Replacing the Banker with a Function

A Mathematician's Guide to Decentralized Finance

Shen-Ning Tung

National Tsing Hua University

NCCU Mathematics Colloquium

Outline

I. The Global Picture — Architecture & Vision

"Trust is not a leap of faith; it is the deterministic output of a distributed state machine."

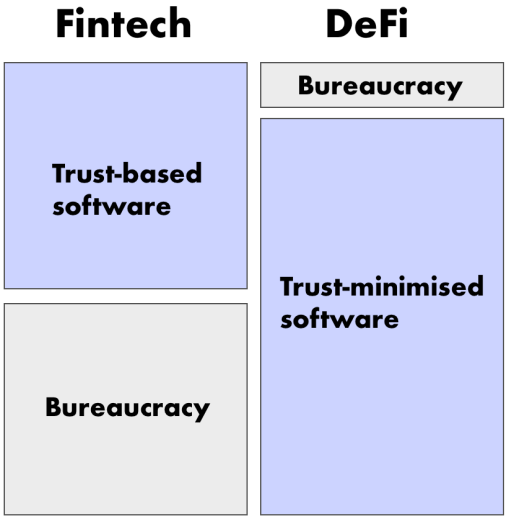
Defining the Transformation: Fintech vs. DeFi

Before we dive into the math, we must distinguish between *digitalising* a bank and *rebuilding* finance.

- **Fintech (Digitised TradFi):** Centralised institutions using modern tech stacks. You have an app, but the bank still holds the *Master Ledger* and the *Kill Switch*.
- **DeFi (Decentralised Finance):** A peer-to-peer ecosystem where *Smart Contracts* act as the custodian and clearinghouse.
- **The Paradigm Shift:** Moving from *Subjective Trust* (Law/Reputation) to *Objective Truth* (Cryptographic Proofs).

References: What is DeFi? | DeFi Beyond the Hype (Wharton) | CCAF DeFi Ecosystem Map

Defining the Transformation: Fintech vs. DeFi



The Blockchain as a Global State Machine

For a mathematician, a blockchain is simply a **distributed, deterministic transition system**.

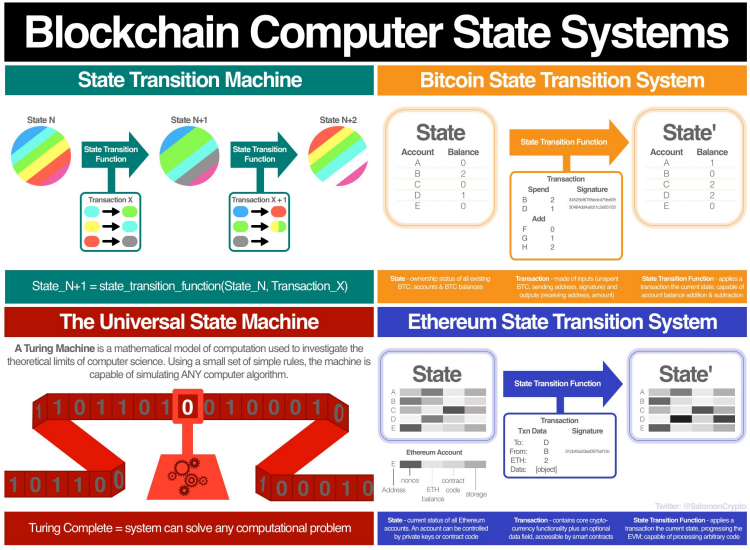
- **The State** (S): A snapshot of all balances and contract data at time t.
- **The Transition Function** (f): The rules of the protocol (the code).
- **The Formula:**

$$S_{t+1} = f(S_t,\ \Delta)$$

  where Δ represents a block of transactions.
- **Integrity**: Thousands of nodes must compute the same S_{t+1} and reach consensus.

