CRYPTO PRAGMATIST PRO

All views are solely my opinions. This is written exclusively for informational purposes. It is not an inducement to invest nor is it advice to follow any particular investment strategy. Data points are taken from various online sources that may or may not be accurate as of publication.

Optimistic vs. ZK Rollups

We are currently in the midst of an unsurprising Arbitrum Season, where many Arbitrum native projects are highly outperforming the entire crypto space. TVL on the rollup has been up and to the right the last month, creating a DeFi Llama dashboard resemblant of the bull run:

Name	Category	Chains	1d Change	7d Change	1m Change
1 GMX	Derivatives		+0.17%	+6.35%	+11.39%
2 Curve	Dexes	+7	+0.83%	+4.31%	+60.74%
3 Uniswap V3	Dexes		-1.77%	-5.76%	+24.46%
4 Sushi		+16	-2.12%	+4.66%	+29.15%
5 Stargate	Cross Chain	+2	-2.55%	-0.07%	-23.49%
6 Synapse	Cross Chain	+11	-2.77%	-1.70%	+23.24%
7 AAVE V3	Lending		-1.55%	+12.20%	+22.74%
8 SwapFish	Dexes		+11.92%	+59.75%	
9 Radiant	Lending		-3.15%	+3.40%	+0.51%
10 Vesta Finance	CDP		-0.30%	+9.18%	+54.80%
11 Beefy	Yield Aggregator	+10	-0.69%	+11.75%	+38.89%
12 Dopex	Options		-6.25%	+3.42%	+32.49%

https://defillama.com/chain/Arbitrum

Dopex, JonesDAO, TreasureDAO, and GMX prices have all increased by more than 2x since the FTX collapse, only proving Arbitrum's relative strength in a time of weakness. While other blockchain ecosystems have been slowly bleeding out their value and project specific tokens head to zero, what makes Arbitrum different?

Despite Arbitrum (and Optimism) feeling like they have been around for a while, they are really just now beginning to see an inflection point of adoption. Arbitrum recently launched their Nitro upgrade which increased compatibility and reduced transaction costs, bringing more protocols and users to the network. And finally, GMX (Arbitrum native) is catching all positive momentum with the crash of FTX and distrust in any centralized crypto entity.

With all these positives regarding optimistic rollups, why should we consider the alternative solution in ZK's? When looking at differences between optimistic and zero-knowledge rollups, there are some clear tradeoffs that can make ZK superior in some aspects, and vice versa. Rather than explain everything in long form here, we will dive into the details of ZK below and save a summarized comparison to optimistic rollups for the end.

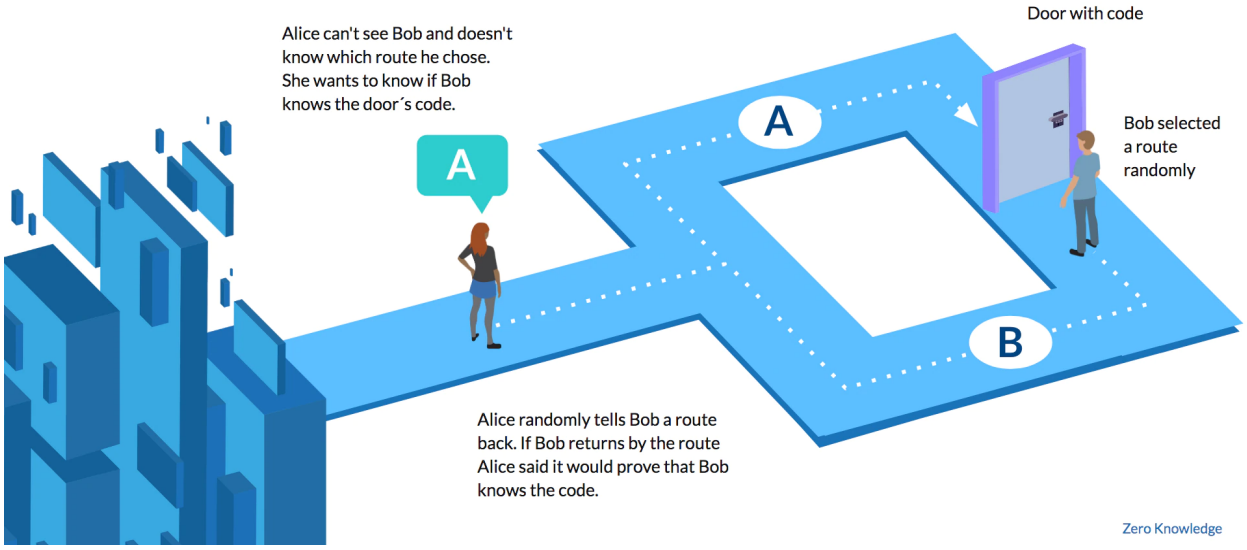
Overall Architecture of Zero-Knowledge Proofs

In the world of blockchain and cryptocurrencies, privacy is a widely accepted core ethos. Individuals use pseudonyms to hide their true identity and crypto twitter memes about finding the best non-KYC trading platforms. But 100% privacy can be difficult to achieve, and as soon

as there is any evidence doxxing a person's name or wallet address, everything is out in the open due to the public ledger of blockchain explorers.

You're probably wondering, what does this have to do with zero-knowledge proofs? Sometimes, people want to keep certain information private, like their annual income. But that information is necessary in order to, say, find out if that person qualifies for a loan from a bank. Zero knowledge proofs would be able to determine if a person qualifies for a loan **without actually revealing their income level.** This obviously protects the individual from not having their private information publicized, but it also protects the bank from not having that valuable information on hand and being the target of a cybercrime.

Zero Knowledge | Intuitive Example



bbva

In this example, Alice wants to know if Bob knows the code to the door. With no knowledge of Bob's previous route, Alice will instruct Bob to return down either side, A or B. If Bob did not know the code, he could only return from the same side he went down. Thus, this action can be repeated x number of times to increase the confidence in Bob's knowledge of the code and make sure he's not guessing (creating a probabilistic certainty.)

In this example there are two parties, a **prover** and **verifier.**

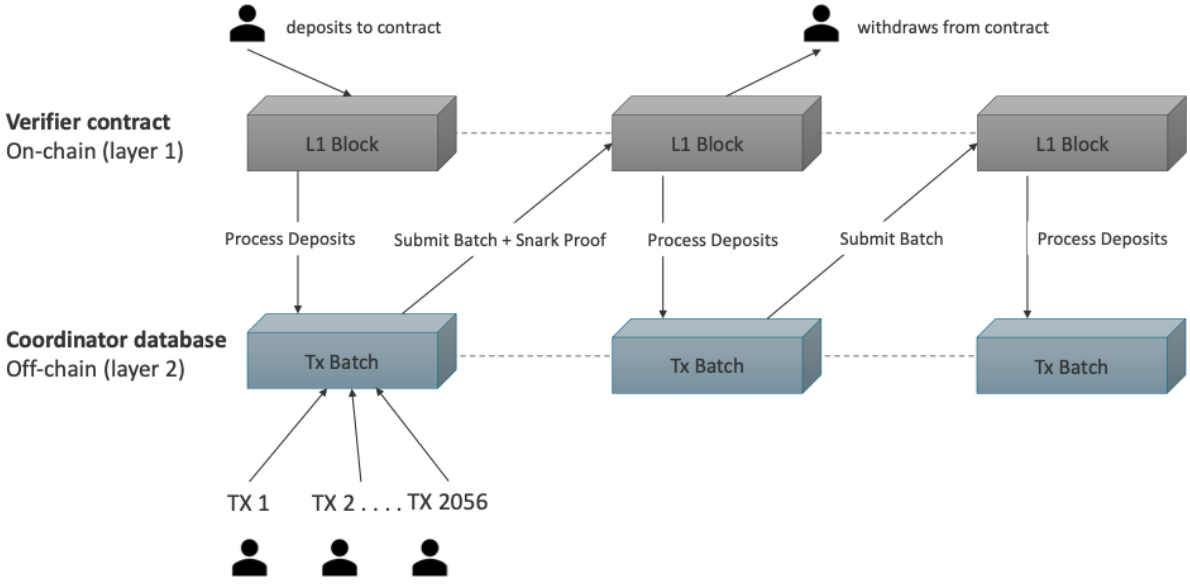
Prover: Bob wants to prove that he knows the code to the door without revealing the code itself.
Verifier: Alice is the verifier, who wants to ensure Bob knows the code.
Witness: Another important component is the secret information, the door code in our example. This is referred to as the "witness".

