Layer 2 Scaling Solutions: Scaling Ethereum and Beyond

Standalone Technical Lecture

Prof. Dr. Joerg Osterrieder

University Lecture Series

March 5, 2026

Lecture Roadmap

1. Scaling Problem & L2 Fundamentals
2. Optimistic Rollups
3. Zero-Knowledge Rollups
4. Sidechains & Alt Scaling

Learning Objectives

- Understand the blockchain trilemma and scalability constraints
- Analyse optimistic and zero-knowledge rollup mechanisms
- Evaluate state channels, sidechains, and Plasma trade-offs
- Assess data availability and proof system designs
- Survey the live L2 ecosystem and future directions

Prerequisites

- Lessons 1–5: Blockchain fundamentals, smart contracts, DeFi
- Basic familiarity with Ethereum and EVM execution

Durat

90 minutes — 5 sections — ˜55 frames — Prerequisite: Lessons 1–5

Table of Contents

Navig

through 5 sections covering the scaling problem to the L2 ecosystem and future

Learning Objectives

By the end of this lecture, you will be able to:

1. **Explain** the blockchain trilemma and why Layer 2 scaling is necessary
2. **Compare** optimistic rollups and ZK rollups (fraud proofs vs. validity proofs)
3. **Analyze** the economics of rollup sequencers, provers, and data availability
4. **Evaluate** bridge security risks and cross-L2 communication challenges
5. **Assess** the modular blockchain thesis (execution, settlement, data availability, consensus)

Bloom

taxonomy levels: Remember → Understand → Apply → Analyze → Evaluate → Create

Section 1: The Scaling Problem & Layer 2 Fundamentals

Why blockchains can't scale natively and how Layer 2 solves it

What You Will Learn

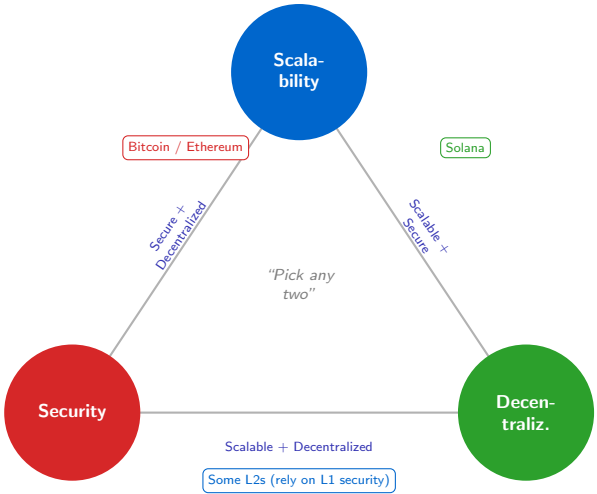
- The blockchain trilemma and its implications for scalability
- Why Ethereum L1 is limited to ~15 TPS
- Layer 2 taxonomy: rollups, state channels, sidechains, validiums
- State channels, Plasma, and early L2 approaches

Frames in This Section

Secti

1 of 5 — Frames 4–14

The Blockchain Trilemma



The Trilemma Explained

- **Scalability**: High throughput, low latency, low fees
- **Security**: Resistant to attacks; honest-minority assumption
- **Decentralization**: No single point of control or failure

Implications

- Optimizing two properties degrades the third
- Ethereum chose security + decentralization
- Result: ˜15 TPS, high gas fees at peak demand
- Layer 2 aims to *break* the trilemma by moving execution off-chain while settling on L1

Vitali Buterin formalized the trilemma – Layer 2 solutions aim to break it by separating execution from consensus

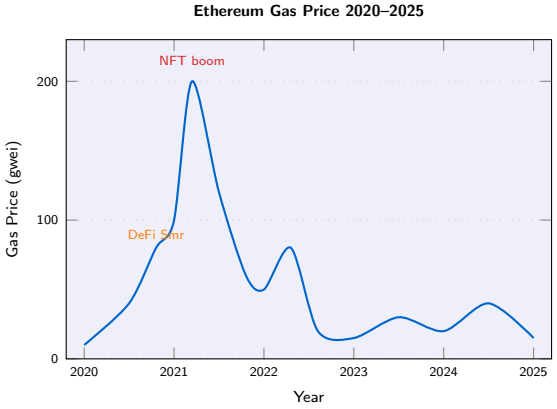
Ethereum's Scaling Bottleneck

Hard Limits at L1

- **~15 TPS** – theoretical maximum under normal load
- **Block gas limit**: 30M gas (target 15M after EIP-1559)
- **Simple transfer**: 21,000 gas
- **DEX swap**: ~150,000 gas
- **Complex DeFi tx**: 200,000–500,000 gas

Economic Consequences

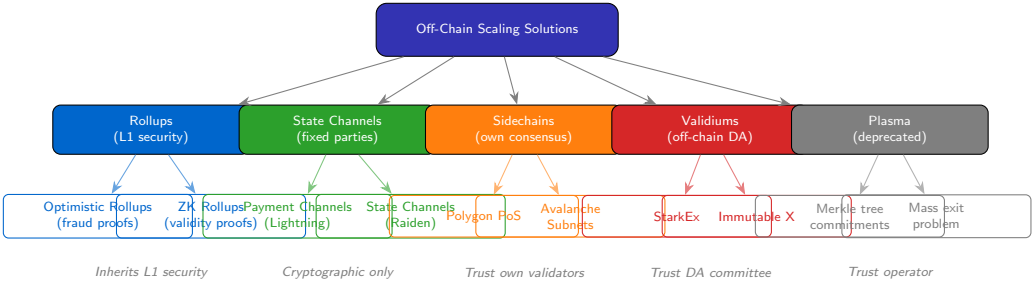
- Peak demand: base fee spikes to 100+ gwei
- Gas cost: **$50–$500** per transaction at peaks
- NFT mint frenzies: gas wars drove fees to **$1,000+**
- DeFi Summer 2020 congestion: weeks of 100+ gwei
- Blocks fill in milliseconds; users outbid each other



The Comparison

Ethereum: ~1.3M tx/day
Visa: ~150M tx/day (115× more)

L2 Taxonomy Overview



Rollu

are the dominant paradigm because they inherit L1 security while supporting general-purpose computation

State Channel Contract

Listing 1: Simple Payment Channel

```
// Simple Payment Channel
contract PaymentChannel {
    address public sender;
    address public recipient;
    uint256 public expiration;

    constructor(address _to,
                uint256 duration)
        payable {
        sender = msg.sender;
        recipient = _to;
        expiration = block.timestamp
            + duration;
    }

    function close(uint256 amount,
                   bytes memory sig)
        external {
        require(msg.sender == recipient);
        require(verify(amount, sig));
        payable(recipient).transfer(
            amount);
        selfdestruct(
            payable(sender));
    }
}
```

How State Channels Work

- **Open**: Sender deposits ETH into the contract (1 on-chain tx)
- **Transact**: Parties exchange signed messages off-chain – no gas, instant finality
- **Close**: Recipient submits final signed state; contract verifies and distributes (1 on-chain tx)
- **Dispute**: If sender stops responding, recipient can close after expiration

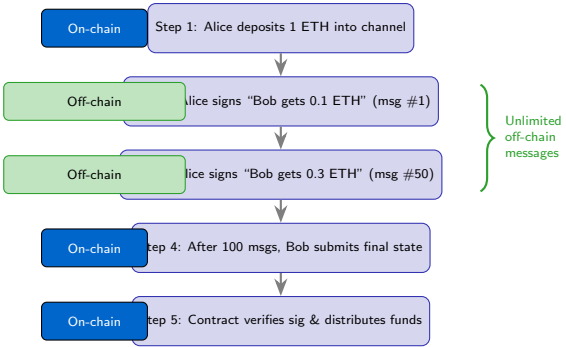
Key Properties

- Only 2 on-chain transactions regardless of how many off-chain messages
- Privacy: intermediate states never touch the chain
- Limitation: participants must be known in advance; capital locked for channel lifetime
- Best for: micropayments, gaming, repeated bilateral interactions

State

channels are ideal for repeated payments between two parties – e.g., micropayments, gaming moves

