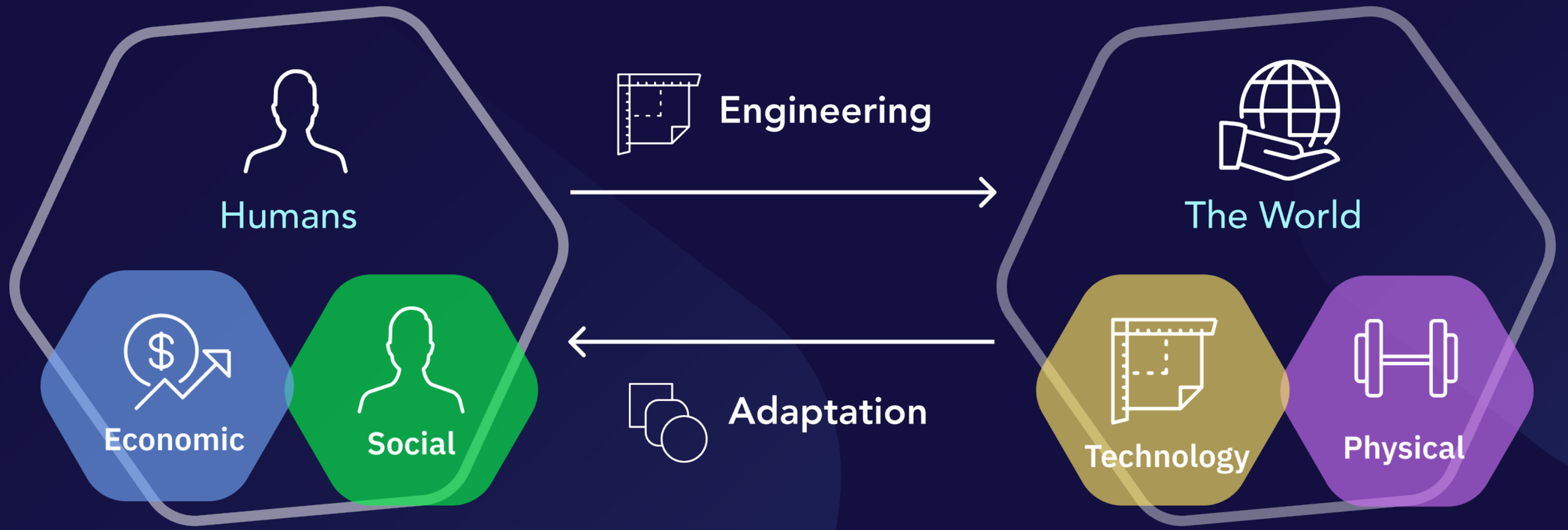
BLOCKSCIENCE

Scientific Approach to Token Systems Simulation

*Dynamic Stochastic **Non-Equilibrium** Models*

Dr. Michael Zargham
@mZargham

April 11, 2019
Complex Systems Engineering Master Class
Groningen, Netherlands



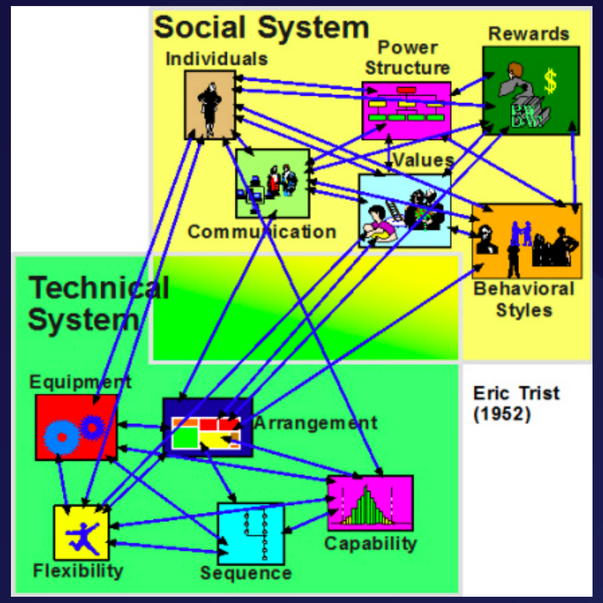
Complex Systems:
Temporal Dynamics & Closure under Behavioral Adaptation

Tools, Skills, and Domains of knowledge are required for design and analysis come from a diverse set of fields

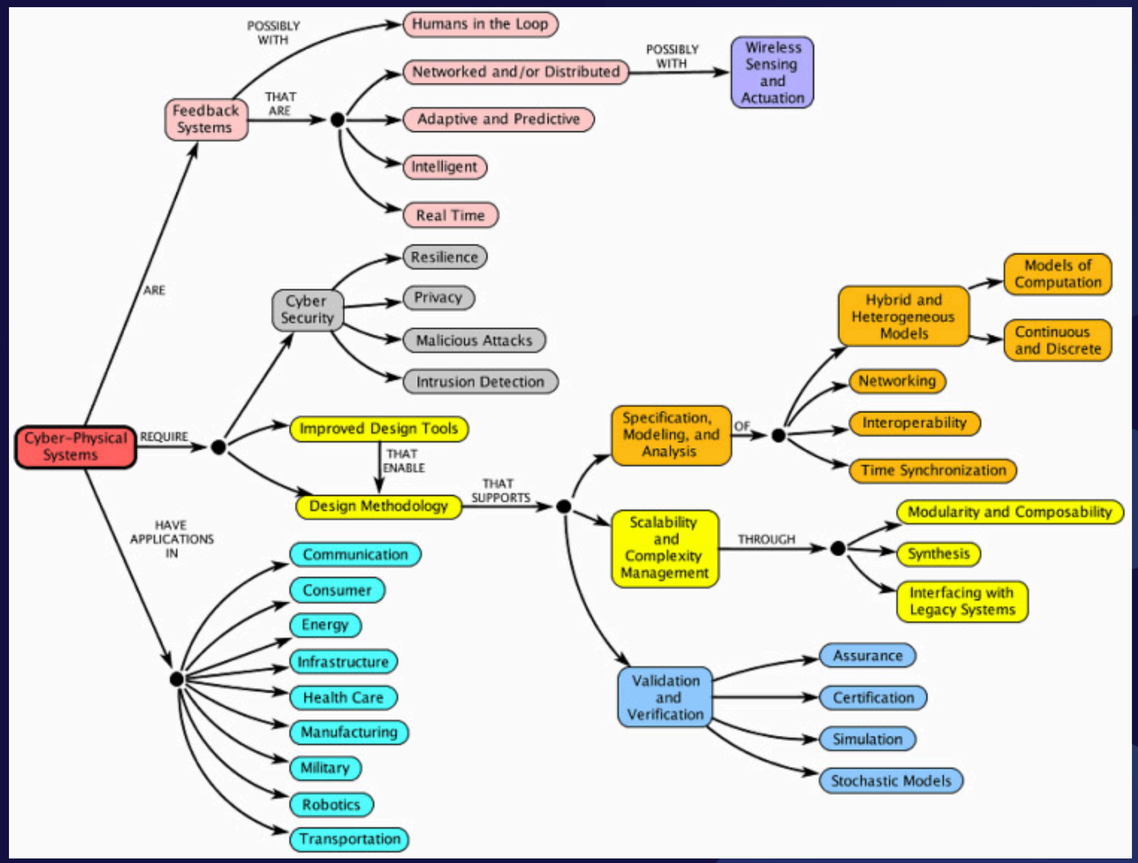
BLOCKSCIENCE

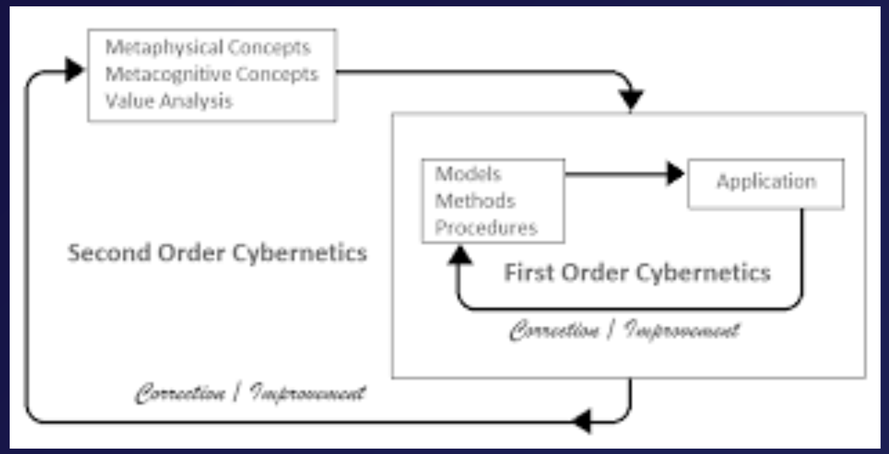
Paradigms for Complexity in Engineering

SocioTechnical Systems
Eric Trist, Ken Bamforth and Fred Emery (1952)



Cyberphysical Systems
Helen Gill (2006)





Second Order Cybernetics

Wiener, Bateson & Mead (1973)

BLOCKSCIENCE

BLOCKSCIENCE

Interconnectedness of Technology and Behavior

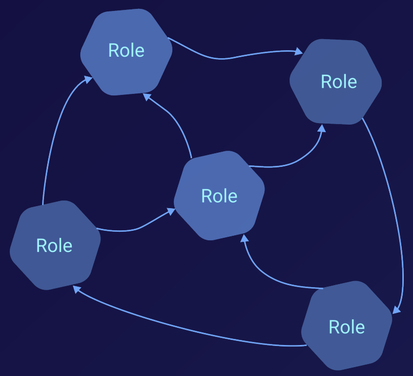
Sustainable flow of value amongst Roles

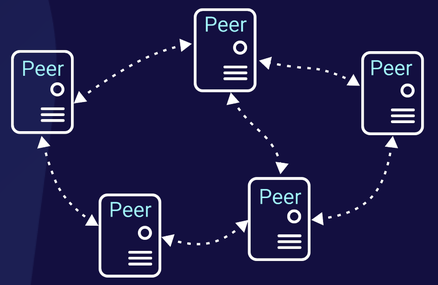
Interaction Patterns & Local Behavior of Agents

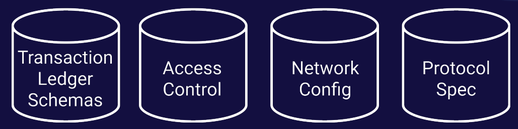
Peer to Peer communication and computation network

Software Enforcing low level immutable “information provenance”

Requires







Enables

BLOCKSCIENCE

Three Kinds of Scientific Results

- **Analytical:**

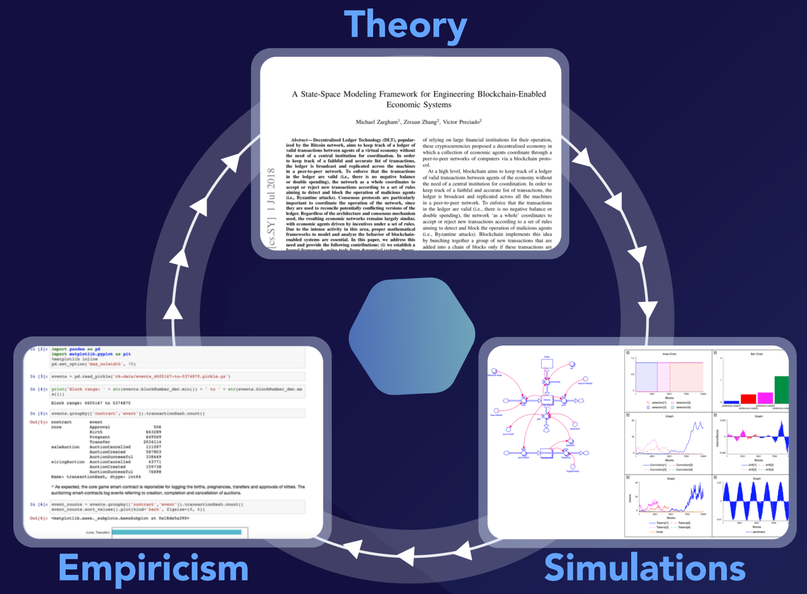
  Formal Assumptions —> Formal Conclusions

