

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 28 April 2022

J. Arkko  
Ericsson  
T. Hardie  
Cisco  
T. Pauly  
Apple  
M. Kühlewind  
Ericsson  
25 October 2021

Considerations on Application - Network Collaboration Using Path Signals  
draft-arkko-iab-path-signals-collaboration-01

Abstract

Encryption and other security mechanisms are on the rise on all layers of the stack, protecting user data and making network operations more secured. Further, encryption is also a tool to address ossification that has been observed over time. Separation of functions into layers and enforcement of layer boundaries based on encryption supports selected exposure to those entities that are addressed by a function on a certain layer. A clear separation supports innovation and also enables new opportunities for collaborative functions. RFC 8558 describes path signals as messages to or from on-path elements. This document states principles for designing mechanisms that use or provide path signals and calls for actions on specific valuable cases.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 28 April 2022.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Past Guidance . . . . .	4
3. Principles . . . . .	5
3.1. Intentional Distribution . . . . .	6
3.2. Minimum Set of Entities . . . . .	7
3.3. Consent of Parties . . . . .	7
3.4. Minimum Information . . . . .	8
3.5. Carrying Information . . . . .	9
3.6. Protecting Information and Authentication . . . . .	9
4. Further Work . . . . .	10
5. Acknowledgments . . . . .	11
6. Informative References . . . . .	11
Authors' Addresses . . . . .	14

## 1. Introduction

Encryption, besides its important role in security in general, provides a tool to control information access and protects against ossification by avoiding unintended dependencies and requiring active maintenance. The increased deployment of encryption provides an opportunity to reconsider parts of Internet architecture that have rather used implicit derivation of input signals for on-path functions than explicit signaling, as recommended by RFC 8558 [RFC8558].

RFC 8558 defines the term path signals as signals to or from on-path elements. Today path signals are often implicit, e.g. derived from in-clear end-to-end information by e.g. examining transport protocols. For instance, on-path elements use various fields of the TCP header [RFC0793] to derive information about end-to-end latency as well as congestion. These techniques have evolved because the information was simply available and use of this information is

easier and therefore also cheaper than any explicit and potentially complex cooperative approach.

As such, applications and networks have evolved their interaction without comprehensive design for how this interaction should happen or which information would be desired for a certain function. This has lead to a situation where sometimes information is used that maybe incomplete, incorrect, or only indirectly representative of the information that was actually desired. In addition, dependencies on information and mechanisms that were designed for a different function limits the evolvability of the protocols in question.

The unplanned interaction ends up having several negative effects:

- \* Ossifying protocols by introducing unintended parties that may not be updating
- \* Creating systemic incentives against deploying more secure or private versions of protocols
- \* Basing network behaviour on information that may be incomplete or incorrect
- \* Creating a model where network entities expect to be able to use rich information about sessions passing through

For instance, features such as DNS resolution or TLS setup have been used beyond their original intent, such as in name-based filtering. MAC addresses have been used for access control, captive portal implementations that employ taking over cleartext HTTP sessions, and so on.

Increased deployment of encryption can and will change this situation. For instance, QUIC replaces TCP for various application and protects all end-to-end signals to only be accessible by the endpoint, ensuring evolvability [RFC9000]. QUIC does expose information dedicated for on-path elements to consume by design explicit signal for specific use cases, such as the Spin bit for latency measurements or connection ID that can be used by load balancers [I-D.ietf-quic-manageability] but information is limited to only those use cases. Each new use cases requires additional action.

Explicit signals that are specifically designed for the use of on-path function leave all other information is appropriately protected. This enables an architecturally clean approach and evolvability, while allowing an information exchange that is important for improving the quality of experience for users and efficient management of the network infrastructure built for them.

This draft discusses different approaches for explicit collaboration and provides guidance on architectural principles to design new mechanisms. Section 2 discusses past guidance. Section 3 discusses principles that good design can follow. This section also provides some examples and explanation of situations that not following the principles can lead to. Section 4 points to topics that need more to be looked at more carefully before any guidance can be given.

## 2. Past Guidance

Incentives are a well understood problem in general but perhaps not fully internalised for various designs attempting to establish collaboration between applications and path elements. The principle is that both receiver and sender of information must acquire tangible and immediate benefits from the communication, such as improved performance.

A related issue is understanding whether a business model or ecosystem change is needed. For instance, relative prioritization between different flows of a user or an application does not require agreements or payments. But requesting prioritization over other people's traffic may imply that you have to pay for that which may not be easy even for a single provider let alone across many.

But on to more technical aspects.

The main guidance in [RFC8558] is to be aware that implicit signals will be used whether intended or not. Protocol designers should consider either hiding these signals when the information should not be visible, or using explicit signals when it should be.

[RFC9049] discusses many past failure cases, a catalogue of past issues to avoid. It also provides relevant guidelines for new work, from discussion of incentives to more specific observations, such as the need for outperforming end-to-end mechanisms (Section 4.4), considering the need for per-connection state (Section 4.6), taking into account the latency involved in reacting to distant signals, and so on.

There are also more general guidance documents, e.g., [RFC5218] discusses protocol successes and failures, and provides general advice on incremental deployability etc. Internet Technology Adoption and Transition (ITAT) workshop report [RFC7305] is also recommended reading on this same general topic. And [RFC6709] discusses protocol extensibility, and provides general advice on the importance of global interoperability and so on.

### 3. Principles

This section attempts to provide some architecture-level principles that would help future designers and recommend useful models to apply.

A large number of our protocol mechanisms today fall into one of two categories: authenticated and private communication that is only visible to the end-to-end nodes; and unauthenticated public communication that is visible to all nodes on a path. RFC 8558 explores the line between data that is protected and path signals.

There is a danger in taking a position that is too extreme towards either exposing all information to the path, or hiding all information from the path.

Exposed information encourages pervasive monitoring, which is described in RFC 7258 [RFC7258]. Exposed information may also be used for commercial purposes, or form a basis for filtering that the applications or users do not desire.

But a lack of all path signaling, on the other hand, may be harmful to network management, debugging, or the ability for networks to provide the most efficient services. There are many cases where elements on the network path can provide beneficial services, but only if they can coordinate with the endpoints. It also affects the ability of service providers and others observe why problems occur [RFC9075].

This situation is sometimes cast as an adversarial tradeoff between privacy and the ability for the network path to provide intended functions. However, this is perhaps an unnecessarily polarized characterization as a zero-sum situation. Not all information passing implies loss of privacy. For instance, performance information or preferences do not require disclosing user or application identity or what content is being accessed, network congestion status information does not have reveal network topology or the status of other users, and so on.

This points to one way to resolve the adversity: the careful of design of what information is passed.

Another approach is to employ explicit trust and coordination between endpoints and network devices. VPNs are a good example of a case where there is an explicit authentication and negotiation with a network path element that's used to optimize behavior or gain access to specific resources.

The goal of improving privacy and trust on the Internet does not necessarily need to remove the ability for network elements to perform beneficial functions. We should instead improve the way that these functions are achieved. Our goals should be:

- \* To ensure that information is distributed intentionally, not accidentally;
- \* to understand the privacy and other implications of any distributed information;
- \* to ensure that the information distribution targets the intended parties; and
- \* to gate the distribution of information on the consent of the relevant parties

These goals for distribution apply equally to senders, receivers, and path elements.

We can establish some basic questions that any new network path functions should consider:

- \* What is the minimum set of entities that need to be involved?
- \* What is the minimum information each entity in this set needs?
- \* Which entities must consent to the information exchange?

If we look at many of the ways network path functions are achieved today, we find that many if not most of them fall short the standard set up by the questions above. Too often, they send unnecessary information or fail to limit the scope of distribution or providing any negotiation or consent.

Going forward, new standards work in the IETF needs to focus on addressing this gap by providing better alternatives and mechanisms for providing path functions. Note that not all of these functions can be achieved in a way that preserves a high level of user privacy from the network; in such cases, it is incumbent upon us to not ignore the use case, but instead to define the high bar for consent and trust, and thus define a limited applicability for those functions.

### 3.1. Intentional Distribution

This guideline is best expressed in RFC 8558:

"Fundamentally, this document recommends that implicit signals should be avoided and that an implicit signal should be replaced with an explicit signal only when the signal's originator intends that it be used by the network elements on the path. For many flows, this may result in the signal being absent but allows it to be present when needed."

This guideline applies also in the other direction as well. For instance, a network element should not unintentionally leak information that is visible to endpoints. An explicit decision is needed for a specific information to be provided, along with analysis of the security and privacy implications of that information.

### 3.2. Minimum Set of Entities

It is recommended that a design identify the minimum number of entities needed to share a specific signal required for an identified function. In some cases this will be a very limited set, e.g. when the application needs to provide a signal to a specific gateway function. In other cases, such as congestion control, a signal might be shared with every router along the path, since each should be aware of the congestion.

While it is tempting to consider removing these limitations in the context of closed, private networks, each interaction is still best considered separately, rather than simply allowing all information exchanges within the closed network. Even in a closed network with carefully managed components there may be compromised components, as evidenced in the most extreme way by the Stuxnet worm that operated in an airgapped network. Most "closed" networks have at least some needs and means to access the rest of the Internet, and should not be modeled as if they had an impenetrable security barrier.

### 3.3. Consent of Parties

Consent and trust must determine the distribution of information. The set of entities that need to consent is determined by the scope and specificity of the information being shared.

Three distinct types of consent are recommended for collaboration or information sharing:

- \* A corollary of the intentional distribution is that the sender needs to agree to sending the information. Or that the requester for an action needs to agree to make a request; it should not be an implicit decision by the receiver that information was sent or a request was made, just because a packet happened to be formed in a particular way.

- \* At the same time, the recipient of information or the target of a request should agree to wishing to receive the information. It should not be burdened with extra processing if it does not have willingness or a need to do so. This happens naturally in most protocol designs, but has been a problem for some cases where "slow path" packet processing is required or implied, and the recipient or router did not have willingness for this.
- \* Internet communications are not made for the applications, they are ultimately made on behalf of users. Information relating to the users is something that both networks and applications should be careful with, and not be shared without the user's consent. This is not always easy, as the interests of the user and (for instance) application developer may not always coincide; some applications may wish to collect more information about the user than the user would like.

As a result, typically an application's consent is not the same as the user's consent.

#### 3.4. Minimum Information

Parties should provide only the information that is needed for the other party to perform the collaboration task that is desired by this party, and not more. This applies to information sent by an application about itself, information sent about users, or information sent by the network.

An architecture can follow the guideline from RFC 8558 in using explicit signals, but still fail to differentiate properly between information that should be kept private and information that should be shared.

In looking at what information can or cannot easily be passed, we can look at both information from the network to the application, and from the application to the network.

For the application to the network direction, user-identifying information can be problematic for privacy and tracking reasons. Similarly, application identity can be problematic, if it might form the basis for prioritization or discrimination that the application provider may not wish to happen. It may also have undesirable economic consequences, such as extra charges for the consumer from a priority service where a regular service would have worked.

On the other hand, as noted above, information about general classes of applications may be desirable to be given by application



providers, if it enables prioritization that would improve service, e.g., differentiation between interactive and non-interactive services.

For the network to application direction there is similarly sensitive information, such as the precise location of the user. On the other hand, various generic network conditions, predictive bandwidth and latency capabilities, and so on might be attractive information that applications can use to determine, for instance, optimal strategies for changing codecs. However, information given by the network about load conditions and so on should not form a mechanism to provide a side-channel into what other users are doing.

While information needs to be specific and provided on a per-need basis, it is often beneficial to provide declarative information that, for instance, expresses application needs than makes specific requests for action.

### 3.5. Carrying Information

There is a distinction between what information is passed and how it is carried. The actually sent information may be limited, while the mechanisms for sending or requesting information can be capable of sending much more.

There is a tradeoff here between flexibility and ensuring the minimality of information in the future. The concern is that a fully generic data sharing approach between different layers and parties could potentially be misused, e.g., by making the availability of some information a requirement for passing through a network.

This is undesirable, and our recommendation is to employ very targeted, minimal information carriage mechanisms.

### 3.6. Protecting Information and Authentication

Some simple forms of information often exist in cleartext form, e.g., ECN bits from routers are generally not authenticated or integrity protected. This is possible when the information exchanges are advisory in their nature, and do not carry any significantly sensitive information from the parties.

In other cases it may be necessary to establish a secure channel for communication with a specific other party, e.g., between a network element and an application. This channel may need to be authenticated, integrity protected and encrypted. This is necessary, for instance, if the particular information or request needs to be shared in confidentiality only with a particular, trusted node, or there's

a danger of an attack where someone else may forge messages that could endanger the communication.

However, it is important to note that authentication does not equal trust. Whether a communication is with an application server or network element that can be shown to be associated with a particular domain name, it does not follow that information about the user can be safely sent to it.

In some cases, the ability of a party to show that it is on the path can be beneficial. For instance, an ICMP error that refers to a valid flow may be more trustworthy than any ICMP error claiming to come from an address.

Other cases may require more substantial assurances. For instance, a specific trust arrangement may be established between a particular network and application. Or technologies such as confidential computing can be applied to provide an assurance that information processed by a party is handled in an appropriate manner.

#### 4. Further Work

This is a developing field, and it is expected that our understanding continues to grow. The recent changes with regards to much higher use of encryption at different protocol layers, the consolidation or more and more traffic to the same destinations, and so on have also greatly impacted the field.

While there are some examples of modern, well-designed collaboration mechanisms, clearly more work is needed. Many complex cases would require significant developments in order to become feasible.

Some of the most difficult areas are listed below. Research on these topics would be welcome.

- \* Business arrangements. Many designs - for instance those related to quality-of-service - involve an expectation of paying for a service. This is possible and has been successful within individual domains, e.g., users can pay for higher data rates or data caps in their ISP networks. However, it is a business-wise much harder proposition for end-to-end connections across multiple administrative domains [Claffy2015] [RFC9049].
- \* Secure communications with path elements. This has been a difficult topic, both from the mechanics and scalability point view, but also because there is no easy way to find out which parties to trust or what trust roots would be appropriate. Some application-network element interaction designs have focused on

information (such as ECN bits) that is distributed openly within a path, but there are limited examples of designs with secure information exchange with specific nodes.

- \* The use of path signals for reducing the effects of denial-of-service attacks, e.g., in the form of modern "source quench" designs.
- \* Ways of protecting information when held by network elements or servers, beyond communications security. For instance, host applications commonly share sensitive information about the user's actions with other nodes, starting from basic data such as domain names learned by DNS infrastructure or source and destination addresses and protocol header information learned by all routers on the path, to detailed end user identity and other information learned by the servers. Some solutions are starting to exist for this but are not widely deployed, at least not today [Oblivious] [PDoT] [I-D.arkko-dns-confidential] [I-D.thomson-http-oblivious]. These solutions address also very specific parts of the issue, and more work remains.
- \* Sharing information from networks to applications. Some proposals have been made in this space (see, e.g., [I-D.flinck-mobile-throughput-guidance]) but there are no successful or deployed mechanisms today.

## 5. Acknowledgments

The authors would like to thank everyone at the IETF, the IAB, and our day jobs for interesting thoughts and proposals in this space. Fragments of this document were also in [I-D.per-app-networking-considerations] and [I-D.arkko-path-signals-information] that were published earlier. We would also like to acknowledge [I-D.trammell-stackevo-explicit-coop] for presenting similar thoughts. Finally, the authors would like to thank Adrian Farrell, Toerless Eckert, and Jeffrey Haas for useful feedback in the IABOPEN session at IETF-111.

## 6. Informative References

[Claffy2015]

kc Claffy, . and D. Clark, "Adding Enhanced Services to the Internet: Lessons from History", TPRC 43: The 43rd Research Conference on Communication, Information and Internet Policy Paper , April 2015.

[I-D.arkko-dns-confidential]

Arkko, J. and J. Novotny, "Privacy Improvements for DNS

Resolution with Confidential Computing", Work in Progress, Internet-Draft, draft-arkko-dns-confidential-02, 2 July 2021, <<https://www.ietf.org/archive/id/draft-arkko-dns-confidential-02.txt>>.

[I-D.arkko-path-signals-information]

Arkko, J., "Considerations on Information Passed between Networks and Applications", Work in Progress, Internet-Draft, draft-arkko-path-signals-information-00, 22 February 2021, <<https://www.ietf.org/archive/id/draft-arkko-path-signals-information-00.txt>>.

[I-D.flinck-mobile-throughput-guidance]

Jain, A., Terzis, A., Flinck, H., Sprecher, N., Arunachalam, S., Smith, K., Devarapalli, V., and R. B. Yanai, "Mobile Throughput Guidance Inband Signaling Protocol", Work in Progress, Internet-Draft, draft-flinck-mobile-throughput-guidance-04, 13 March 2017, <<https://www.ietf.org/archive/id/draft-flinck-mobile-throughput-guidance-04.txt>>.

