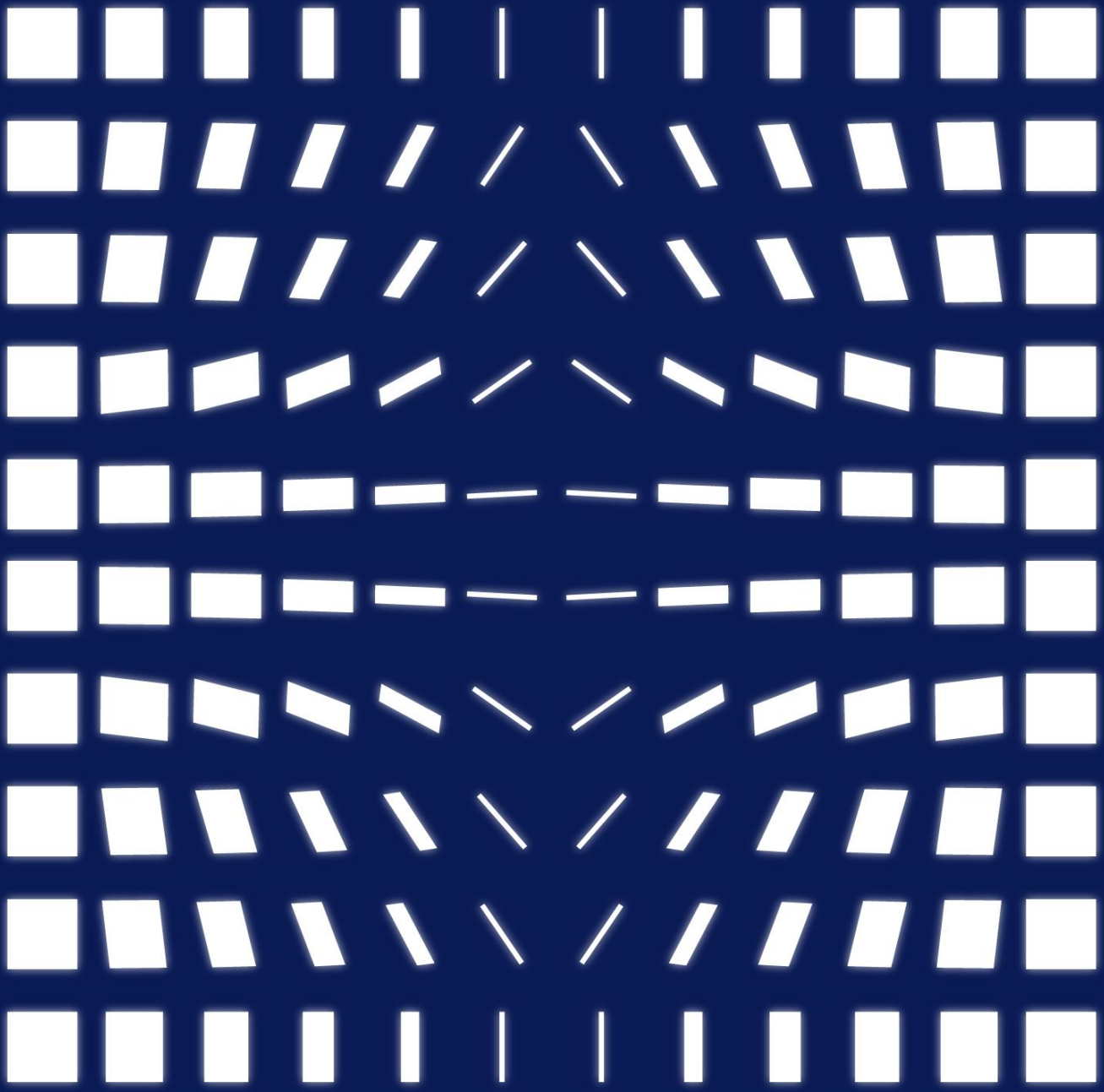
Huobi Research

Will zkEVM be Ethereum's Savior?



Author: Huobi Research Institute / Jiang Mengchu, Han Xiaopeng

Will zkEVM be Ethereum's Savior?

Abstract:

This article compares the features of two mainstream Rollup solutions and concludes that zkRollup has better performance but poor compatibility, thus limiting its application scope. zkRollup needs to include zkEVM to be able to run all kinds of general-purpose smart contracts in order to address its shortcomings. The main aspects of zkEVM and the characteristics of its two different technical routes are then described, and the mainstream zkEVM projects are introduced. Finally, we envision that zkEVM will act as a savior for Ethereum in the near future to help expand capacity, with the possibility to play a role on other public chains in the longer term.

Ethereum has no doubt gained fame as the superstar public chain capable of running smart contracts. However, traffic on the Ethereum network is so busy that “traffic jams” often occur. Even a “highway” like Ethereum must confront the reality: a long queue of transactions waiting to be packed and sent, despite exorbitant tolls (gas fees). The current network congestion was not unexpected. In an earlier incident dating back to 2018, a game named CryptoKitties showcased Ethereum’s weakness in handling mass transactions. Today, the number of users and Dapps that use the Ethereum network have soared to several times that of those in 2018. An update to alleviate the congestion cannot wait.

