Urbit Foundation Operating Plan

2023 and beyond

There are two major parts of this document: the <u>2023 Plan</u> and <u>2022 Review</u>. Each section is preceded by a high-level summary of the contents.

The 2023 Plan discusses our change of course from 2022, why we're doing it, and the major areas we're going to be focusing on, which comprise core development, developing developers, ecosystem coordination, habitual communication, and a governance redux.

The 2022 Review goes deep on what we did for the last year, why we did it, and comprises about 60% of the whole document. It covers our efforts in marketing the ecosystem, performance of our grants program and education efforts, CombineDAO, and includes a report specifically on Assembly 2022.

Contents

- <u>2023 Plan</u>
 - <u>A New Mission</u>
 - Core Development
 - <u>Developing Developers</u>
 - Ecosystem Coordination
- <u>2022 Review</u>
 - <u>Developer Education</u>
 - o <u>Grants</u>
 - <u>CombineDAO</u>
 - <u>Marketing</u>
 - Assembly 2022

2023 Plan

We're leaving 2022 feeling pretty good about the year. <u>Urbit is taking off</u> and <u>people are</u> <u>noticing</u>. Early in the year we set out to <u>match Urbit's appearance to its substance</u> — <u>mission accomplished</u>. What's more, our ecosystem has matured in a remarkably short period of time and is now attracting many developers, new organizations, capital, and general attention.

What is most important now is actually maturing Urbit technically in a world where the technology is being adopted, so we're adopting a new mission: **"OK: Steward Urbit to technical maturity."**

We've brought Ted Blackman on as our CTO to lead a new in-house team of core developers. We'll also adjust our developer education and grants to train developers to work on the core.

Aside from becoming more involved in the development of Urbit, **we'll be implementing a policy of habitual communication at every layer of our organization**. Coordination in a decentralized ecosystem necessitates a steady stream of information made available to galaxies and other stakeholders and serves as the basis of organizational transparency, which the UF needs to have.

With a year of experience under our belt, an ecosystem in full swing, and a more serious long-term mission ahead of us, we're also going to be tightening up the UF's governance. The urbit.org galaxies must be used to fund public goods, and doing so should be administered with careful, informed oversight. We will encourage the galaxies to elect a new board of directors who will take an active role in aiding the UF in achieving its mission.

Contents

- <u>A New MIssion</u>
- <u>Core Development</u>
- Developing Developers
- <u>Ecosystem Coordination</u>

A New Mission

"Urbit consists of that system software which admits of a Kelvin version."

- Philip Monk, Precepts F10

The Urbit Foundation's mission is to steward Urbit to technical maturity, or as we like to say, 0K.

0K, or "Zero Kelvin," refers to Urbit's versioning system that decrements downward to version zero, at which point no further versions will be issued. Reaching 0K represents the completion of the Urbit project and the accomplishment of our mission. We expect to be focused on this for many years.

Core Development

The cornerstone of the Urbit Foundation's "OK" strategy is to build out an internal team of core developers. By Q4 2023, UF plans to have seven core devs in-house — in addition to ~rovnys-ricfer as CTO, long-time UF devs ~tinnus-napbus and ~dinleb-rambep are now working on the kernel, and UF will hire four more core devs over the next year.

The existing core team has nine developers, so adding this seven-member UF team represents a major expansion, with the potential to greatly accelerate the pace of development of the runtime and kernel.

A bigger team, spread across multiple organizations, requires more coordination. To that end, UF has <u>published</u>¹ a technical roadmap for Urbit core. This is a living document, intended both for casual readers and highly technical readers, which will serve as a starting point for anyone interested in staying up to date on what's going on with core dev.

The roadmap addresses one of the main bottlenecks to getting involved with core dev: it's hard to know what's being worked on, what is planned to be worked on, and what architectural changes can be expected from future work. By keeping a frequently updated listing of projects under active development, what's next up in the release pipeline, and links to proposals for future work, UF will ensure that both core devs and app devs all across the Urbit ecosystem can act according to current information and anticipate future changes.

In addition to maintaining the roadmap, UF will also broaden its educational offerings to include advanced material about the runtime and kernel, targeted at devs who already

¹ Okay, not quite yet at time of writing. But the contents of <u>this now outdated document</u> will make their way onto <u>https://roadmap.urbit.org</u> by ~2022.11.14.

know the basics, such as Hoon school graduates, app devs, and grantees. Raising the level of understanding of the deeper parts of the system among ecosystem devs will improve the quality of app code and facilitate core contributions from devs who aren't working on it full-time — reliability, API ergonomics, and developer tooling all stand to gain from more widespread fluency with the internals of the Urbit OS.

To ensure core dev maintains its collegial atmosphere and unified control over the codebase, which must continue to congeal into a tiny, diamond-perfect kernel, the core devs will be forming a "guild". This guild will be responsible for technical deliberation about Urbit OS. While userspace develops as a "bazaar" in traditional open-source terminology, the core needs to remain a "cathedral" to prevent it from increasing in size and scope, which are the natural pressures that arise in multi-party, evolutionary development.

UF is taking these measures in order to minimize technical risk for the Urbit ecosystem as a whole. Many companies are now depending on core development to accelerate to get Urbit "over the hump" into becoming a bulletproof, scalable consumer product. UF will be deploying significant resources toward resolving the bottlenecks in core dev to crush the remaining blockers to adoption.

Developing Developers

Bringing new developers into Urbit has been a mainstay of the Urbit Foundation's activity since its inception and that isn't going to change. There's still a lot of Urbit left to build, and we won't be able to get there without bringing much more talent into the ecosystem. We conceptualize this as a funnel, or pipeline that starts with outreach, proceeds into education and applied opportunities (grants), and ends with full-time work.

