Smartcard Web Server Enabler Architecture

Candidate Version 1.0 – 09 Feb 2007

Open Mobile Alliance

OMA-AD-Smartcard_Web_Server-V1_0-20070209-C

Contents

Figures

1. Scope (Informative)

The Smartcard enables network operators to provide network security to their customers and as a platform to run their services. Several standardization bodies develop smart card toolkit standards in order to fulfill these requirements.

The Smart Card Web Server (SCWS) intends to enable smart card issuers (e.g. Mobile Network Operators) to offer static or dynamic web pages. One operator centric example could be pages generated by applications running in the smart card (e.g. SIM, UICC or R-UIM), enabling local access to content (e.g. questionnaires, FAQs) or security-oriented services requiring keys stored in the smart card.

All these services will be accessible via a Web browser.

This document is an architecture document for the SCWS Enabler (work item presented in [SCWS WID]). It depicts the functionality, interfaces and information flow that is needed to address the requirements related to this work item as described in the Smartcard Web Server Requirements document [SCWS-RD].

This work item addresses the interfaces needed to access and use a web server in the smart card, the HTTP profile that need to be implemented and the access control to this SCWS.

The web server implementation in the smart card is considered out of the scope for this release so this document does not describe the SCWS internal entities within such an implementation.

2. References

2.1 Normative References

[HTTP/1.1] "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, URL: http://www.ietf.org/rfc/rfc2616.txt

[HTTP over TLS] "Hypertext Transfer Protocol over TLS protocol", RFC 2818, May 2000, URL: http://www.ietf.org/rfc/rfc2818.txt

[ISO7816-4] "Information technology - Identification cards - Integrated circuit(s) cards with contacts – Part 4: Interindustry commands for interchange"

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

[RFC1630] "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web", URL: http://www.ietf.org/rfc/rfc1630.txt

[RFC1738] "Uniform Resource Locators (URL)", URL: http://www.ietf.org/rfc/rfc1738.txt

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

[RFC2617] "HTTP Authentication: Basic and Digest Access Authentication", URL: http://www.ietf.org/rfc/rfc2617.txt?number=2617

[SCWS-RD] "SCWS Requirements", Open Mobile Alliance™, OMA-RD-Smartcard_Web_Server-V1_0, URL: http://www.openmobilealliance.org/

[TLS] "Security Transport Protocol", RFC 2246, January 1999, URL: http://www.ietf.org/rfc/rfc2246.txt

[TS31102] "Characteristics of the Universal Subscriber Identity Module (USIM) application", 3rd Generation Partnership Project (3GPP), TS 31.102, URL: http://www.3gpp.org

[TS102221] "Smart Cards; UICC-Terminal interface; Physical and logical characteristics", European Telecommunications Standards Institute (ETSI), TS 102 221,URL: http://www.etsi.org

[TS102223] "Smart Cards; Card Application Toolkit (CAT)", European Telecommunications Standards Institute (ETSI), TS 102 223, http://www.etsi.org

[WAPWAE] "Wireless Application Environment Specification", Open Mobile Alliance™, OMA-WAP-WAESpec-V2_3, URL: http://www.openmobilealliance.org/

[WP HTTP] "Wireless Profiled HTTP", WAP Forum™, WAP-229-HTTP-20010329-a, URL: http://www.openmobilealliance.org/

2.2 Informative References

[ARCH-PRINC] "OMA Architecture Principles", Open Mobile Alliance™, OMA-ArchitecturePrincipes-V1_2, URL: http://www.openmobilealliance.org/

[ARCH-REVIEW] "OMA Architecture Review Process", Open Mobile Alliance™, OMA-ARCHReviewProcess-V1_1, URL: http://www.openmobilealliance.org/

[OMA-DICT] "OMA Dictionary", Open Mobile Alliance™, OMA-Dictionary-V2_1, URL: http://www.openmobilealliance.org/

[SCWS WID] Smartcard web server work item (WID 92)

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 [2119].

