

HOSTING VERSION

Holo: Mainstream Access to High-Powered, Decentralized Apps

ABSTRACT: (To be written after draft is complete) Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Introduction: Why Holo? Core Problems Solved

In recent years, blockchain, and things that run on blockchain, have risen in popularity¹ and entered into more mainstream awareness.^{2 3 4} This has generated excitement about the possibilities of decentralized patterns for currencies, web applications, data ownership⁵, and collective governance⁶. However, there are two substantial hurdles to delivering on this bright promise: one technical and one strategic strategic(?) or “how-do-you-get-there-from-here”.

1. The focus on the specific architecture of blockchain has produced a kind of conceptual tunnel vision that blindly looks past its massive inefficiencies⁷ and scalability hurdles⁸
2. The migration path from the non-decentralized world of centralized apps on web-servers to a fully decentralized world is not trivial, and requires a bridging period, where people running standard web-clients can then access decentralized apps to try them out. This migration also requires a method of accounting for the value flows involved at scale.

Holo solves the second hurdle, and it is built on Holochain, which resolves the first. For scaling distributed applications, Holochain provides fundamentally more viable architecture than the blockchain.⁹ Holo then leverages that scalable architecture to accomplish two primary feats:

¹ Bitcoin and Ethereum some of Google's [most popular searches](#)

² Google Trends search popularity for [Bitcoin](#), [Blockchain](#), [Ethereum](#), [Cryptocurrency](#)

³ Stats on investment by business and banks into blockchain in 2017 TODO

⁴ Stats on ICO investment in 2017, surpassed VC funding TODO

⁵ Some link to governmental projects taking data public (Estonia?) TODO

⁶ Some P2P governance link TODO

⁷ Proof of work and energy usage TODO

⁸ Bitcoin.org scalability page? TODO

⁹ Engineering Efficiency analysis by Phillip Beadle? (not yet published)

1. Enable mainstream access to fully-functional decentralized apps, and
2. Provide the massively scalable crypto-accounting infrastructure required to power it.

This whitepaper details first of these accomplishments, and is published in tandem with another paper detailing the crypto-accounting engine which the distributed apps of Holo use to process all payments and manage transactions.¹⁰ The crypto-credits involved in that accounting system are completely reliant on the distributed computing infrastructure described here for their operation and valuation as a currency backed by computation power.

Both papers assume a thorough understanding of the conceptual and architectural shifts that give Holochain its power and scalability.

[[Failure to make dApps viable because of gas limits¹¹ and costs.¹² In practical terms, what this really means is that “smart contracts” are really just “dumb applications” which are severely hampered in their computational power and flexibility. This frustration should be familiar to anyone who has tried to build practical distributed applications. You have to offload most of the computing to centralized servers to even approach a practical implementation.¹³]]

Technology Overview

This section conveys a basic understanding of Holo’s architecture and approach. Later sections go into greater technical detail of the design of the currency, operations of the network and peered hosts, performance modelling of sub-systems, and security considerations of both the currency and hosted applications.

Built on Holochain

Since Holochain provides the underlying “crypto operating system” for Holo, the claims of what we are implementing in Holo might seem preposterous without a basic understanding of how many of the typical problems in the crypto space are solved by building on top of Holochain.

Probably the most important is the fact that Holochain provides data integrity for peer-to-peer applications without using consensus.

In this paper, the absence of references to Proof-of-Work, or Proof-of-Stake, or leader selection algorithms is not an oversight, nor due to unfamiliarity with established approaches to distributed computing or blockchain architectures. Our approach stems from the fact that Holo doesn’t need to waste computing power on those proofs, nor even on any type of

¹⁰ Reference to the CURRENCY paper

¹¹ Info about current per block limit

¹² Some info about current gas costs for computation. Maybe that article showing it is 400 million time more expensive than AWS

¹³ Someone’s experience of doing this...

global ledger consensus at all. Holochain ensures data integrity for distributed applications by carefully tracking data provenance with content addressability by hashes, cryptographic signatures, and the enforcement of data schemas and application logic.¹⁴

Therefore Holochain establishes an agent-centric pattern for data production, sharing, and management, rather than a data-centric absolutist frame (for which consensus is required).¹⁵ Once a developer learns to do this kind of inversion in their thinking, building a distributed application on Holochain tends to be much easier and many orders of magnitude more computationally efficient.