[I-D.ietf-quic-manageability]

Kuehlewind, M. and B. Trammell, "Manageability of the QUIC Transport Protocol", Work in Progress, Internet-Draft, draft-ietf-quic-manageability-13, 2 September 2021, <<https://www.ietf.org/archive/id/draft-ietf-quic-manageability-13.txt>>.

[I-D.per-app-networking-considerations]

Colitti, L. and T. Pauly, "Per-Application Networking Considerations", Work in Progress, Internet-Draft, draft-per-app-networking-considerations-00, 15 November 2020, <<https://www.ietf.org/archive/id/draft-per-app-networking-considerations-00.txt>>.

[I-D.thomson-http-oblivious]

Thomson, M. and C. A. Wood, "Oblivious HTTP", Work in Progress, Internet-Draft, draft-thomson-http-oblivious-02, 24 August 2021, <<https://www.ietf.org/archive/id/draft-thomson-http-oblivious-02.txt>>.

[I-D.trammell-stackevo-explicit-coop]

Trammell, B., "Architectural Considerations for Transport Evolution with Explicit Path Cooperation", Work in Progress, Internet-Draft, draft-trammell-stackevo-explicit-coop-00, 23 September 2015, <<https://www.ietf.org/archive/id/draft-trammell-stackevo-explicit-coop-00.txt>>.

- [Oblivious] Schmitt, P., "Oblivious DNS: Practical privacy for DNS queries", Proceedings on Privacy Enhancing Technologies 2019.2: 228-244 , 2019.
- [PDoT] Nakatsuka, Y., Paverd, A., and G. Tsudik, "PDoT: Private DNS-over-TLS with TEE Support", Digit. Threat.: Res. Pract., Vol. 2, No. 1, Article 3, <https://dl.acm.org/doi/fullHtml/10.1145/3431171> , February 2021.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.
- [RFC5218] Thaler, D. and B. Aboba, "What Makes for a Successful Protocol?", RFC 5218, DOI 10.17487/RFC5218, July 2008, <<https://www.rfc-editor.org/info/rfc5218>>.
- [RFC6709] Carpenter, B., Aboba, B., Ed., and S. Cheshire, "Design Considerations for Protocol Extensions", RFC 6709, DOI 10.17487/RFC6709, September 2012, <<https://www.rfc-editor.org/info/rfc6709>>.
- [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/info/rfc7258>>.
- [RFC7305] Lear, E., Ed., "Report from the IAB Workshop on Internet Technology Adoption and Transition (ITAT)", RFC 7305, DOI 10.17487/RFC7305, July 2014, <<https://www.rfc-editor.org/info/rfc7305>>.
- [RFC8558] Hardie, T., Ed., "Transport Protocol Path Signals", RFC 8558, DOI 10.17487/RFC8558, April 2019, <<https://www.rfc-editor.org/info/rfc8558>>.
- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/info/rfc9000>>.
- [RFC9049] Dawkins, S., Ed., "Path Aware Networking: Obstacles to Deployment (A Bestiary of Roads Not Taken)", RFC 9049, DOI 10.17487/RFC9049, June 2021, <<https://www.rfc-editor.org/info/rfc9049>>.

[RFC9075] Arkko, J., Farrell, S., Kühlewind, M., and C. Perkins,  
"Report from the IAB COVID-19 Network Impacts Workshop  
2020", RFC 9075, DOI 10.17487/RFC9075, July 2021,  
<<https://www.rfc-editor.org/info/rfc9075>>.

Authors' Addresses

Jari Arkko  
Ericsson

Email: [jari.arkko@ericsson.com](mailto:jari.arkko@ericsson.com)

Ted Hardie  
Cisco

Email: [ted.ietf@gmail.com](mailto:ted.ietf@gmail.com)

Tommy Pauly  
Apple

Email: [tpauly@apple.com](mailto:tpauly@apple.com)

Mirja Kühlewind  
Ericsson

Email: [mirja.kuehlewind@ericsson.com](mailto:mirja.kuehlewind@ericsson.com)

edm  
Internet-Draft  
Intended status: Informational  
Expires: 23 July 2022

C. Eckel  
Cisco Systems  
19 January 2022

Find Code Related to an Internet-Draft or RFC  
draft-eckel-edm-find-code-01

## Abstract

Code related to existing IETF standards and ongoing standardization efforts may exist and be publicly accessible in many places. This document provides a set of practices to make it easier to identify and to find such code.

## Discussion Venues

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

Discussion of this document takes place on the Evolvability, Deployability, & Maintainability mailing list ([edm@iab.org](mailto:edm@iab.org)), which is archived at <https://mailarchive.ietf.org/arch/browse/edm/>.

Source for this draft and an issue tracker can be found at <https://github.com/eckelcu/draft-eckel-edm-find-code>.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 23 July 2022.

## Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Existing IETF Processes and Procedures . . . . .	3
2.1. Implementation Status . . . . .	3
2.2. GitHub . . . . .	3
2.3. Hackathon . . . . .	4
3. Proposal . . . . .	4
3.1. GitHub Repository . . . . .	4
3.2. README . . . . .	5
3.3. Datatracker . . . . .	5
3.4. Implementation Status . . . . .	5
3.5. Inline Errata . . . . .	6
4. Implementation Status . . . . .	6
5. Security Considerations . . . . .	6
6. IANA Considerations . . . . .	6
7. Informative References . . . . .	6
Acknowledgments . . . . .	7
Author's Address . . . . .	7

## 1. Introduction

Code related to existing IETF standards and ongoing standardization efforts may exist and be publicly accessible in many places. One common place is GitHub (<https://github.com/>), but there are many others. The relationship of the code to corresponding IETF standards efforts may be direct, as in the case of a client or server that supports protocol defined by an Internet-Draft (I-D) (<https://www.ietf.org/standards/ids/>). It may be indirect, as in a utility that helps analyze network traffic corresponding to this same protocol. The maturity and status of the code may vary considerably, including something written quickly as a proof of concept during a hackathon, a well established and supported implementation, or a legacy project no longer actively developed or maintained. The code must be publicly available, and preferably open source, though other terms of use may exist as well. In all cases, the code is potentially of interest and beneficial to people contributing to the definition, implementation, or deployment of an existing or evolving IETF standard. This document provides a set of practices make it



easier to identify and to find such code.

## 2. Existing IETF Processes and Procedures

The idea that code related to IETF standards is valuable is not new. Most IETF participants are familiar with the phrase "rough consensus and running code" from the IETF Tao (<https://www.ietf.org/tao.html>). The existence of multiple independently developed and interoperable implementations was explicitly required by [RFC1264] for internet standards on routing protocols. Subsequent updates relaxed this requirement, but the value of running code is still appreciated, and several current RFCs define processes and procedures related to running code.

### 2.1. Implementation Status

A simple process that allows authors of I-Ds to record the status of known implementations by including an Implementation Status section is defined [RFC7942]. The goal of this section is to allow the reader to assign due consideration to I-Ds that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that make the protocols and corresponding documents more mature. However, it is stated that the Implementation Status section should be removed from I-Ds before they are published as RFCs. As a result, the value of the code is limited to that required to develop the standard, and the mechanism does not help find the code once the RFC is published.

### 2.2. GitHub

The IETF chartered the GitHub Integration and Tooling (GIT) (<https://datatracker.ietf.org/wg/git/about/>) working group to establish and document practices and policies for use of GitHub by working groups for managing their work. This resulted in [RFC8874], which provides a set of guidelines for working groups that choose to use GitHub for their work, and [RFC8875], which specifies a set of administrative processes and conventions for such working groups. Within the working group, the concept of work is limited to the development of I-Ds that may eventually become RFCs. Any concept of code is limited to that which appears as text within these documents. In many cases, there is additional code that is closely associated with the documents but not contained within them. This code may be of interest to the community of people contributing to the development of the documents or to the implementation or deployment of eventual standards defined by them.

### 2.3. Hackathon

The IETF Hackathon [I-D.ietf-shmoo-hackathon] encourages the IETF community to collaborate on running code related to existing and evolving Internet standards. Each Hackathon has a wiki that provides a brief description of each project. It is common for there to be one or more I-Ds or RFCs associated with each project, and for there to be one or more related code repositories. These resources are often listed on the wiki, but they are documented and shared with project teams in other ways as well. After the Hackathon, the wiki remains available, but the information within it is typically not updated or maintained.

## 3. Proposal

This section specifies a set of practices that use existing mechanisms to associate code with an I-D or RFC. Following these practices makes it easier for others working with the I-D or RFC to find such code.

### 3.1. GitHub Repository

A GitHub repository (<https://docs.github.com/en/github/getting-started-with-github/quickstart/create-a-repo#create-a-repository>) should be setup for an I-D as outlined in Section 3.2 of RFC 8874 (<https://www.rfc-editor.org/rfc/rfc8874.html#section-3.2>). The `i-d-template` (<https://github.com/martinthomson/i-d-template>) can be used to get started. It provides useful features, including integration with the Datatracker (see Section 3.3). The resulting repository should be associated with the I-D using the Datatracker `github_repo` tag. This should be done even if GitHub is not to be used to collaborate on the I-D.

A GitHub repository typically exists within a GitHub organization (<https://docs.github.com/en/organizations/collaborating-with-groups-in-organizations/about-organizations>). This is not always the case (e.g., a repository in a personal GitHub account), and even when it is, the GitHub organization may not be appropriate to associated with the I-D. In the event there is an appropriate GitHub organization, it should be associated with the I-D using the Datatracker `github_org` tag.

### 3.2. README

The GitHub repository associated with the I-D should include a README (<https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/creating-a-repository-on-github/about-readmes>). The README should include information about the repository, whether or not it is being used to collaborate on the I-D, and any code associated with the I-D. The latter may be achieved by including direct links to such code or by including links to other resources that include information about such code. These resources may be a file, folder, or wiki (<https://docs.github.com/en/communities/>) within the GitHub repository or the GitHub organization associated with the I-D. The QUIC WG's Implementations wiki (<https://github.com/quicwg/base-drafts/wiki/Implementations>) is an example.

### 3.3. Datatracker

The IETF Datatracker (<https://datatracker.ietf.org/>) supports the association of Additional Resources with a Document (e.g., an I-D or RFC) or a Group (e.g., working group (<https://datatracker.ietf.org/wg/>), research group (<https://datatracker.ietf.org/rg/>)). An Additional Resource can be, among others things, a GitHub organization] or a GitHub repository].

It is recommended that this Datatracker mechanism be used to associate an appropriate GitHub organization and repository with an I-D. Ideally these are setup per the guidelines in [RFC8874] and [RFC8875]. In the event the working group or research group is not using GitHub, or the I-D has not yet been adopted by the group, another GitHub organization or repository may be used instead. A GitHub organization is associated with the I-D using the `github_org` tag. A GitHub repository is associated with the I-D using the `github_repo` tag.

### 3.4. Implementation Status

An Implementation Status section, as defined [RFC7942], should be added to an I-D. It should include any GitHub organization or GitHub repository associated with the I-D.

### 3.5. Inline Errata

In the event an I-D becomes an RFC, people looking for code are less likely to reference the Datatracker, and the Implementation Status section may have been removed or require updates. Any GitHub organization or GitHub repository associated with the RFC should be made available as inline errata (<https://mailarchive.ietf.org/arch/msg/edm/ku3cd5xTla7tbtohVYWWW7-XTIg/>). An example of this is RFC 3261 with inline errata (<https://www.rfc-editor.org/rfc/inline-errata/rfc3261.html>). Inline errata views for v3 era RFCs are not supported at this time.

### 4. Implementation Status

The practices proposed in this document are followed by draft-ietf-shmoo-hackathon (<https://datatracker.ietf.org/doc/draft-ietf-shmoo-hackathon/>).

### 5. Security Considerations

TBD.

### 6. IANA Considerations

This document has no IANA actions.

### 7. Informative References

[I-D.ietf-shmoo-hackathon]

Eckel, C., "Running an IETF Hackathon", Work in Progress, Internet-Draft, draft-ietf-shmoo-hackathon-04, 19 January 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-shmoo-hackathon-04>>.

[RFC1264] Hinden, R., "Internet Engineering Task Force Internet Routing Protocol Standardization Criteria", RFC 1264, DOI 10.17487/RFC1264, October 1991, <<https://www.rfc-editor.org/rfc/rfc1264>>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/rfc/rfc7942>>.

[RFC8874] Thomson, M. and B. Stark, "Working Group GitHub Usage Guidance", RFC 8874, DOI 10.17487/RFC8874, August 2020, <<https://www.rfc-editor.org/rfc/rfc8874>>.

[RFC8875] Cooper, A. and P. Hoffman, "Working Group GitHub Administration", RFC 8875, DOI 10.17487/RFC8875, August 2020, <<https://www.rfc-editor.org/rfc/rfc8875>>.

#### Acknowledgments

Vijay Gurbani started (<https://mailarchive.ietf.org/arch/msg/edm/1AV0yGy5cetLjmP6aOu0xyD2kHE/>) the discussion that inspired this effort.

Robert Sparks highlighted a datatracker mechanism ([https://mailarchive.ietf.org/arch/msg/wgchairs/DA-fWpq\\_nsy\\_5kPhJEheBlyaaqI/](https://mailarchive.ietf.org/arch/msg/wgchairs/DA-fWpq_nsy_5kPhJEheBlyaaqI/)) to add a reference to a GitHub repository or organization using the github\_repo or github\_org tag, respectively.

Martin Thompson created the i-d-template (<https://github.com/martinthomson/i-d-template>) repository can be used to setup a GitHub repository for an I-D.

Spencer Dawkins pointed out the RFC editor's ability to inline errata (<https://mailarchive.ietf.org/arch/msg/edm/ku3cd5xTla7tbtohVYWWW7-XTIg/>) and noted that something similar could be done to point to code.

Adam Roach played an important role in enabling the RFC editor's ability to inline errata (<https://mailarchive.ietf.org/arch/msg/edm/ku3cd5xTla7tbtohVYWWW7-XTIg/>).

Mark Nottingham provided an illustrative examples of how the QUIC (<https://github.com/quicwg/base-drafts/wiki/Implementations>) working group uses wiki pages to track early implementations.

Many other people shared thoughts on the email lists for WG Chairs (<https://mailarchive.ietf.org/arch/browse/wgchairs/>) and EDM (<https://mailarchive.ietf.org/arch/browse/edm/>) about how to make it easier to find code. These helped shape the practices outlined in this document.

#### Author's Address

Charles Eckel  
Cisco Systems  
United States of America

Email: [eckelcu@cisco.com](mailto:eckelcu@cisco.com)

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 12 November 2022

M. Thomson  
Mozilla  
D. Schinazi  
Google LLC  
11 May 2022

The Harmful Consequences of the Robustness Principle  
draft-iab-protocol-maintenance-06

Abstract

The robustness principle, often phrased as "be conservative in what you send, and liberal in what you accept", has long guided the design and implementation of Internet protocols. The posture this statement advocates promotes interoperability in the short term, but can negatively affect the protocol ecosystem over time. For a protocol that is actively maintained, the robustness principle can, and should, be avoided.

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-iab-protocol-maintenance/>.

Discussion of this document takes place on the EDM IAB Program mailing list (<mailto:edm@iab.org>), which is archived at <https://www.iab.org/mailman/listinfo/edm>.

Source for this draft and an issue tracker can be found at <https://github.com/intarchboard/draft-protocol-maintenance>.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 12 November 2022.

#### Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

#### Table of Contents

1. Introduction . . . . .	2
2. Fallibility of Specifications . . . . .	3
3. Protocol Decay . . . . .	4
4. Ecosystem Effects . . . . .	5
5. Active Protocol Maintenance . . . . .	7
6. Extensibility . . . . .	8
7. Virtuous Intolerance . . . . .	9
8. Exclusion . . . . .	10
9. Security Considerations . . . . .	11
10. IANA Considerations . . . . .	11
11. Informative References . . . . .	11
Acknowledgments . . . . .	13
Authors' Addresses . . . . .	13

#### 1. Introduction

The robustness principle has been hugely influential in shaping the design of the Internet. As stated in the IAB document on Architectural Principles of the Internet [RFC1958], the robustness principle advises to:

Be strict when sending and tolerant when receiving.  
Implementations must follow specifications precisely when sending to the network, and tolerate faulty input from the network. When in doubt, discard faulty input silently, without returning an error message unless this is required by the specification.

This simple statement captures a significant concept in the design of interoperable systems. Many consider the application of the robustness principle to be instrumental in the success of the Internet as well as the design of interoperable protocols in general.

Time and experience shows that negative consequences to interoperability accumulate over time if implementations apply the robustness principle. This problem originates from an assumption implicit in the principle that it is not possible to affect change in a system the size of the Internet. That is, the idea that once a protocol specification is published, changes that might require existing implementations to change are not feasible.