- **Computational:**

  Formal Assumptions —> Data —> Analysis to Reach Conclusions

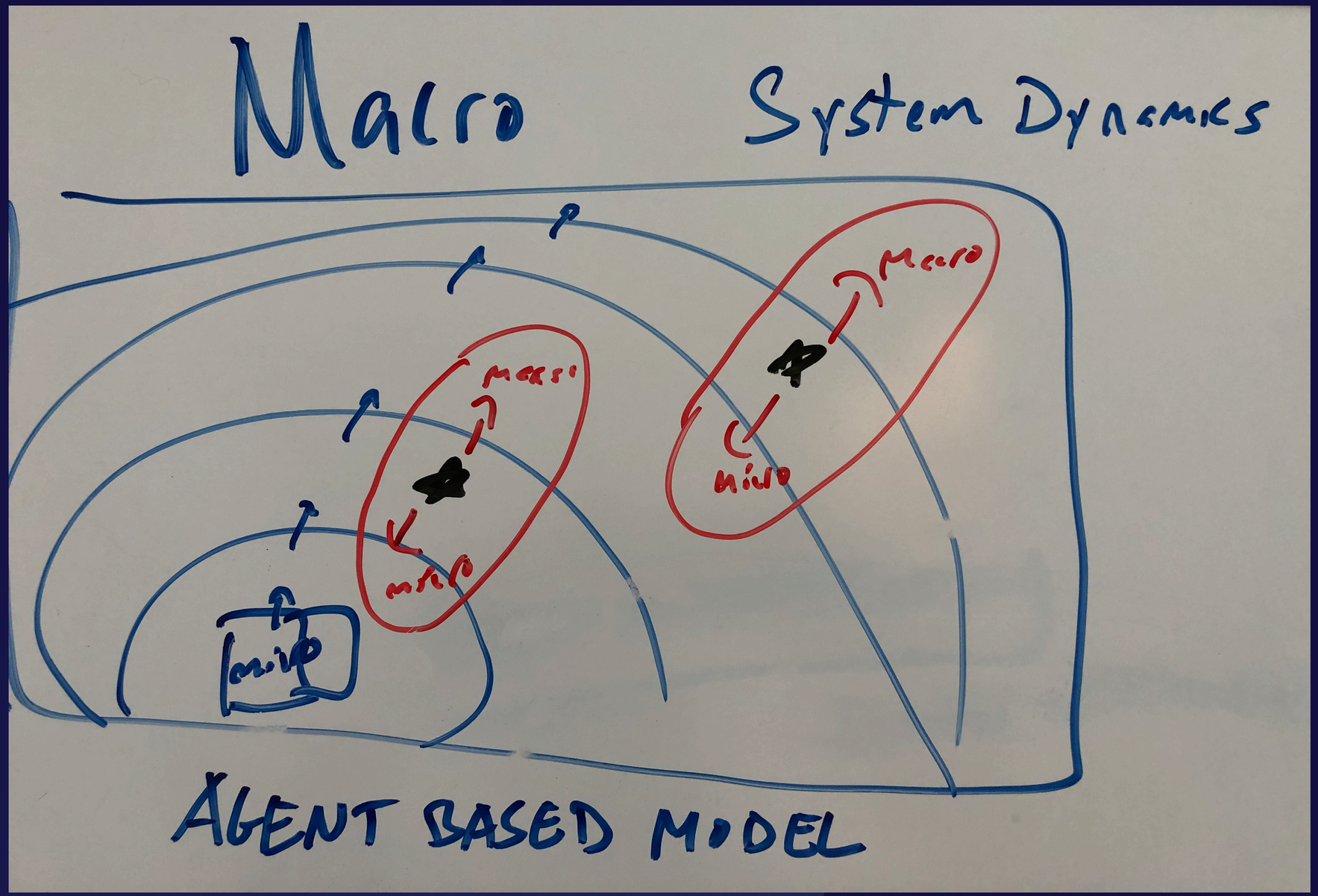
- **Empirical:**

  Measurements & Observations —> Data —> Analysis to Reach Conclusions



BLOCKSCIENCE

An Decision Engineer's Perspective on Economic Modeling



BLOCKSCIENCE

Mechanisms as rules governing allowable actions

Definition 4. *Consider the set of* ***Mechanisms*** *to be* $\mathcal{F}$ *such that any* $f \in \mathcal{F}$ *is an operator*

$$f : \mathcal{X} \times \mathcal{U} \longrightarrow \mathcal{X} \tag{1}$$

where $\mathcal{X}$ *is the space of all possible states* $\mathbf{X}$ *and* $\mathcal{U}$ *is a space of all legal actions associated with the particularly mechanism* f*.*

Definition 6. *The set of all possible* ***transactions*** *is denoted* $\mathcal{T} = \mathcal{A} \times \mathcal{F} \times \mathcal{U}$ *where an element* $t \in \mathcal{T}$ *is defined* $t = (a, f, u)$*. In order for the transaction to be valid, agent* a *must have the right to perform the state update operation* $\mathbf{X}^+ = f(\mathbf{X}, u)$ *given the current state* $\mathbf{X}$*.*

https://arxiv.org/pdf/1807.00955.pdf

Blocks are Batches of ‘local’ economic activity

An atomic update is defined

$$\mathbf{X}^{+} = f_t(\mathbf{X}, u_t) \text{ for any valid } (a_t, f_t, u_t) \text{ given } \mathbf{X} \tag{2}$$

where f_t is the mechanism used in transaction t and u_t is the action taken for transaction t. For a block defined by sequence of transactions $\mathbf{T}$, the state update is

$$\mathbf{X}^{+} = f_N(f_{N-1}(\cdots f_1(f_0(\mathbf{X}, u_0), u_1) \cdots, u_{N-1}), u_N) \tag{3}$$

any valid sequence of transactions $\mathbf{T} = [\ldots, t, \ldots]$, *where* $t = (a, f, u)$ *is valid given* $\mathbf{X}$

$$\mathbf{X}(k+1) = \mathbf{F}_k(X(k)) \tag{5}$$

denoting the closed loop state update accounting implicitly for the actions $u = P((X))$.

BLOCKSCIENCE

Modeling Uncertain or Uncontrolled Behavior

Definition 7. *A* ***policy*** $P : \mathcal{X} \longrightarrow \mathcal{U}$ *is a state dependent strategy over a particular mechanism* $f \in \mathcal{F}$*. An agent* $a \in \mathcal{A}$ *is said to be using policy* P *over mechanism* $f \in \mathcal{F}$ *if it monitors the state* $\mathcal{X}$ *and broadcasts transaction* $t = (a, f, u)$ *associated with action* $u = P(x) \in \mathcal{U}$.

	Possible Outcome	Actually Occurs
Micro Scale	Action Space	Action
Macro Scale	Reachable Space	Realization

BLOCKSCIENCE

Invariant Properties: Designing the Reachable Space

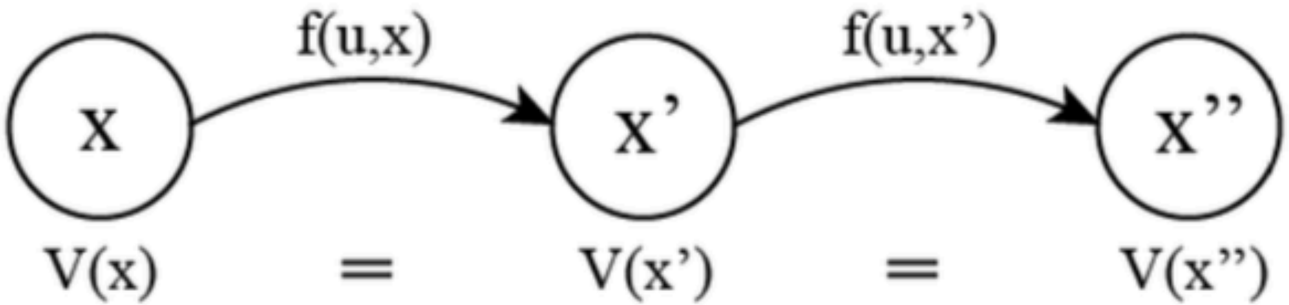
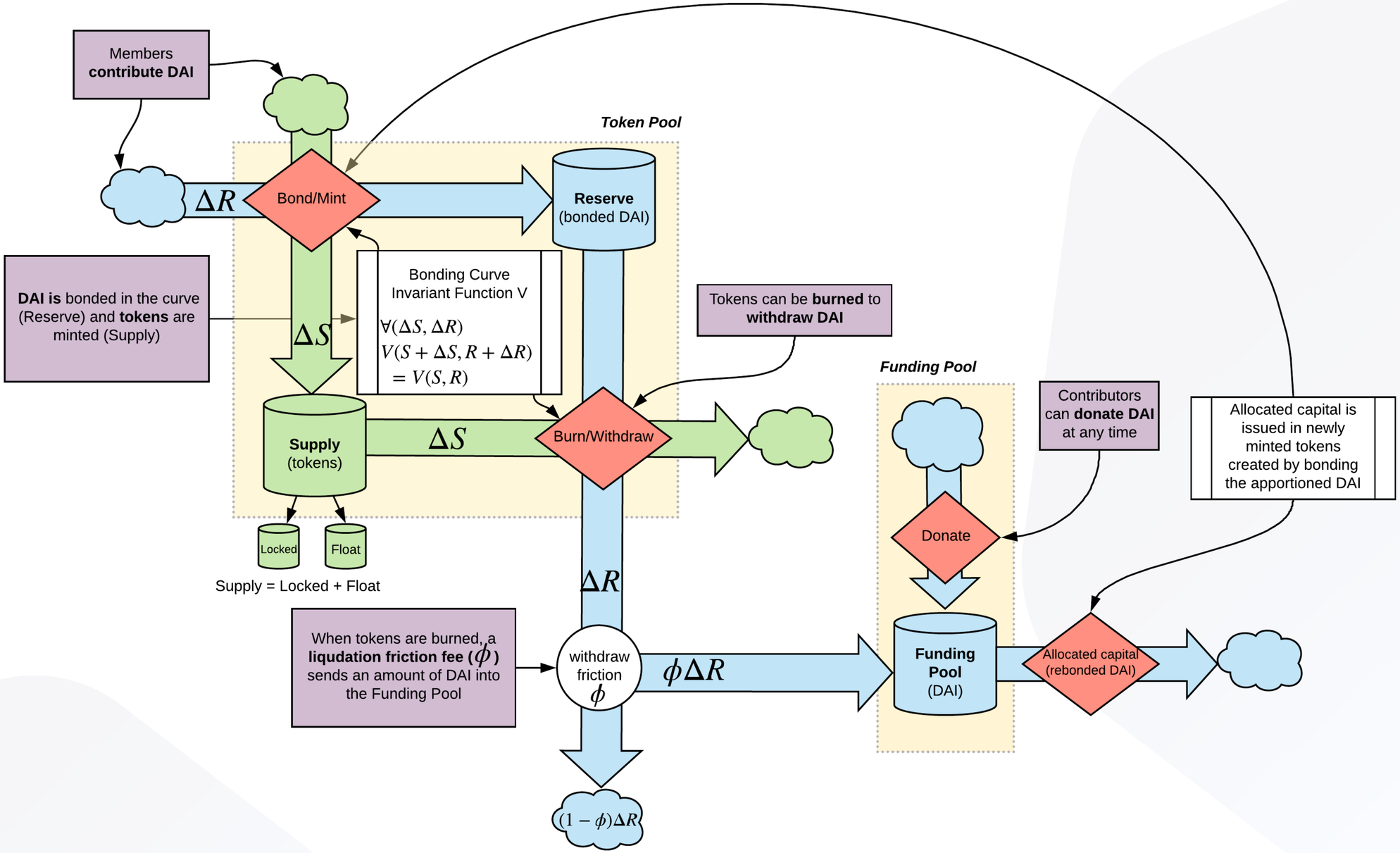


Fig. 2. Illustration of invariant properties. V is invariant under all $f(u, x)$ for all valid actions u.

