                  IAB Architectural and Policy Considerations for
                         Open Pluggable Edge Services

Status of this Memo

   This memo provides information for the Internet community. It does
   not specify an Internet standard of any kind. Distribution of this
   memo is unlimited.

Abstract

   This document includes comments and recommendations by the IAB on
   some architectural and policy issues related to the chartering of
   Open Pluggable Edge Services (OPES) in the IETF. OPES are services
   that would be deployed at application-level intermediaries in the
   network, for example, at a web proxy cache between the origin server
   and the client. These intermediaries would transform or filter
   content, with the explicit consent of either the content provider or
   the end user.

1. Introduction

   Open Pluggable Edge Services (OPES) are services that would be
   deployed in the network, for example, at a web proxy cache between
   the origin server and the client, that would transform or filter
   content. Examples of proposed OPES services include assembling
   personalized web pages, adding user-specific regional information to
   web pages, virus scanning, content adaptation for clients with
   limited bandwidth, language translation, and the like [OPES].

   The question of chartering OPES in the IETF ([OPESBOF1], [OPESBOF2],
   [OPESBOF3]) and the related controversy in the IETF community
   ([Carr01], [CDT01], [Morris01], [Orman01], [Routson01]) have raised
   to the fore several architectural and policy issues about robustness
   and the end-to-end integrity of data (in terms of the disparities
   between what the "origin server" makes available and what the client
   receives). In particular, questions have been raised about the
   possible requirements, for a protocol to be developed and

standardized in the IETF, for that protocol to protect the end-to-end
privacy and integrity of data. This document attempts to address
some of the architectural and policy issues that have been unresolved
in the chartering of OPES, and to come to some common recommendations
from the IAB regarding these issues.

The purpose of this document is not to recommend specific solutions
for OPES, or even to mandate specific functional requirements. This
is also not a recommendation to the IESG about whether or not OPES
should be chartered. Instead, these are recommendations on issues
that any OPES solutions standardized in the IETF should be required
to address, similar to the "Security Considerations" currently
required in IETF documents [RFC2316]. As an example, one way to
address security issues is to show that appropriate security
mechanisms have been provided in the protocol, and another way to
address security issues is to demonstrate that no security issues
apply to this particular protocol. (Note however that a blanket
sentence that "no security issues are involved" is never considered
sufficient to address security concerns in a protocol with known
security issues.)

This document will try to make our concerns underlying integrity,
privacy, and security as clear as possible. We recommend that the
IESG require that OPES documents address integrity, privacy, and
security concerns in one way or another, either directly by
demonstrating appropriate mechanisms, or by making a convincing case
that there are no integrity or privacy concerns relevant to a
particular document.

In particular, it seems unavoidable that at some point in the future
some OPES service will perform inappropriately (e.g., a virus scanner
rejecting content that does not include a virus), and some OPES
intermediary will be compromised either inadvertently or with
malicious intent. Given this, it seems necessary for the overall
architecture to help protect end-to-end data integrity by addressing,
from the beginning of the design process, the requirement of helping
end hosts to detect and respond to inappropriate behavior by OPES
intermediaries.

One of the goals of the OPES architecture must be to maintain the
robustness long cited as one of the overriding goals of the Internet
architecture [Clark88]. Given this, we recommend that the IESG
require that the OPES architecture protect end-to-end data integrity
by supporting end-host detection and response to inappropriate
behavior by OPES intermediaries. We note that in this case by
"supporting end-host detection", we are referring to supporting
detection by the humans responsible for the end hosts at the content
provider and client. We would note that many of these concerns about

the ability of end hosts to detect and respond to the inappropriate
behavior of intermediaries could be applied to the architectures for
web caches and content distribution infrastructures even without the
additional complication of OPES.

Each section of the document contains a set of IAB Considerations
that we would recommend be addressed by the OPES architecture.
Section 6 summarizes by listing all of these considerations in one
place.

In this document we try to use terminology consistent with RFC 3040
[RFC 3040] and with OPES works in progress.