Many problems that might lead to applications of the robustness principle are avoided for protocols under active maintenance. Active protocol maintenance is where a community of protocol designers, implementers, and deployers work together to continuously improve and evolve protocol specifications alongside implementations and deployments of those protocols. A community that takes an active role in the maintenance of protocols will no longer need to rely on the robustness principle to avoid interoperability issues.

There is good evidence to suggest that many important protocols are routinely maintained beyond their inception. In particular, a sizeable proportion of IETF activity is dedicated to the stewardship of existing protocols. This document serves primarily as a record of the hazards inherent in applying the robustness principle and to offer an alternative strategy for handling interoperability problems in deployments.

Ideally, protocol implementations never have to apply the robustness principle. Or, where it is unavoidable, use of the robustness principle is viewed as a short term workaround that needs to be quickly reverted.

## 2. Fallibility of Specifications

The context from which the robustness principle was developed provides valuable insights into its intent and purpose. The earliest form of the principle in the RFC series (the Internet Protocol specification [RFC0760]) is preceded by a sentence that reveals the motivation for the principle:

While the goal of this specification is to be explicit about the protocol there is the possibility of differing interpretations. In general, an implementation should be conservative in its sending behavior, and liberal in its receiving behavior.

This formulation of the principle expressly recognizes the possibility that the specification could be imperfect. This contextualizes the principle in an important way.



An imperfect specification is natural, largely because it is more important to proceed to implementation and deployment than it is to perfect a specification. A protocol benefits greatly from experience with its use. A deployed protocol is immeasurably more useful than a perfect protocol. The robustness principle is a tool that is suited to early phases of system design.

As demonstrated by the IAB document on Successful Protocols [RFC5218], success or failure of a protocol depends far more on factors like usefulness than on technical excellence. Timely publication of protocol specifications, even with the potential for flaws, likely contributed significantly to the eventual success of the Internet.

The problem is therefore not with the premise, but with its conclusion: the robustness principle itself.

### 3. Protocol Decay

The application of the robustness principle to the early Internet, or any system that is in early phases of deployment, is expedient. Applying the principle defers the effort of dealing with interoperability problems, which prioritizes progress. However, deferral can amplify the ultimate cost of handling interoperability problems.

Divergent implementations of a specification emerge over time. When variations occur in the interpretation or expression of semantic components, implementations cease to be perfectly interoperable.

Implementation bugs are often identified as the cause of variation, though it is often a combination of factors. Application of a protocol to uses that were not anticipated in the original design, or ambiguities and errors in the specification are often confounding factors. Disagreements on the interpretation of specifications should be expected over the lifetime of a protocol.

Even with the best intentions, the pressure to interoperate can be significant. No implementation can hope to avoid having to trade correctness for interoperability indefinitely.

An implementation that reacts to variations in the manner recommended in the robustness principle sets up a feedback cycle. Over time:

- \* Implementations progressively add logic to constrain how data is transmitted, or to permit variations in what is received.

- \* Errors in implementations or confusion about semantics are permitted or ignored.
- \* These errors can become entrenched, forcing other implementations to be tolerant of those errors.

A flaw can become entrenched as a de facto standard. Any implementation of the protocol is required to replicate the aberrant behavior, or it is not interoperable. This is both a consequence of applying the robustness principle, and a product of a natural reluctance to avoid fatal error conditions. Ensuring interoperability in this environment is often referred to as aiming to be "bug for bug compatible".

For example, in TLS [TLS], extensions use a tag-length-value format and they can be added to messages in any order. However, some server implementations terminated connections if they encountered a TLS ClientHello message that ends with an empty extension. To maintain interoperability, client implementations were required to be aware of this bug and ensure that a ClientHello message ends in a non-empty extension.

The original JSON specification [RFC4627] demonstrates the effect of specification shortcomings: it did not tightly specify some important details including Unicode handling, ordering and duplication of object members, and number encoding. Consequently, a range of interpretations were used by implementations. An updated JSON specification [RFC7159] did not correct these errors, concentrating instead on identifying the interoperable subset of JSON. I-JSON [RFC7493] takes that subset and defines a new format that prohibits the problematic parts of JSON. Of course, that means that I-JSON is not fully interoperable with JSON. Consequently, I-JSON is not widely implemented in parsers. Many JSON parsers now implement the more precise algorithm specified in [ECMA262].

The robustness principle therefore encourages a chain reaction that can create interoperability problems. In particular, the application of the robustness principle is particularly deleterious for early implementations of new protocols as quirks in early implementations can affect all subsequent deployments.

#### 4. Ecosystem Effects

From observing widely deployed protocols, it appears there are two stable points on the spectrum between being strict versus permissive in the presence of protocol errors:

- \* If implementations predominantly enforce strict compliance with specifications, newer implementations will experience failures if they do not comply with protocol requirements. Newer implementations need to fix compliance issues in order to be successfully deployed. This ensures that most deployments are compliant.
- \* Conversely, if non-compliance is tolerated by existing implementations, non-compliant implementations can be deployed successfully. Newer implementations then have strong incentive to tolerate any existing non-compliance in order to be successfully deployed. This ensures that most deployments are tolerant of the same non-compliant behavior.

This happens because interoperability requirements for protocol implementations are set by other deployments. Specifications and - where they exist - conformance test suites might guide the initial development of implementations, but implementations ultimately need to interoperate with deployed implementations.

For widely used protocols, the massive scale of the Internet makes large-scale interoperability testing infeasible for all but a privileged few. The cost of building a new implementation using reverse engineering increases as the number of implementations and bugs increases. Worse, the set of tweaks necessary for wide interoperability can be difficult to discover. In the worst case, a new implementer might have to choose between deployments that have diverged so far as to no longer be interoperable.

Consequently, new implementations might be forced into niche uses, where the problems arising from interoperability issues can be more closely managed. However, restricting new implementations into limited deployments risks causing forks in the protocol. If implementations do not interoperate, little prevents those implementations from diverging more over time.

This has a negative impact on the ecosystem of a protocol. New implementations are key to the continued viability of a protocol. New protocol implementations are also more likely to be developed for new and diverse use cases and are often the origin of features and capabilities that can be of benefit to existing users.

The need to work around interoperability problems also reduces the ability of established implementations to change. An accumulation of mitigations for interoperability issues makes implementations more difficult to maintain and can constrain extensibility (see also the IAB document on the Long-Term Viability of Protocol Extension Mechanisms [RFC9170]).

Sometimes what appear to be interoperability problems are symptomatic of issues in protocol design. A community that is willing to make changes to the protocol, by revising or extending it, makes the protocol better in the process. Applying the robustness principle instead conceals problems, making it harder, or even impossible, to fix them later.

## 5. Active Protocol Maintenance

The robustness principle can be highly effective in safeguarding against flaws in the implementation of a protocol by peers. Especially when a specification remains unchanged for an extended period of time, incentive to be tolerant of errors accumulates over time. Indeed, when faced with divergent interpretations of an immutable specification, the only way for an implementation to remain interoperable is to be tolerant of differences in interpretation and implementation errors.

From this perspective, application of the robustness principle to the implementation of a protocol specification that does not change is logical, even necessary. But that conclusion relies on an assumption that existing specifications and implementations cannot change. Applying the robustness principle in this way disproportionately values short-term gains over the negative effects on future implementations and the protocol as a whole.

For a protocol to have sustained viability, it is necessary for both specifications and implementations to be responsive to changes, in addition to handling new and old problems that might arise over time.

Maintaining specifications so that they closely match deployments ensures that implementations are consistently interoperable and removes needless barriers for new implementations. Maintenance also enables continued improvement of the protocol. New use cases are an indicator that the protocol could be successful [RFC5218].

Protocol designers are strongly encouraged to continue to maintain and evolve protocol specifications beyond their initial inception and definition. This might require the development of revised specifications, extensions, or other supporting material that documents the current state of the protocol. Involvement of those who implement and deploy the protocol is a critical part of this process, as they provide input on their experience with how the protocol is used.

Most interoperability problems do not require revision of protocols or protocol specifications. For instance, the most effective means of dealing with a defective implementation in a peer could be to

email the developer responsible. It is far more efficient in the long term to fix one isolated bug than it is to deal with the consequences of workarounds.

Early implementations of protocols have a stronger obligation to closely follow specifications as their behavior will affect all subsequent implementations. In addition to specifications, later implementations will be guided by what existing deployments accept. Tolerance of errors in early deployments is most likely to result in problems. Protocol specifications might need more frequent revision during early deployments to capture feedback from early rounds of deployment.

Neglect can quickly produce the negative consequences this document describes. Restoring the protocol to a state where it can be maintained involves first discovering the properties of the protocol as it is deployed, rather than the protocol as it was originally documented. This can be difficult and time-consuming, particularly if the protocol has a diverse set of implementations. Such a process was undertaken for HTTP [HTTP] after a period of minimal maintenance. Restoring HTTP specifications to relevance took significant effort.

Maintenance is most effective if it is responsive, which is greatly affected by how rapidly protocol changes can be deployed. For protocol deployments that operate on longer time scales, temporary workarounds following the spirit of the robustness principle might be necessary. For this, improvements in software update mechanisms ensure that the cost of reacting to changes is much lower than it was in the past. Alternatively, if specifications can be updated more readily than deployments, details of the workaround can be documented, including the desired form of the protocols once the need for workarounds no longer exists and plans for removing the workaround.

## 6. Extensibility

Good extensibility [EXT] can make it easier to respond to new use cases or changes in the environment in which the protocol is deployed.

The ability to extend a protocol is sometimes mistaken for an application of the robustness principle. After all, if one party wants to start using a new feature before another party is prepared to receive it, it might be assumed that the receiving party is being tolerant of unexpected inputs.

A well-designed extensibility mechanism establishes clear rules for the handling of things like new messages or parameters. This depends on precisely specifying the handling of malformed or illegal inputs so that implementations behave consistently in all cases that might affect interoperation. If extension mechanisms and error handling are designed and implemented correctly, new protocol features can be deployed with confidence in the understanding of the effect they have on existing implementations.

In contrast, relying on implementations to consistently apply the robustness principle is not a good strategy for extensibility. Using undocumented or accidental features of a protocol as the basis of an extensibility mechanism can be extremely difficult, as is demonstrated by the case study in Appendix A.3 of [EXT].

A protocol could be designed to permit a narrow set of valid inputs, or it could allow a wide range of inputs as a core feature (see for example [HTML]). Specifying and implementing a more flexible protocol is more difficult; allowing less variability is preferable in the absence of strong reasons to be flexible.

## 7. Virtuous Intolerance

A well-specified protocol includes rules for consistent handling of aberrant conditions. This increases the chances that implementations will have consistent and interoperable handling of unusual conditions.

Choosing to generate fatal errors for unspecified conditions instead of attempting error recovery can ensure that faults receive attention. This intolerance can be harnessed to reduce occurrences of aberrant implementations.

Intolerance toward violations of specification improves feedback for new implementations in particular. When a new implementation encounters a peer that is intolerant of an error, it receives strong feedback that allows the problem to be discovered quickly.

To be effective, intolerant implementations need to be sufficiently widely deployed that they are encountered by new implementations with high probability. This could depend on multiple implementations deploying strict checks.

This does not mean that intolerance of errors in early deployments of protocols have the effect of preventing interoperability. On the contrary, when existing implementations follow clearly specified error handling, new implementations or features can be introduced more readily as the effect on existing implementations can be easily predicted; see also Section 6.

Any intolerance also needs to be strongly supported by specifications, otherwise they encourage fracturing of the protocol community or proliferation of workarounds; see Section 8.

Intolerance can be used to motivate compliance with any protocol requirement. For instance, the `INADEQUATE_SECURITY` error code and associated requirements in HTTP/2 [HTTP/2] resulted in improvements in the security of the deployed base.

## 8. Exclusion

Any protocol participant that is affected by changes arising from maintenance might be excluded if they are unwilling or unable to implement or deploy changes that are made to the protocol.

Deliberate exclusion of problematic implementations is an important tool that can ensure that the interoperability of a protocol remains viable. While compatible changes are always preferable to incompatible ones, it is not always possible to produce a design that protects the ability of all current and future protocol participants to interoperate. Developing and deploying changes that risk exclusion of previously interoperating implementations requires some care, but changes to a protocol should not be blocked on the grounds of the risk of exclusion alone.

Exclusion is a direct goal when choosing to be intolerant of errors (see Section 7). Exclusionary actions are employed with the deliberate intent of protecting future interoperability.

Excluding implementations or deployments can lead to a fracturing of the protocol system that could be more harmful than any divergence resulting from following the robustness principle. The IAB document on Uncoordinated Protocol Development Considered Harmful [RFC5704] describes how conflict or competition in the maintenance of protocols can lead to similar problems.

## 9. Security Considerations

Sloppy implementations, lax interpretations of specifications, and uncoordinated extrapolation of requirements to cover gaps in specification can result in security problems. Hiding the consequences of protocol variations encourages the hiding of issues, which can conceal bugs and make them difficult to discover.

The consequences of the problems described in this document are especially acute for any protocol where security depends on agreement about semantics of protocol elements. For instance, use of unsafe security mechanisms, such as weak primitives [MD5] or obsolete mechanisms [SSL3], are good examples of where forcing exclusion (Section 8) can be desirable.

## 10. IANA Considerations

This document has no IANA actions.

## 11. Informative References

- [ECMA262] "ECMAScript(R) 2018 Language Specification", ECMA-262 9th Edition, June 2018, <<https://www.ecma-international.org/publications/standards/Ecma-262.htm>>.
- [EXT] Carpenter, B., Aboba, B., Ed., and S. Cheshire, "Design Considerations for Protocol Extensions", RFC 6709, DOI 10.17487/RFC6709, September 2012, <<https://www.rfc-editor.org/rfc/rfc6709>>.
- [HTML] "HTML", WHATWG Living Standard, 8 March 2019, <<https://html.spec.whatwg.org/>>.
- [HTTP] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <<https://www.rfc-editor.org/rfc/rfc7230>>.
- [HTTP/2] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <<https://www.rfc-editor.org/rfc/rfc7540>>.
- [MD5] Turner, S. and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms", RFC 6151, DOI 10.17487/RFC6151, March 2011, <<https://www.rfc-editor.org/rfc/rfc6151>>.



- [RFC0760] Postel, J., "DoD standard Internet Protocol", RFC 760, DOI 10.17487/RFC0760, January 1980, <<https://www.rfc-editor.org/rfc/rfc760>>.
- [RFC1958] Carpenter, B., Ed., "Architectural Principles of the Internet", RFC 1958, DOI 10.17487/RFC1958, June 1996, <<https://www.rfc-editor.org/rfc/rfc1958>>.
- [RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, DOI 10.17487/RFC4627, July 2006, <<https://www.rfc-editor.org/rfc/rfc4627>>.
- [RFC5218] Thaler, D. and B. Aboba, "What Makes for a Successful Protocol?", RFC 5218, DOI 10.17487/RFC5218, July 2008, <<https://www.rfc-editor.org/rfc/rfc5218>>.
- [RFC5704] Bryant, S., Ed., Morrow, M., Ed., and IAB, "Uncoordinated Protocol Development Considered Harmful", RFC 5704, DOI 10.17487/RFC5704, November 2009, <<https://www.rfc-editor.org/rfc/rfc5704>>.
- [RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <<https://www.rfc-editor.org/rfc/rfc7159>>.
- [RFC7493] Bray, T., Ed., "The I-JSON Message Format", RFC 7493, DOI 10.17487/RFC7493, March 2015, <<https://www.rfc-editor.org/rfc/rfc7493>>.
- [RFC9170] Thomson, M. and T. Pauly, "Long-Term Viability of Protocol Extension Mechanisms", RFC 9170, DOI 10.17487/RFC9170, December 2021, <<https://www.rfc-editor.org/rfc/rfc9170>>.
- [SSL3] Barnes, R., Thomson, M., Pironti, A., and A. Langley, "Deprecating Secure Sockets Layer Version 3.0", RFC 7568, DOI 10.17487/RFC7568, June 2015, <<https://www.rfc-editor.org/rfc/rfc7568>>.
- [TLS] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/rfc/rfc8446>>.

## Acknowledgments

Constructive feedback on this document has been provided by a surprising number of people including Bernard Aboba, Brian Carpenter, Stuart Cheshire, Mark Nottingham, Russ Housley, Eric Rescorla, Henning Schulzrinne, Robert Sparks, Brian Trammell, and Anne Van Kesteren. Please excuse any omission.

