Understanding L2: Sequencers, Ordering, & Execution
Overview

- What are and why are L2 Sequencers?
- What's The Current State of Things?
- How can State of Things Improve?
- Ordering vs Execution on L2 vs. L1; same challenges or nah?
Me:

Daniel Zachary Goldman / @DZack23

Engineering + Tech Research @ Offchain Labs / Arbitrum
What we want from L2s

- Trustless / L1-Level Security
- Cheap
- Fast
What Rollups Give us

- L2s mostly = Rollups
  - Nice UX; familiar to L1 users
- Rollups key trick is publishing data on L1
- Trustless ✅ Cheap ✅
- Fast?... you can’t go faster than L1 if you’re... publishing data on L1
DZack Trilemma—Pick 2:

- Trustless
- Rollups
- Open Participation
- Sidechains or Something
- Faster Finality than L1
Naive Trusted Solution Vs. Sequencer

- User picks “some random dude” to trust give us fast txs
- Random dude can’t guarantee ordering even if he’s honest
- Instead we enshrine The Sequencer: Sequencer is the only party that post transactions into L2 “directly” (i.e., without a delay)
Sequencers: 3 Phases of Ordering and Execution

SEQUENCER

Validator Asserts State Root

L1 CHAIN

L2 BLOCK

SEQUENCER FEED (SOFT GUARANTEE)

BATCH AND COMPRESS

SETTLE TO L1

TX

TX

TX

TX

TX

TX

TX

TX

TX
“Optional” Sequencer Trust

- “Optional” how?
  - Happy case: wait a bit longer for trustless finality
  - Unhappy case: alternative, fallback “slow inbox” path that circumvents the sequencer entirely
...yeah okay but what even is “the Sequencer?”

- It’s whatever entity we grant short term posting rights to / trust for fast txns
- In principle, it could use whatever mechanism we want (tho can’t interact with L1, so can’t be “truly” trustless/decentralized)
- Currently…
Currently: Sequencers Are Centralized

...This is fine?

Not as bad as you (might) think!

- Limited power, i.e.,
  - Can’t rug the system
  - Can’t lock up user’s funds
- L2s currently have more centralized training wheels ([Arbitrum docs](https://arbitrum.org/docs), [L2Beat](https://l2beat.io))
Centralized Sequencers: ...but it’s not ideal

Risks:

- Honest sequencer
  - Downtime => worse liveness
  - ZKP generation overhead (for ZKR)

- Malicious Sequencer
  - Equivocation
  - (Temporary) censorship
  - MEV!!!!
Ahhhhh MEV

- Side effect of fast txs: Sequencer (by default) has full ordering power
- Philosophical debate — Feature? bug? Somewhere in between?
- Designs for handling MEV at L2 either seek to minimize it or capture it in better ways
Cryptoeconomic Penalties

- Sequencer posts bond; equivocate and bond is slashed
- Helps mitigate equivocation (only)
- Can only punish, not rectify
- Implementation details get a bit messy r.e. L1 reorgs, but doable in principle
Threshold Encryption

- Mitigates Sequencer MEV power (only)
- Keypers: Distributed Key Generation (DKG)
  - Encrypt input data, send to Sequencer, decrypt only after Sequencer commits to ordering
- Potential increased latency / delay attacks
  - “Keypers” need to generate new keys for each round, communicate overhead with clients
  - Keypers semi-trusted (not to withhold key data etc.)

(See [Shutter network](#))
MEV Auctions

- Periodically auctions off sequencing rights over some future interval of time to highest bidder
- Incentive to be sequencer = MEV extraction
- Auctions are infrequent; bidding on predicted “future MEV”

MEV Auctions (cont.)

Potential Downsides

- Latency vs MEV power
- Temporary centralization (Liveness risk / griefing attack)
- Expect practical centralization in practice
- “Ideological” MEV questions: *should* it be captured by the underlying protocol?
Fair Ordering

- Distributed, sequencer committee
- Ordering part is enforced within consensus
- Strict improvement over status quo
- No single-point-of-liveness failure
- Low latency

Potential Downsides

- Honest threshold assumption
- Benefits sophisticated network actors…
...ordering how? (fair ordering cont.)

- “Fair ordering” still leaves open the question of ordering algo
- Simple FIFO incentivizes actors to optimize on the network level, non-ideal
- Can we do better?
Hybrid Ordering Policy: FBA-FCFS.
(fair ordering cont.)

- Separate inputs into discrete time intervals ("fairness granularity")
- Fair ordering / FIFO of intervals, priority fee within intervals
- Active area of research and inquiry!
  - https://research.arbitrum.io/t/hybrid-transaction-ordering-policy/155/1
  - https://research.arbitrum.io/t/transaction-ordering-policy/127/2

TLDR
We argue that frequent batch auction-style FCFS should be adopted in order to make the fairness notion more robust and welfare-maximizing (in sense of providing better UX and making the network long-term incentive aligned with the correct parties).
L1 Status Quo: Ordering 🤝 Execution

Block header

- parentHash
- nonce
- timestamp
- coinbase
- beneficiary
- logsBloom
- difficulty
- extraData
- number
- gasLimit
- gasUsed
- mixHash
- stateRoot
- transactionsRoot
- receiptsRoot
Separating Ordering & Execution on L1

- Different motivation than in L2 world; not interested in faster finality
- Separating transaction ordering => democratizing MEV
- Less economy of scale / pull towards staker centralization
Network Level Ordering / Execution Separation

- MEV-boost!
- De facto separation of tx ordering (builders) and block proposers
- Per-block MEV auctions (sort of)
- Separation of concerns = good for decentralization
- *Not* logically enshrined in consensus (..yet?)
In-protocol Ordering/Execution Separation PBS

- **Proposer builder separation!**
- MEV boost - style, but consensus protects builders/proposers from each other via fancy fork choice rule
- Open research questions remain
More In-protocol Ideas For L1

- Censorship resistant backup path for centralized block builder?
- Threshold commit/reveal for L1?
- ...Fair ordering?
- ...ZK proofs in L1 consensus?
Fin:

- Sequencers give us fast transactions = cool
- Centralized sequencers not terrible but not ideal, trust-minimizing sequencers = cooler
- L1 r&d 🥰 L2 r&d
Thank you!

Daniel z Goldman
Dev/Techie Offchain Labs

@DZack23
orphanned slides:
Alt: Zk-Rollups (usually): Two phases

SEQUENCER

BATCH TXs, ASSERT AND PROVE STATE ROOT

L1 CHAIN

SEQUENCER FEED (SOFT GUARANTEE)
POS Sequencer (remove?)

- Decentralized, but doesn’t
Techniques not mutually exclusive