2. Some history of the controversy about chartering OPES

One view on OPES has been that "OPES is deeply evil and the IETF
should stay far, far away from this hideous abomination" [ODell01].
Others have suggested that "OPES would reduce both the integrity, and
the perception of integrity, of communications over the Internet, and
would significantly increase uncertainly about what might have been
done to content as it moved through the network", and that therefore
the risks of OPES outweigh the benefits [CDT01]. This view of the
risks of OPES was revised in later email, based on the proposals from
[Carr01], "assuming that certain privacy and integrity protections
can be incorporated into the goals of the working group" [Morris01].

One issue concerns the one-party consent model. In the one-party
consent model, one of the end-nodes (that is, either the content
provider or the end user) is required to explicitly authorize the
OPES service, but authorization is not required from both parties.
[CDT01] comments that relying only on a one-party consent model in
the OPES charter "could facilitate third-party or state-sponsored
censorship of Internet content without the knowledge or consent of
end users", among other undesirable scenarios.

A natural first question is whether there is any architectural
benefit to putting specific services inside the network (e.g., at the
application-level web cache) instead of positioning all services
either at the content provider or the end user. (Note that we are
asking here whether there is architectural benefit, which is not the
same as asking if there is a business model.) Client-centric
services suggested for OPES include virus scanning, language
translation, limited client bandwidth adaptation, request filtering,
and adaptation of streaming media, and suggested server-centric
services include location-based services and personalized web pages.

It seems clear that there can indeed be significant architectural
benefit in providing some OPES services inside the network at the
application-level OPES intermediary. For example, if some content is
already available from a local or regional web cache, and the end
user requires some transformation (such as adaptation to a limited-
bandwidth path) applied to that data, providing that service at the
web cache itself can prevent the wasted bandwidth of having to
retrieve more data from the content provider, and at the same time
avoid unnecessary delays in providing the service to the end user.

A second question is whether the architectural benefits of providing
services in the middle of the network outweigh the architectural
costs, such as the potential costs concerning data integrity. This
is similar to the issues considered in RFC 3135 [RFC 3135] of the
relative costs and benefits of placing performance-enhancing proxies
(PEPs) in the middle of a network to address link-related
degradations. In the case of PEPs, the potential costs include
disabling the end-to-end use of IP layer security mechanisms;
introducing a new possible point of failure that is not under the
control of the end systems; adding increased difficulty in diagnosing
and dealing with failures; and introducing possible complications
with asymmetric routing or mobile hosts. RFC 3135 carefully
considers these possible costs, the mitigations that can be
introduced, and the cases when the benefits of performance-enhancing
proxies to the user are likely to outweigh the costs. A similar
approach could be applied to OPES services (though we do not attempt
that here).

A third question is whether an OPES service, designed primarily for a
single retrieval action, has an impact on the application layer
addressing architecture. This is related to the integrity issue
above, but is independent of whether these services are applied in
the middle of the network or at either end.

Most of this document deals with the specific issue of data integrity
with OPES services, including the goal of enabling end hosts to
detect and respond to inappropriate behavior from broken or
compromised OPES intermediaries.

We agree that one-party consent, with one of the end-hosts explicitly
authorizing the OPES service, must be a requirement for OPES to be
standardized in the IETF.

However, as we discuss in the next section of this document, we agree
with [CDT01] that the one-party consent model by itself (e.g., with
one of the end-hosts authorizing the OPES service, and the other
end-host perhaps being unaware of the OPES service) is insufficient
for protecting data integrity in the network. We also agree with

[CDT01] that, regardless of the security and authorization mechanisms
standardized for OPES in the IETF, OPES implementations could
probably be modified to circumvent these mechanisms, resulting in the
unauthorized modification of content. Many of the protocols in the
IETF could be modified for anti-social purposes - transport protocols
could be modified to evade end-to-end congestion control, routing
protocols could be modified to inject invalid routes, web proxy
caches could be used for the unauthorized modification of content
even without OPES, and so on. None of these seem like compelling
reasons not to standardize transport protocols, routing protocols,
web caching protocols, or OPES itself. In our view, it means instead
that the infrastructure needs, as much as possible, to be designed to
detect and defend itself against compromised implementations, and
misuses of protocols need to be addressed directly, each in the
appropriate venue.

Mechanisms such as digital signatures, which help users to verify for
themselves that content has not been altered, are a first step
towards the detection of the unauthorized modification of content in
the network. However, in the case of OPES, additional protection to
ensure the end-to-end integrity of data is desirable as well, for
example, to help end-users to detect cases where OPES intermediaries
were authorized to modify content, but perform inappropriate
modifications. We would note that mechanisms can *help* end-users to
detect compromised OPES intermediaries in some cases even if they do
not *guarantee* that end-users will be able to detect compromised
OPES intermediaries in all cases.

