Network Working Group
Request for Comments: 3752
Category: Informational

A. Barbir
Nortel Networks
E. Burger
Brooktrout Technology, Inc.
R. Chen
AT&T Labs
S. McHenry
Individual Contributor
H. Orman
Purple Streak Development
R. Penno
Nortel Networks

April 2004

Open Pluggable Edge Services (OPES) Use Cases and Deployment Scenarios

Status of this Memo

Copyright Notice

Abstract

This memo provides a discussion of use cases and deployment scenarios for Open Pluggable Edge Services (OPES). The work examines services that could be performed to requests and/or responses.

Table of Contents

1. Introduction

The Open Pluggable Edge Services (OPES) [1] 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. The execution of such services is governed by a set of filtering rules installed on the OPES processor.

The rules enforcement can trigger the execution of service applications local to the OPES processor. Alternatively, the OPES processor can distribute the responsibility of service execution by communicating and collaborating with one or more remote callout [6] servers.

The document presents examples of services in which Open Pluggable Edge Services (OPES) would be useful. There are different types of OPES services: services that modify requests, services that modify responses, and a special case of the latter, services that create responses.

The work also examines various deployment scenarios of OPES services. The two main deployment scenarios, as described by the OPES architecture [1], are surrogate overlays and delegate overlays. Surrogate overlays act on behalf of data provider applications, while delegate overlays act on behalf of data consumer applications. The document also describes combined surrogate and delegate overlays, as one might find within an enterprise deployment.

The document is organized as follows: Section 2 discusses the various types of OPES services. Section 3 introduces OPES deployment scenarios. Section 4 discusses failure cases and service notification. Section 5 discusses security considerations.

The IAB has expressed architectural and policy concerns [2] about OPES. Other OPES documents that may be relevant are, "OPES Service Authorization and Enforcement Requirements" [5]. See references [3, 4] for recommended background reading.

2. Types of OPES services

OPES scenarios involve services that can be performed on requests for data and/or responses. OPES services can be classified into three categories: services performed on requests, services performed on responses, and services creating responses. In Figure 1, the four service activation points for an OPES processor are depicted. The data dispatcher examines OPES rules, enforces policies, and invokes service applications (if applicable) at each service activation point.

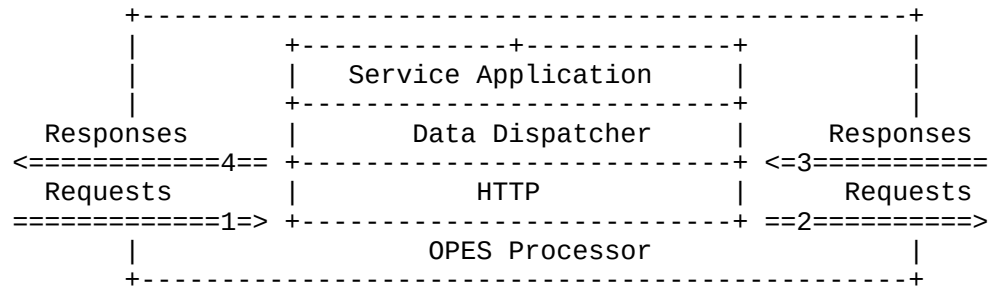

Figure 1: Service Activation Points

2.1. Services performed on requests

An OPES service performed on HTTP requests may occur when a request arrives at an OPES processor (point 1) or when it is about to leave the OPES processor (point 2).

The services performed on requests can further be divided into two cases: those that intend to modify requests and those that do not.

2.1.1. Services intending to modify requests

An OPES processor may modify a service request on behalf of the data consumer for various reasons, such as:

- Owner of a Web access device might need control over what kind of Web content can be accessed with the device, parental control for example.

- Organization may restrict or redirect access to certain web

services based on various criteria such as time of the day or the employee access privileges.

- Hiding the data consumer's identity, user agent, or referrer.
- Adding user preferences or device profile to the service request to get personalized or adapted services.
- Blocking or redirecting a service request due to a corporate policy.

An OPES processor may also modify a service request on behalf of the data provider in several ways, such as:

- Redirecting the request to a different server to reduce the server work load.
- Redirecting image requests to improve access time.

2.1.2. Services *not* intending to modify requests

An OPES processor may invoke useful service applications that do not modify the user requests. Examples include:

- Administrative functions for the data provider, such as service monitoring or usage tracking for billing purposes.
- Useful services for the data consumer, such as user profiling (with the user's consent) for service adaptation later on.