Please consult the [Holochain white paper](http://holochain.org/white-paper)¹⁶ for a more thorough understanding of it's architecture. To clarify, Holochain is not a theory in a paper. At the time of publication, Holochain is already in Alpha release with a number of early applications built on it that have been tested and proven to solve significant problems of scale and usability.

Distributed Cloud Hosting

Holo can be seen as a cloud hosting solution, where "the crowd is the cloud."

Holo runs as an application built on top of Holochain, so it automatically inherits the full peer-to-peer potential of every Holochain app. However, it is made with the express purpose of accomplishing two things that Holochain doesn't do natively: the ability to allow for hosts to support the source chains of others, in order to provide full decentralized services to people not yet running Holochain.

Holochain operates in a fully P2P world, but not everyone is ready to install new software and serve as a microhost in a P2P app. Most people expect to type a web address into their browser and reach a centralized web server where they access the applications made available on that web-site. Holo exists to provide the bridge for this kind of mainstream user to participate in the world of decentralized crypto apps where apps don't live on any single server that DNS could resolve to. Thus, part of the Holo solution includes the **Holo Application Delivery Network (ADN)**, provides DNS services and routes requests to appropriate hosts. This service is built on proven technology similar to how a CDN (Content Delivery Network) directs DNS requests to local servers with cached files.

Additionally, Holo addresses the architectural mismatch between the P2P world, where costs are distributed among the participating nodes, and the centralized world, where the application provider expects to bear the hosting expenses.

The Holo ecosystem consist of the following entities:

¹⁴ Holochain Whitepaper or github repo

¹⁵ Brock, Arthur: The Two Main Fallacies of Distributed Computing and Blockchain (TODO - link)

¹⁶ <http://holochain.org/white-paper>

End-User: a user accessing a crowd-hosted application via a standard web-browser, i.e. users who have not installed and perhaps have no knowledge of Holochain.

Holo-End-User: a user accessing a crowd-hosted app via their own installation of a Holo.host.

Host: a user who has installed both Holochain, and the Holo Holochain application, and as registered as Host with the Holo ADN.

Application-Provider: an entity or person who registers with the Infrastructure Provider as the responsible party for Holo hosted Holochain application. Application-Providers, much like in the centralized world, may either agree to be “invoiced” by Hosts who service requests from End-Users, or design their apps such that those costs are directly passed on to End-Users.

Infrastructure Provider: the Holo organization itself, which:

1. runs the Holo Application Delivery Network to manage bridging from the web to Holochain.
2. runs the compute-power, asset-backed mutual credit currency described in the companion whitepaper, along with accompanying Reserve Accounts. The credits manage the values flows between the various parties and the Reserve Accounts allow Hosts to convert Holo credits to other currencies crypto and federal.

System Components

Hosting

Since “the crowd is the cloud” in Holo, we must address a number of issues for Hosts that maintain hosts agency and security in participation while also ensuring that services are reliably delivered to End-Users.

A Safe Space for dApps (Distributed Apps)

Holo is designed to leverage idle computing capacity. In addition to Holo specific boxes devoted to hosting, people can install Holo on their desktops, or other machines allows them to share their extra space and processing power to support the personal usage and private data of others. We can’t assume that a device is dedicated to Holo hosting, so Holo must be designed to be secure and polite with other device demands.

The fact that a Holochain app runs in a virtual machine inside the Holochain engine provides a layer of isolation from access to any system resources. Holochain apps can only make calls supported by the Holochain engine, which are basically limited to reading & writing to attached Holochain data stores (source chain & DHT). Applications using Holo don’t have direct access to the underlying file system or other system resources.