References: What is a Blockchain? | Ethereum & Bitcoin Tx Visualiser | Dan Boneh: Blockchain Primitives

The Blockchain as a Global State Machine

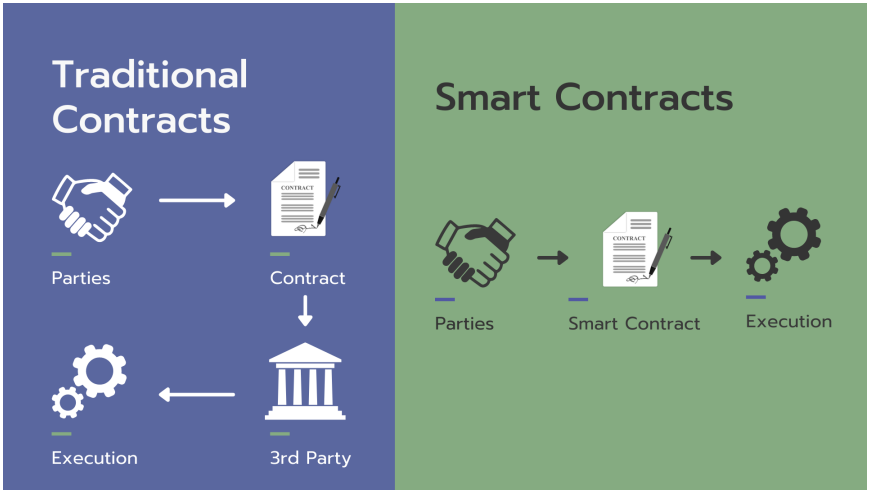


The Engine of DeFi: Ethereum & The EVM

To turn a ledger into a financial system, we need a **Universal Computer** capable of executing arbitrary logic.

- **Smart Contracts:** Self-executing programs whose terms are written directly into code. They are *autonomous* (no middleman) and *immutable* (cannot be changed once deployed).
- **The Ethereum Virtual Machine (EVM):** A global, decentralised CPU ensuring every node executes the same code and arrives at the same result—guaranteeing *mathematical determinism*.
- **Gas & the Halting Problem:** Every operation costs "Gas"—a resource allocation mechanism and a formal constraint on computation in a Turing-complete system.

References: What is Ethereum? | What Are Smart Contracts?


Traditional Contracts
CONTRACT
Parties
Contract
Execution
3rd Party
Smart Contracts
CONTRACT
Parties
Smart Contract
Execution

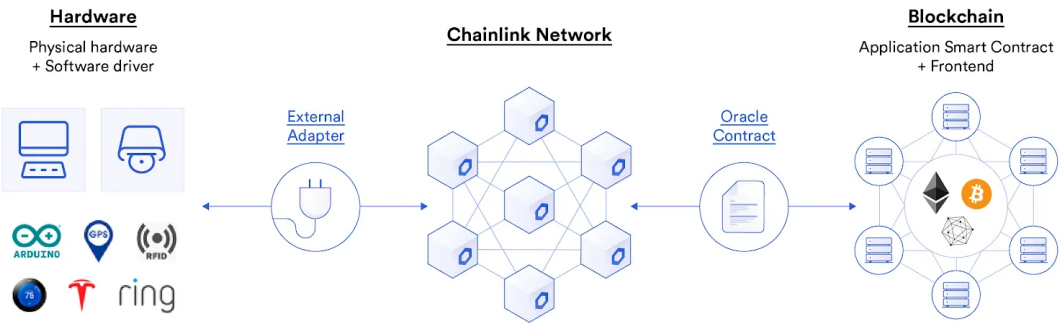
Oracles: The Bridge to Reality

A closed mathematical system cannot natively "know" the price of Gold or the outcome of a game.

- **The Oracle Problem:** How do we bring external data into a deterministic system without introducing a single point of failure?
- **Decentralised Oracles:** Systems like *Chainlink* use a committee of nodes to provide a "consensus" data feed.
- **Application:** A smart contract can trigger a *liquidation* or a payout based on real-world events (e.g. ETH price dropping below \$1800).

References: What Is a Blockchain Oracle? | What Is Chainlink?

