Translated version of this post :

[https://scrapbox.io/ywzx/ETH_Global_Istanbul_:_Prize_list

](https://scrapbox.io/ywzx/ETH_Global_Istanbul_:_Prize_list)

Thank you dmarz for your support during hackathon and best team withvita and Tomo !!

This is our product that we have submitted and details and we will keep building this.

- PEPC-REP | ETHGlobal
- https://twitter.com/keccak254/status/1726770626973536645

Also, two ideas came up in discussions with the team during this hackathon: Relayer inclution list (and paralell auction) and UniswapX Direct fill on SUAVE.

Please feel free to DM us if you are interested.

Introduction

I have attended ETH Global Istanbul!

I wrote a summary of the Flashbot x Uniswap prize winners and comments for those products.

All the projects were unique and full of tips on how to make the most of SUAVE!

We also gained great insight into SUAVE by actually developing with SUAVE.

We wrote about why Flashbot developed SUAVE, SUAVE use cases, SUAVE suggestions, questions and issues, and more.

I would be happy to discuss the insights with the community!

Prize Winner

Uniswap x Flashbot Prize details

- Best Intent UniSUAVE Design and PoC —— $2,000 : Up to 2 teams will receive $1,000
- Best PFOF Design and PoC —— $1,500
- Best UniSUAVE LVR Design and PoC —— $1,500 : ① $800 ② $500 ③ $200
- Best Use of SUAVE —— $2,500> : ① $1,500 ② $750 ③ $250
- Best Use of UniV4 —— $2,500> : ① $1,500 ② $750 ③ $250

Blob Manager

[Blob Merger | ETHGlobal

*Introducing a solution to optimize blob data usage on Ethereum, aligned with EIP-4844. As blobs maintain a fixed size…*ethglobal.com](https://ethglobal.com/showcase/blob-merger-k7m1f)

- Prize : SUAVE 1st place
- Overview
- A single blob can send up to 128KB of data, but even if you use a portion of the blob(less than 128KB of data), you must pay the same fee as for 128KB(a single blob). This is an unnecessary cost for low activity rollups. In addition, since the number of blobs per block is limited, if many rollups emerge in ethereum, rollups have to wait until the data is stored in the blob. Therefore, by allowing blobs of less than 128KB to be submitted together as a single blob, it contributes to the reduction of unnecessary costs and the scaling of rollups.
- The system consists of two components. (1) Registration of blobs (2) Communication management between rollup and builder
- ① is done in this code but I couldn't figure out ②.

- From the commit log, it looks like only blob merge is implemented.
- ① is done in this code but I couldn't figure out ②.
- From the commit log, it looks like only blob merge is implemented.
- A single blob can send up to 128KB of data, but even if you use a portion of the blob(less than 128KB of data), you must pay the same fee as for 128KB(a single blob). This is an unnecessary cost for low activity rollups. In addition, since the number of blobs per block is limited, if many rollups emerge in ethereum, rollups have to wait until the data is stored in the blob. Therefore, by allowing blobs of less than 128KB to be submitted together as a single blob, it contributes to the reduction of unnecessary costs and the scaling of rollups.
- The system consists of two components. (1) Registration of blobs (2) Communication management between rollup and builder
- ① is done in this code but I couldn't figure out ②.
- From the commit log, it looks like only blob merge is implemented.
- ① is done in this code but I couldn't figure out ②.
- From the commit log, it looks like only blob merge is implemented.
- Comments
- This is awsome idea!! I think that using SUAVE like Shared Sequencer is the one that Flashbot was looking for as a usecase of SUAVE.
- I was wondering if the blobs for each rollup could really be combined, and how to match the time difference of block building for each rollup… but I think those problems will be solved eventually.
- This is awsome idea!! I think that using SUAVE like Shared Sequencer is the one that Flashbot was looking for as a usecase of SUAVE.
- I was wondering if the blobs for each rollup could really be combined, and how to match the time difference of block building for each rollup… but I think those problems will be solved eventually.

Uni-Suave intents

[Uni-Suave intents | ETHGlobal

*Intent-solver PoC using Suave. Users submit swap intents, solvers submit solutions. Sensitive data is stored…*ethglobal.com](https://ethglobal.com/showcase/uni-suave-intents-5t2t3)

- Prize: Best Intent UniSUAVE Design and PoC, SUAVE 2nd place
- Overview
- Swapper submits swap intents, order information is public and signatures are stored in a secret area. Solver submits solution tx based on the public information, and the solution is auctioned based on egp.
- The solution will be submitted to builder when the next block's auction starts.
- suave-geth with custom precompile to get goerli's blocknumber to allow auctions within a block.
- Swapper submits swap intents, order information is public and signatures are stored in a secret area. Solver submits solution tx based on the public information, and the solution is auctioned based on egp.
- The solution will be submitted to builder when the next block's auction starts.
- suave-geth with custom precompile to get goerli's blocknumber to allow auctions within a block.
- Comments
- The idea itself was simple, but the architecture and code was super clean.
- Especially the part where the tx of the previous action is sent with the current action was great.
- Especially the part where the tx of the previous action is sent with the current action was great.
- I didn't know that submitBundleJsonRPC() can throw a bundle to the builder.