https://arxiv.org/pdf/1807.00955.pdf

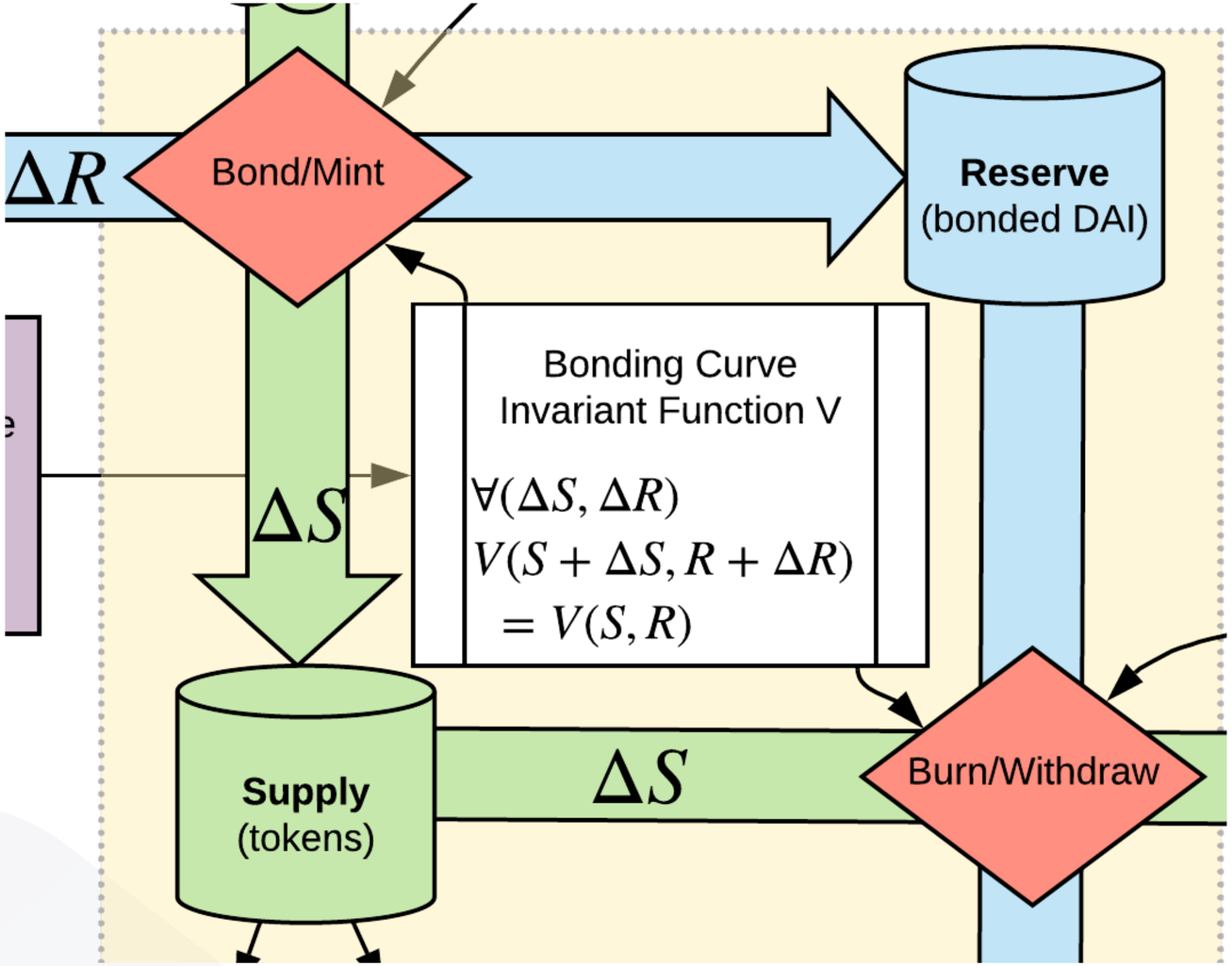
Augmented Bonding Curve for Community Tokens



GIVETH

BLOCKSCIENCE

Augmented Bonding Curve for Community Tokens



GIVETH

BLOCKSCIENCE

Augmented bonding Curve is defined by an Invariant Property

Invariant Preserving Deposit-to-Mint

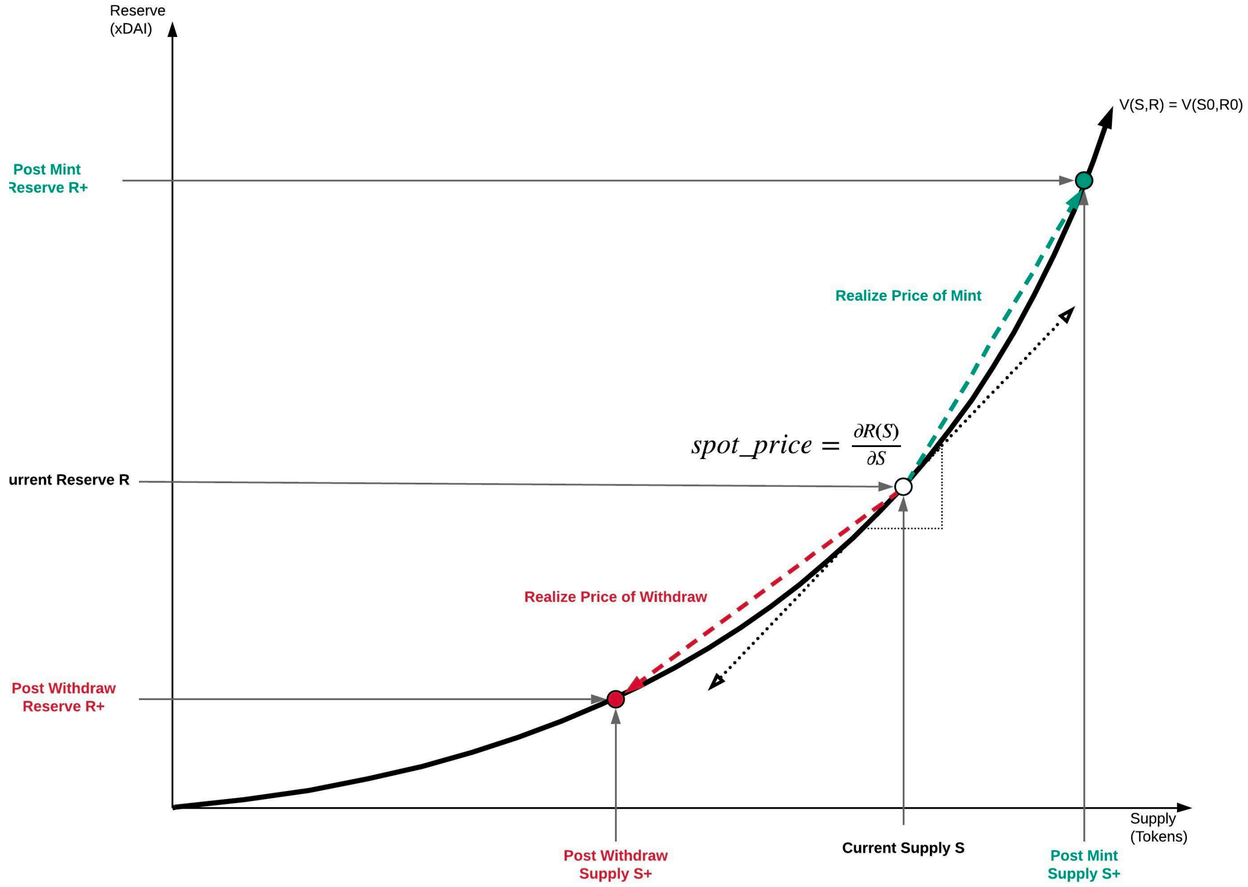
- Deposit ΔR xdai
- Conservation equation: $V(R+\Delta R, S+\Delta S) = \frac{(S+\Delta S)^\kappa}{R+\Delta R} = V_0$
- Derived Mint equation: $\Delta S = mint\Big(\Delta R; (R,S)\Big) = \sqrt[\kappa]{V_0(R+\Delta R)} - S$
- Realized Price is: $\bar{P}(\Delta R) = \frac{\Delta R}{\Delta S} = \frac{\Delta R}{\sqrt[\kappa]{V_0(R+\Delta R)} - \sqrt[\kappa]{V_0(R)}} \to \left(\frac{\partial S(R)}{\partial R}\right)^{-1}$ as $\Delta R \to 0$
- The limiting price is the spot price:

$$\lim_{\Delta R \to 0} \bar{P}(\Delta R) = \left(\frac{\partial S(R)}{\partial R}\right)^{-1} = \left(\frac{V_0^{1/\kappa} \cdot R^{1/\kappa - 1}}{\kappa}\right)^{-1} = \frac{\kappa R^{1-1/\kappa}}{V_0^{1/\kappa}} = \frac{\kappa R^{(\kappa-1)/\kappa}}{V_0^{1/\kappa}} = P(R)$$

Invariant Preserving Burn-to-Withdraw

- Burn ΔS tokens
- Conservation equation: $V(R-\Delta R, S-\Delta S) = \frac{(S-\Delta S)^\kappa}{R-\Delta R} = V_0$
- Derived Withdraw equation: $\Delta R = withdraw\Big(\Delta S; (R,S)\Big) = R - \frac{(S-\Delta S)^\kappa}{V_0}$
- Realized Price is: $\bar{P}(\Delta S) = \frac{\Delta R}{\Delta S} = \frac{\frac{S^\kappa}{V_0} - \frac{(S-\Delta S)^\kappa}{V_0}}{\Delta S} \to \frac{\partial R(S)}{\partial S}$ as $\Delta S \to 0$
- The limiting price is the spot price:

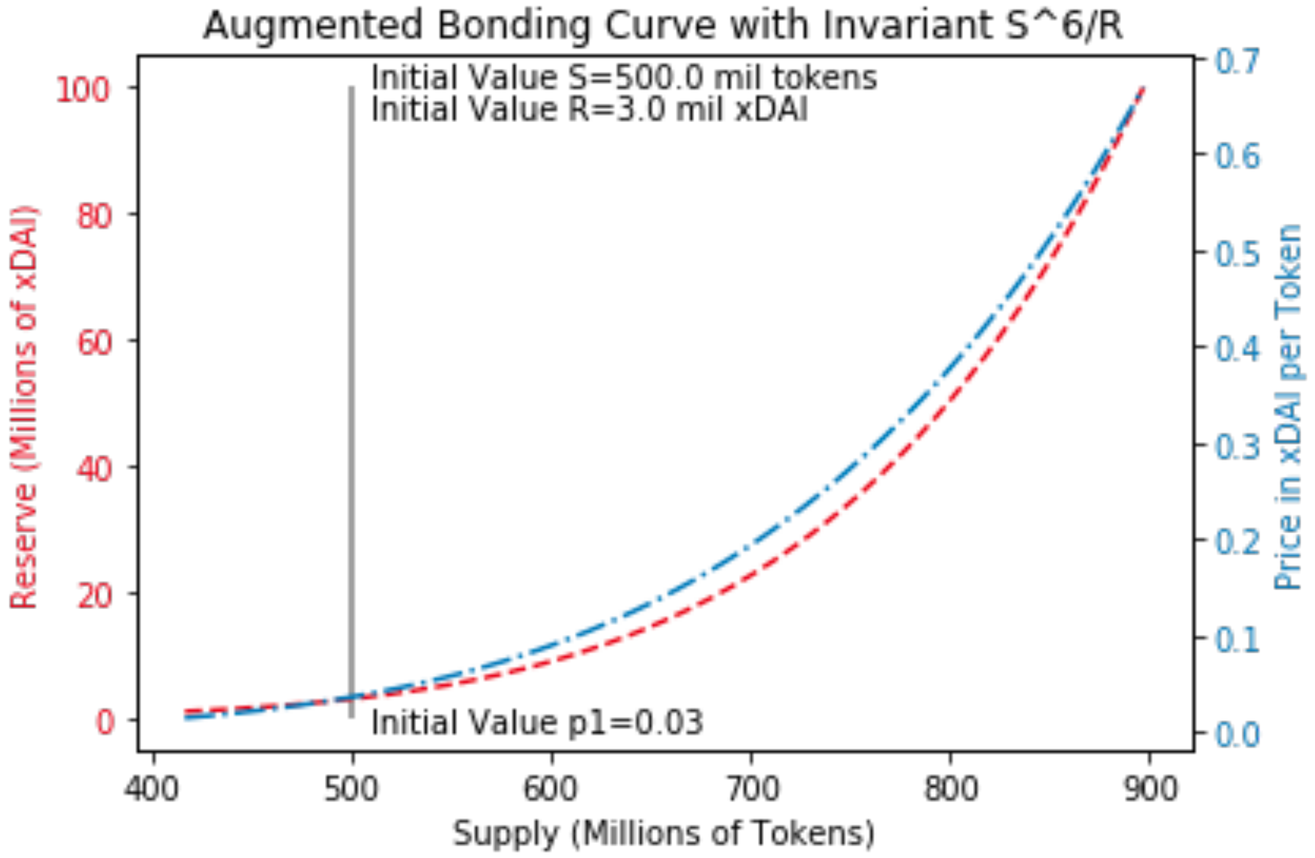
$$\lim_{\Delta S \to 0} \bar{P}(\Delta S) = \frac{\partial R(S)}{\partial S} = \frac{\kappa S^{\kappa-1}}{V_0} = \frac{\kappa \cdot (\sqrt[\kappa]{V_0 R})^{\kappa-1}}{V_0} = \frac{\kappa R^{(\kappa-1)/\kappa}}{V_0^{1/\kappa}} = P(R)$$