Early implementations of ZK proofs were seen in privacy focused blockchain networks like Zcash and Monero. Once the development of ZK-SNARKS advanced, however, Ethereum Foundation researchers began exploring using zero-knowledge proofs to help the scaling

problem on Ethereum. And as a result it is now widely accepted that these ZK scaling solutions will have a significant role in the short, medium, and even long-term scaling of Ethereum.

ZK and Ethereum

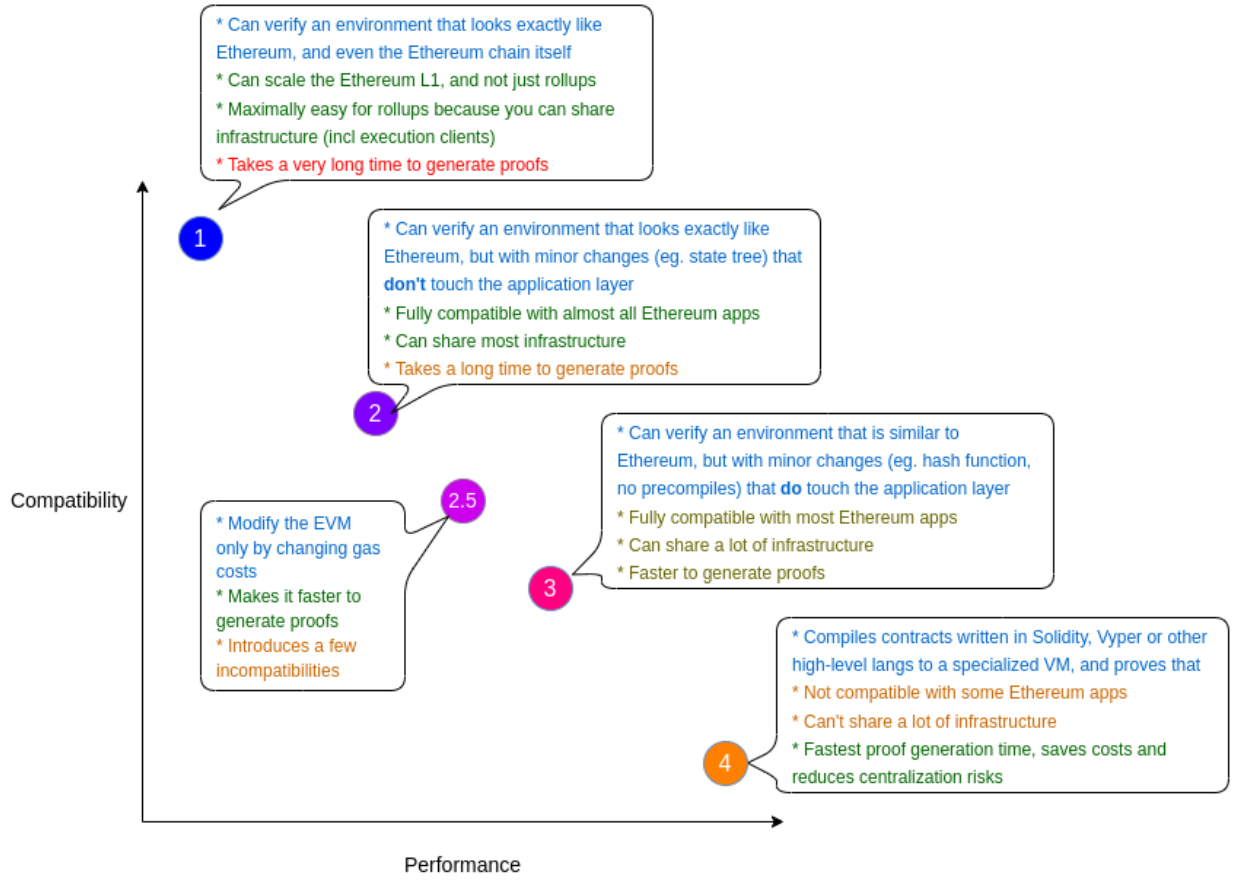
The above is a good way to conceptualize exactly how/why ZK proofs exist, but what about in the context of Ethereum? The ZK proof, or more specifically validity proofs, are what tell Ethereum what the accurate state of the L2 rolled up batch is. These state changes are maintained/confirmed on a smart contract that is stored on Ethereum, and importantly, allow for the rollup batches to have near *instant settlement* on Ethereum mainnet (whereas Optimistic rollups take much longer.)



Simon Brown

The initial research behind ZK rollups mainly supplied use-cases around simple transfers of assets. But as time has gone on, there has been an increase in smart contract support on these layer 2 chains. The current discussion is focused around EVM-equivalence, which specifies the exact level that layer 2s are interoperable with Ethereum and the EVM.

In this context, you have probably come across Vitalik's blog post dissecting the different types of zkEVMs, and what they mean in terms of Ethereum's scalability:



Vitalik Blog

This graph simply looks at EVM equivalence in terms of a tradeoff between generating the proof for a given batch of transactions (longer prover time and higher cost) and seamless integration with Ethereum. The *type one* zkEVMs mean that everything at an execution level is consistent with Ethereum, which allows for infrastructure (block explorers, audit tooling, and applications) to be reused for the L2 without any changes.

Consequently, the *type 4* zkEVMs are built in a way that requires an EVM language (Solidity or Vyper) to be converted to a zk-SNARK friendly custom language. Now that we have finally gotten to the point of SNARKS, we can address the subtle, but important difference between two types of zero-knowledge proofs.

SNARKs	STARKs
• Succinct • Non-Interactive • Argument (of) • Knowledge	• Scalable • Transparent • Argument (of) • Knowledge

For *SNarks*, the difference lies in the succinct and non-interactive nature of the proof. Non-interactive refers to the way that the proof can be verified by a third party. Rather than having the verifier interact with the prover, a smart contract can handle the interaction. Succinct simply means that the proof size is smaller than the witness statement (the secret information.)

When generating the SNARK-based proofs, there needs to be a shared key, referred to as a Common Reference String (CRS) in order for the prover and verifier to communicate. The details are complicated and not important for us users, but the takeaway is that SNARKS require a *trusted party* to set up and not compromise the CRS.