State Channels Deep Dive



Economics

- **2 on-chain txs**: open + close
- **Unlimited** off-chain transactions
- No gas for intermediate states
- Capital locked for channel duration

Lightning Network (Bitcoin)

- 5,000+ BTC locked capacity
- ˜16,000 routing nodes
- Millisecond payment finality
- Multi-hop routing via HTLCs

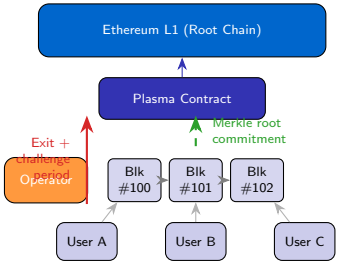
Raiden Network (Ethereum)

- ERC-20 payment channels
- Same HTLC routing model

Plasma Chains

Plasma Architecture

- **Child chains** run independently with their own operators and blocks
- Child chain operators periodically **commit Merkle roots** to L1
- Users can **exit** with a Merkle proof showing their funds on the child chain
- L1 enforces exit validity through a **challenge period** (typically 7 days)



Why Plasma Failed

- **Mass exit problem**: if operator turns malicious, all users must exit simultaneously – L1 cannot process them all
- **Data availability**: operator can withhold block data, preventing users from constructing valid exit proofs
- **Limited generality**: complex smart contract logic is difficult to prove in exit games

Historical Note

Plasma proposed 2017 (Buterin & Poon); superseded by rollups which solve the data availability problem elegantly

Plasm

was Ethereum's first L2 scaling proposal (2017) but was superseded by rollups which solve the data availability problem

Rollups vs Other L2 Approaches

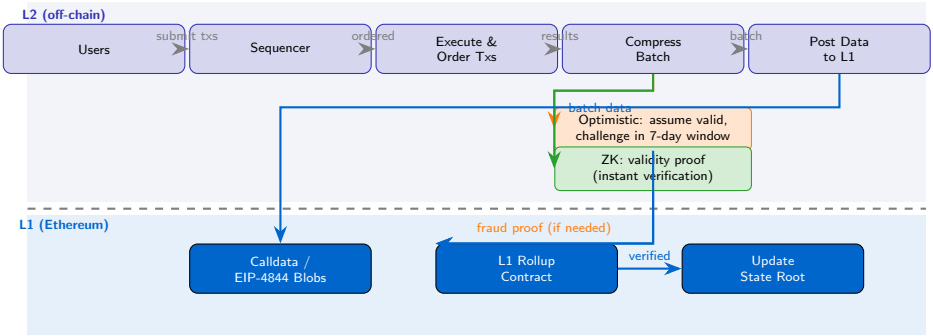
Approach	Security Model	Data Avail.	Generality	Finality	Status
Rollups	L1 (inherited)	On-chain	High (EVM)	7d / instant	Dominant
State Channels	Cryptographic	Off-chain	Low (fixed)	Instant	Niche
Sidechains	Own validators	Off-chain	High (EVM)	Fast	Active
Validiums	ZK proofs	DA committee	Medium	Instant	Specialized
Plasma	Merkle proofs	Operator	Low	7d challenge	Deprecated

Strong Moderate Weak

Rollu

are the dominant L2 paradigm because they inherit L1 security while supporting general-purpose computation

How Rollups Work (General)

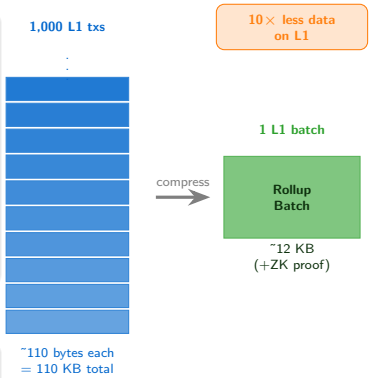


Both rollup types post transaction data to L1 – the key difference is how correctness is proven: fraud proofs vs validity proofs

Data Compression in Rollups

How Rollups Compress Data

- **Full L1 transaction**: ˜110 bytes
  - Signature (65 bytes), nonce (3), gas (3), to (20), value (8), data (varies)
- **Rollup compressed**: ˜12 bytes
  - Address delta (4 bytes), nonce (2), amount (8)
- **Compression ratio**: ˜10×
- **Signature aggregation**: BLS signatures or ZK validity proofs eliminate per-tx signatures entirely
- **EIP-4844 blobs**: Separate cheaper data market vs calldata (˜10× cheaper again)



Cost Savings Calculation

Scenario	1,000 Txs	Cost (est.)
L1 native	1,000 txs	$50,000
Rollup (calldata)	1 batch	$500
Rollup (blob)	1 batch	$50

Savings: 100×–1,000× depending on blob availability

Data compression is the key economic advantage of rollups – less data on L1 means lower costs per transaction

Section 1 Summary

1. The blockchain trilemma forces tradeoffs between scalability, security, and decentralization – optimizing two degrades the third.

2. Ethereum L1 is limited to ˜15 TPS; peak demand drives gas prices to \$50–\$500 per transaction, making many use-cases economically unviable.

3. Layer 2 solutions process transactions off-chain and settle on L1 for security – inheriting Ethereum's trust guarantees at a fraction of the cost.

4. Rollups are the dominant L2 paradigm, offering L1 security with 10–100$\times$ cost reduction via data compression and proof systems.

5. State channels work well for fixed parties with repeated interactions; Plasma was abandoned due to data availability and mass exit problems.

Secti

1 complete – next: Optimistic Rollups (Arbitrum, Optimism) and fraud proof mechanisms

Section 2: Optimistic Rollups

Fraud proofs, challenge periods, and the optimistic approach to scaling

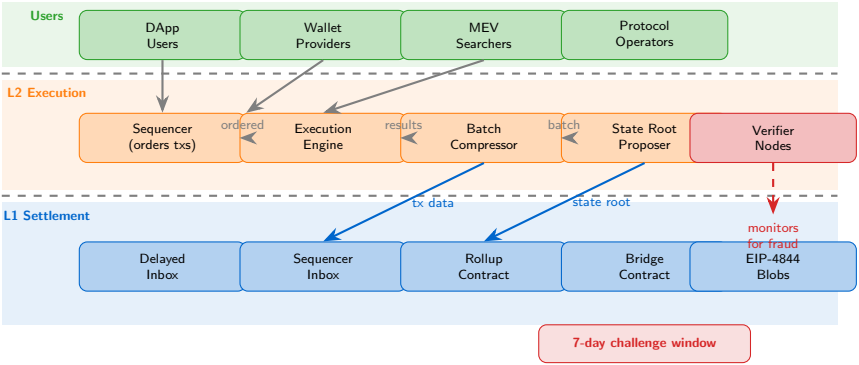
What You Will Learn

- ✓ How optimistic rollups assume validity and verify via fraud proofs
- ✓ Sequencer role, centralization risks, and decentralization roadmaps
- ✓ Challenge period mechanics and dispute resolution
- ✓ Arbitrum, Optimism, and Base architecture comparisons

Frames in This Section

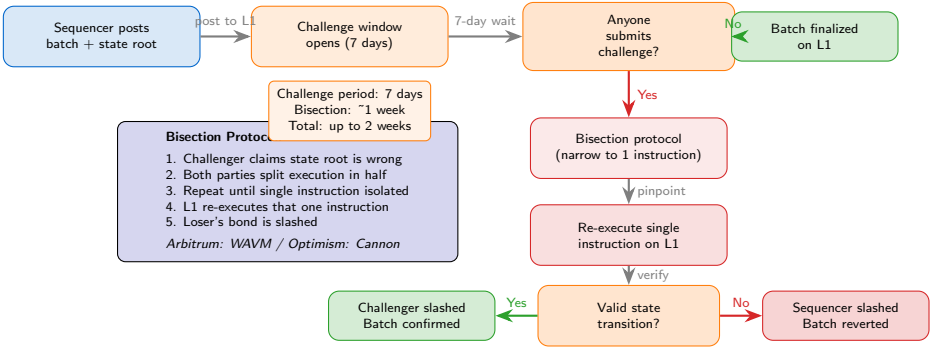
- Frame 16: Optimistic Rollup Architecture
- Frame 17: Fraud Proof Mechanism
- Frame 18: Fraud Proof Challenge (Code)
- Frames 19–24: Sequencer, Arbitrum, Optimism, Base, Fees, Comparison
- Frame 25: Section Summary