If OPES is chartered, the OPES working group will also have to
explicitly decide and document whether the OPES architecture must be
compatible with the use of end-to-end encryption by one or more ends
of an OPES-involved session. If OPES was compatible with end-to-end
encryption, this would effectively ensure that OPES boxes would be
restricted to ones that are known, trusted, explicitly addressed at
the IP layer, and authorized (by the provision of decryption keys) by
at least one of the ends. Compatibility with end-to-end encryption
would also help to prevent the widespread deployment of yet another
set of services that, to benefit from, require one to keep one's
packet contents in the clear for all to snoop.

IAB Considerations:

(2.1) One-party consent: An OPES framework standardized in the IETF
must require that the use of any OPES service be explicitly
authorized by one of the application-layer end-hosts (that is, either
the content provider or the client).

(2.2) IP-layer communications: For an OPES framework standardized in
the IETF, the OPES intermediary must be explicitly addressed at the
IP layer by the end user.

We note that (2.2) is not intended to preclude a chain of
intermediaries, with the first intermediary in the chain explicitly
addressed at the IP layer by the end user.

3. End-to-end Integrity

The proposed OPES services have several possible forms, including
server-centric services, such as the dynamic assembling of web pages,
explicitly authorized by the content provider; client-centric
services such as virus scanning or language translation explicitly
authorized by the end user to act on the response from the content
provider; and client-centric services such as privacy-based services
or content-filtering explicitly authorized by the end user to act on
the request from the end user to the content provider. We consider
the issue of the end-to-end integrity of data separately for these
different classes of services.

For each specific service, the question arises of whether it is
necessary for both the content provider and the end user to be able
to detect and respond to inappropriate behavior by OPES
intermediaries, or if it is sufficient for just one of the two end-
hosts to have this ability. We don't attempt a general answer, but
we do discuss the issues further in the sections below.

3.1. Data integrity with client-centric OPES services on responses

Why is there any concern about the end-to-end integrity of data in a
client-centric OPES service acting on a response from a content
provider? If the client requests a service such as virus scanning or
language translation, why is that of any concern to the content
provider one way or another? One answer is that one of the proper
concerns of the IETF is to design architectures that enable end-hosts
to detect and respond to inappropriate actions in the network. This
seems of particular importance for powerful devices in the network
such as OPES intermediaries, which are authorized by one of the end-
nodes to act on or transform data in the network, but other than that
are not under the direct control of that end-node.

Consider as an example the services of virus scanning or language
translation. The end user has reasonable power in detecting and
dealing with imperfect or corrupted virus scanners or language
translators that are under her direct control (e.g., on her own
machine). The end user knows exactly what program is installed, and
has direct access to the content before and after the service is

applied. The end user would have less control over similar services
offered by OPES in the network itself, where the end user's only
control might be the binary one of authorizing or not authorizing the
service. (We also note that services deployed on the end host in a
self-contained fashion, such as a local virus scanning program, are
not a service in the network, and therefore are not in the province
of the IETF one way or another.)

For a OPES service such as virus scanning or language translation,
the end user could detect a corrupted intermediary, but only through
a "black-box" approach of comparing the input with the output. This
is also imprecise and requires some effort, compared to the effort
required to detect a corrupted virus scanner installed on one's own
machine. For example, the user could retrieve the "non-OPES" version
of the content directly from the content provider, if there is a
"non-OPES" version, and compare this with the "OPES" version of the
content available from the OPES intermediary. However, in the case
of dynamic content, the "non-OPES" version of the content retrieved
by the user directly from the content provider might not necessarily
be the same as the "non-OPES" version of the content considered by
the OPES intermediary. This limited control by the end user of the
OPES service, and the limited ability of the end user to detect
imperfect or corrupted intermediaries, argues for an architecture
that helps the content provider to detect and respond to imperfect or
corrupted OPES intermediaries as well.