Oracles: The Bridge to Reality



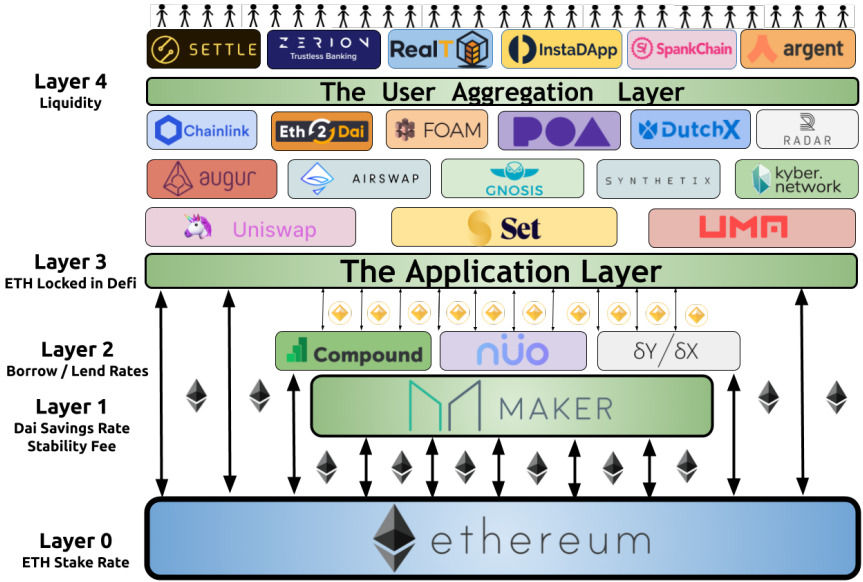
Summary: The Architecture of Integrity

Feature	Traditional System	DeFi System
Enforcement	Legal Prose (Ambiguous)	Mathematical Code (Precise)
Custodian	Centralised Bank	Self-Custody / Smart Contract
Verification	Post-hoc Audits	Real-time Cryptographic Proofs
Interoperability	Siloed APIs	Open Composability ("Money Legos")

The "Money Lego" Framework

DeFi protocols are modular by design. Layer 1 provides settlement, tokens serve as raw materials, and lending/trading protocols act as building blocks. Developers "snap" these bricks together to create complex strategies without seeking institutional permission.

Summary: The Architecture of Integrity



II. Market Mechanics — How DeFi Works

"Liquidity is an invariant, and price is the slope of a curve defined by the balance of reserves."

Beyond Order Books: Automated Market Makers (AMMs)

In traditional markets, a trade requires a counterparty (Buyer $\leftrightarrow$ Seller). In DeFi, you trade against a **Liquidity Pool**.

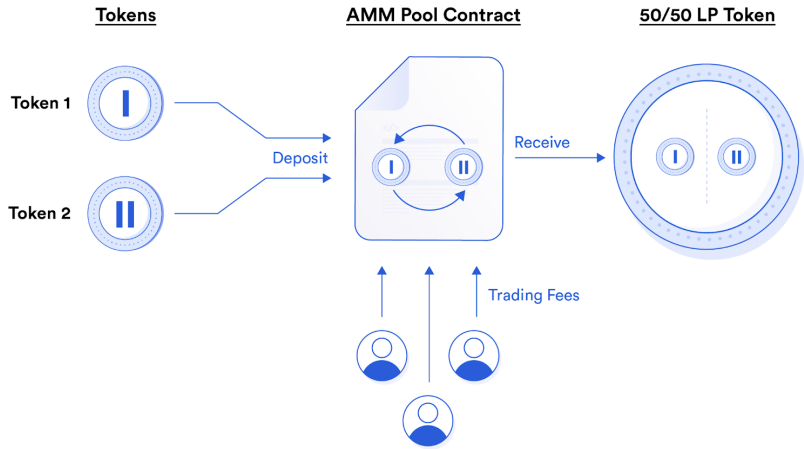
- **The Liquidity Pool:** A smart contract holding a pair of tokens (e.g. X and Y).
- **The Constant Product Invariant:**

$$x \cdot y = k$$

  where x, y are token reserves and k is a constant maintained during a trade.
- **The "Peer-to-Pool" Model:** Anyone can become a *Liquidity Provider (LP)* by depositing assets, earning a pro-rata share of trading fees.