In contrast, *STarks* are **transparent** in the context of the CRS. These parameters are publicly verifiable and **do not rely on a trusted party.** Additionally, they are *scalable*, which refers to how fast proofs can be generated with an increase in the size of the witness. As the witness increases, STARKs become significantly faster, and thus more scalable than SNARKs.

The takeaway: STARKs can scale and do not rely on a trusted party. SNARKs are faster when the witness statement is smaller and are non-interactive proofs.

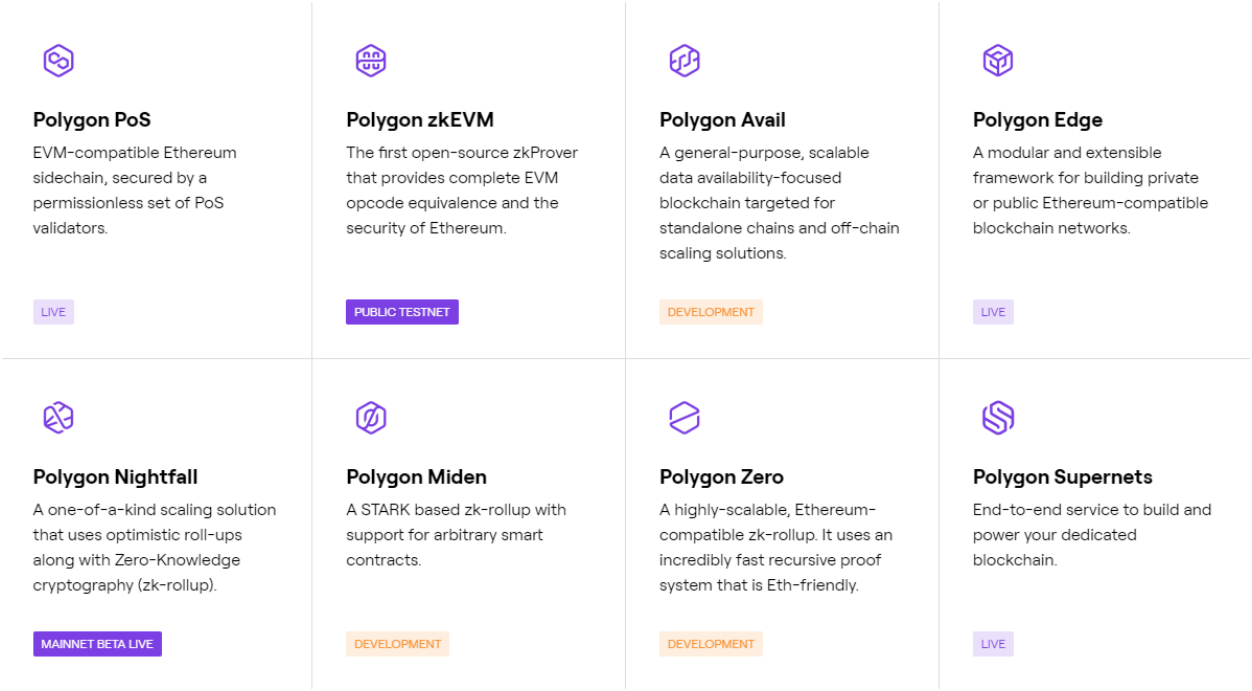
Hopefully this gives a good enough overview of the world of ZK proofs. Here is further reading relevant to the above material:

- SNARKs vs. STARKs
- Everything about Rollups
- Ethereum blog zk rollups

Now, we will take a look at some of the projects currently using zero knowledge proofs to help scale Ethereum.

Polygon zkEVM
Polygon is creating a broad suite of Ethereum scaling solutions. We won't be able to dive into all of them today, but they are worth looking into to see how $MATIC will be involved and what each distinct rollup aims to solve. One of the most interesting is Polygon Avail, which is a data availability blockchain, aiming to have a world of application specific execution chains that utilize

it as a data availability layer. This will be a competitor to Celstia, and more can be read about blockchain modularity in this essay.



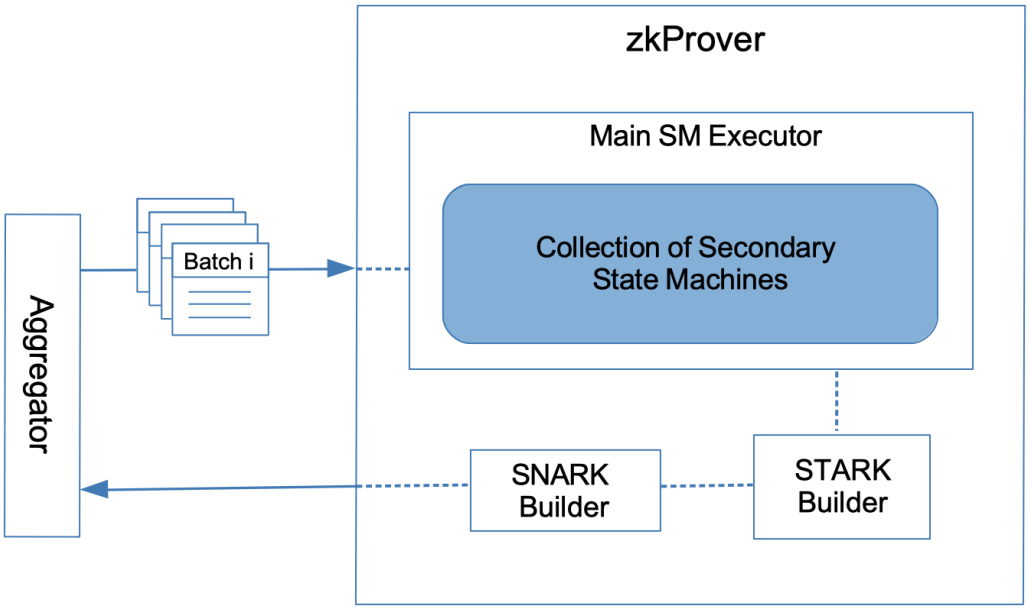
Polygon Scaling Solutions

Today we are focusing on zkEVM scaling solutions, and Polygon's is a significant leader in the space. According to our performance and compatibility tradeoff from above, Polygon zkEVM is currently a *type 3* rollup, with plans to move towards a type 2. All application level transfers would be very simple, but there would be some changes to the *data structures* that hold the Ethereum state. This simply means that devs can copy and paste their applications, and a majority of infrastructure related dev tools can be ported over easily as well (highly compatible, while still leaving some optimization for proof generation.)

Moving Towards Decentralization

This rollup solution offers a unique consensus model to help decentralize the ZK proof generation process (which is by default centralized,) by using the Proof of Efficiency (PoE) consensus model. This determines who will be submitting the next batch of transactions, illustrated below. The other unique component is the zkProver, which involves both sequencers and aggregators.

Sequencer: The sequencer proposes transaction batches to be included. Anyone can be a sequencer by running the relevant software (zkNode.)
Aggregators: The aggregators check the validity of transaction batches and provide the proofs to the smart contract on Layer 1.