However, since third party security testing is still pending, and since users can install custom skins and user interfaces, unknown vulnerabilities might be introduced. Thus, Holo isolates the Holochain engine and all installed apps by running in a container (such as Docker). The container can be configured to run “nice” and give processing priority to local applications

rather than the remote services being provided. The owner can also throttle the overall usage in terms of bandwidth, storage, and CPU use.

Selecting Hosted Applications

Other than the underlying Holochain engine for cryptographic services and the Holo application for usage tracking and payment processing, no Holo host is required to host any particular application. Hosts can explicitly install particular apps, and also block specific applications. In addition, Holo applications are categorized when added to the framework so hosts can set their own priorities and filters by app categories, price brackets, and usage demands.

If a host doesn't have much time or interest in customizing their applications, they can enable an app selection autopilot that will trigger installation of applications with more demand than available hosting power. Autopilot is a good way for the host to increase their revenue by bringing capacities online as they are needed.

Setting Hosting Prices

Hosts can also set their own prices. Some may opt to host certain applications for free (think of supporting a P2P wikipedia, or SETI, or other projects they just want to share spare computing power with). Some people may have higher thresholds than others, thinking they don't want to spend any bandwidth or electricity unless they are being paid above a certain amount.

Again, just like in application selection, hosts who don't want to pay much attention to pricing can configure an auto-pricing app. They can set basic priorities such as trying to serve the greatest demand, seeking the highest payers, or adjusting toward a nice middle of market zone to get some hosting volume without being flooded by it.

Bridging to the Centralized Web

The first web server was the personal computer sitting on Tim Berners-Lee's desk. HTTP was initially designed to be peer-to-peer. However, our network architectures, name resolution patterns, and economic models have steered the pattern of web hosting toward operating on centralized servers, on racks in data centers.

Most people don't think of the web as centralized because of its adaptive routing and the fact that different sites run on different servers. Let's look closer at how reaching a web site really works. In order to reach [holo.host](#), for instance, everyone has to resolve this name in **one** global namespace. This resolution requires asking a DNS **root** server for the nameserver for that domain to retrieve **one** address (in **one** global address space) to find the server to send the HTTP request to. That is still a centralized architecture. Many scalability strategies have emerged to circumvent the bottlenecks in this approach, including multiple root servers, multiple nameservers, rotating IPs in name resolution, load balancing IP routers, virtualizing servers

across multiple machines, etc.¹⁷ Also content delivery networks intercede at many levels to try to improve performance. However, if your native architecture was already a distributed application delivery network, then all the kludges to eliminate bottlenecks become obsolete.

In contrast to the traditional centralized approach, to use a Holochain app, each person can speak to the app UI on localhost (their own device). The app requests from the DHT (supported across everyone's installations of the app) whatever data it needs that isn't cached locally. There are **no bottlenecks or traffic hot spots**.¹⁸ The only single point of failure is someone's internet connection, in which case they could still access the UI and all data they've created, cached, or is stored in their shard of the DHT.

Holo deck

If the app isn't installed, no problem, Holochain comes with its own distributed app store. You can download and install any Holochain app from a fully distributed application. If you don't have Holochain installed, then you have to go back to the centralized web to download it from a web site.

Since the mission of Holo is to service people who don't have Holochain installed, we have to be able to translate between web browser requests traversing the "centralized" infrastructure and connect them with a variety of decentralized Holo hosts. So someone who wants to run The End User experience of a decentralized Twitter-like Holochain app on Holo could type an address into their browser and interact with it just like it was stored on a web server.

Application Delivery Network

In a similar way that a CDN (Content Delivery Network) directs DNS requests to local servers with cached files, the Holo ADN (Application Delivery Network) provides DNS services and routes requests to appropriate hosts.¹⁹ Holo sorts the list of hosts according to weights specified by the application provider (the person, group, or organization who has published the application and is responsible for maintaining it).

