Workgroup: Network Working Group
Internet-Draft:
draft-nottingham-avoiding-internet-centralization-00
Published: 8 December 2021
Intended Status: Informational
Expires: 11 June 2022
Authors: M. Nottingham

# Avoiding Internet Centralization

## Abstract

Avoiding centralization is an important goal for Internet protocols. This document offers a definition of centralization, discusses why it is necessary for Internet protocol designers to consider its risks, identifies different kinds of centralization, catalogues some limitations of current approaches to controlling it, and recommends best practices for protocol designers.

## About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-nottingham-avoiding-internet-centralization/.

Source for this draft and an issue tracker can be found at https://github.com/mnot/avoiding-internet-centralization.

## Status of This Memo

**Table of Contents**

1. Introduction
  1.1. Notational Conventions
2. Why Avoid Centralization
3. Kinds of Centralization
  3.1. Direct Centralization
  3.2. Necessary Centralization
  3.3. Indirect Centralization
  3.4. Inherited Centralization
  3.5. Platform Centralization
4. The Limits of Decentralization
  4.1. Federation isn't Enough
  4.2. Multi-Stakeholder Administration is Hard
  4.3. Blockchains Are Not Magical
5. Guidelines for Protocol Designers
  5.1. Allow Intermediation Sparingly
  5.2. Encrypt, Always
  5.3. Reuse Existing Tools
  5.4. Accomodate Limited Domains Warily
  5.5. Target Extensibility
  5.6. Acknowledge the Limits of Protocol Design
6. Security Considerations
7. References
  7.1. Normative References
  7.2. Informative References
Appendix A. Acknowledgements
Author's Address

## 1. Introduction

The Internet is successful in no small part because of its purposeful avoidance of any single controlling entity. While originally this may have been due to a desire to prevent a single

technical failure from having wide impact, it has also enabled the rapid adoption and broad spread of the Internet, because internetworking does not require obtaining permission from or ceding control to another entity -- thereby accommodating a spectrum of requirements and positioning the Internet as a public good.

As a result, Internet protocols share a common design goal: avoiding centralization, which we define as the ability of a single person, company, or government -- or a small group of them -- to observe, control, or extract rent from the protocol's operation or use.

At the same time, the utility of many Internet protocols is enabled or significantly enhanced by ceding some aspect of communication between two parties to a third party -- often, in a manner that has centralization risk. For example, there might be a need for a 'single source of truth' or a rendezvous facility to allow endpoints to find each other. How should such protocols be designed?

Furthermore, many successful proprietary protocols and applications on the Internet are de facto centralized. Some have become so well-known that they are commonly mistaken for the Internet itself. In other cases, Internet protocols seem to favour centralized deployments due to economic and social factors. Should standards efforts attempt to mitigate centralization in these cases, and if so, how?

Finally, some autonomous networks have requirements to control the operation of Internet protocols internally, and some users or groups of users might cede control of some aspect of how they use the Internet to a central authority, either voluntarily or under legal compulsion. In both of these cases, should Internet protocols accommodate such requirements, and if so, how?

This document discusses aspects of centralization with regard to Internet protocol design (note that 'protocol' is used somewhat loosely here, to also encompass what could be considered an application). Section 2 explains why it is necessary for Internet protocols to avoid centralization when possible. Section 3 surveys the different kinds of centralization that Internet protocols might be involved in. Section 4 then catalogues current high-level approaches to mitigating centralization and discusses their limitations. Finally, Section 5 discusses cross-cutting interactions between centralization and protocol design, recommending best practices where appropriate.

Engineers who design and standardize Internet protocols are the primary audience for this document. However, designers of proprietary protocols can benefit from considering aspects of centralization, especially if they intend their protocol to be

considered for standardisation. Likewise, policymakers can use this document to help identify and remedy inappropriately centralized protocols and applications.

### 1.1. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 2. Why Avoid Centralization

Centralization is undesirable in the design of Internet protocols for many reasons -- in particular, because it is counter to the nature of the Internet, because it violates the purpose of the Internet from the perspective of its end users, and because of the many negative effects it can have on the networks operation and evolution.