Take a deep breathe; this about the same difficulty level as high school physics

BLOCKSCIENCE

Invariant Property Simulation: Validation by Direct Method



Detailed Example:

Initial Supply = 500 Million tokens
Initial Reserve = 3 Million xDAI

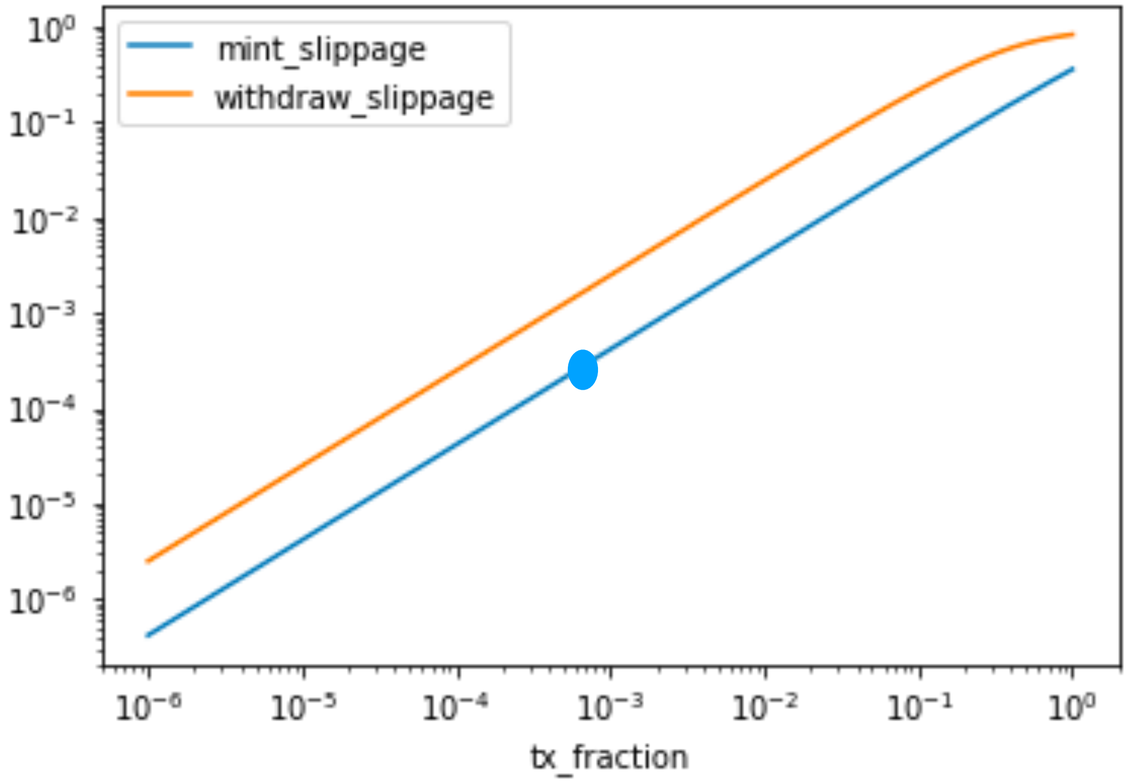
Bonding Curve Invariant:

$V(S,R) = S^6/R = 1.3 \times 10^{21}$

Initial Price = 0.03 xDAI per Token

BLOCKSCIENCE

Implications of Invariant Preserving Mechanisms



Detailed Example:

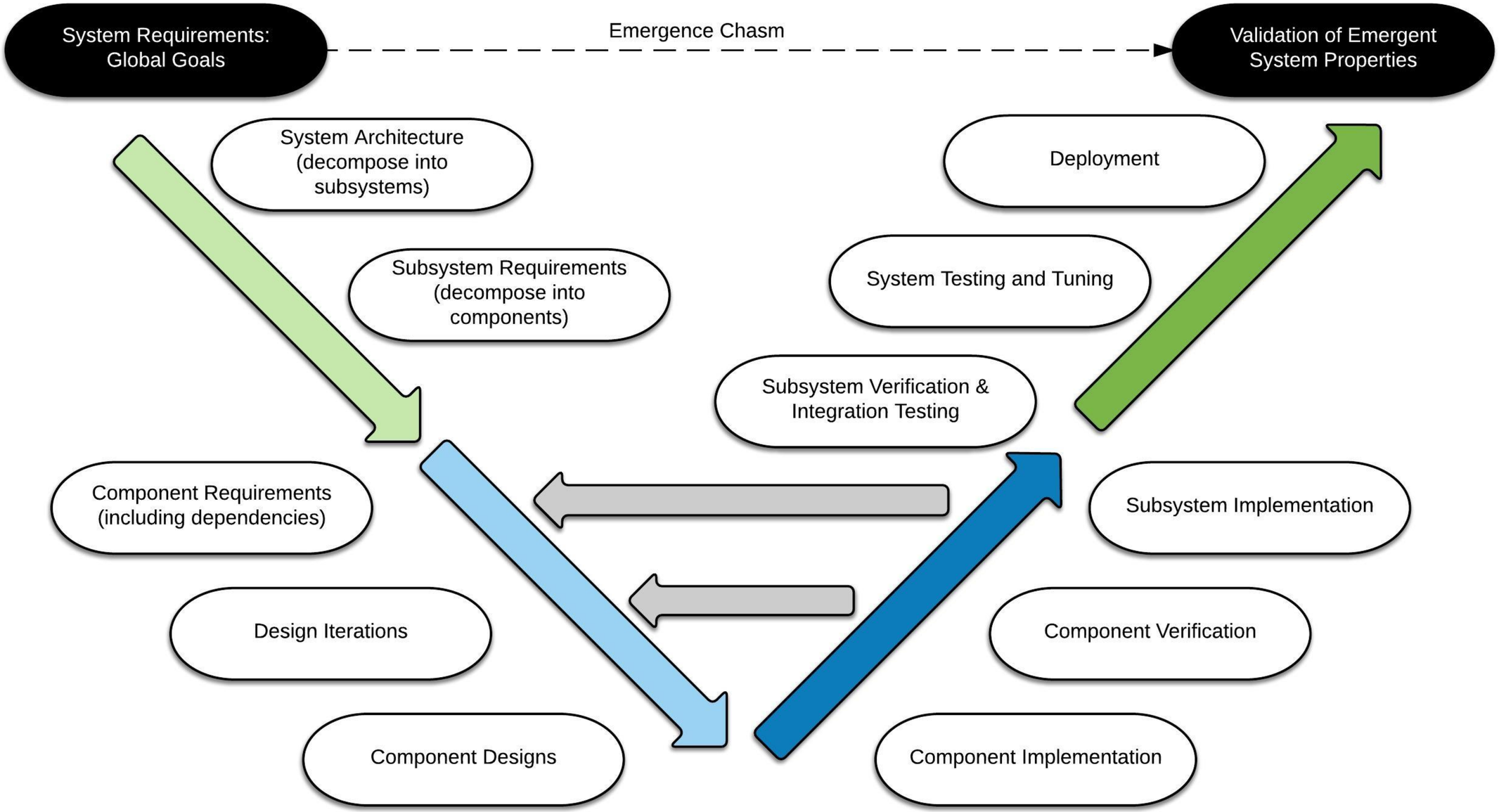
Suppose I “Deposit to Mint” an amount of xDAI that increases the reserve by a “tx_fraction” 0.1%

Then “realize” a price slippage of approximately 0.01%

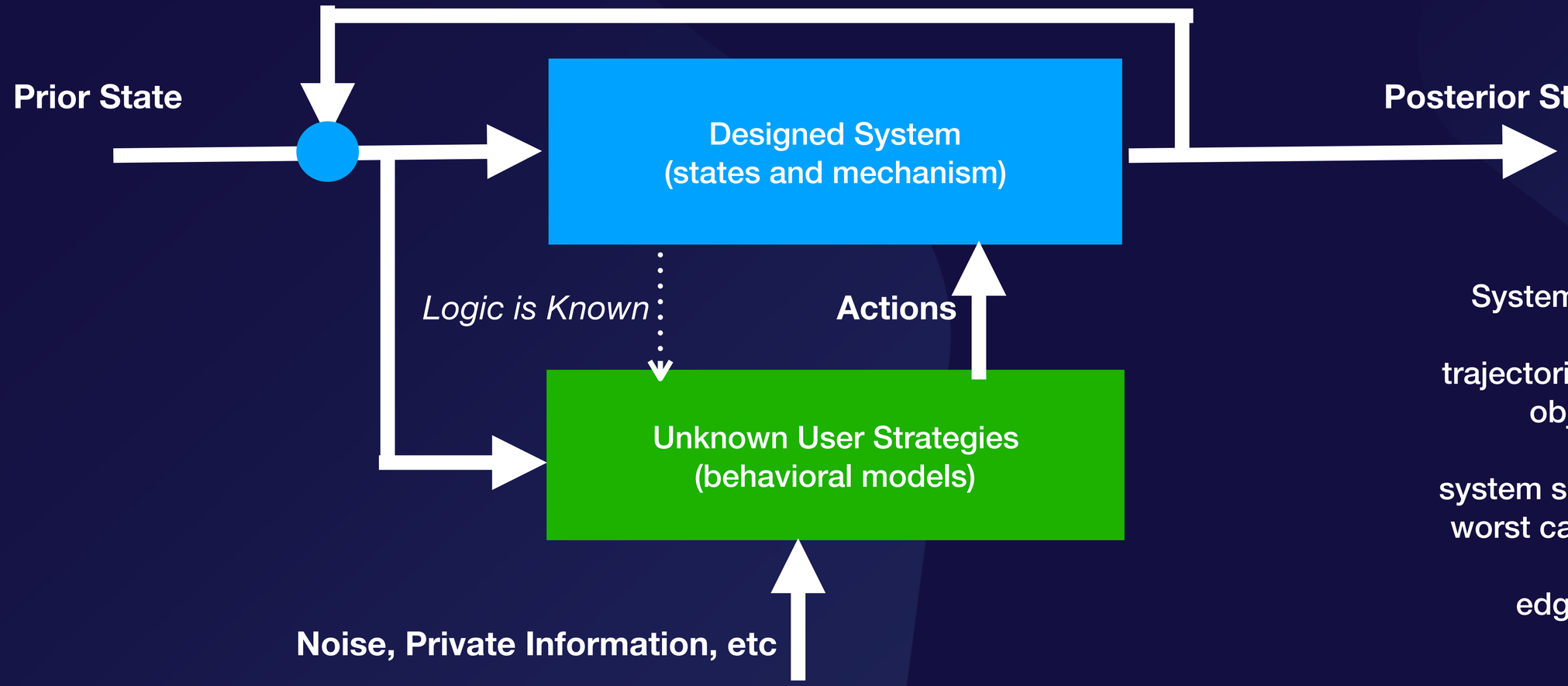
That is to say the effective price of my new tokens is 1 Basis Point greater than the spot price

More context on question being asked

BLOCKSCIENCE


System Requirements: Global Goals
Emergence Chasm
Validation of Emergent System Properties
System Architecture (decompose into subsystems)
Subsystem Requirements (decompose into components)
Component Requirements (including dependencies)
Design Iterations
Component Designs
Component Implementation
Component Verification
Subsystem Implementation
Subsystem Verification & Integration Testing
System Testing and Tuning
Deployment

Closing the Gap: Generalized Differential Equations



System Goals met?

trajectories attracted to objectives?

system safety: bounded worst case scenarios?

edge cases ?

