

L42: Flash Loans and Composability

Module F: Advanced Topics

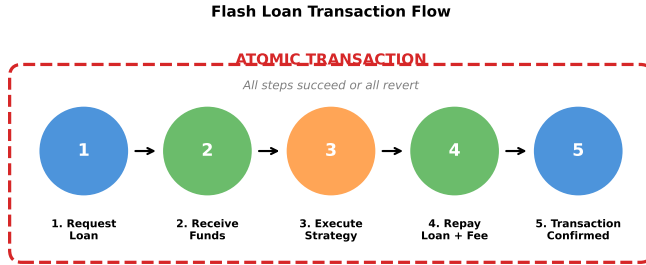
Blockchain & Cryptocurrency Course

December 2025

- Understand flash loan mechanics and atomicity
- Analyze legitimate flash loan use cases
- Examine flash loan attack vectors and notable exploits
- Explore DeFi composability (“money legos”)
- Understand MEV and its relationship to flash loans

What is a Flash Loan?

- **Definition:** Uncollateralized loan borrowed and repaid within single transaction
- **Key Property:** *Atomicity* – loan and repayment are all-or-nothing
- **If repayment fails:** Entire transaction reverts, lender loses nothing
- **No Collateral Required:** Enabled by smart contract execution model
- **Loan Size:** Unlimited (constrained only by liquidity pool)
- **Duration:** Fraction of a second (one block)
- **Unique to DeFi:** Impossible in traditional finance

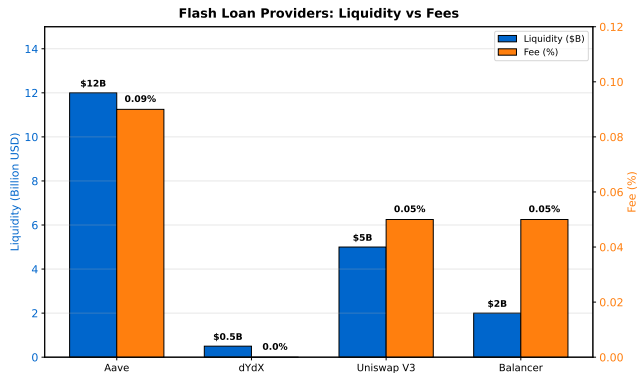


If any step fails, the entire transaction reverts – zero risk to lender

Traditional Loans vs Flash Loans

Property	Traditional Loan	Flash Loan
Collateral	Required (often $\geq 100\%$)	None
Duration	Days/months/years	Single transaction
Creditworthiness	Required (KYC)	Not required
Repayment Guarantee	Legal contracts	Smart contract atomicity
Risk to Lender	Default risk	Zero (reverts)

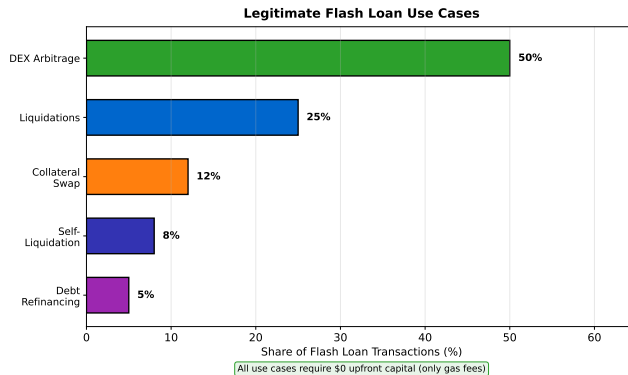
Paradigm Shift: Code execution guarantees replace legal enforcement



Aave is largest provider; dYdX offers free flash loans

- **Composability:** DeFi protocols are like “money legos”
- **Permissionless Integration:** Any contract can call any other contract
- **Atomic Transactions:** Multiple protocol interactions in one transaction
- **Examples of Composability:**
 - ① Swap on Uniswap → Deposit into Aave → Borrow
 - ② Flash loan → Liquidate → Swap collateral → Repay
 - ③ Borrow from Compound → Yield farm on Curve
- **Risk:** Cascading failures, expanded attack surface

Legitimate Use Cases



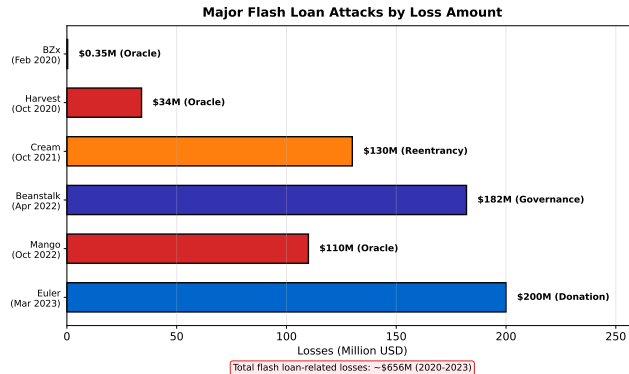
All use cases democratize capital-intensive strategies to anyone

