Client Side Content Screening Framework Architecture

Approved Version 1.0 – 14 Jun 2007

Open Mobile Alliance

OMA-AD-Client_Side_CS_FW-V1_0-20070614-A

Contents

Figures

Tables

1. Scope (Informative)

The scope of the client side content screening framework architecture is to define the architecture of the content screening framework in mobile terminals to screen malicious content. The architecture described in this document is restricted to client/terminal deployments only. Other possible deployments of this architecture (e.g. proxy, server, etc.) are not within the scope of this architecture. Formal definition of malicious content is not in the scope of this architecture and thus is left to each implementation. The architecture described in this document includes a Scan Engine functional component. It is recognized that such a component is likely to have an associated database of profiles to encapsulate information such as its processing rules. However, such a database and its profiles are not in the scope of this architecture. This architecture is based on the requirements listed in [**CSCSF-RD-v1**].

2. References

2.1 Normative References

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[PRIVACY] "OMA Privacy Requirements for Mobile Services", Version 1.0, Open Mobile Alliance™, OMA-RD_Privacy-V1_0, URL:http://www.openmobilealliance.org/

[CSCSF-RD-v1] "OMA Client Side Content Screening Framework Requirements", Version 1.0, Open Mobile Alliance™, OMA-RD-Client_Side_CS_FW-V1_0, URL:http://www.openmobilealliance.org/

2.2 Informative References

[OMA-DICT] "Dictionary for OMA Specifications", Version 2.3, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_3, URL:http://www.openmobilealliance.org/

[ARCH-OSE] "OMA Service Environment", Version 1.0, Open Mobile Alliance™, OMA-Service-Environment-V1_0, URL:http://www.openmobilealliance.org/

[ARCH-PRINC] "OMA Architecture Principles", Version 1.1.1, Open Mobile Alliance™, OMA-ArchitecturePrinciples-V1_1_1, URL:http://www.openmobilealliance.org/

[ARCH-INVEN] "Inventory of Existing Architectures to OMA", Version 1.0.1, Open Mobile Alliance™, OMA-Inventory-of-Architectures-and-Services-V1_0_1, URL:http://www.openmobilealliance.org/

[ARCH-REVIEW] "OMA Architecture Review Process", Version 1.3, Open Mobile Alliance™, OMA-ORG-ARCHReviewProcess-V1_3, URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

For the purposes of this document, the terms and definitions given in [**OMA-DICT**] apply and the following also apply:

Client Side Content Screening	Content screening performed at the mobile terminal.
Client Side Content Screening Framework	An abstract conceptual structure used as the basis for contructing interaction model between OMA/non-OMA enablers and content scanning functionality through a set of interfaces with the ultimate goal of bringing forth content screening capability to the mobile terminal.
Content	Data or code delivered to an end-user and/or end-user’s terminal.
Content Scanning	The actual operation of looking at the data to determine whether it is a potential candidate for screening and level of severity if found to be as such. What this operation consist of would vary according to how content scanning functionality is implemented and falls outside the scope of this document.
Content Scanning Functionality	Content scanning performed for OMA/non-OMA enabler wishing to determine whether a content under consideration is undesirable or not. This performance is accessed by a set of interfaces specified by the content screening framework.
Content Screening	The act of protecting an end-user and/or end-user’s terminal from undesirable content by blocking access to the said content. This act may be in the form of warning message, confirmation of deletion, notification of deletion, silent deletion without notification, etc. Exact detail would vary according to severity level reported, I/O capability of mobile terminal, user preferences, etc.
Mobile Terminal	A device that receives content as part of its normal running operation.
Scan Engine	Component of client side content screening framework that performs content scanning service to OMA/non-OMA enablers related to end-user content delivery and/or processing.
Screening Action	The act of blocking an undesirable content (see ‘Content Screening’).
Server Side Content Screening	Content screening performed at the network by servers with content screening functionality. E.g. Proxy server, mail server, firewall, etc.

