Lesson 06 — Decentralized Finance (DeFi)

Study Notes

Cryptoeconomics — BSc Level

Joerg Osterrieder

2025

Contents

Learning Objectives

By the end of this lesson, students should be able to:

1. **Define** DeFi and explain how it recreates traditional financial services using smart contracts on public blockchains.
2. **Derive** the constant-product AMM formula $x \cdot y = k$ and calculate prices, slippage, and impermanent loss for given trade sizes.
3. **Explain** how lending protocols (Aave, Compound) determine supply and borrow rates via utilization-based interest rate models.
4. **Analyze** liquidation mechanics including health factors, liquidation penalties, and cascade scenarios.
5. **Evaluate** the role of oracles in DeFi and the manipulation risks associated with on-chain price feeds.
6. **Describe** flash loans, their atomic execution model, and their use cases and attack vectors.
7. **Calculate** APY for yield farming strategies and assess the sustainability of liquidity mining rewards.
8. **Interpret** EIP-1559 gas economics including base fee, priority fee, and the implications for miners and users.

DeFi Ecosystem Overview

Decentralized Finance (DeFi)

DeFi refers to financial services—trading, lending, borrowing, derivatives, insurance—built on public blockchains using smart contracts, operating without traditional intermediaries such as banks or brokers. Key characteristics: *permissionless* (anyone with a wallet can participate), *transparent* (all transactions on-chain), *composable* (protocols can be combined), and *non-custodial* (users control assets).

The DeFi Stack

DeFi is organized as a layered architecture:

Layer	Description
Settlement	Underlying blockchain (Ethereum, Solana, Avalanche)
Asset	Tokens: ETH, ERC-20 stablecoins, wrapped assets
Protocol	Smart contracts implementing AMMs, lending, derivatives
Application	User-facing dApps (Uniswap interface, Aave dashboard)
Aggregation	Cross-protocol routing and optimization (1inch, Yearn)

Composability and Money Legos

Composability means any protocol can call any other protocol within a single transaction. A yield aggregator (Yearn) can deposit into a lending protocol (Aave), which issues aTokens that can be used as collateral in a third protocol—all atomically. This modularity is often called **money legos**.

- **Benefit**: Rapid innovation; new products combine existing primitives without starting from scratch.
- **Risk**: Cascading failures—a bug in one protocol can propagate to all protocols that depend on it.

Key Metrics

Total Value Locked (TVL)

TVL is the aggregate value of crypto assets deposited in DeFi protocols at a given time. It is the standard measure of DeFi ecosystem size. Note: TVL is denominated in USD and therefore changes with both deposit quantity and asset prices.

- **DEX (Decentralized Exchange)**: A peer-to-peer trading venue that operates via smart contracts, with no central authority holding order books or custody of funds.
- **Liquidity Pool (LP)**: A smart contract holding reserves of two or more tokens that traders swap against.

Automated Market Makers

The Constant-Product Formula

Traditional exchanges use *order books*: buyers and sellers post bids and asks; trades execute when orders match. AMMs replace order books with a *liquidity pool* and an *invariant*.

Constant-Product AMM

Let x be the pool reserve of token A and y the reserve of token B. The **constant-product invariant** requires $x \cdot y = k$ (k constant). A trader who buys Δx of token A must deposit Δy of token B such that $(x - \Delta x)(y + \Delta y) = k$, giving: $\Delta y = \frac{k}{x - \Delta x} - y = \frac{y \cdot \Delta x}{x - \Delta x}$. The marginal price of A in terms of B is $P_A = y/x$.

Uniswap Mechanics

Uniswap v2 (launched 2020) implements the constant-product formula with a 0.3% trading fee. The fee is collected by adding it to the pool reserve before applying the invariant, effectively increasing k over time and accruing to liquidity providers.

Steps for a swap of token A for token B:

1. Trader sends Δx of A to the pool contract.
2. Contract applies 0.3% fee: effective input $= 0.997 \cdot \Delta x$.
3. Contract computes Δy from the invariant.
4. Contract sends Δy of B to the trader.

Slippage

Slippage

Slippage is the difference between the price at which a trade is initiated and the price at which it executes, caused by the curvature of the AMM price curve. For a constant-product pool:

$$\text{Execution price} = \frac{\Delta y}{\Delta x} = \frac{y}{x - \Delta x} > \frac{y}{x} = \text{Spot price}$$

Slippage increases with trade size relative to pool depth.

