On the Utility of Azimuth and Decentralized Identity

Stuart Christoph ~sarlev-sarsen
~tocwex.syndicate

Abstract

This article considers the concept of identity within the context of Urbit and its identity layer, Azimuth. Comparison is made to self-sovereign identity (SSI) and decentralized identity (DID) design principles. We consider how these systems implement decentralized identity, examining their technical foundations, practical applications, and the broader implications for digital self-sovereignty. Understanding these elements sheds light on the future of identity in a decentralized world.

Contents

Urbit Systems Technical Journal I:2 (2024): 9–33.

Address author correspondence to ~sarlev-sarsen.

1 Introduction

The nature of identity has captivated human attention since prehistory. Cave painting self-portraits and handprints reveal an early sense of 'self' in even the earliest hominids. More recently, Descartes' *cogito, ergo sum* laid the groundwork for modern philosophy. As we move into the digital age, the concept of 'identity' continues to evolve in order to extend into online networks.

This article considers the concept of identity within the context of Urbit and its identity layer, Azimuth. We explore how these systems implement decentralized identity, examining their technical foundations, practical applications, and the broader implications for digital self-sovereignty. By understanding these elements, we aim to shed light on the future of identity in a decentralized world.

1.1 Decentralized Identity

Urbit isn't the only project working on the concept of decentralized identity. As a starting point, we will examine two perspectives on decentralized identity:

1. Self-Sovereign Identity
2. Decentralized Identifiers

Blockchain developer Christopher Allen coined the term 'self-sovereign identity' (SSI) to concretize a notion of digital decentralized identity built for individual control and self-determination (Allen, 2016). He noted that the evolution of online identity has passed through eras of centralized administrative control, oligarchic federated systems, and unfortunately, institutionally captured "user-centric" designs. The next era, designated the era of SSI, thankfully aims to return us to a more intuitive understanding of digital identity. Allen defined ten key principles for SSI:[1]

- Existence
- Control
- Access
- Transparency
- Persistence
- Portability
- Interoperability
- Consent
- Minimalization
- Protection

However, we contend that each of the pre-SSI eras has critically pulled us away from the sense of 'self' that is vital to identity–from the way we experience our identity in the world of atoms and the digital experiences which are becoming ingrained in our day-to-day lives.

Separate from Allen's principles, W3C has presented a standard for Decentralized Identifiers (DIDs). According to "Decentralized Identifiers (DIDs) v1.0", in part (World Wide Web Consortium, 2022):

> Decentralized identifiers (DIDs) are a new type of identifier that enables verifiable, decentralized digital identity. A DID refers to any subject (e.g., a person, organization, thing, data model, abstract

[1] The appendix contains an excerpt defining the principles more completely.

> entity, etc.) as determined by the controller of the DID. …
>
> While other parties might be used to help enable the discovery of information related to a DID, the design enables the controller of a DID to prove control over it without requiring permission from any other party. DIDs are URIs that associate a DID subject with a DID document allowing trustable interactions associated with that subject.

While the Urbit project may not traditionally rely on 'Earth standards,' the DID recommendation also defines ten design goals that are informative for understanding decentralized identity.[2]

- Decentralization
- Control
- Privacy
- Security
- Proof-based
- Discoverability
- Interoperability
- Portability
- Simplicity
- Extensibility

We will revisit some of these principles and design goals in the context of Azimuth and Urbit, but first, a reminder to developers: defaults matter.

1.2 Defaults Matter

As `~wicdev-wisryt` noted of the public key infrastructure (PKI) idea maze (see `~wicdev-wisryt` (2024), pp. 1–7 in this issue), the long history of PKIs largely splits between being 'decentralized and unused' or 'centralized and easy to subject oneself to'. `~wicdev-wisryt` explored this idea maze along the branches of global consistency, permanence, and self-ownership, to which we add a fourth branch: sane defaults.

PGP, GPG, and Web of Trust solutions have existed for decades but are used by effectively nobody, and in some sense can be dismissed due to a lack of sane defaults. One could argue the shortcoming of these systems is a lack of global consistency, but in practice global consistency is a necessary but

[2] See the appendix for a more complete excerpt explicating the design goals

not sufficient condition for a self-sovereign identity. If the system does not also provide sane defaults the global consistency remains an illegible mess to the typical user. The Web of Trust and similar solutions may be glossed over as demonstrably insufficient thanks to this shortcoming.

If we consider global consistency, permanence, and self-ownership as useful goals for the PKI, we must recognize that they fail to explain some of the more controversial elements of Azimuth's hierarchical model for a certain subset of critics. In fact, alternate conceptions for PKIs and identity more broadly are being explored by both Kinode[3] and Pallas.[4] These projects aim to satisfy each of the three branches but take different approaches to the concept of sane defaults. So why does Urbit's Azimuth retain a hierarchical address space?