3.3 Abbreviations

OMA	Open Mobile Alliance

4. Introduction (Informative)

The purpose of this document is to define an architecture for content screening framework within the mobile terminal to provide OMA and non-OMA enablers the capability to detect and screen malicious content through a set of interfaces to a common content scanning functionality. By defining the architecture for client side content screening framework, it aims to meet the urgent market demand for an effective countermeasure to the growing amount of malicious content delivered to mobile terminals before more lethal variants, such as self-spreading viruses and worms, create havoc on networks and users as richer content become available.

4.1 Target Audience

The target audience for this document includes but is not limited to the following:

- The Working Group(s) that will create specifications based on this subject matter
- Working Groups that need to understand the architecture of this subject matter
- Architecture Working Group (e.g. during Architecture Reviews as defined in [**ARCH-INVEN**], to determine compliance of [**ARCH-PRINC**], etc.)
- Interoperability Working Group (e.g. for early analysis of interoperability requirements)
- Security Working Group

4.2 Use Cases

Two use cases are provided to illustrate the functions and roles of various system elements in the client side content screening framework. Both of the use cases concern a case where a message received by an email client contains a virus. The first use case illustrates using the pull model for content delivery while the latter illustrates using the push model. For more example use cases, please refer to [**CSCSF-RD-v1**].

4.2.1 Pull Model

1. End-user activates email client software on mobile terminal in order to access a new email message.
2. Email client software requests (“pulls”) new email message from email server.
3. Email client software downloads the new email message (which contains a virus) from Email server.
4. Before displaying the downloaded email message to the end-user, the email client software forwards the message to scan engine for maliciousness.
5. Scan engine analyzes the content forwarded by the email client software.
6. Scan engine detects presence of a virus within the content.
7. Scan engine responds to the email client software with a scan result that indicates that the forwarded content is a malicious content.
8. Email client software screens the email message from user access.
9. Email client software notifies the end-user (via warning dialog message or other appropriate user interface) that the email message was screened because it was found to be a malicious.

4.2.2 Push Model

1. Email server transmits (“pushes”) new email message (which contains a virus) to email client software on end-user’s mobile terminal.

2. Before displaying the newly arrived email message to the end-user, the email client software forwards the message to scan engine for maliciousness.
3. Scan engine analyzes the content forwarded by the email client software.
4. Scan engine detects presence of a virus within the content.
5. Scan engine responds to the email client software with a scan result that indicates that the forwarded content is a malicious content.
6. Email client software screens the email message.
7. Email client software may notify the end-user (via warning dialog message or other appropriate user interface) that the email message was screened because it was found to be a malicious.

4.3 Requirements

Requirement ID/Number	Phase Met	Section(s)
6.1.1 #1	1.0	5, 6
6.1.1 #2	1.0	5, 6
6.1.1 #3	1.0	5, 6
6.1.1 #4	1.0	5, 6
6.1.1 #5	1.0	5, 6
6.1.1 #6	1.0	5, 6
6.1.1 #7	1.0	5, 6
6.1.2 #1	1.0	5, 6
6.1.2 #2	1.0	5, 6
6.1.2 #3	1.0	5, 6
6.1.3 #1	1.0	5, 6
6.1.3 #2	1.0	5, 6
6.1.3 #3	1.0	5, 6
6.1.4 #1	1.0	5, 6
6.1.4 #2	1.0	5, 6
6.1.4 #3	1.0	5, 6
6.2 #1	1.0	5, 6
6.2.1 #1	1.0	5, 6
6.2.3 #1	1.0	5, 6
6.2.4 #1	1.0	5, 6
6.2.4 #2	1.0	5, 6
6.2.4 #3	1.0	5, 6
6.2.5 #1	1.0	5, 6
6.2.6 #1	1.0	5, 6
6.2.6 #2	1.0	5, 6

Table 1: Client Side Content Screening Framework Requirements

4.4 Planned Phases

The architecture described in this document concerns phase 1.0.