2.2. Services performed on responses

An OPES service performed on HTTP responses may occur when a response arrives at an OPES processor (point 3) or when it is about to leave the OPES processor (point 4). In the case of a caching proxy, the former service may be an encoding operation before the content is stored in the cache, while the latter may be a decoding operation before the content is returned to the data consumer.

The services performed on responses can further be divided into two cases: those that intend to modify responses and those that do not.

2.2.1. Services intending to modify responses

There are several reasons why responses from the data providers might be modified before delivery to the data consumer:

- Content adaptation: the data provider may not have all the device

profiles and templates necessary to transcode the original content into a format appropriate for mobile devices of limited screen size and display capabilities.

- Language translation: the data provider may not have all the translation capabilities needed to deliver the same content in multiple languages to various areas around the world. An OPES processor may perform the language translation or it may invoke different callout servers to perform different language translation tasks.

2.2.2. Services *not* intending to modify responses

An OPES service may be performed on the responses without modifying them. Examples include:

- Logging/Monitoring: Each response may be examined and recorded for monitoring or debugging purposes.

- Accounting: An OPES processor may record the usage data (time and space) of each service request for billing purposes.

2.3. Services creating responses

Services creating responses may include OPES services that dynamically assemble web pages based on the context of the data consumer application.

Consider a content provider offering web pages that include a local weather forecast based on the requestor's preferences. The OPES service could analyze received requests, identify associated user preferences, select appropriate templates, insert the corresponding local weather forecasts, and would then deliver the content to the requestor. Note that the OPES processor may perform the tasks with or without direct access to the weather data. For example, the service could use locally cached weather data or it could simply embed a URL pointing to another server that holds the latest local weather forecast information.

3. OPES deployment scenarios

OPES entities can be deployed over an overlay network that supports the provisioning of data services in a distributed manner. Overlay networks are an abstraction that creates a virtual network of connected devices layered on an existing underlying IP networks in order to perform application level services.

The use of overlay networks creates virtual networks that via OPES

entities enables the necessary network infrastructure to provide better services for data consumer and provider applications. At the application level, the resulting overlay networks are termed OPES Services Networks.

There are two parties that are interested in the services that are offered by OPES entities, the delegate and the surrogate. Delegates are authorized agents that act on behalf of data consumers. Surrogates are authorized agents that act on behalf of data providers.

All parties that are involved in enforcing policies must communicate the policies to the parties that are involved. These parties are trusted to adhere to the communicated policies.

In order to delegate fine-grained trust, the parties must convey policy information by implicit contract, by a setup protocol, by a dynamic negotiation protocol, or in-line with application data headers.

3.1. Surrogate Overlays

A surrogate overlay is a specific type of OPES service network, which is delegated the authority to provide data services on behalf of one or more origin servers. Such services include, but are not limited to, dynamic assembling of web pages, watermarking, and content adaptation.

The elements of surrogate overlays act on behalf of origin severs and logically belong to the authoritative domain of the respective origin servers. The scenario is depicted in Figure 2.

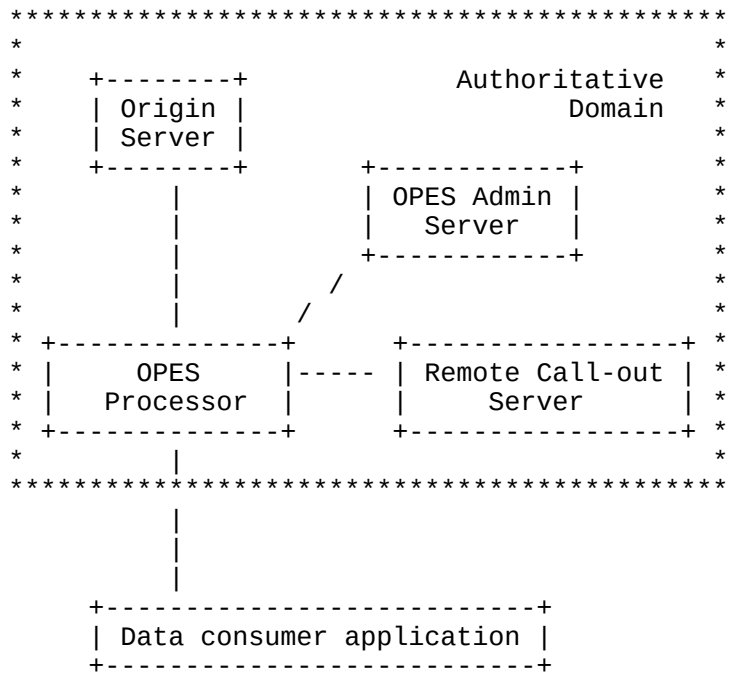

Figure 2: Authoritative Domains for Surrogate Overlays

3.2. Delegate Overlays

A delegate overlay is a specific type of OPES service network, which is delegated the authority to provide data services on behalf of one or more data consumer applications.