Aside from being a clean way of naming a finite address space, Azimuth's hierarchy provides a *sane default* for packet routing, peer discovery, and third-party service provision. It gives users a starting point, and a set of alternatives that are tractable. It serves as the crisis moment ('It's dangerous to go alone! Take this') before embarking on the journey of self-sovereign identity.

Some critics frame this structure as 'digital feudalism' but fail to understand two other elements of the system:

1. The default PKI includes a permissionless right to exit, requiring no approval from the network node up the hierarchy from you.

2. Even the hierarchical model is *just a sane default* and does not forbid an alternative.

While no alternative exists at present,[5] there is no reason that a particular Urbit ship must exclusively ask its sponsor for the

[3] As of writing, Kinode's `.os` identity implementation leans towards allowing users to bring their own identity system, and handling packet routing by designated and self-declared routing nodes.

[4] As of writing, Pallas, formerly Plunder, has not implemented an identity system. In the author's conversations with developers working on the project, Pallas aims to be unopinionated about either identity system or packet routing and peer discovery, instead deferring that role to projects building on top of the system.

[5] As a last minute addition, we note the "Groundwire" proposal (by

addresses of its peers. Nor is there any technical restriction on Urbit's routing packets using a different model. In summary, *the hierarchy is a sane default in an otherwise infinite game.*

2 The Current State of Azimuth

The need to place the PKI on a distributed ledger with global consensus and recognize the hierarchical model as a sane default is obvious. But what about the current address space and its technical implementation? Since early 2019, Azimuth has been on hosted on the Ethereum blockchain, and in 2022 the contracts were updated to support a "naive" Layer 2 rollup. Returning to the principles and design goals of SSI and DIDs, this on-chain presence provides benefits such as existence, control, transparency, persistence, decentralization, and discoverability. It also supports many of the other stated goals, if not satisfying them purely through an on-chain presence. Many of these principles were either explicitly or implicitly considered with both the initial Azimuth contracts (`~ravmel-ropdyl`, 2019), and the move to supporting the Layer 2 rollup (`~datnut-pollen`, 2021). However, the current state of Azimuth falls short in terms of providing interoperability and extensibility.

2.1 Layer 1 IDs on the Ethereum network

Azimuth's deployment on Ethereum was one of the first uses of the ERC-721 standard (Entriken et al., 2018). It is implemented across two core contracts: Azimuth, which holds the registry of identity ownership, and Ecliptic, which defines the business logic for updating the state of the registry. To this day, these contracts serve as one of the most technically 'useful' ERC-721s on the Ethereum network, powering the peer-to-peer interactions of thousands of nodes (also known as urbit ships) in an

`~hastuc-dibtux`, `~tondes-sitrym`, et al.), which was published just before this issue went to print. Groundwire suggested a project for a Gall agent enabling usage of Bitcoin and the comet level address space for an alternative PKI without hard-forking Azimuth.

encrypted off-chain network. As an ERC-721, Urbit ID experienced a significant increase in valuation during the NFT boom of 2021, with the market cap of address space breaking $2B at its peak.

However, the core downside of Azimuth's presence on Ethereum should not be understated. In exchange for the benefits of NFT hype cycles, Urbit users must pay a tax to the Ethereum network in the form of gas fees. This interaction can be particularly expensive because the original versions of Ecliptic required all state transition computations to occur on-chain. While this means Azimuth inherits the security benefits of Ethereum's global consensus model, there is a contention that it fundamentally extracts value from the Urbit network and deposits it into Ethereum in an unnecessary exchange.

2.2 Layer 2 IDs on the naive rollup

The transaction costs on Ethereum were so great that, at times, it cost over $200 worth of ETH just to get on the network. For a budding distributed system project, such costs can be unbearable. Thus, the solution to the 'gas crisis' was introduced (`~datnut-pollen`, 2021): a custom 'naive' rollup. The key feature of the naive rollup is that, instead of paying Ethereum miners (now validators) to verify the state transitions of Azimuth, each Urbit node can validate those transitions themselves. This again brings us back to the idea of sane defaults in a decentralized or distributed network.

That is, in running one's Urbit instance, the canonical "Azimuth" is nothing more than a sane default. This is trivially exemplified by the fact that one could run a fake ship network locally and act as any Urbit ID one chose, or even deploy an alternative version of Azimuth onto mainnet Ethereum. However, because everyone else is likely operating on the default version of `/app/azimuth`, which assumes `azimuth.eth` as The One True PKI, any of one's potential peers will discard one's packets as invalid. The Urbit network will ignore an unsignable interloper. Whether or not the Urbit network pays the Ethereum network for state transition computation, it remains true that Urbit's off-chain interactions are voluntary and governed by

sane defaults.