By its very nature, the Internet must avoid centralization. As a 'large, heterogeneous collection of interconnected systems' [BCP95] the Internet is often characterised as a 'network of networks'. As such, these networks relate as peers who agree to facilitate communication, rather than having a relationship of subservience to others' requirements or coercion by them.

However, many Internet protocols allow a third party to be interposed into communication between two other parties. In some cases, this is not intended by the protocol's designers; for example, intervening networks have taken advantage of unencrypted deployment of HTTP [HTTP] to interpose 'interception proxies' (also known as 'transparent proxies') to cache, filter, track, or change traffic. In cases where interposition of a third party is a designed feature of the protocol, it is often characterised as *intermediation*, and is typically used to help provide the protocol's functions -- sometimes including those that are necessary for it to operate.

Whether or not interposition of a third party into communication is intentional, the 'informational and positional advantages' [INTERMEDIARY-INFLUENCE] gained can be used to observe behavior (the 'panopticon effect') and shape or even deny behaviour (the 'chokepoint effect') -- which can be used those parties (or the states that have authority over them) for coercive ends. [WEAPONIZED-INTERDEPENDENCE]

As Internet protocols' first duty is to the end user [RFC8890], allowing such power to be concentrated into few hands is counter to

the IETF's mission of creating an Internet that 'will help us to build a better human society.' [BCP95]

Additionally, concentration of power has deleterious effects on the Internet itself, including:

* *Limiting Innovation*: Centralization can preclude the possibility of 'permissionless innovation' -- the ability to deploy new, unforeseen applications without requiring coordination with parties other than those you are communicating with.

* *Constraining Competition*: The Internet and its users benefit from robust competition when applications and services are available from many different providers -- especially when those users can build their own applications and services based upon interoperable standards. When dependencies are formed on a centralized service or platform, it effectively becomes an essential facility, which encourages abuse of power.

* *Reducing Availability*: The Internet's availability (as well as applications and services built upon it) improves when there are many ways to obtain access to it. While centralized services typically benefit from the focused attention that their elevated role requires, when they do fail the resulting loss of availability can have disproportionate impact.

* *Creating Monoculture*: At the scale available to a centralized service or application, minor flaws in features such as recommendation algorithms can be magnified to a degree that can have broad (even societal) consequences. Diversity in these functions is significantly more robust, when viewed systemically. [POLYCENTRIC]

* *Self-Reinforcement*: As widely noted (see, eg., [ACCESS]), a centralized service benefits from access to data which can be used to further improve its offerings, while denying such access to others.

To summarize, we avoid centralization because it would allow the Internet (or some part of it) to be captured, effectively turning it into a 'walled garden' that fails to meet both architectural design goals and users' expectations, while endangering the viability of the Internet at the same time.

## 3. Kinds of Centralization

Not all centralization of Internet protocols is equal; there are several different types, each with its own properties. The subsections below list some.

### 3.1. Direct Centralization

The most straightforward kind of centralized protocol creates a fixed role for a specific party.

For example, most proprietary messaging, videoconferencing, chat, and simliar protocols operate in this fashion.

While it has been argued that such protocols are simpler to design, more amenable to evolution, and more likely to meet user needs, [MOXIE] this approach most often reflects commercial goals -- in particular, a strong desire to capture the financial benefits of the protocol by 'locking in' users to a proprietary service.

Directly centralised protocols and applications are not considered to be part of the Internet per se; instead, they are more properly characterized as proprietary protocols that are built on top of the Internet. As such, they are not regulated by the Internet architecture or standards, beyond the constraints that the underlying protocols (e.g., TCP, IP, HTTP) impose.

### 3.2. Necessary Centralization

Some protocols require the introduction of centralization risk that is unavoidable by nature.

For example, when there is a need a single, globally coordinated 'source of truth', that facility is by nature centralized. The most obvious instance is seen in the Domain Name System (DNS), which allows human-friendly naming to be converted into network addresses in a globally consistent fashion.