Delegate overlays provide services that would otherwise be performed by the data consumer applications. Such services include, but are not limited to, virus scanning and content filtering.

The elements of delegate overlays logically belong to the authoritative domain of the respective data consumer application. The situation is illustrated in Figure 3.

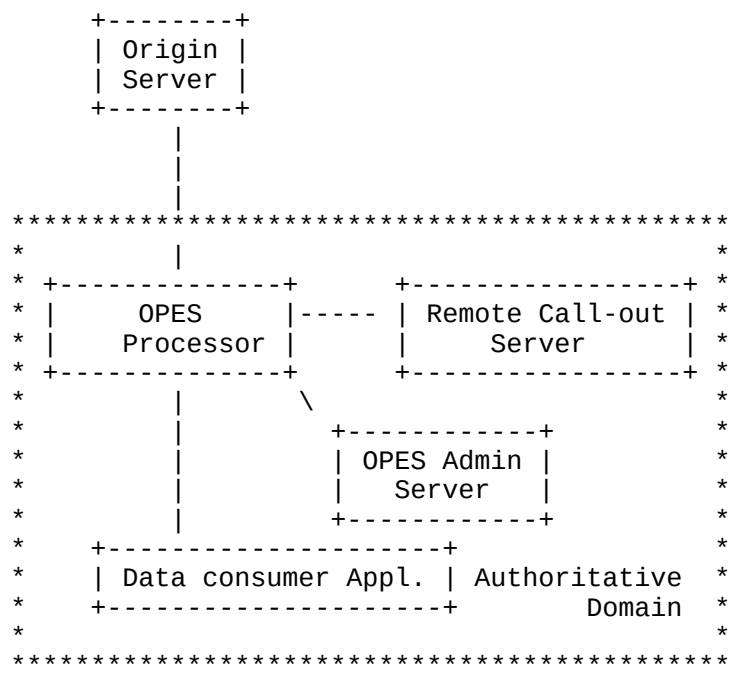

Figure 3: Authoritative Domains for Delegate Overlays

3.3. Enterprise environment

Deployment of OPES services in an enterprise environment is unique in several ways:

- Both data providers and data consumers are in the same administrative domain and trust domain. This implies that the logical OPES administrator has the authority to enforce corporate policies on all data providers, data consumers, and OPES entities.

- In the case when a callout server outside the corporate firewall is invoked for services (such as language translation) that cannot be performed inside the corporation, care must be taken to guarantee a secure communication channel between the callout server and corporate OPES entities. The callout server must also adhere to all corporate security policies for the services authorized.

3.4. Callout Servers

In some cases the deployment of OPES services can benefit from the use of callout servers that could distribute the workload of OPES processors or to contract specialized services from other OPES providers.

In general, operations such as virus scanning that operate on large objects are better handled through the use of a dedicated callout server that is better designed to perform the memory intensive task than what an OPES processor could handle.

3.5. Chaining of OPES data filters and callout servers

OPES data processors can be "chained" in two dimensions: along the content path or along the callout path. In the latter case, the callout servers can themselves be organized in series for handling requests. Any content that is touched by more than one data processor or more than one callout server has been handled by a "chain".

NOTE: Chaining of callout servers is deferred from version 1 of the Protocol. The discussion of chaining is included here for completeness.

3.5.1. Chaining along the content path

An OPES provider may have assigned OPES services to a set of processors arranged in series. All content might move through the series, and if the content matches the rules for a processor, it is subjected to the service. In this way, the content can be enhanced by several services. This kind of chaining can be successful if the services are relatively independent. For example, the content might be assembled by a service early in the chain and then further decorated by a later service.

3.5.2. Chaining along the callout path

Alternatively, an OPES data processor might act as a content-level switch in a cluster of other data processors and callout servers.