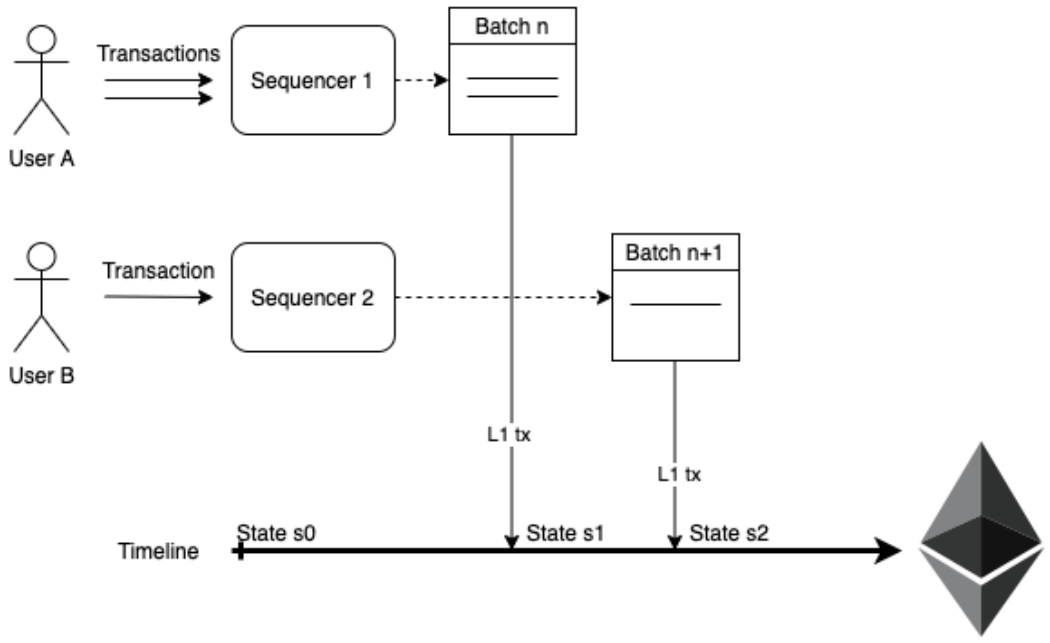


Polygon Docs

Anyone can be a sequencer on the network by putting up a stake of $MATIC tokens and running the zkEVM software, which is an important part of trying to maintain sufficient decentralization. In practice, these sequencers can be wallet platforms, a game, DeFi application, exchange, etc.

Aggregators on the other hand, require specialized hardware to be able to create the proofs for the transaction batches. This creates a level of centralization, which was already going to be the case because aggregators race one another to generate and deliver the proofs fastest, so it would end up that the best wins anyway. But the PoE model aims to decentralize this as best they can, and it is a great start for the current stage of ZK rollups.

Here, anyone can accept a batch from a sequencer and race to generate the proof, with the PoE smart contract on Ethereum's layer 1 accepting the first proof posted:

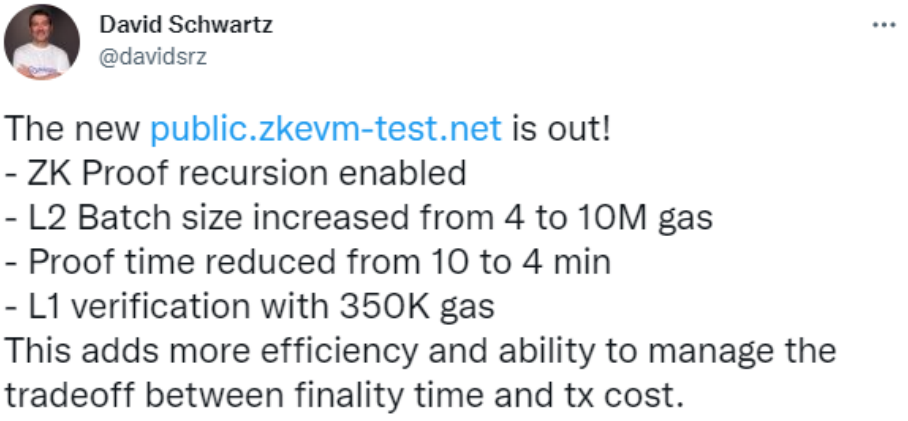


Eth Research

Once the proof is accepted, the Layer 1 contract will update the states without re-executing the transactions: which is why the gas consumption, speed, and network usage is optimized under a ZK rollup structure. Thousands of transactions can be batched into a single one on the Ethereum mainnet.

Usage and Production Timeline

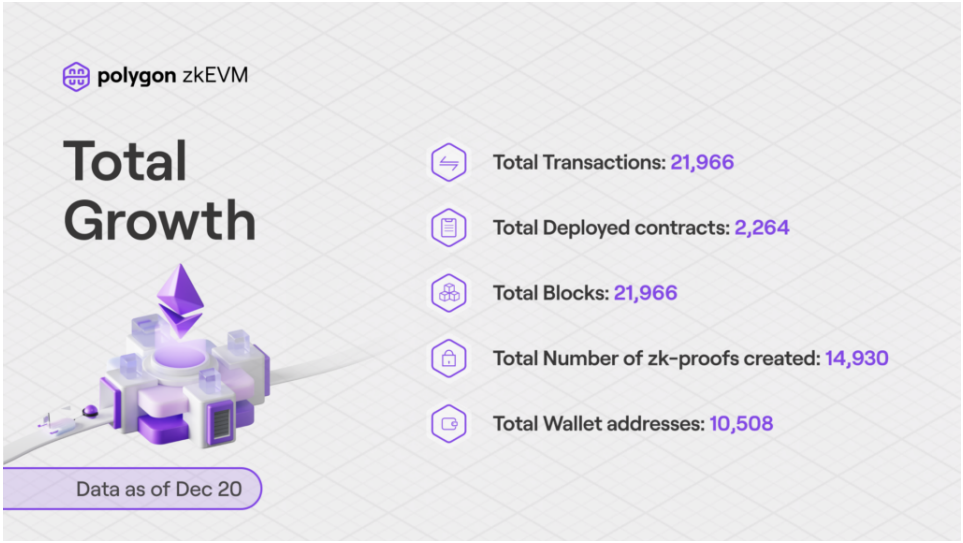
As we mentioned in the intro, a lot of these ZK rollups are just now reaching an inflection point of development and beginning to onboard developers to their respective testnets. Polygon had their zkEVM pubic testnet launch in October, and just recently announced the launching of their *final testnet* as their mainnet is undergoing the final audits. This final release comes with a handful of improvements from the last testnet as well:



Twitter

$MATIC will be used for staking and governance within the Polygon zkEVM, which will likely be similar across the other scaling solutions being built by the team. We have been keeping an eye

on $MATIC for this reason. The testnet statistics aren't that significant of data, but it does show that there is activity and developer interest:



Polygon Blog

Looking ahead, we can reasonably anticipate that the ZK scaling space will be a playground for new DeFi and GameFi applications. Much like the growth on Arbitrum, there will be many developers looking to build unique applications on these rollups in order to tap into the interested and growing Ethereum user base. As these ZK rollups ship, however, they will be competing with the well-grounded Optimistic rollups in Arbitrum and Optimism, and competition will be stiff in order to attract capital. To overcome this competition, we will likely see a similar playbook to the GMX stack playout: a native project seeding a large majority of innovation and interest with many other developer groups building on top of that.

