Open Pluggable Edge Services
Internet-Draft
Expires: February 25, 2004

A. Barbir
Nortel Networks
A. Rousskov
The Measurement Factory
August 27, 2003

OPES Treatment of IAB Considerations
draft-ietf-opes-iab-01

Status of this Memo

Copyright Notice

Abstract

IETF Internet Architecture Board (IAB) expressed nine architecture-level considerations when Open Pluggable Edge Services (OPES) working group was being chartered at the IETF. The working group was chartered under the condition that IAB considerations were addressed by the group. This document describes how OPES addresses those considerations.

Table of Contents

1. Introduction

The Open Pluggable Edge Services (OPES) architecture [I-D.ietf-opes-architecture], enables cooperative application services (OPES services) between a data provider, a data consumer, and zero or more OPES processors. The application services under consideration analyze and possibly transform application-level messages exchanged between the data provider and the data consumer.

In the process of chartering OPES, the IAB made recommendations on issues that OPES solutions should be required to address. These recommendations were formulated in the form of specific IAB considerations [RFC3238]. IAB emphasized that its considerations did not recommend specific solutions and did not mandate specific functional requirements. Addressing an IAB consideration may involve showing appropriate protocol mechanisms or demonstrating that the issue does not apply. Addressing a consideration does not necessarily mean supporting technology implied by the consideration wording.

The primary goal of this document is to show that all IAB considerations are addressed by OPES, to the extent those considerations can be addressed by an IETF working group. Limitations of OPES working group ability to address certain aspects of IAB considerations are explicitly documented.

There are nine IAB considerations [RFC3238] that OPES has to address. In the core of this document are the corresponding nine "Consideration" sections. For each IAB consideration, its section contains general discussion as well as references to specific OPES mechanisms relevant to the consideration.

2. Terminology

This document does not introduce any new terminology but uses terminology from other OPES documents it quotes.

3. Consideration (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)."[RFC3238]

OPES architecture requires that "OPES processors MUST be consented to by either the data consumer or data provider application" [I-D.ietf-opes-architecture]. This requirement alone cannot prevent consent-less introduction of OPES processors. In [I-D.ietf-opes-end-comm], the OPES architecture enables concerned parties to detect unwanted OPES processors by examining OPES traces. The use of traces in OPES is mandatory.

Tracing mechanism on its own is unable to detect processors that are in violation of OPES specifications. Examples include OPES processors operating in stealth mode. However, the OPES architecture allows the use of content signature to verify the authenticity of performed adaptations. Content signatures is a strong but expensive mechanism that can detect any modifications of signed content provided the content provider is willing to sign the data and the client is willing to either check the signature or relay received content to content provider for signature verification.

OPES adaptations may include copying and other forms of non-modifying access to content. These kinds of adaptations cannot be detected by the above mentioned mechanisms. Thus, "passive" OPES processors can operate without consent. If presence of such processors is a concern, content encryption can be used. A passive processor is no different from a proxy or intermediary operating outside of OPES framework. No OPES mechanism can prevent non-modifying access to content.

4. Consideration (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."[RFC3238]

OPES architecture requires that "OPES processors MUST be addressable at the IP layer by the end user (data consumer application)" [I-D.ietf-opes-architecture]. IAB and the architecture draft mention an important exception: addressing the first OPES processor in a chain of processors is sufficient. That is, a chain of OPES processors is viewed as a single OPES "system" at the address of the first chain element.

The notion of a chain is not strictly defined by IAB. For the purpose of addressing this consideration, we consider a group of OPES processors working on a given application transaction. Such a group would necessarily form a single processing chain, with a single "exit" OPES processor (the processor that adapted the given message last). OPES architecture essentially requires that last OPES processor to be explicitly addressable at the IP layer by the end user. Note that chain formation, including its exit point may depend on an application message and other dynamic factors such as time of day or system load.