References: What Are AMMs? | Mastering AMMs (Three Sigma)

Beyond Order Books: Automated Market Makers (AMMs)

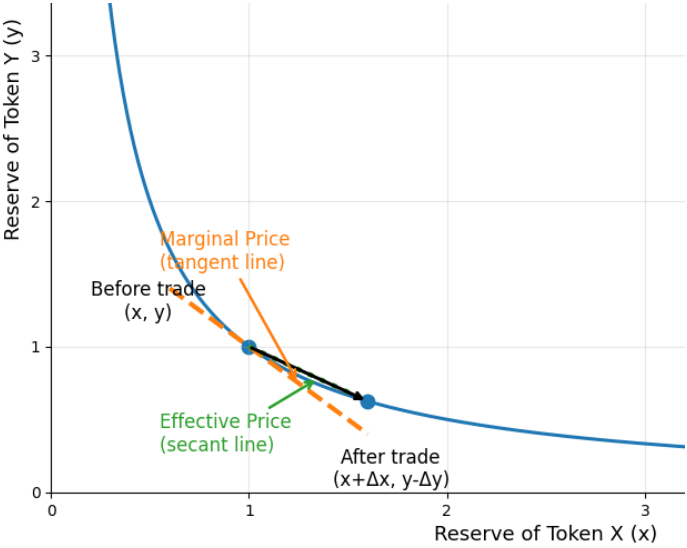


The Geometry of Price Discovery

Price is not "set" by an admin; it is a **mathematical consequence** of the reserve ratio.

- **Marginal Price:** For $y = k/x$,
$$P = \frac{dy}{dx} = -\frac{y}{x}$$
- **Slippage:** Because the curve is a hyperbola, every trade shifts y/x.
  - *Infinitesimal trades:* Price stays near spot y/x.
  - *Large trades:* Significant displacement along the curve.
- **Asymptotic Liquidity:** The pool can never be fully depleted—as quantity $\to 0$, price $\to \infty$.

The Geometry of Price Discovery



Worked Example: Trading on the Curve

Setup

Pool holds 100 ETH (x) and 200,000 USDC (y).
$k = 100 \times 200{,}000 = 20{,}000{,}000$. Spot price $P = y/x = 2{,}000$ per ETH.

Trade: A trader wants to buy 10 ETH.

- New ETH reserve: $x' = 100 - 10 = 90$.
- New USDC reserve: $y' = 20{,}000{,}000 \,/\, 90 = 222{,}222.22$.
- Cost: $222{,}222.22 - 200{,}000 = 22{,}222.22$ USDC.

Result

Effective price: \$2,222.22 per ETH $\implies$ $\approx 11.1\%$ slippage from spot.

Reference: Uniswap University

Algorithmic Lending & Money Markets

Protocols like **Aave** replace loan officers with a *Utilisation-based Interest Rate Model*.

- **Utilisation Rate:**

$$U = \frac{\text{Total Borrows}}{\text{Total Liquidity}}$$

- **Equilibrium Logic:**
  - Low U: Excess supply $\Rightarrow$ low rates to incentivise borrowing.
  - High U: Scarcity $\Rightarrow$ sharp spikes to attract lenders and force debt repayment.

The Kinked Interest Rate Model

The rate function $R(U)$ is piecewise linear around an *Optimal Utilisation* U_{opt}:

Healthy Zone ($U < U_{\text{opt}}$)

