title: Validium description: An introduction to Validium as a scaling solution currently utilized by the Ethereum community. lang: en sidebarDepth: 3

Validium is a scaling solution that enforces integrity of transactions using validity proofs likeZK-rollups, but doesn't store transaction data on the Ethereum Mainnet. While off-chain data availability introduces trade-offs, it can lead to massive improvements in scalability (validiums can process ~9,000 transactions, or more, per second).

Prerequisites {#prerequisites}

You should have read and understood our page onEthereum scaling and layer 2.

What is validium? {#what-is-validium}

Validiums are scaling solutions that use off-chain data availability and computation designed to improve throughput by processing transactions off the Ethereum Mainnet. Like zero-knowledge rollups (ZK-rollups), validiums publish zero-knowledge proofs to verify off-chain transactions on Ethereum. This prevents invalid state transitions and enhances the security guarantees of a validium chain.

These "validity proofs" can come in the form of ZK-SNARKs (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) or ZK-STARKs (Zero-Knowledge Scalable Transparent ARgument of Knowledge). More on zero-knowledge proofs.

Funds belonging to validium users are controlled by a smart contract on Ethereum. Validiums offer near-instant withdrawals, much like ZK-rollups do; once the validity proof for a withdrawal request has been verified on Mainnet, users can withdraw funds by providing Merkle proofs. The Merkle proof validates the inclusion of the user's withdrawal transaction in a verified transaction batch, allowing the on-chain contract to process the withdrawal.

However, validium users can have their funds frozen and withdrawals restricted. This can happen if data availability managers on the validium chain withhold off-chain state data from users. Without access to transaction data, users cannot compute the Merkle proof required to prove ownership of funds and execute withdrawals.

This is the major difference between validiums and ZK-rollups—their positions on the data availability spectrum. Both solutions approach data storage differently, which has implications for security and trustlessness.

How do validiums interact with Ethereum? {#how-do-validiums-interact-with-ethereum}

Validiums are scaling protocols built on top of the existing Ethereum chain. Although it executes transactions off-chain, a validium chain is administered by a collection of smart contracts deployed on Mainnet including:

1. **Verifier contract**: The verifier contract verifies the validity of proofs submitted by the validium operator when making state updates. This includes validity proofs attesting to the correctness of off-chain transactions and data availability proofs verifying the existence of off-chain transaction data.

2. **Main contract**: The main contract stores state commitments (Merkle roots) submitted by block producers and updates the validium's state once a validity proof is verified on-chain. This contract also processes deposits to and withdrawals from the validium chain.

Validiums also rely on the main Ethereum chain for the following:

Settlement {#settlement}

Transactions executed on a validium cannot be fully confirmed until the parent chain verifies their validity. All business conducted on a validium must eventually be settled on Mainnet. The Ethereum blockchain also provides "settlement guarantees" for validium users, meaning off-chain transactions cannot be reversed or altered once committed to on-chain.

Security {#security}

Ethereum, acting as a settlement layer, also guarantees the validity of state transitions on validium. Off-chain transactions executed on the validium chain are verified via a smart contract on the base Ethereum layer.

If the on-chain verifier contract deems the proof invalid, the transactions are rejected. This means operators must satisfy validity conditions enforced by the Ethereum protocol before updating the validium's state.

How does validium work? {#how-does-validium-work}

Transactions {#transactions}

Users submit transactions to the operator, a node responsible for executing transactions on the validium chain. Some validiums may use a sole operator to execute the chain or rely on a proof-of-stake (PoS) mechanism for rotating operators.

The operator aggregates transactions into a batch and sends it to a proving circuit for proving. The proving circuit accepts the transaction batch (and other relevant data) as inputs and outputs a validity proof verifying that the operations were performed correctly.

State commitments {#state-commitments}

The state of the validium is hashed as a Merkle tree with the root stored in the main contract on Ethereum. The Merkle root, also known as the state root, acts as a cryptographic commitment to the current state of accounts and balances on the validium.

To perform a state update, the operator must compute a new state root (after executing transactions) and submit it to the on-chain contract. If the validity proof checks out, the proposed state is accepted and the validium switches to the new state root.

Deposits and withdrawals {#deposits-and-withdrawals}

Users move funds from Ethereum to a validium by depositing ETH (or any ERC-compatible token) in the on-chain contract. The contract relays the deposit event to the validium off-chain, where the user's address is credited with an amount equal to their deposit. The operator also includes this deposit transaction in a new batch.

