Ideal Properties of Rollup Escape Hatches

Jan Gorzny, Ph.D.
Head of L2 Scaling, Quantstamp

Lin Po-An
Research Engineer, Quantstamp

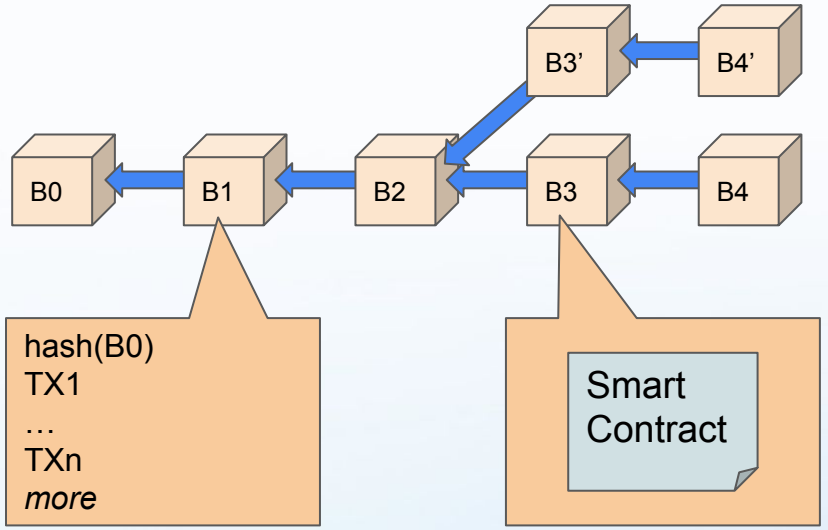
Martin Derka, Ph.D.
Head of New Initiatives, Quantstamp

8 Nov 2022
DICG 2022
Quebec, QC, Canada

Blockchains, Smart Contracts, and Rollups

Blockchains & Smart Contracts

- Append-only distributed ledger
- Users interact via transactions
  - *Blocks* record which transactions are included/processed
- Blocks are determined by some consensus algorithm (e.g., Proof-of-Work)
- Transactions can invoke *smart contracts*

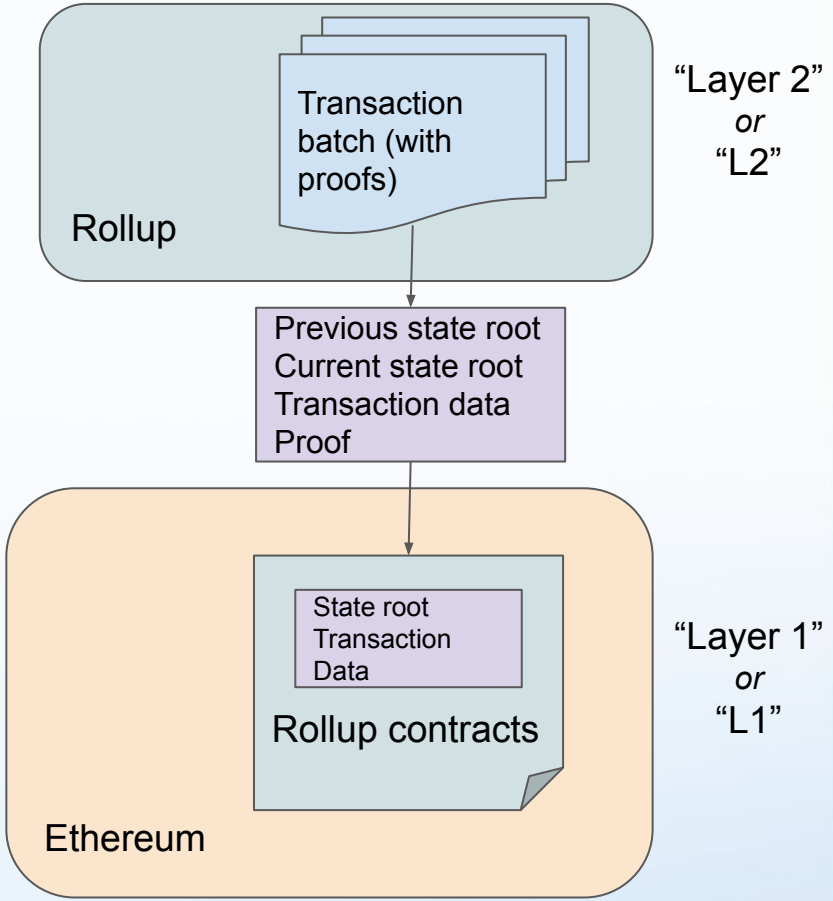


Rollups

a.k.a. "commit-chains" and "validating bridges"