Example: AMM Trade Calculation

Pool: $x = 100$ ETH, $y = 200{,}000$ USDC, $k = 20{,}000{,}000$.
Spot price: $P = 200{,}000/100 = \$2{,}000$ per ETH.

Buy 1 ETH (ignore fee for simplicity):

$$(100 - 1)(y') = 20{,}000{,}000 \;\Rightarrow\; y' = 202{,}020.2$$

Cost $= 202{,}020.2 - 200{,}000 = \$2{,}020.2$
Slippage $= (2{,}020.2 - 2{,}000)/2{,}000 \approx 1.01\%$

Buy 10 ETH:

$$(90)(y') = 20{,}000{,}000 \;\Rightarrow\; y' = 222{,}222.2$$

Cost = \$22,222.2; average price = \$2,222.2 per ETH; slippage ≈ 11.1%.

Concentrated Liquidity (Uniswap v3)

Uniswap v3 (2021) introduced **concentrated liquidity**: LPs can specify a price range $[P_a, P_b]$ within which their capital is active. Outside this range, the LP earns no fees and holds only one token.

Capital Efficiency

By concentrating liquidity near the current price, an LP can achieve the same effective depth as a full-range v2 position with a fraction of the capital. If the current price stays within $[P_a, P_b]$, a concentrated position earns fees proportional to its share of *in-range* liquidity, not the entire pool.

Key parameters in Uniswap v3:

- **Tick**: The logarithmic price grid; each tick represents a 0.01% price increment.
- **Fee tiers**: 0.01% (stable pairs), 0.05% (major pairs), 0.3% (standard), 1% (exotic tokens).
- **Range orders**: A position entirely above or below the current price acts as a limit order—it fills automatically as price crosses the range.

Common Pitfall

Misconception: Concentrated liquidity always improves LP returns.
Correction: Concentration amplifies *both* fee income and impermanent loss. A narrow range that is frequently out-of-range earns no fees and requires active management. Passive LPs often do better with wider ranges.

Impermanent Loss

Formula Derivation

When a LP deposits tokens at price ratio $P_0 = y/x$ and the price changes to $P_1 = r \cdot P_0$ (price ratio $r = P_1/P_0$), the AMM rebalances the pool automatically.

Let the initial deposit be 1 unit of token A and P_0 units of token B (equal value). After price change to $r \cdot P_0$:

- New pool ratio: $x' = x/\sqrt{r}$, $y' = y \cdot \sqrt{r}$ (from $x'y' = k$ and $y'/x' = rP_0$).
- LP position value: $V_{\text{LP}} = x' \cdot rP_0 + y' \cdot 1 = 2y\sqrt{r} = 2P_0\sqrt{r}$.
- HODL value (just holding): $V_{\text{HODL}} = x \cdot rP_0 + y = P_0(1 + r)$.

Impermanent loss is the relative difference:

$$\boxed{IL = \frac{V_{\text{LP}}}{V_{\text{HODL}}} - 1 = \frac{2\sqrt{r}}{1+r} - 1} \tag{1}$$

Note: $IL \leq 0$ for all $r \neq 1$, with $IL = 0$ at $r = 1$ and $IL \to -1$ as $r \to \infty$ or $r \to 0$.

Worked Examples

Example: Impermanent Loss: price ratio r = 2 (price doubles)

$$IL = \frac{2\sqrt{2}}{1+2} - 1 = \frac{2.828}{3} - 1 \approx -0.0572$$

The LP position is worth approximately **5.72% less** than holding both tokens in the original ratio. Impermanent loss is symmetric: a halving ($r = 0.5$) also gives $IL \approx -5.72\%$.

Example: Impermanent Loss: price ratio r = 4 (4x price increase)

$$IL = \frac{2\sqrt{4}}{1+4} - 1 = \frac{4}{5} - 1 = -0.20$$

A fourfold price increase results in a **20% loss** relative to holding. Whether the LP net profits depends on whether trading fees exceed 20%.

Price ratio r	IL	LP value / HODL value
0.25	−20.00%	80.0%
0.50	−5.72%	94.3%
1.00	0.00%	100.0%
2.00	−5.72%	94.3%
4.00	−20.00%	80.0%
9.00	−40.00%	60.0%

Common Pitfall

Impermanent loss is not truly impermanent. It becomes permanent the moment an LP withdraws during price divergence. The name is misleading; a better term is *divergence loss* or *rebalancing loss*.

