Intro to Layer 2 Rollups

Pankaj Gupta
pankaj@coinbase.com

For MS&E 339, Stanford University
Nov 29, 2022

Overview of this talk

- Scaling blockchains(with focus on Ethereum)
- History of Layer 2 technologies
- Why Rollups are exciting
- Types of Rollups
- Optimistic Rollups
- ZK Rollups
- Open problems

Hat-Tip and Excellent reading: “An incomplete guide to rollups”, Vitalik Buterin,
https://vitalik.ca/general/2021/01/05/rollup.html
See also: SOK: Layer-Two Blockchain Protocols (2019, doesn’t mention rollups)

ABOUT ME

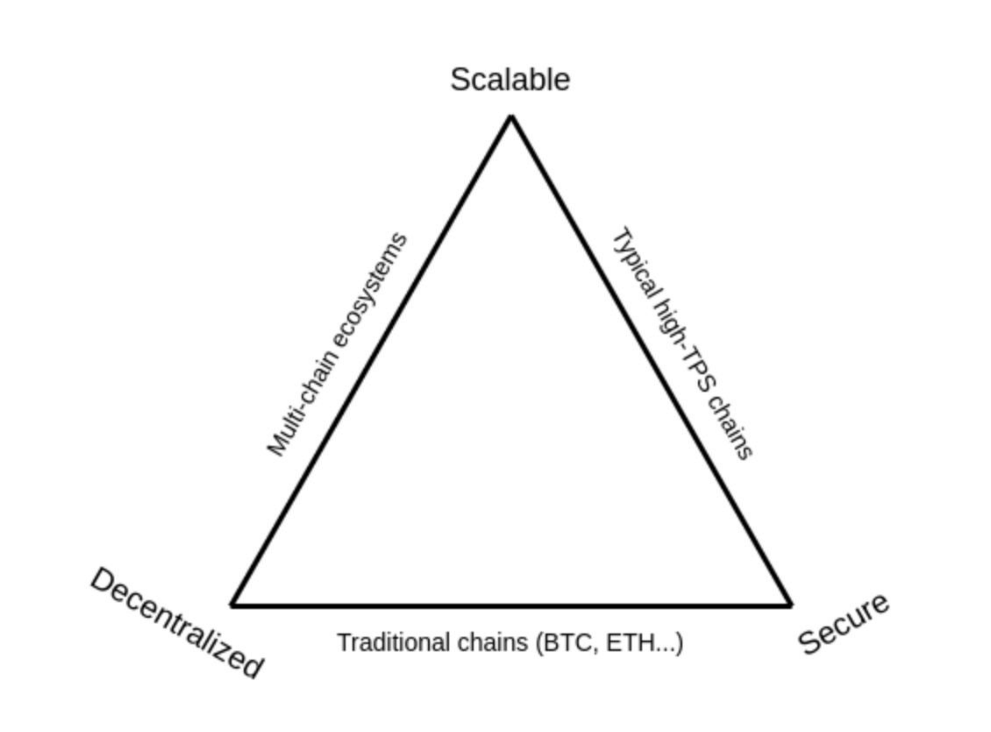
- Pankaj Gupta
  - @pankaj (Twitter)
- VP Engineering at Coinbase
  - Leading engineering for Coinbase consumer products (apps, www.coinbase.com)
- Previously:
  - Google (2017-21) - led eng for Google Pay (Consumer) and Google Pay India
  - Twitter (2009-14) - Search & Recommendations
  - Co-founded 2 startups in between
- PhD Stanford CS 2001

Overview of scaling blockchains

Blockchains are slow

- Bitcoin: 3-7 TPS (Transactions per sec)
- Ethereum: 10-25 TPS (Transfer Transactions)
- Contrast with most payment networks (eg Visa) at ~10K-50K TPS

Why are Blockchains slow?



Scalability Trilemma. https://vitalik.ca/general/2021/04/07/sharding.html

ON PERFORMANCE

- What is performance?
    - Throughput - TPS
    - Latency - quicker “settlement finality”
- Why do we want to improve blockchain performance?
    - Faster payments - currently in O(minutes)
    - If blockchain is a “world computer”, higher performance -> more stuff cheaper -> more decentralized

https://www.paradigm.xyz/2022/07/consensus-throughput

What is the constrained resource?

- Decentralization means:
  - Your normal laptop can run a full node
    - Laptop should be able to store and process the whole blockchain
    - Possible today for Ethereum but nearing limits
  - We do not want only “industrial-grade” computers to be blockchain nodes
- Constrained resources:
  - Computation (laptop CPU)
  - Data network bandwidth
  - Disk storage (this is the **main bottleneck today**)

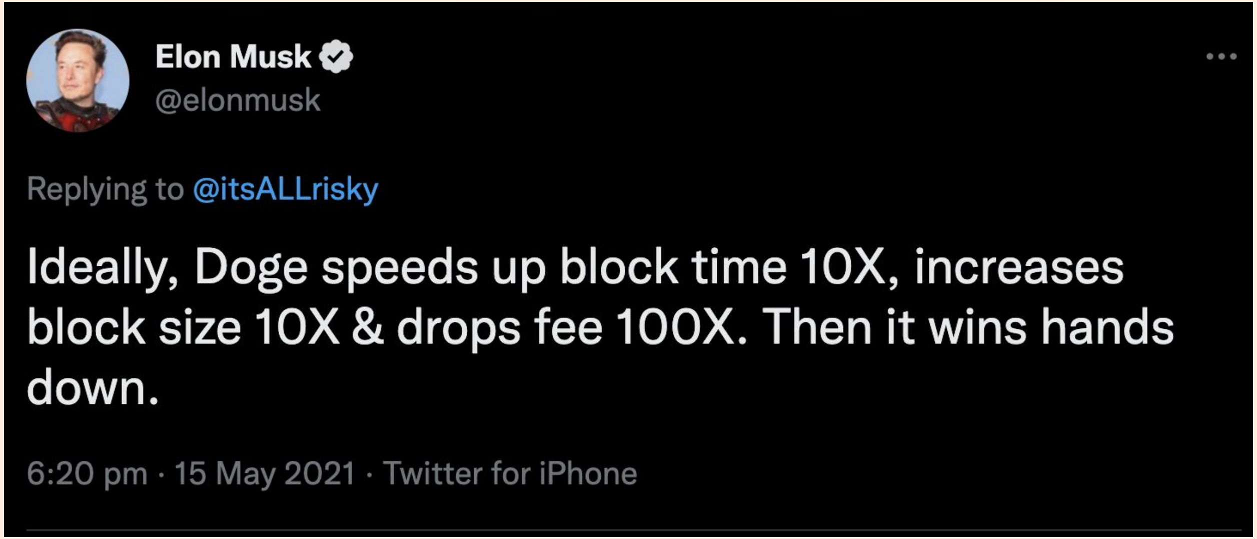
https://vitalik.ca/general/2021/05/23/scaling.html

Three ways to scale blockchains

Hint: A little bit like scaling traditional databases

1/ Vertical scaling

Make blockchain parameters “bigger”



Elon Musk
@elonmusk

Replying to @itsALLrisky

Ideally, Doge speeds up block time 10X, increases block size 10X & drops fee 100X. Then it wins hands down.

