@ferranbt just recently posted a PR to suave-geth that replaces the signature scheme used for signing CCRs with an EIP712 signature scheme. This allows us to sign messages from EIP-1193-based wallets (e.g. M*tamask) without having to rely on the user's wallet RPC endpoint being set to a suave RPC.

I hacked up an implementation to show that it works (web app code here, requires this branch of suave-viem), but there are a couple caveats that we need to be aware of:

1. We cannot use the chainId

parameter in the EIP712 domain. This is because M*tamask (and presumably other wallets) will validate the chainId in the message against the chainId of the connected RPC. But we can simply omit that parameter, which sidesteps the error, but potentially introduces a replay-attack vector. I've gone ahead and done this here, but I'm still uncertain whether it's replay-safe. To prevent replay attacks, we may be able to set the verifyingContract

parameter to a constant in suave-geth, but I haven't tested this yet.

1. While we use the window.ethereum

provider to sign the EIP712 message with the user's wallet, SUAPP developers need to use a separate web3 provider to connect to the SUAVE RPC. This is how we send the EIP712 signature given to us by the wallet, to SUAVE. We'll also use it to query SUAVE chain for state updates like logs, CCR receipts, etc.

With these minor changes, SUAPPS can get signatures from users and send CCRs without the user ever having to connect their wallet to the SUAVE RPC. This makes for a much smoother UX.

Here's a demo ("localhost" in m*tamask is connected to my local suave-geth devnet):

[

](https://www.youtube.com/watch?v=pUV2j3GVcLQ)

You can see that we switch to mainnet, sign an EIP712 message (and send it in a CCR behind the scenes), then switch to localhost and do the same thing, and both work the same. The SUAPP shown here is simple – it just stores a number read from confidentialInputs

. Code here.

Open questions:

- is removing chainId

from the EIP712 message domain replay-safe in this context?

- can a suave-geth-defined verifyingContract

constant be used in place of chainId

?

Further reading:

- Proposal: Implementing EIP-712 for Confidential Compute Requests