$$R_t = R_0 + \left(\frac{U_t}{U_{\text{opt}}}\right) R_{\text{slope1}}$$

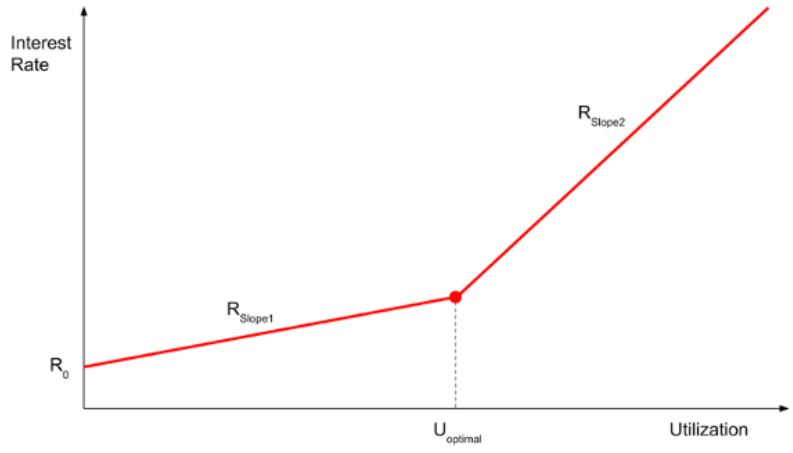
Emergency Zone ($U \geq U_{\text{opt}}$)

$$R_t = R_0 + R_{\text{slope1}} + \left(\frac{U_t - U_{\text{opt}}}{1 - U_{\text{opt}}}\right) R_{\text{slope2}}$$

The "Kink": R_{slope2} is usually an order of magnitude higher than R_{slope1} (e.g. jumping from 4% to 100%)—a mathematical "emergency brake" to prevent pool depletion.

References: DeFi Money Markets 2024 | Aave Docs | Compound Docs

The Kinked Interest Rate Model



Risk Management: Over-collateralisation

In a pseudonymous system without legal recourse, all credit is secured by a **Collateral Invariant**.

- **Collateralisation Ratio:**

$$CR = \frac{\sum(\text{Collateral Assets} \times \text{Price})}{\text{Total Debt Value}}$$

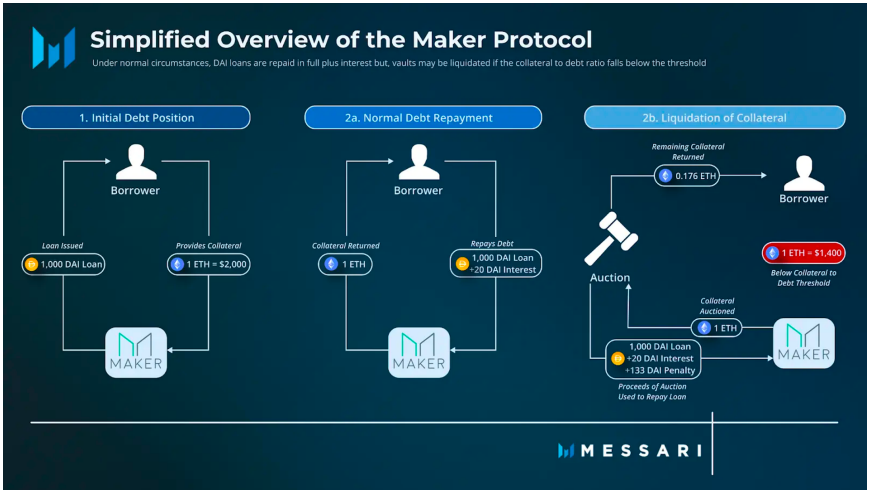
- **Loan-to-Value (LTV):** Maximum leverage allowed (e.g. $\text{LTV} = 75\%$ implies a minimum CR of 133%).
- **Oracle Dependency:** System safety is strictly bound to the accuracy and latency of external price feeds.

The Liquidation Function

The "fail-safe" mechanism that maintains system-wide solvency.

- **Trigger:** If $CR <$ Liquidation Threshold, the position is flagged as under-collateralised.
- **Forced Exit:** Automated "Liquidator" bots repay the debt in exchange for the collateral at a *Liquidation Bonus* (e.g. 5%–10% discount).
- **Mathematical Solvency:** By liquidating before $CR < 100\%$, the protocol ensures that every dollar of debt is always backed by more than one dollar of market value.