We consider the specific example of virus scanning, authorized by the
end user as an OPES service. One could imagine virus scanning as a
widely deployed OPES service, augmenting the virus scanning done on
the end host itself. If I ask for, say, a paper by Steve Bellovin on
security and viruses in the network, and am informed by my authorized
OPES virus-scanning service that this content does not pass the
virus-scan, there are a number of possibilities:

(1) Unknown to Steve, the content (that is, Steve's paper) contains a
    harmful virus.

(2) Steve inserted a harmful virus in the content on purpose, with
    playful or malicious intent.

(3) The OPES virus scanner can't distinguish between a true harmful
    virus, and Steve's paper about harmful viruses.

(4) My local OPES virus scanner has been hacked, with malicious
    intent, to reject all content from Steve Bellovin.

At some point, for some content, some widely-deployed implementation
of some OPES virus scanner is likely to result in problem (3), and
some OPES implementation is likely to be corrupted to result in
problem (4). Because the end user has limited control over the OPES
virus scanner, the end user also is limited in its ability to detect
problems (3) or (4) in the OPES virus scanner. In addition, the
content provider is probably the one with the strongest incentive to
detect problems (3) or (4) in the OPES virus scanner. (The content
provider generally has a strong incentive to detect problem (1) as
well.) In this case, it seems prudent that the overall OPES
architecture should be carefully designed to prevent the OPES service
of virus scanning, as authorized by the client, from unnecessarily
preventing the distribution of content that in fact does not have
viruses.

Obviously, it is not viable to propose that content providers simply
indicate that some content should be passed to the end user without
virus scanning - the point of virus scanning is for the end user to
exercise control in this regard. However, if some form of end-system
notification allows the content provider to find out that the content
is being rejected by a virus scanning service instead of being
delivered to the end user, then the content provider (Steve, in this
case) might want to inform end users that this content is known by
the content provider not to pass some OPES virus scanning services.
End users could then make their own decisions about whether or not to
retrieve that content bypassing the OPES virus scanning service,
relying on their own virus scanner or an alternate virus scanning
service for this particular content. Such end-system notification to
the content provider, if requested, cannot be enforced, and cannot be
relied upon from corrupted intermediaries, but it seems important
nevertheless.

Of course, malicious users can also use their awareness of the virus
scanning service to perfect their ability to construct malicious
viruses that can evade the virus scanning service. This will be done
anyway, with any virus scanning service, and seems like an acceptable
cost to allow content providers some protection against the vagaries
of imperfect or corrupted OPES services in the network.

Thus, for client-requested services such as virus scanning and
language translation, it is clearly desirable for the origin server
to have notification, if it requests it, that these services are
being performed on its content before the content is sent to the
client. Any such end-system notification might be accompanied by
reduced performance (in terms of overhead, delays, etc.) for the OPES
service applied to that content. But some form of end-system

notification is clearly necessary if content providers are to be able
to detect and respond to actions by OPES intermediaries that are
deemed inappropriate by the content provider.

Similarly for a client-based OPES service of language translation, it
is clearly desirable for content providers to be able to inform end
users when some content is deemed by the content provider to be
incompatible with language translation. In this case, the important
issue is not to prevent the OPES language translation from being
performed on the content, but instead to give the content provider
some mechanism to discover the language translation, and to inform
the end user (or more precisely, to inform the end user's host
computer) if the content provider believes that this language
translation is incompatible with this particular content.

IAB Considerations:

(3.1) Notification: The overall OPES framework needs to assist
content providers in detecting and responding to client-centric
actions by OPES intermediaries that are deemed inappropriate by the
content provider.

3.2. Data integrity with server-centric OPES services

What are the concerns, if any, with the end-to-end integrity of data
in a server-centric OPES service such as location-based services?
For example, CNN could authorize a location-based OPES service, where
the OPES intermediary inserts the weather report or news headline of
regional interest into the requested web page. The same issue of the
detection and response to broken or modified OPES intermediaries
occurs with server-centric OPES as with client-centric OPES services.
We only consider server-centric services on responses, as we are not
aware of any proposals for server-centric OPES services on requests
from the client to the content provider.