- This could be used in UniswapX and other intent based applications. I t seems that SUAVE is like an MEV-share with extensibility.
- This could be used in UniswapX and other intent based applications. I t seems that SUAVE is like an MEV-share with extensibility.
- The idea itself was simple, but the architecture and code was super clean.
- Especially the part where the tx of the previous action is sent with the current action was great.
- Especially the part where the tx of the previous action is sent with the current action was great.
- I didn't know that submitBundleJsonRPC() can throw a bundle to the builder.
- This could be used in UniswapX and other intent based applications. I t seems that SUAVE is like an MEV-share with extensibility.
- This could be used in UniswapX and other intent based applications. I t seems that SUAVE is like an MEV-share with extensibility.

SUClave

[SUClave | ETHGlobal

*SUClave: A New AMM Design Using SUAVE and Uniswap V4 Arbitrage bots are doing MEV attacks through the Tap of Pool…*ethglobal.com](https://ethglobal.com/showcase/suclave-dwh4f)

https://twitter.com/doganeth_en/status/1726303354857689181

- Prize: LVR 1st place, SUAVE 3rd place
- Overview
- Most of the arbitrage revenue goes only to the botter (MEV-Boost actors), but this is an effort to return those revenues to the LPs as well.
- Conduct ToB auctions on SUAVE, create signatures for the winners to access ToB, and return the backrun proceeds to LPs and botters.
- Most of the arbitrage revenue goes only to the botter (MEV-Boost actors), but this is an effort to return those revenues to the LPs as well.
- Conduct ToB auctions on SUAVE, create signatures for the winners to access ToB, and return the backrun proceeds to LPs and botters.
- Comments
- Similar to McAMM concept, how to implement the auction is the challenging part.
- Similar to McAMM concept, how to implement the auction is the challenging part.

PEPC-REP

[PEPC-REP | ETHGlobal

*The project will design a more flexible and diverse communication market where the proposer exerts preference in block…*ethglobal.com](https://ethglobal.com/showcase/pepc-rep-dytdv)

https://twitter.com/keccak254/status/1726770626973536645

- Prize: LVR 2nd place
- Overview
- PEPC is built on SUAVE.
- Use cases include building McAMM and direct fill of UniswapX.
- PEPC is built on SUAVE.
- Use cases include building McAMM and direct fill of UniswapX.

- Comments
- This is our product and we will keep building on it.
- This is our product and we will keep building on it.

Toxic Fl-no

[Toxic Fl-no | ETHGlobal

*Reward users who are value adders to the Ethereum ecosystem by evaluating user behavior inside a Suave contract and…*ethglobal.com](https://ethglobal.com/showcase/toxic-fl-no-qsog4)

- Prize: LVR 3rd place
- Overview
- Addresses with valuable on-chain behavior can reduce their fee in UniswapV4.
- Specifically, it allows Axiom to prove off-chain information with SUAVE, and to pass only SUAVE-generated swaps through V4 hooks.
- It doesn’t seem to be fully complete yet, so I’m looking forward to further implementation.
- It doesn’t seem to be fully complete yet, so I’m looking forward to further implementation.
- Addresses with valuable on-chain behavior can reduce their fee in UniswapV4.
- Specifically, it allows Axiom to prove off-chain information with SUAVE, and to pass only SUAVE-generated swaps through V4 hooks.
- It doesn’t seem to be fully complete yet, so I’m looking forward to further implementation.
- It doesn’t seem to be fully complete yet, so I’m looking forward to further implementation.
- Comments
- Using SUAVE as an execution environment for an intent based application seems to be a good idea. I think this is one of the main usecase of SUAVE.
- Using SUAVE as an execution environment for an intent based application seems to be a good idea. I think this is one of the main usecase of SUAVE.

Aggeragator_Hook

[Aggeragator_Hook | ETHGlobal

*Introducing the Aggregate_Hook: a hook for DEXs, seamlessly integrating with Uniswap for efficient liquidity…*ethglobal.com](https://ethglobal.com/showcase/aggeragator-hook-rfy6g)

- Prize: UniswapV4 1st place
- Overview
- Monitor LP information of other DEXs in hooks, and add/remove liquidity before swap if necessary, so that swap can be done at a reasonable price.
- Monitor LP information of other DEXs in hooks, and add/remove liquidity before swap if necessary, so that swap can be done at a reasonable price.
- Comments
- Simple but I hadn’t thought of it! I need more time to think about the extendability of hooks.
- I think that the positional relationship of “aggregator → DEXs including Uniswap” can be changed to “Uniswap V4 → DEXs including Uniswap” by hooks, and I think that Uniswap wants to take away the touch point with Swapper by V4 from aggregator.
- Simple but I hadn’t thought of it! I need more time to think about the extendability of hooks.
- I think that the positional relationship of “aggregator → DEXs including Uniswap” can be changed to “Uniswap V4 →

DEXs including Uniswap” by hooks, and I think that Uniswap wants to take away the touch point with Swapper by V4 from aggregator.