Furthermore, if OPES processing is an internal processing step at data consumer or provider side, then the last OPES processor may reside in a private address space of the side's network and may not be explicitly addressable. In such situations, the processing side must designate an addressable point on the same processing chain. That designated point may not be, strictly speaking, an OPES processor, but it will suffice as such as far as IAB considerations are concerned -- the other side will be able to address it explicitly at the IP layer and it will represent the OPES processing chain to the outside world.

Designating an addressable processing point avoids the conflict between narrow interpretation of IAB consideration and real system designs: It is irrational to expect a content provider to provide access to internal hosts participating in content generation, whether OPES processors are involved or not. Moreover, providing such access would serve little practical purpose because internal OPES processors are not likely to be able to answer any end user queries, being completely out of content generation context. For example, an OPES processor adding customer-specific information to XML pages may not understand or be aware of any final HTML content that the end user receives and may not be able to map end user request to any internal user identification. Since OPES requires the end of the message

processing chain to be addressable, the conflict does not exist -- OPES places no requirements on the internal architecture of data producer systems while requiring the entire OPES-related content production "system" to be addressable at the IP layer.

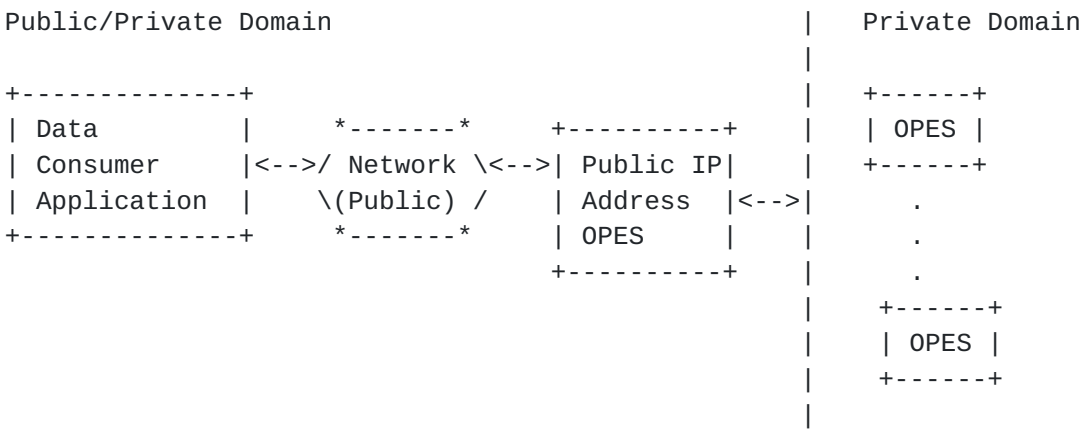

Figure 1

(XXX: should we add a picture showing internal and external OPES intermediaries? more pictures showing other OPES layouts? Move to architecture draft?)

5. Notification Considerations

This section discusses how OPES framework addresses IAB Notification considerations 3.1 and 3.2.

5.1 Notification versus trace

Before specific considerations are discussed, the relationship between IAB notifications and OPES tracing has to be explained. OPES framework concentrates on tracing rather than notification. The tracing specification [I-D.ietf-opes-end-comm] defines "OPES trace" as "application message information about OPES entities that adapted that message" and "OPES tracing" as "the process of including, manipulating, and interpreting an OPES trace" (XXX: keep these in sync). Thus, OPES trace follows the application message it traces. The trace is for the recipient of the application message. Traces are implemented as extensions of application protocols being adapted and traced.

As opposed to an OPES trace, provider notification (as implied by IAB) notifies the sender of the application message rather than the recipient. Thus, notifications propagate in the opposite direction of traces. Supporting notifications directly would require a new protocol. Figure XXX illustrates the differences between a trace and notification from a single application message point of view.

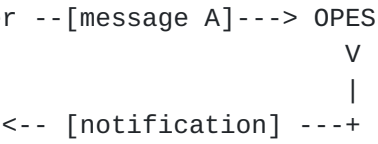

Figure 2

Since notifications cannot be piggy-backed to application messages, they create new messages and may at least double the number of messages the sender has to process (more if several intermediaries on the message path emit notifications). Moreover, associating notifications with application messages may require duplicating application message information in notifications and/or maintaining a sender state until notification is received, increasing performance overhead of notifications. These concerns call for optional notification, with a special protocol to enable notifications when needed.