Optimistic Rollup Architecture



rollups assume all batches are valid – anyone can challenge with a fraud proof within the 7-day window

Fraud Proof Mechanism



Fraud proofs need only 1 honest verifier to catch cheating – the bisection protocol narrows the dispute to a single instruction

Fraud Proof Challenge

Listing 2: Simplified Fraud Proof

```
// Simplified Fraud Proof
contract OptimisticRollup {
    uint256 public challengePeriod
        = 7 days;
    mapping(bytes32 => uint256)
        public stateRoots;

    function submitBatch(
        bytes32 stateRoot,
        bytes calldata txData
    ) external onlySequencer {
        stateRoots[stateRoot] =
            block.timestamp;
        emit BatchSubmitted(
            stateRoot, txData);
    }

    function challengeBatch(
        bytes32 stateRoot,
        bytes calldata proof
    ) external {
        require(block.timestamp <
            stateRoots[stateRoot]
            + challengePeriod,
            "Challenge period over");
        require(verifyFraud(proof),
            "Invalid proof");
        delete stateRoots[stateRoot];
        // Slash sequencer bond
    }
}
```

Contract Analysis

- **submitBatch**: Sequencer posts a new state root along with compressed transaction data
- **challengePeriod**: 7-day window during which anyone can dispute
- **challengeBatch**: Verifier submits a fraud proof; if valid, the batch is deleted and sequencer bond is slashed
- **Key insight**: Only the *challenge path* is expensive – the happy path costs minimal gas

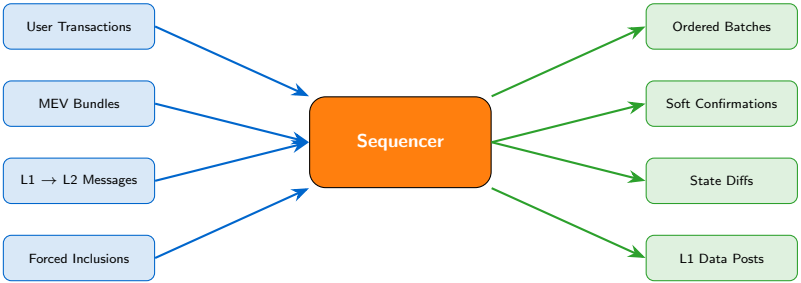
Security Assumptions

- Requires at least **1 honest verifier** monitoring the chain
- Verifier must have enough time to submit proof (7 days)
- L1 must remain censorship-resistant for the proof to land
- Sequencer bond must exceed potential gains from fraud

Trust Model

"Optimistic" = assume honest unless proven otherwise. Security is 1-of-N: only one honest watcher needed

The Sequencer Role



Centralization Risks:

- Single point of failure (liveness)
- MEV extraction / transaction censorship
- Regulatory capture of sequencer operator
- No permissionless block production

Decentralization Roadmap:

- Shared sequencing (Espresso, Astria)
- Based sequencing (L1 proposers order L2 txs)
- Sequencer auctions / rotation
- Forced inclusion via L1 delayed inbox

Today

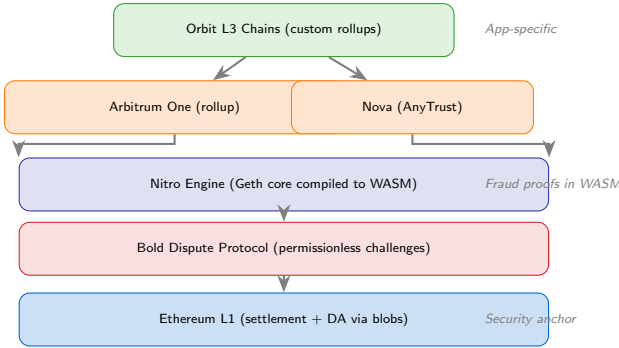
most optimistic rollups run a single centralized sequencer – decentralization is the next major milestone

Arbitrum Ecosystem

- **Arbitrum One**: Flagship optimistic rollup; largest L2 by TVL
- **Arbitrum Nova**: AnyTrust chain for gaming/social (off-chain DA via committee)
- **Arbitrum Orbit**: Framework for launching custom L3 chains
- **Nitro**: Current execution engine – compiles Geth to WASM for fraud proofs
- **Bold**: New permissionless dispute protocol (replaces single-challenger model)

Key Metrics

Metric	Value
Peak TPS	~4,000
Avg Fee	$0.01–$0.10
TVL	~$18B (2024 peak)
Challenge Period	7 days
Token	ARB (governance)



Arbit

One is the largest L2 by TVL – Nitro compiles Geth to WASM so fraud proofs re-execute exact EVM logic

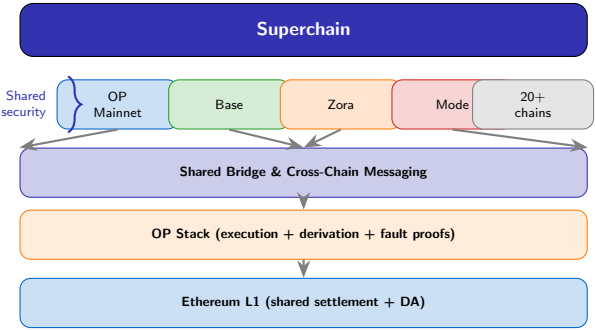
Optimism & the OP Stack

Optimism Ecosystem

- **OP Mainnet**: Original optimistic rollup; second-largest by ecosystem
- **OP Stack**: Open-source modular rollup framework – the "Linux of rollups"
- **Superchain**: Vision for interconnected OP Stack chains sharing security and messaging
- **Fault Proofs**: Cannon/op-program – permissionless fault proof system (live 2024)
- **Governance**: OP token + bicameral system (Token House + Citizens' House)

OP Stack Adopters

- **Base** (Coinbase) – largest OP Stack chain
- **Zora** – NFT-focused L2
- **Mode** – DeFi-focused L2
- **World Chain** (Worldcoin) – identity L2
- **20+ chains** deployed or announced



All Superchain members share the same bridge, sequencing, and upgrade governance

The OP Stack enables a "Superchain" of interoperable rollups – Base, Zora, Mode, and 20+ chains share security

Base & Coinbase L2

Base Overview

- **Built on**: OP Stack (Optimism's modular framework)
- **Operator**: Coinbase – leverages 110M+ verified users
- **No native token**: Revenue from sequencer fees; no token governance
- **Mission**: Bring the next 1 billion users on-chain
- **Launched**: August 2023

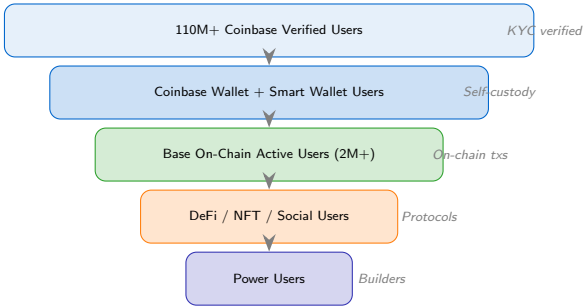
Key Metrics

Metric	Value
Daily Active Addresses	2M+ (2024 peak)
Avg Fee	$0.001–$0.01
TVL	~$8B (2024 peak)
Sequencer	Coinbase (centralized)
Superchain member	Yes

No-Token Model

Base has no native token. Coinbase earns revenue from sequencer fees and contributes to OP Collective governance via OP tokens.