How are the end-nodes to detect inappropriate actions from OPES
services authorized by the content provider? The OPES service is
being performed at an OPES intermediary in the network itself, and
not under the direct control of the content provider; in particular,
the content provider might not have the ability to monitor directly
the output of the OPES intermediary. One could argue that the
content provider and server-centric OPES intermediary are part of a
single distributed application, and can be responsible on their own
for detecting and dealing with broken or modified OPES
intermediaries, without involving the end user. But this is
unconvincing, basically arguing that standardizing protocols for
performing OPES services is a network issue properly in the domain of
the IETF, but the ensuring the overall integrity of the service is a

distributed application matter, and not in the province of the IETF
at all. It would seem to us that you can't have it both ways.
Simply labeling the content provider and the OPES intermediary as
part of the same distributed application does not give the content
provider the ability to monitor the actions of the OPES intermediary.

However, if the end user receives some form of notification that
these OPES services have been provided, and has some mechanism for
receiving the "non-OPES" content from the content provider without
the OPES intermediary's modifications (if there is such a thing as a
non-OPES version of the content), then the end user is in a better
position to detect and react to inappropriate actions from
compromised or poorly-designed OPES intermediaries. Thus, it is
clear that some form of end-system notification is required to allow
the end user to detect and respond to broken or modified OPES
intermediaries. If the end user has notification of action by OPES
intermediaries, it could "veto" an OPES service simply by throwing
the OPES-modified content away. And if the client wants to talk
directly to the origin server to receive the "non-OPES" version, and
the origin server is configured to allow this, then the OPES
intermediary must be designed to permit this end-to-end
communication.

In addition to concerns about detecting and responding to faulty or
compromised OPES intermediaries, there are purely policy-based
concerns about the integrity of data. If the content provider looks
at the source IP address from the HTTP request, or tosses a coin, in
order to decide what content to provide, then that is the content
provider's business. But if there exists a "non-OPES" version of
some content available from the content provider, and also modified
versions available from OPES intermediaries, then it is important
that end users would be able to discover that they are receiving a
modified version from the network, and not the "non-OPES" version
that is also available from the content provider directly.

IAB Considerations:

(3.2) Notification: The overall OPES framework should assist end
users in detecting the behavior of OPES intermediaries, potentially
allowing them to identify imperfect or compromised intermediaries.

(3.3) Non-blocking: If there exists a "non-OPES" version of content
available from the content provider, the OPES architecture must not
prevent users from retrieving this "non-OPES" version from the
content provider.

3.3. Data integrity with client-centric OPES services on requests

   There have also been proposals for OPES services authorized by the
   client on requests from the client to the content provider. Examples
   include services that remove fields from the HTTP header for added
   privacy, and content-filtering services that filter requests based on
   the requested URL. For such services, there is still a need for end
   hosts to be assisted in detecting and responding to imperfect or
   corrupted intermediaries, but it seems less clear to what extent this
   applies to the content provider, and to what extent it applies to the
   end user that authorized the service. The requirements will probably
   have to be determined by the OPES and wider IETF communities on a
   case-by-case basis for each specific service.

4. Application Layer Addresses

   Most application layer addressing revolves around URIs, which, for
   the most part, give a structured method to refer to a single data
   entity on a remote server. URIs are universal in that, in principle,
   the same result is obtained irrespective of the location of the
   client performing the resolution.

   Practice often differs from this theory -- ad-strippers remove data
   from pages at the client end; web server farms redirect clients to
   one of several potential target machines for load-balancing or to
   give the user "localized" content.

   However, from an architectural standpoint, it is important to be
   clear about what is being done here. In all cases, URI resolution
   standards (as defined for individual URI schemes, such as HTTP) apply
   unchanged between the client and the OPES intermediary. What the
   intermediary does to fulfill the request is not material to the
   discussion, and must produce a result that is compliant with the
   applicable URI scheme definition. In this sense, the OPES
   intermediary is the "endpoint" of URI resolution.

   In client-centric OPES, the intermediary is resolving the URI on
   behalf of the client, and then applying client-requested services to
   provide a data response to the client. The client gets the data it
   wanted, but it did not carry out the URI resolution.

   In server-centric OPES, the "origin server" cedes its authority to
   the intermediary to determine what is the "appropriate" content to
   supply for a given URI. The client may well perform standard URI
   resolution, but that reaches no further than the intermediary.

   With those distinctions firmly in mind, there are two particular
   areas of concern for OPES-like services.

The first is the consideration of the effect of a series of
interactions, over time and location (i.e., not just one document
retrieval).   Potential problems include inconsistencies in intra-
and inter-document references -- depending on what content is
changed, references from one version of a document might not exist in
a modified target, etc.