6:20 pm · 15 May 2021 · Twitter for iPhone

Problem: Makes it beyond normal laptops

2/ Layer 2

- Introduce hierarchy of computation/data
  - Like putting caches in front of DB
- “Off-chain” ie, offload computation from main L1 chain for performance
- But still derive security from L1 chain
  - Store some compressed data on L1
  - Allow anyone to verify that L2 processed transactions correctly

3/ Sharding

- Today: every node has to store and process all blockchain txns and state
- Future: A node only handles a subset
- Sharding is required eventually for horizontal scaling

Problem: Complex to get right. We will get there eventually in a series of steps.

History of layer 2s

State channels (2015)

- Eg Lightning network on Bitcoin
- Open a 1:1 payment channel between A and B
  - A pre-funds (“locks up”) say 1 BTC on L1
  - A signs off-chain “I-promise-to-pay” messages and sends directly to B: “0.1”, then “0.2” etc.
  - To settle the payment, either A or B can close the channel, and L1 ensures they get their dues
- Powerful technique, but downsides:
  - Complex to generalize to arbitrary computation
  - Participants (or delegates) have to be online, even if just receiving

https://www.jeffcoleman.ca/state-channels/

Plasma , Commit-chains (2017-19)

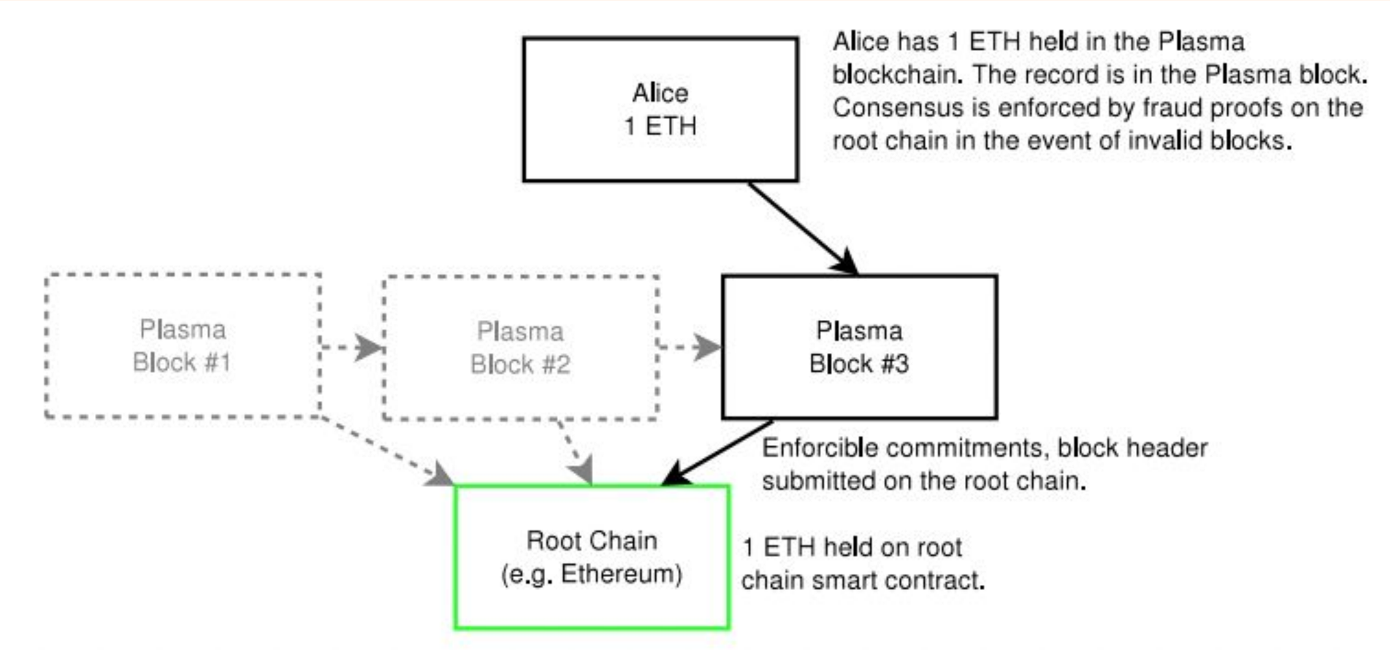
- Computation + state is in a different chain – aka *child* (L2) chain or *plasma* chain
- Child chain’s operator can be untrusted
- A smart contract on root (L1) chain manages the L2 chain
- “Commitments” to L2 chain state (Plasma block header hashes or Merkle roots of balances) are published to L1 periodically

https://eprint.iacr.org/2018/642
https://plasma.io/

Plasma, Commit-chains (contd.)

- If there is proof of fraud submitted on L1, then the block is rolled back and the block creator is penalized
- Any user can deposit into or withdraw from the L2 chain
- On operator misbehavior, a user can do a “forced exit” via the smart contract

Plasma , Commit-chains (contd.)



Alice has 1 ETH held in the Plasma blockchain. The record is in the Plasma block. Consensus is enforced by fraud proofs on the root chain in the event of invalid blocks.

Enforcible commitments, block header submitted on the root chain.

1 ETH held on root chain smart contract.

Block withholding problem

- What if no L2 block is produced by the operator(s)
- A user can completely exit the Plasma chain and withdraw their assets by submitting a Merkle proof of their ownership of assets

Data availability problem

- Only commitments are available on L1
- Is full data to produce those hashes available somewhere for any user to be able to verify or challenge the commitment?

