The goal of the session was to brainstorm some more specific applications that we can at least work out at some conceptual. Thanks to everyone who participated!

We worked out some subset of applications we can choose to further refine. The initial thing we should do with application concepts is brainstorm to level of Public Signal, write a blog post.

Applications

1. Knowledge graph
2. distributed version of obsidian or remote research or Emacs but with distributed storage, permissions. A bunch of different documents which are doubly linked. Documents can be owned by different people or different groups, use compositional identity for convenience, can be linked to each other and people can see portions of the whole graph. CRDTS. Mergable replicated data store could be aback end for this knowledge document.
3. Because we have permissions - the test users could be;
4. Informal Systems
5. Black Sky
6. Radicle Exchange
7. Heliax (internal knowledge wiki)

8. Informal Systems
9. Black Sky
10. Radicle Exchange
11. Heliax (internal knowledge wiki)

12. Quadratic Funding
13. we need to figure out how multiple funders work. Gitcoin is single funder - is this similar or different from Public Signal? Aspects of Gitcoin - are real humans anti-Sybil if we have an anti-Sybil mechanism if we have Gitcoin passport then it does - we can do with quadratic funding curves - many communities do have quadratic funding.
14. Multi-chat
15. distributed slack network without a central server operator and we can come up with a better name. The difference between slack, discourse and the forums are mostly different interfaces operated by different companies, disentangle protocol from operator. (Decentralized discourse and discord - a permissioning scheme not designed for hierarchical organization).
16. The key thing which this has is permissions and information flow control in a heterogeneous graph, knowledge graph is trying to build persistent. If you are chatting, it's ephemeral. Users don't think about papers and chat as the same question.
17. complex permissioning
18. message expiry
19. encryption/privacy
20. designed to handle heterogeneous organization graphs just like when you have discords or slacks you have some shared channels managed on centralized servers.
21. Share data communicate over p2p network about orgs you share information with - just work out conceptual basis
22. The user for this could be Heliax/Informal/Osmosis/Blacksky/Binary builders

23. complex permissioning
24. message expiry
25. encryption/privacy
26. designed to handle heterogeneous organization graphs just like when you have discords or slacks you have some shared channels managed on centralized servers.

27. Share data communicate over p2p network about orgs you share information with - just work out conceptual basis
28. The user for this could be Heliax/Informal/Osmosis/Blacksky/Binary builders
29. LunchBot
30. Determine where we go for lunch
31. people submit their preference function - there is not an obvious solution to lunch intents, the best solver can extract the best solutions
32. People input their lunch preferences - time, location, what to eat
33. The advantage of LunchBot is that it can be used in a cross-organizational fashion to meet up with others from Heliax. Anyone in the org can hack together a makeshift frontend
34. LunchBot application - pooled payments this is a cool integration to know about. Split wise.
35. You privately commit you rather have these three people pay. You need to cooperate with untrusted parties, but they also defect (interesting payment selection mechanisms)
36. People like eating lunch.
37. The second application is you resolve the underlying problems you found.
38. Can later pair with a reservation function like an Anoma native Open Table
39. Flash Liquidity Network
40. an iteration of scale free kudos. Block builders often need access to capital otherwise the only builders will be highly centralized. You could build an application on SUAVE such that Anoma scale free kudos network lends kudos on demand to block builders or solvers. The terms of the loan can be defined at the outset, including repaying the loan at the end of the building sequence or even over a long-term time horizon. Loans can be based on reputation.
41. Type checked Github
42. Storing code, loading code, software updates - visualize data so data is stored such that a user can make data structures - want to use Nockma code to visualize data on chain and do stuff.
43. Intent visualizer
44. (Network data visualization). Visualized front end data store, you want to identify latent demand, which can be explicit with intents.
45. This is for users and potential people who will build things that intents will be satisfied by like restaraunters. Some information can be aggregated without sacrificing too much privacy.
46. They can figure out if its a good idea to build a restaurant at a given location, given demand for example. - Users disclose only the aggregate demand over one-week period for different cuisines per block. The social optima is not everything is completely private.
47. Want the visualization logic written in Juvix; e.g. all the code in Urbit is written in Nock. Most of the tooling should be ideally within the system and shouldn't be outside. The VM shouldn't provide custom logic to make this work.
48. Promise graph
49. see GitHub.
50. The Anoma Phone
51. some open-source piece of hardware which connects directly to the Anoma network. An Anoma microkernel. We can make a microkernel to just run Anoma, unikernals. Microkernel of Linux which runs Anoma on top. More research is tbd. Connects to the P2P network and then the appstore on the phone is served over Anoma. (P2P app store). We want to use the trust information here. The difference between this and Solana phone, there is no android. All apps distrusted over Anoma and written in Nock.
52. Decentralized scientific archive
53. jstor or libgen - integrate with IPFS but also the organization of the whole thing they are indexed and organized. It could include open peer review with different permissions to manage different subfields.
54. Anoma doesn't need to store the data, we just want to unify the interfaces. S3 may or may not run Anoma. (Liberal science) - working on research tools publicly funding research.

55. The science could be indexed for ease of rpgf.

56. We are pro DeSci. What is currently there is quite underwhelming.

57. What do we do instead of peer review? We can do a peer review system on it; e.g., 3/5 sign and now your paper is published to ID space and can be consumed.

58. The reason why the resource_logics

are interesting is you can write lattices - lattices on steroids as you can check other logic. Can do a lattice system, depending on each department you can do some interesting things.

1. Z

2. alt twitter. Mastadon ties data storage together with permissions.See here.

3. Decentralized virtual office

. Gather.town with distributed permissions. There is a federated gather.town connected to other gather.towns, and you can move them around based on permissions. Secret places you can only go if you have special permissions.

1. Decentralized Music curation

2. steer clear of thinking about the incentive models of art. We can explore it. We can do decentralize music curation - have friend graphs curate your music rather than Spotify or algorithms - when sorting articles.

3. Giant decentralized art gallery

4. art exhibition - have it distributed permissions versions. Museum curation and peer review have a certain similarity, just curating different archives. If you want to bring in the media theorists, you can unify this.

5. eVoting

6. very controversial. I like the idea, I think it’s totally doable. Switzerland has the best eVoting in the world as they open sourced their entire stack. It’s done by Swiss post in Java. Turned over into DAOs. Basic polling functionality where the user states preferences. Not binary voting on specific topics, you express your preference function and preferences indicate votes. No one has experimenting with user set preference functions over public funding.

7. The Anoma organizational stack

8. Different kinds of governance systems based on user preferences. It’s a combination of many concepts, but it’s worth covering the Anoma organizational stack as a single thing. We are just a software provider which doesn’t operate anything.

9. Microblogs

10. I want to make a microblog - do some blogging - microblog curation.

11. Games

12. Chess / MUD - match each other in mud and setup chess games in each other. Space station 13. Text interfaces where people chat with each other and go to different rooms.

13. In system IDE

14. run IDE where I prevent memory. It’s a repl, so I can visualize a bunch of stuff. It should be fully integrated want desktops and stuff.