Lending and Borrowing

Aave and Compound Overview

DeFi lending protocols (Aave, Compound) allow users to:

- **Supply (lend)**: Deposit assets into a pool and earn a variable supply APY. Receive interest-bearing tokens (aTokens in Aave, cTokens in Compound) representing the claim.

- **Borrow**: Deposit collateral exceeding the loan value (overcollateralized) and receive borrowed assets. Pay a variable borrow APY.

No credit checks are required; the smart contract enforces solvency through mandatory overcollateralization.

Utilization Curve and Interest Rates

The borrow and supply rates are functions of the **utilization rate** U = Total Borrowed/Total Supplied.

Aave Interest Rate Model

For a pool with optimal utilization U_{opt} (typically 80%):

$$R_{\text{borrow}}(U) = R_0 + \begin{cases} \dfrac{U}{U_{\text{opt}}} \cdot R_{\text{slope1}} & \text{if } U \leq U_{\text{opt}} \\ R_{\text{slope1}} + \dfrac{U - U_{\text{opt}}}{1 - U_{\text{opt}}} \cdot R_{\text{slope2}} & \text{if } U > U_{\text{opt}} \end{cases}$$

where R_0 is the base rate, R_{slope1} is the gradual slope (e.g., 4% at optimal), and R_{slope2} is the steep slope above optimal (e.g., 75%). The supply rate is:

$$R_{\text{supply}} = R_{\text{borrow}} \times U \times (1 - \text{reserve factor})$$

The "kink" at U_{opt} serves as a safety mechanism: if utilization spikes above optimal, sharply higher borrow rates attract new deposits and discourage further borrowing, restoring liquidity.

Example: Interest Rate at High Utilization

Parameters: $R_0 = 0\%$, $U_{\text{opt}} = 80\%$, $R_{\text{slope1}} = 4\%$, $R_{\text{slope2}} = 75\%$.

At $U = 90\%$ (above optimal):

$$R_{\text{borrow}} = 4\% + \frac{0.90 - 0.80}{1 - 0.80} \times 75\% = 4\% + 0.5 \times 75\% = 41.5\%$$

High rates signal urgency for depositors to add liquidity.

Liquidation Mechanics

Health Factor

Every borrowing position has a **health factor**:

$$HF = \frac{\sum_i (\text{Collateral}_i \times \text{LiquidationThreshold}_i)}{\text{TotalBorrows}} \tag{2}$$

- $HF > 1$: Position is healthy.
- $HF < 1$: Position is undercollateralized; anyone can liquidate.

- Liquidation thresholds by asset: ETH $\approx 82.5\%$, WBTC $\approx 75\%$, stablecoins $\approx 85\%$.

Liquidation Process

When $HF < 1$, a liquidator may repay up to 50% of the outstanding debt and receive collateral at a **liquidation bonus** (5–10% discount):

1. Liquidator calls `liquidationCall(collateralAsset, debtAsset, user, debtAmount)`.
2. Protocol verifies $HF < 1$.
3. Liquidator transfers `debtAmount` of debt token to the protocol.
4. Protocol transfers to liquidator: collateral worth `debtAmount` $\times$ (1 + bonus) at oracle price.
5. Position's HF improves; if still < 1, further liquidations possible.

Cascade Scenarios

A **liquidation cascade** occurs when forced selling by liquidators depresses asset prices, causing more positions to fall below $HF = 1$, triggering further liquidations:

$$\text{Price drop} \to HF < 1 \to \text{Liquidation} \to \text{Forced sell} \to \text{Further price drop} \to \ldots$$

This feedback loop can amplify a minor price move into a major crash. During the May 2021 and November 2022 crypto downturns, hundreds of millions of dollars in positions were liquidated within hours.

Common Pitfall

Oracle dependency: Liquidations trigger at oracle-reported prices. If an oracle is manipulated (e.g., via a flash loan on a low-liquidity asset), positions can be wrongly liquidated. Always check the oracle source before using a lending protocol.

Oracles

Oracle

An **oracle** is a service that delivers off-chain data (asset prices, interest rates, random numbers, event outcomes) to smart contracts on the blockchain. Blockchains are deterministic, isolated systems with no native ability to query external APIs.

Chainlink

Chainlink is the dominant decentralized oracle network. Architecture:

- Multiple independent node operators fetch data from premium data providers.
- Operators stake LINK tokens as collateral.
- An on-chain aggregator contract computes the median of reported values; outliers are rejected.