Host selection criteria include metrics about their historical speed and performance, prices the host is charging, trust metrics related to past behavior, and estimated latency from the requestor. This allows applications to be tuned for performance, tuned for affordability, and many other variations.

The DNS resolution engine is the only semi-centralized element of the Holo hosting architecture. It will also be load-balanced and synchronized across multiple locales, but this bit of centralization is needed to interface with existing DNS architectures²⁰.

¹⁷TODO

¹⁸ Removing DHT hot spots is a problem solved by DSHT S=sloppy <TODO>

¹⁹ TODO

²⁰ TODO

Holoquery and Holo Browser Extension

Most Holo hosts will probably be operating on small Holo boxes rather than on dedicated rackmount servers or in data centers on high speed Internet backbones. Since many of these connections may have slower upload speeds for serving out data, Holo includes a strategy to optimize the speed of an end user's experience in their web browser.

Holo will provide a series of JavaScript libraries so that an application UI can optimize loading times. Just like jQuery is cached on most people's browsers and provides a variety of Javascript services,²¹ Holoquery will be cached by its frequent users. In turn, it updates a cache of IP addresses for host boxes so that it can send many parallel requests to load page elements (i.e. images, video, etc.) from different hosts so that you are not constrained to the bandwidth of a single host. It also supervises loading of elements to try a different host if one is failing to be responsive enough.

Holoquery will provide private/public key generation and storage used for signing new source chain entries, and passing signatures with page requests, which are needed for the host's Proof-of-Service logs and invoices.²²

End-users can also install a Holo browser extension to use applications which require creation and storage of private chain entries. This extension could also be used to maintain the state of private information across multiple applications or devices, and to have Holo store encrypted backups of their keys or source chains.

Thorough performance testing will certainly be required, but through these core mechanisms, Holo should be able to provide enterprise-scale performance on a network of small peered computing devices.

Holochain Upgrades

Although Holochain provides most of the underlying componentry that we need for Holo, there are still some modifications and improvements needed to Holochain's core code.

Source Chain Segmentation

Most importantly is separating the ability to author and sign new source chain entries to a chain from the ability to respond to other peers as a steward of someone's source chain. Hosts will never have user's private keys, so any changes will need to be signed by a user to create new entries on their source chain. However, source chain headers and "public" entries will need to be stored with at least some hosts in order to validate them when they are shared to the DHT.

²¹ TODO

²² See Currency White Paper

As already mentioned, Holochain was designed as a complete P2P architecture, so it doesn't natively accommodate one peer hosting another peer's private chain entries. Holo applications will run in a specific mode that segments functionalities, allowing a peer to operate as a proxy for a user in a read-only capacity. In this mode, peers on the DHT, who make validation requests against proxied source chain data, will accept *communications* signed by proxy keys as long as the content being validated has been signed by the proper user keys.

This network/DHT feature will need to be explicitly enabled in the application settings for apps to run on Holo with proxied source chains. In this situation, the DNS registry will also track the address of a core proxy ID for each source chain ID to efficiently route a returning user's connection. For ease and speed of re-establishing context, only hosts with high uptime records will be assigned as core proxies,. As well, the core proxy functions at the center of a cluster of peers who can also replicate source chain data, so that a person can resume context even when their core proxy is offline.

Deeper Segmentation for Source Chain Privacy

Once private keys and source chain services are performing adequately we have outlined some deeper layers for segmenting source chains. These involve breaking entries into parts, which no single host holds, and are only reassembled en route to authorized requests.

We are delaying this approach to focus on basic performance requirements before adding more layers of fragmentation and complexity. We also don't want to provide an illusion of privacy. As source chains services need to be provided to various peers to validate propagation of data, we cannot encrypt the data to only be released by a known key as every node would need the key.

This is a common PK granularity as well.