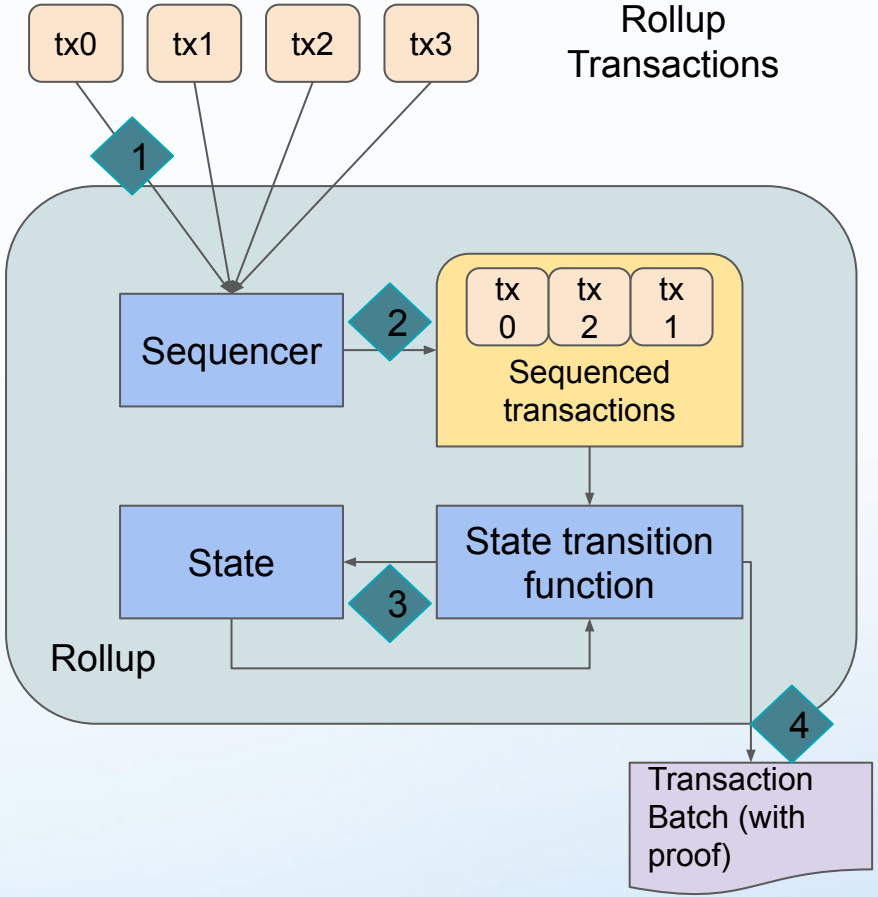
"Layer 2(+)" scaling solution.

- Compute state transitions off-chain
- Aggregate state updates
- Publish (compressed) data to underlying layer for efficiency, verifiability
- Security is tied to underlying layer 1



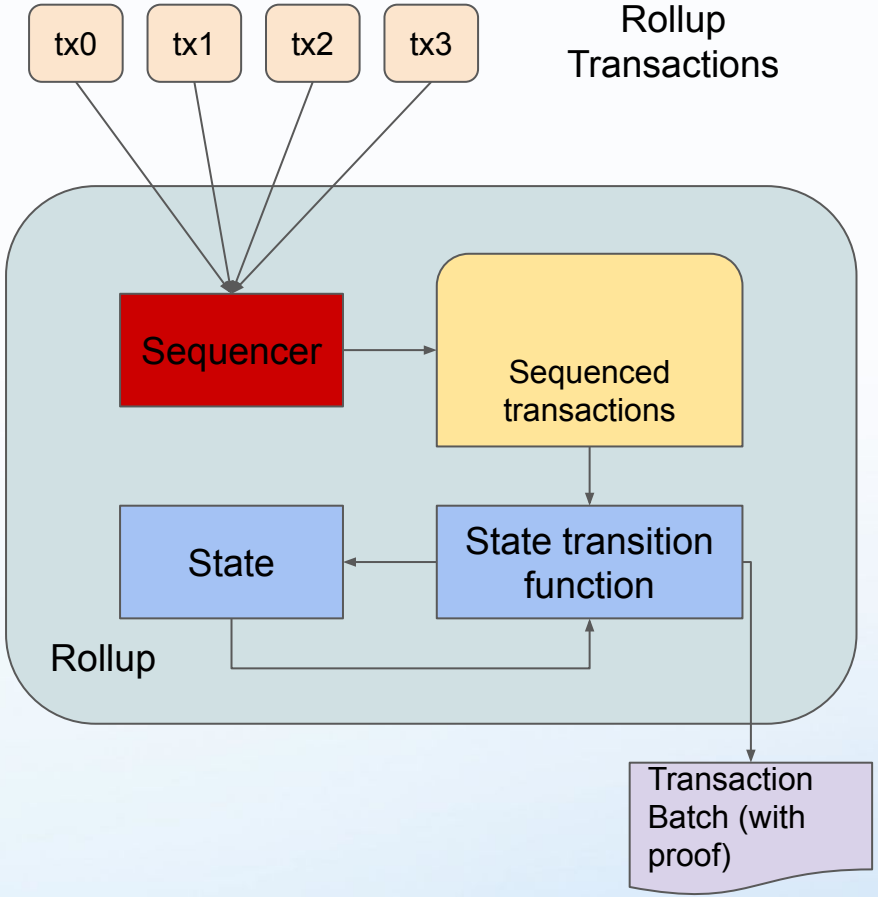
Rollup Behaviour

1. Transactions (tx*i*) are sent to the network
2. A “sequencer” provides an ordering to the transactions and may form blocks
3. The layer 2 state is updated for each transaction
4. The batch (possibly compressed, possibly complemented by a proof) is sent to the layer 1 for records



Escape Hatches

An **escape hatch** is a method by which users of a rollup can recover digital assets or program state from a rollup when the operators (sequencers) are **offline**.