5. Logical Model (Informative)

OMA and non-OMA enablers related to end-user content delivery and/or processing (such as a browser, email client, file installer, HTTP protocol handler, etc) communicate with the scan engine via interface defined by the client side content screening framework. An enabler would pass content to the scan engine for analysis before it is actually processed and/or presented to the end-user. The result of the analysis by the scan engine, whether the content is found to be malicious or not, is returned to the calling enabler. The calling enabler screens the content if the content is found to be malicious. In such a case, it is recommended that screening action be conveyed to the end-user (in the form of warning message, confirmation of deletion, notification of deletion, etc). If the content was found to be non-malicious, then the enabler shall not screen the content but continue with its normal flow of operation.

The logical model of the content screening framework in a mobile terminal is shown in Figure 1.

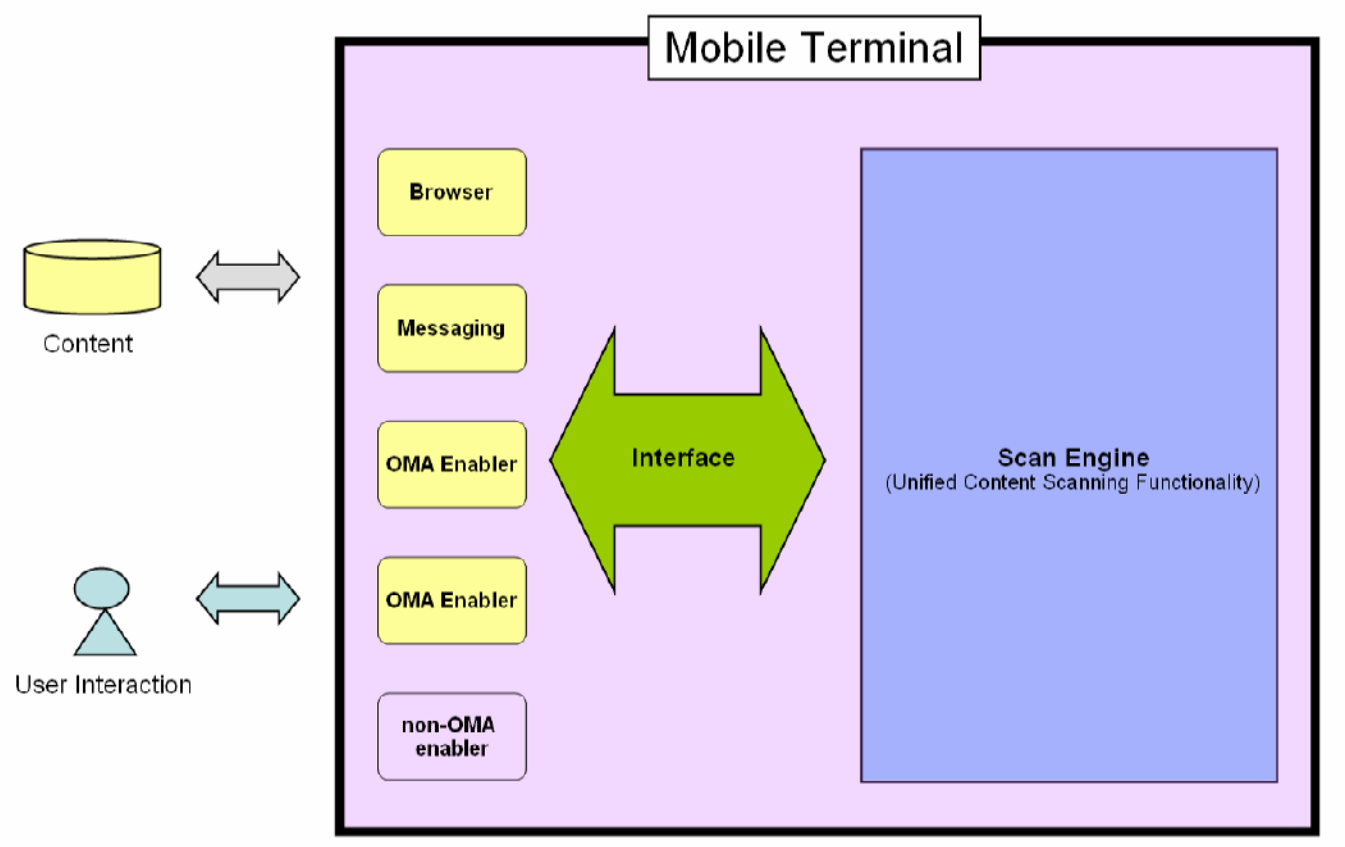


Figure 1: Client Side Content Screening Framework

6. Architectural Model (Normative)

This section describes the architecture model of client side content screening framework by identifying all internal functional components of the framework, the interface used between the components, and communication relationships between the components of the framework with other enablers and applications (including those enablers not defined by OMA). An implementation of an interface may result in software and/or hardware components that are not explicitly identified in this document. Implementation and platform details, however, are not within the scope of this document.

6.1 Dependencies

None.

6.2 Architectural Diagram

Architectural diagram of the client side content screening framework is shown in Figure 2. The blue boxes indicate functional components participating in the framework. The gray boxes indicate functional components that are not participating in the framework but which communicate with those located within. Brief descriptions for each of the functional components are described in Table 1.