Weakness of plasma

- Assets need to have owners
- Complicated to generalize to full EVM
- Fundamental game theory issues around data availability

ROLLUPS

ROLLUPS

- Hybrid L2 scaling mechanism
- Computation is still off-chain in L2
- But (compressed) per-transaction data is kept on L1
  - This eliminates the data availability problem
  - Leverage L1’s consensus and security
  - Anyone can locally process all rollup operations, and thus withdraw/verify/challenge L2’s computation
- However, scaling is limited by L1’s data bandwidth

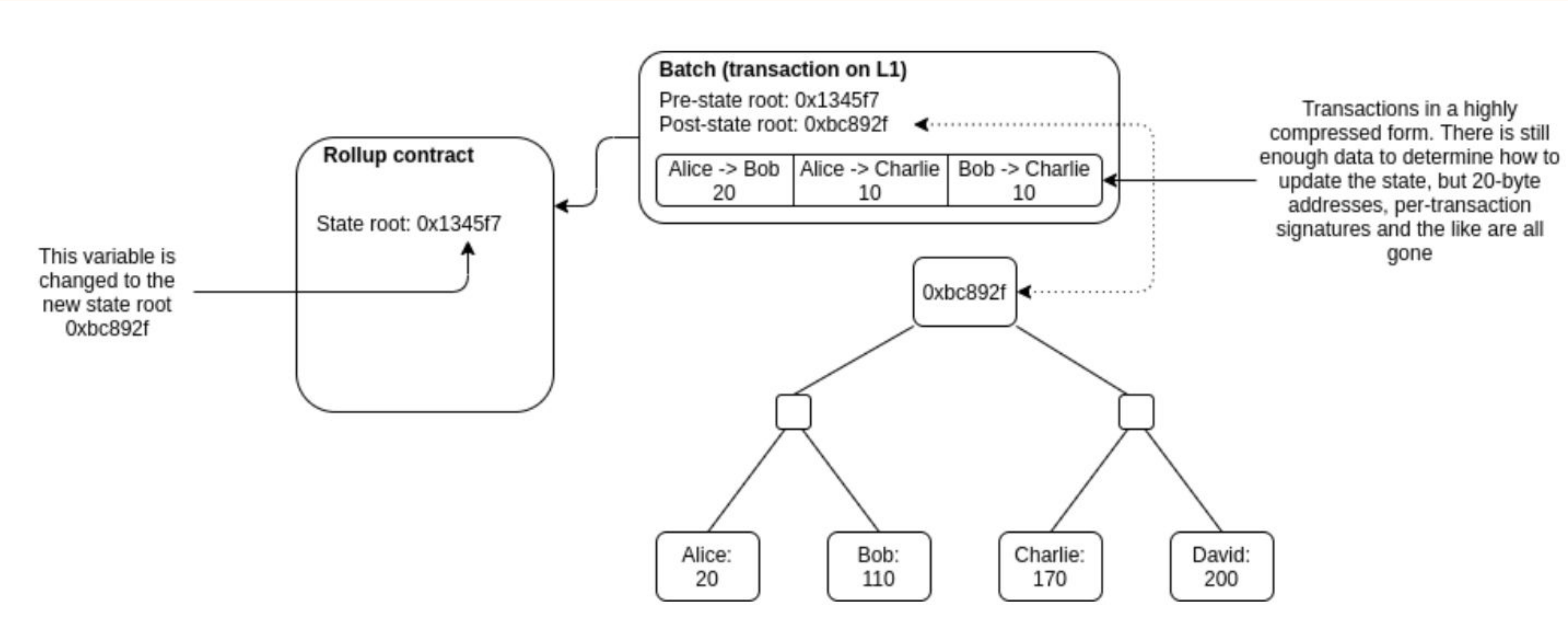
Rollups (contd.)

- Fully general EVM computations can be done on L2
- In fact, many L2 rollups run EVMs
- Dapps that are running on L1 can now be easily ported to run on L2s

Architecture of a rollup

- (Like in commit chains) Smart contracts on L1 that manage the L2, and typically allow
  - Deposit, Withdrawal
  - Publish “assertions” of new state roots
  - In addition, ability to publish a batch of compressed txn data, just sufficient to allow anyone to verify correctness
- An operator (aka validator / aggregator/ sequencer) on L2 which batches txns

Architecture of a rollup



Transactions in a highly compressed form. There is still enough data to determine how to update the state, but 20-byte addresses, per-transaction signatures and the like are all gone

This variable is changed to the new state root 0xbc892f

How to guarantee batch is correct?

Two families of rollups - Optimistic and ZK rollups

Optimistic rollups

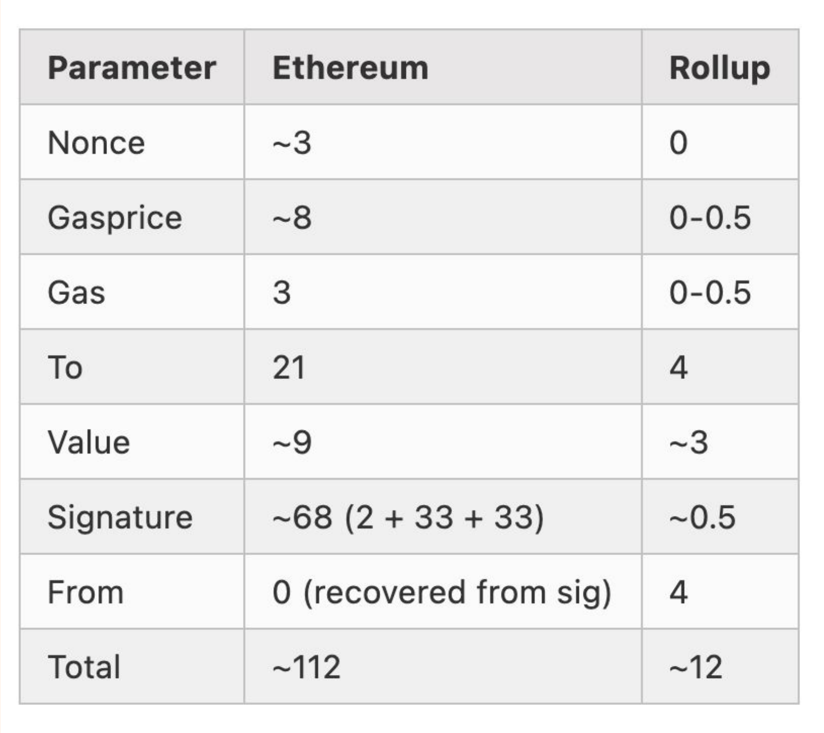
- Eg Arbitrum, Optimism
- Publish compressed txn data (as “calldata”)
  - This incurs a “fixed cost” per batch (eg 21K gas)
- A challenge window of N (~7-14) days for anyone to provide a *“fraud proof”* to rollup’s smart contract
  - If provided and found correct, state roots from then on are invalidated + Publisher’s deposit is slashed

https://ethereum.org/en/developers/docs/scaling/optimistic-rollups/

Fraud proofs

- **Single-round**
  - A verifier L1 contract replays the published txns on top of latest Merkle root to compute new root
  - Expensive
- **Multi-round**
  - To reduce costs
  - Interactive protocol between asserter and challenger
  - “Bisection protocol” like binary search to zero-in on one txn of dispute