## Authors' Addresses

Martin Thomson  
Mozilla  
Email: [mt@lowentropy.net](mailto:mt@lowentropy.net)

David Schinazi  
Google LLC  
1600 Amphitheatre Parkway  
Mountain View, CA 94043  
United States of America  
Email: [dschinazi.ietf@gmail.com](mailto:dschinazi.ietf@gmail.com)

Network Working Group  
Internet-Draft

P. Saint-Andre, Ed.  
16 March 2022

Obsoletes: 8728 (if approved)  
Updates: 7841, 8729, 8730 (if approved)  
Intended status: Informational  
Expires: 17 September 2022

RFC Editor Model (Version 3)  
draft-iab-rfcedp-rfced-model-13

## Abstract

This document specifies version 3 of the RFC Editor Model. The Model defines two high-level tasks related to the RFC Series. First, policy definition is the joint responsibility of the RFC Series Working Group (RSWG), which produces policy proposals, and the RFC Series Approval Board (RSAB), which approves such proposals. Second, policy implementation is primarily the responsibility of the RFC Production Center (RPC) as contractually overseen by the IETF Administration Limited Liability Company (IETF LLC). In addition, various responsibilities of the "RFC Editor Function" are now performed alone or in combination by the RSWG, RSAB, RPC, RFC Series Consulting Editor (RSCE), and IETF LLC. Finally, this document establishes the Editorial Stream for publication of future policy definition documents produced through the processes defined herein.

This document obsoletes RFC 8728. This document updates RFC 7841, RFC 8729, and RFC 8730.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 17 September 2022.

## Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Overview of the Model . . . . .	4
3. Policy Definition . . . . .	5
3.1. Structure and Roles . . . . .	6
3.1.1. RFC Series Working Group (RSWG) . . . . .	6
3.1.2. RFC Series Approval Board (RSAB) . . . . .	7
3.2. Process . . . . .	10
3.2.1. Intent . . . . .	10
3.2.2. Workflow . . . . .	11
3.2.3. Community Calls for Comment . . . . .	14
3.2.4. Appeals . . . . .	15
3.2.5. Anti-Harassment Policy . . . . .	15
3.2.6. RFC Boilerplates . . . . .	16
4. Policy Implementation . . . . .	16
4.1. Roles and Processes . . . . .	16
4.2. Working Practices . . . . .	17
4.3. RPC Responsibilities . . . . .	18
4.4. Resolution of Disagreements between Authors and the RPC . . . . .	19
4.5. Point of Contact . . . . .	20
4.6. Administrative Implementation . . . . .	20
4.6.1. Vendor Selection for the RFC Production Center . . . . .	20
4.6.2. Budget . . . . .	21
5. RFC Series Consulting Editor (RSCE) . . . . .	21
5.1. RSCE Selection . . . . .	22
5.2. RSCE Performance Evaluation . . . . .	22
5.3. Temporary RSCE Appointment . . . . .	23
5.4. Conflict of Interest . . . . .	23
6. Editorial Stream . . . . .	23
6.1. Procedures Request of the IETF Trust . . . . .	24
6.2. Patent and Trademark Rules for the Editorial Stream . . . . .	24
6.3. Editorial Stream Boilerplate . . . . .	24

7.	Historical Properties of the RFC Series . . . . .	25
7.1.	Availability . . . . .	25
7.2.	Accessibility . . . . .	25
7.3.	Language . . . . .	25
7.4.	Diversity . . . . .	26
7.5.	Quality . . . . .	26
7.6.	Stability . . . . .	26
7.7.	Longevity . . . . .	26
8.	Updates to This Document . . . . .	26
9.	Changes from Version 2 of the RFC Editor Model . . . . .	26
9.1.	RFC Editor Function . . . . .	27
9.2.	RFC Series Editor . . . . .	27
9.3.	RFC Publisher . . . . .	28
9.4.	IAB . . . . .	28
9.5.	RFC Series Oversight Committee (RSOC) . . . . .	28
9.6.	RFC Series Advisory Group (RSAG) . . . . .	28
9.7.	Editorial Stream . . . . .	28
10.	Security Considerations . . . . .	29
11.	IANA Considerations . . . . .	29
12.	References . . . . .	29
12.1.	Normative References . . . . .	29
12.2.	Informative References . . . . .	31
	Acknowledgments . . . . .	32
	Author's Address . . . . .	32

## 1. Introduction

The Request for Comments (RFC) Series is the archival series dedicated to documenting Internet technical specifications, including general contributions from the Internet research and engineering community as well as standards documents. RFCs are available free of charge to anyone via the Internet. As described in [RFC8700], RFCs have been published continually since 1969.

RFCs are generated and approved by multiple document streams. Whereas the stream approving body [RFC8729] for each stream is responsible for the content of that stream, the RFC Editor Function is responsible for the production and distribution of all RFCs. The four existing streams are described in [RFC8729]. This document adds a fifth stream, the Editorial Stream, for publication of policies governing the RFC Series as a whole.

The overall framework for the RFC Series and the RFC Editor Function is described in [RFC8729] and is updated by this document, which defines version 3 of the RFC Editor Model. Under this version, various responsibilities of the RFC Editor Function are performed alone or in combination by the RFC Series Working Group (RSWG), RFC Series Advisory Board (RSAB), RFC Production Center (RPC), RFC Series

Consulting Editor (RSCE), and IETF Administration Limited Liability Company (IETF LLC) [RFC8711], which collectively comprise the RFC Editor Function. The intent is to ensure sustainable maintenance and support of the RFC Series based on the principles of expert implementation, clear management and direction, and appropriate community input [RFC8729].

This document obsoletes [RFC8728] by defining version 3 of the RFC Editor Model. This document updates [RFC7841] by defining boilerplate text for the Editorial Stream. This document updates [RFC8729] by replacing the RFC Editor role with the RSWG, RSAB, and RSCE. This document updates [RFC8730] by removing the dependency on certain policies specified by the IAB and RFC Series Editor (RSE). More detailed information about changes from version 2 of the Model can be found under Section 9.

## 2. Overview of the Model

This document divides the responsibilities for the RFC Series into two high-level tasks:

1. Policy definition governing the Series as a whole. This is the joint responsibility of two entities. First, the RFC Series Working Group (RSWG) is an open working group independent of the IETF that generates policy proposals. Second, the RFC Series Approval Board (RSAB) is an appointed body that approves such proposals for publication in the Editorial Stream. The RSAB includes representatives of the streams [RFC8729] as well as an expert in technical publishing, the RFC Series Consulting Editor (RSCE).
2. Policy implementation through publication of RFCs in all of the streams that form the Series. This is primarily the responsibility of the RFC Production Center (RPC) as contractually overseen by the IETF Administration Limited Liability Company (IETF LLC) [RFC8711].

As described more fully in the remainder of this document, the core activities and responsibilities are as follows:

- \* The RSWG proposes policies that govern the RFC Series as a whole, with input from the community, the RSAB, and the RSCE.
- \* The RSAB considers those proposals and either approves them or returns them to the RSWG, which may make further changes or remove them from further consideration.

- \* If approved, such proposals are published as RFCs in the Editorial Stream and thus define the policies to be followed by the RSWG, RSAB, RSCE, and RPC.
- \* The RSCE provides expert advice to the RPC and RSAB on how to implement established policies on an ongoing and operational basis, which can include raising issues or initiating proposed policy changes within the RSWG.
- \* The RPC implements the policies defined by the Editorial Stream in its day-to-day editing and publication of RFCs from all of the streams.
- \* If issues arise with the implementation of particular policies, the RPC brings those issues to the RSAB, which interprets the policies and provides interim guidance to the RPC, informing the RSWG of those interpretations.

This model is designed to ensure public processes and policy documents, clear lines of responsibility and authority, transparent mechanisms for updates and changes to policies governing the RFC Series as a whole, and effective operational implementation of the RFC Series, thus meeting the requirements specified in Section 4 of [RFC8729].

The remainder of this document describes the model in greater detail.

### 3. Policy Definition

Policies governing the RFC Series as a whole are defined through the following high-level process:

1. Proposals must be submitted to, adopted by, and discussed within the RFC Series Working Group (RSWG).
2. Proposals must pass a last call for comments in the working group and a community call for comments (see Section 3.2.3).
3. Proposals must be approved by the RFC Series Approval Board (RSAB).

Policies under the purview of the RSWG and RSAB might include, but are not limited to, document formats, processes for publication and dissemination of RFCs, and overall management of the RFC Series.

### 3.1. Structure and Roles

#### 3.1.1. RFC Series Working Group (RSWG)

##### 3.1.1.1. Purpose

The RFC Series Working Group (RSWG) is the primary venue in which members of the community collaborate regarding the policies that govern the RFC Series.

##### 3.1.1.2. Participation

All interested individuals are welcome to participate in the RSWG (subject to anti-harassment policies as described under Section 3.2.5). This includes but is not limited to participants in the IETF and IRTF, members of the IAB and IESG, developers of software or hardware systems that implement RFCs, authors of RFCs and Internet-Drafts, developers of tools used to author or edit RFCs, individuals who use RFCs in procurement decisions, scholarly researchers, and representatives of standards development organizations other than the IETF and IRTF. The IETF LLC Board members, staff and contractors (especially representatives of the RFC Production Center), and the IETF Executive Director are invited to participate as community members in the RSWG to the extent permitted by any relevant IETF LLC policies. Members of the RSAB are also expected to participate actively.

##### 3.1.1.3. Chairs

The RSWG shall have two chairs, one appointed by the IESG and the other appointed by the IAB. When the RSWG is formed, the chair appointed by the IESG shall serve for a term of one (1) year and the chair appointed by the IAB shall serve for a term of two (2) years; thereafter, chairs shall serve for a term of two (2) years, with no term limits on renewal. The IESG and IAB shall determine their own processes for making these appointments, making sure to take account of any potential conflicts of interest. Community members who have concerns about the performance of an RSWG chair should direct their feedback to the appropriate appointing body via mechanisms such bodies shall specify at the time that the RSWG is formed. The IESG and IAB shall have the power to remove their appointed chairs at their discretion at any time, and to name a replacement who shall serve the remainder of the original chair's term.

It is the responsibility of the chairs to encourage rough consensus within the RSWG and to follow that consensus in their decision making, for instance regarding acceptance of new proposals and advancement of proposals to the RSAB.



#### 3.1.1.4. Mode of Operation

The intent is that the RSWG shall operate in a way similar to that of working groups in the IETF. Therefore, all RSWG meetings and discussion venues shall be open to all interested individuals, and all RSWG contributions shall be subject to intellectual property policies, which must be consistent with those of the IETF as specified in [BCP78] and [BCP79].

When the RSWG is formed, all discussions shall take place on an open email discussion list, which shall be publicly archived.

The RSWG is empowered to hold in-person, online-only, or hybrid meetings, which should be announced with sufficient notice to enable broad participation; the IESG Guidance on Face-to-Face and Virtual Interim Meetings (<https://www.ietf.org/about/groups/iesg/statements/interim-meetings-guidance-2016-01-16/>) provides a reasonable baseline. In-person meetings should include provision for effective online participation for those unable to attend in person.

The RSWG shall operate by rough consensus, a mode of operation informally described in [RFC2418].

The RSWG may decide by rough consensus to use additional tooling (e.g., GitHub as specified in [RFC8874]), forms of communication, and working methods (e.g., design teams) as long as they are consistent with this document and with [RFC2418] or its successors.

Absent specific guidance in this document regarding the operation of the RSWG, the general guidance provided in Section 6 of [RFC2418] should be considered appropriate.

The IETF LLC is requested to provide necessary tooling to support RSWG communication, decision processes, and policies.

The IAB is requested to convene the RSWG when it is first formed in order to formalize the IAB's transfer of authority over the RFC Editor Model.

#### 3.1.2. RFC Series Approval Board (RSAB)

### 3.1.2.1. Purpose

The RFC Series Approval Board (RSAB), which includes representatives of all of the streams, shall act as the approving body for proposals generated within the RSWG, thus providing an appropriate set of "checks and balances" on the output of the RSWG. The only policy-making role of the RSAB is to review policy proposals generated by the RSWG; it shall have no independent authority to formulate policy on its own. It is expected that the RSAB will respect the rough consensus of the RSWG wherever possible, without ceding its responsibility to review RSWG proposals as further described under Section 3.2.2.

### 3.1.2.2. Members

The RSAB consists primarily of the following voting members:

- \* As the stream representative for the IETF stream, an IESG member or other person appointed by the IESG
- \* As the stream representative for the IAB stream, an IAB member or other person appointed by the IAB
- \* As the stream representative for the IRTF stream, the IRTF chair or other person appointed by the IRTF Chair
- \* As the stream representative for the Independent stream, the Independent Submissions Editor (ISE) [RFC8730] or other person appointed by the ISE
- \* The RFC Series Consulting Editor (RSCE)

If and when a new stream is created, the document that creates the stream shall specify if a voting member representing that stream shall also be added to the RSAB, along with any rules and processes related to that representative (e.g., whether the representative is a member of the body responsible for the stream or an appointed delegate thereof).

The RFC Series Consulting Editor (RSCE) is a voting member of the RSAB but does not act as a representative of the Editorial Stream.

To ensure the smooth operation of the RFC Series, the RSAB shall include the following non-voting, ex-officio members:

- \* The IETF Executive Director or their delegate; the rationale is that the IETF LLC is accountable for implementation of policies governing the RFC Series

- \* A representative of the RPC, named by the RPC; the rationale is that the RPC is responsible for implementation of policies governing the RFC Series

In addition to the foregoing, the RSAB may at its discretion include other non-voting members, whether ex-officio members or liaisons from groups or organizations with which the RSAB deems it necessary to formally collaborate or coordinate.

#### 3.1.2.3. Appointment and Removal of Voting Members

The appointing bodies, i.e., the stream approving bodies (IESG, IAB, IRTF chair, and ISE), shall determine their own processes for appointing RSAB members (note that processes related to the RSCE are described under Section 5). Each appointing body shall have the power to remove its appointed RSAB member at its discretion at any time. Appointing bodies should ensure that voting members are seated at all times and should fill any vacancies with all due speed, if necessary on a temporary basis.

In the case that the IRTF chair or ISE is incapacitated or otherwise unable to appoint another person to serve as a delegate, the IAB (as the appointing body for the IRTF chair and ISE) shall act as the temporary appointing body for those streams and shall appoint a temporary member of the RSAB until the IAB has appointed an IRTF chair or ISE, who can then act as an RSAB member or appoint a delegate through normal processes.

#### 3.1.2.4. Vacancies

In the case of vacancies by voting members, the RSAB shall operate as follows:

- \* Activities related to implementation of policies already in force shall continue as normal.
- \* Voting on approval of policy documents produced by the RSWG shall be delayed until the vacancy or vacancies have been filled, up to a maximum of 3 months. If during this 3-month period a further vacancy arises, the delay should be extended by up to another 3 months. After the delay period expires, the RSAB should continue to process documents as described below. Note: this method of handling vacancies does not apply to a vacancy of the RSCE role, only of the stream representatives enumerated above.

### 3.1.2.5. Chair

The RSAB shall annually choose a chair from among its members using a method of its choosing. If the chair position is vacated during the chair's term, the RSAB chooses a new chair from among its members.

### 3.1.2.6. Mode of Operation

The RSAB is expected to operate via an email discussion list, in-person meetings, teleconferencing systems, and any additional tooling it deems necessary.

The RSAB shall keep a public record of its proceedings, including minutes of all meetings and a record of all decisions. The primary email discussion list used by the RSAB shall be publicly archived, although topics that require confidentiality (e.g., personnel matters) may be omitted from such archives or discussed in private. Similarly, meeting minutes may exclude detailed information about topics discussed under executive session, but should note that such topics were discussed.

The RSAB shall announce plans and agendas for their meetings on the RFC Editor website and by email to the RSWG at least a week before such meetings. The meetings shall be open for public attendance and the RSAB may consider allowing open participation. If the RSAB needs to discuss a confidential matter in executive session, that part of the meeting shall be private to the RSAB, but must be noted on the agenda, and must be documented in the minutes with as much detail as confidentiality requirements permit.

The IETF LLC is requested to provide necessary tooling and staff to support RSAB communication, decision processes, and policies.

The IAB is requested to convene the RSAB when it is first formed in order to formalize the IAB's transfer of authority over the RFC Editor Model.

## 3.2. Process

### 3.2.1. Intent

The intent is to provide an open forum by which policies related to the RFC Series are defined and evolved. The general expectation is that all interested parties will participate in the RSWG, and that only under extreme circumstances should RSAB members need to hold "CONCERN" positions (as described under Section 3.2.2).

Because policy issues can be difficult and contentious, RSWG participants and RSAB members are strongly encouraged to work together in a spirit of good faith and mutual understanding to achieve rough consensus (see [RFC2418]). In particular, RSWG members are encouraged to take RSAB concerns seriously, and RSAB members are encouraged to clearly express their concerns early in the process and to be responsive to the community. All parties are encouraged to respect the value of each stream and the long-term health and viability of the RFC Series.

This process is intended to be one of continuous consultation. RSAB members should consult with their constituent stakeholders (e.g., authors, editors, tool developers, and consumers of RFCs) on an ongoing basis, so that when the time comes to consider the approval of a proposal, there should be no surprises. Appointing bodies are expected to establish whatever processes they deem appropriate to facilitate this goal.

