

I am posting this on behalf of [@fiii](#) who is on vacation right now

## Team Information

- Team Members:

Alejo

## Project Repository

- GitHub Link:

[GitHub - fiii/sec](#)

## Project Goal

- Project Name:

## **S.E.C.** (Salary Expectations Checker)

- Brief Description:

SUAPP to align salary expectations without either party (employers and candidates) revealing their number.

- What is the primary goal of your project?

I wanted to familiarize myself with the SUAVE platform, get a temperature check on the state of its development, and produce a simple SUAPP with a realistic use case.

## Challenges

- What were the major challenges your team encountered during the hackathon and how did you overcome them?

backend

the backend part was relatively straightforward. the mossify-prod

branch of suave-geth allows for pushing to confidential store all storage changes enacted by the MEVM. this means that minimal architecture changes are needed to bring an app from “vanilla” to SUAPP, namely dealing with the lack of return values (which are now used for the callbacks). the mossify

branch is experimental work by Ferran, who provided guidance and pointed me to a minor change required for this to actually work (which is the delta to mossify-prod

).

frontend

the devex for the frontend was pretty terrible. part of this is frontend, part of this is ethereum, part of this is SUAVE, and part of this might be this specific branch of suave-geth. I tried to use the suave-viem library recommended in the SUAVE docs, but even with Brock’s help I couldn’t even land a transaction (a CCR actually). I ended up using a salad of libraries and doing a very ugly hack with the help of Miha using his library suave-ethers. not super clear to me to what extent the mossify branch breaks the suave-viem library, but my sense is that it shouldn’t, because CCRs work transparently.

## Current State

- Project Status at the End of Hackathon:
- What features are complete?

the SUAPP is complete with an MVP frontend and runs fine in a local environment.

- What is still a work in progress?

I couldn’t deploy to Rigil because I need a suave-geth node running the mossify branch connected to the network. this is on devops’ TODO.

- Are there any known bugs or issues?

no.

- What features are complete?

the SUAPP is complete with an MVP frontend and runs fine in a local environment.

- What is still a work in progress?

I couldn't deploy to Rigil because I need a suave-geth node running the mossify branch connected to the network. this is on devops' TODO.

- Are there any known bugs or issues?

no.

#### How to Run

- Installation Instructions:
- Prerequisites (e.g., software versions, dependencies):

install suave-geth and compile the mossify-prod branch.

- Step-by-step installation and setup guide:

see repo's README.md

.

- Prerequisites (e.g., software versions, dependencies):

install suave-geth and compile the mossify-prod branch.

- Step-by-step installation and setup guide:

see repo's README.md

.

#### Additional Notes

- Future plans for the project:

#### Deploy on Rigil!

- Acknowledgments:

thanks to Ferran, Brock, and Miha for all the help!

Extra: SUAPP Screenshot!

[

Screenshot 2024-06-17 at 11.17.09 PM

1564×1878 212 KB

](<https://collective.flashbots.net/uploads/default/original/2X/1/1ac65f701870b7e175f0f0202e8f9ecf963aad3c.png>)