Coinbase Onboarding Funnel



Optimistic Rollup Fee Structure

Fee Decomposition

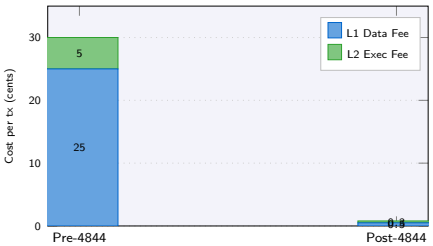
Total Fee = L1 Data Fee + L2 Execution Fee

- **L1 Data Fee** (dominant cost): Posting compressed transaction data to Ethereum
  - Pre-4844: via calldata (˜16 gas/byte)
  - Post-4844: via blobs (˜1 gas/byte equivalent)
- **L2 Execution Fee**: Gas for L2 computation (very cheap)
- **EIP-4844 impact**: 10–100× reduction in L1 data costs

Fee Examples (2024 averages)

Operation	Pre-4844	Post-4844
ETH transfer	$0.10	$0.001
Token swap	$0.30	$0.005
NFT mint	$0.50	$0.01
Complex DeFi	$1.00	$0.02

Fee Breakdown: Pre vs Post EIP-4844



EIP-4844 (March 2024) reduced L2 fees by 10–100× by introducing blob data – a separate cheaper data market

Optimistic Rollup Comparison

Feature	Arbitrum One	Optimism (OP Mainnet)	Base
Technology	Nitro (Geth + WASM)	OP Stack (Geth-based)	OP Stack (fork)
Fraud Proof Type	Bold (interactive bisection)	Cannon (fault proof program)	Inherits OP fault proofs
Challenge Period	7 days	7 days	7 days
TVL (2024 peak)	~$18B	~$7B	~$8B
Peak TPS	~4,000	~2,000	~2,000
Avg Fee (post-4844)	$0.01–$0.10	$0.001–$0.05	$0.001–$0.01
Native Token	ARB (governance)	OP (governance)	None
Sequencer	Offchain Labs (centralized)	OP Foundation (centralized)	Coinbase (centralized)
Key Feature	Largest TVL, Orbit L3s	OP Stack / Superchain	Coinbase onboarding
DA Layer	Ethereum blobs + calldata	Ethereum blobs + calldata	Ethereum blobs + calldata
EVM Compatibility	Full (Geth fork)	Full (Geth fork)	Full (Geth fork)

Arbitrum Strengths

- Largest TVL and DeFi ecosystem
- Orbit enables custom L3 chains
- Bold: permissionless disputes

Optimism Strengths

- OP Stack: open-source standard
- Superchain vision (20+ chains)
- Retroactive Public Goods Funding

Base Strengths

- Coinbase's 110M+ user funnel
- Lowest fees among top L2s
- No-token model (pure utility)

All three use Ethereum for settlement and DA – they differ in execution engines, governance, and go-to-market strategy

Section 2 Summary

1. Optimistic rollups assume all batches are valid and rely on fraud proofs during a 7-day challenge window – only 1 honest verifier is needed for security.

2. The bisection protocol narrows disputes to a single instruction re-executed on L1 – making fraud proofs efficient but introducing a 7-day withdrawal delay.

3. Centralized sequencers are the primary trust assumption today; shared sequencing, based sequencing, and forced inclusion are active areas of decentralization research.

4. Arbitrum (Nitro/Bold), Optimism (OP Stack/Superchain), and Base (Coinbase onboarding) dominate the optimistic rollup landscape with distinct competitive advantages.

5. EIP-4844 blobs reduced L2 fees by 10–100×; the fee structure is dominated by L1 data costs, making data compression and blob adoption critical for economics.

Secti

2 complete – next: Zero-Knowledge Rollups (zkSync, StarkNet, Polygon zkEVM) and validity proof mechanisms

Section 3: Zero-Knowledge Rollups

Validity proofs, SNARKs vs STARKs, and the mathematical approach to scaling

What You Will Learn

- ✓ ZK proof fundamentals: completeness, soundness, zero-knowledge
- ✓ ZK-SNARKs vs ZK-STARKs: tradeoffs and trust assumptions
- ✓ zkSync Era, StarkNet, and Polygon zkEVM architectures
- ✓ Prover costs, hardware requirements, and proving time optimization

Frames in This Section

- Frame 27: ZK Proof Fundamentals
- Frame 28: SNARKs vs STARKs
- Frame 29: ZK Rollup Architecture
- Frame 30: ZK Proof Verification (Code)
- Frames 31–35: zkSync, StarkNet, Polygon, Provers, Comparison
- Frame 36: Section Summary

Secti

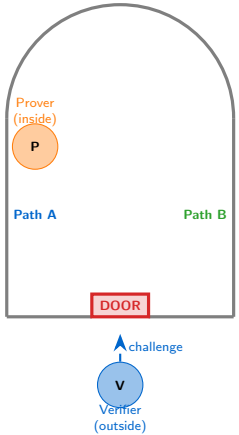
3 of 5 – Zero-Knowledge Rollups: prove validity with math

Three Properties of ZK Proofs

1. **Completeness**: If the statement is true, an honest prover can always convince an honest verifier.
2. **Soundness**: If the statement is false, no cheating prover can convince the verifier (except with negligible probability).
3. **Zero-Knowledge**: The verifier learns *nothing* beyond the truth of the statement – no information about the witness leaks.

Why It Matters for Rollups

ZK rollups use validity proofs: the prover demonstrates that all state transitions in a batch are correct *without* revealing individual transaction details (though in practice, rollups post data for DA).



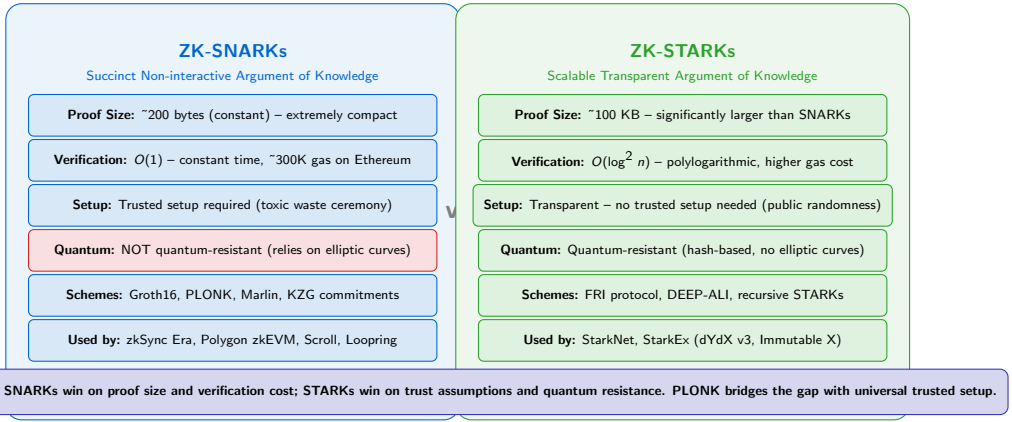
Protocol:
1. P enters cave (A or B)
2. V shouts: "Come out A!"
3. P emerges from A
4. Repeat n times
5. Cheat prob: $(1/2)^n$

After 20 rounds:
cheat prob $< 10^{-6}$
V is convinced
but learns **nothing**
about the secret

Ali

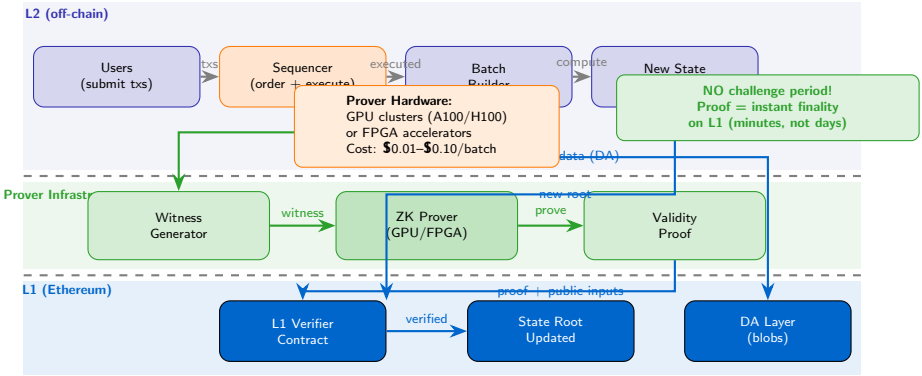
Baba's Cave: the prover proves knowledge of the secret (door password) without ever revealing it to the verifier