### 3.2.2. Workflow

The following process shall be used to formulate or modify policies related to the RFC Series:

1. An individual or set of individuals generates a proposal in the form of an Internet-Draft (which must be submitted in full conformance with the provisions of [BCP78] and [BCP79]) and asks the RSWG to adopt the proposal as a working group item.
2. The RSWG may adopt the proposal as a draft proposal of the RSWG, if the chairs determine (by following working group procedures for rough consensus) that there is sufficient interest in the proposal; this is similar to the way a working group of the IETF would operate (see [RFC2418]).
3. The RSWG shall then further discuss and develop the proposal. All participants, but especially RSAB members, should pay special attention to any aspects of the proposal that have the potential to significantly modify policies of long standing or historical characteristics of the Series as described under Section 7. Members of the RSAB are expected to participate as individuals in all discussions relating to RSWG proposals. This should help to ensure that they are fully aware of proposals early in the policy definition process. It should also help to ensure that RSAB members will raise any issues or concerns during the development of the proposal, and not wait until the RSAB review period. The RSWG chairs are also expected to participate as individuals.

4. At some point, if the RSWG chairs believe there may be rough consensus for the proposal to advance, they will issue a last call for comments within the working group.
5. After a comment period of suitable length, the RSWG chairs will determine whether rough consensus for the proposal exists (taking their own feedback as individuals into account along with feedback from other participants). If comments have been received and substantial changes have been made, additional last calls may be necessary. Once the chairs determine that consensus has been reached, they shall announce their determination on the RSWG discussion list and forward the document to the RSAB.
6. Once consensus is established in the RSWG, the RSAB shall issue a community call for comments as further described under Section 3.2.3. If substantial comments are received in response to the community call for comments, the RSAB may return the draft to the RSWG to consider those comments and make revisions to address the feedback received. In parallel with the community call for comments, the RSAB itself shall also consider the proposal.
7. If the scope of the revisions made in the previous step is substantial, an additional community call for comments should be issued by the RSAB, and the feedback received should be considered by the RSWG.
8. Once the RSWG chairs confirm that concerns received during the community call(s) for comments have been addressed, they shall inform the RSAB that the document is ready for balloting by the RSAB.
9. Within a reasonable period of time, the RSAB will then poll its members for their positions on the proposal. Positions may be as follows:
  - \* "YES": the proposal should be approved
  - \* "CONCERN": the proposal raises substantial concerns that must be addressed
  - \* "RECUSE": the person holding the position has a conflict of interest

Any RSAB member holding a "CONCERN" position must explain their concern to the community in detail. Nevertheless, the RSWG might not be able to come to consensus on modifications that will address the RSAB member's concern.

There are three reasons why an RSAB member may file a position of CONCERN:

- \* The RSAB member believes that the proposal represents a serious problem for one or more of the individual streams.
- \* The RSAB member believes that the proposal would cause serious harm to the overall Series, including harm to the long-term health and viability of the Series.
- \* The RSAB member believes, based on the results of the community call(s) for comments Section 3.2.3, that rough consensus to advance the proposal is lacking.

Because RSAB members are expected to participate in the discussions within the RSWG and to raise any concerns and issues during those discussions, most CONCERN positions should not come as a surprise to the RSWG. Notwithstanding, late CONCERN positions are always possible if issues are identified during RSAB review or the community call(s) for comments.

10. If a CONCERN exists, discussion will take place within the RSWG. Again, all RSAB members are expected to participate. If substantial changes are made in order to address CONCERN positions, an additional community call for comments might be needed.
11. A proposal without any CONCERN positions is approved.
12. If, after a suitable period of time, any CONCERN positions remain, a vote of the RSAB is taken. If at least three voting members vote YES, the proposal is approved.
13. If the proposal is not approved, it is returned to the RSWG. The RSWG can then consider making further changes.
14. If the proposal is approved, a notification is sent to the community, and the document enters the queue for publication as an RFC within the Editorial Stream.

15. Policies may take effect immediately upon approval by the RSAB and before publication of the relevant RFC, unless they are delayed while the IETF LLC resolves pending resource or contract issues.

### 3.2.3. Community Calls for Comment

The RSAB is responsible for initiating and managing community calls for comments on proposals that have gained consensus within the RSWG. The RSAB should actively seek a wide range of input. The RSAB seeks such input by, at a minimum, sending a notice to the "rfc-interest" email list or to its successor or future equivalent. RSAB members should also send a notice to the communities they directly represent (e.g., the IETF and IRTF). Notices are also to be made available and archived on the RFC Editor website. In addition, other communication channels can be established for notices (e.g., via an RSS feed or by posting to social media venues).

In cases where a proposal has the potential to significantly modify policies of long standing or historical characteristics of the Series as described under Section 7, the RSAB should take extra care to reach out to a very wide range of communities that make use of RFCs (as described under Section 3.1.1.2) since such communities might not be actively engaged in the RSWG directly. The RSAB should work with the stream approving bodies and the IETF LLC to identify and establish contacts in such communities, assisted in particular by the RSCE.

The RSAB should maintain a public list of communities that are contacted during calls for comments.

A notice of a community call for comments contains the following:

- \* A subject line beginning with 'Call for Comments:'
- \* A clear, concise summary of the proposal
- \* A URL pointing to the Internet-Draft that defines the proposal
- \* Any explanations or questions for the community that the RSAB deems necessary (using their usual decision-making procedures)
- \* Clear instructions on how to provide public comments
- \* A deadline for comments



A comment period will last not less than two weeks and should be longer if wide outreach is required. Comments will be publicly archived on the RFC Editor website.

The RSAB is responsible for considering comments received during a community call for comments. If RSAB members conclude that such comments raise important issues that need to be addressed, they should do so by discussing those issues within the RSWG or (if the issues meet the criteria specified under Step 9 of Section 3.2.2) lodging a position of "CONCERN" during RSAB balloting.

#### 3.2.4. Appeals

Appeals of RSWG chair decisions shall be made to the RSAB. Decisions of the RSWG chairs can be appealed only on grounds of failure to follow the correct process. Appeals should be made within thirty (30) days of any action, or in the case of failure to act, of notice having been given to the RSWG chairs. The RSAB will then decide if the process was followed and will direct the RSWG chairs as to what procedural actions are required.

Decisions of the RSAB can be appealed on grounds of failure to follow the correct process. Where the RSAB makes a decision in order to resolve a disagreement between authors and the RPC (as described under Section 4.4), appeals can be filed on the basis that the RSAB misinterpreted an approved policy. Aside from these two cases, disagreements about the conduct of the RSAB are not subject to appeal. Appeals of RSAB decisions shall be made to the IAB and should be made within thirty (30) days of public notice of the relevant RSAB decision (typically, when minutes are posted). The IAB shall decide whether a process failure occurred and what if any corrective action should take place.

#### 3.2.5. Anti-Harassment Policy

The IETF anti-harassment policy (<https://www.ietf.org/about/groups/iesg/statements/anti-harassment-policy/>) also applies to the RSWG and RSAB, which strive to create and maintain an environment in which people of many different backgrounds are treated with dignity, decency, and respect. Participants are expected to behave according to professional standards and to demonstrate appropriate workplace behavior. For further information about these policies, see [RFC7154], [RFC7776], and [RFC8716].

### 3.2.6. RFC Boilerplates

RFC boilerplates (see [RFC7841]) are part of the RFC Style Guide, as defined below under Section 4.2. New or modified boilerplates considered under version 3 of the RFC Editor Model must be approved by the following parties, each of which has a separate area of responsibility with respect to boilerplates:

- \* Each applicable stream, which approves that the boilerplate meets its needs
- \* The RSAB, which approves that the boilerplate is not in conflict with the boilerplate used in the other streams
- \* The RPC, which approves that the language of the boilerplate is consistent with the RFC Style Guide
- \* The IETF Trust, which approves that the boilerplate correctly states the Trust's position regarding rights and ownership

## 4. Policy Implementation

### 4.1. Roles and Processes

Publication of RFCs is handled by the RFC Production Center (RPC).

A few general considerations apply:

- \* The general roles and responsibilities of the RPC are defined by RFCs published in the Editorial Stream (i.e., not directly by the RSWG, RSAB, or RSCE), by existing RFCs which apply to the RPC and which have not yet been superseded by Editorial Stream RFCs, and by the requisite contracts.
- \* The RPC is advised by the RSCE and RSAB, and has a duty to consult with them under specific circumstances, such as those relating to disagreements between authors and the RPC as described under Section 4.4.
- \* The RPC is overseen by the IETF LLC to ensure that it performs in accordance with contracts in place.

All matters of budget, timetable, and impact on its performance targets, are between the RPC and IETF LLC.

The RPC shall regularly provide reports to the IETF LLC, RSAB, RSWG, and broader community regarding its activities and any key risks or issues affecting it.

In the event that the RPC is required to make a decision without consultation that would normally deserve consultation, or makes a decision against the advice of the RSAB, the RPC must notify the RSAB.

This document does not specify the exact relationship between the IETF LLC and the RPC; for example, the work of the RPC could be performed by a separate corporate entity under contract to the IETF LLC, it could be performed by employees of the IETF LLC, or the IETF LLC could engage with independent contractors for some or all aspects of such work. The exact relationship is a matter for the IETF LLC to determine.

The IETF LLC is responsible for the method of and management of the engagement of the RPC. Therefore, the IETF LLC has authority over negotiating performance targets for the RPC and also has responsibility for ensuring that those targets are met. Such performance targets are set based on the RPC's publication load and additional efforts required to implement policies specified in the Editorial Stream, in existing RFCs which apply to the RPC and which have not yet been superseded by Editorial Stream RFCs, and in the requisite contracts. The IETF LLC may consult with the community regarding these targets. The IETF LLC is empowered to appoint a manager or to convene a committee to complete these activities.

If individuals or groups within the community have concerns about the performance of the RPC, they can request that the matter be investigated by the IETF LLC Board, the IETF LLC Executive Director, or a point of contact designated by the IETF LLC Board. Even if the IETF LLC opts to delegate this activity, concerns should be raised with the IETF LLC. The IETF LLC is ultimately answerable to the community via the mechanisms outlined in its charter [RFC8711].

#### 4.2. Working Practices

In the absence of a high-level policy documented in an RFC, or in the interest of specifying the detail of its implementation of such policies, the RPC can document working practices regarding the editorial preparation and final publication and dissemination of RFCs. Examples include:

- \* Maintenance of a style guide that defines editorial standards for RFCs; specifically, the RFC Style Guide consists of [RFC7322] and the other documents and resources listed at [STYLEGUIDE].
- \* Instructions regarding the file formats that are accepted as input to the editing and publication process.

- \* Guidelines regarding the final structure and layout of published documents. In the context of the XML vocabulary [RFC7991], such guidelines could include clarifications regarding the preferred XML elements and attributes used to capture the semantic content of RFCs.

#### 4.3. RPC Responsibilities

The core responsibility of the RPC is the implementation of RFC Series policies through publication of RFCs (including the dimensions of document quality, timeliness of publication, and accessibility of results), while taking into account issues raised by the community through the RSWG and by the stream approving bodies. More specifically, the RPC's responsibilities at the time of writing include the following:

1. Editing documents originating from all RFC streams to ensure that they are consistent with the editorial standards specified in the RFC Style Guide.
2. Creating and preserving records of edits performed on documents.
3. Identifying where editorial changes might have technical impact and seeking necessary clarification.
4. Establishing the publication readiness of each document through communication with the authors, IANA, or stream-specific contacts, supplemented if needed by the RSAB and RSCE.
5. Creating and preserving records of dialogue with document authors.
6. Requesting advice from the RSAB and RSCE as needed.
7. Providing suggestions to the RSAB and RSCE as needed.
8. Participating within the RSWG in the creation of new Editorial Stream RFCs that impact the RPC, specifically with respect to any challenges the RPC might foresee with regard to implementation of proposed policies.
9. Identifying topics and issues that they encounter while processing documents or carrying out other responsibilities on this list for which they lack sufficient expertise, and identifying and conferring with relevant experts as needed.
10. Providing reports to the community on its performance and plans.

11. Consulting with the community on its plans.
12. Negotiating its specific plans and resources with the IETF LLC.
13. Providing sufficient resources to support reviews of RPC performance by the IETF LLC.
14. Coordinating with IANA to ensure that RFCs accurately document registration processes and assigned values for IANA registries.
15. Assigning RFC numbers.
16. Liaising with stream approving bodies and other representatives of the streams as needed.
17. Publishing RFCs, which includes:
  - \* posting copies to the RFC Editor site both individually and in collections
  - \* depositing copies with external archives
  - \* creating catalogs and catalog entries
  - \* announcing the publication to interested parties
18. Providing online access to RFCs.
19. Providing an online system to facilitate the submission, management, and display of errata to RFCs.
20. Maintaining the RFC Editor website.
21. Providing for the backup of RFCs.
22. Ensuring the storage and preservation of records.
23. Authenticating RFCs for legal proceedings.

#### 4.4. Resolution of Disagreements between Authors and the RPC

During the process of editorial preparation and publication, disagreements can arise between the authors of an RFC-to-be and the RPC. Where an existing policy clearly applies, typically such disagreements are handled in a straightforward manner through direct consultation between the authors and the RPC, sometimes in collaboration with stream-specific contacts.

However, if it is unclear whether an existing policy applies, or if it is unclear how to interpret an existing policy, the parties may need to consult with additional individuals or bodies (e.g., RSAB, IESG, IRSG, or stream approving bodies) to help achieve a resolution. The following points are intended to provide more specific guidance.

- \* If there is a conflict with a policy for a particular stream, to help achieve a resolution the RPC should consult with the relevant stream approving body (such as the IESG or IRSG) and other representatives of the relevant stream as appropriate.
- \* If there is a conflict with a cross-stream policy, the RPC should consult with the RSAB to achieve a resolution.
- \* The disagreement might raise a new issue that is not covered by an existing policy or that cannot be resolved through consultation between the RPC and other relevant individuals and bodies, as described above. In this case, the RSAB is responsible for (a) resolving the disagreement in a timely manner if necessary so that the relevant stream document(s) can be published before a new policy is defined and (b) bringing the issue to the RSWG so that a new policy can be defined.

#### 4.5. Point of Contact

From time to time, individuals or organizations external to the IETF and the broader RFC Series community may have questions about the RFC Series. Such inquiries should be directed to the `rfc-editor@rfc-editor.org` (mailto:`rfc-editor@rfc-editor.org`) email alias or to its successor or future equivalent and then handled by the appropriate bodies (e.g., RSAB, RPC) or individuals (e.g., RSWG chairs, RSCE).

#### 4.6. Administrative Implementation

The exact implementation of the administrative and contractual activities described here are a responsibility of the IETF LLC. This section provides general guidance regarding several aspects of such activities.

##### 4.6.1. Vendor Selection for the RFC Production Center

Vendor selection is done in cooperation with the streams and under the final authority of the IETF LLC.

The IETF LLC develops the work definition (the Statement of Work) for the RPC and manages the vendor selection process. The work definition is created within the IETF LLC budget and takes into account the RPC responsibilities (as described under Section 4.3), the needs of the streams, and community input.

The process to select and contract for the RFC Production Center and other RFC-related services is as follows:

- \* The IETF LLC establishes the contract process, including the steps necessary to issue an RFP when necessary, the timing, and the contracting procedures.
- \* The IETF LLC establishes a selection committee, which will consist of the IETF Executive Director and other members selected by the IETF LLC in consultation with the stream approving bodies. The committee shall select a chair from among its members.
- \* The selection committee selects the vendor, subject to the successful negotiation of a contract approved by the IETF LLC. In the event that a contract cannot be signed, the matter shall be referred to the selection committee for further action.

#### 4.6.2. Budget

Most expenses discussed in this document are not new expenses. They have been and remain part of the IETF LLC budget.

The RFC Series portion of the IETF LLC budget shall include funding to support the RSCE, the RFC Production Center, and the Independent Stream.

The IETF LLC has the responsibility to approve the total RFC Editor budget (and the authority to deny it). All relevant parties must work within the IETF LLC budgetary process.

#### 5. RFC Series Consulting Editor (RSCE)

The RFC Series Consulting Editor (RSCE) is a senior technical publishing professional who will apply their deep knowledge of technical publishing processes to the RFC Series.

The primary responsibilities of the RSCE are as follows:

- \* Serve as a voting member on the RSAB
- \* Identify problems with the RFC publication process and opportunities for improvement

- \* Provide expert advice within the RSWG regarding policy proposals
- \* Provide expert advice to the RPC and IETF LLC

Matters on which the RSCE might provide guidance could include the following (see also Section 4 of [RFC8729]):

- \* Editing, processing, and publication of RFCs
- \* Publication formats for the RFC Series
- \* Changes to the RFC Style Guide
- \* Series-wide guidelines regarding document content and quality
- \* Web presence for the RFC Series
- \* Copyright matters related to the RFC Series
- \* Archiving, indexing, and accessibility of RFCs

The IETF LLC is responsible for the method of and management of the engagement of the RSCE, including selection, evaluation, and the timely filling of any vacancy. Therefore, whether the RSCE role is structured as a contractual or employee relationship is a matter for the IETF LLC to determine.

