RESEARCH REPORT

Decentralization Off The Shelf: 7 Maxims

Authors

Eileen Wagner (Simply Secure)
Karissa McKelvey (Digital Democracy)
Kelsie Nabben (RMIT University / DXOS.org)

With funding support from SamsungNEXT, DXOS.org, and the Shuttleworth Foundation.

Executive Summary

A growing number of technologists are challenging the consolidation of power over information systems by creating decentralized protocols and applications. Where government and corporate control are causing harm, decentralized technologies can bring about autonomy, resilience, and equity. However, there is a significant gap between decentralized protocols and the applications that users want to adopt. Until now, there hasn't been a systematic survey of the needs and challenges for the people and projects involved.

Through a series of interviews and focus groups with technology designers and builders, we have identified 7 areas where projects can improve their own practice; where targeted research is necessary; and where funders need to step in to enable collaborative innovations.

1. Collaboration. Projects need to collaborate on a stronger, galvanizing narrative by leveraging strategies from campaigning and movement building.

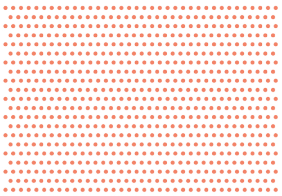
2. Design. Designers need novel patterns and approaches for driving the development of decentralized protocols and applications.

3. Infrastructure. Funders need to prioritize the independent verifiability and resilience of shared digital infrastructure, such as app stores, browsers, hardware, and networking.

4. Developer onboarding. Developers need more accessible education materials and training modules for decentralized architecture patterns.

5. Trust models. Projects need to design with vulnerable populations in mind, and adopt privacy and safety frameworks specific to decentralization.

6. Sustainability. Funders need to innovate on strategies for sustaining projects and attracting talent.



7. Governance. Technologies need to define governance models to align value and build a sustainable culture for long-term project value and stability.

These themes emerged from our research, as well as years of shared experience working in decentralization as protocol developers, UX designers, and researchers.

This research is part of "Decentralization Off The Shelf,"[2] a collective initiative to identify needs, synthesize priorities, provide resources, and coordinate efforts to further the development and deployment of decentralized technologies. By addressing these issues, we aim to support the design and development of better user-facing tools that are backed by decentralized architecture and increase the overall quantity and quality of decentralized applications.

1. https://decentpatterns.xyz

Table of Contents

Executive Summary 2

Table of Contents 4

Introduction 5

- Key Terms 6
- Approach 8
- Attribution 8

Maxims 9

- **Collaboration.** Recoginize the context 10
- **Design**. New user experience patterns 12
- **Infrastructure**. Independent verifiability and resilience 16
- **Developer onboarding**. Training and education 19
- **Trust models**. Safety and privacy 23
- **Sustainability.** Attracting and retaining talent 26
- **Governance.** Decision-making, stability, and participation 28

Conclusion 31

Appendix 32

- Research Participants 32

Introduction

"Decentralization" is a loaded term. At face value, it describes technical and social architectures that are not centralized—either because they have no central control or authority, or because they have many centers of power. But digging deeper in the research areas, developer communities, and political movements, you will not find a uniform or consistent description of the values and approaches to decentralization. On what level (networking, application, providers) should decentralization happen? Is decentralization best realized by peer-to-peer or federated models? These and many more questions divide the community.

Perhaps more polarized than technical disagreements are political ones: practitioners see decentralization both as a way towards self-determination—gaining independence from large corporations or governments—as well as means towards more equity and co-ownership. Some projects focus on revolutionizing the banking economy, others fight against censorship and surveillance of human rights defenders, while yet others work to make science more open and efficient. Many practitioners believe that decentralization is the next step of human progress, but some also believe it will be the last working technology in a dystopian future. Decentralization is, to say the least, a mixed bag of ideologies.

While the motivations for decentralization are heterogenous, the goal, however, is clear: better usability and more adoption. Despite the rapid growth of some decentralized protocols, key challenges remain across the domains. It is our hope to identify those common challenges and work towards generalized solutions. This report serves as a starting point, and we welcome comments and feedback from the community.

Key Terms

Decentralization
Network architecture that avoids reliance on a single party. Encompasses peer-to-peer, blockchain, federated, and distributed technologies that involve many individual users.

Peer-to-Peer (p2p)
Peers make a portion of their resources, such as processing power, disk storage or network bandwidth, directly available to other network participants, without the need for central coordination by servers or stable hosts. Popularized by BitTorrent, Napster, and Bitcoin.[2]

Federated
Federation allows separate deployments of a service to communicate with each other through a common protocol, for instance a mail server run by Google federates with a mail server run by Microsoft when you send an email from @gmail.com to @hotmail.com.[3] Each deployment may host multiple users.