Formalization of design space allows explicit consideration of
both **Objectives** and **Constraints** under temporal dynamics and uncertainty

BLOCKSCIENCE

Bitcoin as a Networked Dynamical System

$$x(k+1) = A_k x(k) + B_k u(k) + \mu_k v(k).$$

Dimension of *x* is all “Accounts” — balances, positive real

Discrete time system, *k* is block height

The *A* matrix is Identity

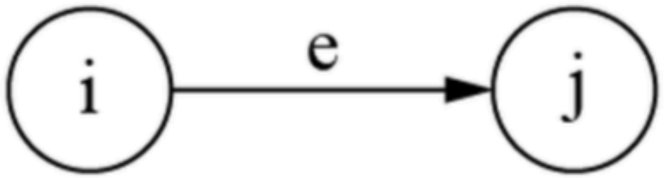
The *B* matrix is an Incidence Matrix

The Input *u* is the amount sent, (i,j)

The inflationary mining rewards μ

Stochastic vector ν defining which accounts get the reward

$$[B_k]_{ie} = \begin{cases} 1 & \text{if } e = (j,i) \text{ for any } j \\ -1 & \text{if } e = (i,j) \text{ for any } j \\ 0 & \text{otherwise} \end{cases}$$



Bitcoin Conservation and Systemic Property

No Double Spend rule is a local conserved flow constraint

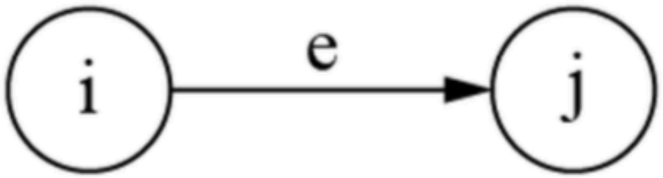
during block k. Viewed from the perspective of account i the local constraint a flow balance

$$x_i(k) + \sum_j u_{(j,i)}(k) - u_{(j,i)}(k) \geqslant 0. \tag{19}$$

In the practice, the transactions encoded by the inputs u are processed with a strict ordering that can be enforced with only the sender's state

$$u_{(i,j)} \leqslant x_i \tag{20}$$

$$[B_k]_{ie} = \begin{cases} 1 & \text{if } e = (j,i) \text{ for any } j \\ -1 & \text{if } e = (i,j) \text{ for any } j \\ 0 & \text{otherwise} \end{cases}$$

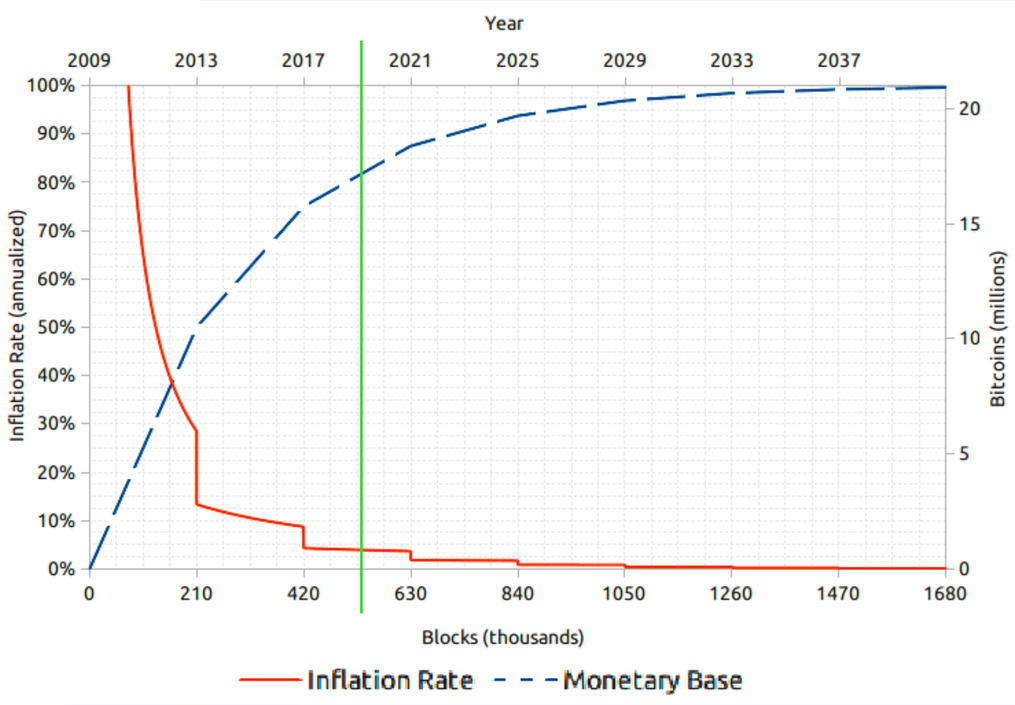


Comparing Analytical Results to a live Economic System

The local conservation Law
guarantees a global conservation property!

$$y(k) = \mathbf{1}'x(k) = \sum_i x_i(k). \quad \Rightarrow \quad y(K) = \sum_{k=1} \mu_k$$

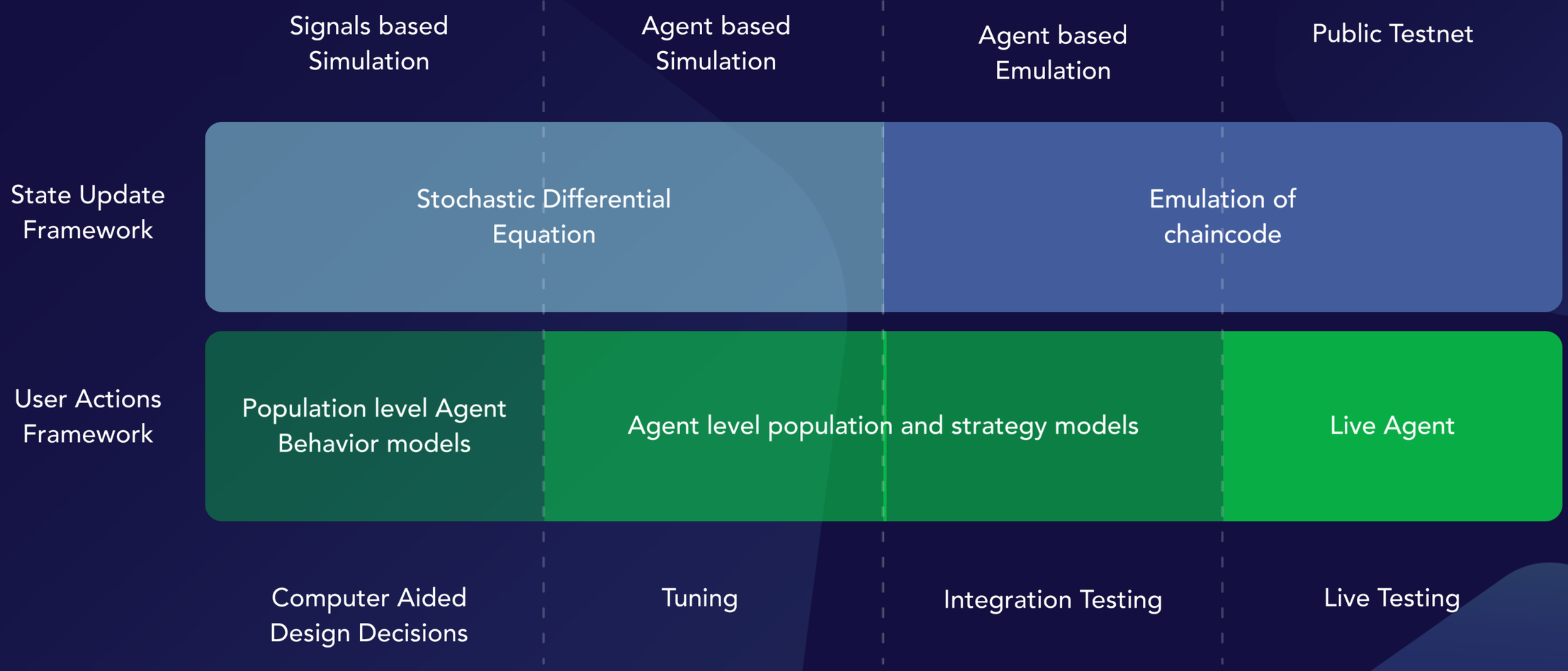
$$\mu_k = \frac{\lfloor \frac{50 \cdot 10^8}{2^i} \rfloor}{10^8} \text{where } k \in r_i \quad \Rightarrow \quad y_\infty = \lim_{k \to \infty} y(k) = \sum_{k=1}^{\infty} \mu_k$$



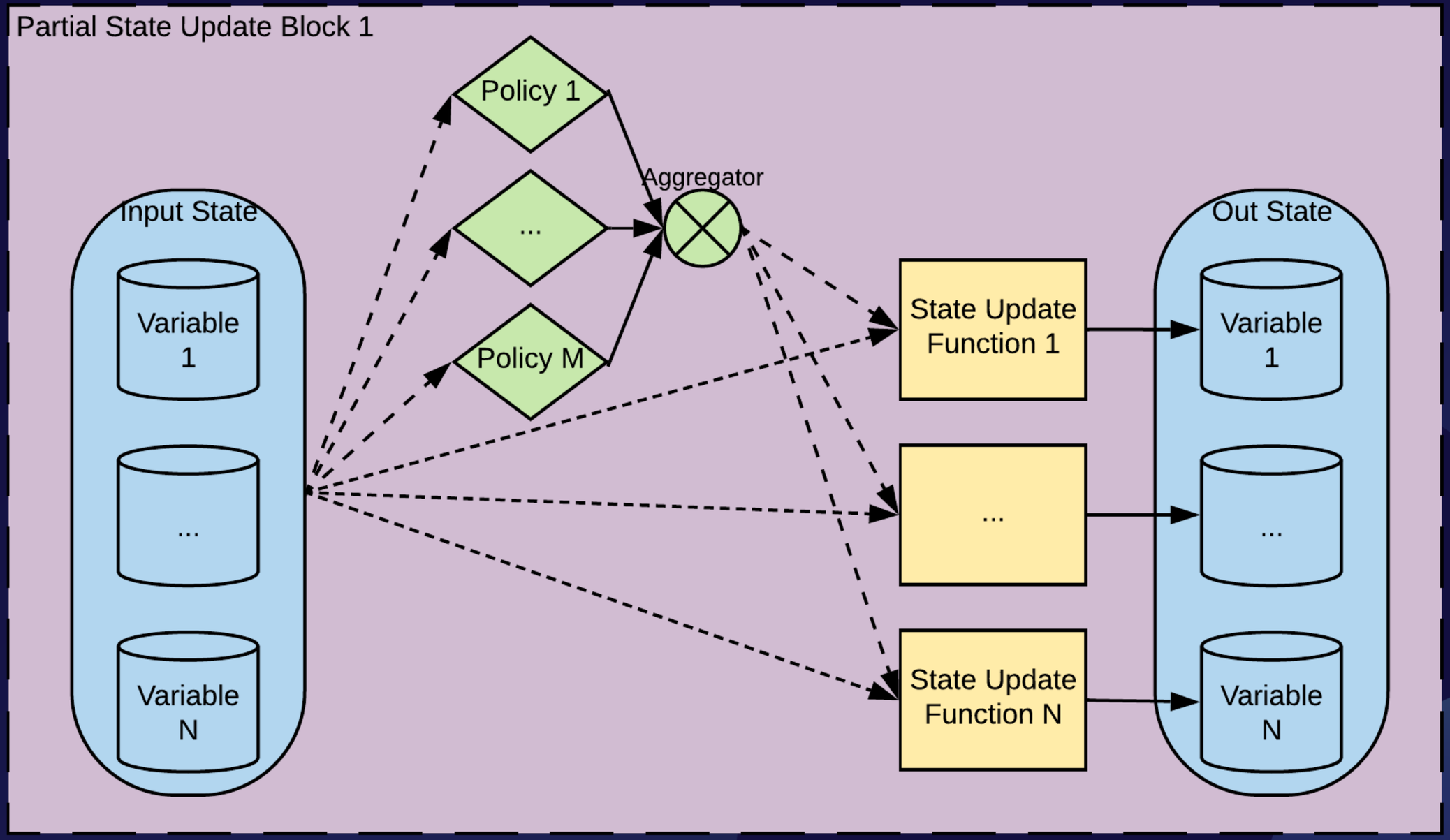
$$y_\infty = \lim_{k \to \infty} y(k) = \sum_{k=1}^{\infty} \mu_k$$ = 20999999.9769 ~ 21 Million BTC