Twitter - and check out this dashboard comparing vault activity

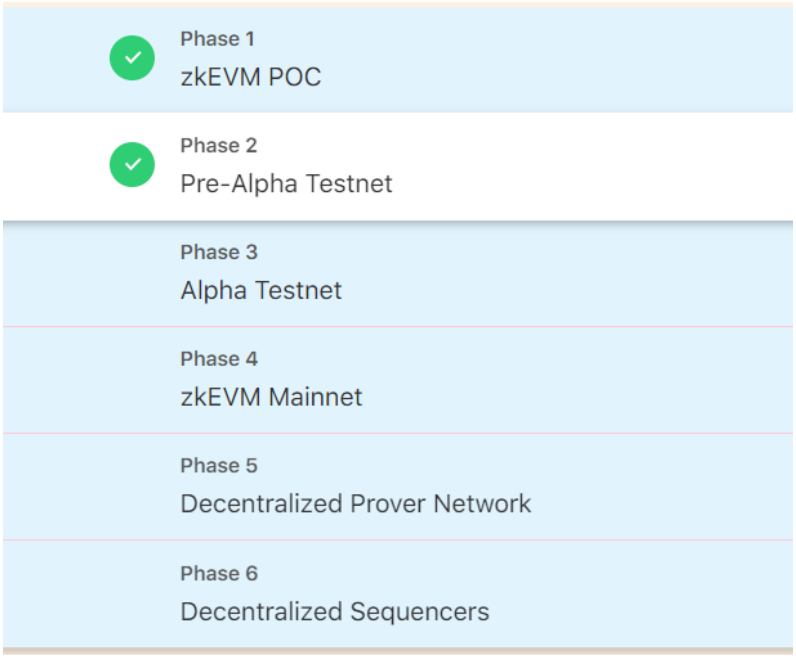
With a sector like ZK scaling in such an early stage, speed really matters. Obviously, these teams are not sacrificing quality just to be first, but it is quite clear that teams are moving fast to try to be first to market in their zkEVM. Polygon seems to be on a solid track to be one of the first groups.

Scroll

Scroll's zkEVM is most similar to Polygon in terms of its EVM equivalence. Scroll's proving architecture is referred to as "Roller network", which aims to also create a permissionless and decentralized network of prover nodes. More specifically, an excerpt from Scroll's mirror post about their zkEVM network:

"The community will be incentivized to build substantially better hardware solutions and run provers themselves instead of relying only on the Scroll team in a centralized way. To bootstrap in the initial phase of the network, we are building GPU prover solutions internally which we will open-source for public usage. As this matures, we are exploring ASIC and FPGA solutions with several hardware companies. In the long run, we look forward to vibrant competition in this domain and firmly believe that latency and cost for proof generation will decrease exponentially."

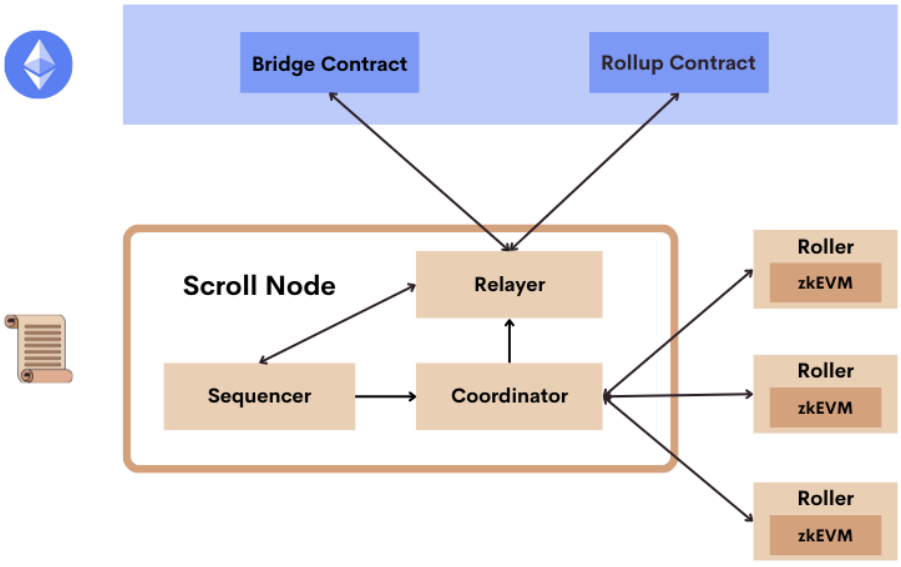
Our takeaway from this essentially means the team is hard at work at finding ways to decentralize the network, but nothing is set in stone yet. Which makes sense given the stage of development the product is in: the timeline for Scroll is as follows



Scroll.io

As you can see, the decentralization of the network will not be happening until later on in its development, but it is still important to understand how it will be implemented. According to Vitalik's post linked above, Scroll is another type 3 at the present moment with a goal of moving towards type 2 in the future.

The architecture of the Scroll zkEVM differs slightly from Polygon, most notably in the selection for who provides the proof:



Scroll Architecture

As noted above, the “Roller” is the prover in the network, which means they generate the proofs. Remember that this requires a high computational effort, thus the default standard is that proof generation is somewhat centralized. Scroll tries to combat this by allowing the **coordinator to randomly select a prover** from the *Roller network*. This strategy seems more decentralized than Polygon’s, where it was a race to generate the proof, meaning the fastest would always win. This ensures that the prover network will stay competitive as anyone will have a chance.

The **rollup contract** provides data availability for Scroll blocks, and once a proof has been verified by the rollup contract, it is considered finalized on Scroll. The bridge contract is separate to allow for secure asset transfers between the L1 and L2 networks.

Next up in the zkEVM space, we move to two projects that are currently considered *type 4* according to our above framework. The most important distinction here is that type 4 zkEVMs use a language compiler to translate EVM programming language (Solidity and Vyper) into another high level language.