Compression in a batch



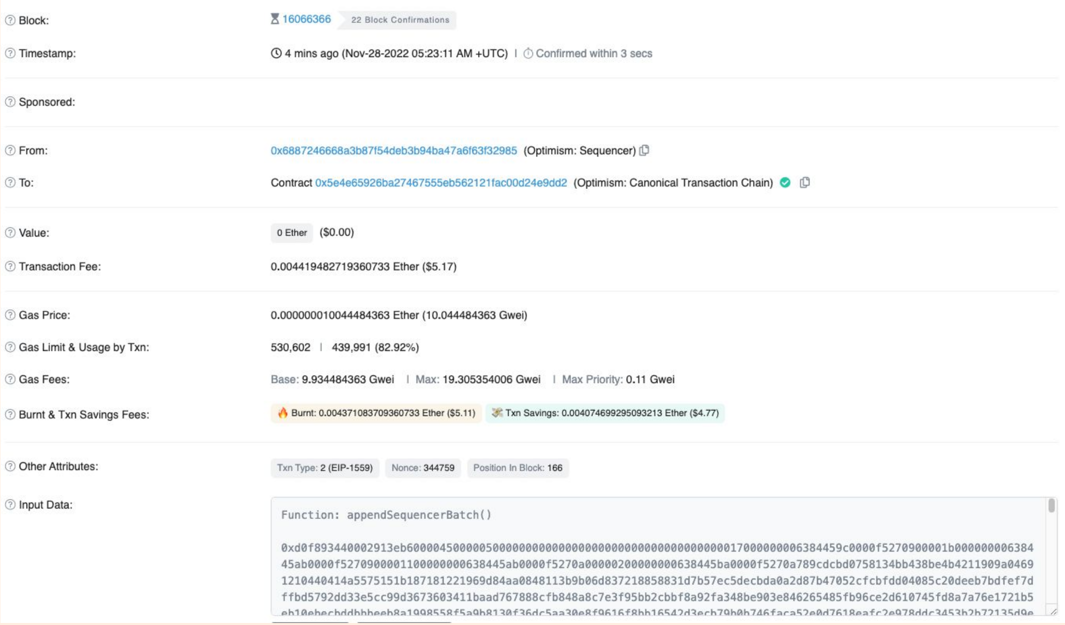
Parameter	Ethereum	Rollup
Nonce	~3	0
Gasprice	~8	0-0.5
Gas	3	0-0.5
To	21	4
Value	~9	~3
Signature	~68 (2 + 33 + 33)	~0.5
From	0 (recovered from sig)	4
Total	~112	~12

Can further use compression
algorithms like Zlib (Optimism)
Or Brotli (Arbitrum)

https://vitalik.ca/general/2021/01/05/rollup.html

Optimism batch submission

https://etherscan.io/tx/0xbc9ca2074024bf74466ba19dd5196528c5ddca435b3d2c2b2df28ecf15b7256a



Block:	⌛ 16066366 22 Block Confirmations
Timestamp:	4 mins ago (Nov-28-2022 05:23:11 AM +UTC) | Confirmed within 3 secs
Sponsored:	
From:	0x6887246668a3b87f54deb3b94ba47a6f63f32985 (Optimism: Sequencer)
To:	Contract 0x5e4e65926ba27467555eb562121fac00d24e9dd2 (Optimism: Canonical Transaction Chain) ✓
Value:	0 Ether ($0.00)
Transaction Fee:	0.004419482719360733 Ether ($5.17)
Gas Price:	0.000000010044484363 Ether (10.044484363 Gwei)
Gas Limit & Usage by Txn:	530,602 | 439,991 (82.92%)
Gas Fees:	Base: 9.934484363 Gwei | Max: 19.305354006 Gwei | Max Priority: 0.11 Gwei
Burnt & Txn Savings Fees:	🔥 Burnt: 0.004371083709360733 Ether ($5.11) Txn Savings: 0.004074699295093213 Ether ($4.77)
Other Attributes:	Txn Type: 2 (EIP-1559) Nonce: 344759 Position In Block: 166
Input Data:	

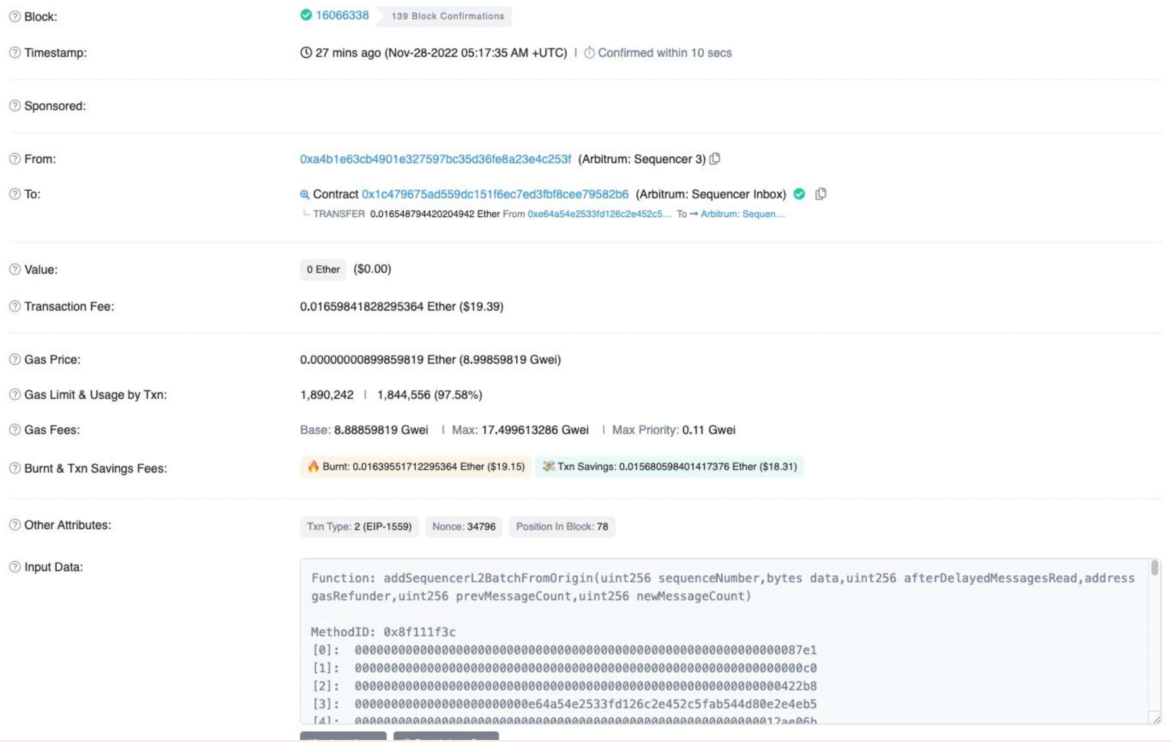
```
Function: appendSequencerBatch()

0xd0f893440002913eb60000450000050000000000000000000000000000000000000170000000006384459c0000f5270900001b000000006384
45ab0000f5270900001100000000638445ab0000f5270a000002000000000638445ba0000f5270a789cdcbd0758134bb438be4b4211909a0469
1210440414a5575151b187181221969d84aa0848113b9b06d837218858831d7b57ec5decbda0a2d87b47052cfcbfdd04085c20deeb7bdfef7d
ffbd5792dd33e5cc99d3673603411baad767888cfb848a8c7e3f95bb2cbbf8a92fa348be903e846265485fb96ce2d610745fd8a7a76e1721b5
eb10ebecbddbbbeeb8a1998558f5a9b8130f36dc5aa30e8f9616f8bb16542d3ecb79b0b746faca52e0d7618eafc2e978ddc3453b2b72135d9e
```