BLOCKSCIENCE

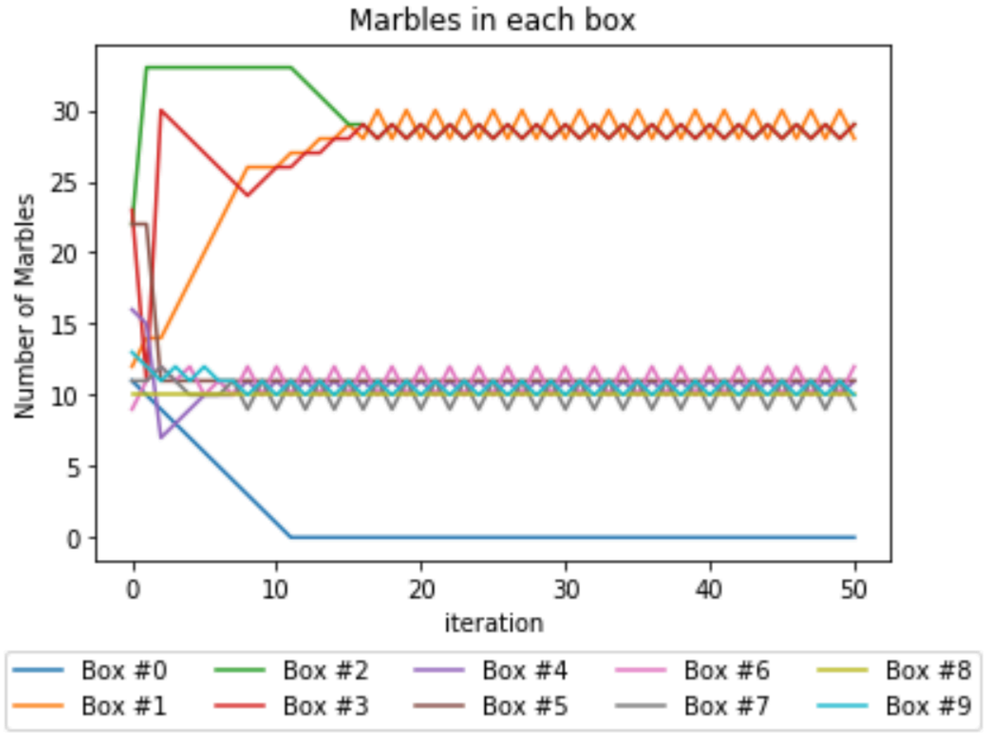
Modeling & Testing Systems with Agents in the Loop

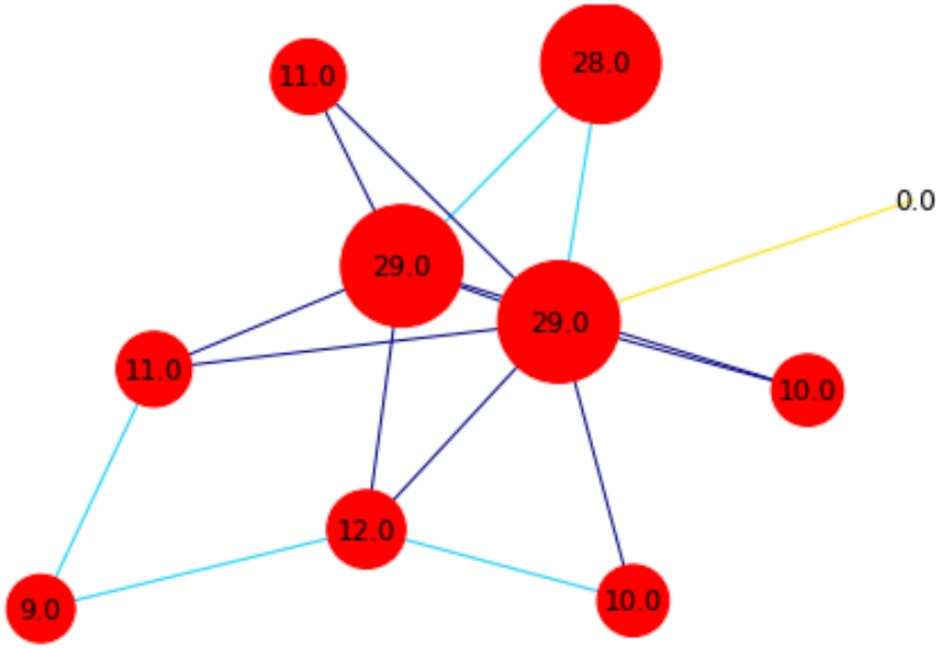


BLOCKSCIENCE



BLOCKSCIENCE





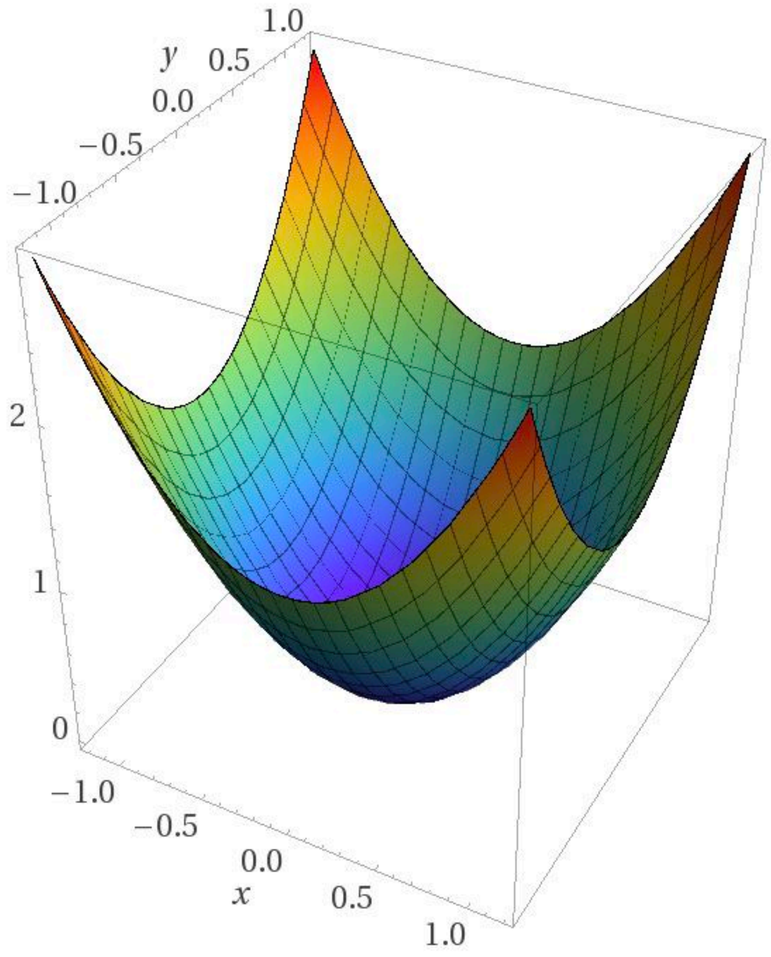
In [6]:

```python
# this time lets make three kinds of robots
def greedy_robot(src_balls, dst_balls):

    #robot wishes to accumlate balls at its source
    #takes half of its neighbors balls
    if src_balls < dst_balls:
        delta = -np.floor(dst_balls/2)
    else:
        delta = 0

    return delta

def fair_robot(src_balls, dst_balls):

    #robot follows the simple balancing rule
    delta = np.sign(src_balls-dst_balls)

    return delta


def giving_robot(src_balls, dst_balls):

    #robot wishes to gice away balls one at a time
    if src_balls > 0:
        delta = 1
    else:
        delta = 0

    return delta
```

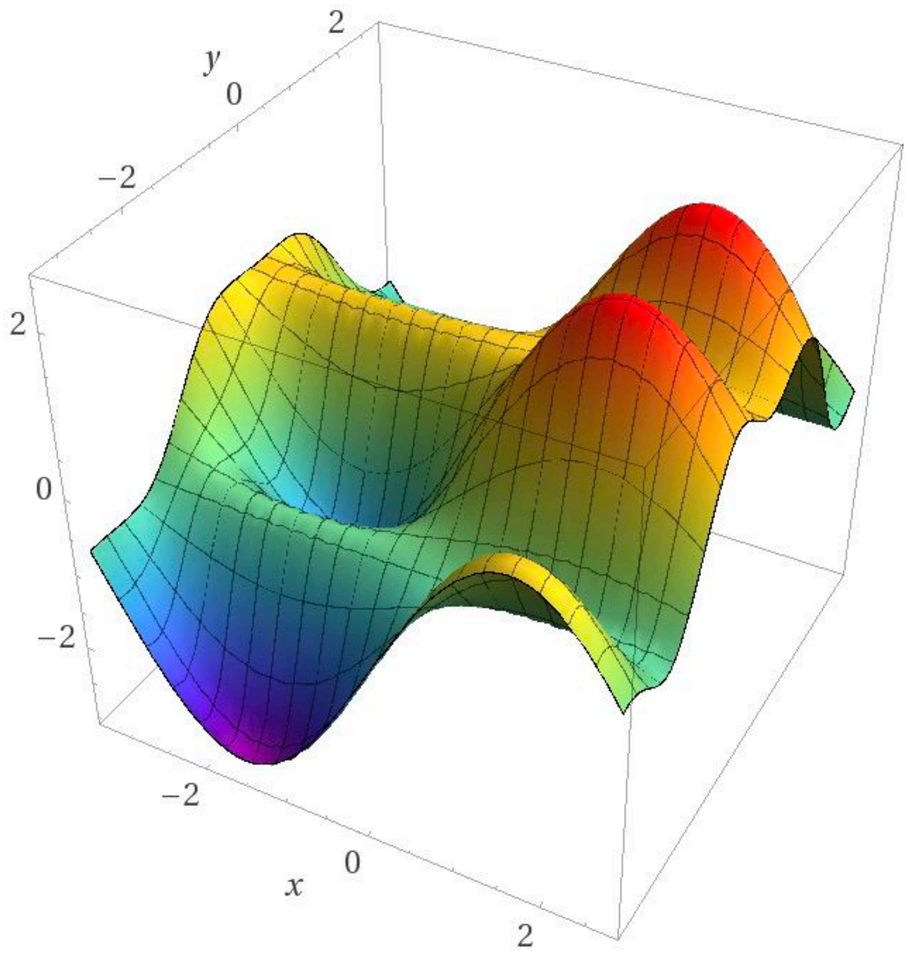
https://github.com/BlockScience/SimCAD-Tutorials/tree/master/demos/robot-marbles-network

BLOCKSCIENCE

Potential Fields = Incentive Fields?