There also seems to be little practical way to prevent nodes with different parts from colluding and sharing the data, or in fact, all be controlled by one party. See the later section on "Security Considerations for Applications" for a thorough explication of privacy issues.

Other Important Milestones

There are a few improvements in the development queue for Holochain that Holo will benefit from and will be important for Holo hosts.

Reputation Upgrades: Holochain already has basic infrastructure for nodes to reject data as fraudulent, and to report fraudulent behavior to the neighboring peers of a bad actor. Evidence is provided in the form of a "warrant" that includes the original fraudulent data signed by its author. The native immune system of Holochain can use the accumulation of these warrants as the basis for blacklisting corrupted nodes. Additional enhancements are planned for collecting better performance metrics of your neighborhood peers, and enabling the possibility of defining node service classes.

Pluggable Governance modules that enable applications to manage forking of their DNA are planned for Holochain's Alpha 1 release. This will be helpful for application providers when managing new releases of their distributed software, and enabling easy upgrades for their users. Within this governance process, upgraded apps can write a closing entry onto a user's old source chain that points to their new source chain in the updated app. The new source chain in the updated app will also have opening entries that point back to the old source chain. This provides continuity of function and identity with clean versioning of code and protocols.

Scalability, Security, & Optimization: Holochain will undergo increasingly stringent scalability testing, security vetting, and performance optimizations that will continue to mature the platform and ensure its reliability for mission critical applications.

System Details

Countersigned Transactions

All transactions are recorded as double-entry accounting records to the source chains of each agent in the transaction. As a general use example (not specifically for hosting), Alice agrees to sell a bike to Bob for some Holo credits. They communicate with each other via node-to-node messaging to build a transaction, one party initiates, leaving empty the fields for the counterparty to complete.

After constructing the transaction, and exchanging the data required to validate the transaction and audit the spenders chain to validate they have the credits they're spending, then the receiver signs it by encrypting the hash of the data with their private key. After the spender responds with their signature, they both commit it to their own source chains with the record of the counterparty's signature.

Note that BOTH parties actively participate in the transaction, and must sign it to their chains. This is not the spending of a coin by a single keyholder, but a mutually agreed upon transaction with the opportunity to validate the other party's state before transacting. This again underscores why no global consensus is needed. All nodes have the same validation rules, and if Alice can't validate that Bob can spend the credits he's spending, then her app will reject the transaction. It doesn't matter if others before her colluded with Bob. Every non-colluding actor will reject illegal operations.

Pre-Authorized Tokens for Automated Transactions

The pre-authorization tokens are essentially a mechanism to optimize the speed of transactions which require mutual consent, by having one party provide explicit permission in advance. This massively reduces transaction time to computer automated speeds, eliminating the need to notify a person and obtain their approval. Pre-authorization tokens turns a process which is normally synchronous, into an asynchronous one -- or rather, minimize the synchronous messaging to providing a current top-hash and digital signature.

Roles and Responsibilities,

Holo credits are designed to power distributed applications operated by a network of hosts who provide computing power, therefore, specific roles and responsibilities are defined within this ecosystem. Most of these roles can be stacked such that an agent (a node with private/public keys and the ability to interact with Holo and its hosted applications) may hold multiple roles. However, the last two roles listed below are reserved for special functions and cannot be stacked with other functions.

Hosts: Most of this paper has been about how Holo hosts operate, so this role may largely be clear. However, the terms hosts are agreeing to, and the privileges they receive may not be clear. Hosts have the rights to copy and run applications on Holochain, and copy and publish user data, but that does not grant them ownership of that data. If an Application Provider removes an app, an End-User removes their account, the host must purge that app and that user data from their system. This is easy to verify and non-compliant hosts can be blacklisted.

End-Users: Holo is designed to help them reach their applications, to keep their data out of centralized services, and to make it safe and easy to do crypto-transactions. There are no special responsibilities they have, nor additional privileges they receive, like all user roles they agree to follow the rules encoded in the Holo app DNA, which includes payment of transaction fees when they accumulate to the payment threshold.