Blockchain
A distributed ledger that can record transactions between multiple parties efficiently and in a verifiable and permanent way.[4]

Distributed systems
Academic topic within the discipline of Computer Science which is concerned with the design of computer systems that consist of many individual computers connected over a network. Peer-to-peer networks and blockchains are examples of distributed systems architectures.

WebRTC
A protocol standard for establishing connections in a web browser where data passes directly between users.

TCP/UDP
The two foundational transport protocols used on the Internet. Common protocols used to send data between two computers.

DHT
Distributed hash table, used in some projects to connect peers to each other by storing information in the form of key-value pairs in a distributed manner.

---

2. Rüdiger Schollmeier, A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications, Proceedings of the First International Conference on Peer-to-Peer Computing, IEEE (2002).
3. Sheth and Larson (1990). "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases". ACM Computing Surveys, Vol. 22, No.3. pp. 183–236.
4. Iansiti, Marco; Lakhani, Karim R. (January 2017). "The Truth About Blockchain". Harvard Business Review. Harvard University. Archived from the original on 18 January 2017. Retrieved 17 January 2017

IP address
A number of a computer or network which is unique and thus can be used to address it.

Hash
A number, usually displayed as a string of letters and numbers. It can serve as a ‘fingerprint’ uniquely identifying data.

UX
User experience, the overall experience of a person using a product or a service, especially in terms of how easy it is to use.

End users
Mythical creatures that have been known to visit your website or app. We’ll be using “users” contextually for anyone using a product or service (including a protocol documentation), and emphasize “end users” where we mean individuals and organizations that ultimately adopt an application.

Approach

This report is based on two independent research efforts in mid- and late-2019, conducted by an interdisciplinary group of designers, researchers, and technologists: Kelsie Nabben (RMIT University), Eileen Wagner (Simply Secure), and Karissa McKelvey (Digital Democracy). We investigated the technical, design and organizational challenges as well as aspirations in the decentralized technology community.

The methodology of this research is qualitative, involving focus groups, interviews, and participant observation. In total, we interviewed 57 protocol developers, project leads, and designers. In addition, we facilitated four workshops (Dat UX workshop, May 2019 Berlin, Redecentralize and MozFest, October 2019 London, P2P Summit at Ethereum DevCon by DXOS.org, October 2019 Osaka), with a total of around 85 participants. Based on these findings, we surfaced and synthesized key themes.

All quotations in this report are attributed to our research participants.

This work was funded by SamsungNEXT and DXOS.org, with organizational support from Simply Secure and Digital Democracy. A Shuttleworth Flash Grant also enabled some travel.

Attributions

Thanks for helpful comments on the draft to Niels ten Oever, Dan Hassan, Peter van Hardenberg, Antonela Debiasi, Ricardo J. Méndez, Iryna Nezhynska, Gerben, Benedict Lau, Paul d'Aoust, Jan Dittrich, Jay Graber, Ross Schulman, Abbey Titcomb, Scott Moore, substack, Paul Frazee, Betsy Cooper, Matthew Slipper, Darius Kazemi, Matthew Wild, Dietrich Ayala, Przemysław Idzkiewicz, Harry Lachenmayer, Chris Adams, Paul Gardner-Stephen, Mark Nadal, Pospi, Vincent Ahrend and Allon Bar.

Thanks to Georgia Bullen for strategy and fundraising support.
Thanks to Cade Diehm, Kira Oakley, Jay Graber, Irina Bolychevsky for continued camaraderie.
Thanks to Ignatius Gilfedder for a stunning visual design, and Ngọc Triệu and Vincent Ahrend for laying it out in print and web.
Thanks to Allon Bar for the name "off the shelf".
Thanks to DXOS.org for initiating the P2P Summit, arranging some interviews and financial support. Additional thanks to SamsungNEXT and the Shuttleworth Foundation for financial support.

Maxims

1 Collaboration

Recognize the context.

Decide on whether and how to communicate your values; a value proposition can be political, too.

Collaborate with other projects that are targeting the same users and use cases, and pattern match with regard to design and architecture choices.

Learn from adjacent movements; work with community organizers, coalition builders, and campaigners.

A stronger, galvanizing narrative for the decentralization space is needed. This could be a number of aspects: cost-saving, ownership, agency, openness, resilience, collaboration, autonomy, and self-determination are just a few of the possible value propositions. Adjacent movements have developed strategies for challenging the status quo, and the decentralization ecosystem will benefit from adopting these strategies.