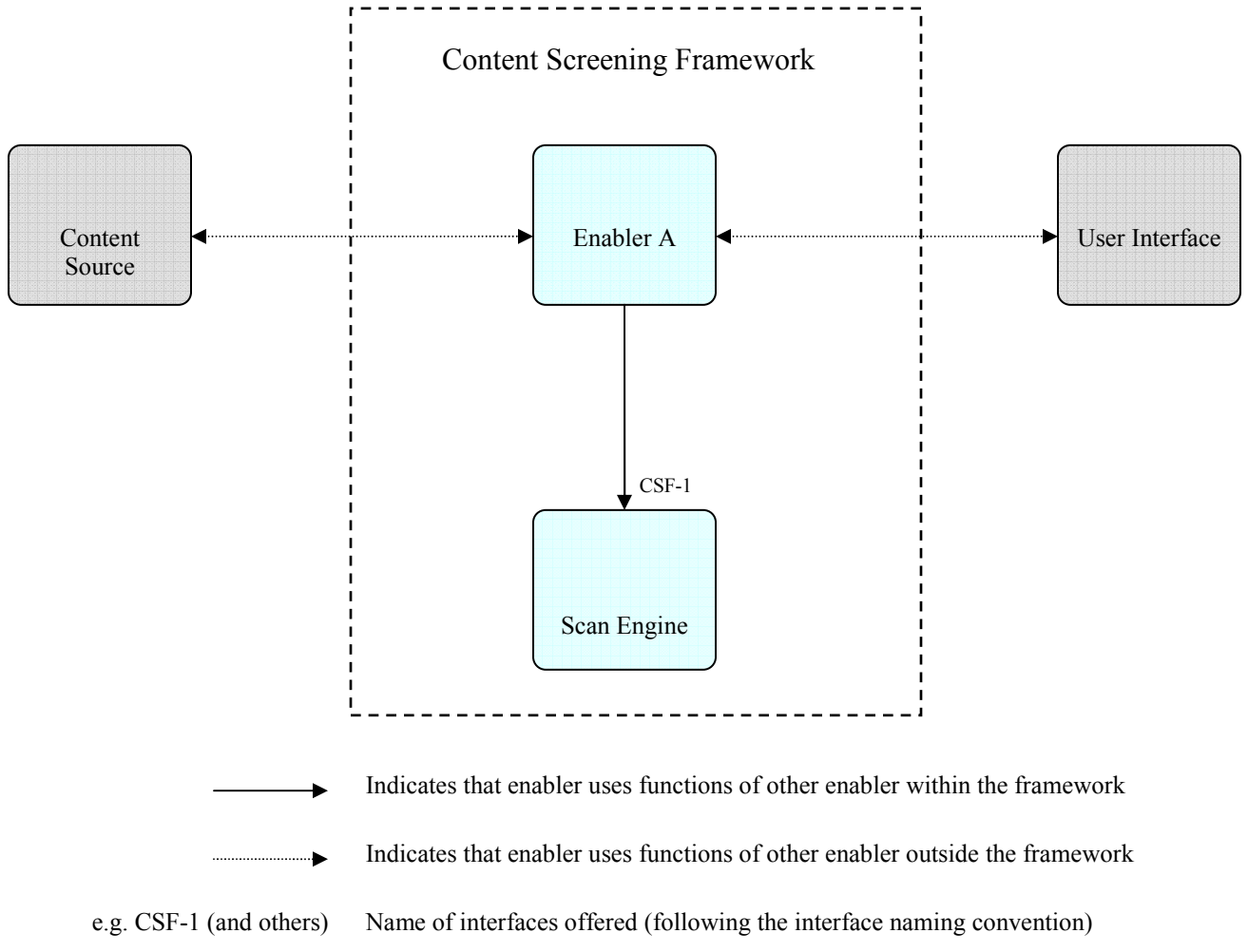


Figure 2: Architectural Model of Client Side Content Screening Framework

Functional Component Name	Description	Located in the client side content screening framework?	Implementation and platform details within the scope of this AD?
Content Source	An abstract term for a component that provides content to Enabler A. E.g. Web server serving Web pages, mail server serving email messages, bluetooth terminal sending bluetooth messages, or even removable media inserted into the end-user's terminal serving photo images.	No	No
Enabler A	OMA and non-OMA enablers related to end-user content delivery and/or processing. Enabler A resides in the terminal. E.g. Browser, MMS client, Email client, file installer, HTTP protocol handler, etc.	Yes	No
Scan Engine	Component of the client side content screening framework that provides content scanning service to Enabler A.	Yes	No
User Interface	An abstract term for end-user input/output interaction. E.g. LCD screen, keypad, speaker.	No	No

Table 2: Functional Components of the Architectural Diagram

6.3 Functional Components and Interfaces

Name	CSF-1
Description	Client Side Content Screening Framework Interface for Requesting Scanning of Content
Responsibility	This interface allows enabler related to end-user content delivery and/or processing to forward content to the scan engine to determine whether it is malicious or not. The scan engine analyzes the forwarded content and returns the result to the calling enabler. The calling enabler screens the content based on the result from the scan engine.

6.4 Flows

The purpose of this section is to describe the high-level logical flows between the architectural entities described in the architectural diagram for client side content screening framework. Two flows are presented, one using the Pull model (see Figure 3) and the other the Push model (see Figure 4). For both cases, the steps are identified by their corresponding numbers in their respective figures.

6.4.1 Pull Model

1. Enabler A receives request for content access from the User Interface.
2. Enabler A requests ("pulls") the content from Content Source.
3. Content Source responds to Enabler A with the requested content (but the content contains a virus).
4. Enabler A requests scanning of the downloaded content from Scan Engine via interface CSF-1.
5. Scan Engine scans the content and detects presence of a virus within the content.

6. Scan Engine responds to Enabler A with the result of the scan that the content is malicious.
7. Enabler A screens the content based on the result from Scan Engine.
8. Enabler A responds to User Interface that access to requested content was denied because it was found to be malicious.