To move funds back to Mainnet, a validium user initiates a withdrawal transaction and submits it to the operator who validates the withdrawal request and includes it in a batch. The user's assets on the validium chain are also destroyed before they can exit the system. Once the validity proof associated with the batch is verified, the user can call the main contract to withdraw the remainder of their initial deposit.

As an anti-censorship mechanism, the validium protocol allows users to withdraw directly from the validium contract without going through the operator. In this case, users need to provide a Merkle proof to the verifier contract showing an account's inclusion in the state root. If the proof is accepted, the user can call the main contract's withdrawal function to exit their funds from the validium.

Batch submission {#batch-submission}

After executing a batch of transactions, the operator submits the associated validity proof to the verifier contract and proposes a new state root to the main contract. If the proof is valid, the main contract updates the validium's state and finalizes the results of transactions in the batch.

Unlike a ZK-rollup, block producers on a validium are not required to publish transaction data for transaction batches (only block headers). This makes validium a purely off-chain scaling protocol, as opposed to "hybrid" scaling protocols (i.e., layer 2) that publish state data on the main Ethereum chain as `calldata`.

Data availability {#data-availability}

As mentioned, validiums utilize an off-chain data availability model, where operators store all transaction data off Ethereum Mainnet. Validium's low on-chain data footprint improves scalability (throughput isn't limited by Ethereum's data processing capacity) and reduces user fees (the cost of publishing `calldata` is lower).

Off-chain data availability, however, presents a problem: data necessary for creating or verifying Merkle proofs may be unavailable. This means users may be unable to withdraw funds from the on-chain contract if operators should act maliciously.

Various validium solutions attempt to solve this problem by decentralizing the storage of state data. This involves forcing block producers to send the underlying data to "data availability managers" responsible for storing off-chain data and making it available to users on request.

Data availability managers in validium attest to the availability of data for off-chain transactions by signing every validium batch. These signatures constitute a form of "availability proof" which the on-chain verifier contract checks before approving state updates.

Validiums differ in their approach to data availability management. Some rely on trusted parties to store state data, while others use randomly assigned validators for the task.

Data availability committee (DAC) {#data-availability-committee}

To guarantee the availability of off-chain data, some validium solutions appoint a group of trusted entities, collectively known as a data availability committee (DAC), to store copies of the state and provide proof of data availability. DACs are easier to implement and require less coordination since membership is low.

However, users must trust the DAC to make the data available when needed (e.g., for generating Merkle proofs). There's the possibility of members of data availability committees getting compromised by a malicious actor who can then withhold off-chain data.

More on data availability committees in validiums.

Bonded data availability {#bonded-data-availability}

Other validiums require participants charged with storing offline data to stake (i.e., lock up) tokens in a smart contract before assuming their roles. This stake serves as a “bond” to guarantee honest behavior among data availability managers and reduces trust assumptions. If these participants fail to prove data availability, the bond is slashed.

In a bonded data availability scheme, anyone can be assigned to hold off-chain data once they provide the required stake. This expands the pool of eligible data availability managers, reducing the centralization that affects data availability committees (DACs). More importantly, this approach relies on cryptoeconomic incentives to prevent malicious activity, which is considerably more secure than appointing trusted parties to secure offline data in the validium.

More on bonded data availability in validiums.

Volitions and validium {#volitions-and-validium}

Validiums offer many benefits but come with trade-offs (most notably, data availability). But, as with many scaling solutions, validiums are suited to specific use-cases—which is why volitions were created.

Volitions combine a ZK-rollup and validium chain and allow users to switch between the two scaling solutions. With volitions, users can take advantage of validium's off-chain data availability for certain transactions, while retaining the freedom to switch to an on-chain data availability solution (ZK-rollup) if needed. This essentially gives users the freedom to choose trade-offs as dictated by their unique circumstances.

A decentralized exchange (DEX) may prefer using a validium’s scalable and private infrastructure for high-value trades. It can also use a ZK-rollup for users who want a ZK-rollup's higher security guarantees and trustlessness.

Validiums and EVM compatibility {#validiums-and-evm-compatibility}

Like ZK-rollups, validiums are mostly suited to simple applications, such as token swaps and payments. Supporting general computation and smart contract execution among validiums is difficult to implement, given the considerable overhead of proving EVM instructions in a zero-knowledge proof circuit.