DeFi Omni Dapp

[DeFi Omni Dapp | ETHGlobal

A single app to interact with all popular DeFi protocols from once place

ethglobal.com](https://ethglobal.com/showcase/defi-omni-dapp-z8sd3)

- Prize: UniswapV4 2nd Place
- Overview
- UniswapV4 can be used in ShellProtocol.
- UniswapV4 can be used in ShellProtocol.
- Comments
- I didn’t know about ShellProtocol.
- I felt that understanding DeFi is also essential in terms of block building and MEV.
- I didn’t know about ShellProtocol.
- I felt that understanding DeFi is also essential in terms of block building and MEV.

FastSwaps

[FastSwaps | ETHGlobal

*First mobile cross-chain exchange. Our platform empowers users to send tokens effortlessly from one network to another…*ethglobal.com](https://ethglobal.com/showcase/fastswaps-izqiq)

- Prize: UniswapV4 3rd Place
- Overview
- Building a swap without the need for a wallet.
- Building a swap without the need for a wallet.

Insights that I gained through this hackathon

I gained a variety of insights through this hackathon.

I will write about UniswapV4 and SUAVE insights, Suggestions to SUAVE, Questions & future issues of SUAVE.

UniswapV4

- It was good to have a better understanding of the scalability and role of UniswapV4.
- For more information: UniswapV4 - ywzx.
- However, I have a feeling that UniswapV4 is not something that crypto developers can handle. I felt I need to have deepen knowledge of traditional finance.

SUAVE

Through this hackathon, I gained a deeper understanding of “(1) Why Flashbot developed SUAVE

” and “(2) The usecase of SUAVE

”.

① Why Flashbot developed SUAVE

Despite having established a leading position in the industry by MEV-Boost, they are trying to develop SUAVE, which breaks that position.

I think that the main reason is that MEV's revenues are sure to decline, they predict a future where MEV-boost will no longer be used.

There are two reasons for the decline in MEV revenues.

First reason is that DEX protocols are trying to eliminate MEV.

DEX protocols has been updated to reduce price fluctuation during a swap,

also Intend based applications which prevents MEVs are emerging.

Second, the overall Ethereum execution environment is expected to sift to L2 from L1, so L1 transaction will be decrease.

If MEV revenues decrease, there is a possibility that MEV-Boost will not be used as there will be fewer MEV-Boost participants.

Therefore, Flashbot is trying to develop products that are related to a wider range of "block building" from those that were specific to "MEV".

In other words, if SUAVE can be used to build blocks for products that do not involve MEV, there is a great possibility that SUAVE will be used even if MEV-boost is no longer used.

Of course, there is a reason why SUAVE was created simply because the current MEV-boost has some problems, but this was known to us, so I just skipped it.

② The usecase of SUAVE

Prior to participating in the hackathon, I had considered SUAVE as a highly programable MEV-Share or OFA (Order flow auction) platform.

However, through implementation examples and actual development, I felt that SUAVE could be used in many areas such as shared sequencers, data availability, trust outsourcing, and execution environment for intent based applications

.

Shared sequencer

By implementing precompile, you can fetch L1 information in the MEVM contract, making the connection between L1 and SUAVE stronger and enabling shared sequencer-like behavior.

Uni-Suave intents is a good example of Shared Sequencer because it send an intent to a builder by looking at the blocknumber of L1.

Similar services: Celestia, Espresso

Data availability

The blob one is a good example of a DA that could be more powerful by adding a shared sequencer function.

Similar services: Celestia, Espresso

Outsourcing trust

The block construction so far needs to trust that the builder will not unbundle. It is "trust me bro".By substituting some or all of the builder's activities for SUAVE, trust can be outsourced (replaced) to SUAVE.

SUAVE also requires a certain level of trust, but it is more reliable than a single builder.

Similar services: MEV-Share, HyperOracle

Intent based Application execution environment

Not only at the builder, but also at the application layer.

Especially since the current Intent based application requires a certain degree of trust, replacing the Intent execution environment with SUAVE will allow Intent to be built in a more transparent and scalable.

Similar services: Anoma, Essential

Suggestion to SUAVE

I felt it would be much more extensible as “interoperability” if we could create transaction on SUAVE, something like ICP.

Questions & future issues

① Will it be actually used in block construction in L1 and L2s?

As for L1, SUAVE is useless unless it can win the block auction of the existing MEV-Boost, but it seems difficult for SUAVE to win the block-auction due to latency issues.

In L2, it is important to show merits for choosing SUAVE among the many competitors and for migrating the block construction environment to SUAVE, but there does not seem to be a decisive factor that L2s used SUAVE.

② The SUAVE chain itself is almost identical to the private chain, but since it runs on TEE, no one seems to care about that.

I would like to know a detailed analysis of how much we have to trust with SAUAVE (up to what economic scale it can withstand), etc.

③(Simply looking for reasons for my thoughts) There are Shared sequencer and DA products such as Celestia and Espresso, but why SUAVE attracts me more?

Previous achievements so far (MEV-geth, MEV-Boost, etc.)? Is it because I sense a more fundamentalist Ethereum vibe from Flashbot, or is it the strength of the community around MEV?