What if there was a Layer 2 solution that could offer fast and secure transactions at low gas fees, that is highly compatible with the vast majority of smart contracts and Dapps, yet private enough without the controversial week-long waiting time? Would this make the wish list? The truth is, a Layer 2 solution of this nearly perfect nature could probably be ‘live’ in the market in the near future.

What could Layer 2 and Rollup Do?

The transition to ETH 2.0 is best viewed as a massive renovation of “the highway”; the process will be an arduous one since the upgrade is naturally difficult to implement, with the interests of various parties to be taken into consideration. A Layer 2 solution can be viewed as an indirect remedy in that it creates overpasses above “the highway” in order to achieve scaling. The mainstream Layer 2 scaling technique is known as Rollup: it handles more transactions without taking up space, by compiling data. In other words, larger data packs could commute like a bus on "the highway" instead of cars with limited capacity; thus, more passengers, a.k.a. data in this case, can be transported.

Rollup can be classified according to whether it interacts with proofs submitted to Layer 1. There are 2 Rollup technical tracks. The first is Interactive Rollup. For short, we shall classify one-round interaction and multi-round interaction as Optimistic Rollup (OP Rollup). Arbitrum, Boba Network and Optimism are some outstanding projects in this category: the number of projects in the ecosystem and TVL have skyrocketed of late, and the TVL of these three projects occupy almost 70% of the entire Layer 2 market. The other is Non-interactive Rollup, also known as ZK Rollup; examples of projects in this track are dYdX, Loopring and zkSync. These have a comparatively smaller market share.

Just as how it performs in Optimistic Rollup: the fraud proof could still be flawed even

Table 1：Comparison of ETH2.0, zkRollup and Optimistic Rollup

	ETH2.0	ZK Rollup	Optimistic Rollup
Data Stored On-chain (Data Availability)	Yes	Yes	Yes
Compatibility with Common Smart Contract	Easy	Hard	Easy
On-chain transaction costs（gas fee）	Extremely Low	5% of the current Ethereum L1	30% of the current Ethereum L1
Exit Time	No	A few minutes	1 week
Security	High	High	High
Difficulty of Development	Highest	High	Low

Source: Huobi Research Institute

after a week-long verification period. However, ZK Rollup prefers a much faster and result-driven "state proof". As the nature of verification differs, ZK Rollup outshines OP Rollup with better performance, lower transaction fees and unconstrained exit times.

ZK Rollup has an absolute advantage and could be a superstar and inject more energy into Ethereum scaling, but why does it remain reclusive in this booming market?

Disadvantages of ZK Rollup

The answer: compatibility, the main disadvantage of ZK Rollup compared to OP Rollup. As illustrated below, ZK Rollup is only compatible with payment and transaction applications, while OP Rollup supports a greater variety. ZK Rollup lacks appropriate development, which has led to its comparatively low 3.6% market share of the Layer 2 Ethereum market, TVL-wise.

Figure 1: Rankings of Projects in TVL

No.	Name	TVL	Breakdown	7d Change	Market share	Purpose	Technology
1.	Arbitrum	$2.65B		-5.94%	40.04%	Universal	Optimistic Rollup
2.	Boba Network OP	$1.31B		-8.77%	19.81%	Universal	Optimistic Rollup
3.	dYdX	$896M		-6.70%	13.54%	Exchange	ZK Rollup
4.	Loopring	$642M		-14.59%	9.71%	Payments, Exchange	ZK Rollup
5.	Optimism OP	$466M		-4.13%	7.05%	Universal	Optimistic Rollup
6.	ZKSwap V2	$202M		-13.36%	3.06%	Payments, Exchange	ZK Rollup
7.	ImmutableX	$188M		-28.50%	2.84%	NFT, Exchange	Validium
8.	DeversiFi	$103M		+22.40%	1.56%	Exchange	Validium
9.	Metis Andromeda OP	$60.42M		+46.06%	0.91%	Universal	Optimistic Rollup
10.	zkSync	$54.89M		+27.36%	0.83%	Payments	ZK Rollup

Source: L2beats

ZK Rollup submits Zero Knowledge Proof to the Ethereum mainnet as an effective proof; and the inherent complexity of generating Zero Knowledge Proof triggers low compatibility with most applications. In this intricate process, logic of codes must be first converted to a mathematical circuit, not only including basic calculations such as plus or minus, but also accompanied with convoluted logic such as “and”, “or”, “Not”,

Hash, bit operation, and other operations on smart contracts. Moreover, this diagram can only support plus and multiplication calculations, and Ethereum opcode is not Zero Knowledge Proof friendly, as it was not designed to be. Furthermore, the most frequently deployed Hashing methods, such as AEW-128 or SHA-256, consist of enormous bit operations ("and" and "or" commands); it would be extremely complicated and substantial when converted to gate constraints in the circulation.

With the diffusion of Zero Knowledge Proof, Zero Knowledge powered solutions cannot successfully kick off without the help of the rising star - zkEVM, which plays a fundamental role in building the second "overpass".