2.3 When to Use Global Consensus

As noted in ~wicdev-wisryt's piece, blockchains allow users to distinguish two cryptographically valid messages, specifically which was signed first, without requiring a centralized party. Both Layer 1 and Layer 2 implementations rely on this affordance. Effectively, they both need the chain to give them a timestamp and a signed message to discern key ownership. Whether the cosmputation of key rotation is done on-chain or client side, anyone within the Urbit network that is continuing to use the system defaults will ultimately get the correct result for validating a packet's sender. It is the core affordance of timestamped messages for which Urbit must pay *some* blockchain to have a viable decentralized PKI.

However, Urbit-side key validation and peer discovery are not the only benefits of having the PKI on Ethereum. The cost of on-chain computation for global consensus is undoubtedly a hurdle to low-cost user adoption of Urbit. However, it would be a failure of imagination to claim there is no contingent loss of benefits. Specifically, Ethereum is a growing network that is building on top of the same ERC-721 standard on which Urbit ID is defined. Improvements to the capabilities of the Ethereum network can enhance the capabilities of the Urbit network—*if Urbit maintains, or increases, its legibility to the Ethereum Network*. The current naive rollup does not achieve this. An L2 Urbit ID, and it's ownership in particular, is functionally illegible to the outside world. From the perspective of Ethereum, global consistency is compromised, and key rotation is impossible to validate.

While we could explore the technical details of these two systems, instead let's return to examine the SSI principles and DID design goals of interoperability and extensibility. In a sense, these just mean that you want your identity to work in as many places as possible. To be as legible to as many systems or stakeholders as is possible.

A playful example of this is whether one would trade one's state-issued driver's license for a high school ID card (~sarlev-

sarsen, 2024). In spite of one's driver's license being more costly to replace or update, one's response is almost certainly, "No way." A licensee knows intuitively that a driver's license is a more credible and legible identifier than a high school ID, in large part due to the authority of the centralized issuing authority. Imagine achieving such a level of credibility and legibility in a self-sovereign manner. This tradeoff is not merely a thought experiment, as dependency on centralized identity issuance authorities has real world impacts:

> In the last year, self-sovereign identity has also entered the sphere of *international policy*. This has largely been driven by the refugee crisis that has beset Europe, which has resulted in many people lacking a recognized identity due to their flight from the state that issued their credentials. However, it's a long-standing international problem, as foreign workers have often been abused by the countries they work in due to the lack of state-issued credentials. (Allen, 2016)

One may reasonably ask at this junction whether, if Urbit IDs will form the foundation for self-sovereign digital identity, they should be more legible or less legible?

While this may run counter to some Urbit purists who want to avoid anything related to 'Earth code,' the reality for the foreseeable future is that interoperability with other networks is both desirable and necessary for Urbit's continued survival. With time, as Urbit evolves, it may become the most legible system, and the cruft of Earth code will fall by the wayside. But that future is still a long way off.

Improving legibility, or in other words, protecting the interoperability and extensibility of Urbit ID, does not imply loyalty to any particular chain. It simply claims that increasing legibility to the largest set of networks should be included in the objective function of Urbit ID.

3 Alternative Options for the PKI

How does Urbit continue to improve the utility of Azimuth? While in 2019 the options for hosting the PKI were still limited, the technology landscape has drastically evolved in the intervening years. Several Urbit-related blockchain projects have been proposed and explored recently which bear on how the PKI could be hosted in the future.

3.1 Self-Host the PKI on `%chain`

The most 'philosophically straightforward' answer would be for Urbit to self-host its PKI. While this requires further technical development, it is conceptually tractable and the `%chain` project, primarily executed by `~tiller-tolbus` and `~midsum-salrux` as a skunkworks effort, has shown initial successes on implementing an Urbit-native blockchain. This option could make use of identity-backed consensus models, and would be the most direct pathway to the Urbit network not having to pay a 'tax' to another network for its consensus mechanism. In a sense, this is the most 'independent' or 'self-contained' option. However, it would likely result in decreased capital access and practical legibility to external networks.

While reasonable minds may differ on the net benefit of Urbit's proximity to the Ethereum network's capital pools, there is no doubt that such proximity has materially impacted on address space valuation. This includes the ~$33,000 peak for star-level identities. Distancing Urbit ID from Ethereum by moving the PKI to a self-hosted network would limit the opportunity to access this capital by increasing the barriers to exchange.