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

3.2 Definitions

Application	The implementation of a well-defined and related set of functions that perform useful work on behalf of the user. It may consist of software and or hardware elements and associated user interfaces.
BIP	Bearer Independent Protocol as defined in ETSI [TS102223].
Browser	A program used to view (x) HTML or other media type documents.
Content Provider	An entity that provides data that forms the basis of a service.
Device	In this context, a Device is a voice and/or data terminal that uses a Wireless Bearer for data transfer. Device types may include (but are not limited to): mobile phones (GSM, CDMA, 3GSM, etc.), data-only terminals, PDAs, laptop computers, PCMCIA cards for data communication and unattended data-only Devices (e.g., vending machines). Smart cards are not considered as part of the device within the context of the Smart Card Web Server.
Local services	Services that reside in the smart card web server.
Network Operator	An entity that is licensed and allocated frequency to operate a public mobile wireless telecommunications network for the purpose of providing publicly available commercial services.
Smart card	This is a portable tamper resistant device with an embedded microprocessor chip. A smart card is used for storing data (e.g. access codes, user subscription information, secret keys etc.) and performing typically security related operations like encryption and authentication. A smart card may contain one or more network authentication applications like the SIM (Subscriber Identification Module), USIM, R-UIM (Removable – User Identification Module).
Smart card application	An application that executes in the smart card.
Smart card issuer	The entity that gives/sales the smart card to the user (e.g. network operator for a SIM card).
UICC	UICC is the smart card defined for the 3G standard [TS102221].
URI	Uniform Resource Identifiers (URI, see [RFC1630]) provides a simple and extensible means for identifying a resource. URI syntax all widely used to address Internet resources over the web but is also adapted to local resources over a wide variety of protocols and interfaces.
URL	The specification is derived from concepts introduced by the World-Wide Web global information initiative, whose use of such objects dates from 1990 and is described in "Universal Resource Identifiers in WWW", [RFC1630]. The specification of URLs (see [RFC1738]) is designed to meet the requirements laid out in "Functional Requirements for Internet Resource Locators".
User	Person who interacts with a user agent to view, hear or otherwise use a resource.
Web Page	A document viewable by anyone connected to the page server who has a web browser.
Web server	A server process running on a processor, which sends out web pages in response to HTTP requests from browsers.

3.3 Abbreviations

AACL	Application Access Control List
ACP	Access Control Policy
APDU	Application Protocol Data Units

IP	Internet Protocol
NACL	Network Access Control List
OMA	Open Mobile Alliance
R-UIM	Removable User Identity Module
SCWS	Smart Card Web Server
TCP	Transmission Control Protocol
USIM	Universal Subscriber Identity Module

4. Introduction

A Smart Card Web Server (SCWS) is a HTTP server implemented in the smart card embedded in the mobile device (e.g. SIM, (U)SIM, UICC). It will allow network operators to offer smart card based services to their customers by using the widely deployed [HTTP/1.1] protocol.

This solution integrates well in the Internet and the OMA architecture and affects the device and the smart card itself. The goal of this architecture is to have a minimum impact on the device and other system elements like remote servers. The main scope of the WI is to allow a local communication between the device WEB browser and the Smart Card Web Server. This will allow the user to browse static and dynamic content on the Smart Card Web Server and the implementation of dynamic web applications in the smart card. The security constraints are expressed in the Requirement document and the architecture and solution itself should accommodate them.

As the solution relies on well-known Internet protocols, it mainly concentrates on specifying the needed modules/gateways, in the device and smart card, to allow an HTTP communication between the device and the smart card. It is also aimed to have no change in the device browser in order to make the SCWS browsing as transparent as the browsing of any other remote Web server. The architecture takes into account possible security vulnerabilities coming from the SCWS connection to the mobile device network stack.

A Smart card-URI is used in order to communicate with a web server that is embedded in the smartcard (SCWS). We limit our discussion to smart card platforms such as (U)SIM (Subscriber Identification Module), UICC, R-UIM (Removable – User Identification Module) in a mobile phone.

The current work is phase 1.0.

As the SCWS connectivity is provided by UICC commands, it will also follow new connectivity solutions that could be specified by the ETSI SCP. The architecture described in this document takes into account the possible evolutions of the UICC connectivity solution.