- Nodes earn LINK rewards for timely, accurate responses; dishonest nodes are slashed.

TWAP Oracles

A **Time-Weighted Average Price (TWAP)** oracle computes the average price over a window $[t - T, t]$:

$$\text{TWAP} = \frac{\int_{t-T}^{t} P(\tau)\, d\tau}{T} \approx \frac{1}{N} \sum_{i=1}^{N} P_i$$

TWAP oracles resist spot manipulation because an attacker must sustain a price deviation over the entire time window, which is capital-intensive.

Oracle Manipulation Risks

- **Flash loan spot price attack**: Borrow large amounts, move the price on a low-liquidity DEX, trigger liquidations, repay in same block. Only works against protocols using spot prices as oracles.
- **Oracle extractable value (OEV)**: MEV specifically arising from oracle update timing. Searchers watch for oracle update transactions and front-run them.
- **Low-liquidity asset manipulation**: Tokens with thin markets can be price-moved cheaply, enabling collateral inflation attacks.

Flash Loans

Flash Loan

A **flash loan** is an uncollateralized loan that is borrowed and repaid within a single blockchain transaction. If the loan is not repaid (plus fee) before the transaction completes, the entire transaction reverts, as if the loan never occurred.

Atomic Execution Model

The atomicity of Ethereum transactions makes flash loans possible. A flash loan transaction typically:

1. Calls `flashLoan(asset, amount, receiverContract, ...)` on the lending pool.
2. Pool transfers `amount` to `receiverContract`.
3. `receiverContract.executeOperation()` runs arbitrary logic (arbitrage, liquidation, collateral swap, etc.).
4. Pool checks that `amount + fee` has been returned; if not, the entire transaction reverts.

Use Cases

- **Arbitrage**: Borrow ETH, buy on DEX A, sell on DEX B at a higher price, repay loan, keep profit—all in one transaction.
- **Liquidation**: Borrow the debt asset, liquidate an unhealthy position, receive discounted collateral, sell collateral to repay the flash loan.
- **Collateral swap**: Replace ETH collateral with WBTC collateral atomically (borrow new collateral, repay loan, withdraw old collateral, repay flash loan).
- **Self-liquidation**: A borrower facing liquidation can use a flash loan to repay their own debt and recover collateral before a liquidator claims the bonus.

Attack Vectors

Flash loans dramatically lower the capital requirement for attacks:

- **Oracle manipulation**: Borrow $100M, manipulate a low-liquidity oracle, drain a protocol exploiting the false price, repay loan.
- **Governance attacks**: Borrow governance tokens, vote on a malicious proposal, return tokens—all in one transaction. Defeated by snapshot-based voting and time locks.

Example: Flash Loan Arbitrage

ETH trades at $2,000 on Uniswap and $2,010 on SushiSwap.

1. Flash borrow 1,000 ETH from Aave (fee: 0.05% = 0.5 ETH).
2. Sell 1,000 ETH on Uniswap for 2,000,000 USDC.
3. Buy 1,000 ETH on SushiSwap for $1{,}000 \times 2{,}010 = 2{,}010{,}000$ USDC. Wait—arbitrage goes the other direction: buy cheap, sell dear.
4. Buy 1,000 ETH on Uniswap using 2,000,000 USDC, sell for 2,010,000 USDC on SushiSwap.
5. Repay 1,000 ETH + 0.5 ETH fee to Aave.
6. Profit: $\approx$ $10,000 minus gas costs.

No upfront capital required; the entire operation is atomic.

Yield Farming and Liquidity Mining

Mechanics

Yield Farming

Yield farming is the practice of actively moving assets across DeFi protocols to maximize returns. **Liquidity mining** is a specific incentive structure where protocols distribute governance tokens to LPs to bootstrap liquidity.

A typical yield stack:

1. Deposit ETH and USDC into a Uniswap pool → earn trading fees + UNI governance tokens.
2. Deposit the Uniswap LP token into a yield aggregator (Yearn) → Yearn auto-compounds rewards.
3. Use aTokens from Aave lending as collateral to borrow USDC → reinvest for leverage.

APY Calculation

For simple yield without compounding:

$$\text{APY}_{\text{simple}} = \frac{\text{Annual Reward Tokens} \times P_{\text{token}}}{\text{Total Value Deposited}}$$

For continuously compounded yield:

$$\text{APY} = \left(1 + \frac{r}{n}\right)^n - 1$$

where r is the annual rate and n is the number of compounding periods per year. As $n \to \infty$: $\text{APY} = e^r - 1$.