Moving the PKI in this way would also decrease interoperability and extensibility by making it harder for the Ethereum network to access and validate data about ownership of Urbit IDs. Technically speaking, this constraint could be reduced by self-hosted RPC endpoints of `%chain` or Azimuth state that could be accessed over HTTP. However, the more prominent issue would be the coordination constraint it would generate. If the PKI moves, what motivation do members of the Ethereum

Network have to bother with Urbit RPC endpoints instead of just staying within the Ethereum tooling ecosystem?

In a world in which Urbit is more powerful and offers a compelling, market-legible alternative to the Ethereum virtual machine, one can see the draw of a bootstrapped `%chain`. However, the ~$70,000 per year in gas fees that Urbit users collectively pay to maintain the PKI on Ethereum seems a worthwhile exchange for the legibility it affords.

3.2 Zenith: UrbitChain on Cosmos SDK

Moving the PKI to a self-hosted, urbit-native solution is not the only option should the Urbit network want to reduce its dependency on outside networks. One standing proposal, Zenith, establishes an off-Urbit blockchain governed by the galaxy-level address space and implemented as its own L2 Ethereum rollup using the Cosmos SDK.

In terms of legibility, this pathway essentially splits the difference. The Cosmos ecosystem is smaller than Ethereum as a whole, but comes in second place on metrics for current Total Value Locked (TVL) (Fernau, 2022). The Cosmos ecosystem and SDK could give some of the benefits of a self-hosted system. For example, it would reduce the outflow of capital from the Urbit network, while still achieving some interoperability with the external world thanks to the affordances granted by Cosmos. Peg zones and the Inter-Blockchain Communication (IBC) protocol would enable ready access to a network of blockchain networks. These networks are likely to have more operational friction for capital flows than remaining on L1 Ethereum, but may be a valid tradeoff. Additionally, the Zenith proposal makes a nod to the opportunity to use the Zenith rollup as a general purpose urbit-aware smart contracting platform, including a "Scry Oracle Contract" which would enable canonization of certain subsets of the scry namespace and create some on-chain enforcement mechanisms of the referentially transparent nature of the urbit namespace.

3.3 Mayflower: A Recolonization Effort

Also presented as an option for the future of the Azimuth PKI is a proposal under the codename "Mayflower". While there is currently a directional debate between the Zenith and Mayflower plans, due to disagreements on implementation and design of related non-fungible tokens and governing bodies, the Mayflower proposal similarly suggests use of an Ethereum L2 rollup and visions of future urbit native gas tokens and contracts.[6] The two main distinctions at this time appear to be a preference to avoid 'rolling your own' rollup and leaning towards something like the existing Base rollup and Optimism Superchain, plus a focus on 'rezoning' urbit address space, such that more focused development will occur within the readily available 'digital land'. Proponents of this plan have noted that it neither precludes an Urbit-native chain, nor path to an Urbit specific rollup. As of writing, specific details on this plan are outstanding and in active debate. Notably, both these proposals aim towards using existing Ethereum layer 2 technology, but depending on implementation details the portability of identities between L1 and L2 is undetermined.

3.4 Bridge the PKI across multiple chains

As we consider the tradeoffs of moving the PKI, it is worth revisiting our 'sane defaults' branch in the idea maze. If `azimuth.eth` and with it `/app/azimuth.hoon`[7] are just sane defaults, why not also support some sane *alternatives*? One of the final Urbit precepts reminds us that "communities are autonomous" (`~wicdev-wisryt`, 2020).

To this end, we should use Urbit OS as the coordination technology to allow users and communities to decide with which other networks they desire to maintain legibility. As with the current PKI's split between L1 Ethereum and the naive rollup, so too could we allow people to move their Urbit ID

[6] The author of this piece makes no statement about the financial viability of these two options and notes that they both appear to recognize the illegibility of the way the PKI is currently fractured and aim at rectifying it.

[7] The Jael secrets vane of Urbit OS merely subscribes for the PKI state, rather than storing it in Arvo itself.

from L1 Ethereum to any other chain with a viable way to post signature data.

Some may choose to remain on mainnet Ethereum to take advantage of features like ERC-6551 Token Bound Account (Windle et al., 2023).[8] Others may want to inscribe their ownership as an Ordinal on Bitcoin. Alternatively, Solana offers some benefits for the more crypto-degenerate who are looking for speed of trading and market volatility to help make their wealth.

