What would the application requirements look like for “minimum viable distributed Anoma coordination”? First, let’s define what this means. I define “minimum viable distributed Anoma coordination” as a hybrid setup where essential parts of the coordination mechanisms can be run on Anoma (and thereby distributed), while other parts still likely not run on Anoma (e.g. video calls) can be performed either with existing E2E encrypted centralized or decentralized software - and where all the key data required to continue coordination is distributed and autonomously controlled.

By this definition, essential parts include:

- payments / compensation for work performed, e.g. scale-free kudos
- promise tracking, e.g. promise graph
- knowledge aggregation, e.g. knowledge graph
- code management, e.g. git hosting and canonical reference tracking
- ephemeral structured chat, e.g. multichat

Inessential parts include:

- E2E encrypted video and audio calls, for which sufficiently secure/decentralized alternatives exist, and which we can easily switch (no long-term data or ontology)
- Email, which will always be a legacy system, and for which sufficiently secure/decentralized alternatives exist, and which we can easily switch (no long-term data or ontology)

We should also consider that we may be able to hook up Anoma as a “distributed database backend” to existing frontends which have solved many of the frontend problems. For example, something like Anytype (which has their own, albeit less general, synchronization protocol), could potentially be a suitable frontend for Knowledge Graph.

An interesting question is whether these essential applications should have separate or unified frontends.

cc @apriori @degregat @tg-x for brainstorming input