Outreach

The UF will undertake a few new efforts to bring more people into the developer ecosystem. One such new effort will be producing video content about core development — walkthroughs of modules, lectures on advanced topics, question-and-answer sessions with core devs, and live streams of core devs sharing their screen as they work.

We will also be laying bigger nerd traps. Urbit has as much intellectual depth as any software project. Its unsolved problems, once explained, often capture the imagination of high-caliber developers, so this can become one of our primary recruitment tools.

We will write open letters to famous developers comparing their work with various parts of Urbit, and post them to twitter and hackernews for exposure. We will post challenges with large star bounties for difficult work items. We will be iterating on how to use the attention economy to our advantage. Among other successes, the cohort classes have demonstrated that new developers can begin with Hoon as a first language and Urbit as a first platform, challenging preconceptions from subgroups of the legacy developer community. Continuing to showcase success stories — of developers learning Urbit successfully, building cool things things, and speaking to their experiences — is our best line of defense at showing developers that Urbit is worth the investment.

Education

Urbit has a large learning surface area: a new language, programming paradigm, operating system, and naming conventions. Convincing developers to learn all of this requires, among other things, an uncommonly good educational experience. There are two entry points: documentation or cohort-based classes.

Documentation

Our documentation has been <u>heavily reworked over the course of 2022</u> to provide a clear path from newcomer (Hoon School) to app developer (App School). Many developers are autodidacts that can learn a new stack by consuming documentation at their own pace. While we feel mostly satisfied with the content of our docs, we believe that they can be further improved by including professional video content that enriches the learning experience beyond what written content can provide. ~lagrev-nocfep will be working on recording short videos explaining specific topics that become interspersed throughout the major courses.

Cohort-based Classes

Instructor-led classes like Hoon School Live and App School Live <u>have been successful</u> <u>throughout 2022</u>. For students, they provide camaraderie with and support from other learners, and access to an instructor that can answer questions and provide feedback. Among other successes, the cohort classes have demonstrated that new developers can begin with Hoon as a first language and Urbit as a first platform, challenging preconceptions from subgroups of the legacy developer community.

While classes like this don't scale as well as self-led documentation, they provide our team with valuable insights about teaching the material that are crucial for this stage and remain manageable at the scale we're dealing with. We expect to ramp up the frequency of live courses that are conducted over the next year to ensure that at least one course is always happening every quarter on a regular basis, giving us the ability to constantly add new developers to the funnel on an ongoing basis.

In light of our new focus on core development and the success thus far with existing courses, we'll be expanding the scope of our course offerings to include core

development — Arvo School. We expect this to take some time to develop and likely won't run such a course until Q3 of 2022.

Grants

Our grants program is <u>working well</u> — we've built solid systems that set a high standard for grant workers, which keep quality high and abandonment low. We're not planning any major changes to the way our grants program works, but we do plan to shift the emphasis of grants from app development to infrastructure development.

The three categories of grants are proposals, bounties and apprenticeships. Here's a breakdown of how they work:

Apprenticeships

An apprenticeship awards a single star for roughly three months of work to a contributor that's paired with a Mentor. The Mentor is responsible for defining the project, selecting the contributor, and providing mentorship throughout the process. Mentors can be any past contributor or employed developer. Most apprenticeships have been infrastructure-focused throughout the year, and have also been used for maintenance of existing products (e.g. Port, Bitcoin Wallet). Apprenticeships have historically been the least numerous type of grant.

In 2023 we plan to continue using apprenticeships in the fashion we've been using them, although with more of a focus on keeping products — especially those that can be considered infrastructure, like Bridge — maintained.

Bounties

Bounties are contracts for well-defined work. They involve one or more contributors and a Sponsor. The Sponsor is the one that wrote the bounty and stands to benefit from its completion. Bounties have spanned infrastructure contributions, hosting meetups, developing apps, adding features to apps, and producing content.

In 2023 we plan to utilize bounties for incentivizing high-value contributions to infrastructure work. The goal of this is twofold: produce lasting contributions, and capture much higher-caliber developers by presenting them with interesting problems that have high rewards. With Ted at the helm and our own developers increasingly focused on infrastructure, writing technical specifications that can be turned into bounties of this sort will be much more viable.

Proposals

Proposals are projects proposed by contributors. In order to be accepted a proposal must have a Champion — a member of the community that has previously made substantial

contributions and continues to be engaged in working on Urbit. Proposals throughout 2022 have typically been for app development, which makes sense: new contributors can see the path to building an app much more readily than they can figure out how to contribute to infrastructure development, since the learning curve is lower and the problem space larger.

Our hope is that the presence of the <u>public roadmap</u> (goes live on November 14th) will make infrastructure proposals more common, since developers will know what areas to direct their contributions towards. Additionally, our grants budget will now be more heavily skewed towards infrastructure contributions, placing a restriction on the amount of funding available for app development.

Why continue encouraging app development at all? We've seen numerous cases of app developers eventually transitioning to infrastructure development and don't wish to close a pathway that works. For a more quantitative view, see below:

| Total Full-time Developers | 58 |
|--|----------|
| Started in userspace | 29 / 50% |
| Started in grants | 27 / 48% |
| Contribute to infrastructure | 22 / 38% |
| Infrastructure that started in userspace | 12 / 21% |
| Infrastructure that started in grants | 11 / 19% |

Full-time Developers as of ~2022.11

We don't know that app development is the most effective way to produce infrastructure developers, but we do know that it works and has produced good results in the past. This year we'll be trying other tactics while keeping known pathways open.

Ecosystem Coordination

Raising efficiency in a rapidly expanding ecosystem is a lot like increasing efficiency in a high-growth company — coordination costs grow exponentially and require continuous improvements in communication. We're committed to making communication across a variety of mediums a core competency of our organization. We believe that, in so doing, the Urbit ecosystem will be able to work more effectively: more quickly in complementary directions.