The other concern is whether this leads to the creation of content
that is exclusively accessible through the use of an intermediary.
That is, there is no "non-OPES" version.   Either this should not be
allowed, or this would argue for an extension to the Internet
application layer addressing architecture.

IAB Considerations:

(4.1) URI resolution: OPES documentation must be clear in describing
these services as being applied to the result of URI resolution, not
as URI resolution itself.

(4.2) Reference validity: All proposed services must define their
impact on inter- and intra-document reference validity.

(4.3) Any services that cannot be achieved while respecting the above
two considerations may be reviewed as potential requirements for
Internet application addressing architecture extensions, but must not
be undertaken as ad hoc fixes.

5. Privacy

   Intermediaries in the middle of the network increase the number of
   locations where the privacy of an end-to-end transaction could be
   compromised.   Some of these privacy concerns apply to web caches and
   CDNs in general as well as specifically to OPES intermediaries.   It
   seems a reasonable requirement, for OPES to be chartered in the IETF,
   that the issue of providing mechanisms for end users to determine the
   privacy policies of OPES intermediaries should be addressed.   These
   mechanisms could be quite different for client-centric and server-
   centric OPES services.

   For a complex issue such as an OPES architecture, which interacts
   with protocols from other standards bodies as well as from other IETF
   working groups, it seems necessary to keep in mind the overall
   picture while, at the same time, breaking out specific parts of the
   problem to be standardized in particular working groups.   Thus, a
   requirement that the overall OPES architecture address privacy
   concerns does not necessarily mean that the mechanisms for this need
   to be developed in the IETF, or in the OPES working group (if it is
   chartered).

IAB Considerations:

(5.1) Privacy: The overall OPES framework must provide for mechanisms
for end users to determine the privacy policies of OPES
intermediaries.

6. Summary of IAB Considerations

(2.1) One-party consent: An OPES framework standardized in the IETF
must require that the use of any OPES service be explicitly
authorized by one of the application-layer end-hosts (that is, either
the content provider or the client).

(2.2) IP-layer communications: For an OPES framework standardized in
the IETF, the OPES intermediary must be explicitly addressed at the
IP layer by the end user.

(3.1) Notification: The overall OPES framework needs to assist
content providers in detecting and responding to client-centric
actions by OPES intermediaries that are deemed inappropriate by the
content provider.

(3.2) Notification: The overall OPES framework should assist end
users in detecting the behavior of OPES intermediaries, potentially
allowing them to identify imperfect or compromised intermediaries.

(3.3) Non-blocking: If there exists a "non-OPES" version of content
available from the content provider, the OPES architecture must not
prevent users from retrieving this "non-OPES" version from the
content provider.

(4.1) URI resolution: OPES documentation must be clear in describing
these services as being applied to the result of URI resolution, not
as URI resolution itself.

(4.2) Reference validity: All proposed services must define their
impact on inter- and intra-document reference validity.

(4.3) Any services that cannot be achieved while respecting the above
two considerations may be reviewed as potential requirements for
Internet application addressing architecture extensions, but must not
be undertaken as ad hoc fixes.

(5.1) Privacy: The overall OPES framework must provide for mechanisms
for end users to determine the privacy policies of OPES
intermediaries.

7. Conclusions

   This document includes comments and recommendations by the IAB on
   some architectural and policy issues related to the chartering of
   OPES in the IETF.

8. Acknowledgements

   This document has benefited from discussions with members of the IAB
   and the IESG, contributors to OPES, John Wroclawski, and others.
   However, this is a document of the IAB, and we do not claim that the
   other people listed above agree with the contents.

9. References

   [Carr01]   Wayne Carr, "Suggested OPES Requirements for Integrity,
                Privacy and Security", email to ietf-openproxy@imc.org,
                August 16, 2001. URL "http://www.imc.org/ietf-
                openproxy/mail-archive/msg00869.html".

   [CDT01]   Policy Concerns Raised by Proposed OPES Working Group
                Efforts, email to the IESG, from the Center for Democracy
                & Technology, August 3, 2001. URL
                "http://www.imc.org/ietf-openproxy/mail-
                archive/msg00828.html".