The level of available details in notifications versus provider interest in supporting notification is another concern. Experience shows that content providers often require very detailed information about user actions to be interested in notifications at all. For

example, Hit Metering protocol [XXX] has been designed to supply content providers with proxy cache hit counts, in an effort to reduce cache busting behavior which was caused by content providers desire to get accurate site "access counts". However, the Hit Metering protocol is currently not widely deployed because the protocol does not supply content providers with information such as client IP addresses, browser versions, or cookies.

Hit Metering experience is relevant because Hit Metering protocol was designed to do for HTTP caching intermediaries what OPES notifications are meant to do for OPES intermediaries. Performance requirements call for state reduction via aggregation of notifications while provider preferences call for state preservation or duplication. Achieving the right balance when two sides belong to different organizations and have different optimization priorities may be impossible.

Thus, instead of explicitly supporting notifications on a protocol level, OPES concentrates on tracing facilities and supports notifications indirectly, using those tracing facilities. In other words, the IAB choice of "Notification" label is interpreted as "Notification assistance" (i.e. making notifications meaningful) and is not interpreted as a "Notification protocol".

5.2 An example of an OPES trace for HTTP

The example below illustrates adaptations done to HTTP request at an OPES intermediary operated by the client ISP. Both original (as sent by an end user) and adapted (as received by the origin web server) requests are shown. The primary adaptation is the modification of HTTP "Accept" header. The secondary adaptation is the addition of an "OPES-Via" HTTP extension header.

```
GET /pub/WWW/ HTTP/1.1
Host: www.w3.org
Accept: text/plain
```

Figure 3

... may be adapted by an ISP OPES system to become:

```
GET /pub/WWW/ HTTP/1.1
Host: www.w3.org
Accept: text/plain; q=0.5, text/html, text/x-dvi; q=0.8
OPES-Via: http://www.isp-example.com/opes/?client-hash=1234567
```

Figure 4

The example below illustrates adaptations done to HTTP response at an OPES intermediary operated by a Content Distribution Network (CDN). Both original (as sent by the origin web server) and adapted (as received by the end user) responses are shown. The primary adaptation is the conversion from HTML markup to plain text. The secondary adaptation is the addition of an "OPES-Via" HTTP extension header.

```
HTTP/1.1 200 OK
Content-Length: 12345
Content-Encoding: text/html

<html><head><h1>Available Documenta...
```

Figure 5

... may be adapted by a CDN OPES system to become:

```
HTTP/1.1 200 OK
Content-Length: 2345
Content-Encoding: text/plain
OPES-Via: http://www.cdn-example.com/opes/?site=7654321&service=h2t

AVAILABLE DOCUMENTA...
```

Figure 6

In the above examples, "OPES-Via" header values contain URLs that may point to OPES-specific documents such as description of the OPES operator and its privacy policy. Those documents may be parameterized to allow for customizations specific to the transaction being traced (e.g., client or even transaction identifier may be used to provide more information about performed adaptations). Traced OPES URLs may be later used to request OPES bypass. (XXX: OPES specs will need to define OPES-Via format and semantics)

5.3 Consideration (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."[RFC3238]

OPES tracing mechanisms assist content providers in detecting client-centric actions by OPES intermediaries. Specifically, a compliant OPES intermediary or system notifies a content provider of its presence by including its tracing information in the application protocol requests. An OPES system MUST leave its trace (XXX quote tracing draft) [I-D.ietf-opes-end-comm]. Detection assistance has

its limitations. Some OPES intermediaries may work exclusively on responses and may not have a chance to trace the request. Moreover, some application protocols may not have explicit requests (e.g., a content push service).

