decentralized economic systems

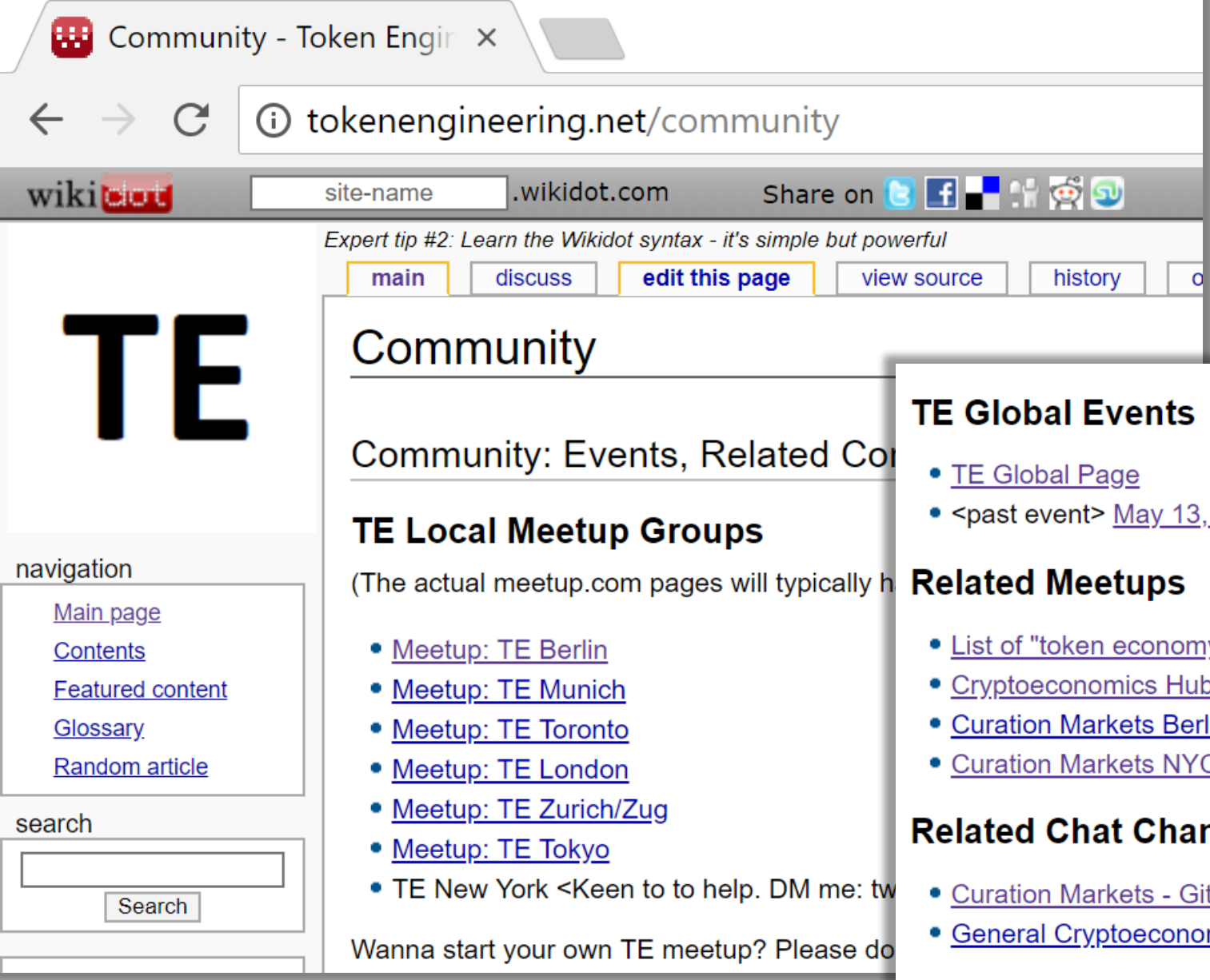
Reading Resources

Related Disciplines

- Mechanism Design
- Algorithmic Game Theory
- Economic Systems Design
- Game Mechanics Design
- Systems Engineering
- Publicy Policy Analysis & Design
- Swarm Robotics
- Operations Research

Reading Resources

- Alex Evans, A Crash Course in Mechanism Design for Cryptoeconomic Applications, Oct 2017
- Trent McConaghy, Towards a Practice of Token Engineering Part I, Part II, Part III, Feb 2018
- Michael Zargham, On Engineering Economic Systems Part I, Part II, Mar 2018
- Elad Verbin, Behavioral Crypto-Economics: The Challenge and Promise of Blockchain Incentive Design, Mar 2018
- Jacob Horne, The Emergence of Cryptoeconomic Primitives
- Chris Burniske, The Crypto J-Curve, Aug 2017
- Chris Burniske, Cryptoasset Valuations, Sept 2017
- Adrian Colyer, Designing Secure Ethereum Smart Contracts - a Finite State Machine Approach, Mar 2018
- «add to me» «and start to group these more:) »



TE Global Events

- TE Global Page
- <past event> May 13, 2018 - NYC

Related Meetups

- List of "token economy" meetups
- Cryptoeconomics Hub Madrid meetup, YouTube Channel
- Curation Markets Berlin
- Curation Markets NYC

Related Chat Channels

- Curation Markets - Gitter
- General Cryptoeconomics Group - Telegram, by Cryptoeconomics Hub Madrid

Token Engineering YouTube

Subscribe to our channel & be the first to know when new content is available.

- Global Gathering NYC

Ways to Participate

- Edit this wiki and impart your wisdom! Add blocks, tools, readings.
- Tweet with #tokenengineering hashtag
- Attend a meetup (see Events). Or: start your own!
- Subscribe to the TE mailing list:

Email Address

Subscribe

;-) dan-coincrunch
;-) achill rudolph
;-) Eden Dhaliwal
;-)
morganmoskalyk
;-) Frederik Bussler
;-) hodlhands
;-)
AngelaKreitenweis
;-) adriankrion
;-) cburniske
 Or Luis Shemtov
 bmann
;-)

Conclusion

Conclusion:
Towards a Practice of #TokenEngineering

- Token design ≈ optimization design
- So, approach token design as optimization design!
  1. Formulate problem. Objectives, constraints.
  2. Try existing patterns. Iterate.
  3. If needed, try new design.
- This process helped a *lot* for designing Ocean (so far)
- Token Engineering = Theory + practice + tools + responsibility
- A community is forming! Tokenengineering.net

Trent McConaghy
@trentmc0

ocean