Design and marketing strategies. Projects need to decide whether and how to communicate their values as a key feature of their design and marketing campaign. While almost all projects think of their work as political, opinions diverge when it comes to user-facing communication. For projects that identify with concrete political goals, such as building an alternative Internet, it is crucial for users to acknowledge the politics involved. Campaigns like the Tor Project's "Take Back the Internet"[5] come to mind. For others, this narrative is alarmist and alienating, and designs omit a political framing to focus solely on meeting user needs, whatever they may be: "A decentralized Spotify should look and feel like Spotify."

Pattern match. Every project has its own considerations for target users and use cases while aiming to push the boundaries of what is possible. Each use case begs for a different design, from architecture to user interface—and yet there are patterns in the ecosystem that exist across projects. Projects should investigate their own work to make these design decisions explicit and seek collaboration with other groups that share similarities.

Movement building. For decentralized applications to reach their full potential, there needs to be much more collective strategizing, coalition building, and community organizing—not just around the particular technologies, but around the overarching mission and values of the movement towards decentralization. Given the economic threat that decentralization poses to existing power structures, we can expect more measures—legal or political—to be put in place as decentralized technologies grow in popularity. Yet many projects are focused on their products and not on this wider concern.

> To change the distribution of authority, we should study how authority works. We should ask: When is authority within a computer network appropriate? How should it be assigned? Once assigned, how can it be constrained?[6]

5. https://blog.torproject.org/take-back-internet-us
6. Frazee, Paul. Information Civics: Deconstructing the power structures of large-scale social computing networks. https://infocivics.com/

2 Design

New user experience patterns.

Apply best practices in human-centered design, and **test early and often** with end users—even when you are not developing an end user-facing product.

Use accessible and commonly used language and interfaces in the ecosystem to reduce necessity for re-explanation of familiar and common patterns.

Develop protocols as products through application co-development.

Every piece of software needs some amount of interface, content, and service design. This is no different in decentralization. What is different, however, is that decentralization introduces concepts and scenarios that are diverging from today's dominant, centralized paradigms. Decentralized technologies require new, generalizable design patterns.

Onboarding and explanations. It is difficult to onboard and manage expectations for new users given the complexities introduced in decentralized contexts. Areas that are most confusing to end users include:

Mental models of decentralization and networking in general
- What data is stored where?
- Where are the boundaries between apps, archives, mounts?

Relationship between protocols and clients
- What is a protocol, what is a client?
- What does it mean for a protocol to have multiple clients? (and vice versa)

Agency & identity
- Who else knows who I am? Can I be anonymous?
- What happens when I lose my password?
- Can I use multiple devices? Can I share a device with others?

Security & authentication
- Who can see what, and for how long?
- Why can't certain data be deleted?
- What happens if a device is lost?

Online status & synchronization & availability
- Does this network require the Internet to work?
- What does it mean to optionally use the Internet?
- Will my content be available at all times? How reliable is this service?

Licensing & intellectual property
- What does it mean to have other people's data on my machine?

Governance & content moderation
- Who has control over shared content and infrastructure?
- How can I block someone?
- Who can I call if I run into a problem?

The difficulty starts with naming and descriptions, as many projects need to add context and background to their onboarding.

> Every decision you make as a designer is an abstraction and comes with trade-offs. Calling something a 'location' instead of a 'device', as Apple has done recently, is an example of a trade-off between avoiding jargon and adding ambiguity.

> Words can be different, but the concepts have to be the same. Think about how you learned about volcanoes as a child vs as an adult. You are learning about the same concept, just in different ways.

A human-centered approach[7] to application and protocol design means addressing these concerns by (1) listening to users to understand existing social practices, habits, and mental models, (2) following design heuristics around user control and error prevention,[8] and (3) iteratively user testing different explanations. Metaphors and stories best describe new technology in an accessible and clear way. There is also value in sharing terminology across different projects, when it comes to establishing foundational concepts such as seeding or network health.

Protocols as products. The protocol's specifications impact user experience. It is wrong and harmful to think that end users are not affected by protocol development.[9] For instance, features such as gossip and eager content replication make it difficult to delete information on the network, making information control effectively impossible.

Unusual safety and privacy properties aren't the only aspect that affect end users. The overall design approach, such as openness and extensibility, will also impact usability. For example, XMPP defines a basic core protocol on top of which different extensions ("XEPs") are defined for different features. This improves the agility of the protocol; however, with an open client ecosystem this can lead to user confusion when different XMPP clients cannot fully interoperate due to missing or incompatible extensions.