Habitual Communication

From this point forward we'll be publishing monthly² reports in our <u>new public group</u> across a few different areas:

- **Ecosystem:** Long-form written updates about what's going on with particular organizations in the ecosystem, what they've got coming up, and how you can help them. We'll also be surfacing open roles and contributors that are looking for work.
- Infrastructure: Updates on the state of infrastructure development, discussions of current core development strategy, updates in the context of the new <u>public</u> <u>roadmap</u>, outstanding issues and what's being done about them, and areas where you can get involved. Updates will specifically reference the new
- **Developer Pipeline:** Monthly reports focused on developer content, new projects, and metrics about developer funnel and grants, like: grants by status, stars distributed, total number of developers, engaged developers, and employed developers.

We'll continue publishing the two regular newsletters that we already do: the general Urbit Monthly newsletter and [battery payload] (the developer newsletter). Keeping these running will be even easier with the rest of our organization regularly engaged in providing the above information.

Quarterly Reports

Wherever possible we'll be publishing metrics at regular intervals for time-based analysis in the future by anyone that needs them to make decisions about their involvement with Urbit — be it hiring, investing, building, or using. These will be based on network-wide statistics like number of online ships, Azimuth activity and asset prices, as well as UF metrics about developers, infrastructure development velocity, grants, and fund distribution.

² At minimum.

Events: IRL and Virtual

We believe that events, both IRL and virtual, are major drivers of growth, connection and progress.

Events like last year's <u>Town Hall</u>, <u>Assembly</u>, and the upcoming <u>Developer Week</u> provide teams with natural timelines to work towards, and with a decentralized ecosystem like ours, can result in many new products and updates getting delivered all at once. Consider that Assembly showcased product launches from Tlon, Native Planet, Holium, Uqbar, Quartus, Tirrel, Mars Review of Books, and more — all of these happening at once has a compounding effect. Developer Week is motivating the completion of new documentation, six blog posts, and at least two product launches.

IRL events are especially good at building connections. The Foundation has built strong relationships with Reserve Protocol, Vienna Hypertext, Coinfund, Delphi, and numerous other organizations and individuals through the facilitation of IRL events. Companies get the chance to meet contributors and make hires. Organizations form partnerships and make plans.

We'll continue to produce virtual and IRL events that rally the ecosystem. In addition to producing Assembly 2023, we will continue ensuring representation at major Web3 conferences, support events like the developer-focused Volcano Summit, and in general, provide opportunities to get together.

2022 Review

Assembly represented the conclusion of a year-long arc.

A year ago we set out to <u>"support, engage, and grow a nascent ecosystem."</u> We adopted the mission to *ensure wide-scale adoption of Urbit as a technology* and spent the year focused on bringing this about. We built a pipeline for new developers to <u>learn</u>, <u>grow their</u> <u>skills</u>, <u>invested in several companies</u>, and <u>promoted Urbit and its ecosystem</u> throughout the course of the year in order to <u>match Urbit's appearance to its substance</u>.

These efforts culminated with the <u>Assembly in September</u>. Our ecosystem has now come into existence, is deeply engaged across many different vectors, and is growing faster than we can keep up with.

The rest of this document is dedicated to a deep dive on the major things we did throughout the year with backing outcomes wherever we measured them.

Contents

- Developer Education
- Grants
- <u>CombineDAO</u>
- <u>Marketing</u>
- <u>Assembly 2022</u>

Developer Education

We've focused our efforts to provide better education around Urbit development on <u>documentation</u>, <u>courses</u>, <u>hackathons</u>, <u>and more</u>. Our efforts here have been led by ~lagrev-nocfep, who joined us part-time at the beginning of the year, and ~tinnus-napbus who has been working on documentation at the UF since late 2021.

Documentation

We've made major strides on documentation throughout the year. Perhaps the largest and most visible update is in the launch of <u>developers.urbit.org</u> (DUO) in June that streamlined documentation into a concrete learning pathway (the "Core Curriculum") and separated guides from reference material. That launch was the culmination of over half a year of documentation work and has since quelled the classic and popular complaint: "the docs suck." <u>Areas for improvement</u> hit our attention from time to time, but by and large we now hear that the docs are, in fact, <u>quite good</u>.

Here are some of the major new pieces of documentation completed over the year:

App School I: A twelve-part guide that covers Arvo basics and all the specifics of writing Gall apps in detail. This is designed to be the natural follow-up from Hoon School that provides working knowledge of Gall agents without requiring any JavaScript knowledge.

App School II: A nine-part guide that walks the reader through creating their own Journal app. It involves building a Gall agent, a React-based frontend, and then publishing it for distribution. This guide is end-to-end and leaves the reader ready to write their own apps.

<u>%docs app</u>: We knew a while back that we wanted to orient DUO around Urbit documentation, not app documentation. After all, the number of apps developed will continually increase, they'll be built by all kinds of people, and it would be neither feasible nor desirable to centralize them on any urbit.org domain. Instead we leveraged software distribution to build an application that sources application documentation directly from installed applications. This means that a developer simply has to include their documentation with their app, and developers can access documentation through their Urbit by installing the desired app. This author knows of at least 11 applications documented in this way, including Tlon's new Groups app — this has become a standard way to document Urbit apps.

Hoon School Dead: The Hoon School documentation up until the release of DUO was roughly four years old with few major updates beyond syntactic fixes as the Hoon language advanced. ~lagrev-nocfep's experience teaching Hoon School Live (not to mention his experience teaching in general) led him to believe that a significantly better version of the Hoon docs — Hoon School Dead — would be valuable to replace the aging

guides with. The course features 19 lessons with diagrams, numerous exercises, code walkthroughs, and an expanded set of covered topics.