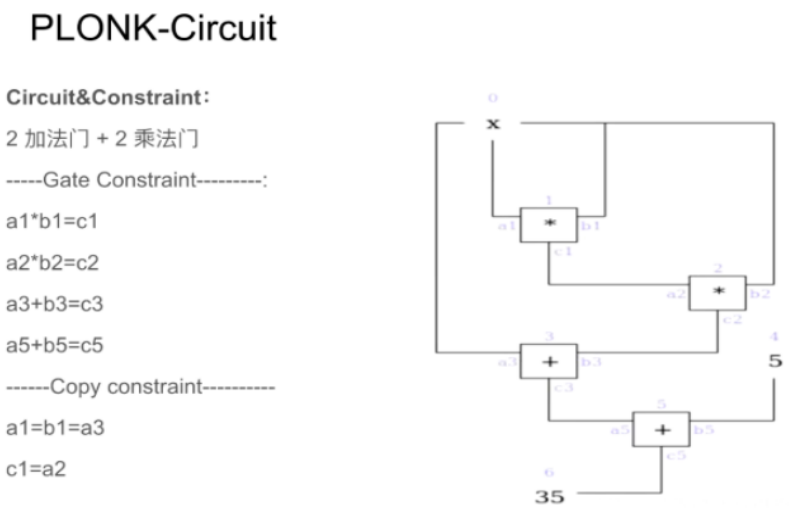
We need zkEVM

The current ZK Rollup has constructed a parallel path above the Ethereum boulevard. However, it is more like a bike trail that can only carry out simple transactions, whereas loaded trucks (a.k.a smart contracts) are too heavy to pass. In this case, should ZK Rollup conquer a larger market share by running smart contracts and produce proofs on Layer 2 in order for verification on the mainnet to pass faster, a special virtual machine, zkEVM, must be in place.

There are two key requirements for such a circumstance. First, that zkEVM be compatible with current EVM so codes on Layer 1 can be executed immediately on Layer 2. Second, zkEVM must be capable of producing proofs for various operations while consuming less computation and storage resources.

Fortunately, thanks to the vast development of Zero Knowledge Proof, a new algorithm, “Plonk Zero Knowledge Proof”, has arrived, accelerating the uptake of zkEVM. “Plonk Zero Knowledge Proof” does not produce the proof by performing calculations on the entire circuit from head to toe; instead, it only verifies constraints in the circuit. As shown in the following chart, so long as the gate constraint and the copy constraint are verified, the whole circuit can be verified. In addition, the new algorithm sets a trust mark to the whole unit instead of partially trusted; the verification process thus speeds up.

Figure 2：Plonk circuit



Source: Huobi Research Institute

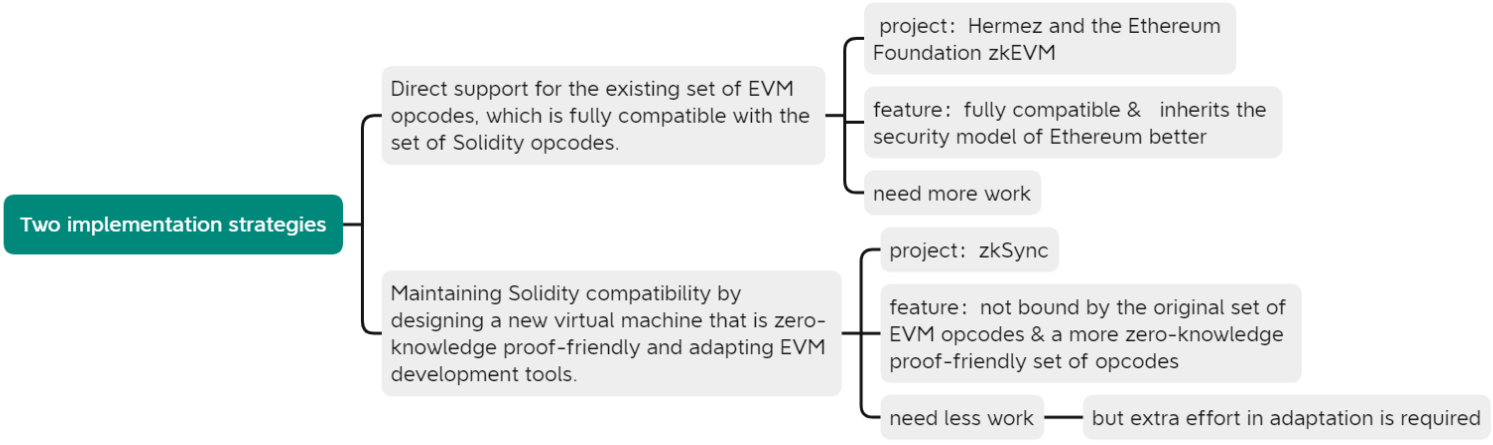
Tech Tracks of zkEVM

There are two mainstream tech tracks:

First, EVM friendly projects which embed Zero Knowledge Proof in current EVM. It aims to provide further support to native EVM opcode so codes are still executed in EVM and completely compatible with solidity commands. Applied ZKP and Hermez draw the most attention in this track.

Second, Zero Knowledge Proof friendly projects that build EVM with the foundation on Zero Knowledge Proof friendly opcode. This track focuses on the redesign of the virtual machine, so codes executed here can generate Zero Knowledge Proof more easily. That is to say, the original Zero Knowledge Proof unfriendly codes will be modified, adapting EVM developer tools in order to maintain compatibility with solidity. Matter Labs (zkSync 2.0) represents most projects in this track.