ZK-SNARKs vs ZK-STARKs



Both are validity proofs – they differ in cryptographic assumptions, proof size, and trust requirements

ZK Rollup Architecture



ZK

rollups replace the 7-day challenge window with a mathematical proof – instant L1 finality once the proof is verified

ZK Proof Verification

Listing 3: Simplified ZK Verifier

```
// Simplified ZK Verifier
contract ZKRollupVerifier {
    bytes32 public stateRoot;
    uint256 public batchNumber;

    function verifyAndUpdate(
        bytes32 newStateRoot,
        uint256[2] memory proofA,
        uint256[2][2] memory proofB,
        uint256[2] memory proofC,
        uint256[] memory publicInputs
    ) external {
        require(
            verifyProof(
                proofA, proofB, proofC,
                publicInputs
            ),
            "Invalid ZK proof"
        );
        stateRoot = newStateRoot;
        batchNumber++;
        emit StateUpdated(
            batchNumber, newStateRoot);
    }
}
```

Pairing-Based Verification

- **Groth16**: 3 elliptic curve pairings, ˜200K gas, circuit-specific trusted setup
- **PLONK**: Universal setup (reusable), slightly higher gas (˜300K), more flexible
- **Verification**: L1 contract checks $e(A, B) = e(\alpha, \beta) \cdot e(L, \gamma) \cdot e(C, \delta)$
- **Public inputs**: Previous state root, new state root, batch hash

Gas Cost Comparison

- ZK proof verification: ˜300K gas (fixed cost per batch)
- Optimistic batch posting: ˜40K gas (but 7-day delay)
- ZK amortized per tx: **$**0.001–**$**0.01 (spread across 1000s of txs)
- Cost dominated by **prover computation**, not verification

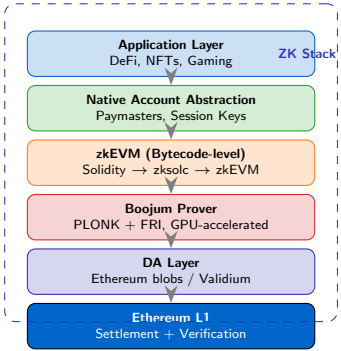
Key Insight

Verification is $O(1)$ regardless of batch size – a batch of 10 txs costs the same to verify as 10,000 txs. The prover bears the scaling cost.

ZK

Architecture Overview

- **zkEVM type**: Bytecode-level compatible (compiles Solidity via zksolc to custom bytecodes, then proves)
- **Proof system**: Boojum (PLONK-based with FRI), replaced previous Bellman prover
- **Native Account Abstraction**: Every account is a smart contract – no EOAs, built-in paymasters
- **ZK Stack**: Open-source modular framework for building custom ZK chains (Hyperchains)
- **Data Availability**: Posts to Ethereum L1 (rollup mode) or DA committee (validium mode)
- **Token**: ZK (governance, launched June 2024)



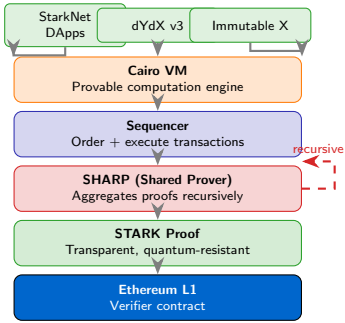
Key Innovation

Native AA means gasless transactions, session keys, and social recovery are protocol-level features, not afterthoughts.

Era pioneered native account abstraction and the ZK Stack framework for building sovereign ZK rollups

Architecture Overview

- **Proof system**: ZK-STARKs – no trusted setup, quantum-resistant, transparent
- **Language**: Cairo – purpose-built for provable computation (not EVM-compatible by default)
- **Recursive proofs**: SHARP (Shared Prover) aggregates proofs from multiple apps into one L1 proof
- **Execution model**: Cairo VM (not EVM) – different programming paradigm optimized for proof generation
- **Appchains**: StarkNet Stack allows building application-specific chains with shared proving
- **Token**: STRK (governance + fees, launched Feb 2024)



Key Innovation

SHARP lets multiple applications (StarkNet, dYdX v3, Immutable X, Sorare) share one prover and split verification costs on L1.

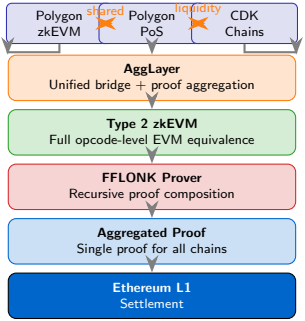
Stark

uses STARKs for quantum-resistant proofs and Cairo for provable computation – SHARP amortizes L1 costs across apps

Polygon zkEVM

Architecture Overview

- **zkEVM type**: Type 2 – full EVM equivalence at the opcode level (existing Solidity contracts deploy without changes)
- **Proving time**: ˜10 minutes per batch (improving with recursive proofs and hardware acceleration)
- **Proof system**: Custom SNARK (FFLONK variant with recursive composition)
- **AggLayer**: Unified bridge and proof aggregation layer connecting all Polygon chains
- **CDK (Chain Development Kit)**: Build sovereign ZK L2s with shared liquidity via AggLayer
- **Polygon 2.0**: Vision for network of ZK-connected chains with unified liquidity



Key Innovation

Type 2 zkEVM means zero migration cost – any Ethereum dApp works on Polygon zkEVM with no code changes. AggLayer unifies liquidity.

zkEVM offers full Solidity compatibility with zero migration cost – AggLayer connects all Polygon chains with shared liquidity

Prover Economics

Cost Breakdown Per Batch

GPU/FPGA Cluster: $10K–$100K+ capital (A100/H100 nodes)

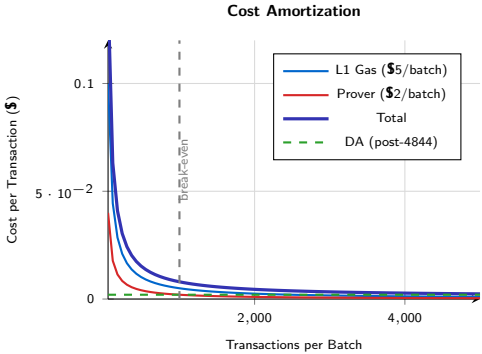
Electricity: $0.005–$0.05 per batch (high compute workload)

L1 Verification Gas: ~300K gas ≈ $3–$30 per batch

DA Cost (blobs): ~$0.01–$1 per batch (post-4844)

Amortized per tx: $0.001–$0.01 at scale (1000+ txs/batch)

Larger batches ⇒ lower cost per tx
ZK proving is the major ongoing expense
Hardware acceleration is a competitive moat



ZK prover costs are amortized across all transactions in a batch – economies of scale make larger batches dramatically cheaper per tx

ZK Rollup Comparison

Feature	zkSync Era	StarkNet	Polygon zkEVM
Proof System	PLONK + FRI (Boojum)	STARKs (SHARP)	FFLONK (recursive)
EVM Compatibility	Bytecode-level (Type 4)	None (Cairo VM)	Opcode-level (Type 2)
Smart Contract Language	Solidity (via zksolc)	Cairo	Solidity (native)
Trusted Setup	Yes (universal/PLONK)	No (transparent)	Yes (universal/PLONK)
Quantum Resistant	No	Yes	No
Finality (to L1)	~1 hour	~2–4 hours	~30 min
TVL (2024 peak)	~$1B	~$200M	~$100M
Avg Fee (post-4844)	$0.01–$0.10	$0.01–$0.05	$0.01–$0.05
Key Innovation	Native AA, ZK Stack	SHARP, Cairo lang	Type 2 zkEVM, AggLayer
Appchain Framework	ZK Stack (Hyperchains)	StarkNet Stack	CDK (Chain Dev Kit)
Prover Hardware	GPU (Boojum)	CPU + GPU (Stone)	GPU (FFLONK)

zkSync Strengths

- Native Account Abstraction
- ZK Stack for sovereign chains
- Boojum: efficient GPU proving

StarkNet Strengths

- No trusted setup (transparent)
- Quantum-resistant proofs
- SHARP: shared proving costs

Polygon zkEVM Strengths

- Full Solidity compatibility
- AggLayer: unified liquidity
- CDK for custom L2 chains

Each

ZK rollup makes different tradeoffs – zkSync (AA), StarkNet (STARKs), Polygon (EVM compat) – all settling on Ethereum

Section 3 Summary

1. Zero-knowledge proofs provide completeness, soundness, and zero-knowledge – enabling validity proofs that verify state transitions without re-executing them.