Lightning Tutorials: In addition to the tutorials above that are designed to give in-depth understanding of Urbit application development over what should take multiple days, we also released a series of three "lightning" tutorials that eschew teaching the stack in favor of giving a high-level sense of the code required to build an app in as short a time as possible. There's commentary on the interesting parts of the applications in order to showcase how little it takes to build Urbit apps, and the three all build on each other to demonstrate Urbit's ability to "stack" apps by sharing backends.

In addition to the above a series of smaller but still important pieces of documentation were produced for developers: a conceptual outline of the <u>scry</u> and <u>subscription</u> APIs, a ton of updates to the <u>Glossary</u>, and more recently an <u>end-to-end example of building a</u> <u>version of Flappy Bird that combines a %pals-based leaderboard</u>.

While not necessarily developer-oriented, we also rewrote the <u>Cloud Hosting Guide</u> and the <u>Getting Started Guide</u>.

Courses

The most high-profile move in Urbit education this year was the revival of Hoon School. For those that aren't familiar, Hoon School has been taught as an instructor-led cohort-based course several times over the years. Then taught the first one (afaik) in 2019, a couple of subsequent ones in early 2020, and the community-led Hooniversity took it over in late 2020. No cohort-based courses were taught in 2021.

The version that ~lagrev-nocfep taught called Hoon School Live (HSL) was the most popular of any that preceded it. The first batch started in February and ran for ten weeks. ~lagrev-nocfep, who has a background in engineering pedagogy and has taught computer science for over a decade, built a brand-new curriculum for this batch which was discussed on this developer call.

Following HSL was another newly-designed course for app developers called App School Live (ASL) that culminated in a hackathon held at Assembly. The hackathon featured 10 teams spanning 19 individuals, all of whom finished a project. Another cohort of HSL started immediately following Assembly and is still underway. This cohort features an accelerated schedule from 10 weeks to six weeks.

Below are some metrics on signups and completion of the various courses. We unfortunately don't have any data on the Tlon-led courses from 2020:

| Course | Signups | Starters ³ | Finishers⁴ |
|--------------------|---------|-----------------------|------------|
| Hoon School ~2019 | 17 | unknown | 5 |
| Hoon School ~2020 | unknown | unknown | unknown |
| Hooniversity ~2020 | 32 | 32 | 20 |
| HSL ~2022.2 | 253 | 147 | 61 |
| ASL ~2022.7 | 101 | 54 | 28 |
| HSL ~2022.9 | 278 | 1135 | ongoing |

The following information about prior programming experience was also collected for the first batch of HSL and will be collected again following the conclusion of the fall cohort of HSL:



³ The number of people that turned in the first homework. Set to the number of signups when unknown, which is the case for older courses.

⁴ For HSL and ASL, this is the number of people that turn in all of the homeworks.

⁵ This is probably the final number, although more submissions may arrive

Hackathons

Throughout the year we've participated in or run three hackathons and are about to run a fourth.

- 1. NFT Berlin: focused on uses of Urbit ID
- 2. HackFS: integrations between Urbit and IPFS
- 3. Assembly: opportunity for ASL students to build applications
- 4. <u>encode.club</u>: a four week hackathon starting in November

NFT Berlin had one team that engaged in the Urbit project, members of which signed up for Hoon courses and showed up at Assembly. HackFS produced nothing of note. The Assembly hackathon that was run for our community had great turnout and produced excellent results (see below), and we've got high hopes for our collaboration with encode.club. Our experience in the last year has taught us that participation in external Hackathons is only possible if the challenge is broad enough for people without Hoon knowledge to partake in (NFTBerlin).

Assembly Hackathon Results

The Assembly 2022 Hackathon received 10 submissions from 16 total participants. Every team had something to submit. The following projects were submitted:

- Gather, a meetups invite and coordination application
- Whom, a contacts app the builds on the concepts initially explorer by %pals
- %classifieds, a tool for selling material goods to people in your community
- **Cue**, a bookmarking tool that resurfaces previously saved articles
- %ahoy, a peer ship status and connectivity monitoring tool
- Temple O Chess, an homage to Terry Davis with Urbit native matchmaking
- **DukeBox**, a DOS game emulator with Doom and Mortal Combat distributions
- Plug, an online storefront for selling hardgoods and inventory tracking
- Herd, a dependency management system built to improve developer experience.
- Quorum, a Stack Overflow clone with search and voting functionality

These ten submissions competed for four possible demo slots at Assembly, which were granted to Whom, %classifieds, Herd, and Quorum based on a selection criteria across ease of installation, quality of documentation, potential impact of project, and front end experience.

The reception of these projects from the community stretches beyond just those that were selected for demos, with %ahoy even being extended to include an automated `|pack` / `|meld` function by way of popular demand. Entrant ~midsum-salrux has

received job offers from both Tlon and Holium, and entrant ~litlep-nibbyt is now working on an apprenticeship with Quartus.

For more detail about the submissions, the operational group has been opened to the public: ~martyr-sarlev-sarsen/assembly-hackers

[battery payload]

The Urbit developer community has slowly grown beyond Tlon and core development. With the rise of third party application development, it became clear that discussion was spilling the boundaries of what an email list like urbit-dev could handle. We launched the [battery payload] group (~dister-dozzod-lapdeg/battery-payload) and newsletter to orient the developer community around a particular set of official resources from the Urbit Foundation (such as a Quorum choral explanations board).

Education → Grants

The live cohort courses are serving as an effective pipeline to the Grants program. Out of 501 total signups to the 2022 HSL and ASL courses, there are 49 Grant Contributors⁶ who participated in any of the live cohort courses for an approximately 10% conversion rate overall.

⁶ This metric is not purely sequential, and thus may include contributors who participated in the grants program and then returned to participate in Hoon or App School Live.