Considering this path, it is worthwhile to elaborate on how the current L2 solution works. The original version of the Ethereum contracts held ownership data in the Azimuth contracts and business logic in the Ecliptic contracts. In 2021, these contracts were updated to support two new affordances:

- Azimuth points sent to the Ethereum deposit address (`0x1111...`) are assessed by the default Urbit client as being owned on layer 2.
- Ecliptic posts *time-stamped* signature data to the Ethereum chain concerning the change of ownership of a point owned on layer 2. The default Urbit client reviews and validates the state change before applying the change to the Urbit's own `/app/azimuth`.

This mechanism means that these points are still cryptographically controlled for purposes of anyone running the default Urbit OS or relevant code (namely, `/app/eth-watcher`, `/lib/naive.hoon`, `/app/azimuth-tracker` and `/app/azimuth`), and it is easy to imagine how this same mechanism could be used to support other chains as hosts for portions of the PKI:

1. Define a holding address for the ERC-721 on mainnet ethereum for other host chains (i.e. sending the L1 NFT to `0x2222...` indicates sending the ownership to Bitcoin).
2. Implement a watcher for the subject chain that looks for posted signature data.

[8] ERC-6551 implements an interface and registry for smart contract accounts deterministically controlled by one's ERC-721 Urbit ID.

3. Update and/or add to the relevant Urbit OS code for client side validation of data posted to the new chain.

This approach would increase legibility to other chains, giving communities more options for what their members can achieve with their Urbit ID.

However, as the PKI fractures, maintaining global consistency becomes increasingly complex. In the above model, the primary solution for maintaining this consistency becomes that official 'global consensus' is on mainnet Ethereum, while any other alternative is subject only to the client side validation. Depending on the affordances of the subject chain, the level of effort put into building the subsystem, and any cost concerns for network fees on the specific subject chain, the result is likely that *moving to any other chain is a one-way trip*. Just as a move to the current naive rollup is a one way trip.

The biggest reason for this being the case is the question: 'what timestamp is the canonical one?'.

3.5 Move the PKI to Nockchain

While Ethereum has made significant progress in interchain bridges and L2 rollups since the implementation of the Naive rollup, making two-way bridges increasingly possible, they still introduce a host of security and consensus questions. They also are likely to exacerbate the capital outflow concerns of the current L1 Ethereum implementation.

But if we return to the philosophical perspective, assuming we want our identities to be legible to different networks, don't we also want to potentially modify that legibility? Just as individuals can emigrate between nation-states, it seems self-evident that we should be able to move between digital networks, bringing our self-sovereign identities with us.

What is the best way to enable this portability (another shared principle/design goal for SSIs/DIDs) while maintaining global consistency? The reader who is familiar with Nockchain will anticipate the punchline: zero-knowledge proofs.

Using Zorp's EDEN and its associated Nock prover (`~tacryp-socryt` et al., 2023), any Urbit instance can prove

valid computation of any 'client-side' Azimuth state change and use that proof as the data to post to any given blockchain. It is likely that the 'easiest' place to do that would be Nockchain. However, external legibility may incentivize these proofs to be posted elsewhere. An Azimuth contract on a subject chain could use the proof of valid Nock computation to assign ownership of an Urbit ID crypto-asset native to a given address. This could include whatever affordances native asset ownership on that chain may bestow to an Urbit operator.

In the long run, this pathway would likely lead more Urbit operators to move their Urbit ID's primary 'citizenship' to Nockchain. This would occur without unnecessarily losing legibility to other networks, and retains the possibility of absorbing external capital into the Urbit network along the way.

4 Other Considerations for Azimuth

While Azimuth's on-chain nature is that of a public key infrastructure, a PKI alone does not constitute a complete identity system. In fact, it is the pairing of Urbit ID with Urbit OS that presents the best opportunity to serve as a fully-fledged decentralized identity system. Whether we look at implementing W3C's standardized DIDs, or adhering to the principles of SSIs, not everything about Azimuth is, or should be, on-chain.

4.1 GroundWire, comets, and Bitcoin Ordinals

An early design consideration of the 21e8 Bitcoin Ordinals project (Ordinals, 2024) was reportedly to be able to host an Azimuth-like PKI system, and the desire to tie Urbit's fortunes more closely to those of Bitcoin persists in segments of the Urbit community. A recent debate throughout the ecosystem about the Azimuth PKI, namely between Zenith and Mayflower, has instigated a wide-ranging discussion about alternative ways of handling identiy across the Urbit network. GroundWire proposes to experiment with both new PKI governance models, and to make use of the comet-level address space in a more robust way (`~hastuc-dibtux` et al., 2024). Im-

plementation details are still forthcoming, but the initial plan is to prototype a Gall agent to support a PKI on Bitcoin that would allow comets to rotate their keys (updating the current 'self-attestation' model), and make use of different non-hierarchical models for routing and peer-discovery. Additionally, instead of using the finite 'higher level' address space for spam resistance, this proposal would use mechanisms like L1 Bitcoin fee costs, or burning and timelocking coins.