2. SNARKs offer tiny proofs and $O(1)$ verification but require trusted setup; STARKs are transparent and quantum-resistant but produce larger proofs with $O(\log^2 n)$ verification.

3. ZK rollups achieve instant L1 finality (no 7-day wait) by submitting validity proofs – the prover bears the computational cost, while L1 verification is cheap and constant-time.

4. zkSync Era (native AA, ZK Stack), StarkNet (STARKs, Cairo, SHARP), and Polygon zkEVM (Type 2, AggLayer) represent three distinct approaches to ZK scaling.

5. Prover economics are dominated by hardware costs (GPU/FPGA clusters); amortization across large batches drives per-transaction costs below \$0.01 at scale.

Secti

3 complete – next: Sidechains & Alternative Scaling approaches

Section 4: Sidechains & Alternative Scaling

Beyond rollups: sidechains, modular blockchains, and data availability innovations

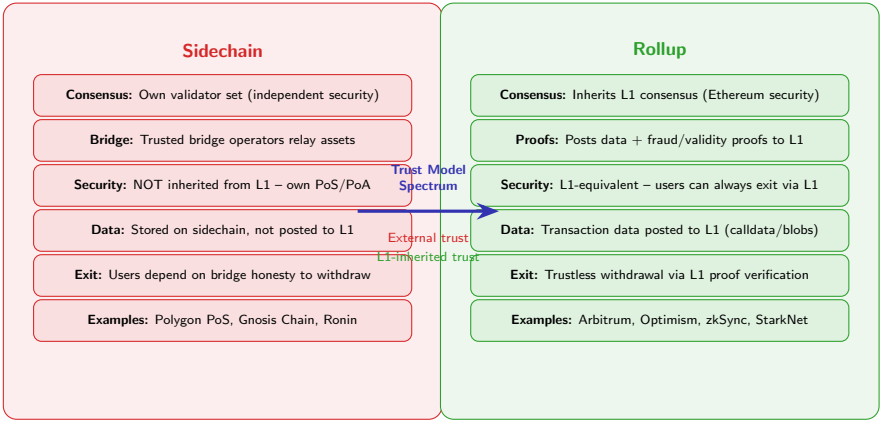
What You Will Learn

- ✓ Sidechains vs rollups: trust models and security tradeoffs
- ✓ Polygon PoS and Avalanche subnet architectures
- ✓ The modular blockchain thesis and separation of concerns
- ✓ EIP-4844 blobs, Celestia, and data availability innovations

Frames in This Section

- Frame 38: Sidechains vs Rollups
- Frame 39: Polygon PoS
- Frame 40: Avalanche Subnets
- Frame 41: Modular Blockchain Thesis
- Frame 42: Bridge Contract (Code)
- Frames 43–46: EIP-4844, Celestia, Danksharding, Validiums
- Frame 47: Section Summary

Sidechains vs Rollups



Sidec

have independent security; rollups inherit L1 security – the key distinction is the trust model for withdrawals

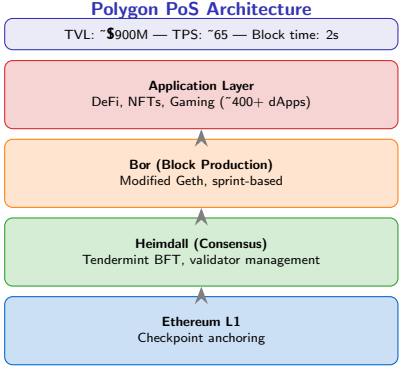
Polygon PoS

Architecture Overview

- **Validator Set**: ˜100 validators running Proof of Stake
- **Heimdall Layer**: Tendermint-based consensus for checkpointing
- **Bor Layer**: Block production (modified Geth, EVM-compatible)
- **Checkpoints**: Periodic state commitments to Ethereum L1
- **Bridge**: PoS bridge for asset transfers (trusted validator set)

Transition to zkEVM

- Polygon PoS transitioning to a **validium** secured by ZK proofs
- **AggLayer**: Unified bridge aggregating ZK proofs from multiple chains
- Goal: Sidechain convenience with rollup-level security
- POL token replacing MATIC for expanded utility



Polyg

PoS is a sidechain with its own consensus, transitioning toward ZK-secured validium via the AggLayer

Avalanche Subnets

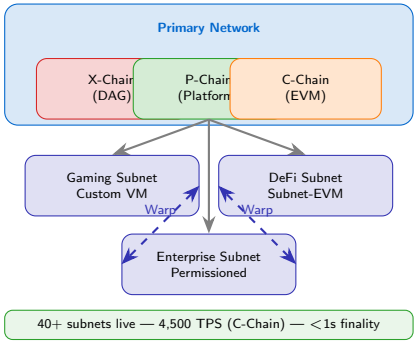
Multi-Chain Architecture

- **X-Chain**: DAG-based, for asset creation and exchange (Avalanche consensus)
- **P-Chain**: Platform chain for validator coordination and subnet management
- **C-Chain**: Contract chain, EVM-compatible (Snowman consensus)
- **Subnets**: Custom blockchains with independent validator sets and VMs

Key Innovations

- **Elastic Subnets**: Stake AVAX to validate custom chains
- **Custom VMs**: Any virtual machine (Subnet-EVM, Move VM, WASM)
- **Warp Messaging**: Native cross-subnet communication protocol
- **Sub-second finality**: Snowball consensus reaches finality in <1 second

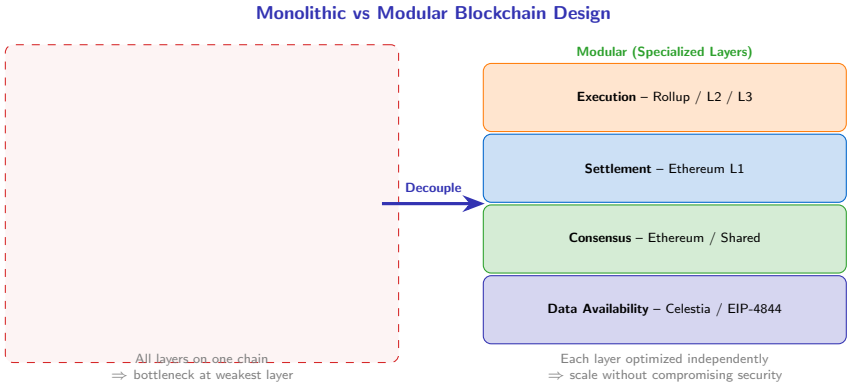
Avalanche Subnet Architecture



Avala

subnets allow custom blockchain deployments with independent VMs and validators, connected via Warp messaging

The Modular Blockchain Thesis



Modu

blockchains separate execution, settlement, consensus, and data availability into specialized optimized layers

Bridge Contract

Listing 4: L1 Bridge: Lock Tokens

```
// L1 Bridge: Lock tokens
contract L1Bridge {
    mapping(bytes32 => bool)
        public processedDeposits;

    function deposit(
        address token,
        uint256 amount,
        address l2Recipient
    ) external {
        IERC20(token).transferFrom(
            msg.sender,
            address(this),
            amount
        );
        bytes32 depositHash =
            keccak256(abi.encode(
                token, amount,
                l2Recipient,
                block.number
            ));
        emit DepositInitiated(
            depositHash,
            token, amount,
            l2Recipient);
    }
}
```

Lock-and-Mint Mechanism

- **L1 Lock**: User deposits tokens into the L1 bridge contract; tokens are held in escrow
- **L2 Mint**: Bridge relayer detects deposit event; L2 contract mints equivalent wrapped tokens
- **Withdraw**: Reverse – burn wrapped tokens on L2, unlock originals on L1
- **Deposit hash**: Unique identifier prevents replay attacks