~100 transactions
~4K gas/txn
~$0.05/txn

Arbitrum batch submission

https://etherscan.io/tx/0x99936cfde6004708b61124de0af93c79b5fae4890b17dd02b5ca6506e77d9402



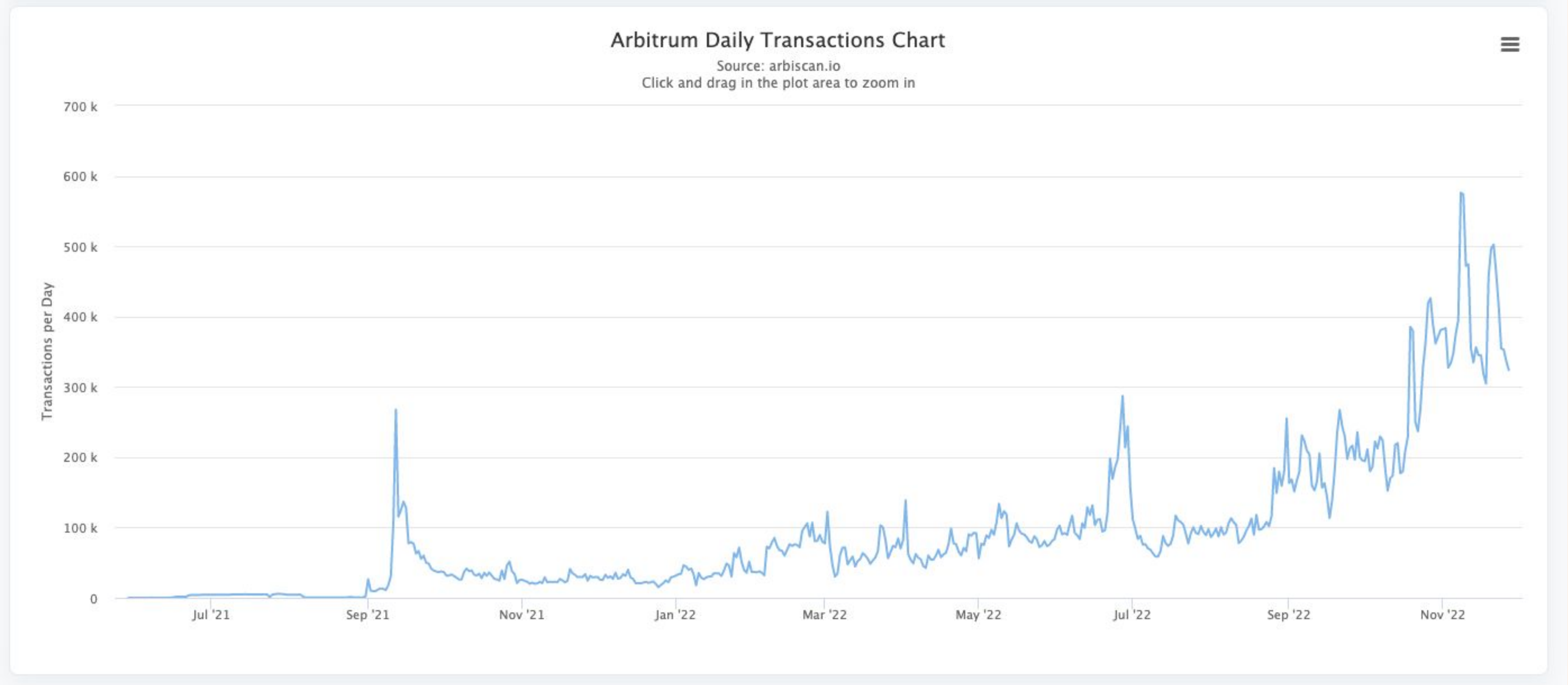
ⓘ Block:	✓ 16066338 139 Block Confirmations
ⓘ Timestamp:	27 mins ago (Nov-28-2022 05:17:35 AM +UTC) | Confirmed within 10 secs
ⓘ Sponsored:	
ⓘ From:	0xa4b1e63cb4901e327597bc35d36fe8a23e4c253f (Arbitrum: Sequencer 3)
ⓘ To:	Contract 0x1c479675ad559dc151f6ec7ed3fbf8cee79582b6 (Arbitrum: Sequencer Inbox) ✓ └ TRANSFER 0.016548794420204942 Ether From 0xe64a54e2533fd126c2e452c5… To → Arbitrum: Sequen…
ⓘ Value:	0 Ether ($0.00)
ⓘ Transaction Fee:	0.01659841828295364 Ether ($19.39)
ⓘ Gas Price:	0.00000000899859819 Ether (8.99859819 Gwei)
ⓘ Gas Limit & Usage by Txn:	1,890,242 | 1,844,556 (97.58%)
ⓘ Gas Fees:	Base: 8.88859819 Gwei | Max: 17.499613286 Gwei | Max Priority: 0.11 Gwei
ⓘ Burnt & Txn Savings Fees:	Burnt: 0.01639551712295364 Ether ($19.15) Txn Savings: 0.015680598401417376 Ether ($18.31)
ⓘ Other Attributes:	Txn Type: 2 (EIP-1559) Nonce: 34796 Position In Block: 78
ⓘ Input Data:	

```
Function: addSequencerL2BatchFromOrigin(uint256 sequenceNumber,bytes data,uint256 afterDelayedMessagesRead,address
gasRefunder,uint256 prevMessageCount,uint256 newMessageCount)

MethodID: 0x8f111f3c
[0]:  00000000000000000000000000000000000000000000000000000000000087e1
[1]:  00000000000000000000000000000000000000000000000000000000000000c0
[2]:  00000000000000000000000000000000000000000000000000000000000422b8
[3]:  000000000000000000000000e64a54e2533fd126c2e452c5fab544d80e2e4eb5
[4]:  000000000000000000000000000000000000000000000000000000000012ae06b
```