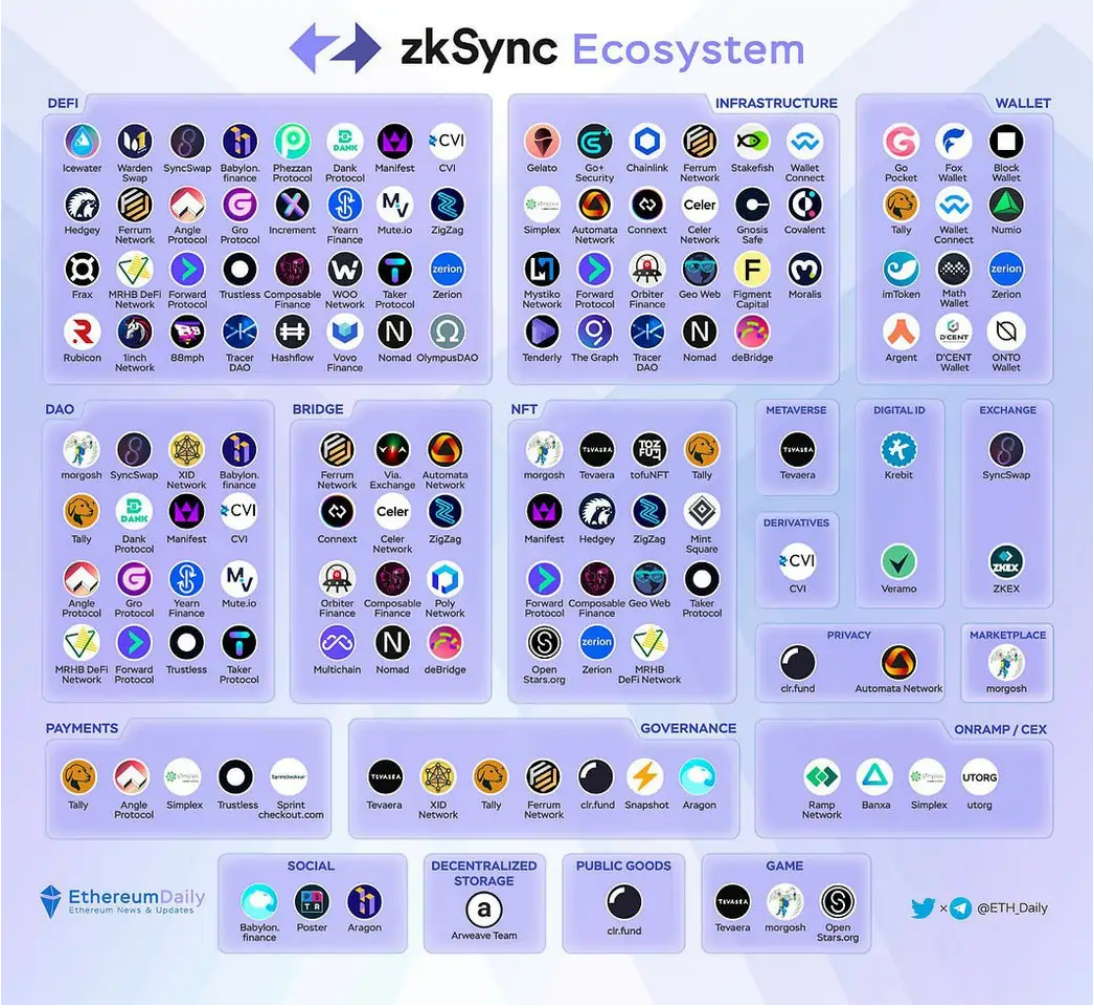
Matter Labs - zkSync
As a SNARK based rollup, zkSync relies on the trust assumptions associated with the CRS. The team decided to use a “universal setup” which means the CRS is not application-specific. Thus, they argue that zkSync can be referred to as a fully trustless protocol.

Even though zkSync claims to be an EVM based ecosystem, it still requires that Solidity be transpiled to Zinc, the programming language that is designed specifically for ZK-SNARK based smart contracts. Vitalik outlines the advantages and disadvantages to this type of ZK rollup design:

Advantage: Much faster prover times due to the ZK-SNARK friendly bytecode, which helps ease the decentralization efforts to produce proofs compared to what we discussed above.

Disadvantage: Contracts may not have the same addresses, EVM bytecode does not port over directly, and debugging infrastructure cannot be carried over.

Nonetheless, the Zinc transpiler has made it easy and possible for a lot of projects to build with zkSync, as it boasts one of the largest suites of applications for all ZK rollups:



Chemix.io

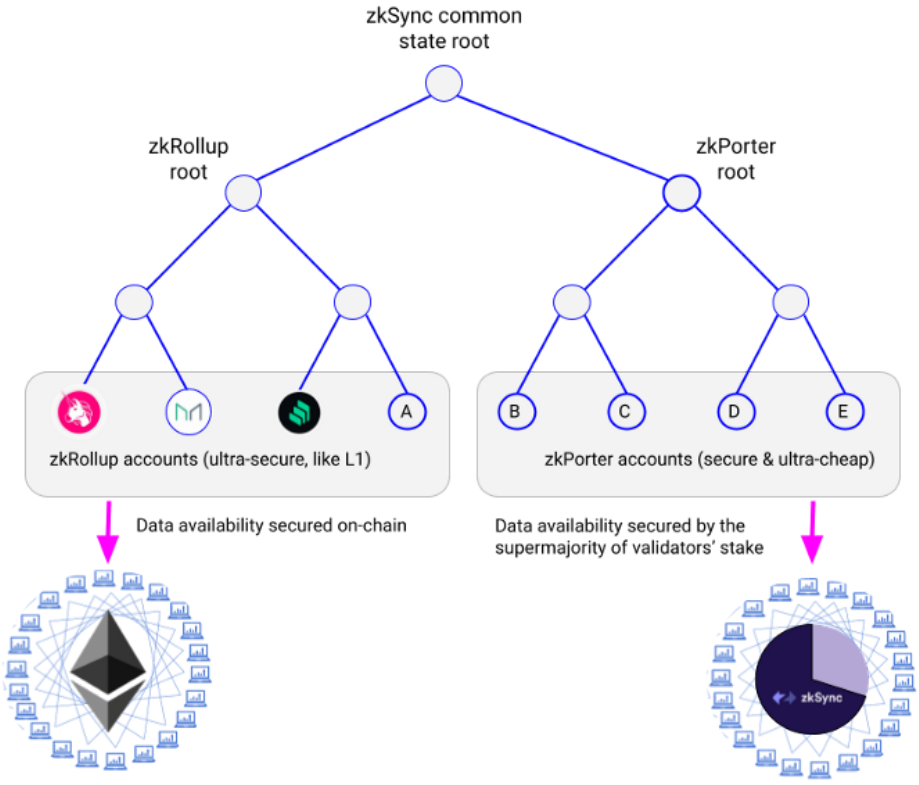
Token

A lot of the hype around zkSync is due to the fact that they have publicly disclosed that there will be a token associated with the network. There is no set timeline, nor any clear indication on how it will be distributed. This means there is a possibility that there is *no airdrop*, but we think it is worth it to play around with the network anyway to ensure that you qualify in the event the token is airdropped.

Here is a detailed guide on how you might be able to qualify, but in general the best strategy is to just find what you like on the network and stay involved.

zkPorter

The token (we can just deem it $ZKS for simplicity) plays a significant role with one of the main unique selling points of zkSync: zkPorter. $ZKS will be used to secure the data availability of the zkPorter accounts, which is a hyper-scalable branch connected to the SNARK based rollup root:



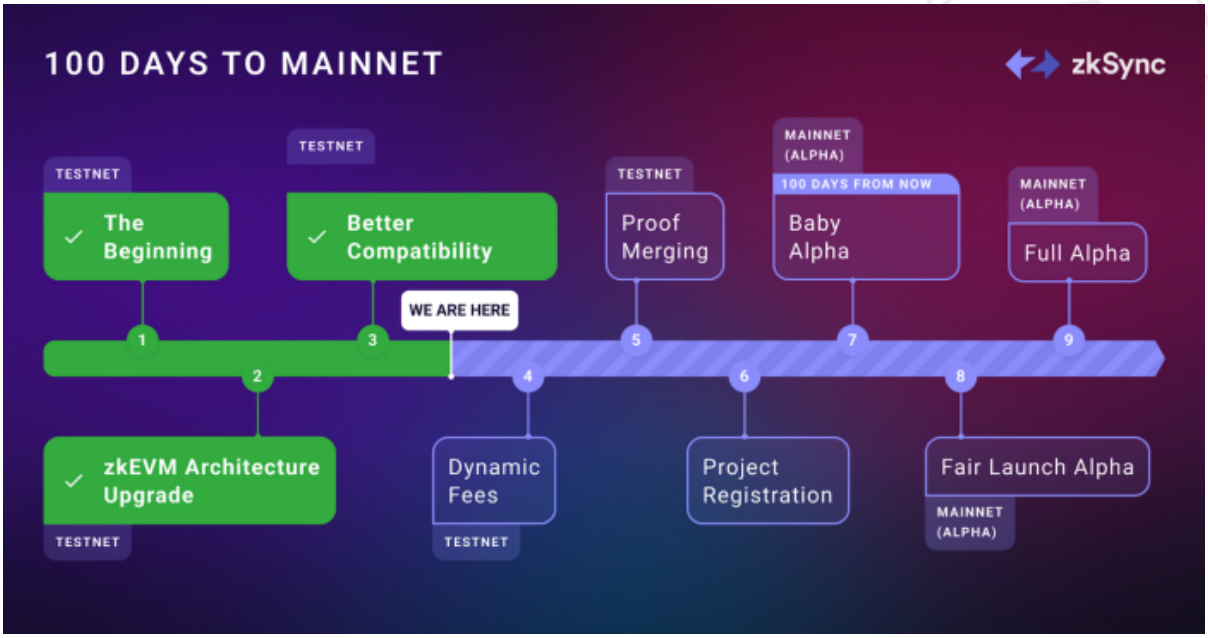
Matter Labs Blog

The common state root pictured above allows for the zkPorter and zkRollup side to share a state and communicate with one another seamlessly. The post assures that the PoS in zkSync is more secure than other PoS systems because malicious $ZKS stakers, called Guardians, will only be able to freeze the zkPorter state, which negates any economic profit they could extract (they would freeze their own stake in the process.)

The zkPorter is an L3, which is important to know because, in general, the farther that we move away from Ethereum mainnet, the less security we have. The nuances of this conversation are beyond our scope, but Vitalik illustrates this point well here.

Timeline

Like the other projects we have talked about, the timeline is opaque and consists of words like "baby alpha", which is just another phrase for still in development. This graphic was initially launched with the "100 days from now" date landing in mid-October, and since that day came and went with no baby alpha launch, who knows how far we are from the actual launch:



They released a post stating that the fair onboarding alpha would be hit in Q1 2023 and fair launch landing in Q2 due to audits that are currently going on. After raising a $200M Series C, their investors are probably not too happy with the slowdown, but no one should be surprised.

Account Abstraction

One of our recurring discussions throughout this year has been the benefit that social recovery wallets, like Argent, bring to the lay-person's crypto experience. But diving into the details of this social recovery leads us to the concept of account abstraction (AA), which has been a large focus of zkSync, and the next project we are discussing, Starknet.

Account abstraction benefits a network by simplifying the user's experience with wallets. Rather than having both externally owned accounts (user stores private key) and smart contract accounts, AA only uses contract accounts. The nitty gritty detail is summarized well in this blog post, but we find some of these use cases game-changing for the user and development experience:

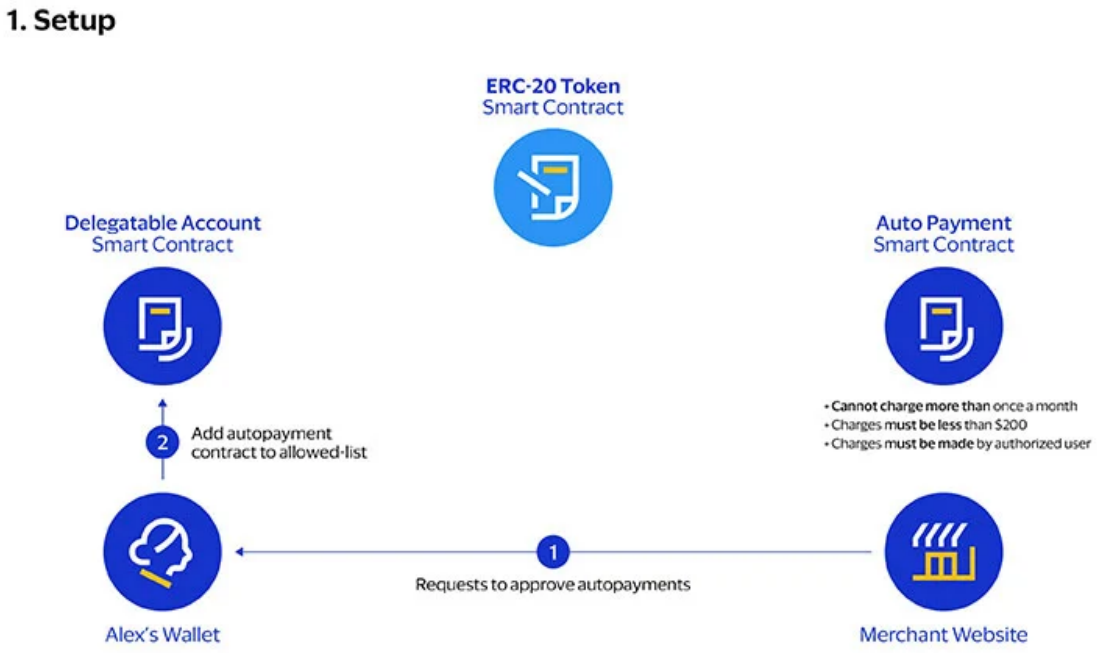
- Paying gas fees for someone else
- Paying gas fees in a different token than $ETH
- Batching transactions via session keys for one gas payment
- Differentiating valid transactions for different keys within the same account

AA is obviously going to have a notable impact on the UX of blockchains in general, and the development therein will be important to watch for the growth of ZK rollups.

Starknet

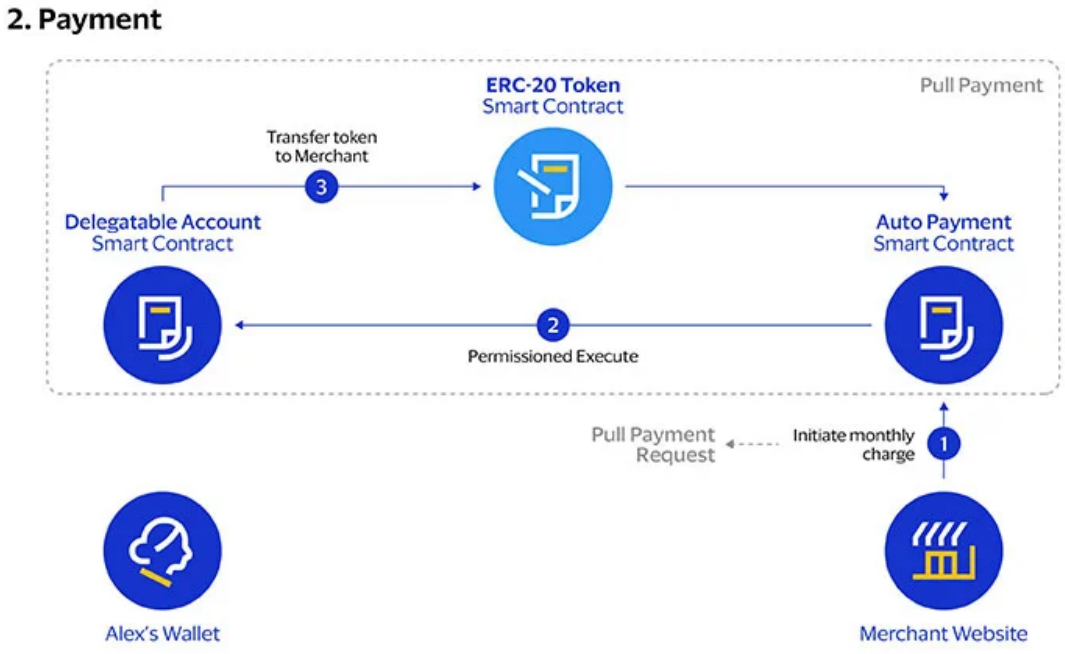
As the name implies, Starknet is a STARK based ZK rollup, which has the benefits of scalability and trustless CRS setup. It's most notable adoption with (currently) dYdX and Immutable, with additional strong L1 applications working on its testnet. But perhaps one of the most eye-catching developments in this report is Visa beginning to explore Stakware's L2. On

Tuesday, Catherine Gu announced that Visa had used AA discussed above to create a unique way for customers to set up a recurring payment:



Visa Paper

And after setting this up, the smart contract account is able to execute permissionless payments at a regular interval.