Maker Portocol



Liquidation Walkthrough: A "Hard Money" Event

1. **Open Position:** Deposit 1 ETH ($2,000). Borrow 1,500 USDC ($\mathrm{LTV} = 75\%$).
2. **Current State:** $CR = 2{,}000/1{,}500 = 133.3\%$ (Healthy).
3. **Market Shock:** ETH drops to $1,700.
4. **Breach:** $CR = 113.3\%$. Threshold is 120% $\Rightarrow$ position is now **liquidatable**.
5. **Execution:** A bot detects the breach, repays 1,500 USDC, and claims the ETH at a discount.

Takeaway

Debt is cleared by code, not by credit committees or bailouts. The math is the ultimate arbiter of solvency.

References: What Are Stablecoins? | Sky Protocol: Multi-Collateral Dai

III. Synthesis — Mechanism Design & The Future

"The bedrock of future finance is not the institution,
but the formally verified proof."

Mechanism Design: Engineering Rational Behaviour

In a permissionless system, we cannot "ban" bad actors; we must make it **mathematically unprofitable** to be one.

- **Agent-Based Economics:** Model the protocol as a game where autonomous agents maximise their own utility.
- **Slashing & Incentives:**
  - *Positive:* Yield and rewards for honest participation.
  - *Negative:* Malicious behaviour results in automated forfeiture of staked capital (*Slashing*).
- **Sybil Resistance:** Proof of Stake imposes an economic cost that makes it prohibitively expensive to subvert the system via multiple identities.

References: 8 Reasons Why Blockchain Mechanism Design Is Hard | Permissionless Mechanism Design

When the Math Breaks: The Terra/Luna Collapse

In May 2022, the Terra ecosystem provided a \$40 billion case study in **mechanism failure**.

- **The Setup:** TerraUSD (UST) was an *algorithmic stablecoin* backed by a game-theoretic arbitrage loop with LUNA.
- **The Death Spiral:**

$$\text{UST} < \$1 \Rightarrow \text{burn UST, mint LUNA} \Rightarrow \text{LUNA hyper-inflates} \Rightarrow \text{UST loses backing}$$

- **The Lesson:** The system was *locally stable* but *globally fragile*. It assumed a constant presence of rational arbitrageurs and failed to account for "bank run" dynamics.

Reference: Luna Brothers, Inc. (BitMEX)

Privacy via Zero-Knowledge Proofs (ZKP)

The Transparency Paradox: Blockchains are public, but finance requires privacy.

- **ZKP Definition:** A cryptographic method where a *Prover* convinces a *Verifier* that a statement is true (e.g. "I am solvent") without revealing the underlying data (e.g. "my balance").
- **Mathematical Applications:**
  - *Private Credit:* Prove $\mathrm{Collateral} > \mathrm{Debt}$ without revealing wallet history.
  - *Identity:* Prove you are verified or >18 without sharing your name.
  - *Scalability (zk-Rollups):* Bundle 10,000 transactions into a single succinct proof verifiable in milliseconds.

References: ZKP: An Illustrated Primer | Decentralised Identity Guide 2025

Open Problems: The Frontiers of DeFi

An invitation for students to apply advanced mathematics to unsolved challenges:

- **Formal Verification:** Using formal methods (e.g. TLA$^+$, Coq) to *prove* a smart contract is bug-free before it handles billions of dollars.
- **Systemic Risk & Contagion:** Modelling the "Money Lego" effect—how the failure of one protocol propagates through the hyper-connected liquidity network.
- **MEV (Maximal Extractable Value):** Studying the game theory of transaction ordering to prevent sophisticated bots from front-running users.

Your Career: Beyond the Classroom

Digital Finance is a rare field where **pure math translates directly into production infrastructure**.

- **Protocol Architect:** Designing the next generation of invariant curves and interest rate models.
- **Risk Quantitative (Quant):** Running Monte Carlo simulations for firms like Gauntlet or Chaos Labs to stress-test protocol parameters.
- **The Opportunity:** You are at the intersection of *Cryptography*, *Game Theory*, and *Quantitative Finance*. The "Wild West" era is ending; the *Era of Rigor* is beginning.

Final Closing Thought

"Digital Finance is not about 'disrupting' banks;
it is about building a financial system that is as predictable
and transparent as the laws of mathematics."

Thank you!

Scan for slides