---

7. https://www.oreilly.com/content/ux-for-beginners-key-ideas/
8. https://www.nngroup.com/articles/ten-usability-heuristics/
9. https://trac.tools.ietf.org/html/rfc8280

In line with open source development processes overall, we observed a general gap in design and product thinking. Reasons for this vary, but overwhelmingly teams are rarely thinking about protocols as products: that protocols should be treated as products for application developers, applying user stories and tight feedback loops to inform development.

One model to design, develop, and test protocols is to "live in" them during development, as done by Secure Scuttlebutt.[10] This would avoid the build now, deploy later approach. A more systematic model for doing this is Matrix's development of Riot, their "flagship client".

> Riot allows us to push Matrix forward by experimenting with any suggested changes to the protocol. This allows us to get down the ivory tower of protocol development.

By having an in-house, go-to client that is being co-developed with the protocol, end user needs and pain points are considered during the protocol development cycle. One challenge with this approach is that applications can become a myopic version of what protocols can do—so it might be worth thinking about developing a variety of applications at the same time.

> Protocols that do not co-design with a target user base risk irrelevance.

10. https://scuttlebutt.nz/

3 Infrastructure

Independent verifiability and resilience.

Fund research & development for resilient infrastructure, including connectivity, hardware, storage, and end-to-end protocol testing.

Design infrastructure layers with modular interfaces to enable the reusability of components across protocols.

Avoid centralized dependencies when possible, and otherwise consider them to be a key limitation of infrastructure design.

Decentralized protocols are still dependent on centralized points of failure, such as cloud infrastructure, web browsers, app stores, and proprietary hardware. App stores, networking equipment vendors, and web browsers need to adopt new technology policies that enable decentralization. Investment should be made to improve independent verifiability and resilience of this common digital infrastructure.

Connectivity. Projects should collaborate on connectivity toolkits, making them flexible enough for a variety of use cases. The widespread adoption of NAT (Network Address Translation) and other middlebox technologies negatively influenced users' ability to connect directly to other users without a centralized server or platform. To mitigate this, projects deploy cloud services that introduce two peers to each other or relay content over DNS/STUN/TURN/ICE. These strategies provide a better end-user experience, but also create a dependency on centralized infrastructure maintained by one or more third parties. This introduces points of failure and causes leakage of metadata, such as IP addresses and application data. Although connectivity is a common issue, many projects are still using bespoke solutions, as no single connectivity library or framework has gained widespread traction among decentralized protocols.

Storage. Applications should remove the dependency on external storage services by incorporating peer-to-peer protocols directly when possible All peer-to-peer applications suffer from the potential unavailability of application data, especially when used on multiple devices (e.g., mobile). To mitigate this, centralized servers (sometimes called "gateways") can be used to store application data. Because of these challenges, some decentralized projects abandon peer-to-peer altogether and adopt a federated approach instead. However, it is often unclear how these services prioritize data retention policies, on-disk encryption, and metadata access. Some projects are aiming to enable on-disk encryption for decentralized storage, but it is considered difficult to provide good user experience and performance in real-world scenarios.[11]

Where popular browsers[12] like Google Chrome, Apple Safari, Microsoft's Edge (IE), and Mozilla Firefox are moving slowly on incorporating peer-to-peer protocols, other browsers are filling the gap. Beaker Browser has built-in peer-to-peer networking and storage for Hypercore (Dat), and Brave Browser has an extension for direct access to IPFS and Tor, without gateways or relays.[13] There are a few individual campaigns to improve WebRTC as well as support TCP and UDP directly in web applications to help close these gaps.[14]

Modularity. Developers should craft protocols on a variety of modular libraries that can be extended and improved over time without changing underlying protocol behavior. This is particularly important for the long-term sustainability and flexibility of decentralized technologies, which can be more difficult to upgrade over time as no single party can control the entire network. Developers often design protocols and libraries for a particular use case, rather than for general use. Because of this, there are few protocols that are flexible enough to be repurposed in different scenarios. It is common software engineering practice to refactor components for use outside of their original purpose, but early design decisions are oftentimes binding in decentralized networks.

---

11. Such as IPFS, Filecoin, Dat, DDRP, and Swarm
12. https://beakerbrowser.com/
13. https://brave.com/
14. https://discourse.wicg.io/t/filling-the-remaining-gap-between-websocket-webrtc-and-webtranspor/4366

> The immediate priority is organic, modular libraries that are designed to do a specific set of things and are usable, that do not care about the context in which they're used and aren't trying to comply within a framework.

