2020 Q3 Hyperledger Burrow

Project

Hyperledger Burrow

Project Health

This quarter has certainly been a fallow quarter for Burrow, mostly as a consequence of Covid and in the case of Monax the need to refocus on our product built on top of Burrow. We do have more work planned and another developer coming in to work on the SQL mapping layer Vent at the end of this year to extend its functionality.

Inevitably this raises considerations that have lingered around Burrow and the level of development activity from a wider community. I think it would be premature at this stage to talk about end-of-life for Burrow as I am aware of other companies taking an interest (such as CertiK). However I think the next two quarters will be critical for Burrow and on that timescale we should revisit this question and decide whether there is a realistic prospect of Burrow leaving incubation (chiefly in having sufficient contribution-weighted contributor diversity).

I do expect Q1 2021 to be quite active in terms of work that Monax needs to do on Burrow so one way or another Burrow does need an active development home in the medium term, but I defer to the TSC to judge whether Hyperledger is that home. If we are able to garner greater community involvement then I think it could, but so far with the time and resources I have available to me for community building that has not really happened.

Questions/Issues for the TSC

Just any reflections on the above.

Releases

- [v0.30.5] Provide `BytesToHex` for Vent projections

Overall Activity in the Past Quarter and Current Plans

There has been a certain amount of thinking activity away from lines of code:

- Paper design for some features to Vent that make it more capable to form projections over events - generated columns, literal columns, JSON support. You can define Solidity events (via EVM log) and form log-structured tables or realised projections (using SQL upsert idioms) to reflect your blockchain-stored evidence in a traditional SQL schema read-only, combined with other write tables, blended with views
- Idea to support event emission via AssemblyScript WASM constracts
- Using JSON Schema and tooling from the Typescript world (particularly the excellent https://github.com/gcanti/io-ts) to generate smart contract stub code, event schema, and database mappings

The theme of all this design work is geared towards emphasising a schema-first (OpenAPI spec/JSON schema) and code-generation heavy way of working with event streams and having Burrow specialise in storing streams of events with relatively lightweight (much more lightweight that what we have typically done in Solidity) contracts to maintain integrity of our events. Also the idea to move away from Solidity/EVM and provide a way of working with events in AssemblyScript (keeping the events isomoporphic to Solidity events).

Maintainer Diversity

See: https://github.com/hyperledger/burrow/blob/master/MAINTAINERS.md

Contributor Diversity

No new contributors

Reviewed by

- ✓ Angelo De Caro
- ✓ Arnaud J Le Hors
- ✓ Christopher Ferris

- [x] Dan Middleton
- [x] Gari Singh
- [ ] Hart Montgomery
- [x] Mark Wagner
- [x] Nathan George
- [x] Swetha Repakula
- [x] Tracy Kuhrt
- [x] Troy Ronda