Bridge Risks

- **Smart contract bugs**: Bridge contracts hold billions in TVL – a single vulnerability is catastrophic
- **Validator compromise**: Trusted bridges rely on honest multisig or validator set
- **Censorship**: Bridge operators can freeze or censor withdrawals
- **Liquidity fragmentation**: Wrapped tokens $\neq$ native tokens across chains

Lock-and-mint bridges are the most common cross-chain mechanism – they hold billions in TVL and are prime targets for exploits

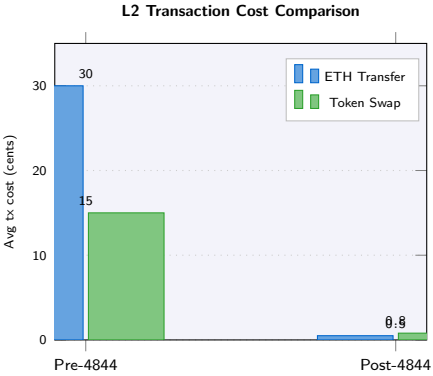
EIP-4844 & Blob Transactions

Proto-Danksharding (EIP-4844)

- **New transaction type**: Type-3 blob-carrying transactions
- **Blob size**: ~128 KB per blob, up to 6 blobs per block
- **Separate fee market**: Blob gas price independent of execution gas
- **Pruned after ~18 days**: Not permanently stored on-chain
- **Cost reduction**: 10–100× cheaper than calldata for L2s

Impact on L2 Economics

- Pre-4844: L2s paid $0.10–$1.00 per tx for calldata
- Post-4844: $0.001–$0.01 per tx via blobs
- **Target**: 3 blobs/block, max 6 blobs/block
- Blob fee adjusts via EIP-1559-style mechanism
- Launched **March 13, 2024** (Dencun upgrade)



Result: L2 fees dropped ~98% post-Dencun

EIP-4844 introduced blob transactions (March 2024) – a dedicated cheap data layer that reduced L2 costs by 10–100×

Celestia & Modular DA

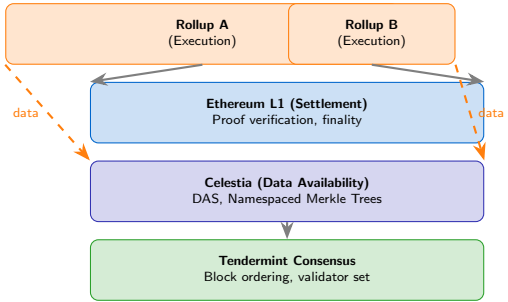
Celestia Architecture

- **Purpose-built DA layer**: Only orders and publishes data – no execution
- **Data Availability Sampling (DAS)**: Light nodes sample random chunks to verify availability
- **Namespaced Merkle Trees (NMTs)**: Each rollup reads only its own namespace
- **Throughput**: ˜6.67 MB/s data throughput target
- **Security model**: Honest majority of light nodes ensures DA

DA vs Execution

- Celestia provides **only data availability** – no smart contracts, no execution
- Rollups post data to Celestia instead of (or in addition to) Ethereum
- Tradeoff: Cheaper DA but weaker security guarantees than Ethereum DA
- Sovereign rollups: Rollups that settle on Celestia without an L1 settlement layer

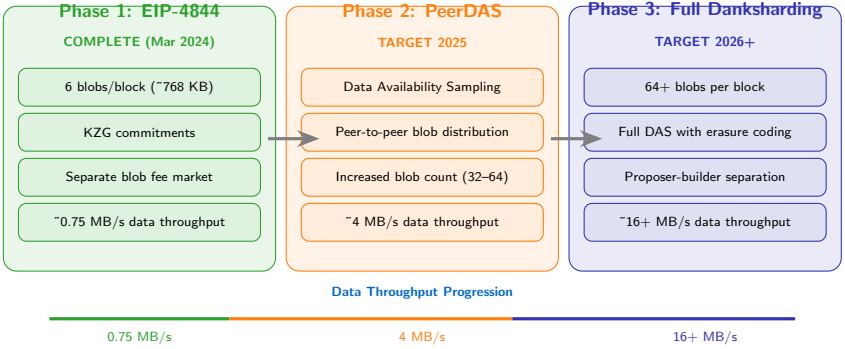
Modular Stack with Celestia



Celes

is a purpose-built data availability layer – rollups post data to Celestia for cheap DA while settling proofs on Ethereum

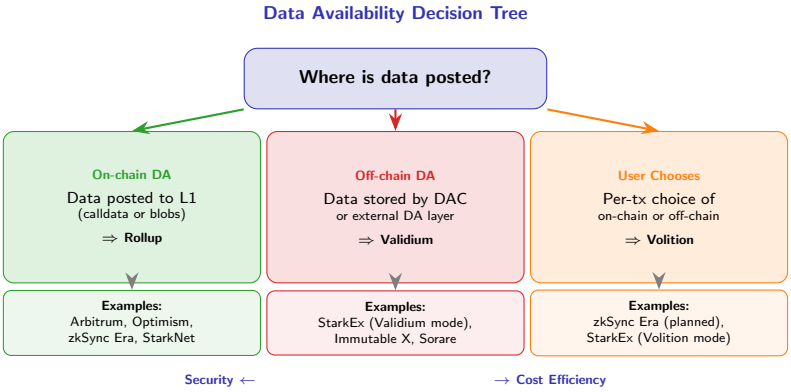
Danksharding Roadmap



Dank

is Ethereum's DA roadmap: EIP-4844 (done) → PeerDAS (2025) → Full Danksharding (2026+) with 20× more throughput

Validiums & Volitions



Rollu

store data on-chain (max security); validiums store off-chain (cheaper); volitions let users choose per transaction

Section 4 Summary

1. Sidechains have their own consensus and security model; rollups inherit L1 security – the fundamental distinction is the trust model for user withdrawals.

2. Polygon PoS (˜100 validators, checkpoints) and Avalanche subnets (custom VMs, sub-second finality) offer high throughput with independent security assumptions.

3. The modular thesis separates execution, settlement, consensus, and data availability into specialized layers – each optimized independently for maximum throughput.

4. EIP-4844 blob transactions reduced L2 costs by 10–100×; the Danksharding roadmap targets 16+ MB/s data throughput via PeerDAS and full sharding.

5. Validiums store data off-chain for cost savings; volitions let users choose per-transaction – the DA spectrum trades security for efficiency.

Secti

4 complete – next: L2 Ecosystem & Future (bridges, economics, L3s, account abstraction)

Section 5: L2 Ecosystem & Future

Bridge security, L2 economics, L3s, and the rollup-centric roadmap

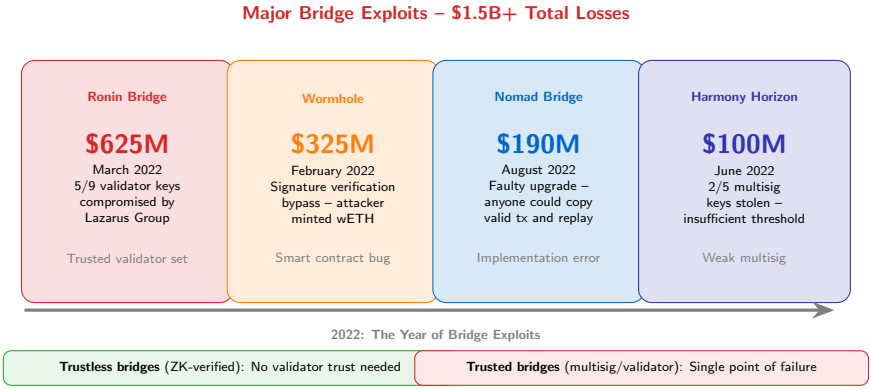
What You Will Learn

- ✓ Bridge security vulnerabilities and major hack post-mortems
- ✓ Cross-L2 communication: shared sequencing and intent protocols
- ✓ L2 economics: sequencer revenue, MEV, and cost structures
- ✓ L3s, account abstraction, and the rollup-centric endgame