~1000 transactions
~1.5K gas/txn
~$0.02/txn

L2 transactions are growing



Zero knowledge rollups

- Eg zkSync, Starknet
- Prover publishes a “Validity Proof” proving that submitted transactions are valid
- Typically via ZK-SNARKS (or STARKS)
- Verification of the proof takes much less time than to redo the entire computation in the batch
- Not automatically privacy preserving

https://ethereum.org/en/developers/docs/scaling/zk-rollups/

ZK rollups (contd.)

- Data availability
  - Changes in balance are often included with proofs
- Instant finality
  - No need to wait for challenge period of 7-14 days
- But proof takes time to construct: O(mins)
  - Fixed cost is high
- Recursive proofs (“proofs-of-proofs”) allow even higher scalability

L2 rollups rely on L1 for:

- Data availability
- Censorship resistance
  - If L2 operator goes offline or misbehaves, user can do a “forced exit” by submitting its own batch on L1 chain
- Dispute resolution (Optimistic) or Validity (ZK)
- Settlement

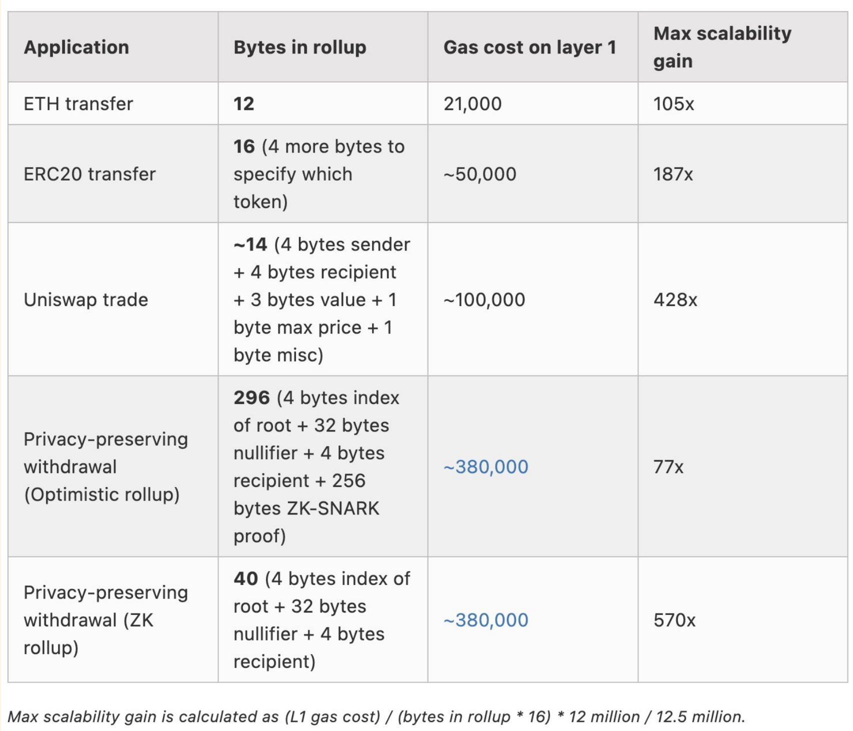
Optimistic vs ZK rollups

	ZK Rollups	**Optimistic Rollups**
Fixed gas cost	~500K (verification of ZK proof takes time)	~21K
Variable (per-txn cost)	Lower	Higher
Finality	Instant (as soon as next batch)	Challenge period time (~1 week) [Liquidity providers can help]
Technology complexity/immaturity	High	Straightforward

Optimistic vs ZK rollups (contd)

	ZK Rollups	Optimistic Rollups
Off-chain cost	High - creating a ZK proof is expensive. A block of 1K tx could take O(10-20 minutes)	Low
General-purpose?	ZK proofs of general purpose EVM computations is harder	Easier

Scalability with rollups



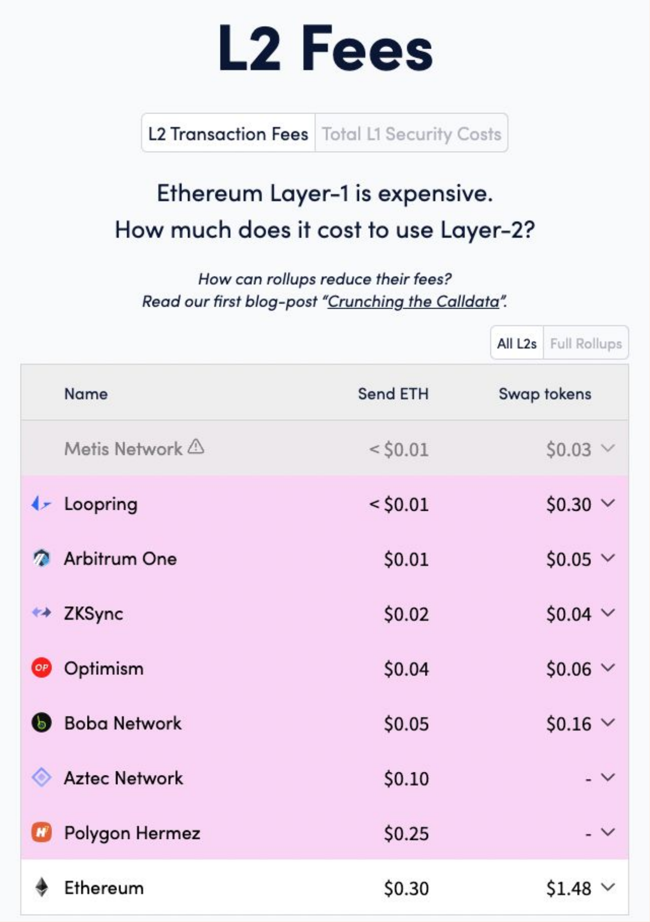
Application	Bytes in rollup	Gas cost on layer 1	Max scalability gain
ETH transfer	**12**	21,000	105x
ERC20 transfer	**16** (4 more bytes to specify which token)	~50,000	187x
Uniswap trade	**~14** (4 bytes sender + 4 bytes recipient + 3 bytes value + 1 byte max price + 1 byte misc)	~100,000	428x
Privacy-preserving withdrawal (Optimistic rollup)	**296** (4 bytes index of root + 32 bytes nullifier + 4 bytes recipient + 256 bytes ZK-SNARK proof)	~380,000	77x
Privacy-preserving withdrawal (ZK rollup)	**40** (4 bytes index of root + 32 bytes nullifier + 4 bytes recipient)	~380,000	570x

*Max scalability gain is calculated as (L1 gas cost) / (bytes in rollup * 16) * 12 million / 12.5 million.*

Current rollups can do ~2000 (Transfer) TPS

https://vitalik.ca/general/2021/01/05/rollup.html

L2FEES



L2 Fees

L2 Transaction Fees | Total L1 Security Costs

Ethereum Layer-1 is expensive.
How much does it cost to use Layer-2?