#### 5.1. RSCE Selection

Responsibility for making a recommendation to the IETF LLC regarding the RSCE role will lie with a selection committee. The IETF LLC should propose an initial slate of members for this committee, making sure to include community members with diverse perspectives, and consult with the stream representatives regarding the final membership of the committee. In making its recommendation for the role of RSCE, the selection committee will take into account the definition of the role as well as any other information that the committee deems necessary or helpful in making its decision. The IETF LLC is responsible for contracting or employment of the RSCE.

#### 5.2. RSCE Performance Evaluation

Periodically, the IETF LLC will evaluate the performance of the RSCE, including a call for confidential input from the community. The IETF LLC will produce a draft evaluation of the RSCE's performance for review by RSAB members other than the RSCE, who will provide feedback to the IETF LLC.



### 5.3. Temporary RSCE Appointment

In the case that the currently appointed RSCE is expected to be unavailable for an extended period, the IETF LLC may appoint a Temporary RSCE through whatever recruitment process it considers appropriate. A Temporary RSCE acts as the RSCE in all aspects during their term of appointment.

### 5.4. Conflict of Interest

The RSCE is expected to avoid even the appearance of conflict of interest or judgment in performing their role. To ensure this, the RSCE will be subject to a conflict of interest policy established by the IETF LLC.

The RPC service provider may contract services from the RSCE service provider, and vice versa including for services provided to the IETF LLC. All contracts between the two must be disclosed to the IETF LLC.

Where those services are related to services provided to the IETF LLC, IETF LLC policies shall apply, including publication of relevant parts of the contract.

## 6. Editorial Stream

This document creates the Editorial Stream as a separate space for publication of policies, procedures, guidelines, rules, and related information regarding the RFC Series as a whole.

The Editorial Stream shall be used only to specify and update policies, procedures, guidelines, rules, and related information regarding the RFC Series as a whole; no other use of the Editorial Stream is authorized by this memo and no other streams are so authorized. This policy may be changed only by agreement of the IAB, IESG, and IETF LLC.

All documents produced by the RSWG and approved by the RSAB shall be published as RFCs in the Editorial Stream with a status of Informational. (Note that the Editorial Stream is not authorized to publish RFCs that are Standards Track or Best Current Practice, since such RFCs are reserved to the IETF Stream [RFC8729].) Notwithstanding the status of "Informational", it should be understood that documents published in the Editorial Stream define policies for the RFC Series as a whole.

The requirements and process for creating any additional RFC streams are outside the scope of this document.

### 6.1. Procedures Request of the IETF Trust

The IAB requests that the IETF Trust and its Trustees assist in meeting the goals and procedures set forth in this document.

The Trustees are requested to publicly confirm their willingness and ability to accept responsibility for the Intellectual Property Rights (IPR) for the Editorial Stream.

Specifically, the Trustees are asked to develop the necessary boilerplate to enable the suitable marking of documents so that the IETF Trust receives the rights as specified in [BCP78]. These procedures need to also allow authors to indicate either no rights to make derivative works, or preferentially, the right to make unlimited derivative works from the documents. It is left to the Trust to specify exactly how this shall be clearly indicated in each document.

### 6.2. Patent and Trademark Rules for the Editorial Stream

As specified above, contributors of documents for the Editorial Stream are expected to use the IETF Internet-Draft process, complying therein with the rules specified in the latest version of [BCP9]. This includes the disclosure of Patent and Trademark issues that are known, or can be reasonably expected to be known, to the contributor.

Disclosure of license terms for patents is also requested, as specified in the most recent version of [BCP79]. The Editorial Stream has chosen to use the IETF's IPR disclosure mechanism, <https://www.ietf.org/ipr/>, for this purpose. The IAB would prefer that the most liberal terms possible be made available for Editorial Stream documents. Terms that do not require fees or licensing are preferable.

Non-discriminatory terms are strongly preferred over those that discriminate among users. However, although disclosure is required and the RSWG and the RSAB may consider disclosures and terms in making a decision as to whether to submit a document for publication, there are no specific requirements on the licensing terms for intellectual property related to Editorial Stream publication.

### 6.3. Editorial Stream Boilerplate

This document specifies the following text for the "Status of This Memo" section of RFCs published in the Editorial Stream. Any changes to this boilerplate must be made through the RFC Series Policy Definition process specified in this document.

Because all Editorial Stream RFCs have a status of Informational, the first paragraph of the "Status of This Memo" section shall be as specified in Appendix A.2.1 of [RFC7841].

The second paragraph of the "Status of This Memo" section shall be as follows:

This document is a product of the RFC Series Policy Definition process. It represents the consensus of the RFC Series Working Group approved by the RFC Series Approval Board. Such documents are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

The third paragraph of the "Status of This Memo" section shall be as specified in Section 3.5 of [RFC7841].

## 7. Historical Properties of the RFC Series

This section lists some of the properties that have been historically regarded as important to the RFC Series. Proposals that affect these properties are possible within the processes defined in this document. As described under Section 3.2.2 and Section 3.2.3, proposals that might have a detrimental effect on these properties should receive heightened scrutiny during RSWG discussion and RSAB review. The purpose of this scrutiny is to ensure that all changes are deliberate and that the consequences of a proposal, as far as they can be identified, have been carefully considered.

### 7.1. Availability

Documents in the RFC Series have been available for many decades, with no restrictions on access or distribution.

### 7.2. Accessibility

RFC Series documents have been published in a format that was intended to be as accessible as possible to people with disabilities, e.g., people with impaired sight.

### 7.3. Language

All existing RFC Series documents have been published in English. However, since the beginning of the RFC series, documents have been published under terms that explicitly allow translation into languages other than English without asking for permission.

#### 7.4. Diversity

The RFC series has included many types of documents including standards for the Internet, procedural and informational documents, thought experiments, speculative ideas, research papers, histories, humor, and even eulogies.

#### 7.5. Quality

RFC Series documents have been reviewed for subject matter quality and edited by professionals with a goal of ensuring that documents are clear, consistent, and readable [RFC7322].

#### 7.6. Stability

Once published, RFC Series documents have not changed.

#### 7.7. Longevity

RFC Series documents have been published in a form intended to be comprehensible to humans for decades or longer.

### 8. Updates to This Document

Updates, amendments, and refinements to this document can be produced using the process documented herein, but shall be published and operative only after (a) obtaining the agreement of the IAB and the IESG, and (b) ensuring that the IETF LLC has no objections regarding its ability to implement any proposed changes.

### 9. Changes from Version 2 of the RFC Editor Model

The processes and organizational models for publication of RFCs have changed significantly over the years. Most recently, in 2009 [RFC5620] defined the RFC Editor Model (Version 1) and in 2012 [RFC6635] defined the RFC Editor Model (Version 2), since modified slightly in 2020 by [RFC8728].

However, the community experienced several problems with version 1 and version 2, including a lack of transparency, a lack of avenues for community input into policy definition, and unclear lines of authority and responsibility.

To address these problems, in 2020 the IAB formed the RFC Editor Future Development Program to conduct a community discussion and consensus process for the further evolution of the RFC Editor Model. Under the auspices of this Program, the community considered changes that would increase transparency and community input regarding the

definition of policies for the RFC Series as a whole, while at the same time ensuring the continuity of the RFC Series, maintaining the quality and timely publication of RFCs, ensuring document accessibility, and clarifying lines of authority and responsibility.

This document is the result of discussion within the Program and describes version 3 of the RFC Editor Model while remaining consistent with [RFC8729].

The following sections describe the changes from version 2 in more detail.

### 9.1. RFC Editor Function

Several responsibilities previously assigned to the "RFC Editor" or, more precisely, the "RFC Editor Function" are now performed by the RSWG, RSAB, RPC, RSCE, and IETF LLC (alone or in combination). These include various aspects of strategic leadership (Section 2.1.1 of [RFC8728]), representation of the RFC Series (Section 2.1.2 of [RFC8728]), development of RFC production and publication (Section 2.1.3 of [RFC8728]), development of the RFC Series (Section 2.1.4 of [RFC8728]), operational oversight (Section 3.3 of [RFC8729]), policy oversight (Section 3.4 of [RFC8729]), the editing, processing, and publication of documents (Section 4.2 of [RFC8729]), and development and maintenance of Series-wide guidelines and rules (Section 4.4 of [RFC8729]). Among other things this changes the dependency on the RFC Series Editor (RSE) included in Section 2.2 of [RFC8730] with regard to "coordinating work and conforming to general RFC Series policies as specified by the IAB and RSE." In addition, various details regarding these responsibilities have been modified to accord with the framework defined in this document.

### 9.2. RFC Series Editor

Implied by the changes outlined in the previous section, the responsibilities of the RFC Series Editor (RSE) as a person or role (contrasted with the overall "RFC Editor Function") are now split or shared among the RSWG, RSAB, RSCE, RPC, and IETF LLC (alone or in combination). More specifically, the responsibilities of the RFC Series Consulting Editor (RSCE) under version 3 of the RFC Editor Model differ in many ways from the responsibilities of the RFC Series Editor under version 2 of the Model. In general, references in existing documents to the RSE can be taken as referring to the "RFC Editor Function" as described herein, but should not be taken as referring to the RSCE.

### 9.3. RFC Publisher

In practice the RFC Production Center (RPC) and RFC Publisher roles have been performed by the same entity and this practice is expected to continue; therefore this document dispenses with the distinction between these roles and refers only to the RPC.

### 9.4. IAB

Under earlier versions of the RFC Editor Model, the IAB was responsible for oversight of the RFC Series and acted as a body for final conflict resolution regarding the Series. The IAB's authority in these matters is described in the IAB's charter ([RFC2850] as updated by [I-D.draft-carpenter-rfced-iab-charter]). Under version 2 of the Model, the IAB delegated some of its authority to the RFC Series Oversight Committee (see Section 9.5). Under version 3 of the Model, authority for policy definition resides with the RSWG as an independent venue for work by members of the community (with approval of policy proposals as the responsibility of the RSAB, representing the streams and including the RSCE), whereas authority for policy implementation resides with the IETF LLC.

### 9.5. RFC Series Oversight Committee (RSOC)

In practice, the relationships and lines of authority and responsibility between the IAB, RSOC, and RSE have proved unwieldy and somewhat opaque. To overcome some of these issues, this document dispenses with the RSOC. References to the RSOC in documents such as [RFC8730] are obsolete because this document disbands the RSOC.

### 9.6. RFC Series Advisory Group (RSAG)

Version 1 of the RFC Editor Model [RFC5620] specified the existence of the RFC Series Advisory Group (RSAG), which was no longer specified in version 2 of the Model. For the avoidance of doubt, this document affirms that the RSAG has been disbanded. (The RSAG is not to be confused with the RFC Series Approval Board (RSAB), which this document establishes.)

### 9.7. Editorial Stream

This document creates the Editorial Stream in addition to the streams already described in [RFC8729].

## 10. Security Considerations

The same security considerations as those in [RFC8729] apply. The processes for the publication of documents must prevent the introduction of unapproved changes. Because multiple entities described in this document (most especially the RPC) participate in maintenance of the index of publications, sufficient security must be in place to prevent these published documents from being changed by external parties. The archive of RFC documents, any source documents needed to recreate the RFC documents, and any associated original documents (such as lists of errata, tools, and, for some early items, originals that are not machine-readable) need to be secured against data storage failure.

The IETF LLC (along with any other contracting or contracted entities) should take these security considerations into account during the implementation and enforcement of any relevant contracts.

## 11. IANA Considerations

The RPC is responsible for coordinating with the IANA to ensure that RFCs accurately document registration processes and assigned values for IANA registries.

The IETF LLC facilitates management of the relationship between the RPC and IANA.

This document does not create a new registry nor does it register any values in existing registries, and no IANA action is required.

## 12. References

### 12.1. Normative References

- [BCP78] Bradner, S., Ed. and J. Contreras, Ed., "Rights Contributors Provide to the IETF Trust", BCP 78, RFC 5378, November 2008.  
  
<<https://www.rfc-editor.org/info/bcp78>>
- [BCP79] Bradner, S. and J. Contreras, "Intellectual Property Rights in IETF Technology", BCP 79, RFC 8179, May 2017.  
  
<<https://www.rfc-editor.org/info/bcp79>>
- [BCP9] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

Dusseault, L. and R. Sparks, "Guidance on Interoperation and Implementation Reports for Advancement to Draft Standard", BCP 9, RFC 5657, September 2009.

Housley, R., Crocker, D., and E. Burger, "Reducing the Standards Track to Two Maturity Levels", BCP 9, RFC 6410, October 2011.

Resnick, P., "Retirement of the "Internet Official Protocol Standards" Summary Document", BCP 9, RFC 7100, December 2013.

Kolkman, O., Bradner, S., and S. Turner, "Characterization of Proposed Standards", BCP 9, RFC 7127, January 2014.

Dawkins, S., "Increasing the Number of Area Directors in an IETF Area", BCP 9, RFC 7475, March 2015.

Halpern, J., Ed. and E. Rescorla, Ed., "IETF Stream Documents Require IETF Rough Consensus", BCP 9, RFC 8789, June 2020.

<<https://www.rfc-editor.org/info/bcp9>>

- [RFC2418] Bradner, S., "IETF Working Group Guidelines and Procedures", BCP 25, RFC 2418, DOI 10.17487/RFC2418, September 1998, <<https://www.rfc-editor.org/info/rfc2418>>.
- [RFC7154] Moonesamy, S., Ed., "IETF Guidelines for Conduct", BCP 54, RFC 7154, DOI 10.17487/RFC7154, March 2014, <<https://www.rfc-editor.org/info/rfc7154>>.
- [RFC7322] Flanagan, H. and S. Ginoza, "RFC Style Guide", RFC 7322, DOI 10.17487/RFC7322, September 2014, <<https://www.rfc-editor.org/info/rfc7322>>.
- [RFC7776] Resnick, P. and A. Farrel, "IETF Anti-Harassment Procedures", BCP 25, RFC 7776, DOI 10.17487/RFC7776, March 2016, <<https://www.rfc-editor.org/info/rfc7776>>.
- [RFC7841] Halpern, J., Ed., Daigle, L., Ed., and O. Kolkman, Ed., "RFC Streams, Headers, and Boilerplates", RFC 7841, DOI 10.17487/RFC7841, May 2016, <<https://www.rfc-editor.org/info/rfc7841>>.
- [RFC8716] Resnick, P. and A. Farrel, "Update to the IETF Anti-Harassment Procedures for the Replacement of the IETF Administrative Oversight Committee (IAOC) with the IETF



Administration LLC", BCP 25, RFC 8716,  
DOI 10.17487/RFC8716, February 2020,  
<<https://www.rfc-editor.org/info/rfc8716>>.

- [RFC8729] Housley, R., Ed. and L. Daigle, Ed., "The RFC Series and RFC Editor", RFC 8729, DOI 10.17487/RFC8729, February 2020, <<https://www.rfc-editor.org/info/rfc8729>>.
- [RFC8730] Brownlee, N., Ed. and B. Hinden, Ed., "Independent Submission Editor Model", RFC 8730, DOI 10.17487/RFC8730, February 2020, <<https://www.rfc-editor.org/info/rfc8730>>.

## 12.2. Informative References

- [I-D.draft-carpenter-rfced-iab-charter]  
Carpenter, B. E., "IAB Charter Update for RFC Editor Model", Work in Progress, Internet-Draft, draft-carpenter-rfced-iab-charter-08, 15 March 2022,  
<<https://www.ietf.org/archive/id/draft-carpenter-rfced-iab-charter-08.txt>>.
- [RFC2850] Internet Architecture Board and B. Carpenter, Ed., "Charter of the Internet Architecture Board (IAB)", BCP 39, RFC 2850, DOI 10.17487/RFC2850, May 2000,  
<<https://www.rfc-editor.org/info/rfc2850>>.
- [RFC5620] Kolkman, O., Ed. and IAB, "RFC Editor Model (Version 1)", RFC 5620, DOI 10.17487/RFC5620, August 2009,  
<<https://www.rfc-editor.org/info/rfc5620>>.
- [RFC6635] Kolkman, O., Ed., Halpern, J., Ed., and IAB, "RFC Editor Model (Version 2)", RFC 6635, DOI 10.17487/RFC6635, June 2012, <<https://www.rfc-editor.org/info/rfc6635>>.
- [RFC7991] Hoffman, P., "The "xml2rfc" Version 3 Vocabulary", RFC 7991, DOI 10.17487/RFC7991, December 2016,  
<<https://www.rfc-editor.org/info/rfc7991>>.
- [RFC8700] Flanagan, H., Ed., "Fifty Years of RFCs", RFC 8700, DOI 10.17487/RFC8700, December 2019,  
<<https://www.rfc-editor.org/info/rfc8700>>.
- [RFC8711] Haberman, B., Hall, J., and J. Livingood, "Structure of the IETF Administrative Support Activity, Version 2.0", BCP 101, RFC 8711, DOI 10.17487/RFC8711, February 2020,  
<<https://www.rfc-editor.org/info/rfc8711>>.