Example: APY Calculation

A pool has \$10M TVL. The protocol distributes 1,000 GOV tokens per day, each worth \$100. Daily reward = \$100,000.

$$\text{APY} = \frac{100{,}000 \times 365}{10{,}000{,}000} = 36.5\%$$

If 20% more liquidity is attracted (TVL → \$12M) and GOV price drops by 30% (GOV → \$70):

$$\text{APY} = \frac{70{,}000 \times 365}{12{,}000{,}000} \approx 21.3\%$$

APY is extremely sensitive to token price and TVL changes.

Sustainability

Liquidity mining APYs exceeding 100% are almost never sustainable:

- High APY attracts liquidity → APY falls (TVL denominator grows).
- Reward tokens are sold by farmers → token price drops → APY falls further.

- When APY falls below the risk-adjusted opportunity cost, LPs exit → TVL collapses → token price collapses.

Sustainable DeFi yields (5–20%) come from genuine economic activity: trading fees, lending interest, and options premiums.

Gas Economics

Gas and Gwei

Gas

Gas is the unit measuring computational effort required to execute an Ethereum operation. Simple ETH transfers cost 21,000 gas; complex DeFi interactions can cost 200,000–500,000+ gas. **Gwei** ($= 10^{-9}$ ETH) is the standard unit for gas prices. Transaction fee = Gas Used × Gas Price (in Gwei).

EIP-1559 (London Upgrade, 2021)

Before EIP-1559, gas markets ran as a first-price auction: users bid gas prices and miners selected the highest bids. This was unpredictable and wasteful.

EIP-1559 introduced a two-component fee:

Component	Description
Base Fee	Algorithmically set by the protocol. Burned (destroyed). Increases by up to 12.5% if the previous block was more than 50% full; decreases if less than 50% full.
Priority Fee (tip)	Optional tip paid directly to the validator (block proposer). Incentivizes inclusion during low-congestion periods.

Max fee: Users set `maxFeePerGas` = base fee + priority fee cap. If the actual base fee is lower, the difference is refunded.

Deflationary pressure: When the network is busy, the base fee and therefore ETH burn rate can exceed new issuance, making ETH net deflationary. Track at `https://ultrasound.money`.

Gas Estimation and MEV

DeFi transactions face **priority gas auctions (PGAs)**: bots competing to front-run profitable transactions by bidding higher priority fees. This caused "gas wars" pre-EIP-1559. Private transaction relays (Flashbots Protect) allow users to submit transactions directly to block builders, bypassing the public mempool and avoiding front-running.

Practice Problems

1. **(AMM Math)** A constant-product pool has 500 ETH and 1,000,000 USDC.

   a. What is the current spot price of ETH in USDC?

   b. A trader buys 10 ETH from the pool (ignore fees). How much USDC does she pay, and what is the effective price per ETH?

   c. Calculate the slippage percentage.

   Solution:

   a. $P = 1{,}000{,}000/500 = \$2{,}000$ per ETH.

   b. $k = 500 \times 1{,}000{,}000 = 500{,}000{,}000$. New ETH reserve: $500 - 10 = 490$. New USDC reserve: $500{,}000{,}000/490 = 1{,}020{,}408.16$. USDC paid: $1{,}020{,}408.16 - 1{,}000{,}000 = \$20{,}408.16$. Effective price: $\$20{,}408.16/10 = \$2{,}040.82$ per ETH.

   c. Slippage: $(2{,}040.82 - 2{,}000)/2{,}000 \approx 2.04\%$.

2. **(Impermanent Loss)** An LP deposits \$5,000 worth of ETH and \$5,000 USDC into a constant-product pool when ETH = \$1,000.

   a. How many ETH and USDC does the LP deposit?

   b. ETH rises to \$4,000. Calculate the LP's position value and the value of simply holding the initial tokens.

   c. Calculate the impermanent loss using the formula (1).

   Solution:

   a. 5 ETH and 5,000 USDC.

   b. $r = 4{,}000/1{,}000 = 4$. New pool ratio: $x' = 5/\sqrt{4} = 2.5$ ETH, $y' = 5{,}000 \times \sqrt{4} = 10{,}000$ USDC. LP value: $2.5 \times 4{,}000 + 10{,}000 = \$20{,}000$. HODL value: $5 \times 4{,}000 + 5{,}000 = \$25{,}000$.

   c. $IL = 2\sqrt{4}/(1+4) - 1 = 4/5 - 1 = -20\%$. The LP is \$5,000 (20%) worse off than simply holding.