Grants

The grants program has been a mainstay of the Foundation's operations since its inception. The idea is simple — give out address space to people that contribute meaningfully to the development of Urbit. Our means of execution have <u>focused primarily</u> <u>on the contributors themselves</u> more so than the work that they're doing, which is worth digging into.

To date, grants have been mostly awarded for things built using Urbit rather than for contributions to Urbit. Our thesis is that because the loci for contributions to Urbit are difficult to find and understand, and the bar for contributing to core is very high (see our <u>plans for grants in 2023</u> for what we're doing to challenge this assumption). Because Urbit is ultimately meant for developers to build applications with, and application development has become possible to a much greater degree over the last two years, this is a much easier pathway (in relative terms) for interested contributors.

In 2022 the major change we enacted to the grants program was the introduction of **community-based oversight of grants**. As stated in last year's plan:

Our approach to running grants heavily emphasizes (and incentivizes) a team-based approach to tackling them. We believe that having people work together is more effective than having them work alone, both for those doing the work and for everyone who stands to benefit from their work. A nice side effect of encouraging people to work together in teams is that they naturally form community/ies over the shared sense of purpose.

This ended up working as we'd hoped. It distributed the load of overseeing work, raised the bar for project quality, decreased abandonment rate, and led to a number of great collaborations.

Outcomes

We'll break a discussion of outcomes into three sections:

- 1. <u>Qualitative</u> analysis of collaborations stimulated and applications built.
- 2. Quantitative analysis of outcomes produced.
- 3. Review of the <u>underlying data</u>.

Grants, qualitatively

We've felt throughout the year that our system is producing good results. Collaboration amongst contributors is very high, which has greatly decreased the amount of direct oversight that we need to have over individual projects while increasing the quality of what's being produced. Here are some examples of what we mean by that: **Native Planet.** The <u>Native Planet</u> project's <u>first two</u> phases started in the grants program. ~dalhec-banler recruited <u>~sitful-hatred</u>, <u>~mopfel-winrux</u> and <u>~nallux-dozryl</u> to his team, who were all previously funded through grants. ~sitful-hatred and ~mopfel-winrux each received a gift for having brought <u>Urbit to Umbrel</u>. The whole group is now heavily engaged in a cross-organizational team sponsored by chorus.one and led by <u>~tiller-tolbus</u> to discuss hosting standards.

Quartus. The Quartus team received 20 stars over the course of the year to fund the development of their many apps. This was their primary source of funding throughout the year. Quartus has become an integral part of the Urbit ecosystem, having produced applications like <u>peat</u> that are becoming key parts of userspace infrastructure. Two of their three contributors, ~rabsef-bicrym and ~wicrum-wicrun, have either made or are now actively making infrastructure contributions.

Productivity task force. While many successful grant projects and relationships can be captured by the grantee being hired on full time to an Urbit company, there are other more fluid relationships that grow out of grantee contributions. Often, these come in the form of larger sets of contributors working in parallel on related problems. For example, the <u>Membrane</u> (a basic spreadsheet app) and <u>%nested-goals</u> (a shareable task tracker) grant teams have crafted a shared relationship and interest in building productivity tools, and are beginning to shape an approach for uCal that will present a view into urbit calendaring that can understand the ways that people often use spreadsheets or to-do lists to manage time-dependent projects, tasks, and teams.

Bitcoin working group. Similarly, interrelated Bitcoin wallet grants have generated a tight-knit <u>community</u> of <u>contributors</u> and a pathway towards pitching Urbit to Bitcoin companies, such as Lightning Labs, on why Bitcoin needs to adopt Urbit in order to compete in the ever expanding blockchain space. This work is also offering a proof-of-concept approach for audience specific Urbit pitches and Urbit.org landing pages such as urbit.org/bitcoin. Through this cultivation of informal working groups we can leverage contributors' audience-specific understanding, and Urbit technical knowledge, to pinpoint a value prop and curate content that will help increase adoption of Urbit as a technology in that specific vertical.

Holium ecosystem bootstrapping. An interesting dynamic is also emerging around how to get independent developers building apps that offer a unified look and feel. Specifically, Holium <u>has written</u> a <u>set</u> of <u>bounties that we've</u> funded surrounding basic apps to be released alongside their launch of Realm, with different teams of contributors working on apps with a unified front end experience, but with a path towards independence rather than centralization under the Holium organization. This dynamic is still evolving, but what emerges in this semi-cooperative, semi-competitive landscape will

be interesting to follow as these contributors have been building relationships with one another but don't have any official obligations to one another, nor is the only success pathway to be hired by the company championing their grant.

Here's an incomplete list⁷ of some of the actively-maintained apps that have come out of the grants program:

- <u>Chess:</u> Fully decentralized, peer-to-peer Chess app for Urbit
- Faux: Faux, a bridge between discord and urbit groups.
- Forms: Create and fill out forms and surveys on Urbit.
- Go: The classic board-game Go, now on Urbit!
- <u>Funes Notes</u>: Funes Notes is a mobile notetaking application available for iOS
- <u>Peat:</u> Peat is Urbit's first backup utility.
- <u>Quorum:</u> Quorum is a Stack Overflow/Quora style application that allows individuals to submit and vote on questions and answers in threads.
- <u>Slam:</u> Slam is a social arcade game where you battle waves of enemies that are generated from your pals' gorae.
- <u>Trill:</u> Trill is an Urbit native microblog.
- <u>Astrolabe</u>: Astrolabe is a one-stop shop app for exploring Urbit address space.
- <u>Books</u>: Books is an application for managing your crypto wallets, transactions, and counterparties.

