

We'll be using the forum to keep an ongoing record of the changes we're shipping.

We'll begin with the first two weeks of this year, and post similar updates every ~2 weeks from now.

Jan 1 - 19

suave-geth

Initial release: v0.1.0

Installing suave-geth

is now as easy as:

```
$ curl -L https://suaveup.flashbots.net | bash $ suave-geth --suave.dev
```

This change also goes along with deprecating the current Docker setup.

suave-std

[suave-std

](<https://github.com/flashbots/suave-std>) is a collection of Solidity contracts and libraries to help you create Suapps.

This will likely become the main entrypoint for developers who want to build SUAPPs.

Retiring SuaveForge.sol

[SuaveForge.sol](#) was a helper contract we initially used to assist with running smart contracts intended for SUAVE in Forge. With the extension of capabilities now introduced by suave-std

, this is no longer necessary or optimal.

Forge Cheat Codes

[This PR](#) adds the confidentialInputs

precompile as a "cheat code" in Forge, meaning you can both mock and reset inputs which are intended to be confidential in both contracts and their tests.

[This follow-up PR](#) adds the support required for resetting the confidential store to enable cleaner, better tests.

New MevShare Library

One of the core examples we use at this early stage to illustrate what SUAVE can do is an implementation of the current MEV-Share protocol in a SUAVE smart contract.

[This PR](#) improves how we implement that example and makes it more realistic.

eth_sendBundle Support

[This PR](#) adds support for the [eth_sendBundle RPC supported by the Flashbots Builder](#) and, once again, demonstrates how services which currently rely on trusted third parties can be implemented as smart contracts on SUAVE.

[WIP] EIP-1559 Transaction Support

Our [first community contribution](#) in the suave-std

repo!

You can find another great community contribution in the [SUAVE playground](#).

suapp-examples

With the implementation of suave-std

, our SUAPP examples repo has also been overhauled.

Change of submodules

It was once necessary to include suave-geth

as a submodule in this repo in order to get everything working, but we can now just [use the much lighter suave-std repo](<https://github.com/flashbots/suapp-examples/pull/18>), making life easier, better, and quicker for everyone.

DataRecords, not Bids

The above PR also implements some of the last changes we shipped in suave-geth

in 2023, changing the name of a low-level data object from Bid

(which was causing a lot of confusion) to DataRecord

, which is a bit more wordy, but also more clear.

[WIP] UniSUAPP

Brock is working on an [implementation of Uniswap V2 on SUAVE](#). We'll discuss further details in the next update once this is finished and merged.

Docker setup

It's [now possible to run a Docker setup](#) when working with the examples that will start everything you need for more complex SUAPP development. Our initial docker setup in suave-geth

only started the SUAVE chain, and left some of the other pieces off by default. This new Docker file fixes that.

Public network support

A number of early testers and tinkerers have run into issues trying to build SUAPPs which depend on data from other chains, or post data to other chains, or do some mix of both.

[This PR](#) shifts a lot of the configuration details in env variables and makes the changes necessary to enable anyone to more easily set up their work, especially if it depends on multiple chains.

Miscellaneous

Precompile docs automation

We now [update our list of available precompiles](#) automatically whenever a new one is merged so that it is always up-to-date.