Allocation of IP addresses is another example of a necessary facility being a centralization risk. Internet routing requires addresses to be allocated uniquely, but if the addressing function were captured by a single government or company, the entire Internet would be at risk of abuse by that entity.

Similarly, the need for coordination in the Web's trust model brings centralization risk, because a Certificate Authority (CA) can control communication between the Web sites that they sign certificates for and users whose browsers trust the CA's root certificates.

Protocols that need to solve the 'rendezvous problem' to coordinate communication between two parties that are not in direct contact also suffer from this kind of centralization risk. For example, chat protocols need a way to coordinate communication between two parties that wish to talk; while the actual communication can be direct

between them (so long as the protocol facilitates that), the endpoints' mutual discovery typically requires a third party.

Internet protocols currently tend to mitigate necessary centralization using measures such as mandated federation Section 4.1 and multi-stakeholder administration Section 4.2.

### 3.3. Indirect Centralization

Even when a protocol avoids direct centralization and does not exhibit any necessary centralization, it might become centralized in practice when external factors influence its deployment.

Indirect centralization can be caused by factors that encourage use of a central facility despite the absence of such a requirement in the protocol itself. Such factors might be economic, social, or legal.

For example, cloud computing is used to deploy many Internet protocols. Although the base concepts and control protocols for it avoid centralization in the sense that there is no need for a single, central cloud provider, the economics of providing compute at scale as well as some social factors regarding developer familiarity and comfort encourage convergence on a small number of cloud providers.

Often, the factors driving indirect centralization are related to the network effects that are so often seen on the Internet. While in theory every node on the Internet is equal, in practice some nodes are much more connected than others: for example, just a few sites drive much of the traffic on the Web. While expected and observed in many kinds of networks [SCALE-FREE], network effects award asymmetric power to nodes that act as intermediaries to communication.

Left unchecked, these factors can cause a potentially decentralized application to become directly centralised, because the central facility has leverage to 'lock in' users. For example, social networking is an application that is currently supplied by a small number of directly centralized, proprietary platforms despite standardization efforts (see, e.g., [W3C.CR-activitystreams-core-20161215]), due to the powerful network effects associated.

By its nature, indirect centralization is difficult to avoid in protocol design, and federated protocols are particularly vulnerable to it (see Section 4.1).

### 3.4. Inherited Centralization

Most Internet protocols depend on other, 'lower-layer' protocols. The features, deployment, and operation of these dependencies can surface centralization risk into protocols operating 'on top' of them.

For example, the network between endpoints can introduce centralization risk to application-layer protocols, because it is necessary for communication and therefore has power over it. A given network might block access to, slow down, or modify the content of various application protocols or specific services for financial, political, operational, or criminal reasons, thereby creating pressure to use other services, which can in turn result in centralization.

Inherited centralization risk is only present when users cannot use an alternative means of accessing the desired service. For example, users often have flexibility in choice of Internet access, so they could just 'route around' a network that impacts their chosen service. However, such choices are often not available in the moment, and the Internet's topology means that a 'choke point' upstream could still affect their Internet access.

Usually, inherited centralization -- both existing and anticipated -- is a factor to work around in protocol design, just as any other constraint would be. One effective tool for doing so is encryption, discussed further in Section 5.2.

### 3.5. Platform Centralization

The complement to inherited centralization is platform centralization -- where a protocol does not directly define a central role, but could facilitate centralization in the applications it supports.

For example, HTTP [HTTP] in itself is not considered a centralized protocol; interoperable servers are relatively easy to instantiate, and multiple clients are available. It can be used without central coordination beyond that provided by DNS, as discussed above.

However, applications built on top of HTTP (as well as the rest of the 'Web Platform') often exhibit centralization. As such, HTTP is an example of a platform for centralization -- while the protocol itself is not centralized, it does facilitate the creation of centralized services and applications.

Like indirect centralization, platform centralization is difficult to completely avoid in protocol design. Because of the layered nature of the Internet, most protocols are designed to allow

considerable flexibility in how they are used, often in a way that it becomes attractive to form a dependency on one party's operation. Notably, this can happen even if the protocol does not accommodate intermediation explicitly.