App Providers: App Providers are responsible for maintenance and security of apps they publish on Holo. They also agree to timely payment of Proof-of-Service invoices. Since unpaid invoices are visible to all, it is easy enough for a host to demonstrate a failure to pay. No central authority needs to intervene, and no smart-contract needs to enforce payment. Hosts can configure their app selection preferences to filter on age of unpaid invoices. Hosts will simply stop serving their app if they fail to pay. The benefit of accountability is that everything doesn't need to be automated into a smart contract, removing all human choice and agency. Holo optimizes the feedback loops so people can make smarter choice and still maintain their power to choose. Maybe you want to support a start up project that can't pay yet. Maybe you're friends with them. Maybe you can take the risk with slow payers and just charge them a little more. Why should a one-size-fits-all smart contract replace people's power to choose. When it's practical to have a smart contract be as sophisticated as the entire Holo system, you can both automate optimal functioning and include human choice.

App Developers: *Uh... I'm tired of this section and don't feel like writing this right now. TODO*

Reserve Accounts: These accounts are used to enable hosting providers to redeem Holo credits for other currencies which have been used to purchase Holo. Redeemed credits must have been earned for providing hosting, which can be confirmed by auditing their Proof-of-Service entries and transaction history. Reserve Accounts have a special algorithm for expanding their

credit limit by committing a Proof-of-Reserve record to their chain. This record demonstrates receipt of a payment for Holo credits which is then held in reserve for later redemption by Holo hosts who have met the redemption requirements (providing transfer methods, and probably some forms of real world identity verification).

Infrastructure Provider: In this case, the Holo organization is the infrastructure provider and the main benefit they receive is transaction fees for maintaining and updating the software, the DNS services, and other centralized components required for interfacing with the “centralized” web.

Intelligent Self-Governance

The most fundamental “terms and conditions” of the system are the code written into the Holo app DNA. However, Holo is also a hosted commons with expected standards of behavior. You cannot encode all social contracts into smart contracts -- you shouldn’t even try²³. Instead, you can facilitate better collective intelligence and healthy feedback loops that enhance people’s ability to choose who they trust and detect unexpected, unpredictable cheating and fraud. All types of users are incentivized not to defraud others on the system with the understanding that when they are caught all privileges on Holo may be revoked. Keep in mind that EVERY communication and data element is signed by its author to their immutable chain, or it cannot propagate. If you are a bad actor, you have published a non-repudiable record of your actions -- you’ve left your digital fingerprints all over the scene of the crime.

This enables the Holo ecosystem to have a high-functioning “immune system,” because any node can create a “warrant” which flags fraudulent behaviors and provides the original signed records of the fraudster as proof. As new types of fraud are identified and able to be detected, it is easy to determine who has already done those things, then spread warrants as proof of fraud, so other nodes can opt to blacklist them.

Another agreement for all Holo users is to not generate automated or artificial traffic (to boost your own or your buddy’s hosting fees). If a developer needs to test app scalability they must do that on Holochain or on a distinct Holo testing network where test nodes (likely supplied by app providers and developers) agree to provide each other free computing space purely for testing purposes.

²³ On not being able to reduce social process to a smart contract. (If we can’t find something then we can reference Art’s article ([or more specific response](#)) on not taking humans out of the equation.

Modelling Holo Dynamics (This is where we put the math and technical stuff Eric wants included)

App Performance

- Comparison of Computational Efficiency: Holochain vs. Ethereum
- Browser side Holoquery assistance with local computation and balancing
- Optimizations: Minimal latency. Simultaneous requests. Segmentation of static/dynamic content.
- Semi-Centralized functions: Round-robin IPs for DNS resolution, client DNS caching minimizes hits on central infrastructure,