3. **(Health Factor)** A borrower deposits 10 ETH (price \$2,500, liquidation threshold 80%) and borrows 15,000 USDC.

   a. Calculate the initial health factor.

   b. ETH price drops to \$2,000. What is the new health factor?

   c. At what ETH price does the position become liquidatable ($HF < 1$)?

   Solution:

   a. $HF = (10 \times 2{,}500 \times 0.80)/15{,}000 = 20{,}000/15{,}000 \approx 1.33$.

b. $HF = (10 \times 2{,}000 \times 0.80)/15{,}000 = 16{,}000/15{,}000 \approx 1.07$.

c. Liquidation when $10 \times P \times 0.80 = 15{,}000 \Rightarrow P = 15{,}000/8 = \$1{,}875$.

4. **(Gas Economics)** A DeFi swap costs 200,000 gas. The base fee is 30 Gwei and the priority fee is 2 Gwei. ETH = \$2,000.

   a. Calculate the total transaction cost in ETH and USD.

   b. How much ETH is burned? How much goes to the validator?

   Solution:

   a. Total gas price = 32 Gwei. Fee = $200{,}000 \times 32 \times 10^{-9}$ ETH = 0.0064 ETH = \$12.80.

   b. Burned (base fee): $200{,}000 \times 30 \times 10^{-9} = 0.006$ ETH = \$12.00. To validator (priority): $200{,}000 \times 2 \times 10^{-9} = 0.0004$ ETH = \$0.80.

Key Takeaways

1. **DeFi recreates traditional finance** permissionlessly using smart contracts. Composability (money legos) enables rapid innovation but also cascading failure risk.

2. **AMMs replace order books** with a constant-product invariant $x \cdot y = k$. Prices are determined algorithmically; larger trades face higher slippage. Uniswap v3 concentrates liquidity for capital efficiency at the cost of active management.

3. **Impermanent loss** is the cost paid by LPs for providing liquidity when prices diverge. Formula: $IL = 2\sqrt{r}/(1+r) - 1$. It is symmetric, permanent upon withdrawal, and can exceed fee income for volatile pairs.

4. **Lending protocols** use utilization-based interest rate models with a kink at the optimal utilization to balance supply and demand. Health factors and liquidation bonuses protect solvency.

5. **Oracles** are critical infrastructure but also the most common DeFi attack vector. TWAP oracles resist flash loan manipulation; spot price oracles do not.

6. **Flash loans** enable uncollateralized, atomic borrowing of any amount. Legitimate uses include arbitrage and collateral swaps; malicious uses include oracle manipulation and governance attacks.

7. **Yield farming APYs** are reflexive: high APY attracts capital, diluting the yield. Only yields backed by genuine economic activity (trading fees, lending spreads) are sustainable.

8. **EIP-1559** replaced first-price gas auctions with a burned base fee plus optional priority tip, improving fee predictability and making ETH potentially deflationary during high network usage.

Further Reading

- Adams, H., Zinsmeister, N., & Robinson, D. (2020). *Uniswap v2 Core*. https://uniswap.org/whitepaper.pdf
  The original whitepaper describing the constant-product AMM and fee mechanics.
- Adams, H. et al. (2021). *Uniswap v3 Core*. https://uniswap.org/whitepaper-v3.pdf
  Introduces concentrated liquidity, tick-based positions, and multiple fee tiers.
- Aave Protocol (2020). *Aave Protocol Whitepaper V2*. https://github.com/aave/aave-protocol/blob/master/docs/Aave_Protocol_Whitepaper_v1_0.pdf
  Full specification of health factors, interest rate models, and flash loans.
- Buterin, V. et al. (2019). *EIP-1559: Fee Market Change*. https://eips.ethereum.org/EIPS/eip-1559
  Original proposal for the base fee / priority fee mechanism.
- Leshner, R. & Hayes, G. (2019). *Compound: The Money Market Protocol*. https://compound.finance/documents/Compound.Whitepaper.pdf
  Describes the utilization-based interest rate model and cToken mechanics.
- DeFi Llama: https://defillama.com/ — Live TVL, yields, and analytics across all major DeFi protocols.
- Rekt News: https://rekt.news/leaderboard/ — Database of DeFi hacks; essential reading for understanding attack vectors.