## 4. The Limits of Decentralization

### 4.1. Federation isn't Enough

A widely known technique for avoiding centralization in Internet protocols is federation - that is, designing them in such a way that new instances of any intermediary or otherwise centralized function are relatively easy to create, and they are able to maintain interoperability and connectivity with other instances.

For example, SMTP [RFC5321] is the basis of the e-mail suite of protocols, which has two functions that are necessarily centralized:

1. Giving each user a globally unique address, and

2. Routing messages to the user, even when they change network locations or are disconnected for long periods of time.

E-mail reuses DNS to mitigating first risk (see Section 5.3). To mitigate the second, it defines an intermediary role for routing users' messages, the Message Transfer Agent (MTA). By allowing anyone to deploy a MTA and defining rules for interconnecting them, the protocol's users avoid the need for a single, central router.

Users can (and often do) choose to delegate that role to someone else, or run their own MTA. However, running your own mail server has become difficult, due to the likelihood of a small MTA being classified as a spam source. Because large MTA operaters are widely known and have greater impact if their operation is affected, they are less likely to be classified as such, thereby indirectly centralizing the protocol's operation (see Section 3.3).

This illustrates that while federation can be effective at avoiding direct centralization and managing necessary centralization, federated protocols are still vulnerable to indirect centralization, and may exhibit platform centralization.

Another example of a federated Internet protocol is XMPP [RFC6120], supporting 'instant messaging' and similar functionality. Like e-mail, it reuses DNS for naming and requires federation to facilitate rendezvous of users from different systems.

While some deployments of XMPP do support truly federated messaging (i.e., a person using service A can interoperably chat with someone using service B), many of the largest do not. Because federation is

voluntary, some operators made a decision to capture their users into a single service, rather than provide the benefits of global interoperability.

The examples above show that federation can be a useful technique to avoid direct centralization, but on its own is not sufficient to avoid indirect centralization. If the value provided by a protocol can be captured by a single entity, they may use the protocol as a platform to obtain a 'winner take all' outcome -- a significant risk with many Internet protocols, since network effects often promote such outcomes. Likewise, external factors (such as spam control) might naturally 'tilt the table' towards a few operators of these protocols.

### 4.2. Multi-Stakeholder Administration is Hard

Delegating the administration of a necessarily centralized function (see Section 3.2) to a multi-stakeholder body is an onerous but sometimes necessary way to mitigate the undesirable effects.

A multi-stakeholder body is an institution that includes representatives of the different kinds of parties that are affected by the system's operation ('stakeholders') in an attempt to make well-reasoned, broadly agreed-to, and authoritative decisions.

The most relevant example of this technique is the administration of the Domain Name System [RFC1035], which as a 'single source of truth' requires centralization of the naming function. To mitigate centralization, this task is carried out by multiple root servers that are administered by separate operators -- themselves diverse in geography and a selection of corporate entities, non-profits and government bodies from many jurisdictions and affiliations. Furthermore, those operators are regulated by ICANN, which is defined as a globally multi-stakeholder body with representation from a end users, governments, operators, and others.

Another example of multi-stakeholderism is the standardization of Internet protocols themselves. Because a specification effectively controls the behavior of implementations that are conformant with it, the standardization process can be seen as a single point of control. As a result, Internet standards bodies like the IETF allow open participation and contribution, make decisions in an open and accountable way, have a well-defined process for making (and when necessary, appealing) decisions, and take into account the views of different stakeholder groups [RFC8890].

Yet another example is the administration of the Web's trust model, implemented by Web browsers as relying parties and Certificate Authorities as trust anchors. To assure that all parties meet the

operational and security requirements necessary to provide the desired properties, the CA/Browser Forum was established as an oversight body that involves both of those parties as stakeholders.

In each of these examples, setup and ongoing operation of a multi-stakeholder organization is not trivial. This is the major downside of such an approach. Additionally, the legitimacy of such an organization cannot be assumed, and may be difficult to establish and maintain (see, eg, [LEGITIMACY-MULTI]). This concern is especially relevant if the function being coordinated is broad, complex, and/or contentious.