How can rollups reduce their fees?
Read our first blog-post "Crunching the Calldata".

All L2s | Full Rollups

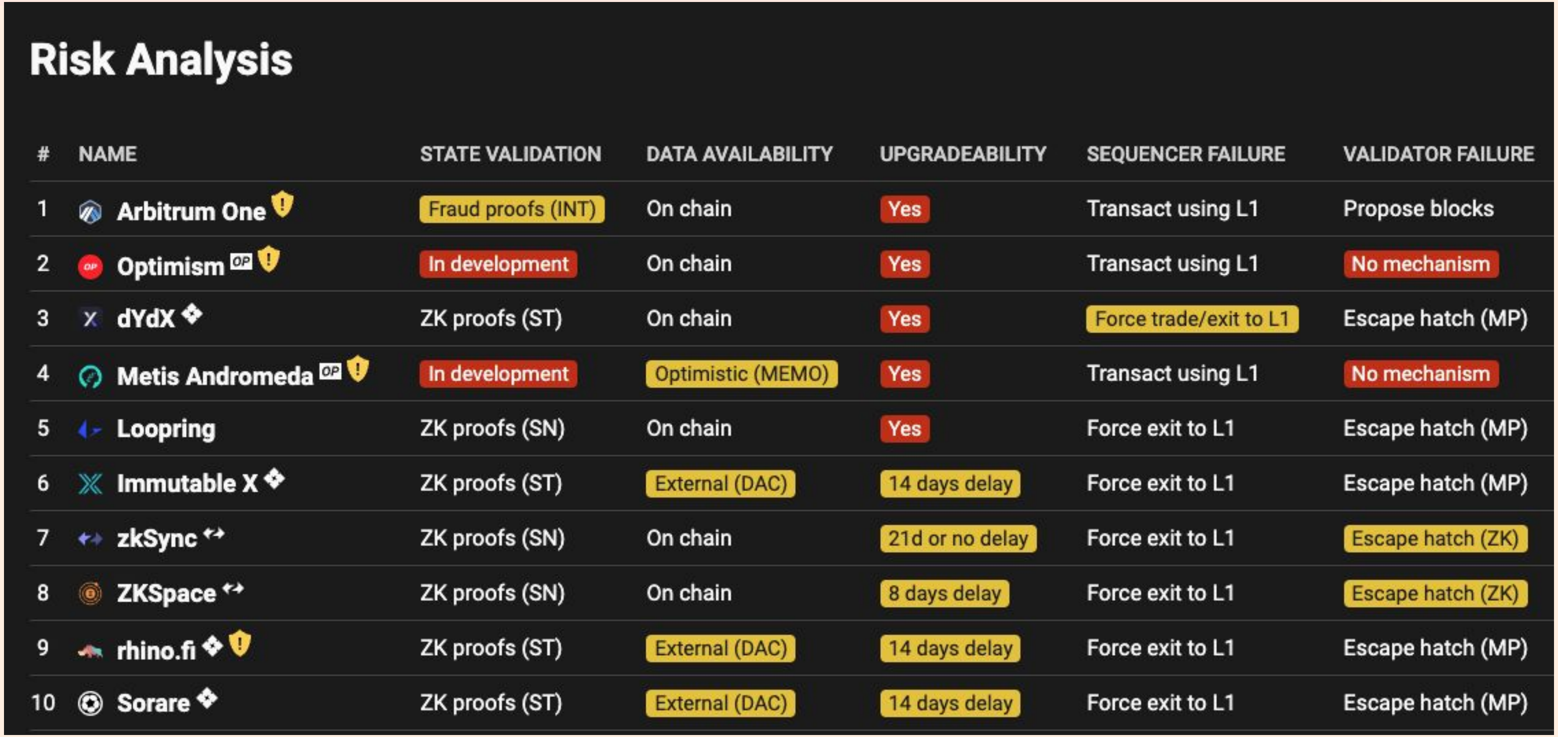
Name	Send ETH	Swap tokens
Metis Network ⚠	< $0.01	$0.03
Loopring	< $0.01	$0.30
Arbitrum One	$0.01	$0.05
ZKSync	$0.02	$0.04
Optimism	$0.04	$0.06
Boba Network	$0.05	$0.16
Aztec Network	$0.10	-
Polygon Hermez	$0.25	-
Ethereum	$0.30	$1.48

https://l2fees.info

Types of operators/batch producers

- Anyone can submit a batch
    - Potentially wasted effort on computing batches in parallel
- Centralized sequencer(s)
    - Allowlist of one or more actors
- Sequencer auction
    - Auction for who is sequencer for next day
- Proof-of-Stake (PoS)
- Delegated PoS (DPoS)

Rollups are on training wheels



Risk Analysis

#	NAME	STATE VALIDATION	DATA AVAILABILITY	UPGRADEABILITY	SEQUENCER FAILURE	VALIDATOR FAILURE
1	Arbitrum One	Fraud proofs (INT)	On chain	Yes	Transact using L1	Propose blocks
2	Optimism OP	In development	On chain	Yes	Transact using L1	No mechanism
3	dYdX	ZK proofs (ST)	On chain	Yes	Force trade/exit to L1	Escape hatch (MP)
4	Metis Andromeda OP	In development	Optimistic (MEMO)	Yes	Transact using L1	No mechanism
5	Loopring	ZK proofs (SN)	On chain	Yes	Force exit to L1	Escape hatch (MP)
6	Immutable X	ZK proofs (ST)	External (DAC)	14 days delay	Force exit to L1	Escape hatch (MP)
7	zkSync	ZK proofs (SN)	On chain	21d or no delay	Force exit to L1	Escape hatch (ZK)
8	ZKSpace	ZK proofs (SN)	On chain	8 days delay	Force exit to L1	Escape hatch (ZK)
9	rhino.fi	ZK proofs (ST)	External (DAC)	14 days delay	Force exit to L1	Escape hatch (MP)
10	Sorare	ZK proofs (ST)	External (DAC)	14 days delay	Force exit to L1	Escape hatch (MP)