OPES tracing mechanisms assist content providers in responding to client-centric actions by OPES intermediaries. Specifically, OPES traces MUST include identification of OPES systems and SHOULD include a list of adaptation actions performed on provider's content. This tracing information may be included in the application request. Usually, however, this information will be included in the application response, an adapted version of which does not reach the content provider. If OPES end points cooperate, then notification can be assisted with traces. Content providers that suspect or experience difficulties can do any of the following:

- Check whether requests they receive pass through OPES intermediaries. Presence of OPES tracing info will determine that. This check is only possible for request/response protocols. For other protocols (e.g., broadcast or push), the provider would have to assume that OPES intermediaries are involved until proven otherwise.

- If OPES intermediaries are suspected, request OPES traces from potentially affected user(s). The trace will be a part of the application message received by the user software. If users cooperate, the provider(s) have all the information they need. If users do not cooperate, the provider(s) cannot do much about it (they might be able to deny service to uncooperative users in some cases).

- Some traces may indicate that more information is available by accessing certain resources on the specified OPES intermediary or elsewhere. Content providers may query for more information in that case.

- If everything else fails, providers can enforce no-adaptation policy using appropriate OPES bypass mechanisms and/or end-to-end encryption mechanisms.

OPES detection and response assistance is limited to application protocols with support for tracing extensions. For example, HTTP [RFC2616] has such support while DNS over UDP does not.

(XXX: should we prohibit adaptation of application protocols that do not allow for tracing?)

5.4 Consideration (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."[RFC3238]

OPES tracing mechanisms assist end users in detecting OPES intermediaries. Specifically, a compliant OPES intermediary or system notifies an end user of its presence by including its tracing information in the application protocol messages sent to the client. An OPES intermediary MUST leave its trace (XXX quote tracing draft) [I-D.ietf-opes-end-comm]. Detection assistance has its limitations. Some OPES intermediaries may work exclusively on requests and may not have a chance to trace the response. Moreover, some application protocols may not have explicit responses (e.g., event logging service).

OPES detection assistance is limited to application protocols with support for tracing extensions. For example, HTTP [RFC2616] has such support while DNS over UDP does not.

(XXX: should we prohibit adaptation of application protocols that do not allow for tracing?)

5.5 Consideration (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."[RFC3238]

OPES intermediaries MUST support a bypass feature (XXX quote bypass draft) [I-D.ietf-opes-end-comm]. If an application message includes bypass instructions and an OPES intermediary is not configured to ignore them, the matching OPES intermediary will not process the message. An intermediary may be configured to ignore bypass instructions only if no non-OPES version of content is available. Bypass may generate content errors since some OPES services may be essential but may not be configured as such.

Bypass support has limitations similar to the two notification-related considerations above. (XXX: but it is possible to instruct all OPES intermediaries to bypass an application message without knowing all OPES intermediaries IDs).

(XXX: Ideally, this section need to be polished further -- if there is no non-OPES version of the content, most IAB considerations probably do not apply because there is really no adaptation, only

creation of content; and we should not restrict content creation.)

6. Consideration (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."[RFC3238]

"OPES Scenarios and Use Cases" specification [I-D.ietf-opes-scenarios] documents content adaptations that are in scope of the OPES framework (XXX provide a quote). These adaptations do not include URI resolution (XXX check). In some environments, it is technically possible to adapt URIs (and other kinds of identifiers or addresses) using documented OPES mechanisms.

7. Consideration (4.2) Reference validity

"All proposed services must define their impact on inter- and intra-document reference validity."[RFC3238]

OPES working group does not propose adaptation services. However, OPES tracing requirements include identification of OPES intermediaries and services (for details, see "Notification" consideration sections in this document). It is required that provided identification can be used to locate information about the OPES intermediaries, including the description of impact on reference validity (XXX quote tracing draft) [I-D.ietf-opes-end-comm].

8. Consideration (4.3) Addressing extensions

"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."[RFC3238]

OPES framework does not contain ad hoc fixes. This and other OPES documents should be sufficient to inform service creators of IAB considerations. If a service does URI resolution or silently affects document reference validity, the authors are requested to review service impact on Internet application addressing architecture and work within IETF on potential extension requirements. Such actions would be outside of the current OPES framework.