### 4.3. Blockchains Are Not Magical

Increasingly, distributed consensus technologies such as the blockchain are touted as a solution to centralization issues. A complete survey of this rapidly-changing area is beyond the scope of this document, but at a high level, we can generalise about their properties.

These techniques avoid centralization risk by distributing intermediary or otherwise potentially centralized functions to members of a large pool of protocol participants. Verification of proper performance of a function is typically guaranteed using cryptographic techniques (often, an append-only transaction ledger). The assignment of a particular task to a node for handling usually cannot be predicted or controlled. To assure diversity in the pool of participants (thereby preventing Sybil attacks), techniques such as proof-of-work (where each participant has to demonstrate significant consumption of resources) or proof-of-stake (where each participant has some other incentive to execute correctly) are used.

As such, these techniques purposefully disallow direct centralization and are robust against inherited centralization. Depending upon the application in question, indirect and platform centralization may still be possible, but in general these techniques do not lend themselves to these ends as readily as federated systems do.

However, distributed consensus technologies have several potential shortcomings that may make them inappropriate -- or at least difficult to use -- for many Internet applications, because their use conflicts with other important goals:

1. Distributed consensus protocols can have significant implications for privacy. Because activity (such as queries or transactions) are shared with many unknown parties, they have very different privacy properties than traditional client/server protocols. Mitigations (e.g., Private Information

Retrieval; see, eg, [PIR]) are still not suitable for broad deployment.

2. Their complexity and 'chattiness' typically results in significantly less efficient use of the network. When distributed consensus protocols use proof-of-work, energy consumption can become significant (to the point where some jurisdictions have banned its use).

3. Distributed consensus protocols are still not proven to scale to the degree expected of successful Internet protocols. In particular, relying on unknown third parties to deliver functionality can introduce variability in latency, availability, and throughput. This is a marked change for applications with high expectations for these properties (e.g., commercial Web services).

4. By design, distributed consensus protocols diffuse responsibility for a function among several, difficult-to-identify parties. While this may be an effective way to prevent many kinds of centralization, it also means that making someone accountable for how the function is performed is impossible, beyond the bounds of the protocol's design.

It is also important to recognise that a protocol can use distributed consensus for some functions, but still have centralization risk elsewhere. Even when distributed consensus is used exclusively (which is uncommon, due to the associated costs), some degree of coordination is still necessary -- whether that be through governance of the function itself, creation of shared implementations, or documentation of shared wire protocols. That represents centralization risk, just at a different layer (inherited or platform, depending on the circumstances).

These potential shortcomings do not rule out the use of distributed consensus technologies for every use case. They do, however, caution against relying upon these technologies uncritically.

## 5. Guidelines for Protocol Designers

While the following recommendations are not a complete guide, they can be a starting point for avoiding or mitigating centralization in Internet protocols.

### 5.1. Allow Intermediation Sparingly

The introduction of an intermediary role -- i.e., one that performs a function but is not a first party to communication -- adds centralization risk to Internet protocols, because it brings opportunities for control and observation. Even when the protocol is

federated (see Section 4.1) to avoid direct centralization, significant indirect centralization risks exist when intermediation is allowed.

However, intermediation can sometimes add significant value to a protocol, or enable what is considered a necessary function. In such cases, the centralized function SHOULD be as minimal as possible, and expose only the information and pontential for control necessary for that function to be performed. Protocol designers SHOULD consider the likely deployment patterns for those intermediaries and how network effects and other factors will influence them.

Such predictions can be difficult. For example, an intermediary interposed by the end user of a protocol might allow them to delegate functions to a party they trust, thereby empowering them. However, if an intervening network is able to force users to delegate to a particular intermediary, inherited centralization could result.

When carefully considered, intermediation can be a powerful way to enforce functional boundaries -- for example, to reduce the need for users to trust potentially malicious endpoints, as seen in the so-called 'oblivious' protocols currently in development (e.g., [I-D.pauly-dprive-oblivious-doh]) that allow end users to hide their identity from services, while still accessing them.