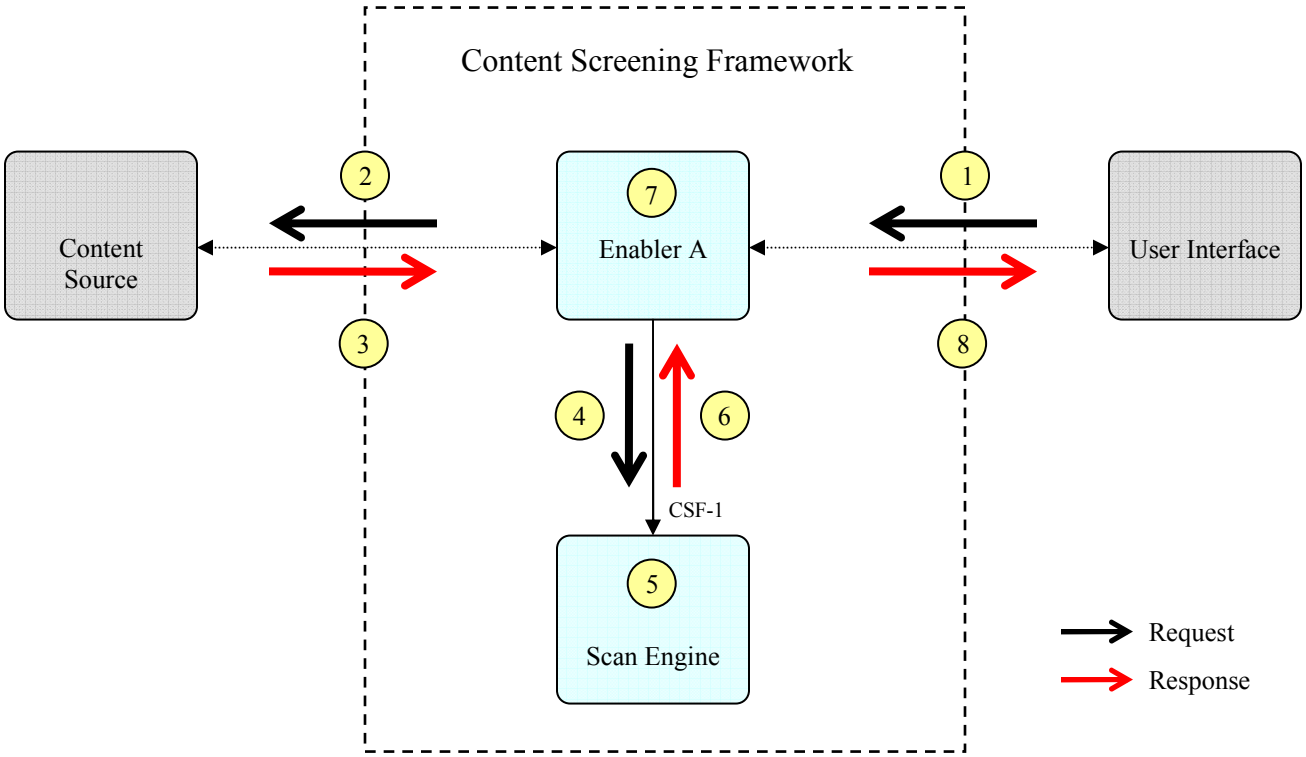
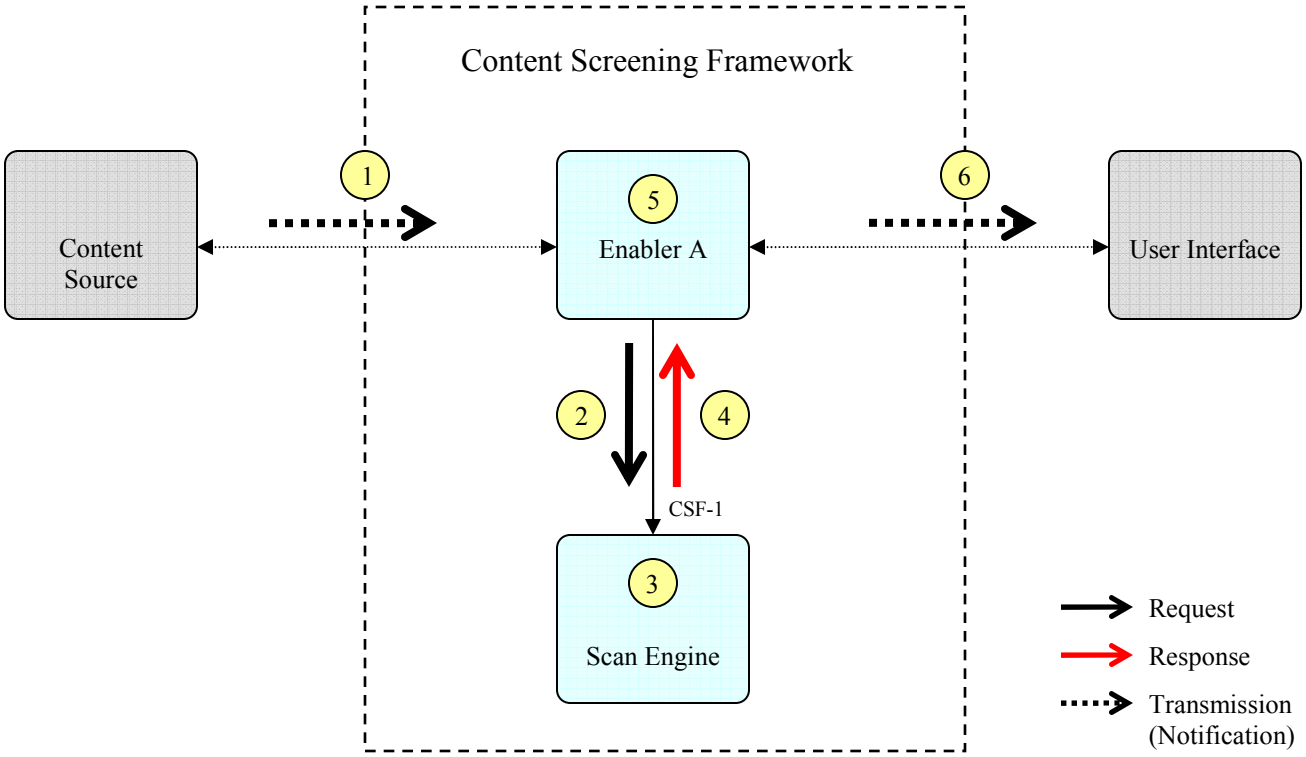


Figure 3: Flow of Architecture Model of Client Side Content Screening Framework using the Pull Model

6.4.2 Push Model

1. Content Source transmits ("pushes") content to Enabler A (but the content contains a virus).
2. Enabler A requests scanning of the newly arrived content from Scan Engine via interface CSF-1.
3. Scan Engine scans the content and detects presence of a virus within the content.
4. Scan Engine responds to Enabler A with the result of the scan that the content is malicious.
5. Enabler A screens the content based on the result from Scan Engine.
6. Enabler A may transmit notification to User Interface that content transmitted from the server was screened because it was found to be malicious.



Indicates that enabler uses functions of other enabler within the framework

Indicates that enabler uses functions of other enabler outside the framework

e.g. CSF-1 (and others) Name of interfaces offered (following the interface naming convention)

Figure 4: Flow of Architecture Model of Client Side Content Screening Framework using the Push Model

Appendix A. Change History (Informative)

A.1 Approved Version 1.0 History

Reference	Date	Description
OMA-AD-Client_Side_CS_FW-V1_0	14 Jun 2007	Status changed to Approved by TP OMA-TP-2007-0221- INP_ERP_Client_Side_CS_FW_V1_0_for_Final_Approval