Figure 3：Two Tracks of zkEVM



Source: Huobi Research Institute

The appeal of an EVM friendly track is compatibility. It is completely compatible with current ecosystem and developer tools, yet has inherited the credited security model from Ethereum. From the overview of Ethereum ecosystem, this type of update would be so gradual that current projects can be transferred smoothly. However, as mentioned above, Zero Knowledge Proof will also be generated for those commands, deviating from the generation of proofs, which may consume enormous resources along the way.

A Zero Knowledge Proof friendly track wins flexibility-wise. It does not strictly generate proofs for every single command, but codes to a set of commands that is more Zero Knowledge Proof friendly instead. Throughout the transformation of codes, Zero Knowledge Proofs are generated while sustaining smart contract functions. As a result, avant-garde projects are more likely to be attracted to participate by the tempting benefits of considerately smaller workloads and less difficulties encountered. However, extra workloads may be conducted to transform EVM codes to intermediary codes, especially when replacing the most frequently deployed Keccak Hash functions with other functions. It remains untested as to whether this transformation process can be flawless or bring extra security and compatibility problems to the table:

Table 2：Advantages, shortcomings and outstanding projects from the two zkEVM technology tracks

Technical Track	EVM-friendly technical track	Zero-Knowledge Proofs friendly technical track
Advantage	Better compatibility and safety	Better flexibility
Shortcoming	Part of Opcodes are difficult to generate ZKP, and the workload is large	Additional adaptation is required, which may cause security risks
Projects	Hermez; Applied ZKP (the Ethereum Foundation EVM)	zkSync 2.0;

Source: Huobi Research Institute

Typical projects

zkSync 2.0

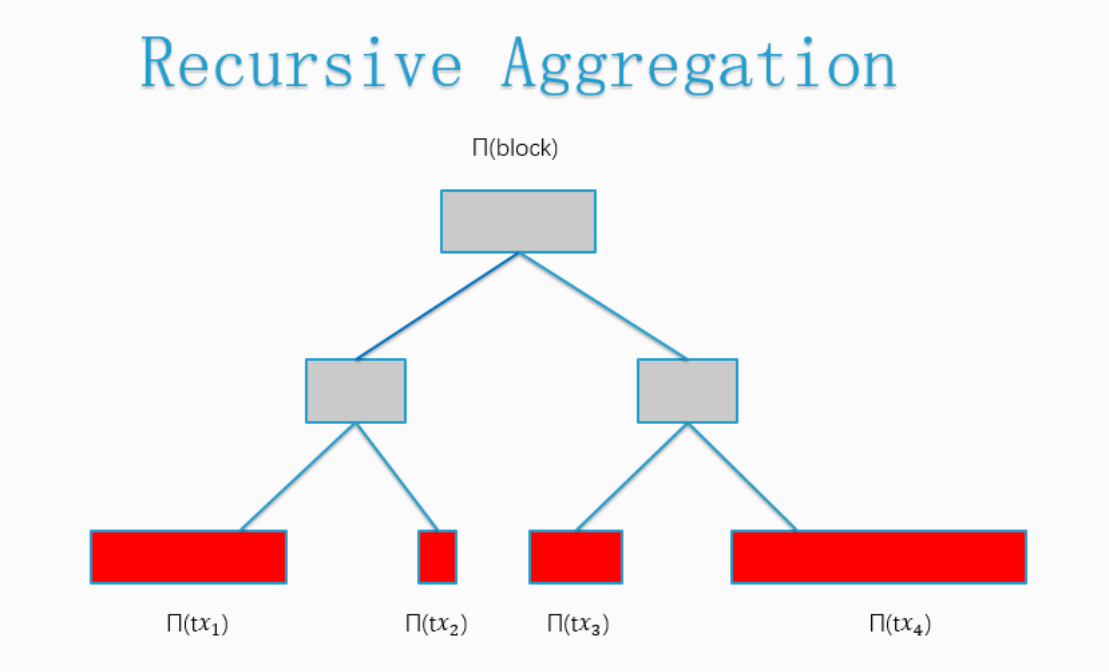
ZkSync 2.0 is is a Zero Knowledge Proof friendly project. The updated version of zkSync 1.0, zkSync 2.0, is EVM compatible. Matter Labs, its creators, explored various technical implementation plans, including TinyRam, Optimized Special Ops, Recursive Aggregation, etc., in order to realize zkEVM on zkSync 2.0.

TinyRam is a simple and traditional random browser for the R1SC circuit (a commonly used circuit in Zero Knowledge Proof). It processes logic from smart contracts and generate circuits for common opcodes. However, its resource consumption could be tremendous: the number of gates in TinyRam could be a thousand times more than that in normal fixed circuit. In other words, in a normal fixed circuit, an “add” calculation could be done via just one gate, whereas 1000 gates must be involved in TinyRam; the more gates, the higher the gas fee. Even though TinyRam is somewhat inefficient, it has minor influence on the whole consumption structure because a rather small percentage is reserved for dealing with logic. This is a tradeoff between efficiency and compatibility, and the latter carries more weight for zkEVM.

However, there are some longer commands in EVM, such as CALL, DATACOPY, EXP, CREATE, and so on. These commands are naturally unfriendly to circuit proof. For these special commands, zkEVM or zkSync inserts Optimized Special Ops accordingly in order to facilitate the expression of these longer commands in EVM via specially designed codes as intermediary.