Some validium projects attempt to sidestep this problem by compiling EVM-compatible languages (e.g., Solidity, Vyper) into creating custom bytecode optimized for efficient proving. A drawback of this approach is that new zero-knowledge proof-friendly VMs may not support important EVM opcodes, and developers have to write directly in the high-level language for an optimal experience. This creates even more problems: it forces developers to build dapps with an entirely new development stack and breaks compatibility with current Ethereum infrastructure.

Some teams, however, are attempting to optimize existing EVM opcodes for ZK-proving circuits. This will result in the development of a zero-knowledge Ethereum Virtual Machine (zkEVM), an EVM-compatible VM that produces proofs to verify the correctness of program execution. With a zkEVM, validium chains can execute smart contracts off-chain and submit validity proofs to verify an off-chain computation (without having to re-execute it) on Ethereum.

More on zkEVMs.

How do validiums scale Ethereum? {#scaling-ethereum-with-validiums}

1. Off-chain data storage {#off-chain-data-storage}

Layer 2 scaling projects, such as optimistic rollups and ZK-rollups, trade the infinite scalability of pure off-chain scaling protocols (e.g., Plasma) for security by publishing some transaction data on L1. But this means the scalability properties of rollups is limited by data bandwidth on Ethereum Mainnet (data sharding proposes to improve Ethereum's data storage capacity for this reason).

Validiums achieve scalability by keeping all transaction data off-chain and only post state commitments (and validity proofs) when relaying state updates to the main Ethereum chain. The existence of validity proofs, however, gives validiums higher security guarantees than other pure off-chain scaling solutions, including Plasma and sidechains. By reducing the amount of data Ethereum has to process before validating off-chain transactions, validium designs greatly extend throughput on Mainnet.

2. Recursive proofs {#recursive-proofs}

A recursive proof is a validity proof that verifies the validity of other proofs. These "proof of proofs" are generated by recursively aggregating multiple proofs until one final proof verifying all previous proofs is created. Recursive proofs scale blockchain processing speeds by increasing the number of transactions that can be verified per validity proof.

Typically, each validity proof the validium operator submits to Ethereum for verification validates the integrity of a single block. Whereas a single recursive proof can be used to confirm the validity of several validium blocks at the same time—this is possible since the proving circuit can recursively aggregate several block proofs into one final proof. If the on-chain verifier contract accepts the recursive proof, all the underlying blocks are finalized immediately.

Pros and cons of validium {#pros-and-cons-of-validium}

Pros	Cons
Validity proofs enforce integrity of off-chain transactions and prevent operators from finalizing invalid state updates.	Producing validity proofs requires special hardware, which poses a centralization risk.
Increases capital efficiency for users (no delays in withdrawing funds back to Ethereum)	Limited support for general computation/smart contracts; specialized languages required for development.
Not vulnerable to certain economic attacks faced by fraud-proof based systems in high-value applications.	High computational power required to generate ZK proofs; not cost-effective for low throughput applications.
Reduces gas fees for users by not posting calldata to Ethereum Mainnet.	Slower subjective finality time (10-30 min to generate a ZK proof) but faster to full finality because there is no dispute time delay.
Suitable for specific use-cases, like trading or blockchain gaming that prioritize transaction privacy and scalability.	Users can be prevented from withdrawing funds since generating Merkle proofs of ownership requires off-chain data to be available at all times.
Off-chain data availability provides higher levels of throughput and increases scalability.	Security model relies on trust assumptions and cryptoeconomic incentives, unlike ZK-rollups, which purely rely on cryptographic security mechanisms.

Use Validium/Volitions {#use-validium-and-volitions}

Multiple projects provide implementations of Validium and volitions that you can integrate into your dapps:

StarkWare StarkEx - *StarkEx is an Ethereum Layer 2 (L2) scalability solution that is based on validity proofs. It can operate in either ZK-Rollup or Validium data-availability modes.*

- Documentation
- Website

Matter Labs zkPorter- *zkPorter is a Layer 2 scaling protocol tackling data availability with a hybrid approach that combines the ideas of zkRollup and sharding. It can support arbitrarily many shards, each with its own data availability policy.*

- Documentation
- Website

Further reading {#further-reading}

- Validium And The Layer 2 Two-By-Two — Issue No. 99
- ZK-rollups vs Validium
- Volition and the Emerging Data Availability spectrum
- Rollups, Validiums, and Volitions: Learn About the Hottest Ethereum Scaling Solutions