Computed by Wolfram|Alpha



Computed by Wolfram|Alpha

BLOCKSCIENCE

Potential Based Pricing

$$V(S, R, P, \tilde{P}) = \left[\gamma(P - \tilde{P})^2 - \log\left(\frac{R}{S}\right) \right]_+$$

State Space:

$(S, R, P, \tilde{P}, \hat{P})$

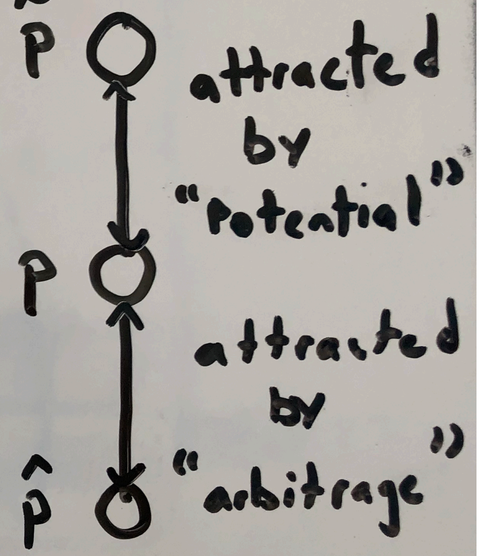
Supply S

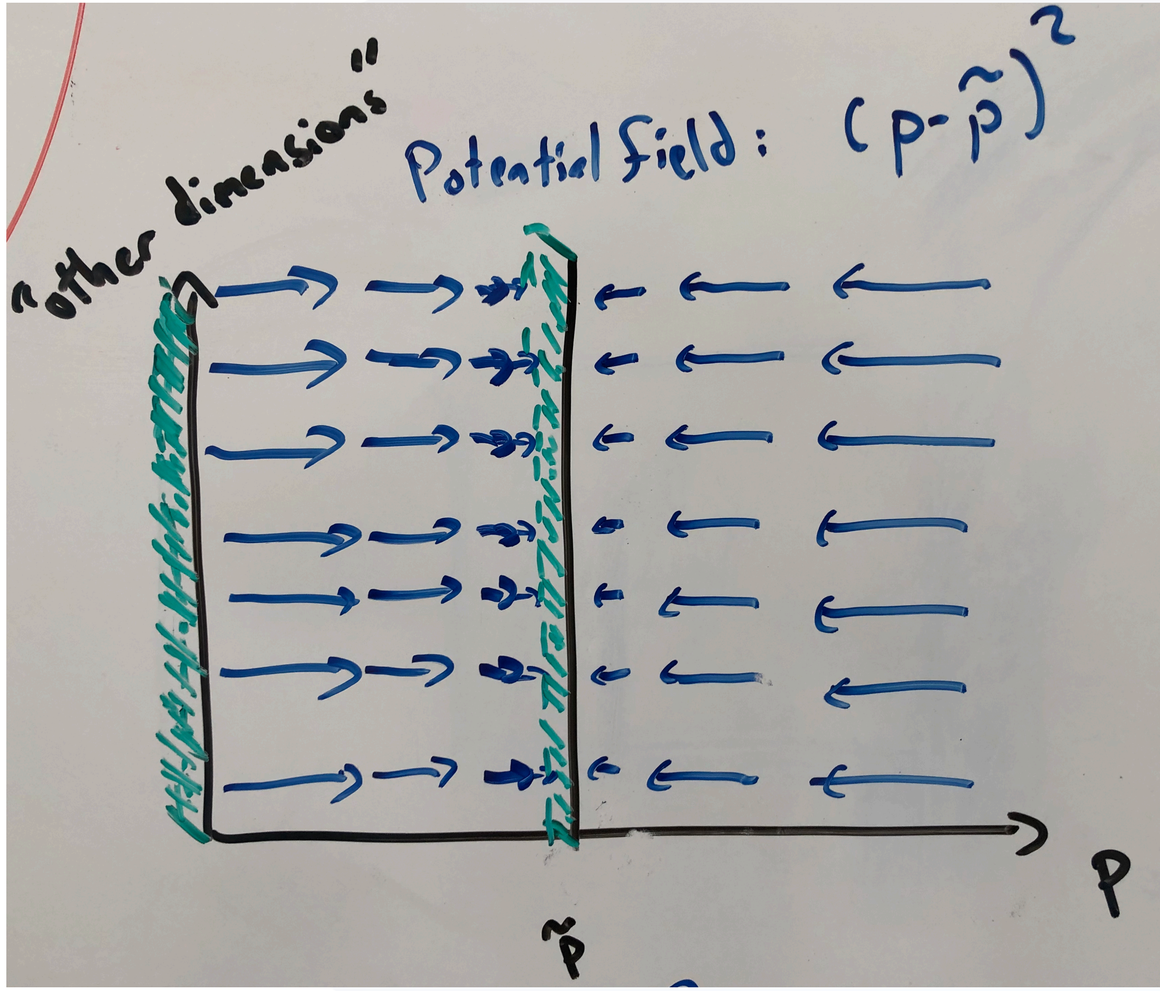
Reserve R

Spot Price P

Smooth Avg Price $\tilde{P}$

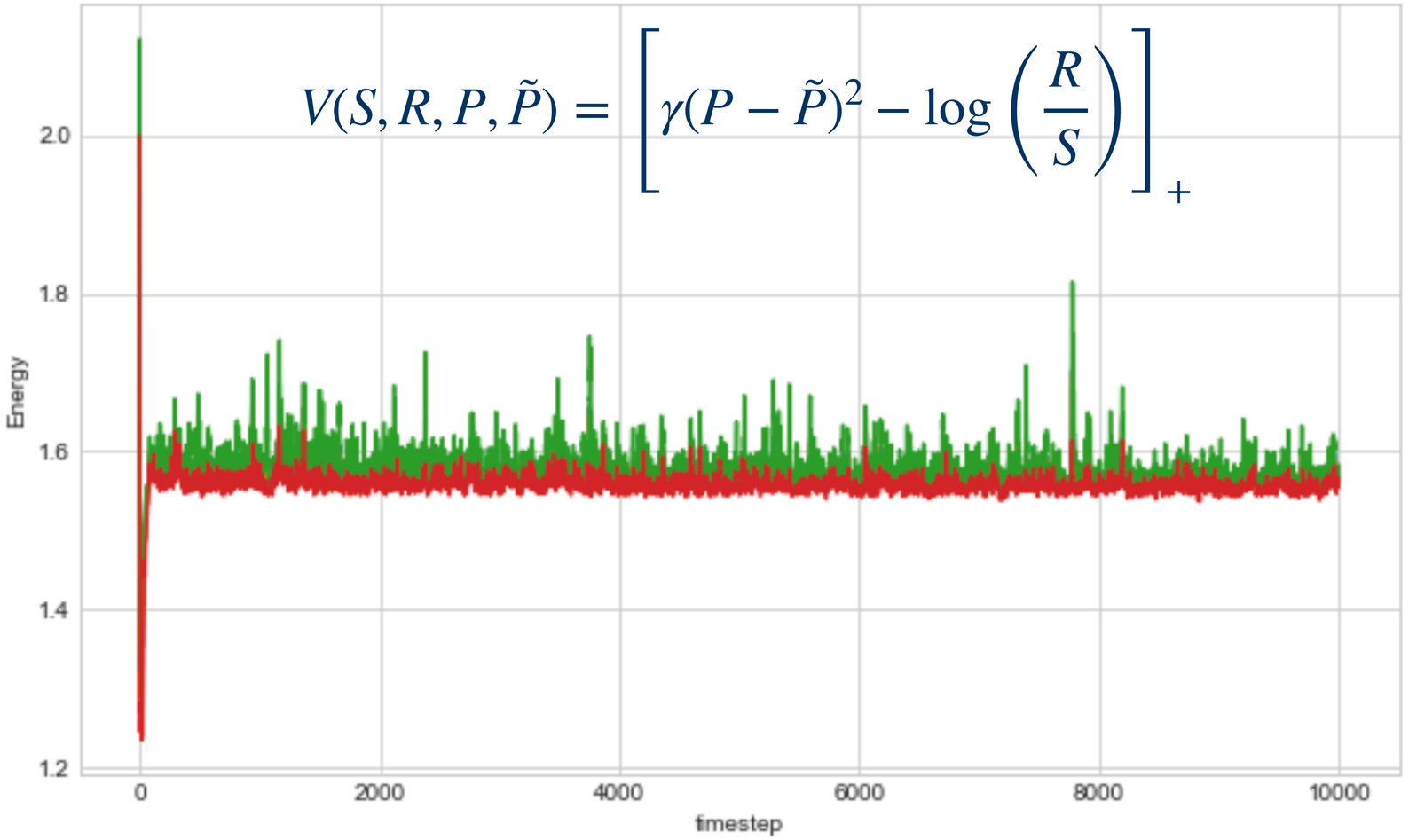
Ext. Market Price $\hat{P}$





Stabilization through Energy Dissipation

$$V(S, R, P, \tilde{P}) = \left[\gamma(P - \tilde{P})^2 - \log\left(\frac{R}{S}\right)\right]_+$$



Detailed Example:

System Initialized

Price P =1
Attractor $\tilde{P}$=0.01

Behavior is randomized but is distributed around the belief

$$\mathbb{E}[\hat{P}_k] = 1$$

BLOCKSCIENCE

What happens to Price?

New Price is determined by action

$$P^+ = P + \Delta P(action, state) = \frac{\Delta R}{\Delta S}$$

Average is updated atomically

$$\tilde{P}^+ = (1 - \alpha)P^+ + \alpha\tilde{P}$$

New State becomes the state:

$$S \leftarrow S + \Delta S$$

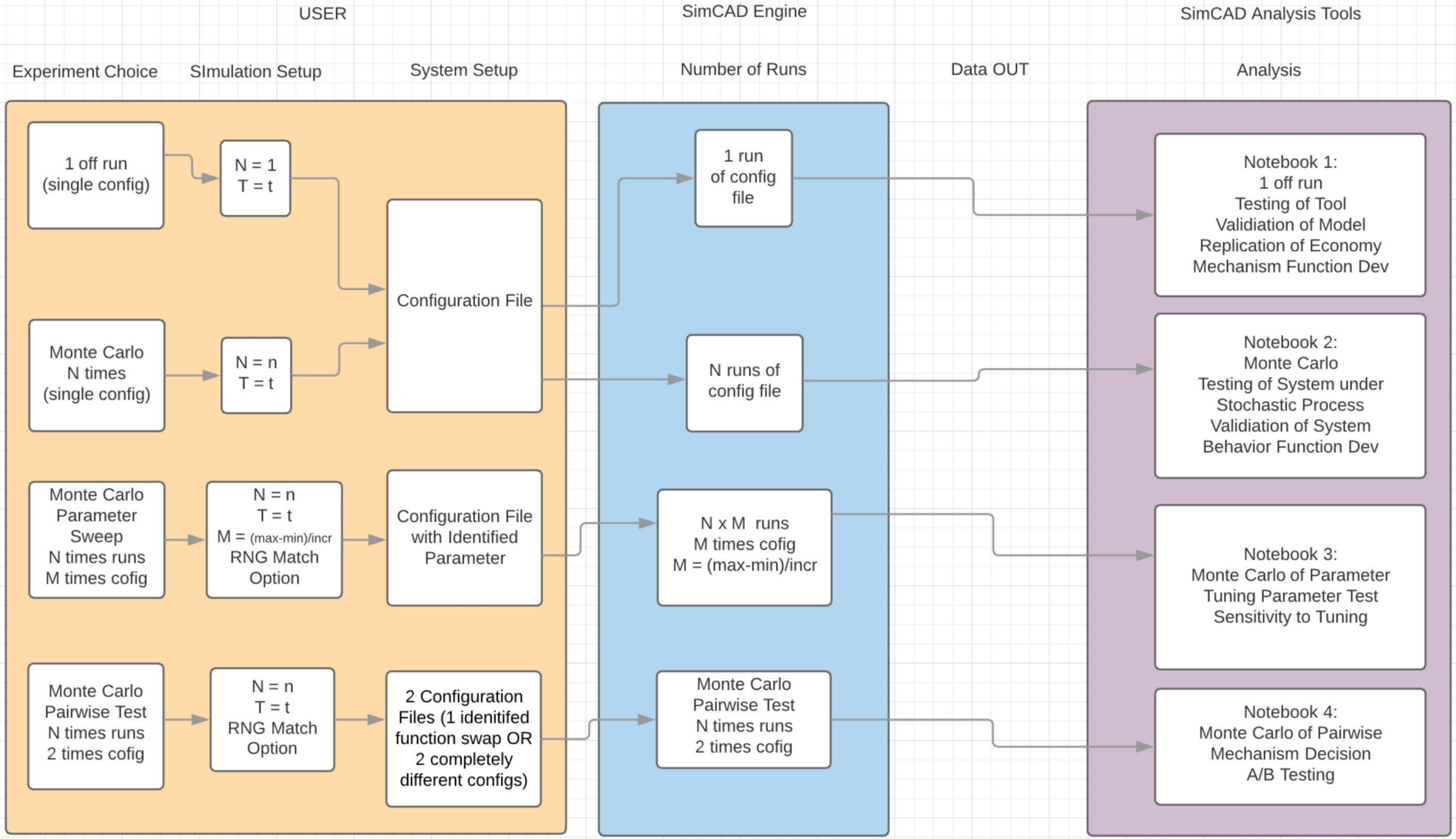
$$R \leftarrow R + \Delta R$$

$$P \leftarrow P^+$$

$$\tilde{P} \leftarrow \tilde{P}^+$$

USER-ENGINE-OUTPUT

Matt Barlin



BLOCKSCIENCE

Impositions and Consequences of Potential Field Designs?