Source: https://l2beat.com

Proposal to take wheels off (Nov '22)

- Stage 0: Full training wheels
    - All rollup txns go on-L1, one full node
    - Operator can not freeze or steal users' assets
    - No active fraud/validity proof
- Stage 1: Limited training wheels
    - Running fraud or validity proof scheme
    - Upgrade mechanism allowed to exist for bugs
- Stage 2: no training wheels

https://ethereum-magicians.org/t/proposed-milestones-for-rollups-taking-off-training-wheels/11571

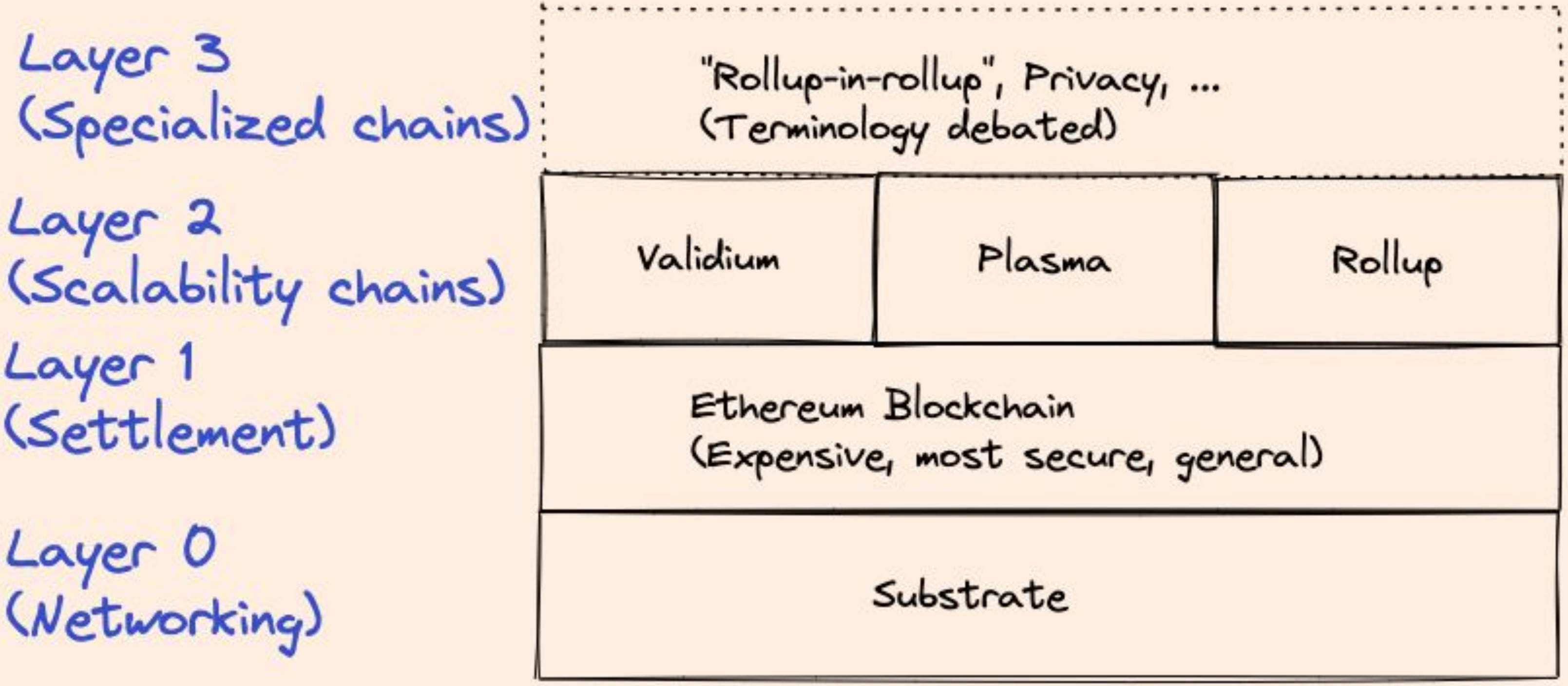
Related L2 schemes

Validium = Plasma + ZK-rollups

- Like ZK rollups, but off-chain data
- Only block header hashes are published (not individual txns) like Plasma
- This makes them cheaper+private, at the expense of less security
- Data at “Data Availability Committee”
- Many enterprise blockchain use cases can use validiums instead

https://twitter.com/EliBenSasson/status/1267453759170523136

Blockchains-within-blockchains



Layer 3
(Specialized chains)

"Rollup-in-rollup", Privacy, ...
(Terminology debated)

Layer 2
(Scalability chains)

Validium

Plasma

Rollup

Layer 1
(Settlement)

Ethereum Blockchain
(Expensive, most secure, general)

Layer 0
(Networking)

Substrate

https://medium.com/starkware/fractal-scaling-from-l2-to-l3-7fe238ecfb4f

SIDECHAINS

- “Pure” off-chain scaling protocols
- Separate blockchains independent of L1
- Connected to L1 by a 2-way bridge
- Easy dApps portability if sidechain uses same VM as L1
- Have own security model (Do not derive security from L1)
- Eg Polygon

Open problems with rollups

- Need at least one online honest node to verify and challenge published assertions
  - Are current incentives enough?
- Can zkEVMs be efficient *and* fully general?
- Exploration of design space between
  - ZK/Optimistic rollups/Validiums (Volitions)
  - Rollups with various “Data Availability modes”
- Exploration of failure modes
  - With centralized sequencers
  - DOS attacks (eg too many challenges, disputes)

Open problems (contd)

- What is the right fee structure for an individual transaction, given batching
- Low fee chains (eg Solana, Polygon) still have vastly lower fees than Rollups
    - Sharding will help: EIP-4844 (“Proto-danksharding”) introduces new, cheaper *blob-carrying transaction* type for rollups to cheaply publish arbitrary large blobs (~125KB) in the consensus layer (and blob commitments in the execution layer)
    - Coinbase is actively contributing to EIP-4844

Conclusions

- L2 Rollups seen as the way forward in the short/medium (may be even long term) for Ethereum scaling because of generality and easy dApps portability
- Current L2 schemes are still immature
    - But lots of activity in the industry / ecosystem
- Sharding is the long term way for scaling blockchain performance