[RFC8728] Kolkman, O., Ed., Halpern, J., Ed., and R. Hinden, Ed.,  
"RFC Editor Model (Version 2)", RFC 8728,  
DOI 10.17487/RFC8728, February 2020,  
<<https://www.rfc-editor.org/info/rfc8728>>.

[RFC8874] Thomson, M. and B. Stark, "Working Group GitHub Usage  
Guidance", RFC 8874, DOI 10.17487/RFC8874, August 2020,  
<<https://www.rfc-editor.org/info/rfc8874>>.

[STYLEGUIDE]  
RFC Editor, "Style Guide", 26 October 2021,  
<<https://www.rfc-editor.org/styleguide/>>.

#### Acknowledgments

Portions of this document were borrowed from [RFC5620], [RFC6635], [RFC8728], [RFC8729], the Frequently Asked Questions of the IETF Trust, and earlier proposals submitted within the IAB's RFC Editor Future Development Program by Martin Thomson, Brian Carpenter, and Michael StJohns. Thanks to Eliot Lear and Brian Rosen in their role as chairs of the Program for their leadership and assistance. Thanks also for feedback and proposed text to Jari Arkko, Sarah Banks, Carsten Bormann, Scott Bradner, Nevil Brownlee, Ben Campbell, Jay Daley, Martin Duerst (note: replace "ue" with U+00FC before publication), Wesley Eddy, Lars Eggert, Adrian Farrel, Stephen Farrell, Sandy Ginoza, Bron Gondwana, Joel Halpern, Wes Hardaker, Bob Hinden, Russ Housley, Christian Huitema, Ole Jacobsen, Sheng Jiang, Benjamin Kaduk, John Klensin, Murray Kucherawy, Mirja Kuehlewind, Ted Lemon, John Levine, Lucy Lynch, Jean Mahoney, Andrew Malis, Larry Masinter, S. Moonesamy, Russ Mundy, Mark Nottingham, Tommy Pauly, Colin Perkins, Julian Reschke, Eric Rescorla, Alvaro Retana, Adam Roach, Dan Romascanu, Alice Russo, Doug Royer, Rich Salz, John Scudder, Stig Venaas, Tim Wicinski, and Nico Williams.

#### Author's Address

Peter Saint-Andre (editor)  
Email: [stpeter@stpeter.im](mailto:stpeter@stpeter.im)

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 16 April 2022

M. Thomson  
Mozilla  
T. Pauly  
Apple  
13 October 2021

Long-term Viability of Protocol Extension Mechanisms  
draft-iab-use-it-or-lose-it-04

Abstract

The ability to change protocols depends on exercising the extension and version negotiation mechanisms that support change. This document explores how regular use of new protocol features can ensure that it remains possible to deploy changes to a protocol. Examples are given where lack of use caused changes to be more difficult or costly.

Discussion Venues

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

Discussion of this document takes place on the EDM Program mailing list ([edm@iab.org](mailto:edm@iab.org)), which is archived at <https://mailarchive.ietf.org/arch/browse/edm/>.

Source for this draft and an issue tracker can be found at <https://github.com/intarchboard/use-it-or-lose-it>.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 16 April 2022.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Imperfect Implementations Limit Protocol Evolution . . . . .	3
2.1. Good Protocol Design is Not Itself Sufficient . . . . .	4
2.2. Disuse Can Hide Problems . . . . .	5
2.3. Multi-Party Interactions and Middleboxes . . . . .	5
3. Active Use . . . . .	6
3.1. Dependency is Better . . . . .	7
3.2. Version Negotiation . . . . .	7
3.3. Falsifying Active Use . . . . .	8
3.4. Examples of Active Use . . . . .	9
3.5. Restoring Active Use . . . . .	10
4. Complementary Techniques . . . . .	10
4.1. Fewer Extension Points . . . . .	10
4.2. Invariants . . . . .	11
4.3. Limiting Participation . . . . .	11
4.4. Effective Feedback . . . . .	12
5. Security Considerations . . . . .	12
6. IANA Considerations . . . . .	13
7. Informative References . . . . .	13
Appendix A. Examples . . . . .	17
A.1. DNS . . . . .	17
A.2. HTTP . . . . .	18
A.3. IP . . . . .	18
A.4. SNMP . . . . .	19
A.5. TCP . . . . .	19
A.6. TLS . . . . .	19
Acknowledgments . . . . .	20
Authors' Addresses . . . . .	20

## 1. Introduction

A successful protocol [SUCCESS] needs to change in ways that allow it to continue to fulfill the changing needs of its users. New use cases, conditions and constraints on the deployment of a protocol can render a protocol that does not change obsolete.

Usage patterns and requirements for a protocol shift over time. In response, implementations might adjust usage patterns within the constraints of the protocol, the protocol could be extended, or a replacement protocol might be developed. Experience with Internet-scale protocol deployment shows that each option comes with different costs. [TRANSITIONS] examines the problem of protocol evolution more broadly.

An extension point is a mechanism that allows a protocol to be changed or enhanced. This document examines the specific conditions that determine whether protocol maintainers have the ability to design and deploy new or modified protocols via their specified extension points. Section 2 highlights some historical examples of difficulties in transitions to new protocol features. Section 3 argues that ossified protocols are more difficult to update and describes how successful protocols make frequent use of new extensions and code-points. Section 4 outlines several additional strategies that might aid in ensuring that protocol changes remain possible over time.

The experience that informs this document is predominantly at "higher" layers of the network stack, in protocols with limited numbers of participants. Though similar issues are present in many protocols that operate at scale, the tradeoffs involved with applying some of the suggested techniques can be more complex when there are many participants, such as at the network layer or in routing systems.

## 2. Imperfect Implementations Limit Protocol Evolution

It can be extremely difficult to deploy a change to a protocol if implementations with which the new deployment needs to interoperate do not operate predictably. Variation in how new codepoints or extensions are handled can be the result of bugs in implementation or specifications. Unpredictability can manifest as abrupt termination of sessions, errors, crashes, or disappearances of endpoints and timeouts.

The risk of interoperability problems can in turn make it infeasible to deploy certain protocol changes. If deploying a new codepoint or extension makes an implementation less reliable than others, even if only in rare cases, it is far less likely that implementations will adopt the change.

Deploying a change to a protocol could require implementations to fix a substantial proportion of the bugs that the change exposes. This can involve a difficult process that includes identifying the cause of these errors, finding the responsible implementation(s), coordinating a bug fix and release plan, contacting users and/or the operator of affected services, and waiting for the fix to be deployed.

Given the effort involved in fixing problems, the existence of these sorts of bugs can outright prevent the deployment of some types of protocol changes, especially for protocols involving multiple parties or that are considered critical infrastructure (e.g., IP, BGP, DNS, or TLS). It could even be necessary to come up with a new protocol design that uses a different method to achieve the same result.

This document only addresses cases where extensions are not deliberately blocked. Some deployments or implementations apply policies that explicitly prohibit the use of unknown capabilities. This is especially true of functions that seek to make security guarantees, like firewalls.

The set of interoperable features in a protocol is often the subset of its features that have some value to those implementing and deploying the protocol. It is not always the case that future extensibility is in that set.

## 2.1. Good Protocol Design is Not Itself Sufficient

It is often argued that the careful design of a protocol extension point or version negotiation capability is critical to the freedom that it ultimately offers.

RFC 6709 [EXTENSIBILITY] contains a great deal of well-considered advice on designing for extension. It includes the following advice:

This means that, to be useful, a protocol version-negotiation mechanism should be simple enough that it can reasonably be assumed that all the implementers of the first protocol version at least managed to implement the version-negotiation mechanism correctly.

There are a number of protocols for which this has proven to be insufficient in practice. These protocols have imperfect implementations of these mechanisms. Mechanisms that aren't used are the ones that fail most often. The same paragraph from RFC 6709 acknowledges the existence of this problem, but does not offer any remedy:

The nature of protocol version-negotiation mechanisms is that, by definition, they don't get widespread real-world testing until after the base protocol has been deployed for a while, and its deficiencies have become evident.

Indeed, basic interoperability is considered critical early in the deployment of a protocol. A desire to deploy can result in early focus on a reduced feature set, which could result in deferring implementation of version negotiation and extension mechanisms. This leads to these mechanisms being particularly affected by this problem.

## 2.2. Disuse Can Hide Problems

There are many examples of extension points in protocols that have been either completely unused, or their use was so infrequent that they could no longer be relied upon to function correctly.

Appendix A includes examples of disuse in a number of widely deployed Internet protocols.

Even where extension points have multiple valid values, if the set of permitted values does not change over time, there is still a risk that new values are not tolerated by existing implementations. If the set of values for a particular field or the order in which these values appear of a protocol remains fixed over a long period, some implementations might not correctly handle a new value when it is introduced. For example, implementations of TLS broke when new values of the `signature_algorithms` extension were introduced.

## 2.3. Multi-Party Interactions and Middleboxes

One of the key challenges in deploying new features is ensuring compatibility with all actors that could be involved in the protocol. Even the most superficially simple protocols can often involve more actors than is immediately apparent.

The design of extension points needs to consider what actions middleboxes might take in response to a protocol change, as well as the effect those actions could have on the operation of the protocol.

Deployments of protocol extensions also need to consider the impact of the changes on entities beyond protocol participants and middleboxes. Protocol changes can affect the behavior of applications or systems that don't directly interact with the protocol, such as when a protocol change modifies the formatting of data delivered to an application.

### 3. Active Use

The design of a protocol for extensibility and eventual replacement [EXTENSIBILITY] does not guarantee the ability to exercise those options. The set of features that enable future evolution need to be interoperable in the first implementations and deployments of the protocol. Implementation of mechanisms that support evolution is necessary to ensure that they remain available for new uses, and history has shown this occurs almost exclusively through active mechanism use.

Only by using the extension capabilities of a protocol is the availability of that capability assured. "Using" here includes specifying, implementing, and deploying capabilities that rely on the extension capability. Protocols that fail to use a mechanism, or a protocol that only rarely uses a mechanism, could lead to that mechanism being unreliable.

Implementations that routinely see new values are more likely to correctly handle new values. More frequent changes will improve the likelihood that incorrect handling or intolerance is discovered and rectified. The longer an intolerant implementation is deployed, the more difficult it is to correct.

Protocols that routinely add new extensions and code points rarely have trouble adding additional ones, especially when the handling of new versions or extensions are well defined. The definition of mechanisms alone is insufficient; it is the assured implementation and active use of those mechanisms that determines their availability.

What constitutes "active use" can depend greatly on the environment in which a protocol is deployed. The frequency of changes necessary to safeguard some mechanisms might be slow enough to attract ossification in another protocol deployment, while being excessive in others.



### 3.1. Dependency is Better

The easiest way to guarantee that a protocol mechanism is used is to make the handling of it critical to an endpoint participating in that protocol. This means that implementations must rely on both the existence of extension mechanisms and their continued, repeated expansion over time.

For example, the message format in SMTP relies on header fields for most of its functions, including the most basic delivery functions. A deployment of SMTP cannot avoid including an implementation of header field handling. In addition to this, the regularity with which new header fields are defined and used ensures that deployments frequently encounter header fields that they do not yet (and may never) understand. An SMTP implementation therefore needs to be able to both process header fields that it understands and ignore those that it does not.

In this way, implementing the extensibility mechanism is not merely mandated by the specification, it is crucial to the functioning of a protocol deployment. Should an implementation fail to correctly implement the mechanism, that failure would quickly become apparent.

Caution is advised to avoid assuming that building a dependency on an extension mechanism is sufficient to ensure availability of that mechanism in the long term. If the set of possible uses is narrowly constrained and deployments do not change over time, implementations might not see new variations or assume a narrower interpretation of what is possible. Those implementations might still exhibit errors when presented with new variations.

### 3.2. Version Negotiation

As noted in Section 2.1, protocols that provide version negotiation mechanisms might not be able to test that feature until a new version is deployed. One relatively successful design approach has been to use the protocol selection mechanisms built into a lower-layer protocol to select the protocol. This could allow a version negotiation mechanism to benefit from active use of the extension point by other protocols.

For instance, all published versions of IP contain a version number as the four high bits of the first header byte. However, version selection using this field proved to be unsuccessful. Ultimately, successful deployment of IPv6 over Ethernet [RFC2464] required a different EtherType from IPv4. This change took advantage of the already-diverse usage of EtherType.

Other examples of this style of design include Application-Layer Protocol Negotiation ([ALPN]) and HTTP content negotiation (Section 12 of [HTTP]).

This technique relies on the codepoint being usable. For instance, the IP protocol number is known to be unreliable and therefore not suitable [NEW-PROTOCOLS].

### 3.3. Falsifying Active Use

"Grease" was originally defined for TLS [GREASE], but has been adopted by other protocols, such as QUIC [QUIC]. Grease identifies lack of use as an issue (protocol mechanisms "rusting" shut) and proposes reserving values for extensions that have no semantic value attached.

The design in [GREASE] is aimed at the style of negotiation most used in TLS, where one endpoint offers a set of options and the other chooses the one that it most prefers from those that it supports. An endpoint that uses grease randomly offers options - usually just one - from a set of reserved values. These values are guaranteed to never be assigned real meaning, so its peer will never have cause to genuinely select one of these values.

More generally, greasing is used to refer to any attempt to exercise extension points without changing endpoint behavior, other than to encourage participants to tolerate new or varying values of protocol elements.

The principle that grease operates on is that an implementation that is regularly exposed to unknown values is less likely to be intolerant of new values when they appear. This depends largely on the assumption that the difficulty of implementing the extension mechanism correctly is as easy or easier than implementing code to identify and filter out reserved values. Reserving random or unevenly distributed values for this purpose is thought to further discourage special treatment.

Without reserved greasing codepoints, an implementation can use code points from spaces used for private or experimental use if such a range exists. In addition to the risk of triggering participation in an unwanted experiment, this can be less effective. Incorrect implementations might still be able to identify these code points and ignore them.

In addition to advertising bogus capabilities, an endpoint might also selectively disable non-critical protocol elements to test the ability of peers to handle the absence of certain capabilities.

This style of defensive design is limited because it is only superficial. As greasing only mimics active use of an extension point, it only exercises a small part of the mechanisms that support extensibility. More critically, it does not easily translate to all forms of extension points. For instance, HMSV negotiation cannot be greased in this fashion. Other techniques might be necessary for protocols that don't rely on the particular style of exchange that is predominant in TLS.

Grease is deployed with the intent of quickly revealing errors in implementing the mechanisms it safeguards. Though it has been effective at revealing problems in some cases with TLS, the efficacy of greasing isn't proven more generally. Where implementations are able to tolerate a non-zero error rate in their operation, greasing offers a potential option for safeguarding future extensibility. However, this relies on there being a sufficient proportion of participants that are willing to invest the effort and tolerate the risk of interoperability failures.

### 3.4. Examples of Active Use

Header fields in email [SMTP], HTTP [HTTP] and SIP [SIP] all derive from the same basic design, which amounts to a list name/value pairs. There is no evidence of significant barriers to deploying header fields with new names and semantics in email and HTTP as clients and servers generally ignore headers they do not understand or need. The widespread deployment of SIP B2BUAs, which generally do not ignore unknown fields, means that new SIP header fields do not reliably reach peers. This does not necessarily cause interoperability issues in SIP but rather causes features to remain unavailable until the B2BUA is updated. All three protocols are still able to deploy new features reliably, but SIP features are deployed more slowly due to the larger number of active participants that need to support new features.

As another example, the attribute-value pairs (AVPs) in Diameter [DIAMETER] are fundamental to the design of the protocol. Any use of Diameter requires exercising the ability to add new AVPs. This is routinely done without fear that the new feature might not be successfully deployed.

These examples show extension points that are heavily used are also being relatively unaffected by deployment issues preventing addition of new values for new use cases.

These examples show that a good design is not required for success. On the contrary, success is often despite shortcomings in the design. For instance, the shortcomings of HTTP header fields are significant enough that there are ongoing efforts to improve the syntax [HTTP-HEADERS].

### 3.5. Restoring Active Use

With enough effort, active use can be used to restore capabilities.

EDNS [EDNS] was defined to provide extensibility in DNS. Intolerance of the extension in DNS servers resulted in a fallback method being widely deployed (see Section 6.2.2 of [EDNS]). This fallback resulted in EDNS being disabled for affected servers. Over time, greater support for EDNS and increased reliance on it for different features motivated a flag day [DNSFLAGDAY] where the workaround was removed.

