

CHANGELOG / SUAVE Dev Updates

a.k.a Simple Updates About Virtually Everything

We're regularly posting SUAVE updates to the community. The format is open to experiments.

Past editions:

- [#1 - 2024-01-18](#)
- [#2 - 2024-02-15](#)
- [#3 - 2024-03-25](#)

Progress on SUAVE has been swift and inspiring. There are a tonne of new features based on all the feedback we got from a team gathering in Vienna and the EthGlobal London hackathon.

suave-geth

New version released: 0.1.5

The focus of this release is primarily Developer Experience. It includes much better support for the spell

command, which will help you deploy contracts and send CCRs, as well as better error messages, and various lower-level fixes to make creating cool SUAPPs much smoother.

- Feat: [rename dns registry to service alias by jinmel · Pull Request #223 · flashbots/suave-geth · GitHub](#)
- Feat: [Return error if the CCR was done to an empty contract by ferranbt · Pull Request #227 · flashbots/suave-geth · GitHub.](#)
- Feat: [Return valid error from forgecmd calls by ferranbt · Pull Request #228 · flashbots/suave-geth · GitHub](#)
- Feat: [Add confidential inputs to spell cmd by ferranbt · Pull Request #230 · flashbots/suave-geth · GitHub](#)
- Feat: [Embed the precompile error in the error message returned in the ccr by ferranbt · Pull Request #231 · flashbots/suave-geth · GitHub.](#)
- Bug: [Do not error on state change for transient objects by ferranbt · Pull Request #229 · flashbots/suave-geth · GitHub](#)
- Bug: [Fix forgecmd read empty config from forge.toml by ferranbt · Pull Request #226 · flashbots/suave-geth · GitHub](#)

suave-std

We've formalised what SUAVE-STD is: a library of essential patterns for building SUAPPs. As such, we removed many of the e2e examples, which now live in their own repos.

These examples include [Unisuapp](#), a [Next starter template](#), and a [Vite starter template](#).

We've added a few important patterns to our essentials: especially the Gateway.sol contract:

- Feat: [Add Gateway contract to make calls to external contracts by ferranbt · Pull Request #72 · flashbots/suave-std · GitHub.](#)
- Feat: [Add jsonrpc call\(\) method by ferranbt · Pull Request #71 · flashbots/suave-std · GitHub](#)
- Feat: [Show precompile error message by ferranbt · Pull Request #70 · flashbots/suave-std · GitHub](#)
- Feat: [Add tool to validate the correctness of the Solidity code snippets by ferranbt · Pull Request #68 · flashbots/suave-std · GitHub.](#)
- Bug: [Fix suavelib import path in some protocols by ferranbt · Pull Request #69 · flashbots/suave-std · GitHub](#)
- Bug: [Fix to allow address\(this\) in peekers for forge confstore by ferranbt · Pull Request #73 · flashbots/suave-std · GitHub.](#)

suapp-examples

We continue to extend these to illustrate the various different ways you can leverage the essential patterns illustrated in

SUAVE-STD to build novel and powerful Suapps.

- New: [Add example of a suapp storing and using a private key by ferranbt · Pull Request #48 · flashbots/suapp-examples · GitHub](#).
- New: [Example using a private key generated from the Suapp by ferranbt · Pull Request #50 · flashbots/suapp-examples · GitHub](#).
- New: [Add example of suave-std/Gateway.sol by ferranbt · Pull Request #49 · flashbots/suapp-examples · GitHub](#)
- New: [brock/nftee by zeroXbrock · Pull Request #52 · flashbots/suapp-examples · GitHub](#)

suave-execution-geth

A quick refresher: this is a minimal fork of Geth which gives you access to the SUAVE Execution Namespace. It's useful for running e2e multi-chain examples, for instance: trying to build blocks on other chains.

- Feat: [Merge upstream by ferranbt · Pull Request #30 · flashbots/suave-execution-geth · GitHub](#)

suave-docs

- Full overhaul of all [tutorials](#)
- Local first
- Using spell
- Begin with basic programming model and the thing most builders have struggled with so far: the onchain-offchain dynamic. Extend this step by logical step through all the different primitives required for building Suapps.
- 3-layered approach. suave-std

is abstracted, essential patterns. The docs illustrate core patterns there in fleshed out examples with narrative to explain “why” and make them stick. We introduce a “Community Directory” to illustrate the many divergent ways in which the bottom two layers can be applied.

- Local first
- Using spell
- Begin with basic programming model and the thing most builders have struggled with so far: the onchain-offchain dynamic. Extend this step by logical step through all the different primitives required for building Suapps.
- 3-layered approach. suave-std

is abstracted, essential patterns. The docs illustrate core patterns there in fleshed out examples with narrative to explain “why” and make them stick. We introduce a “Community Directory” to illustrate the many divergent ways in which the bottom two layers can be applied.

- New [Community Directory](#), showcasing the work of everyone who has contributed to SUAVE so far.

suavex-foundry

- Wee also maintain a fork of Foundry called [suavex-foundry](#).
- Brock demos this in both the build-eth-block

example in Suapp Examples, and in Unisuapp. You can read more about it in [this post](#).

Twitter

- https://twitter.com/suave_bot. We finally have a twitter presence

! Appropriately, it is a bot. * It's purpose is to repost new forum threads, so post on our forum to get CT excited about your work. The bot may also pass you butter.

- It's purpose is to repost new forum threads, so post on our forum to get CT excited about your work. The bot may also pass you butter.