To enhance verification efficiency, EVM adds Recursive Aggregation. Through Recursive Aggregation, those proofs, which originally must undergo the verification process separately, only need to be transformed into a binary tree as root proof and verified; it is sufficient to verify that all proofs from leaf nodes are correct. Thus, verification efficiency is enhanced.

Figure 4: Recursive Aggregation



Source: Huobi Research Institute

As mentioned above, the Zero Knowledge Proof friendly tech track employs the method of recreating a set of opcode to support Zero Knowledge Proof, which makes it improbable that ZkSync 2.0 might be the first mature product in the market. Matter Labs has already launched closed beta for zkSync 2.0 that runs Uniswap V2; a trial experience to the Testnet is recommended if more relevant information is desired. The team has not announced the official schedule for the mainnet launch, and it is likely that more tests must be conducted in order to confirm its security requirements.

Hermez

Hermez is devoted to obtaining full compatibility with DApps of Ethereum by exploiting

EVM's native command set. It enables an optimized implementation from existing tools on Ethereum with higher security.

However, some native EVM commands are reluctant to Zero Knowledge Proof. The team altered the tough codes to some intermediary codes and micro opcode, to express the same logic that could be accepted. This type of intermediary codes are customized and optimized; Zero Knowledge Proof is more likely to be generated in this case. Therefore, a balance is more likely to be achieved between maximizing Zero Knowledge Proof generation and optimizing EVM compatibility, a middle point between the two tracks.

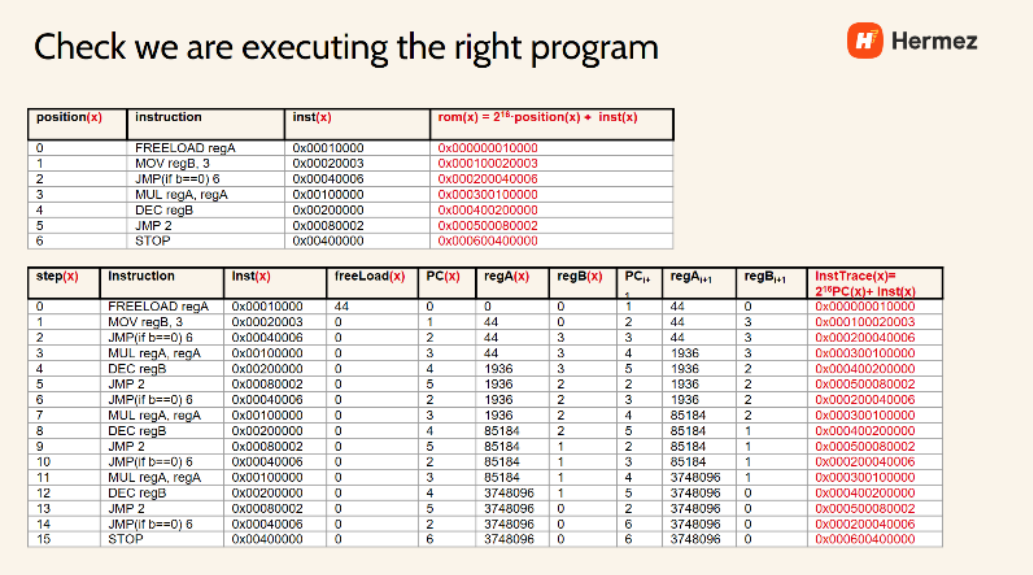
As native opcode of EVM requires executing environment, intermediary opcode is subject to a typical environment-uVM. UVM is composed by ROM and Main SM, and Main SM consists of various SM that could handle multiple functions.

Let's dive into how Hermez produces proofs according to the program logic. A program has to be executed step-by-step no matter how complicated the logic. For instance, extracting a number, completing a calculation, responding to a condition, jumping to another string, etc. requires the program to reiterate various execution of commands until an ending condition is encountered. Although the trace, which represents the pathway of execution and number of executions, could be random, the result would still be within a certain scope of possible outcomes. That is to say, even if the program cannot determine the specific route to go through, it will proceed on one of the predetermined routes all the same: No matter which route the program chooses, the

program execution will be considered complete as long as the actual execution coincides with the predetermined conditions.

Hermez stores these certain intermediary opcode in ROM, and configures responding codes according to different storage locations and commands characteristics, namely rom(x). During program execution, a real opcode will be generated according to each operation, namely instTrace(x). Plookup algorisms can be utilized in the process of verifying instTrace(x) is a subset of rom(x).

Figure 5: Hermez's method for determining the correct execution of a program



position(x)	instruction	inst(x)	rom(x) = 2^{16}·position(x) + inst(x)
0	FREELOAD regA	0x00010000	0x000000010000
1	MOV regB, 3	0x00020003	0x000100020003
2	JMP(if b==0) 6	0x00040006	0x000200040006
3	MUL regA, regA	0x00100000	0x000300100000
4	DEC regB	0x00200000	0x000400200000
5	JMP 2	0x00080002	0x000500080002
6	STOP	0x00400000	0x000600400000

