Blockchain Project

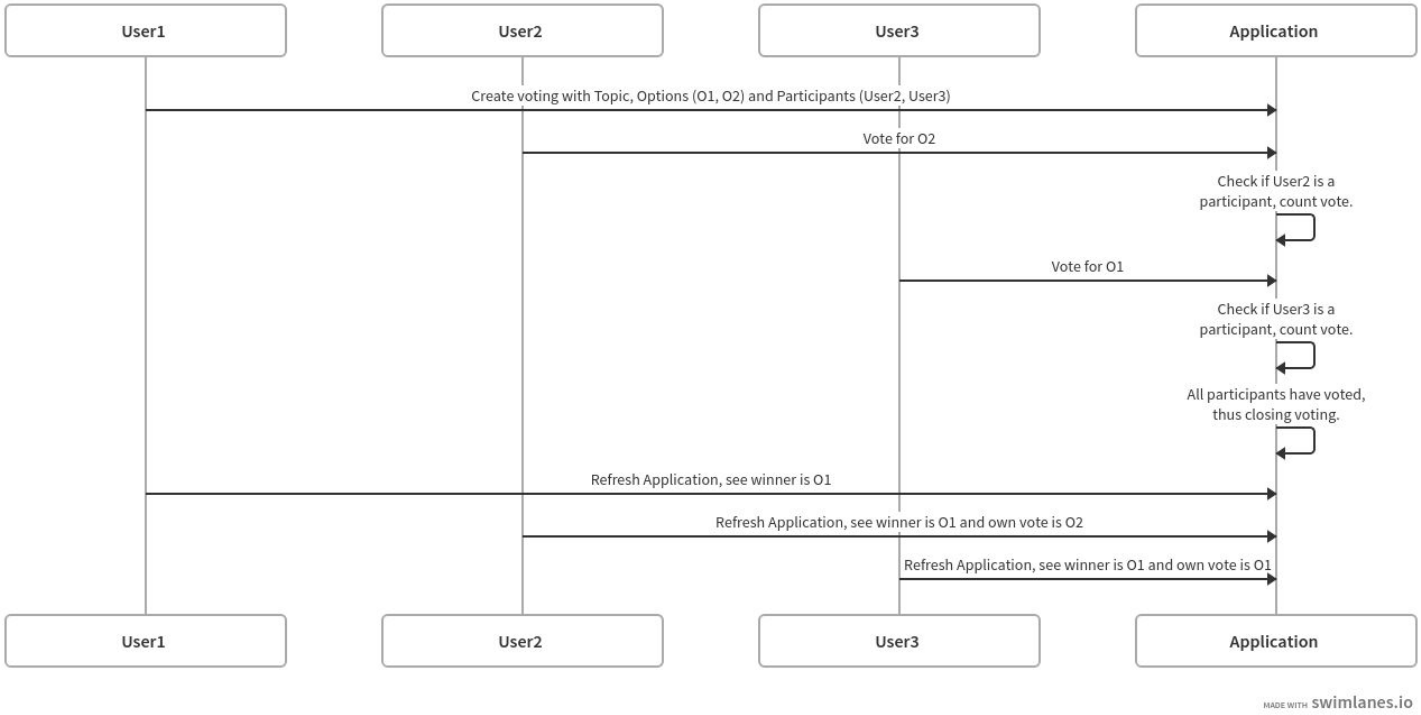
Laura Thoma, Mona Panchaud,
Stefanie Jäger

Use Cases

Our goal was to develop a simple voting application, with a web frontend and smart contract backend.
Key Features:

- Any user can start a new voting, with a topic, options, and a whitelist of participants.
- All whitelisted participants can cast a vote for one of the options.
- The voting automatically ends once every participant on the whitelist has submitted their vote.
- The winning option is the one that receives the highest number of votes.
- In the event of a tie, the option listed first during creation will be declared the winner.

Voting process



Ideas we didn’t implement

We had some ideas, which we did not end up implementing:

- The ability for the creator to end a voting at any time.
- Handle events for participants and creator on the start or end of a voting.
- Allowing a user to log in as an admin or a participant, with a different UI.
- Skip Login, get address from Metamask instead.
- Showing the current vote count next to each option for all votings.