Grants, quantitatively

Last year we stated that our grants KPIs were: number of engaged contributors, number of repeat contributors, contributor placement in full-time positions, and completion rate, and that our goal was to *double grants volume without significant drops in retention, placement or completion rate.* We didn't exactly hit this goal, although we did have other successes outside of the original scope.

You'll notice that our metrics are heavily **contributor-focused rather than contribution-focused** — this is in line with our long-standing policy of using grants to

produce capable contributors that will go on to either join product companies or tackle infrastructure projects.

Key Metrics

- Number of contributors in grants is up by 29%. We didn't double it, but it did increase.
- Number of repeat contributors in grants is down by 3%. More total repeat contributors, but fewer in relation to the total.
- Number of grants is up by 11%. Nowhere close to doubled.

⁷ Limited to applications that have public links input on urbit.org

- Placement (hired) is down from 31% to 16%, although the total is higher. 17 hires were made over the last year, which is more than the last year. Markets likely impacted the amount of hiring.
- 48% of full-time Urbit developers started in the grants program.
- Abandonment rate dropped by 10%. 16% of proposals failed to gain community approval. We had only two abandoned grants throughout the year. Community-based approval filters out casual contributors.
- **Completion rate is 30%.** This is lower than this time last year, but many more grants are still being actively worked on.
- **53% of applicants to bounties or apprenticeships are rejected.** We're generally fairly selective.
- 67% of grants were for userspace work, 25% for infrastructure, and 8% were cultural. The vast majority were technical, and mostly app development.

See below for tables containing details on where these numbers came from.

Full-time Developers as of ~2022.11

| Total Full-time Developers | 58 |
|--|----------|
| Started in userspace ⁸ | 29 / 50% |
| Started in grants | 27 / 48% |
| Contribute to infrastructure ⁹ | 22 / 38% |
| Infrastructure that started in userspace ¹⁰ | 12 / 21% |
| Infrastructure that started in grants | 11 / 19% |

Count of Type



⁸ Includes contributors like ~hastuc-dibtux and ~tacryt-socryp who began their tenure as JavaScript programmers and are now among the most productive infrastructure engineers we have.

⁹ Either full-time or with occasional PRs to the repo, but are otherwise employed full-time working at Urbit companies.

¹⁰ Number of current infrastructure developers that began as userspace developers.

| | 2022 | 2021 |
|--|-------------------|----------|
| Distinct (grant) contributors | 110 ¹¹ | 78 |
| Total grants | 108 | 97 |
| Repeat grantees | 22 / 20% | 18 / 23% |
| Total stars granted | 203 | 150 |
| Outstanding star liabilities | 97 | - |
| Average stars per grantee | 1.85 | 3.125 |
| Completed grants \rightarrow full-time Urbit | 17 / 16% | 11 / 31% |
| Apps Produced | 20 ¹² | - |
| Acceptance rate to open listings | 47% (60 / 127) | _ |

*2021*¹³ Apprentice Proposal 2022 Bounty Total 17 34 56 108 97 Completed 37 / 38% 5 / 29% 22 / 32% 16 / 28% 32 / 30% In Progress¹⁴ 9 / 53% 15 / 44% 22 / 39% 46 / 43% 33 / 34% Canceled¹⁵ 9 / 16% 9 / 8% _ _ 15 / 15% Draft¹⁶ 5 / _ 5 / 9% 5% 11 / 11% Abandoned¹⁷ 2 / 12% _ _ 2 / 2% 12 / 12% Listed¹⁸ 1 / 6% 7 / 21% _ 8 / 7% _ Replaced¹⁹ _ 3% 5 / 9% 6 / 6% 1 /

¹¹ This number includes champions and mentors of proposals and apprenticeships.

¹² Based on a tally of grants that have known distributed apps.

Grants & Contributor Data: 2021.Q4-2022.Q3

¹⁶ A proposal that is still awaiting a champion.

¹⁸ A bounty or apprenticeship that is awaiting a worker.

¹⁹ A grant that was later superseded by a new grant.

¹³ 2021's metrics report this number as 107, but the total across the categories is actually 97. This error was discovered during compilation of this year's report.

¹⁴ The contributor is still actively working on the project.

¹⁵ A proposal that was never started due to lack of approval from the community.

¹⁷ A grant that the contributor ceased working on after receiving approval.

Combine(DAO)

The initial idea behind the Combine initiative at the Foundation was to find a good way to fund early stage companies building on Urbit. At the time, there was not a ton of outside interest from VCs or large investors and the star price was quite high, so it seemed like a good and necessary use of address space to try to seed fund small companies to build on the platform. The most obvious approach seemed to be to figure out a YC-like deal where we would have a fixed deal in exchange for helping small projects with resources such as: technical hiring, marketing, business development, strategy, etc. We were smart enough at least, to let this "YC-like" idea be a guiding principle but not to overcommit to it.

Almost immediately, we encountered problems with our plan. The most obvious is that, while we had a lot of interest, very few projects were ready for prime time. This was the first inkling of an issue that we're still dealing with, which is that Urbit has a lot of engineers and few "product" people. Second, the companies that did need investment and were investable were at all different stages. In other words, the deal flow was heterogeneous enough that trying to have a fixed approach hamstrung us.

Even so, we made some initial investments in Holium, Mars Review of Books, Uqbar and dcSpark. We also facilitated the continued funding of companies like ~tirrel, who operated as vendors to the UF.

The DAO transition

The decision to convert the Combine from an initiative of the Foundation to a DAO was borne of a few related considerations. The first was that we began to hear from other smaller investors in Urbit that they were having difficulty taking full advantage of the information that was being shared within the internal Combine group. The second, obsolete now, was the idea that it would be beneficial to have an actively-managed treasury to be able to participate in the WSTR market as part of a liquidity solution. The third was that, at this point, the price of stars had halved from when we began considering the initiative in the first place. It was suggested then to create a DAO to pool the assets of a few investors. Doing so would also provide us with some learnings about creating and running such a DAO on Urbit, a use case that had been spoken about at great length, but hadn't actually materialized yet.