Holo Network Scalability

- Each app runs in its own holochain. Not monolithic single ledger. Only the parties running that chain need to become eventually consistent with their shard of the data. Doing async computation to validate
- Holo is also a holochain app. Every host is running in the holo app, but it is separate/distinct from the apps which use Holo for service accounting. They do not block each other.
- Holo neighborhoods track peer reputation on performance (uptime, bandwidth, latency, proc speed, etc) & behavior (re: trust, fraud, payments, promptness, etc.)
- App neighborhoods track peer reputation according to rules of that app
- Periodic (daily?) reporting by peer of their own stats and nearest neighbors.
- Modular structure to Holo app to enable evolution of component parts (reputation tracking, logging, invoicing, transactions, app selector, price selector, etc.)

*Small
want
reputation
Spent
Selling*

Expected Host Box Performance

- Formula for modeling Simultaneous App Services on variants of hardware configurations (which happen to match the specs for our 3 boxes)
 - Expected performance on low-power device - Initial load testing on Raspberry Pi.
 - Holo vs. Ethereum minimum system requirements for earning // Approx \$2100²⁴ for full node (mining rig) 8GB RAM + 3GB per GPU
- Expected mining income²⁵

*Holo producing 3 Places
of B.N. What are they
target? Some Design
Available?
Price?*

²⁴ Good Ethereum hardware [config](#) post

²⁵ Ethereum mining profitability calculator <http://badmofo.github.io/ethereum-mining-calculator/>

Security Considerations

Holo Application Security

Privacy on Holo vs. Holochain

Remember that Holo is bridging backward to the “centralized” web. As such, data and computation that is one peer’s responsibility to hold and perform is being delegated to other peers. If you really don’t like the idea of your data being hosted by other unknown netizens, then all you have to do is install Holochain and host yourself. There’s certainly no privacy problem there.

Additionally, one of the core issues of centralized app providers is not just that they can see your data, but that they can analyze your complete usage patterns.²⁶ Realistically, these meta-patterns of data usage are far more likely to be exploited and exploitable by centralized corporations doing meta-analysis on their millions/billions of users, than by a handful of small hosters who only see a small portion of the traffic for an application or a user. Holo hosts will have a small subset of users’ data, and will certainly tend to be much less sophisticated analysts.

Network Communications

End to end encryption + gossip obscuring actual traffic

Node-to-Node Messaging

Can be Encrypted... but if you want to hide traffic patterns of communications, they would need to be split among various proxies. This is certainly feasible for Holoquery to do, since it will have address info for multiple proxies. This feature is being considered for later releases of Holo.

Public Entries / Published Data

Already public, hosts just have public source chain collected in one place instead of having to assemble them from different nodes in the DHT (which anyone could theoretically do if headers are published).

Encryption vs. Private Data

Blockchain only allows encryption. Holochain enables you to share encrypted content, OR hold private entries which are only shared by direct interaction with the author. This variability has many advantages over encryption as the only option. Encryption only methods will probably be compromised, Keys can be compromised, and data once hidden can be selectively released.

²⁶ Article about what (telegram?) knows or doesn’t just from usage pattern data

Vulnerability of Source Chains

Nobody can create new entries without keys. Proxies might be able to observe usage patterns, but much usage traffic will be divvied up between many other nodes.

Private Entries on Source Chains

In theory, it may seem easy to break source chain entries into parts and distribute those parts among different hosts, and provide a method for a requestor to reassemble the needed components. However, in practice preventing nodes from colluding or being under unified control seems out of scope for Holo at this stage in its life cycle.

Private chain entries could

Conclusions / Resulting System Dynamics

Puts scalable infrastructure capacities into the hands of normal people, not just well-capitalized mega-corps

Enables app providers to scale hosting/computing needs with minimal capital investment.

When bandwidth is available, you can install Holo to tap into idle computing capacity for productive use. Enabling passive income from sharing computing resources.

Establishes an asset-backed, value-stable, currency with natural "breathing" supply - provides the stability for next generation of adopters to cross the chasm to crypto adoption

Movement from the petro-dollar to processing-credits