TL;DR:

we propose Adamantium, a protocol for Autonomous Data Availability, which retains the scaling benefits of off-chain data availability, while removing all trust assumptions for any willing user. Willing to do what? To be online; and if they aren't online, their funds cannot be stolen, nor frozen - rather, the funds are moved from L2 back to an Ethereum address under the user's control.

Background

Validium relies on a DA Committee (DAC), made up of a set of reputable players in the blockchain space. DAC members store off-chain a copy of the account balances, and attest to the availability of its state S by signing the Merkle root of S after every batch processed by the StarkEx Operators.

Validium's trust assumptions: Validium requires users to trust DAC members in one very particular scenario, which we call the Escape Hatch. In case the StarkEx Operators censor a user's withdrawal request, users trust at least one DAC member to publish a current copy of the latest state S (read a complete description of the protocol here). Can Validium be improved and made completely trustless? It can, and we call the improved protocol Adamantium.

[

Adamantium Description

1400×1375 118 KB

](https://ethresear.ch/uploads/default/original/2X/0/04df68a90f0b25c1ca14e3bdc97d9e525cd85e78.png)

Description

In Adamantium, users can operate in a fully trustless manner, by choosing to become a Power User (PU). The funds of a PU are always in her custody: typically a PU provides a signature that she has access to her own off-chain data, thus allowing her to activate her personal Escape Hatch with the Application Smart Contract on L1. Absent that timely signature, the PU's funds are automatically withdrawn back on-chain (aka Protective Withdrawal).

What about users who do not wish to become a PU? With Adamantium, they have a wider set of choices. They will no longer be restricted to trusting a DAC member - they can opt to trust any Power User willing to serve as a watchtower on their behalf (and would have to authorize that PU to do so).

Participants:

- DAC: The DAC continues to operate, and offer its services to any interested users (i.e. app users)
- Users:
- Regular Users: Users can continue to operate as they did previously, and rely on the DAC to fill its role, as described above.
- Power User (PU): A user who trusts no one - not the DAC nor anyone else.

- Regular Users: Users can continue to operate as they did previously, and rely on the DAC to fill its role, as described above.
- Power User (PU): A user who trusts no one - not the DAC nor anyone else.

System Design Implications & Economics:

- PU (Power User):
- A PU has one or more Merkle tree vaults mapped to it - these are the vaults she signs for
- A PU is generally expected to be online:
- Response time: a PU needs to provide her signature within a proof-generation time frame, so her response time is measured in minutes, not seconds.
- Cost: they have enough at stake, and care enough not to trust other parties, to warrant the hassle and expense.
- When on-line: PU performs the same computational work as a DAC member: they need to hold the balance tree and verify the Merkle tree up to the root.

We estimate the PU's monthly computational cost to be a few $100s/month.

- When going off-line: a Protective Withdrawal is executed.

Protective Withdrawal - the protocol-enforced withdrawal of funds back on-chain as call data - is the key innovation in Adamantium. Absent a timely cryptographic signature from the PU, the Operator is forced to make the funds available to the PU on mainnet.

In a given proof batch cycle, the Operator pays for call data only for those users who went off-line during that cycle. Importantly, this gas expense does not scale with the number of transactions in a given batch, nor with all users who are merely still off-line. Naturally, the Operator may charge the PU for this sequence.

- When on-line: PU performs the same computational work as a DAC member: they need to hold the balance tree and verify the Merkle tree up to the root.

We estimate the PU’s monthly computational cost to be a few $100s/month.

- When going off-line: a Protective Withdrawal is executed.

Protective Withdrawal - the protocol-enforced withdrawal of funds back on-chain as call data - is the key innovation in Adamantium. Absent a timely cryptographic signature from the PU, the Operator is forced to make the funds available to the PU on mainnet.

In a given proof batch cycle, the Operator pays for call data only for those users who went off-line during that cycle. Importantly, this gas expense does not scale with the number of transactions in a given batch, nor with all users who are merely still off-line. Naturally, the Operator may charge the PU for this sequence.

- Response time: a PU needs to provide her signature within a proof-generation time frame, so her response time is measured in minutes, not seconds.

- Cost: they have enough at stake, and care enough not to trust other parties, to warrant the hassle and expense.

- When on-line: PU performs the same computational work as a DAC member: they need to hold the balance tree and verify the Merkle tree up to the root.

We estimate the PU’s monthly computational cost to be a few $100s/month.

- When going off-line: a Protective Withdrawal is executed.

Protective Withdrawal - the protocol-enforced withdrawal of funds back on-chain as call data - is the key innovation in Adamantium. Absent a timely cryptographic signature from the PU, the Operator is forced to make the funds available to the PU on mainnet.

In a given proof batch cycle, the Operator pays for call data only for those users who went off-line during that cycle. Importantly, this gas expense does not scale with the number of transactions in a given batch, nor with all users who are merely still off-line. Naturally, the Operator may charge the PU for this sequence.

- When on-line: PU performs the same computational work as a DAC member: they need to hold the balance tree and verify the Merkle tree up to the root.

We estimate the PU’s monthly computational cost to be a few $100s/month.

- When going off-line: a Protective Withdrawal is executed.

Protective Withdrawal - the protocol-enforced withdrawal of funds back on-chain as call data - is the key innovation in Adamantium. Absent a timely cryptographic signature from the PU, the Operator is forced to make the funds available to the PU on mainnet.