- **Scenario:** ETH trades at \$2000 on Uniswap, \$2020 on SushiSwap
- **Flash Loan Arbitrage:**
 - 1 Borrow 1000 ETH via flash loan
 - 2 Buy 1000 ETH on Uniswap (\$2,000,000)
 - 3 Sell 1000 ETH on SushiSwap (\$2,020,000)
 - 4 Repay flash loan + fee (\$2,001,000)
 - 5 Profit: \$19,000 in one transaction
- **Capital Required:** Only gas fees (\$50-\$200)
- **Democratization:** Anyone can be arbitrageur, not just whales

- **Problem:** User has debt collateralized with Asset A, wants Asset B
- **Flash Loan Solution:**
 - 1 Borrow Asset A via flash loan
 - 2 Repay existing debt
 - 3 Withdraw original collateral
 - 4 Deposit new collateral (Asset B)
 - 5 Borrow Asset A again
 - 6 Repay flash loan
- **Result:** Collateral swapped atomically, zero liquidation risk
- **Fee:** Only flash loan fee (0.05-0.09%)

- **Dark Side:** Flash loans enable large-scale attacks with zero capital
- **Attack Pattern:**
 - ① Borrow massive amount via flash loan
 - ② Manipulate protocol state (price oracle, governance)
 - ③ Exploit manipulation for profit
 - ④ Repay flash loan
- **Impact:** \$500M+ stolen via flash loan attacks (2020-2023)
- **Key Insight:** Flash loans amplify existing vulnerabilities

Major Flash Loan Attacks

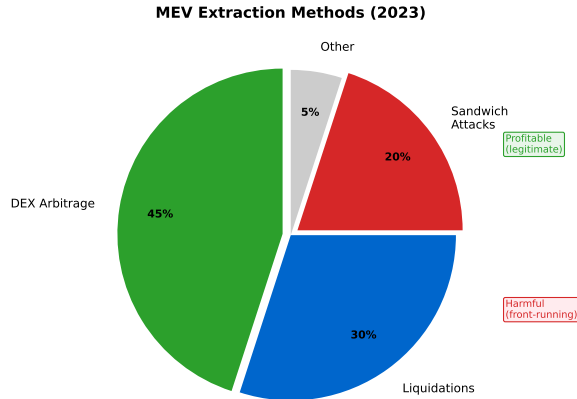


Oracle manipulation is most common attack vector

Attack Example: Oracle Manipulation

- **Vulnerable Protocol:** Uses single DEX price as oracle
- **Attack Steps:**
 - 1 Borrow 10,000 ETH via flash loan
 - 2 Buy all TOKEN on Uniswap (10x price increase)
 - 3 Protocol oracle reads inflated price
 - 4 Borrow stablecoins using overvalued TOKEN
 - 5 Sell TOKEN back (price normalizes)
 - 6 Repay flash loan + keep stablecoins
- **Real Example:** Harvest Finance (\$34M stolen)
- **Mitigation:** Use TWAP or Chainlink oracles

- ❶ **Decentralized Oracles:** Use Chainlink (not single DEX)
- ❷ **Time-Weighted Average Price (TWAP):** Average over blocks
- ❸ **Reentrancy Guards:** Prevent recursive contract calls
- ❹ **Governance Delays:** Timelock on parameter changes
- ❺ **Vote Locking:** Require tokens locked before voting
- ❻ **Circuit Breakers:** Pause if anomalous activity detected



Flash loans amplify both legitimate arbitrage and harmful MEV extraction

- **MEV (Maximal Extractable Value):** Profit from transaction ordering
- **Flash Loans + MEV:** Amplify arbitrage and liquidation profits
- **Techniques:**
 - **Front-running:** Place transaction before victim's
 - **Back-running:** Place transaction after victim's
 - **Sandwich Attacks:** Front-run + back-run
- **Flashbots:** Democratize MEV extraction, reduce gas wars
- **MEV Volume:** \$600M+ extracted in 2023

Positive Effects

- Democratize arbitrage
- Increase market efficiency
- Enable capital-efficient refinancing
- Liquidation bots improve protocol health

Negative Effects

- Enable zero-capital attacks
- Amplify protocol vulnerabilities
- MEV extraction harms users
- Governance manipulation risk

Net Assessment: Powerful tool that magnifies both good and bad protocol design

- **Legal Gray Area:** No clear regulatory framework
- **Key Questions:**
 - Are flash loan attacks theft or code exploitation?
 - Is the protocol or attacker liable?
 - Code is law vs legal enforcement
- **Precedent:** Mango Markets attacker arrested (Oct 2022)
 - Charged with market manipulation, not flash loan use
- **Protocol Responsibility:** Bug bounties, audits, insurance

Key Takeaways:

- Flash loans: Uncollateralized loans repaid in single atomic transaction
- Enabled by smart contract atomicity: Revert on failure = zero lender risk
- Use cases: Arbitrage, collateral swaps, self-liquidation, refinancing
- Composability: DeFi protocols as interoperable building blocks
- Attacks: Oracle manipulation, governance takeover (\$500M+ stolen)
- Flash loans amplify existing protocol vulnerabilities (not root cause)
- Defenses: Decentralized oracles, TWAP, vote locking, circuit breakers

- ❶ Why can flash loans exist in DeFi but not in traditional finance?
- ❷ How do flash loans democratize arbitrage opportunities?
- ❸ What makes oracle manipulation the most common attack vector?
- ❹ Should flash loan attackers be prosecuted if they exploit code bugs?
- ❺ How can protocols defend against flash loan governance attacks?