As with the Zenith and Mayflower proposals, the GroundWire proposal makes statements about the governance and fundrasing directions of the Urbit project. These topics are out of scope for the focus of this article, but merit ongoing discussion in the appropriate forums.

4.2 Illegible subnets

Globally available and transparent chain state greatly enhances legibility. Openly available sponsorship hierarchies inform unconnected peers about how to reach you, and global claims can be used for various purposes. However, not every application requires global consensus, and Turing-complete smart contracts can lead to the risk of open-ended cost overruns. Furthermore, many networks may wish to become or remain illegible to the outside world. Private access control lists, gossip networks, and packet routing mechanisms enable use cases that would not benefit from the cost or transparency of on-chain or globally consistent legibility.[9] Urbit's off-chain and encrypted peer-to-peer messaging affordances open up an intriguing landscape of 'illegible subnets' while still benefiting from associations with specific pseudonymous Urbit identities.

The value of illegible subnets also builds on the insight that not everything can, or should, be distilled down to its lowest common denominator, tokenized, and put on the blockchain. The more complex the interaction, the more costly the distillation process, and *the more likely there is no consensus to*

[9]It is worth noting that illegible subnets may contend with the SSI principle of consent, in that they enable opaque data-sharing networks and other similar 'darknet' use cases.

be found around its 'valuation'. Instead of demanding timestamped global consensus in all our peer interactions, we must support emergent, implied-consensus networks. These networks are governed not by immutable chain state but by willing association and participation by sovereign peers. Systems where the network forms and dissolves on the fly, based on the ongoing discovery and refinement of the values of its participants.

4.3 Peer-to-peer attestations

Another affordance that does not require a blockchain is peer-to-peer attestation. There are many use cases where the privacy and cost benefits of cryptographically verifiable peer-to-peer attestations are desirable. They can be used independently, e.g. "`~sarlev-sarsen` attests to having received a message from `~lagrev-nocfep` containing a valid passcode", or even in conjunction with global blockchain state.

For example, `%fund` by `~tocwex.syndicate`,[10] uses peer-to-peer attestation begin to build a subjective reputation graph. An Urbit can hold a signed message from an Ethereum wallet attesting contol, e.g. "`~sarlev-sarsen` controls wallet `0x6789...`", and can even share this signed message with other peers.

```
::  $prof: user profile data
::
++  prof
  $:  ship-url=@t
      wallets=(map addr sigm)
      favorites=(set flag)
  ==
::
```

Anyone receiving that attestation can then use it in constructing an understanding of the relationships between different Urbit IDs, off-chain financial promises, on-chain transactions, reputation, or other economic activity, without it thereby becoming a canonical, globally available understanding.

[10] `https://tocwexsyndicate.com`

Such shared understandings granted by peer-to-peer attestations can thus create pockets of privileged information networks, and the attendant competitive advantages of a more opaque, yet verifiable, system.

4.4 Self-verified credentials

As ~wicdev-wisryt notes, one offshoot of the permanence branch of the PKI idea maze makes available a mechanism for self-attestation. This currently only exists for the portion of Azimuth's address space where comet identities reside. This self-attestation is useful for a variety of purposes—cost, anonymity, etc.—but doesn't afford a more long-lasting system for self-verified credentials due to the transient nature of a comet identity. It is likely that expanding the capacity for self-attestations would be of benefit to the Urbit network. In particular, the concept of Verifiable Credentials (VC) may be worth pursuing (World Wide Web Consortium, 2022).

Urbit ID does not currently have a system of verifiable credentials, and arguably, the PKI aspect of Azimuth should never include this. However, it should be reasonably straightforward to produce either a VC attestation layer utilizing Azimuth or incorporate knowledge of Azimuth into a VC solution.

One mechanism that could be integrated into an Urbit-native VC solution is the Urbit Hierarchical Deterministic (HD) Wallet. This is the technology used to secure ownership of an Urbit ID point and could satisfy the SSI doctrinal requirement that, "an SSI digital wallet should implement open standards for portable, self-sovereign verifiable credentials and other sensitive private data" (Preukschat and Reed, 2021). Without VC, Urbit does not yet qualify for SSI under this definition.