The same advice applies in these cases; the observation and control potential SHOULD be as minimal as possible, while still meeting the design goals of the protocol.

See [I-D.thomson-tmi] for more guidance.

### 5.2. Encrypt, Always

When deployed at scale, encryption can be an effective technique to reduce many inherited centralization risks. By reducing the number of parties who have access to content of communication, the ability of lower-layer protocols and intermediaries at those layers to interfere with or observe is precluded. Even when they can still prevent communication, the use of encryption makes it more difficult to discriminate the target from other traffic.

Note that the benefits are most pronounced when the majority (if not all) traffic is encrypted. As a result, protocols SHOULD be encrypted by default.

See also [RFC7258].

### 5.3. Reuse Existing Tools

When a protocol function has necessary centralization risk and there exists an already-deployed solution with appropriate mitigations, that solution should be reused in favour of inventing a new one.

For example, if a protocol requires a coordinated, global naming function, reusing the Domain Name System is preferable to establishing a new system, because its centralization risk is known and understood (see Section 4.2).

### 5.4. Accomodate Limited Domains Warily

[RFC8799] explores a class of protocols that operate in 'limited domains' -- that is, they are not intended to be 'full' Internet protocols with broad applicability, but instead operation within a particular network or other constrained environment.

Often, limited-domain protocols address network requirements -- for example, imposing security policy, integrating services or application functions into the network, or differentiating different classes of network services.

Such network-centric requirements can introduce the risk of inherited centralization when they allow the network to interpose itself and its requirements between the endpoints of a given communication.

These risks can be partially mitigated by requiring such functions to be opted into by one or both endpoints (once both the network and the endpoint are authenticated to each other), so that the network is acting on their behalf. However, this approach is still vulnerable to indirect centralization, because the endpoints may be pressured to acquiesce to a network's demands.

### 5.5. Target Extensibility

An important feature of Internet protocols is their ability to evolve over time, so that they can meet new requirements and adapt to new conditions without requiring a 'flag day' to convert users. Typically, protocol evolution is accommodated through extension mechanisms, where optional features can be added over time in an interoperable fashion.

Protocol extensions can bring risk of platform centralization if a powerful entity can change the target for meaningful interoperability by adding proprietary extensions to a standard protocol. This is especially true when the core standard does not itself provide sufficient utility to be appealing on its own.

For example, the SOAP protocol [SOAP] was an extremely flexible framework, allowing vendors to attempt to capture the market by requiring use of their preferred extensions to interoperate.

This kind of centralization risk can be mitigated in a few ways. First and foremost, Internet protocols SHOULD provide concrete utility to the majority of their users as published; 'framework' standards facilitate this kind of risk.

Furthermore, Internet protocols SHOULD NOT make every aspect of their operation extensible; extension points SHOULD be reasoned, appropriate boundaries for flexibility and control. When extension points are defined, they SHOULD NOT allow an extension to declare itself to be mandatory-to-interoperate, as that pattern invites abuse.

### 5.6. Acknowledge the Limits of Protocol Design

Centralization cannot be prevented through protocol design and standardization efforts alone. While the guidelines above may forestall some types of centralization, indirect and platform centralization are often outside the control of a protocol's architecture.

Thankfully, architecture is not the only form of regulation; legal mechanisms combined with changing norms and the resulting market forces have their own regulatory effects. [NEW-CHICAGO]

In this view, the job of a protocol designer is to avoid centralization with architecture where possible, but where it is not, to create affordances for these other regulating forces.

## 6. Security Considerations

This document does not have direct security impact on Internet protocols. However, failure to consider centralization risks might result in a myriad of security issues.

## 7. References

### 7.1. Normative References


**[RFC2119]** Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

**[RFC8174]** Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

### 7.2. Informative References


**[ACCESS]** Vestager, M., "Defending Competition in a Digitised World, Address at the European Consumer and Competition Day", April 2019, <https://wayback.archive-it.org/12090/20191129202059/https://ec.europa.eu/commission/commissioners/2014-2019/vestager/announcements/defending-competition-digitised-world_en>.