Frames in This Section

- Frame 49: Bridge Security & Major Hacks
- Frame 50: Cross-L2 Communication
- Frame 51: L2 Economics
- Frame 52: MEV on Layer 2
- Frame 53: L3s & App-Specific Chains
- Frame 54: Account Abstraction on L2
- Frame 55: Key Takeaways and Course Summary

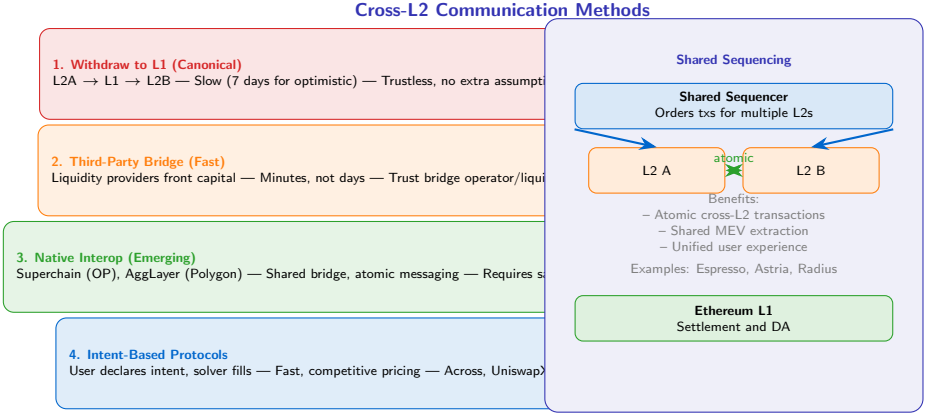
Bridge Security & Major Hacks



Bridg

exploits caused $1.5B+ in losses in 2022 alone – trustless ZK-verified bridges are the industry response to validator trust risks

Cross-L2 Communication



Cross

L2 communication ranges from slow canonical routes to fast intent-based protocols – shared sequencing enables atomic cross-rollup transactions

L2 Economics

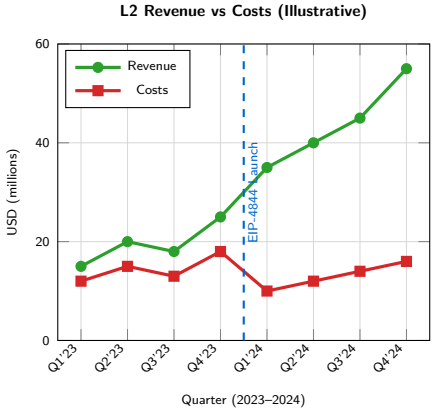
Revenue vs Cost Structure

Revenue Sources

- **Transaction Fees:** Base fee + priority fee from users
- **MEV Extraction:** Sequencer captures ordering value
- **Blob Arbitrage:** Profit from L2 fee > L1 blob cost

Cost Components

- **L1 Data Posting:** Blob/calldata fees to Ethereum
- **Proof Verification:** ZK proof gas or fraud proof bonds
- **Infrastructure:** Sequencer, prover, node operations



L2

economics: revenue from fees and MEV vs costs of L1 data posting and proofs – EIP-4844 dramatically improved margins

MEV on Layer 2

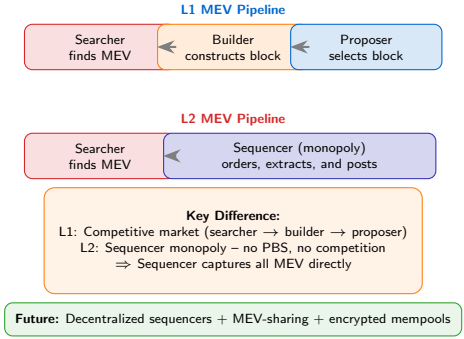
L2 MEV Context

- **Sequencer monopoly**: Single sequencer controls transaction ordering – no open mempool
- **MEV types**: Arbitrage, liquidations, sandwich attacks, NFT sniping
- **Scale**: Estimated $100M+ annual MEV across major L2s
- **Fair ordering**: FCFS, encrypted mempools, MEV-sharing proposed

Mitigation Approaches

- **FCFS ordering**: Process transactions in arrival order (Arbitrum)
- **MEV-Share**: Return MEV to users via kickback mechanisms
- **Threshold encryption**: Encrypt txs until ordering is committed
- **Decentralized sequencing**: Remove single-party ordering power

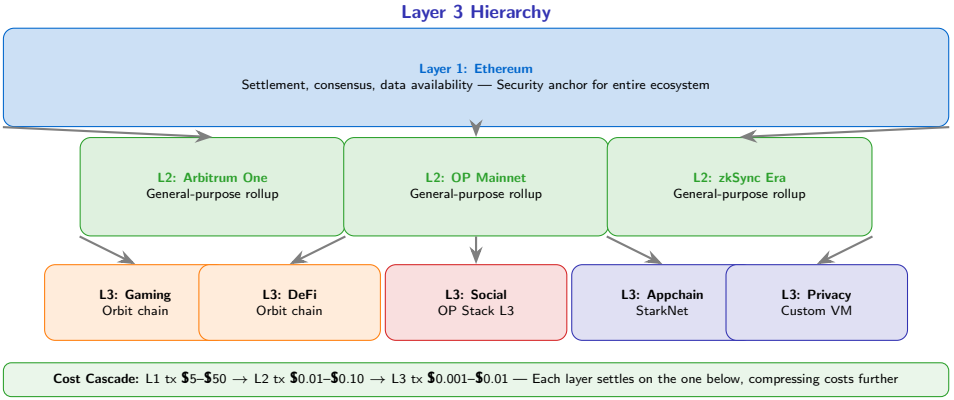
MEV Flow Comparison



L2

sequencers hold monopoly over transaction ordering and MEV extraction – decentralized sequencing and fair ordering are active research areas

L3s & App-Specific Chains



L3s settle on L2s (which settle on L1) – app-specific chains via Orbit, OP Stack, or StarkNet Appchains for ultra-low-cost domain-specific execution

Account Abstraction on L2

What is Account Abstraction?

- **ERC-4337**: Smart contract wallets as first-class accounts (L1 standard)
- **Native AA**: Built into the protocol (zkSync Era, StarkNet)
- **Key insight**: Every account is a smart contract – programmable validation logic

Benefits for Users

- **Gasless transactions**: Paymasters sponsor gas on behalf of users
- **Social recovery**: Recover wallet via trusted contacts (no seed phrase)
- **Session keys**: Time-limited keys for dApps (no per-tx signing)
- **Batched operations**: Multiple actions in a single transaction
- **Multi-sig built-in**: Require N-of-M signatures natively

UX Comparison

Traditional EOA Flow

- User manages seed phrase (12–24 words)
- Must hold ETH for gas before any action
- Sign every single transaction manually
- Lost key = lost funds (no recovery)

Account Abstraction Flow

- Email/social login, biometric auth
- Paymaster covers gas – zero ETH needed
- Session keys: approve once, interact freely
- Social recovery via guardians

Acco

abstraction transforms L2 UX: gasless txs, social recovery, session keys – L2s with native AA (zkSync, StarkNet) lead adoption

Key Takeaways and Course Summary

1. Scaling Fundamentals: The blockchain trilemma limits L1 throughput; Layer 2 solutions inherit L1 security while processing transactions off-chain at 10–100$\times$ lower cost.

2. Optimistic Rollups: Fraud proofs with 7-day challenge windows; Arbitrum, Optimism, and Base dominate with the OP Stack enabling a Superchain of interoperable L2s.

3. ZK Rollups: Validity proofs provide instant finality without challenge periods; SNARKs vs STARKs tradeoffs, with zkSync, StarkNet, and Polygon zkEVM leading adoption.

4. Alternative Scaling: Sidechains (own security), modular design (specialized layers), and EIP-4844 blobs fundamentally changed L2 economics with the Danksharding roadmap ahead.

5. L2 Ecosystem: Bridge security remains critical ($1.5B+ lost); L3s, account abstraction, and shared sequencing are shaping the rollup-centric future of Ethereum.

Next: Advanced topics in cross-chain interoperability, shared sequencing, and based rollups

Cours

complete – 5 sections, 55 frames — The rollup-centric roadmap is Ethereum's path to global-scale decentralized computation