5 Conclusion

Much work remains to be done on both the on-chain implementations and off-chain affordances of Azimuth and the Urbit ID system. Each area of work must be considered alongside the effort required for implementation:

- Balancing optionality in the PKI for end users against the ongoing maintenance of a more complex codebase.
- Weighing legibility for external networks against measures that reduce capital outflows from the network.
- Allocating development time to adherence with DID, SSI, and VC standards against the practical usefulness of their implementation.
- Coordinating systems for illegible subnets and peer-to-peer attestations against other features that can be productized and marketed to consumers.

Regardless of the specific focus, Urbit ID and the Azimuth address space offer developers a platform to build systems that enhance individual sovereignty and provide flexibility in creating community networks that control their own legibility and interactions.

6 Appendix

This appendix serves as a reference for the more complete excerpts of referenced articles.

6.1 The Path to Self-Sovereign Identity

The principles of SSI are as follows ("Ten Principles of Self-Sovereign Identity," 26 April 2016), all quoted:

> A number of different people have written about the principles of identity. Kim Cameron wrote one of the earliest "Laws of Identity", while the aforementioned Respect Network policy and W3C Verifiable Claims Task Force FAQ offer additional perspectives on digital identity. This section draws on all of these ideas to create a group of principles specific to self-sovereign identity. As with the definition itself, consider these principles a departure

point to provoke a discussion about what's truly important.

These principles attempt to ensure the user control that's at the heart of self-sovereign identity. However, they also recognize that identity can be a double-edged sword — usable for both beneficial and maleficent purposes. Thus, an identity system must balance transparency, fairness, and support of the commons with protection for the individual.

1. **Existence**. *Users must have an independent existence.* Any self-sovereign identity is ultimately based on the ineffable "I" that's at the heart of identity. It can never exist wholly in digital form. This must be the kernel of self that is upheld and supported. A self-sovereign identity simply makes public and accessible some limited aspects of the "I" that already exists.

2. **Control**. *Users must control their identities.* Subject to well-understood and secure algorithms that ensure the continued validity of an identity and its claims, the user is the ultimate authority on their identity. They should always be able to refer to it, update it, or even hide it. They must be able to choose celebrity or privacy as they prefer. This doesn't mean that a user controls all of the claims on their identity: other users may make claims about a user, but they should not be central to the identity itself.

3. **Access**. *Users must have access to their own data.* A user must always be able to easily retrieve all the claims and other data within his identity. There must be no hidden data and no gatekeepers. This does not mean that a user can necessarily modify all the claims associated with his identity, but it does mean they should be aware of them. It also does not mean that users have equal access to others' data, only to their own.

4. **Transparency**. *Systems and algorithms must be transparent.* The systems used to administer and operate a

network of identities must be open, both in how they function and in how they are managed and updated. The algorithms should be free, open-source, well-known, and as independent as possible of any particular architecture; anyone should be able to examine how they work.

5. **Persistence**. *Identities must be long-lived.* Preferably, identities should last forever, or at least for as long as the user wishes. Though private keys might need to be rotated and data might need to be changed, the identity remains. In the fast-moving world of the Internet, this goal may not be entirely reasonable, so at the least identities should last until they've been outdated by newer identity systems. This must not contradict a "right to be forgotten"; a user should be able to dispose of an identity if he wishes and claims should be modified or removed as appropriate over time. To do this requires a firm separation between an identity and its claims: they can't be tied forever.

6. **Portability**. *Information and services about identity must be transportable.* Identities must not be held by a singular third-party entity, even if it's a trusted entity that is expected to work in the best interest of the user. The problem is that entities can disappear — and on the Internet, most eventually do. Regimes may change, users may move to different jurisdictions. Transportable identities ensure that the user remains in control of his identity no matter what, and can also improve an identity's persistence over time.

7. **Interoperability**. *Identities should be as widely usable as possible.* Identities are of little value if they only work in limited niches. The goal of a 21st-century digital identity system is to make identity information widely available, crossing international boundaries to create global identities, without losing user control. Thanks to persistence and autonomy these widely available identities can then become continually available.

8. **Consent**. *Users must agree to the use of their identity.* Any identity system is built around sharing that identity and its claims, and an interoperable system increases the amount of sharing that occurs. However, sharing of data must only occur with the consent of the user. Though other users such as an employer, a credit bureau, or a friend might present claims, the user must still offer consent for them to become valid. Note that this consent might not be interactive, but it must still be deliberate and well-understood.

9. **Minimalization**. *Disclosure of claims must be minimized.* When data is disclosed, that disclosure should involve the minimum amount of data necessary to accomplish the task at hand. For example, if only a minimum age is called for, then the exact age should not be disclosed, and if only an age is requested, then the more precise date of birth should not be disclosed. This principle can be supported with selective disclosure, range proofs, and other zero-knowledge techniques, but noncorrelatibility is still a very hard (perhaps impossible) task; the best we can do is to use minimalization to support privacy as best as possible.