Visa Paper

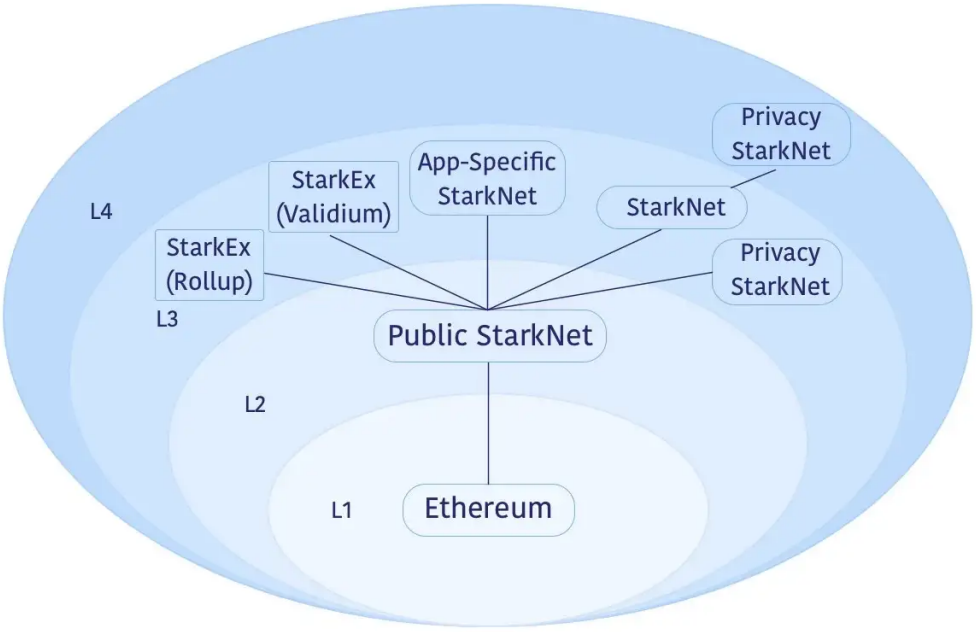
The significance of this cannot be overstated. Not only does it showcase a true, non-gamified, non-token oriented use case of crypto/blockchain technology, but it also taps into features that are only available on these unique ZK scaling solutions. Both Ethereum and ZK get mainstream credibility points for this one.

Cairo

If you have heard of Starknet, it is probable that you've come across the word "Cairo". This is the native language that the Starkware team developed that is more efficient with ZK SNARK proofs just like zkSync's Zinc language. There is currently no compiler that will allow for devs to move over their Solidity based applications to Cairo, but that will change when Nethermind's Warp project launches. When that does happen, we will probably hear a lot more buzz around Starknet and its spot in the zkEVM race.

Fractal Scaling/Hyperscaling

With Starknet, they have also emphasized their focus on L3 scaling, which is often referred to as fractal or hyperscaling. The architecture of the L3 ⟷ L2 is similar to that of L2 ⟷ L1, with a validity proofs submitted from L3 to L2. The benefits of this can be thought of as similar to the Cosmos ecosystem, with application specific chains that are highly customizable:



Starkware Blog

The development of this, similar to zkSync's zkPorter, is dependent on a successful implementation of the L2. Starknet is also working hard on implementing account abstraction with their scaling products, offering zero gas fees on the NFT based rollup Immutable X.

Starknet will also have a token, announced earlier this year (with a projected launch date that has already passed.) Although there is no specific plan on an airdrop, there is an "unallocated" portion of the supply that could imply an airdrop.

Comparing Optimistic Rollups and ZK Rollups

The links provided in this post, specifically this one, do a good job of going in detail on how optimistic and ZK rollups are different from one another. But in our words, optimistic rollups are "optimistic" because they rely on fraud proofs to determine the validity of a batch of transactions on the Layer 2 network. This means that transactions are innocent until proven guilty. Anyone can be watching the batches of transactions, and if someone suspects that a batch contains a malicious transaction, the chain will enter a dispute period and verify which state is correct.

Penalties are in place for both the sequencer and fraud proof submitter to deter bad actors. Accordingly, for transactions to be treated as valid and final on Ethereum mainnet, they must pass the challenge period of roughly 14 days. Why does this strategy not create a permanent solution to Ethereum scaling?

1. Centralization

There are centralized "rollup operators" that are responsible for producing blocks on the Optimistic Layer 2 networks. There are some guardrails in place, such as having a group of nodes ready to fill in and assume the operators role if they were to go offline by using the publicly available on-chain rollup data. But ZK rollup sequencers only batch transactions together, and submit them to a prover, meaning there is no dependency on this specific party.

2. Long time to withdraw funds

The 14 day dispute period is not user friendly, and although there are bridges that have abstracted this away by offering instant liquidity, the core problem still exists. Because ZK proofs offer instant verification of batches (once finality is reached on L1), then the funds can be accessed exponentially faster.

3. Not as flexible as ZK rollups

Although we are a ways away from this becoming a reality, ZK rollups hyperscaling ability is important to the future of Ethereum. GameFi applications rely on two things that have been a focus of ZK rollups: extremely fast TPS and batched transactions (AA.) While optimistic rollups are a significant increase in the TPS, the ceiling for ZKs are much higher.

Conclusion

ZK or optimistic, which is better? The answer, of course, is nuanced, and even at the end of the discussion there is no clear cut right or wrong. In an early blog post, Arbitrum's development team Offchain Labs argued that Optimistic were superior, but in reality both have their own advantages and disadvantages.

In a world where we are increasingly confident that there will be many, not just one, blockchain networks, learning about the stark differences between ZK and optimistic rollups is important. Both will have their respective advantages for developers and users, and be able to command a slice of the market accordingly.

As discussed throughout the post, we believe that there will be a similar level of fragmentation across these ZK rollups as we have seen with optimistic. We will likely see a strong first mover advantage, but from that point forward, liquidity and general popularity/usage will be determined by the network that has the most innovative protocols (GMX and Arbitrum.) Each chain may

have their comparative strengths and weaknesses, but **before those are solidified, it will be high intensity PvP for attracting users.**

What we can more definitively say is that the current adoption that has been going on with optimistic rollups **will be seen with ZK rollups as well**. It might be in 6 months, 12 months, or 18 months, but eventually it will come. Stay aware of protocol updates in these different ecosystems and be ready to capitalize on new opportunities. Happy hunting.