In a given proof batch cycle, the Operator pays for call data only for those users who went off-line during that cycle. Importantly, this gas expense does not scale with the number of transactions in a given batch, nor with all users who are merely still off-line. Naturally, the Operator may charge the PU for this sequence.

- A PU has one or more Merkle tree vaults mapped to it - these are the vaults she signs for

- A PU is generally expected to be online:

- Response time: a PU needs to provide her signature within a proof-generation time frame, so her response time is measured in minutes, not seconds.

- Cost: they have enough at stake, and care enough not to trust other parties, to warrant the hassle and expense.

- When on-line: PU performs the same computational work as a DAC member: they need to hold the balance tree and verify the Merkle tree up to the root.

We estimate the PU’s monthly computational cost to be a few $100s/month.

- When going off-line: a Protective Withdrawal is executed.

Protective Withdrawal - the protocol-enforced withdrawal of funds back on-chain as call data - is the key innovation in Adamantium. Absent a timely cryptographic signature from the PU, the Operator is forced to make the funds available to the PU on mainnet.

In a given proof batch cycle, the Operator pays for call data only for those users who went off-line during that cycle. Importantly, this gas expense does not scale with the number of transactions in a given batch, nor with all users who are merely still off-line. Naturally, the Operator may charge the PU for this sequence.

- When on-line: PU performs the same computational work as a DAC member: they need to hold the balance tree and verify the Merkle tree up to the root.

We estimate the PU's monthly computational cost to be a few $100s/month.

- When going off-line: a Protective Withdrawal is executed.

Protective Withdrawal - the protocol-enforced withdrawal of funds back on-chain as call data - is the key innovation in Adamantium. Absent a timely cryptographic signature from the PU, the Operator is forced to make the funds available to the PU on mainnet.

In a given proof batch cycle, the Operator pays for call data only for those users who went off-line during that cycle. Importantly, this gas expense does not scale with the number of transactions in a given batch, nor with all users who are merely still off-line. Naturally, the Operator may charge the PU for this sequence.

- Response time: a PU needs to provide her signature within a proof-generation time frame, so her response time is measured in minutes, not seconds.
- Cost: they have enough at stake, and care enough not to trust other parties, to warrant the hassle and expense.

- When on-line: PU performs the same computational work as a DAC member: they need to hold the balance tree and verify the Merkle tree up to the root.

We estimate the PU's monthly computational cost to be a few $100s/month.

- When going off-line: a Protective Withdrawal is executed.

Protective Withdrawal - the protocol-enforced withdrawal of funds back on-chain as call data - is the key innovation in Adamantium. Absent a timely cryptographic signature from the PU, the Operator is forced to make the funds available to the PU on mainnet.

In a given proof batch cycle, the Operator pays for call data only for those users who went off-line during that cycle. Importantly, this gas expense does not scale with the number of transactions in a given batch, nor with all users who are merely still off-line. Naturally, the Operator may charge the PU for this sequence.

- When on-line: PU performs the same computational work as a DAC member: they need to hold the balance tree and verify the Merkle tree up to the root.

We estimate the PU's monthly computational cost to be a few $100s/month.

- When going off-line: a Protective Withdrawal is executed.

Protective Withdrawal - the protocol-enforced withdrawal of funds back on-chain as call data - is the key innovation in Adamantium. Absent a timely cryptographic signature from the PU, the Operator is forced to make the funds available to the PU on mainnet.

In a given proof batch cycle, the Operator pays for call data only for those users who went off-line during that cycle. Importantly, this gas expense does not scale with the number of transactions in a given batch, nor with all users who are merely still off-line. Naturally, the Operator may charge the PU for this sequence.

- Application Operator (e.g. the exchange):
- Adamantium Vaults: The Application Smart Contract tracks the vaults mapped to every PU.
- Protective Withdrawals Cost: Tx batching reduces the gas cost of a Protective Withdrawal so the amortized cost approaches the gas cost of placing the data on-chain.
- An Operator can ignore a PU's signature, thus triggering an unwarranted Protective Withdrawal. Repeating this kind of ordeal too many times will simply cause PUs to switch to a competing application.

- Adamantium Vaults: The Application Smart Contract tracks the vaults mapped to every PU.

- Protective Withdrawals Cost: Tx batching reduces the gas cost of a Protective Withdrawal so the amortized cost approaches the gas cost of placing the data on-chain.
- An Operator can ignore a PU's signature, thus triggering an unwarranted Protective Withdrawal. Repeating this kind of ordeal too many times will simply cause PUs to switch to a competing application.

[

Screen Shot 2021-05-31 at 15.03.25

1402×582 32.4 KB

](https://ethresear.ch/uploads/default/original/2X/f/f9621ee8f86ec2a2d793a5b568d872c806d296e6.png)

- Other than running an Ethereum node

** Till you withdraw

Protocol Extension: Followers of a PU

A Follower is a user who chooses to put their trust in one or more PUs that provide them with a watchtower service for a fee. A PU should sign not only for their own vaults, but also for the vaults of their Followers.

A Follower's funds will be withdrawn back on-chain only if none of the PUs it follows have provided their timely cryptographic signature. By following multiple PUs, a Follower reduces the likelihood that a chance disruption of service by any single PU will result in a withdrawal of funds.

A PU could ignore a Follower's signature, thus triggering an unnecessary Protective Withdrawal. We believe this kind of behavior will be very limited in scope, as Followers will simply switch to more reliable PUs.