Proprietary hardware and app stores. Investment should be made in open source hardware and app stores to improve independent verifiability and resilience of our shared infrastructure. Proprietary software and hardware is still a major dependency, which is especially concerning for application security.[15] This is a crucial problem not just for decentralized technologies, but for the security of the breadth of digital infrastructure that runs the Internet as well. This dependency on proprietary hardware and app stores also can lead to increasingly locked down and unpredictable deployment roadmaps.

> In order to remove any dependency on centralized infrastructure we decided to prototype some low-cost hardware devices.

15. https://www.theverge.com/2020/3/6/21167782/intel-processor-flaw-root-of-trust-csme-security-vulnerability

4 Developer onboarding

Training and education.

Provide better user experience for application developers with a focus on toolkits, query languages, and modular libraries that are designed with minimal assumptions.

Describe and specify how protocols are supposed to perform and what they are designed to do, in terms of trade-offs and target use cases.

Support the creation of accessible content as a way for projects to share more about their approach e.g., libraries, UX/UI patterns, design decisions, and an ecosystem map for developers to get started.

Decentralized protocols are still dependent on centralized points of failure, such as cloud infrastructure, web browsers, app stores, and proprietary hardware. App stores, networking equipment vendors, and web browsers need to adopt new technology policies that enable decentralization. Investment should be made to improve independent verifiability and resilience of this common digital infrastructure.

Documentation, libraries and accessible content. To attract and engage more developers, there is a dire need for clear and accessible educational content. Current materials are highly technical and disparate, with a steep learning curve requiring significant time investment. Materials should cover more than simplistic examples and API coverage, but also include behavioral quirks and known issues. This will also benefit ongoing research and development across the ecosystem as developers can better understand what's been done before, what's worked, and what hasn't.

> Because it's a decentralized space that anyone can participate in, there's a lot of education that needs to be put out in terms of best practices when you're engineering and developing these systems since it doesn't work the same way as a traditional client-server architecture.

Technical architecture design. To improve transparency and developer experience, protocols should produce audits about the benefits, trade-offs, and shortcomings present in the technical design. Ideally, these audits would be performed by an independent body, composed of experts from design, product, distributed systems, and security. There needs to be a technical pattern library that highlights the available approaches and trade-offs, giving practical tools for developers to get started.

Significant resources have been poured into some shared technical architectures and frameworks, such as libp2p, with limited success.[16] While funding and coordination efforts were found to be beneficial, those design decisions are not standardized across the ecosystem. This is due to the complexity of the library, continuous development across multiple languages that is difficult to implement, and the cost to pursue different design directions once time and effort was already invested.

16. https://discuss.libp2p.io/t/report-a-study-of-libp2p-and-eth2/229

Consistency and Availability
It is established academically that distributed systems do not all have the same guarantees of consistency and availability. This leads to disagreements about how an application should behave when there is network latency.

- Strong consistency.
  - Main Benefit: The most recent read is the latest information. Useful for transactions between untrusted parties.
  - Main Drawback: Slower user experience.
  - Example: Blockchains
- Eventual consistency.
  - Main Benefit: Faster user experience and higher availability.
  - Main Drawback: The most recent read could be out-of-date or require conflict resolution.
  - Example: CRDTs[17]

Specifications. Rather than documentation, which can be a higher-level overview and introduction to a particular implementation, a specification can be useful for lower-level systems developers. This helps support the development of new implementations and standards. Many protocols do have specifications, but they can easily become out of date, are difficult to find, or are written with inaccessible insider framing. While standards are important for compatibility, interoperability, and scale, these will follow on from better documentation, stable specifications, and maturity.[18]

> We iterate on protocol decisions early and play with them, do testing, and once we feel good about the functionality we will create a specification.

17. Shapiro, Marc; Preguiça, Nuno; Baquero, Carlos; Zawirski, Marek (2011), Conflict-Free Replicated Data Types (PDF), Lecture Notes in Computer Science, 6976, Grenoble, France: Springer Berlin Heidelberg, pp. 386–400, doi:10.1007/978-3-642-24550-3_29, ISBN 978-3-642-24549-718.
18."open specification maintenance is not important to initial success...and standardized later." https://tools.ietf.org/html/rfc5218#page-3

Standard terminology and UX elements. Projects should adopt a common framework for language referring to elements across protocols. Numerous different terms are being used for similar concepts within the context of decentralized applications. Aside from perhaps email, there are few well-resourced and fully-tested decentralized products with widespread adoption as of the time of this report. The ecosystem is largely still learning what decentralization means and experimenting with various approaches.