The EDNS example shows that effort can be used to restore capabilities. This is in part because EDNS was actively used with most resolvers and servers. It was therefore possible to force a change to ensure that extension capabilities would always be available. However, this required an enormous coordination effort. A small number of incompatible servers and the names they serve also became inaccessible to most clients.

## 4. Complementary Techniques

The protections to protocol evolution that come from active use (Section 3) can be improved through the use of other defensive techniques. The techniques listed here might not prevent ossification on their own, but can make active use more effective.

### 4.1. Fewer Extension Points

A successful protocol will include many potential types of extension. Designing multiple types of extension mechanism, each suited to a specific purpose, might leave some extension points less heavily used than others.

Disuse of a specialized extension point might render it unusable. In contrast, having a smaller number of extension points with wide applicability could improve the use of those extension points. Use of a shared extension point for any purpose can protect rarer or more specialized uses.

Both extensions and core protocol elements use the same extension points in protocols like HTTP [HTTP] and DIAMETER [DIAMETER]; see Section 3.4.

#### 4.2. Invariants

Documenting aspects of the protocol that cannot or will not change as extensions or new versions are added can be a useful exercise. Section 2.2 of [RFC5704] defines invariants as:

Invariants are core properties that are consistent across the network and do not change over extremely long time-scales.

Understanding what aspects of a protocol are invariant can help guide the process of identifying those parts of the protocol that might change. [QUIC-INVARIANTS] and Section 9.3 of [TLS13] are both examples of documented invariants.

As a means of protecting extensibility, a declaration of protocol invariants is useful only to the extent that protocol participants are willing to allow new uses for the protocol. A protocol that declares protocol invariants relies on implementations understanding and respecting those invariants. If active use is not possible for all non-invariant parts of the protocol, greasing (Section 3.3) might be used to improve the chance that invariants are respected.

Protocol invariants need to be clearly and concisely documented. Including examples of aspects of the protocol that are not invariant, such as Appendix A of [QUIC-INVARIANTS], can be used to clarify intent.

#### 4.3. Limiting Participation

Reducing the number of entities that can participate in a protocol or limiting the extent of participation can reduce the number of entities that might affect extensibility. Using TLS or other cryptographic tools can therefore reduce the number of entities that can influence whether new features are usable.

[PATH-SIGNALS] also recommends the use of encryption and integrity protection to limit participation. For example, encryption is used by the QUIC protocol [QUIC] to limit the information that is available to middleboxes and integrity protection prevents modification.

#### 4.4. Effective Feedback

While not a direct means of protecting extensibility mechanisms, feedback systems can be important to discovering problems.

Visibility of errors is critical to the success of techniques like grease (see Section 3.3). The grease design is most effective if a deployment has a means of detecting and reporting errors. Ignoring errors could allow problems to become entrenched.

Feedback on errors is more important during the development and early deployment of a change. It might also be helpful to disable automatic error recovery methods during development.

Automated feedback systems are important for automated systems, or where error recovery is also automated. For instance, connection failures with HTTP alternative services [ALT-SVC] are not permitted to affect the outcome of transactions. An automated feedback system for capturing failures in alternative services is therefore necessary for failures to be detected.

How errors are gathered and reported will depend greatly on the nature of the protocol deployment and the entity that receives the report. For instance, end users, developers, and network operations each have different requirements for how error reports are created, managed, and acted upon.

Automated delivery of error reports can be critical for rectifying deployment errors as early as possible, such as seen in [DMARC] and [SMTP-TLS-Reporting].

#### 5. Security Considerations

Many of the problems identified in this document are not the result of deliberate actions by an adversary, but more the result of mistakes, decisions made without sufficient context, or simple neglect. Problems therefore not the result of opposition by an adversary. In response, the recommended measures generally assume that other protocol participants will not take deliberate action to prevent protocol evolution.

The use of cryptographic techniques to exclude potential participants is the only strong measure that the document recommends. However, authorized protocol peers are most often responsible for the identified problems, which can mean that cryptography is insufficient to exclude them.

The ability to design, implement, and deploy new protocol mechanisms can be critical to security. In particular, it is important to be able to replace cryptographic algorithms over time [AGILITY]. For example, preparing for replacement of weak hash algorithms was made more difficult through misuse [HASH].

## 6. IANA Considerations

This document makes no request of IANA.

## 7. Informative References

- [AGILITY] Housley, R., "Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms", BCP 201, RFC 7696, DOI 10.17487/RFC7696, November 2015, <<https://www.rfc-editor.org/rfc/rfc7696>>.
- [ALPN] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301, July 2014, <<https://www.rfc-editor.org/rfc/rfc7301>>.
- [ALT-SVC] Nottingham, M., McManus, P., and J. Reschke, "HTTP Alternative Services", RFC 7838, DOI 10.17487/RFC7838, April 2016, <<https://www.rfc-editor.org/rfc/rfc7838>>.
- [DIAMETER] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <<https://www.rfc-editor.org/rfc/rfc6733>>.
- [DMARC] Kucherawy, M., Ed. and E. Zwicky, Ed., "Domain-based Message Authentication, Reporting, and Conformance (DMARC)", RFC 7489, DOI 10.17487/RFC7489, March 2015, <<https://www.rfc-editor.org/rfc/rfc7489>>.
- [DNSFLAGDAY] "DNS Flag Day 2019", May 2019, <<https://dnsflagday.net/2019/>>.
- [EDNS] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <<https://www.rfc-editor.org/rfc/rfc6891>>.
- [EXT-TCP] Honda, M., Nishida, Y., Raiciu, C., Greenhalgh, A., Handley, M., and H. Tokuda, "Is it still possible to extend TCP?", Proceedings of the 2011 ACM SIGCOMM

conference on Internet measurement conference - IMC '11,  
DOI 10.1145/2068816.2068834, 2011,  
<<https://doi.org/10.1145/2068816.2068834>>.

[EXTENSIBILITY]

Carpenter, B., Aboba, B., Ed., and S. Cheshire, "Design Considerations for Protocol Extensions", RFC 6709, DOI 10.17487/RFC6709, September 2012, <<https://www.rfc-editor.org/rfc/rfc6709>>.

[GREASE]

Benjamin, D., "Applying Generate Random Extensions And Sustain Extensibility (GREASE) to TLS Extensibility", RFC 8701, DOI 10.17487/RFC8701, January 2020, <<https://www.rfc-editor.org/rfc/rfc8701>>.

[HASH]

Bellovin, S. and E. Rescorla, "Deploying a New Hash Algorithm", Proceedings of NDSS '06 , 2006, <<https://www.cs.columbia.edu/~smb/papers/new-hash.pdf>>.

[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>>.

[HTTP-HEADERS]

Nottingham, M. and P-H. Kamp, "Structured Field Values for HTTP", RFC 8941, DOI 10.17487/RFC8941, February 2021, <<https://www.rfc-editor.org/rfc/rfc8941>>.

[HTTP11]

Fielding, R. T., Nottingham, M., and J. Reschke, "HTTP/1.1", Work in Progress, Internet-Draft, draft-ietf-httpbis-messaging-19, 12 September 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-messaging-19>>.

[INTOLERANCE]

Kario, H., "Re: [TLS] Thoughts on Version Intolerance", 20 July 2016, <<https://mailarchive.ietf.org/arch/msg/tls/bOJ2JQc3HjAHFFWCiNTIb0JuMZc>>.

[MPTCP]

Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, DOI 10.17487/RFC6824, January 2013, <<https://www.rfc-editor.org/rfc/rfc6824>>.



## [MPTCP-HOW-HARD]

Raiciu, C., Paasch, C., Barre, S., Ford, A., Honda, M., Duchene, F., Bonaventure, O., and M. Handley, "How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP", April 2012, <<https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/raiciu>>.

## [NEW-PROTOCOLS]

Barik, R., Welzl, M., Fairhurst, G., Elmokashfi, A., Dreibholz, T., and S. Gjessing, "On the usability of transport protocols other than TCP: A home gateway and internet path traversal study", Computer Networks Vol. 173, pp. 107211, DOI 10.1016/j.comnet.2020.107211, May 2020, <<https://doi.org/10.1016/j.comnet.2020.107211>>.

## [PATH-SIGNALS]

Hardie, T., Ed., "Transport Protocol Path Signals", RFC 8558, DOI 10.17487/RFC8558, April 2019, <<https://www.rfc-editor.org/rfc/rfc8558>>.

## [QUIC]

Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/rfc/rfc9000>>.

## [QUIC-INVARIANTS]

Thomson, M., "Version-Independent Properties of QUIC", RFC 8999, DOI 10.17487/RFC8999, May 2021, <<https://www.rfc-editor.org/rfc/rfc8999>>.

## [RAv4]

Katz, D., "IP Router Alert Option", RFC 2113, DOI 10.17487/RFC2113, February 1997, <<https://www.rfc-editor.org/rfc/rfc2113>>.

## [RAv6]

Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, DOI 10.17487/RFC2711, October 1999, <<https://www.rfc-editor.org/rfc/rfc2711>>.

## [RFC0791]

Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/rfc/rfc791>>.

## [RFC0988]

Deering, S., "Host extensions for IP multicasting", RFC 988, DOI 10.17487/RFC0988, July 1986, <<https://www.rfc-editor.org/rfc/rfc988>>.

- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, DOI 10.17487/RFC2464, December 1998, <<https://www.rfc-editor.org/rfc/rfc2464>>.
- [RFC5704] Bryant, S., Ed., Morrow, M., Ed., and IAB, "Uncoordinated Protocol Development Considered Harmful", RFC 5704, DOI 10.17487/RFC5704, November 2009, <<https://www.rfc-editor.org/rfc/rfc5704>>.
- [RRTYPE] Gustafsson, A., "Handling of Unknown DNS Resource Record (RR) Types", RFC 3597, DOI 10.17487/RFC3597, September 2003, <<https://www.rfc-editor.org/rfc/rfc3597>>.
- [SIP] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<https://www.rfc-editor.org/rfc/rfc3261>>.
- [SMTP] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <<https://www.rfc-editor.org/rfc/rfc5321>>.
- [SMTP-TLS-Reporting] Margolis, D., Brotman, A., Ramakrishnan, B., Jones, J., and M. Risher, "SMTP TLS Reporting", RFC 8460, DOI 10.17487/RFC8460, September 2018, <<https://www.rfc-editor.org/rfc/rfc8460>>.
- [SNI] Langley, A., "Accepting that other SNI name types will never work", 3 March 2016, <[https://mailarchive.ietf.org/arch/msg/tls/1t79gzNIItZd71DwwoaqcQQ\\_4Yxc](https://mailarchive.ietf.org/arch/msg/tls/1t79gzNIItZd71DwwoaqcQQ_4Yxc)>.
- [SNMPv1] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol (SNMP)", RFC 1157, DOI 10.17487/RFC1157, May 1990, <<https://www.rfc-editor.org/rfc/rfc1157>>.
- [SPF] Kitterman, S., "Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1", RFC 7208, DOI 10.17487/RFC7208, April 2014, <<https://www.rfc-editor.org/rfc/rfc7208>>.
- [SUCCESS] Thaler, D. and B. Aboba, "What Makes for a Successful Protocol?", RFC 5218, DOI 10.17487/RFC5218, July 2008, <<https://www.rfc-editor.org/rfc/rfc5218>>.

- [TCP] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/rfc/rfc793>>.
- [TFO] Cheng, Y., Chu, J., Radhakrishnan, S., and A. Jain, "TCP Fast Open", RFC 7413, DOI 10.17487/RFC7413, December 2014, <<https://www.rfc-editor.org/rfc/rfc7413>>.
- [TLS-EXT] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <<https://www.rfc-editor.org/rfc/rfc6066>>.
- [TLS12] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/rfc/rfc5246>>.
- [TLS13] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/rfc/rfc8446>>.
- [TRANSITIONS] Thaler, D., Ed., "Planning for Protocol Adoption and Subsequent Transitions", RFC 8170, DOI 10.17487/RFC8170, May 2017, <<https://www.rfc-editor.org/rfc/rfc8170>>.

## Appendix A. Examples

This appendix contains a brief study of problems in a range of Internet protocols at different layers of the stack.

### A.1. DNS

Ossified DNS code bases and systems resulted in new Resource Record Codes (RRCodes) being unusable. A new codepoint would take years of coordination between implementations and deployments before it could be relied upon. Consequently, overloading use of the TXT record was used to avoid effort and delays involved, a method used in the Sender Policy Framework [SPF] and other protocols.

It was not until after the standard mechanism for dealing with new RRCodes [RRTYPE] was considered widely deployed that new RRCodes can be safely created and used.

## A.2. HTTP

HTTP has a number of very effective extension points in addition to the aforementioned header fields. It also has some examples of extension points that are so rarely used that it is possible that they are not at all usable.

Extension points in HTTP that might be unwise to use include the extension point on each chunk in the chunked transfer coding Section 7.1 of [HTTP11], the ability to use transfer codings other than the chunked coding, and the range unit in a range request Section 14 of [HTTP].

## A.3. IP

The version field in IP was rendered useless when encapsulated over Ethernet, requiring a new ethertype with IPv6 [RFC2464], due in part to layer 2 devices making version-independent assumptions about the structure of the IPv4 header.

Protocol identifiers or codepoints that are reserved for future use can be especially problematic. Reserving values without attributing semantics to their use can result in diverse or conflicting semantics being attributed without any hope of interoperability. An example of this is the 224/3 "class E" address space in IPv4 [RFC0988]. This space was originally reserved in [RFC0791] without assigning any semantics and has since been partially reclaimed for use in multicast (224/4), but otherwise has not been successfully reclaimed for any purpose (240/4) [RFC0988].

For protocols that can use negotiation to attribute semantics to values, it is possible that unused codepoints can be reclaimed for active use, though this requires that the negotiation include all protocol participants. For something as fundamental as addressing, negotiation is difficult or even impossible, as all nodes on the network path plus potential alternative paths would need to be involved.

IP Router Alerts [RAv4][RAv6] use IP options or extension headers to indicate that data is intended for consumption by the next hop router rather than the addressed destination. In part, the deployment of router alerts was unsuccessful due to the realities of processing IP packets at line rates, combined with bad assumptions in the protocol design about these performance constraints. However, this was not exclusively down to design problems or bugs as the capability was also deliberately blocked at some routers.

#### A.4. SNMP

As a counter example, the first version of the Simple Network Management Protocol (SNMP) [SNMPv1] defines that unparseable or unauthenticated messages are simply discarded without response:

It then verifies the version number of the SNMP message. If there is a mismatch, it discards the datagram and performs no further actions.

When SNMP versions 2, 2c and 3 came along, older agents did exactly what the protocol specifies. Deployment of new versions was likely successful because the handling of newer versions was both clear and simple.

#### A.5. TCP

Extension points in TCP [TCP] have been rendered difficult to use, largely due to middlebox interactions; see [EXT-TCP].

For instance, multipath TCP [MPTCP] can only be deployed opportunistically; see [MPTCP-HOW-HARD]. As multipath TCP enables progressive enhancement of the protocol, this largely only causes the feature to not be available if the path is intolerant of the extension.

In comparison, the deployment of Fast Open [TFO] critically depends on extension capability being widely available. Though very few network paths were intolerant of the extension in absolute terms, TCP Fast Open could not be deployed as a result.

#### A.6. TLS

Transport Layer Security (TLS) [TLS12] provides examples of where a design that is objectively sound fails when incorrectly implemented. TLS provides examples of failures in protocol version negotiation and extensibility.

Version negotiation in TLS 1.2 and earlier uses the "Highest mutually supported version (HMSV)" scheme exactly as it is described in [EXTENSIBILITY]. However, clients are unable to advertise a new version without causing a non-trivial proportion of sessions to fail due to bugs in server and middlebox implementations.

Intolerance to new TLS versions is so severe [INTOLERANCE] that TLS 1.3 [TLS13] abandoned HMSV version negotiation for a new mechanism.

The server name indication (SNI) [TLS-EXT] in TLS is another excellent example of the failure of a well-designed extensibility point. SNI uses the same technique for extension that is used successfully in other parts of the TLS protocol. The original design of SNI anticipated the ability to include multiple names of different types.

SNI was originally defined with just one type of name: a domain name. No other type has ever been standardized, though several have been proposed. Despite an otherwise exemplary design, SNI is so inconsistently implemented that any hope for using the extension point it defines has been abandoned [SNI].

#### Acknowledgments

Toerless Eckert, Wes Hardaker, Mirja Kuehlewind, Eliot Lear, Mark Nottingham, and Brian Trammell made significant contributions to this document.

#### Authors' Addresses

Martin Thomson  
Mozilla

Email: [mt@lowentropy.net](mailto:mt@lowentropy.net)

Tommy Pauly  
Apple

Email: [tpauly@apple.com](mailto:tpauly@apple.com)