Some Current Approaches

Current Approaches

Transact Using L1.
Execute state updates on L1 directly

Propose Blocks (ZK)
Advance chain using new “sequencer”

Force Exit to L1
Execute specific state updates on L1

Rollup	Escape Hatch Mechanism
Arbitrum Nova [23]/One [24]	Transact Using L1
Aztec [25] (Connect [26])	Propose Blocks* (ZK)
Boba Network [11]	Transact Using L1
dYdX [27]	Force Exit to L1
Fuel (v1) [28]	Propose Blocks
ImmutableX [29]	Force Exit to L1
Layer2.Finance [30]	*None*
Layer2.Finance-zk [31]	Force Exit to L1
Loopring [32]	Force Exit to L1
Metis Andromeda [33]	Transact Using L1
Optimism [15]	Transact Using L1
Polygon Hermez [34]	Force Exit to L1*
rhino.fi [35]	Force Exit to L1
Sorare [36]	Force Exit to L1
StarkNet [12]	*None*
ZKSpace (ZKSwap) [37]	Force Exit to L1
zkSync (v1) [21]	Force Exit to L1

Table 1: Escape hatches for various layer two solutions according to L2Beat.com [38] as of August 2022. We do not distinguish between so-called *Validium* solutions and ZK rollups, as they are similar except that the former is not required to store data on-chain along with their validity proofs.

Ideal Properties

Basic Properties

Modular. They should clearly delineate features and functionality.

Secure. They should not be vulnerable to exploitation; they may have large attack surfaces.

Correcting. Users shouldn’t need to use consecutive escape hatches.

Desired Property: Support Arbitrary State Escape

What is valuable state may not be clear.

Anything should be escapable; it may be necessary to allow users/developers to determine what is valuable.

- ERC20 tokens?

- NFTs/ERC 721 tokens?

- LP ERC 721 tokens??
- Game data?
- Other rollup state?

Desired Property: Built-In

dApp developers should need minimal extra work to be supported by an escape hatch, if their state is to be escaped.

Assets that work on L1 and L2 should have roughly the same code on both layers.

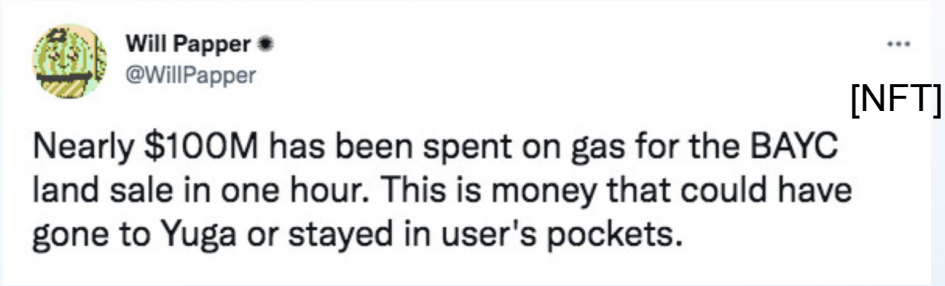
This is important for…

- Security (audits, tests, assumptions)
- Developers (ease of adoption)

Desired Property: (Transaction) Efficient

A gas war on the underlying layer should not clog the escape efforts of the L2 users.

An escape effort from an L2 should not cause a gas war on the underlying layer.



[NFT]

Clever state migration may be helpful or necessary.

Desired Property: Global

Escape hatches shouldn't be application-specific, for the UX.

It's easier to hit "escape" on a network, than every dApp a user is involved in on that network.

Recall…

Assets that work on L1 and L2 should have roughly the same code on both layers.

Desired Property: Automatic & Live

They should always be available when needed, and they should not need manual intervention to “turn on”.

Ideally, escape hatches are activated programmatically.

- Ensures that no users need to wait for (lengthy) social consensus
- Ensure that users know what to do & how to use hatch if necessary
- Can make security assumptions clear from the start

Conclusion

Rollup escape hatches may be hard. The time to start thinking about them is **now**.

Future Work

- Differentiate validator failure from sequencer failures
- Study specific approaches and establish a framework for escape hatches

Thank you!
Questions? Comments?

Email: **jan@quantstamp.com**
Twitter: **@jgorzny**

We're hiring!