10. **Protection**. *The rights of users must be protected.* When there is a conflict between the needs of the identity network and the rights of individual users, then the network should err on the side of preserving the freedoms and rights of the individuals over the needs of the network. To ensure this, identity authentication must occur through independent algorithms that are censorship-resistant and force-resilient and that are run in a decentralized manner.

6.2 Decentralized Identifiers (dids) v1.0

The design goals of DIDs are as follows (Section 1.2, "Version 1.0 of the W3C Recommendation 19 July 2022"), all quoted:

1. **Decentralization**. Eliminate the requirement for centralized authorities or single point failure in identifier management, including the registration of globally unique identifiers, public verification keys, services, and other information.

2. **Control**. Give entities, both human and non-human, the power to directly control their digital identifiers without the need to rely on external authorities.

3. **Privacy**. Enable entities to control the privacy of their information, including minimal, selective, and progressive disclosure of attributes or other data.

4. **Security**. Enable sufficient security for requesting parties to depend on DID documents for their required level of assurance.

5. **Proof-based**. Enable DID controllers to provide cryptographic proof when interacting with other entities.

6. **Discoverability**. Make it possible for entities to discover DIDs for other entities, to learn more about or interact with those entities.

7. **Interoperability**. Use interoperable standards so DID infrastructure can make use of existing tools and software libraries designed for interoperability.

8. **Portability**. Be system- and network-independent and enable entities to use their digital identifiers with any system that supports DIDs and DID methods.

9. **Simplicity**. Favor a reduced set of simple features to make the technology easier to understand, implement, and deploy.

10. **Extensibility**. Where possible, enable extensibility provided it does not greatly hinder interoperability, portability, or simplicity.

References

Allen, Christopher (2016) "The Path to Self-Sovereign Identity". URL: https://www.lifewithalacrity.com/article/the-path-to-self-soverereign-identity/ (visited on ~2024.8.29).

~datnut-pollen, Jonathan Paprocki (2021) "The Gang Solves the Gas Crisis". URL: https://urbit.org/blog/rollups.

Entriken, William et al. (2018) "ERC-721: Non-Fungible Token Standard". URL: https://eips.ethereum.org/EIPS/eip-721 (visited on ~2024.8.29).

Fernau, Owen (2022) "Cosmos Ecosystem Quietly Surges to $17B in TVL in Challenge to Ethereum Layer 2s". URL: https://thedefiant.io/news/markets/cosmos-tvl-surge-ethereum.

~hastuc-dibtux, Liam Fitzgerald et al. (2024) "GroundWire 01". URL: https://straylight.network/groundwire/groundwire-01 (visited on ~2024.8.29).

Ordinals (2024) "Ordinals Handbook". URL: https://docs.ordinals.com/ (visited on ~2024.8.29).

Preukschat, Alex and Drummond Reed (2021). *Self-Sovereign Identity: Decentralized digital identity and verifiable credentials*. Manning Publications. ISBN: 9781617296598.

~ravmel-ropdyl, Galen Wolfe-Pauly (2019) "Azimuth is On-Chain". URL: https://urbit.org/blog/azimuth-is-on-chain.

~sarlev-sarsen, Stuart Cristoph (2024) "Why #l1masterrace is More Than a Meme". URL: https://sarlev-sarsen.rooftopdao.io/blog/why-l1masterrace-is-more-than-a-meme.

~tacryp-socryt, Logan Allen et al. (2023) "EDEN - a practical, SNARK-friendly combinator VM and ISA". URL: https://eprint.iacr.org/2023/1021.

~wicdev-wisryt, Philip C. Monk (2020) "Urbit Precepts (Discussion)". URL:

https://urbit.org/blog/precepts-discussion (visited on ~2024.8.29).

— (2024). "Designing a Permanent Personal Identity: The Public Key Infrastructure Idea Maze." In: *Urbit Systems Technical Journal* 1.2, pp. 1–7.

Windle, Jayden et al. (2023) "ERC-6551: Non-fungible Token Bound Accounts". URL: https://eips.ethereum.org/EIPS/eip-6551 (visited on ~2024.8.29).

World Wide Web Consortium (2022a) "Decentralized Identifiers (DID) v1.0: Core architecture, data model, and representations". URL: https://www.w3.org/TR/did-core/ (visited on ~2024.8.29).

— (2022b) "Verifiable Credentials Data Model 1.1". URL: https://www.w3.org/TR/vc-data-model/ (visited on ~2024.8.29).