These ideas were dropped due to multiple reasons:

- Realizing that they are not necessary.
- Too little time for feature development.
- Solidity and its limitations.

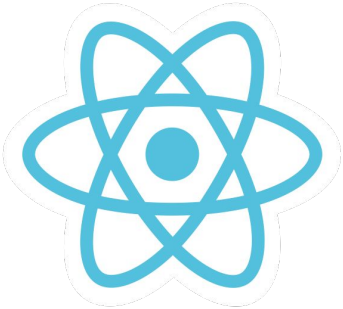
Smart Contract with Solidity

Some of the pitfalls we kept encountering:

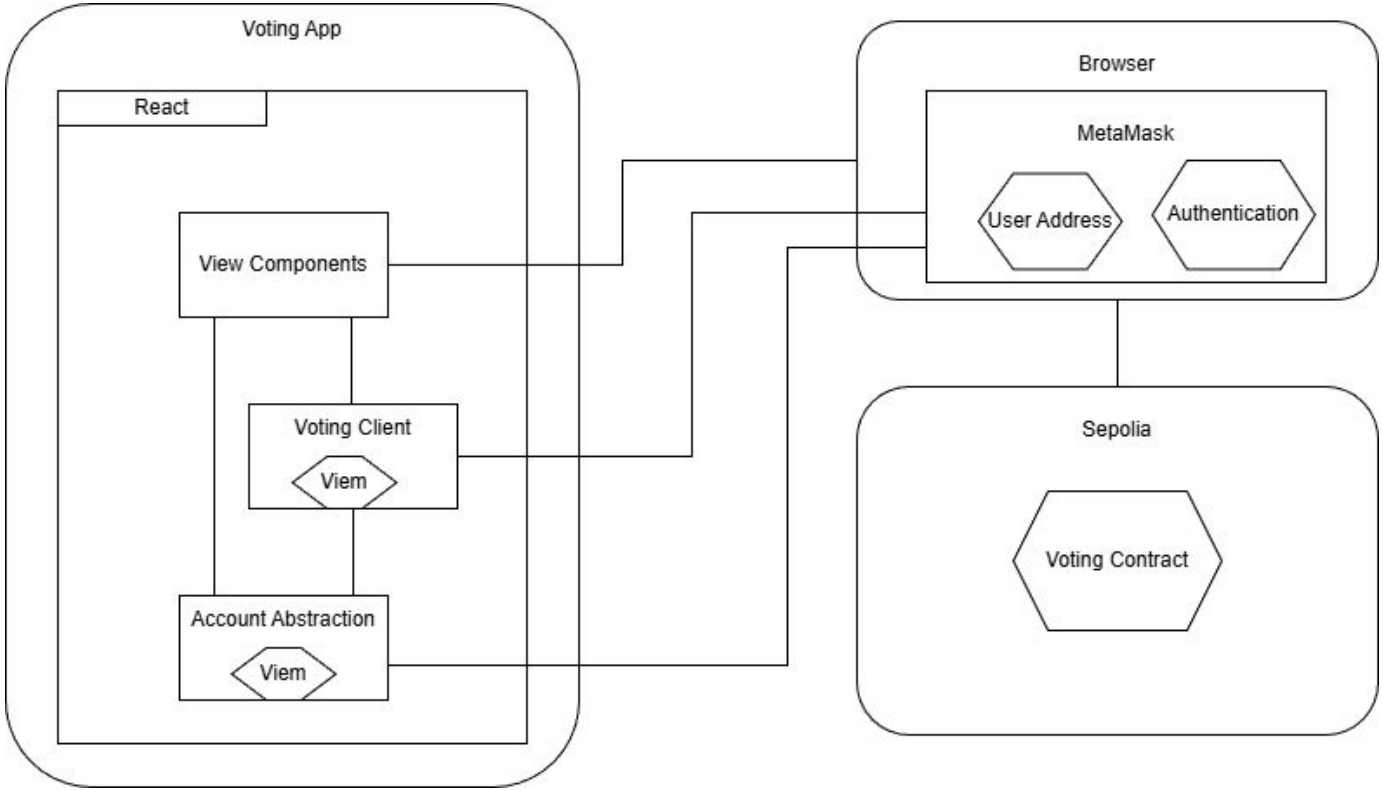
- Maps are efficient, but also limiting:
  - We had to change some properties to arrays, like `participants` and `options`.
  - Just wanting to be able to iterate required adding an additional property: `uint votingsCount;` for `mapping(uint -> Voting) votings;`.
- Arrays are also limiting:
  - Key functions like `filter`/`find` or `map` are missing.
- Cumbersome development process:
  - Testing each change required a cycle of compiling, deploying and updating the frontend.
  - Also resulted in a loss of created votings after changes to contract.

