

Offline inboxing MVP retro & planning

Doc for meeting 12th of December, 2017. Please fill in things that went well things that we can do better.

Retrospective (~30m)

What went well?

- We delivered an MVP with scope we said (even a bit more, tests etc)
- Debugging between Clojure/Go last few days, communication general
- Good research, figured out more details on Whisper internal, strength/weaknesses
- Good narrow scope for focus

What can we do better?

- Clarify blocking interfaces earlier
 - Uncertainty, to learn domain etc
 - Env for end to end testing, ~nodejs (~check after call)
- Get end to end testing earlier, spike style (QA involved just last 5-10%, Clojure mostly last ~20-25%)
 - Better describe feature for QA, one standup a week with QA, involve in process
- Think better about amount of work vs number of devs. On status-go side, two devs would be enough
 - Uncertainty, changes as time passes
- This is my second idea with 50% involvement and I believe it's better to join for one idea for 100%. (Adam)
 - poss fix: sync up week start date across swarms?
 - See board ideas progress for given week, if assigned to two idea got a potential problem

Misc questions/notes1

- Did people think MVP scope and date was reasonable when they committed to idea?
 - Hard to know when big part research
- Lots of research first - make explicit?
 - Probably
- Should Oskar have been more involved in the Go side (main part of MVP work) of things initially?
 - No, more involvement wouldn't have helped

Initial planning (~30m)

Copy paste from <https://github.com/orgs/status-im/teams/status/discussions/5> so we can live edit this and adjust as necessary.

Iteration 1 - Public chat, MVP retainer, and laundry

Goal Date: 22 December (2w)

Description: Main user story: Support public chat.

MVP retainer and laundry: QA of MVP complete and all MVP work merged into develop on status-react and status-go. MVP bug fixing. Basic error handling, refactoring and logging. Documentation for updating wnode-status and API calls introduced. Identify and make bounties of some nice-to-have / upcoming issues. Planning and swarm participants re-organization: likely one more Clojure dev and identify Go dev needs. Clarification on interface (enode). Specify group chat workflow and tests thereof.

Uncertainty: Swarm availability, both w.r.t. other ideas and holidays (seems OK).

- Availability normal for 2 weeks: Adam 2w 22 Dec, Eugene, Boris, Dmitry, Oskar
- After Christmas no more Go dev necessary

Iteration 2 - Joe Blockchain goes to town (multiple wnodes)

Goal Date: TBD - est. +~2w, due to holidays possibly delayed till mid January

State: Draft

Description: Ability to toggle between different wnodes in the app. Show basic connectivity to wnode in interface. The wnodes identifiers are hardcoded in the app. Add by enode (?). Multiple wnodes enabled for redundancy (?).

Uncertainty: Go dev work necessary. Have two parallel track to investigate and support incentive layer?

- Docs and options for wnode-status but other than that not

Different swarm, not part of this

: Iteration 3...n - Joe gets rich (incentive layer)

- **Seems complicated and different**
- **Start research before change iteration**
- **N can be big**
- **Focus on other ideas, step back, notifications, for groups**
- **Other idea**

Goal Date: TBD

State: Early-stage draft

Description: Enable payment layer. Joe run a wnode and he should be compensated in SNT for the work he does. Payment options: chunked (prepaid SIM), per req/work (micropayments) or after completion (credit). Requires some back and forth with wnode to establish “contract” (can be as simple as “you got this week’s password”). Possibly some way of knowing if “contract” has been fulfilled. Might require design. Business decision on how we are going to provide this by default.

Uncertainty: Over 9000.

Not in scope of this swarm

The following ideas are related but separate from this swarm:

Onboarding. Discovery mechanism for finding wnodes. Using wnode for push notifications (requires iteration 3, at a minimum iteration 2). Reputation system. Negotiation layer.