> We are often running into fundamental issues around changing online/offline status (e.g. message bursts), access control, key management, and just explaining what networks are.

Because of this, projects have developed many diverse and creative UX solutions for decentralized technologies. This has resulted in a large variety of terminology, adding to the confusion for both new developers and new end users.

Do	Don't
CatChain devices are called "peers" that add "messages" to their "feed" and share them over a "topic" using a Distributed Hash Table	CatChain devices are called "cats" that add "scratches" to their "post" and share them within a "backyard" using a "DistributedCatTable."

5 Trust models

Safety and privacy.

Train more developers and designers on privacy and security approaches and skills, and include threat modelling in protocol design.

Move from threat models to trust models; understand that every network makes assumptions about trust, and trust may not necessarily scale with the network.

Manage expectations by communicating clearly what a technology can and cannot do and who they are trusting with their data and metadata.

Fund research and development on security and anonymity innovations in networks.

We see increased interest in decentralized technology as a response to surveillance capitalism, security breaches,[19] and tech monopolization; but few projects offer the security and service that end users are looking for from an alternative. Unlike large technology companies, many decentralized projects do not have the resources readily available to implement, communicate, and research good privacy and safety guarantees. Threats to user privacy and safety need to be understood by system designers, application developers, and end users.

19. Zuboff, Shoshana (2019). The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power. New York: PublicAffairs.

Privacy trade-offs. Deeper investigation and research is needed in decentralized privacy, such as mixnets, privacy-preserving DHTs, and encrypted relays.[20] Although privacy may be possible for many protocols, in practice this requires up-front investment of significant resources, and can result in a significantly slower user experience. This trade-off between anonymity (e.g. Tor[21]) and content availability (e.g. IPFS[22]) means that more devices having access to data can increase the attack surface. IP addresses and sometimes even sensitive user data are shared with the operators of decentralized infrastructure. It is possible to enhance privacy, integrity, and availability together, but these features affect system complexity, the properties provided, and degree of decentralization.[23]

Communication. Onboarding processes should correct safety and privacy expectations by providing clear, timely communication. Very rarely are privacy and safety concerns communicated clearly to users. Some protocols have out-of-date or incorrect statements on their websites. Concepts around "decentralization", "open source" and "community-driven" can sometimes create false expectations around security properties.

> From the role cryptocurrencies play in emergent dark web marketplaces, to the well-funded efforts by protocol developers to create faster and more resilient networks, the decentralised community seeks to antagonise the status quo whilst making significant tradeoffs that refuse to acknowledge how societies directly threaten their communities.[24]

20. Claudio A. Ardagna; et al. (2009). "Privacy Preservation over Untrusted Mobile Networks". In Bettini, Claudio et al. (eds.). Privacy In Location-Based Applications: Research Issues and Emerging Trends. Springer. p. 88. ISBN 9783642035111
21. https://www.torproject.org/
22. https://ipfs.io/
23. Troncoso, Carmela, et al. "Systematizing decentralization and privacy: Lessons from 15 years of research and deployments." Proceedings on Privacy Enhancing Technologies 2017.4 (2017): 404-426.
24. Diehm, Cade (2020). This Is Fine: Optimism and Emergency in the Decentralised Network, New Design Congress.

From threat models to trust models. Another way to approach threat modelling is trust modelling: how many people and organizations must one trust to use a certain technology? While this is easy to answer in a centralized context (main operator), the model gets more complex in a decentralized context (infrastructure operator(s), every other peer on the network, etc.). Once these assumptions are made explicit, it will be evident that there is no single solution for all use cases, and so some target users will need to be prioritized.

Threat modelling doesn't just concern server-side security guarantees. Understanding the threats users face in a shared network—from domestic surveillance to online harassment—is essential in designing safe and appropriate technologies.[25] Content moderation is a social problem that requires social solutions; but any application is operating within the technical boundaries defined by the protocol they use. For example, if it is technically impossible to delete content on another peer's device, then social rules enforced by the application design must take this into consideration.

Projects either run default public infrastructure or force users to supply their own infrastructure. This is a trade-off between increased adoption rates and clearer boundaries of trust, ownership, and control. Having fewer people on the network might result in people taking on more trust and responsibility. "What does a client look like if your design constraints are (1) less than 50 people on the network, (2) no scaling, and (3) offline-first?"[26] Understanding that scale and trust are inversely correlated is an important first lesson in designing a decentralized protocol.