**[BCP95]** Alvestrand, H., "A Mission Statement for the IETF", BCP 95, RFC 3935, October 2004. <https://www.rfc-editor.org/info/bcp95>

**[HTTP]** Fielding, R. T., Nottingham, M., and J. Reschke, "HTTP Semantics", Work in Progress, Internet-Draft, draft-ietf-httpbis-semantics-19, 12 September 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-semantics-19>.

**[I-D.pauly-dprive-oblivious-doh]** Kinnear, E., McManus, P., Pauly, T., Verma, T., and C. A. Wood, "Oblivious DNS Over HTTPS", Work in Progress, Internet-Draft, draft-pauly-dprive-oblivious-doh-08, 3 December 2021, <https://datatracker.ietf.org/doc/html/draft-pauly-dprive-oblivious-doh-08>.

**[I-D.thomson-tmi]** Thomson, M., "Principles for the Involvement of Intermediaries in Internet Protocols", Work in Progress, Internet-Draft, draft-thomson-tmi-02, 6 July 2021, <https://datatracker.ietf.org/doc/html/draft-thomson-tmi-02>.

**[INTERMEDIARY-INFLUENCE]** Judge, K., "Intermediary Influence", 2014, <https://scholarship.law.columbia.edu/faculty_scholarship/1856>.

**[LEGITIMACY-MULTI]** Palladino, N. and N. Santaniello, "Legitimacy, Power, and Inequalities in the Multistakeholder Internet Governance", 2020.

**[MOXIE]** Marlinspike, M., "Reflections: The ecosystem is moving", May 2016, <https://signal.org/blog/the-ecosystem-is-moving/>.

**[NEW-CHICAGO]** Lessig, L., "The New Chicago School", June 1998.

**[PIR]** Olumofin, F. and I. Goldberg, "Revisiting the Computational Practicality of Private Information Retrieval", 2010.

**[POLYCENTRIC]** Aligia, P.D. and V. Tarko, "Polycentricity: From Polanyi to Ostrom, and Beyond", April 2012.

**[RFC1035]** Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/rfc/rfc1035>.

**[RFC5321]** Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <https://www.rfc-editor.org/rfc/rfc5321>.

**[RFC6120]** Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, DOI 10.17487/RFC6120, March 2011, <https://www.rfc-editor.org/rfc/rfc6120>.

**[RFC7258]** Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <https://www.rfc-editor.org/rfc/rfc7258>.

**[RFC8799]** Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <https://www.rfc-editor.org/rfc/rfc8799>.

**[RFC8890]** Nottingham, M., "The Internet is for End Users", RFC 8890, DOI 10.17487/RFC8890, August 2020, <https://www.rfc-editor.org/rfc/rfc8890>.

**[SCALE-FREE]** Albert, R., "Emergence of Scaling in Random Networks", October 1999, <https://barabasi.com/f/67.pdf>.

**[SOAP]** Mitra, N. and Y. Lafon, "SOAP Version 1.2 Part 0: Primer (Second Edition)", World Wide Web Consortium Recommendation REC-soap12-part0-20070427, 27 April 2007, <https://www.w3.org/TR/2007/REC-soap12-part0-20070427>.

**[W3C.CR-activitystreams-core-20161215]** Snell, J. and E. Prodromou, "Activity Streams 2.0", World Wide Web Consortium CR CR-activitystreams-core-20161215, 15 December 2016, <https://www.w3.org/TR/2016/CR-activitystreams-core-20161215>.

**[WEAPONIZED-INTERDEPENDENCE]** Farrell, H. and A.L. Newman, "Weaponized Interdependence: How Global Economic Networks Shape State Coercion", 2019, <https://doi.org/10.1162/ISEC_a_00351>.

## Appendix A. Acknowledgements

This document benefits from discussions with Brian Trammell during our shared time on the Internet Architecture Board.

## Author's Address

Mark Nottingham
Prahran
Australia

Email: mnot@mnot.net
URI: https://www.mnot.net/