   [Clark88]   David D. Clark, The Design Philosophy of the DARPA
                Internet Protocols, SIGCOMM 1988.

   [Morris01]   John Morris, "Re: corrected - Suggested OPES
                Requirements for Integrity, Privacy and Security",
                September 28, 2001. Email to ietf-openproxy@imc.org, URL
                "http://www.imc.org/ietf-openproxy/mail-
                archive/msg00935.html".

   [ODell01]   Mike O'Dell, "OPES continuing froth...", Message-Id:
                <200107101341.JAA30276@ccr.org>, July 10, 2001, email to
                ietf@ietf.org. URL "http://www1.ietf.org/mail-
                archive/ietf/Current/msg12650.html".

   [OPES]   Open Pluggable Edge Services (OPES) Web Page,
                "http://www.ietf-opes.org/".

   [OPESBOF1]   OPES BOF, 49th IETF, December 12, 2000. Agenda:
                "http://www.ietf.org/ietf/00dec/opes-agenda.txt".
                Minutes: "http://www.ietf.cnri.reston.va.us/
                proceedings/00dec/toc.htm#P25_256".

   [OPESBOF2] OPES BOF, 50th IETF, March 9, 2001. Minutes:
               "http://www.ietf.org/proceedings/01mar/ietf50-40.htm".

   [OPESBOF3] OPES BOF, 51st IETF, August 2001. Agenda:
               "http://www.ietf.org/ietf/01aug/opes.txt". Minutes:
               "http://www.ietf.org/proceedings/01aug/minutes/OPES.HTM".

   [Orman01]   Hilarie Orman, "Data Integrity for Open Pluggable
               Services", email to ietf-openproxy@imc.org, August 15,
               2001. URL "http://www.imc.org/ietf-openproxy/mail-
               archive/msg00865.html".

   [RFC 2316] Bellovin, S., "Report of the IAB Security Architecture
               Workshop", RFC 2316, April 1998.

   [RFC2401]   Kent, S. and R. Atkinson, "Security Architecture for the
               Internet Protocol", RFC 2401, November 1998.

   [RFC 3040] Cooper, I., Melve, I. and G. Tomlinson, "Internet Web
               Replication and Caching Taxonomy", RFC 3040, January
               2001.

   [RFC 3135] Border, J., Kojo, M., Griner, J., Montenegro, G. and Z.
               Shelby, "Performance Enhancing Proxies Intended to
               Mitigate Link-Related Degradations", RFC 3135, June 2001.

   [Routson01] Joyce Routson, IETF's Edge Standards Controversy, July
               11, 2001, Stardust CDN Week. URL
               "http://www.stardust.com/cdnweek/articles/2001/07/09/
               opes.htm".

10. Security Considerations

   This document does not propose any new protocols, and therefore does
   not involve any security considerations in that sense. However,
   throughout this document there are discussions of the privacy and
   integrity issues of OPES services and the architectural requirements
   created by those issues.

11. IANA Considerations

   There are no IANA considerations regarding this document.

Authors' Addresses

Internet Architecture Board
EMail: iab@iab.org

Membership at time this document was completed:

Harald Alvestrand
Ran Atkinson
Rob Austein
Fred Baker
Steve Bellovin
Brian Carpenter
Jon Crowcroft
Leslie Daigle
Steve Deering
Sally Floyd
Geoff Huston
John Klensin
Henning Schulzrinne

12. Full Copyright Statement

   Copyright (C) The Internet Society (2002). All Rights Reserved.

   This document and translations of it may be copied and furnished to
   others, and derivative works that comment on or otherwise explain it
   or assist in its implementation may be prepared, copied, published
   and distributed, in whole or in part, without restriction of any
   kind, provided that the above copyright notice and this paragraph are
   included on all such copies and derivative works. However, this
   document itself may not be modified in any way, such as by removing
   the copyright notice or references to the Internet Society or other
   Internet organizations, except as needed for the purpose of
   developing Internet standards in which case the procedures for
   copyrights defined in the Internet Standards process must be
   followed, or as required to translate it into languages other than
   English.

   The limited permissions granted above are perpetual and will not be
   revoked by the Internet Society or its successors or assigns.

   This document and the information contained herein is provided on an
   "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING
   TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
   BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION
   HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
   MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

   Funding for the RFC Editor function is currently provided by the
   Internet Society.