The very week that the Combine DAO was legal and funded, the markets slid again and the star price declined from its place at roughly 10k to 5k, immediately halving the treasury in dollar terms. In retrospect, keeping the treasury in USDC would have been a better decision. Since our Treasury was held in Stars, it made the mechanics of funding projects onerous. The worsening market situation made funding anything difficult as well. While the Foundation and ~poldec-tonteg held the majority of the votes, we had an internal objective of making decisions as close to unanimous as possible.

DAO-to-DAO (D2D) Sales

While initiating new investments was difficult, we decided to direct resources toward what was increasingly seeming like a blocking issue: the lack of product teams on Urbit. The fact of the matter is that, while Urbit has great engineering and developer pipelines, this doesn't actually translate to product teams. Product teams are necessary for deal flow.

We decided an elegant solution was to do more direct outreach to DAOs, specifically getting them into the Holium sales funnel. This offered the dual benefit of supporting an ecosystem company as well as building relationships with DAOs. DAOs, we reasoned, would be good sources of talent looking to build novel things in Web3. Unlike direct sales to organizations, many of the most valuable members of DAOs are members of other DAOs and/or have other projects of their own. This means that outreach to this demographic is high leverage.

This outreach was successful. We were able to form a working relationship with Tribute Labs, who are currently spinning up a DAO to focus on Urbit projects. They run several high profile DAOs, so getting them to adopt Urbit as their DAO stack stands to be very beneficial. Several of the DAOs that we approached are in the Holium sales funnel and at least one has become an active investor in the Urbit ecosystem.

Winding Down

Two things have changed over the course of the year that have led us to decide to unwind the Combine and return the stars allocated.

The first is that it has become clear that there is now no lack of investor interest for Urbit companies. The Combine DAO structure membership is universally able and willing to invest separately, so there is less advantage in practice to pooling the funds.

The second is that, with the hiring of a CTO, the Foundation needs to focus more of its budget on development. This was organizationally impossible a year ago, but has quickly become a reality.

Getting out of the funding business will free us up to focus on helping to level up companies working in the ecosystem. This will put them in a better position to take advantage of the funding available.

Since no deals were closed with the Combine DAO, unwinding it is fairly straightforward. ~poldec-tonteg will shift more of his own personal assets into the Combine and turn it into a personal project, using it as a vehicle to fund smaller and creative projects that are at risk of being stranded by the pivot of the Foundation out of funding.

Marketing

Early on in the year we decided to take marketing for Urbit seriously. Three factors influenced this decision:

- 1. The then-widespread belief that the teams responsible for working on Urbit were terrible at marketing the project,
- 2. Balaji's input at our first board meeting that we, those building Urbit, had never made a serious attempt to "win" with users, and
- 3. Our view that the Urbit ecosystem was advancing Urbit to a state in which significantly greater adoption along multiple axes.

By early 2022 Urbit had existed for roughly a decade and, during that time, had been subject to critique during a period in which it was under active development with little to outwardly show. Urbit this year is a qualitatively different thing than the Urbit of 2013-2021, yet general perception of the project was substantially lagging. Our thesis was that, with enough attention drawn to the project in 2022, we could shift the dominant narratives in a significant way, thereby attracting more developers, users, entrepreneurs and investors at a time in which they can be productively deployed.

Urbit NYC

The first step in this process was Urbit NYC, which was predominantly a publicity stunt designed to draw attention by highlighting our eclectic community. We engaged a publicist to gain the NY media's attention which resulted in the Red Scare episode and a (still ongoing) relationship with Kyle Chayka to place a piece in the New Yorker. The week of sponsored events cost a bit above \$50k (for reference, this is the cost of a sponsored image in a new tab of the Brave browser for a single day), resulted in a surge in traffic to urbit.org in Twitter in excess of either of the Assemblies, and had the added benefit of getting much of our community and the UF team together IRL.

Show, Don't Tell

With increased attention hitting the project it became clear that we needed to tell the Urbit story better. At the time many expressed the need for a better explainer of Urbit to replace the aging <u>Overview</u>. We chose instead to take the stance of "show, don't tell." Instead of explaining the project — a well-explored problem space that never seemed to produce satisfactory results — we instead sought to demonstrate what was happening on Urbit by highlighting the identities, applications and communities that were being built.

Public Links

Our first step was to release Public Links on April 30th ahead of Urbit NYC, which allowed Urbit IDs, apps and groups to have public pages on urbit.org that could be linked to from

the outside web. We paid particular attention to Opengraph metadata to ensure that shares on Twitter and other social platforms would be as enticing as possible.



Everyone 🗸

https://urbit.org/groups/~ladtem-filmyr/urbit-ny-week



O Everyone can reply

Each page provides instructions on how to enter the network and either install an app, join a group, or acquire an ID depending on what type of page is being viewed.

Ecosystem Section

Three months later (resources shifted to the launch of developers.urbit.org) we released the <u>ecosystem</u> section, which featured prominently on the landing page of urbit.org, which turned <u>Public Links</u> into a content directory and showcased relevant new content via the monthly <u>Ecosystem Spotlights</u>.

Ecosystem

The Urbit ecosystem is comprised of a wide variety of individuals and organizations, including developers, DAOs, podcasters, and more.





Since launch on Jul 25, 2022 pages under urbit.org/ecosystem account for 22.1% of all urbit.org pageviews²⁰ and 4.8% of all unique pageviews²¹. This implies that each unique visitor that visits the ecosystem section views several distinct pages.