BLOCKSCIENCE

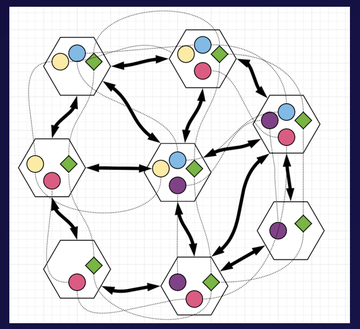
Engineering = { Capability

Responsibility

BLOCKSCIENCE

Mindful of “Subjective” Choice of “Objective” Measures

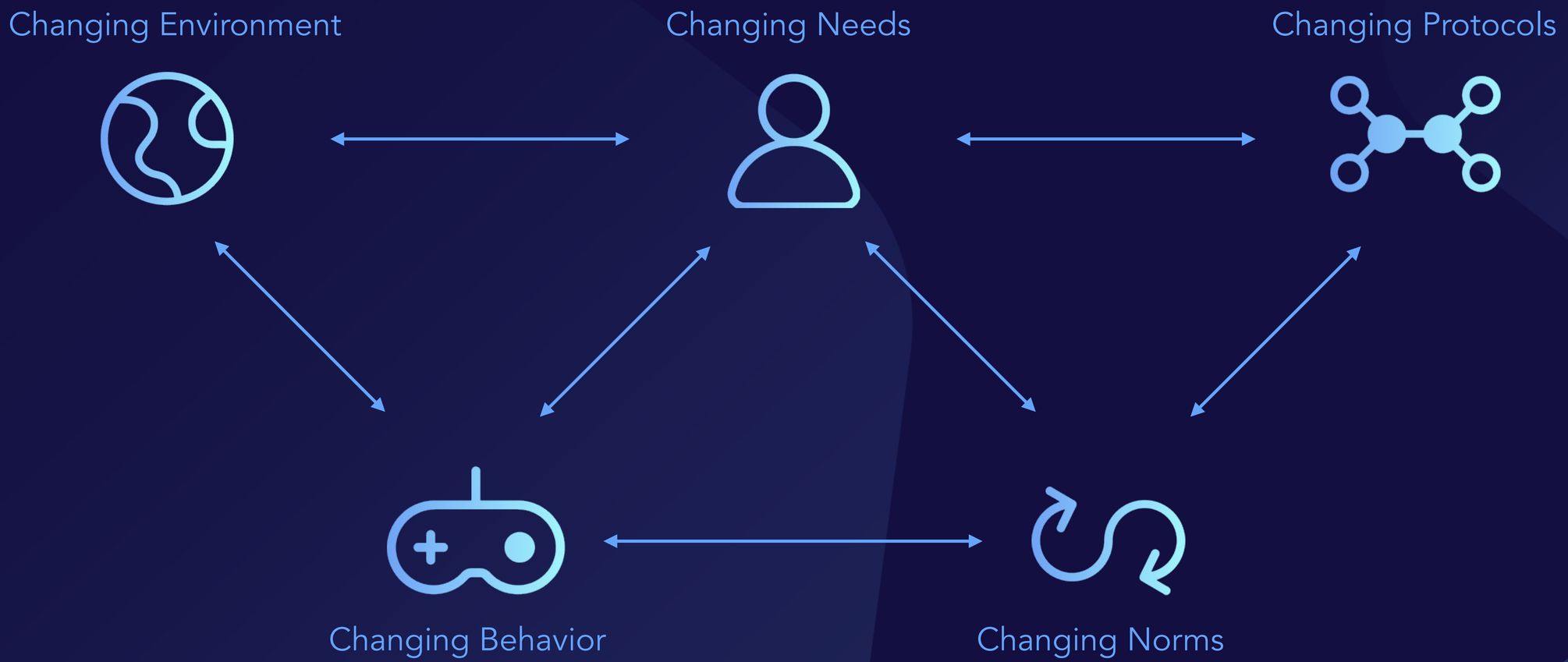
Systems



Choice of Measure is inseparable from the system itself and its outcomes

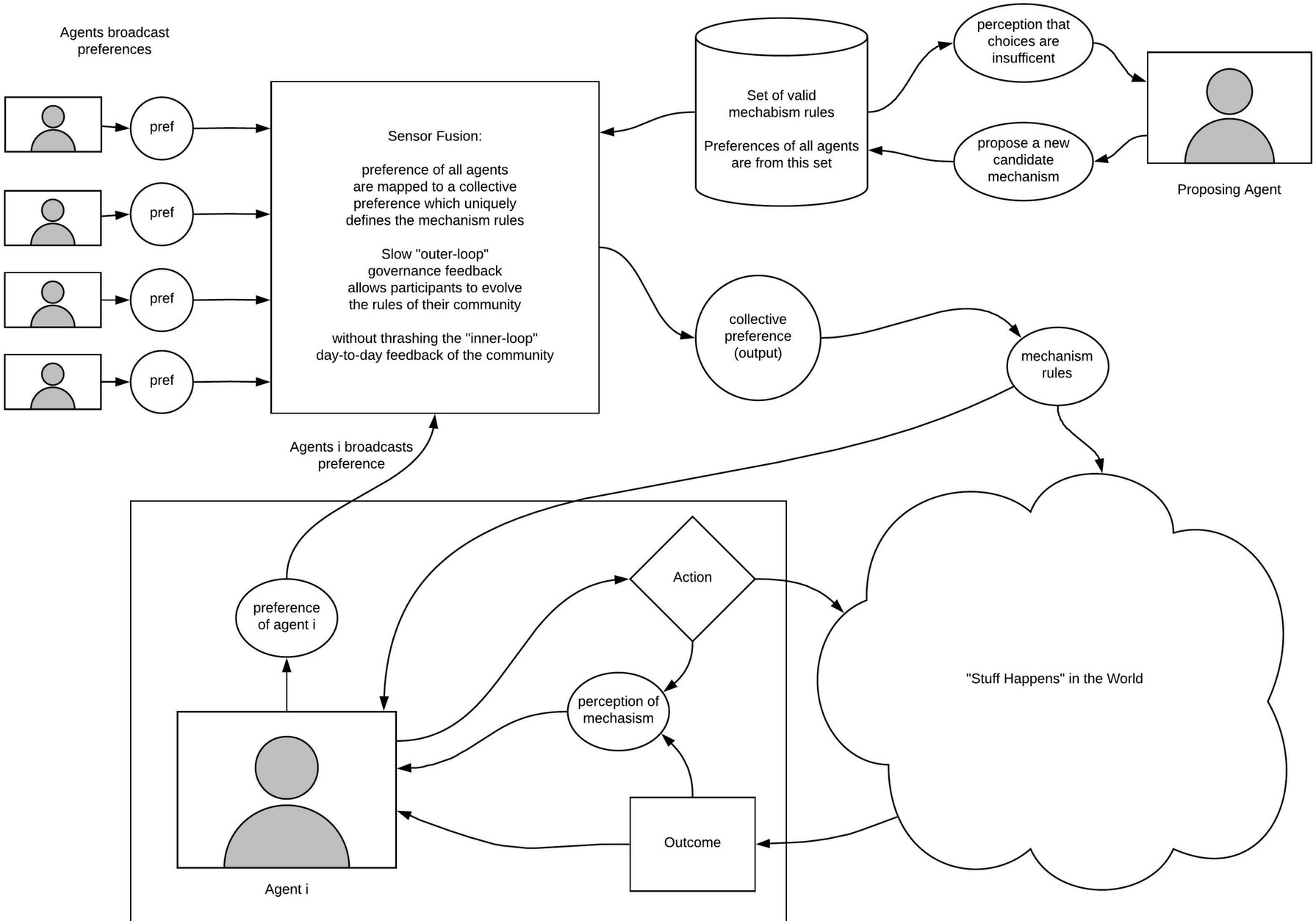
BLOCKSCIENCE

Protocol Governance as Adaptive Maintenance



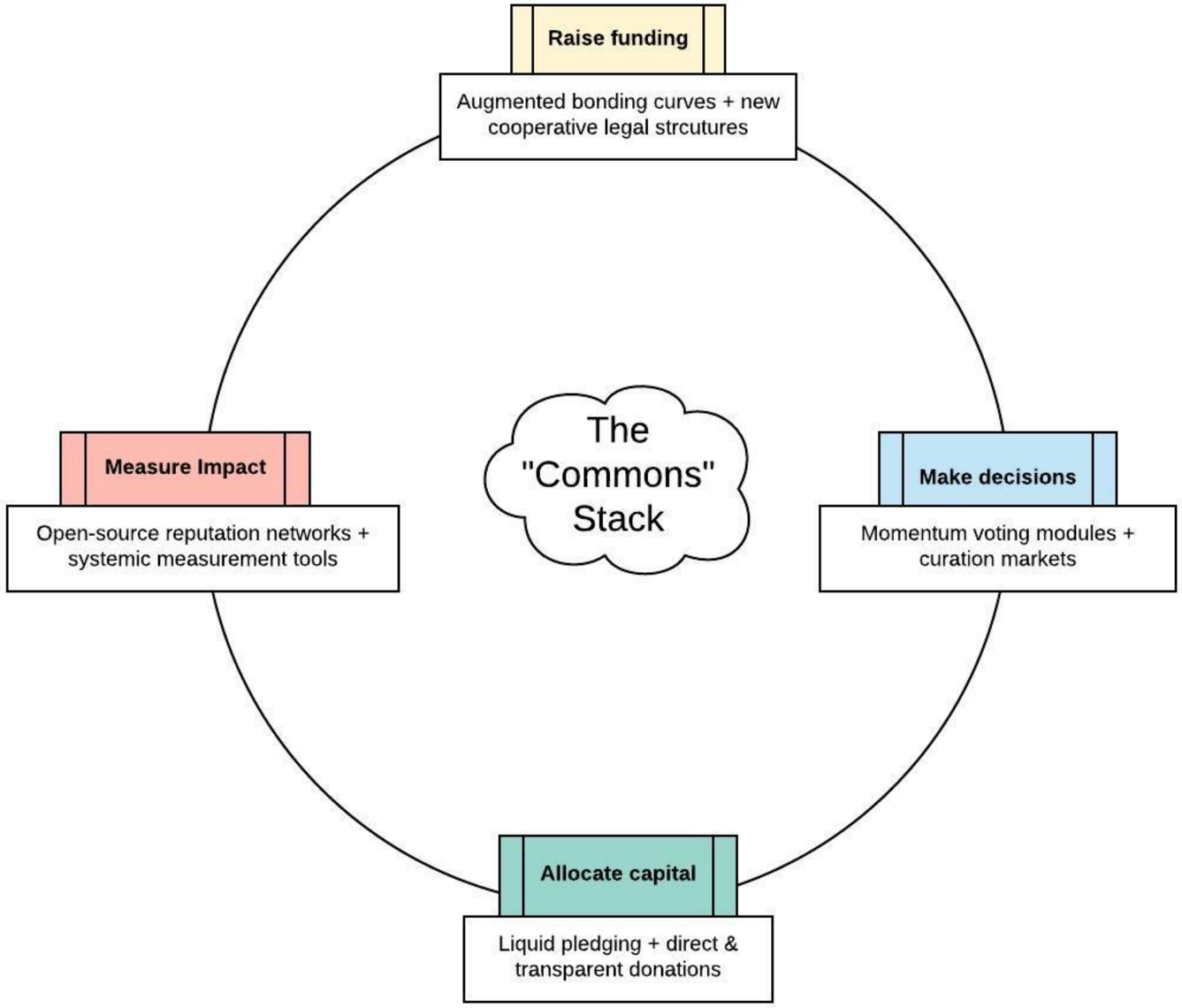
BLOCKSCIENCE

Governance as Multi-scale Adaptation



BLOCKSCIENCE

Autonomous Vs Automated



BLOCKSCIENCE

BLOCKSCIENCE

Thank you

Web page: https://block.science/
Twitter: @mZargham