step(x)	Instruction	Inst(x)	freeLoad(x)	PC(x)	regA(x)	regB(x)	PC_{i+1}	$regA_{i+1}$	$regB_{i+1}$	InstTrace(x)= 2^{16}PC(x)+ Inst(x)
0	FREELOAD regA	0x00010000	44	0	0	0	1	44	0	0x000000010000
1	MOV regB, 3	0x00020003	0	1	44	0	2	44	3	0x000100020003
2	JMP(if b==0) 6	0x00040006	0	2	44	3	3	44	3	0x000200040006
3	MUL regA, regA	0x00100000	0	3	44	3	4	1936	3	0x000300100000
4	DEC regB	0x00200000	0	4	1936	3	5	1936	2	0x000400200000
5	JMP 2	0x00080002	0	5	1936	2	2	1936	2	0x000500080002
6	JMP(if b==0) 6	0x00040006	0	2	1936	2	3	1936	2	0x000200040006
7	MUL regA, regA	0x00100000	0	3	1936	2	4	85184	2	0x000300100000
8	DEC regB	0x00200000	0	4	85184	2	5	85184	1	0x000400200000
9	JMP 2	0x00080002	0	5	85184	1	2	85184	1	0x000500080002
10	JMP(if b==0) 6	0x00040006	0	2	85184	1	3	85184	1	0x000200040006
11	MUL regA, regA	0x00100000	0	3	85184	1	4	3748096	1	0x000300100000
12	DEC regB	0x00200000	0	4	3748096	1	5	3748096	0	0x000400200000
13	JMP 2	0x00080002	0	5	3748096	0	2	3748096	0	0x000500080002
14	JMP(if b==0) 6	0x00040006	0	2	3748096	0	6	3748096	0	0x000200040006
15	STOP	0x00400000	0	6	3748096	0	6	3748096	0	0x000600400000

Source: Hermez, Huobi Research

One would be unable to identify a song if he or she cannot hear the melody that belongs to the particular song. The same thing applies when identifying a program: a program is determined to be falsely executed if codes being executed are not genetically part of the program.

One assumption goes: It would be irrational to consider a program is correctly executed if it follows certain paths but in misplaced order. From my perspective, it is controversial to conclude the status of a program execution merely by how it is executed. zkEVM needs, and only needs to verify the exact codes of smart contracts to be executed on Layer 2. Other possible glitches or the order of code execution should not be part of zkEVM verification; developers who write and deploy the smart contract should be concerned.

In terms of proving the consistency of storage, Hermez utilizes the proof of correlation for key-values. Compared to Applied ZKP, Hermez introduces Hash and Merkle Tree; a considerable amount of hash computations exist. However, the resource consumption for generating Zero Knowledge Proof by Hashing could be enormous, and Hermez has not released any official solutions for such cases. From our perspective, and consistent with the logic mentioned earlier this article, so long as the fixed logic is executed and verified, the degree of completion should not be of concern.

The next proofs from SM could be integrated as a whole to be sent and verified by the verifiers.

Last but not least, Hermez employs a large amount of polynomial promises: the proof being generated is zk-STARK instead of zk-SNARK. Zk-STARK constitutes a large proportion of storage, deviating from the principle that Rollup was born to minimize the amount of data submitted to Layer 1. As a result, Hermez proposed that a proof could

be synthesized for a typical proof: A STARK proof could be generated first, PLONK or Groth 16 could then come to play the role in synthesizing a shorter proof. It is commensurate with two separated compilations that directly reduce the consumption of the verifier and save unnecessary occupation of Layer 1 storage, making it more scalable.

AppliedZKP

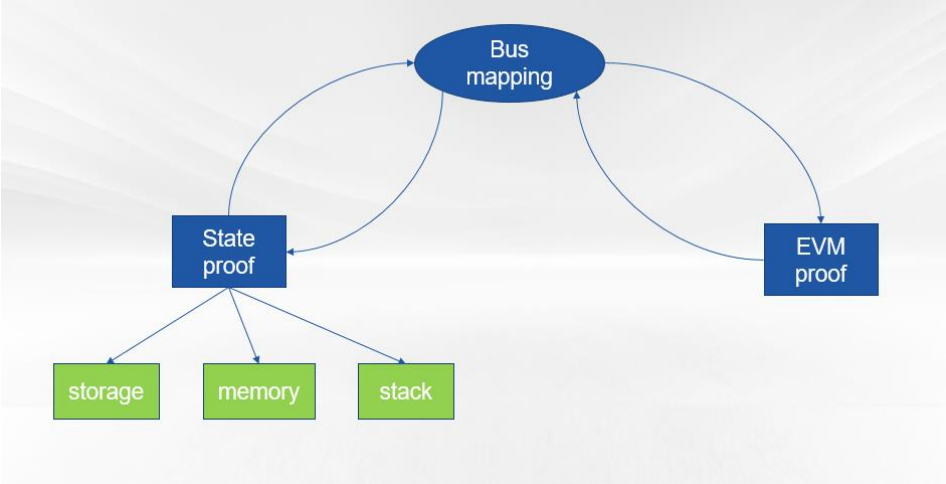
Apart from zkSync 2.0 and Hermez, the construction of the "overpass" cannot be whole without AppliedZKP. An interesting aspect of AppliedZKP: Bus Mapping.