Road to Assembly

Assembly planning started in March with venue selection, team recruitment and research. We decided early on that this event would be the culmination of our marketing strategy that placed Urbit's ecosystem in full view. There's enough to talk about here to devote a whole section to, so see <u>below</u>.

²⁰ 172,233 views out of a total of 778,743 views across the site as of Nov 8, 2022

²¹ 6,449 uniques out of 40,564 uniques as of Nov 8, 2022

Assembly 2022

In 2021 the Tlon executive team talked about preparing to give Urbit a real launch. Assembly 2021 wasn't that launch, although it definitely was a crucial step in preparing for such an event. At the start of the year we anticipated that Urbit's growing ecosystem would be the right occasion for a proper launch event, and that a launch would further help the ecosystem accomplish what it needs to to be successful: ship product, build awareness, raise money, and hire people.

As I discussed in my <u>opening talk</u> it didn't end up being possible to really "launch" Urbit at any specific event, but commemorating the occasion served much the same purpose: the dominant narratives about Urbit have been changed, developer numbers are on the rise, and Urbit companies are either directly benefiting from this or are on track to.

Results

We've seen numerous spikes across the board in metrics that we track, although there's still much that's hard to see or still developing. Here's what we know:

Publicity has been good

We've had <u>six pieces</u> written about Assembly all of which have been generally positive, some more so than others.

Grants and engaged developers saw a spike

10 grants have gone live since Assembly and we've received 16 applications to work on bounties or apprenticeships. The second batch of HSL received 278 applications, which is more than the first one, which is surprisingly high given that the first HSL came after more than a year without any courses provided.

Hires are getting made

At least four hires in the ecosystem have been made that we know of with a few more under discussion.

Social saw a huge spike

Keyword mentions of Urbit on Twitter rose from 1,599 in June, to 4,652 in September, to 5,122 in October.²²

²² Measured through <u>https://sproutsocial.com</u>

Volume ... Identify conversation patterns and changes within this reporting period. Volume Total By Month **(**) 6,000 5,000 4,000 October 2022 3,000 Topic Volume 5,122 2,000 Click Chart to View Messages 1,000 0 JUN JUL AUG SEP ост 2022 Topic Volume

Follower count from July 1st (when we started advertising Assembly) to October 31st rose 12.4% from 28,097 to 31,578 with 1.3 million impressions made²³.

Survey Results

We distributed a survey after Assembly that received 97 responses. Here are the results compared with last year's:

| | 2022 | 2021 ²⁴ |
|-----------------------------|----------|---------------------------|
| Overall Satisfaction | 8.4 / 10 | 8.8 / 10 |
| Well-organized | 8.6 / 10 | 8.1 / 10 |
| Utility of information | 8.1 / 10 | N/A |

We also asked people what they liked the most, what they liked the least, and how they felt about the speakers. Here are the summarized findings:

What did you like the most?

- **71% people/community**: mingling and having the space to do so, diverse personalities
- 25% speakers: variety of thought, different perspectives and topics
- **20% environment:** the aesthetics of the venue, Miami in general, furniture, quality of production

²³ Measured through <u>https://sproutsocial.com</u>

²⁴ 79 responses were recorded for last year's survey

- **19% technical talks**: the presentations on technical subjects and access to technical people
- 18% demos: the things that Urbit companies showed off
- 18% cultural talks: the non-technical talks and panels about art, culture, media
- 4% smoking: Joe's tobacco stand

What did you like the least?

- 32% A/V: the background noise, acoustics, glare
- 30% weather: humidity, heat, lack of A/C
- 10% cultural talks: the non-technical talks and panels about art, culture, media
- 9% location: the venue's lack of child-friendliness, Miami in general
- **8% speakers**: those that were shilling their own thing, were unprepared, weren't on Urbit
- 6% political speakers: edgelords and right-wingers called out specifically

Budget

Our original budget was \$500k and the cost for the event ended up being \$750k. This was primarily due to overages in catering, furniture, AV, decor and hotels. The venue required certain vendors that ended up costing more than expected, and Miami is an expensive city in general.

We did better in sponsorships than expected, pulling in \$230k (versus \$150k expected). Ticket sales did worse than expected, pulling in a total of \$161k — this was due to market conditions that resulted in sale price of star passes and decreased demand for stars in general.

Public Goods

Throughout the year we built and maintained a number of products that can be considered public goods: the <u>Network Explorer</u>, <u>Trial Hosting</u>, and <u>Login with Urbit</u>.

Network Explorer

A lot of under the hood work has happened this year to make the network explorer useful.

First we had to refine Azimuth data in light of the launch of L2. This posed unique challenges around representing spawned ships versus those with networking keys set, as all L2 ships have their networking keys set by default. Drawing a direct comparison with L1 ships required that we not consider networking keys set on L2 ships until the second time.

The second block of work was on showing the number of onlineships on the network. This requires pinging each ship with networking keys set to discover whether or not it's online. Through this process we hit the limits of the Ames network and fixed a few issues in arvo, eventually producing reliable data on real network growth.

Trial Hosting

We quietly launched a new way to join the network at Assembly that provides access to a fresh moon for one day. After a day the moon is breached. This serves as a way for newcomers to trivially try Urbit at no cost.

We've seen moderate usage of this since launch and aren't completely sure how it will evolve, although we can imagine integrations with hosting providers or webterm integrations into the docs as potential next steps. Regardless, the project requires no ongoing maintenance and costs little to keep running.

Login with Urbit

We built two applications leading up to Assembly that enable a traditional Web2 application to provide an Urbit-based authentication mechanism.

We built this with Vienna Hypertext as the pilot case in time for its debut at Assembly, and since then Vienna has received upwards of 500 signups through Urbit and reports those users as among the most active on the service. We've since finished a complete rewrite of the application to further simplify the process, which debuts during Urbit Developer Week.