9. Consideration (5.1) Privacy

"The overall OPES framework must provide for mechanisms for end users to determine the privacy policies of OPES intermediaries."[RFC3238]

OPES tracing mechanisms allow end users to identify OPES intermediaries (for details, see "Notification" consideration sections in this document). It is required that provided identification can be used to locate information about the OPES intermediaries, including their privacy policies.

The terms "privacy" and "privacy policy" are not defined in this context (by IAB or OPES working group). OPES tracing mechanisms allow end users and content providers to identify OPES intermediaries. It is believed that once an intermediary is identified, it would be possible to locate relevant information about that intermediary, including information relevant to requesters perception of privacy policy or reference validity. (XXX: should we move this paragraph into a separate section and expand it? one the other hand, it is probably the job of the architecture draft to define these things so that we can refer to them from here.)

10. Security Considerations

XXX.

11. Compliance

This document may be perceived as a proof of OPES compliance with IAB implied recommendations. However, this document does not introduce any compliance subjects. Compliance of OPES implementations is defined in other OPES documents discussed above.

12. To-do

security section: Does this document have any original security matters worth documenting?

normative IDs: To be normative, OPES Internet-Drafts must be replaced with corresponding RFCs when the latter are published.

architecture draft: Should architecture draft talk about external/ internal OPES intermediaries, OPES systems, and privacy policies? Should this document be limited to a compilation of references from other OPES drafts, or should we introduce/discuss new concepts here?

Appendix A. Change Log

Internal WG revision control ID: $Id: iab-cons.xml,v 1.19 2003/08/28 03:48:32 rousskov Exp $

head-sid15

* Added a figure showing a chain of internal OPES intermediaries behind a public IP address. Needs more work. More cases?

head-sid14

* Rewrote the Introduction to the IP addressing consideration. Do NOT explain how IAB considerations, if interpreted literary, do not satisfy important real-world constraints. Instead, use the "chain of OPES intermediaries" exception introduced by IAB itself to show that OPES architecture addresses IAB concerns as long as the "chain" is defined/formed for a given application message rather than being a statically configured application routing table of sorts. IAB had to add the "chain" exception to cover one of the most obvious real-world usage scenario. We use the very same exception to cover all usage scenarios we care about.

* Polished text explaining the differences between tracing and notification mechanisms.

* Added examples of OPES/HTTP traces.

* Be careful not to imply that all OPES intermediaries must obey bypass instructions. Bypass should be ignored when no non-OPES version of the content exists. Ideally, this may need to be polished further -- if there is no non-OPES version of the content, most IAB considerations probably do not apply because there is really no adaptation, only creation of content (and we should not restrict content creation).

* Added references to OPES "Communications" draft [I-D.ietf-opes-end-comm].

head-sid9

* Polished to meet new xml2rfc strict requirements.

head-sid8

* Added unpolished meat for all nine considerations.

* Added Abbie Barbir as an author.

head-sid7

* Initial revision

Normative References

[I-D.ietf-opes-end-comm]
Barbir, A., "OPES processor and end points communications", draft-ietf-opes-end-comm-00 (work in progress), June 2003.

[I-D.ietf-opes-architecture]
Barbir, A., "An Architecture for Open Pluggable Edge Services (OPES)", draft-ietf-opes-architecture-04 (work in progress), December 2002.

[I-D.ietf-opes-scenarios]
Barbir, A., "OPES Use Cases and Deployment Scenarios", draft-ietf-opes-scenarios-01 (work in progress), August 2002.

[RFC3238] Floyd, S. and L. Daigle, "IAB Architectural and Policy Considerations for Open Pluggable Edge Services", RFC 3238, January 2002.

Informative References

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Nielsen, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

Authors' Addresses

Abbie Barbir
Nortel Networks
3500 Carling Avenue
Nepean, Ontario
CA

Phone: +1 613 763 5229
EMail: abbieb@nortelnetworks.com

Alex Rousskov
The Measurement Factory

EMail: rousskov@measurement-factory.com
URI: http://www.measurement-factory.com/

Intellectual Property Statement

Full Copyright Statement

Acknowledgement

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