The inherent logic of Bus Mapping concerns dealing with storage and computation independently. When corresponding data is correctly read by a group of codes, executed in a preset order and intervened with the account status, it would be considered an effective execution. Proof in this case can be categorized as "Status Proof" and "EVM Proof". "Status Proof" is the final outcome where operations of status/storage/stacks are correctly executed, while EVM confirms the correct codes are executed within a certain time range. With these two proofs in appearance, Ethereum mainnet would be capable of authenticating whether programs are being executed on Layer 2.

To be more specific, "Status Proof" must match the status of operations related to storage completed in the EVM. An EVM storage is composed of three parts: Storage,

Memory and Stack. “Status Proof” has to provide proof to each part respectively. Bus Mapping plays a role as a gateway, transporting data between computation module and storage module. “Status Proof” would be the one that informs the computation module that data transported in Bus Mapping is consistent with that in the storage module.

Figure 5: Bus Mapping



Source: Huobi Research Institute

The reflection of Bus Mapping includes two operations according to status. One is to read the old status, the other is to rewrite a new status. Storage status (applicable for Storage, Memory and Stack) is organized by sorts of key-values, a confirmation of receiving the correct data is equal to a confirmation that data transported by Bus Mapping match the data in storage status. Furthermore, pLookup algorithm would be employed to verify that the keys and values transported by Bus Mapping are inside the source being read; that keys and values match each other. Thus, pLookup finishes the verification process, confirming a subset relationship between two data sets.

pLookup first transforms the proof of bit operation to the verification that if input and output match the default setting in the lookup table, and then to the summary of whether the group of vectors is inclusive of another. By doing so, the number of Gate Constraints could be reduced, therefore efficiency level increases.

Proof of EVM needs to confirm every single operation in the program, including calculation (plus, minus, multiplication and division), logic operation (“and”, “or” and “not”), program redirect (call), etc. Each step requires undergoing an entire process of realizing opcode, defining constraints and EVM execution result, confirming execution steps in order to complete the EVM circuit. While it is common to generate proofs for mathematical and logical calculations, it gets more complicated for program redirection. More details are upcoming.

Summary and Expectations

For Ethereum, the well-known “highway”, ZK Rollup has increased total transaction volume by over 500 times, boosting TPS to 2000, which is on the par with the current VISA payment system. After vast development of zkEVM, ZK Rollup will be capable of handling more cases and providing comprehensive support for various applications, cementing its position as a pioneer in the Rollup market.

Traffic pressure on Ethereum could be enormously alleviated thanks to faster transportation of data. Firstly, this affords Ethereum a better chance amidst the fierce competition of newly emerged public chains, such as Solana, Avalanche, Fantom, etc. As a result, these public chains will lose comparative advantage in terms of performance and will have to differentiate themselves from each other and employ more innovative strategies in order to attract users and grow their ecosystem. Secondly, more newly launched projects could be deployed on Ethereum with a lower cost and still benefit from the mature and powerful Ethereum ecosystem. In all, zkEVM is the key. We believe that with proper development efforts, the above will come to pass. Perhaps, the discussion should not be about “whether zkEVM will come”, but “when zkEVM will come”.

zkEVM does not indicate the finish line for Rollup. The future of Rollup could extend beyond that of a temporary transition. According to Vitalik, for Ethereum, Rollups are

more than likely to be the sole, trustless, scalable solution in the short run, or even long term. The volume of Rollup could be optimized by the reduction of gas fees for the calldata part in the block, cutting cost down by more than five times,which is less than 1% of that in Layer 1; when Sharding is successful, the scalability of Rollup could be amplified exponentially resulting in a transaction fee that is negligibly low.

Rollup could also feature in public chains apart from Ethereum. A newly developed district would be sparsely populated because very few can perceive and embrace the advanced philosophy of design; traffic jams might not be an issue at the start. The same applies for public chains. However, traffic may eventually accumulate to a level that may cause jams, and an “overpass” would be necessary sooner or later to ease the situation. Rollup remains indispensable considering possible consequences that may appear in the future, and zkEVM will play a vital role in the challenges yet to come.

Reference：

1. https://blog.csdn.net/PPIO_Official/article/details/102942680
2. https://medium.com/degate/an-article-to-understand-zkevm-the-key-to-ethereum-scaling-ff0d83c417cc
3. https://zksync.io/zkevm/
4. https://medium.com/matter-labs/zksync-2-0-hello-ethereum-ca48588de179
5. https://www.youtube.com/watch?v=6wLSkpIHXM8
6. https://www.ftftx.com/hot/40269.html
7. https://blog.hermez.io/introducing-hermez-zkevm/
8. https://www.youtube.com/watch?v=17d5DG6L2nw
9. https://medium.com/@sin7y/exploring-popular-zkevm-solutions-appliedzkp-matter-labs-hermez-and-sin7y-d17deb1f8808
10. https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf
11. https://starli.medium.com/zkp-zkevm-a9b046789b4e
12. https://hackernoon.com/appliedzkp-zkevm-circuit-code-a-guide

About Huobi Research Institute

Huobi Blockchain Application Research Institute (referred to as "Huobi Research Institute") was established in April 2016. Since March 2018, it has been committed to comprehensively expanding the research and exploration of various fields of blockchain. As the research object, the research goal is to accelerate the research and development of blockchain technology, promote the application of blockchain industry, and promote the ecological optimization of the blockchain industry. The main research content includes industry trends, technology paths, application innovations in the blockchain field, Model exploration, etc. Based on the principles of public welfare, rigor and innovation, Huobi Research Institute will carry out extensive and in-depth cooperation with governments, enterprises, universities and other institutions through various forms to build a research platform covering the complete industrial chain of the blockchain. Industry professionals provide a solid theoretical basis and trend judgments to promote the healthy and sustainable development of the entire blockchain industry.