25. Wilson, Molly (2020). Design Under Pressure. https://simplysecure.org/designunderpressure/
26. https://*runyourown*.social

6 Sustainability

Attracting and retaining talent.

Research and develop community-driven funding models as they relate to sustainability of people, projects, and protocols.

Utilize bounty platforms and microgrants as an experimental funding mechanism to test appetite and support innovation in this area.

Work with project managers who can introduce business cases, prototyping processes, and more.

Sustainable funding is the biggest factor in the delivery and maintenance of projects (non-profit or commercial). Projects often fail to identify their target audience and develop sustainable funding models. This need has been highlighted in previous research,[27] and is especially pertinent for attracting talent to decentralized application development.

27. Eghbal, Nadia. Roads and bridges: The unseen labor behind our digital infrastructure. Ford Foundation, 2016. https://www.fordfoundation.org/media/2976/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure.pdf

Initial funding. To support growth and widespread adoption, long-term project sustainability should be considered a key piece of project strategy. We uncovered a variety of options for initial funding within open source technology development, from market-driven venture capital to volunteer-run or grant-funded. These offer a start for new research projects, but are not sustainable for products, especially for teams with business overheads. Initial funding allows for the creation of a prototype, but widespread user adoption requires significant time investment which outlives the initial funding period.

Cryptocurrencies. Projects should only adopt cryptocurrencies when their use emerges as an essential component based upon research, design, and market analysis. There are mixed responses towards the application of cryptocurrencies for sustainability. Crypto-economic thinking could provide a framework for incentive structures that ensure sustainability for protocols. For example, market mechanisms such as fees for transit or storage paid to protocol creators are undergoing active experimentation within the blockchain ecosystem.[28] Some projects are wary of "scammy token models" and the creation of false scarcity for founder profit. Others pointed out that crypto-economic mechanisms have a clear use case for some purposes, such as Sybil resistance.

> "[Decentralization] enables a set of use cases that can avoid cryptocurrency and there are some risks (such as lack of clear regulation). The two need not be conflated or inextricably linked.

Business models. To support strategy and growth, project managers should be involved to develop a licensing strategy, business case, and clear migration processes from centralized competitors. Very few projects actively engage with usability and adoption, despite their desire to expand use cases and network volume. Non-profit projects often care about end users in the abstract, but don't have the appropriate team to prioritize this work. Consultancies offering technical support have also proven to be an effective strategy for sustainability; however, these models still require expertise outside of engineering and design. Licenses should also be chosen based upon project sustainability concerns, rather than simply defaulting to any common open source license.

28. https://web.archive.org/web/20200311162809/https://electriccoin.co/blog/funding/

7 Governance

Decision-making and stability.

Include diverse voices on the core team and within the community to make sure that technology works for more than those who develop it.

Innovate on community-centered funding strategies to improve the health of the open source ecosystem.

Craft governance structures for decision-making about protocol upgrades and other ecosystem developments before widespread deployment.

Decentralized protocols require special attention to governance. No single party has control over when and how a decentralized protocol is upgraded across many individualized machines. Protocol updates can cause significant user and developer experience challenges regarding version compatibility, especially when considering limited Internet connectivity or custom protocol extensions. An open governance structure with strong social rules for decision-making is recommended for decentralized protocols.

Structures and cultures. Projects in this space often describe themselves as socio-political reactions to harmful structures created by technology, and those values can be reflected in governance models as well. While technical approaches and issues could be similar across projects, the governance structures and cultures varied with significant similarities to the wider open source ecosystem.[29] When different structures and cultures interact with each other, ideologies, expectations, and interests can come into conflict. Projects should make these aspects explicit through a defined vision and clear governance model as a first step to forming partnerships and coalitions.[30]

Decision-making. Maintainers should define the decision-making process in early stages of protocol development and make that process clear to new contributors and end-users. It is important to establish rules of engagement for how decentralized protocols are changed over time. For many projects, this looks like a dedicated core team that does the heavy lifting and are self-selected by the organizational structure.

Diverse opinions and backgrounds. Project leads should focus on diversifying the decision-making team and open source community to get wider input and adoption. Developers operate in a relatively homogenous environment. Homogeneity both refers to demographics (in line with larger patterns in the technology industry: often people identifying as male and white, working from North America and Europe, with a background in distributed systems[31] [32] [33]), as well as their shared beliefs and attitudes (in resilience, autonomy, and security). This has resulted in technologies that meet the needs of a small portion of the world population, resulting in a general for us, by us mentality.