The first stage might develop a processing schedule for the content and direct it to other OPES data processors and/or callout servers. For example, OPES processor A might handle all services assembling content, OPES processor B might handle all services involving URL translation, and OPES processor C might handle all content security services. The first processor would determine that processors A and

C were needed for a particular content object, and it would direct the content to those processors. In turn, the processors might use several callout servers to accomplish the task.

4. Failure cases and service notification

These are illustrative cases where information about OPES processing can help endpoint users determine where and why content modifications are being performed.

- Content provider uses an OPES data processor to enhance content based only on context local to the provider. The local context might be time of day, local URL, or available advertising, for example. The content provider might find OPES logging to be sufficient for debugging any problems in this case. However, the content provider might also try direct probing by issuing a request for the content and examining headers related to tracing. If unexpected parameters show up in the trace headers, the content provider's administrator can use these to correct the OPES rules or detect the presence of an unexpected OPES processor in the content path.

- Content provider uses an OPES data processor to enhance content based on context related to the requestor. The requestor may notice that his requests do not elicit the same response as another requestor. He may, for example, get an error message. If he believes there is a configuration error on the OPES data processor, he will need to provide information to the administrator of it. If the information includes "OPES service access control, action: blocked", for example, he can inquire about the circumstances that will allow him to be added to the access control list. In another example, if he sees a picture unrelated to the surrounding text, and if the tracing shows "OPES service choose picture, action: insert 640x480 weather.gif", he might complain that the OPES service does not properly recognize his geographic location and inserts the wrong weather map. In any case, if the information is forwarded to the content provider, the problem may be fixed.

- End user has OPES processor available as part of his network access environment. The end user may have selected "translate English to Spanish" as an OPES service. If he sees "OPES service language translation, action: destination language not supported, no action", then he may inquire of the OPES service provider about what languages are supported by the package. If the end user feels that the source language is not properly represented by the

provider, resulting in inability for the service to operate, he (or the language service provider) can contact the content provider.

- If the content provider gets complaints from users about the translation service and feels that the problem is not in the content but in the service, he may recommend that the service not be applied to his pages. He can do that through content headers, for example, with the notation "No OPES service #8D3298EB" or "No OPES class language translation".

- End user's ISP or enterprise uses OPES to control user access based on user profiles. The end user can see that the OPES services are being applied by his ISP, but he cannot control them. If he feels that the transformations bowdlerize the content he can complain to the provider organization.

- The content provider or end user relies on a content distribution network and OPES is used within that network. OPES may be authorized by either the content provider, end user, or both. The content provider may suspect that his access control rules are not being applied properly, for example. He may ask for notification on all accesses to his content through a log. This request and the logfile are outside the OPES architecture; there are security implications for the request, the response, and the resources used by the logfile.

5. Security Considerations

The document presents usage scenarios and deployment cases. Issues related to the overall security of OPES entities are given in [1].

6. Informative References

[1] A. Barbir et al., "An Architecture for Open Pluggable Edge Services (OPES)", Work in Progress, July 2002.

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

[3] Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J. and S. Waldbusser, "Terminology for Policy-Based Management", RFC 3198, November 2001.

[4] 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.

[5] OPES Working Group, "OPES Service Authorization and Enforcement Requirements", Work in Progress, May 2002.

[6] Beck, A., et al., "Requirements for OPES Callout Protocols", Work in Progress, July 2002.

7. Acknowledgements

The authors would like to thank the participants of the OPES WG for their comments on this document.

8. Authors' Addresses

Abbie Barbir
Nortel Networks
3500 Carling Avenue
Nepean, Ontario K2H 8E9
Canada

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

Eric W. Burger
Brooktrout Technology, Inc.
18 Keewaydin Dr.
Salem, NH 03079

EMail: e.burger@ieee.org

Yih-Farn Robin Chen
AT&T Labs - Research
180 Park Avenue
Florham Park, NJ 07932
US

Phone: +1 973 360 8653
EMail: chen@research.att.com

Stephen McHenry
305 Vineyard Town Center, #251
Morgan Hill, CA 95037
US

Phone: +1 408 683 2700
EMail: stephen@mchenry.net

Hilarie Orman
Purple Streak Development

EMail: ho@alum.mit.edu

Reinaldo Penno
Nortel Networks
600 Technology Park Drive
Billerica, MA 01803
US

EMail: rpenno@nortelnetworks.com

9. Full Copyright Statement

Intellectual Property

Acknowledgement

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