Consulting email:

research@huobi.com

Twitter: @Huobi_Research

https://twitter.com/Huobi_Research

Medium: Huobi Research

https://medium.com/huobi-research

Disclaimer

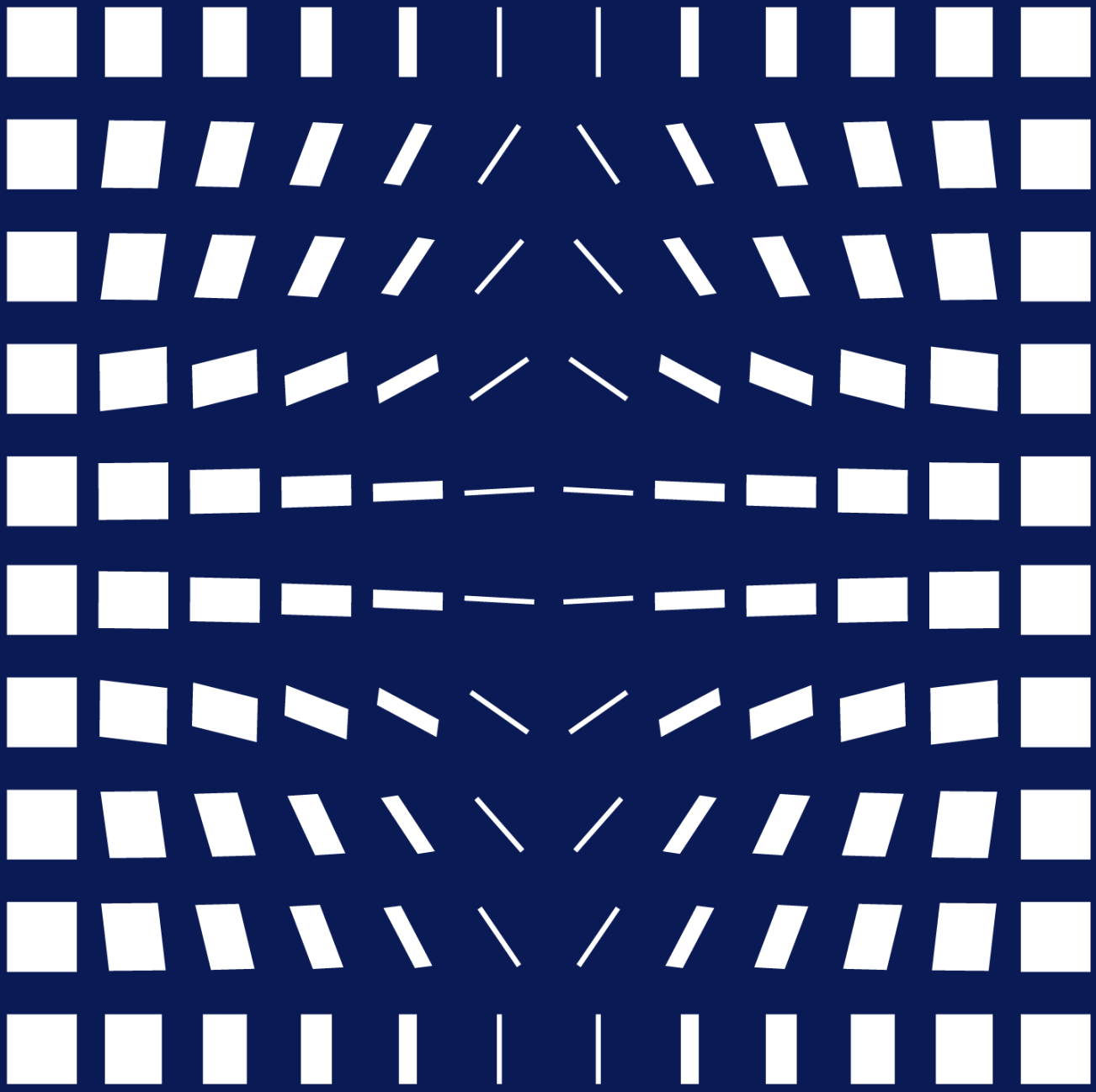
1) The author of this report and his organization do not have any relationship that affects the objectivity, independence, and fairness of the report with other third parties involved in this report.

2) The information and data cited in this report are from compliance channels. The sources of the information and data are considered reliable by the author, and necessary verification have been made for their authenticity, accuracy and completeness, but the author makes no guarantee for their authenticity, accuracy or completeness.

3) The content of the report is for reference only, and the facts and opinions in the report do not constitute business, investment and other related recommendations. The author does not assume any responsibility for the losses caused by the use of the contents of this report, unless clearly stipulated by laws and regulations. Readers should not only make business and investment decisions based on this report, nor should they lose their ability to make independent judgments based on this report.

4) The information, opinions and inferences contained in this report only reflect the judgments of the researchers on the date of finalizing this report. In the future, based on industry changes and data and information updates, there is the possibility of updates of opinions and judgments.

5)



Huobi Research