> “Certain users in the platforms are elevated over others based on decisions and algorithms implemented by the platform. [Ask] where is trust being placed: whether it is in the coders, the developers, those who design and govern mobile devices or apps; and whether trust is in fact being shifted from social institutions to private actors.[34]

29. See also https://opentechstrategies.com/files/pub/MZ+OTS_OS_Archetypes_report_ext_scr.pdf
30. https://communityrule.info/
31. Eghbal, Nadia. *Roads and Bridges*
32.https://www2.deloitte.com/content/dam/Deloitte/se/Documents/risk/DI_2019-global-blockchain-survey.pdf
33. https://www.researchgate.net/publication/326079170_Diversity_in_software_engineering
34. Article 19. (2019) Blockchain and Freedom of Expression. https://www.article19.org/wp-content/uploads/2019/07/Blockchain-and-FOE-v4.pdf

Blockchains. Governance models that include a blockchain as a primary component should also include social and legal considerations outside of the blockchain itself. Blockchains are under active experimentation as a decentralized system for encoding governance structures. The consensus-based system is designed for a scenario where public transactions are mediated between participants which are all potentially malicious and cannot be trusted. Technology is not neutral,[35] which in practice means that even with a blockchain, users need to trust someone at some point—e.g., rules set by the developers, designers, and investors. These roles, again, might represent existing power imbalances.

35. https://www.fordfoundation.org/ideas/equals-change-blog/posts/technology-is-not-neutral-it-s-political/

Conclusion

There is a significant gap between the protocols that define the decentralization space and the applications that users want to adopt. Decentralized applications have the potential to achieve widespread adoption, but there are still key challenges across the ecosystem that have yet to be addressed. Perhaps the most difficult of them all is the general insecurity of our shared digital infrastructure, which affects all software and hardware. By collaborating across projects, the ecosystem will more effectively gain support to evolve and challenge this status quo. Because decentralization is a newly formed and rapidly evolving approach to end-user applications, designers and developers need new educational materials, trust models, and design patterns to help them succeed. Funders need to innovate on community-driven initiatives for sustaining projects as well as attracting and training this talent. These recommendations were crafted with the goal to increase the overall quantity and quality of decentralized applications.

Appendix

Research Participants

- Alexander Cobleigh, Cabal
- Antoine Toulme, Whiteblock
- Antonela Debiasi, Tor Project
- Benedict Lau, Hypha Worker Co-operative
- Chris Waclawek and Rich Burdon, DXOS.org
- Dan Hassan and Kieran Gibb, Magma Collective
- Darius Kazemi, runyourown.social
- Dean Eigermann and Dmitriy Ryajov, Status
- Eletherios Diakomichalis, Alexis Sellier, Abbey Titcomb, Radicle
- Emaline Friedman, Commons Engine
- Fatemeh Shirazi, Web3 Foundation
- Feross Aboukhadijeh, WebTorrent
- Franz Heinzmann, Sonar
- Harry Halpin, Nym
- Harry Lachenmayer, independent
- Holger Krekel, Delta.Chat
- Ira Bolychevsky, Redecentralize
- Iryna Nezhynska, Jolocom / DWeb. Design
- Jay Graber, independent
- Kenneth Ng, Ethereum Foundation
- Kira Oakley, Digital Democracy
- Lars Eggert, IRTF
- Mai Sutton, disaster.radio
- Mark Nadal, gun.eco
- Mathias Buus, Dat / Hyperdivision
- Matthew Slipper, Kyokan
- Matthew Wild, XMPP Standards Foundation / Modern XMPP
- Michael Rogers, Briar
- Michel Bauwens, P2P Foundation
- Nadir Chishtie, Matrix.org / Riot
- Niels ten Oever, University of Amsterdam and Texas A&M University
- Paolo, Optool / DTN Conference
- Paul Frazee, Beaker Browser
- Paul Gardner-Stephen, Serval / MEGA65
- Peter Czaban, Web3 Foundation
- Peter van Hardenberg, Ink & Switch
- Pospi, Holochain
- Przemyslaw Rekucki, Golem
- pukkamustard, openEngiadina
- Roel Roscam Abbing, Malmö University
- Sadie Doreen, I2P
- Scott Moore, Gitcoin
- substack, bits.coop
- Tiberius Brastaviceanu, Sensorica
- Wendy Hanamura, Internet Archive
- Zak Cole, Whiteblock
- Zelf and Mix, Secure Scuttlebutt