Tech Stack

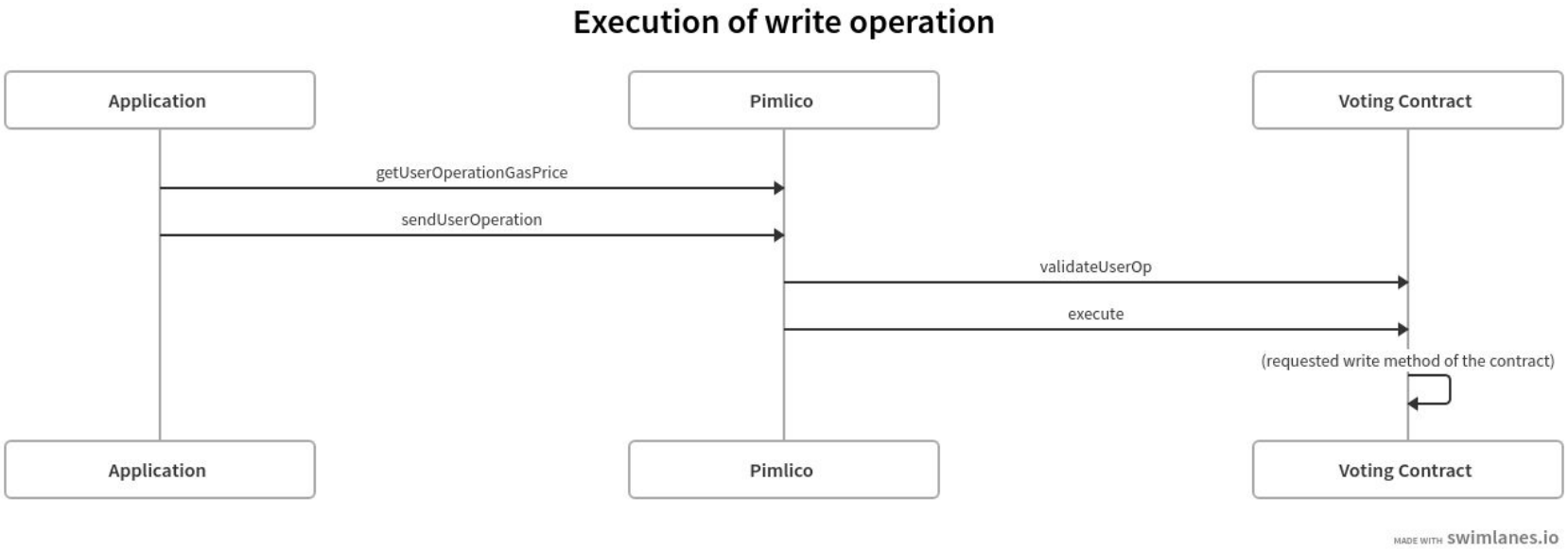
- Frontend: React & Viem
- Pimlico as a Bundler
- Smart Contract with Solidity



Architecture



Account Abstraction



- Voting Contract owns Ethereum which are used to pay for the Gas.

Demo

Checklist

✅ A working prototype demonstrating gasless user interactions ➡️ Demo

✅ Use latest stable releases of chosen libraries and frameworks ➡️ Updated

✅ Interaction with a public blockchain ➡️ Sepolia

✅ At least 2 participants need to be involved ➡️ Demo

✅ Gasless transaction implementation using account abstraction ➡️ Demo

✅ Status and process need to be shown in the frontend ➡️ Votings and Alerts

Thanks 🖤