5. Architectural Model

The SCWS enabler architecture provides a functional description of the SCWS itself and a functional and behavioural description of the OMA SCWS gateway that provides the connectivity of the SCWS to the hosting Device network stack. It also describes the interface with a remote administration application using an end-to-end secure connection. The basic principles of this solution are described hereafter.

The smart card provides a web server for the user to browse using the device WEB browser. This web server is accessible via a gateway that translates the TCP/IP protocol to another local protocol between the device and the smart card. The HTTP requests and responses are then sent directly to the SCWS over the local smartcard-device protocol. The current proposal for the local access URL (from within the device) to the SCWS is to use the loopback address with two TCP port numbers to be assigned for this purpose:

- 20080 for HTTP
- 20443 for HTTPS

Note: These TCP port numbers may be different in the final specification depending on the actual port numbers that will be allocated by IANA. These numbers are kept in this document for clarity in the following chapters.

The architecture should be open to allow the choice of several smartcard-device protocols as the “local bearer” to transport the HTTP requests and responses. One example of such a local bearer relies on a protocol that is already standardized in ETSI/SCP. This protocol is called the Bearer Independent Protocol (BIP).

5.1 Dependencies

- OMA Browser
- HTTP/1.1
- TLS
- HTTPS

5.2 Architectural Diagram

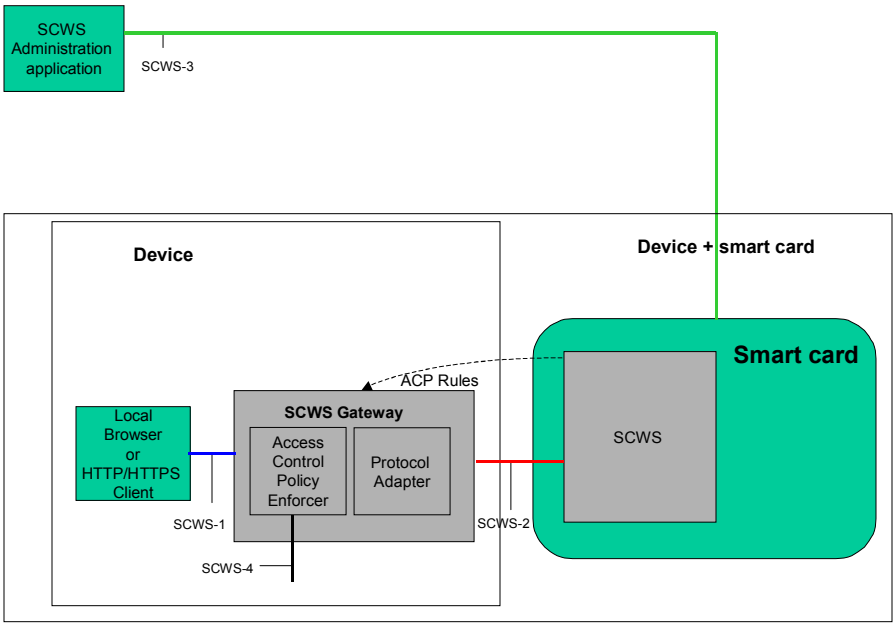

Figure 1: SCWS Connectivity Architectural Model

5.3 Functional Components and Interfaces

5.3.1 Functional Components

5.3.1.1 SCWS

Name: SCWS Server.

Description: SCWS Server

Responsibility: In charge of processing of client requests. This component resides in the smart card and processes HTTP requests.

5.3.1.2 SCWS Gateway

Name: SCWS Gateway

Description: SCWS Gateway

Responsibility: Provides the link between the hosting device network stack and the Smart card Interface and protocol at TCP level. Two main functions are identified within this component:

- Protocol translation from TCP/IP to the local transport protocol between the device and the smart card
- Enforcement of access control policy to the SCWS based on access control rules that are read from the smart card

Note: when the smart card is able to communicate directly over the TCP/IP protocol this component will evolve accordingly.

5.3.1.3 HTTP client

Name: HTTP client

Description: local HTTP client (see "local browser or HTTP client" in the architecture diagram).

Responsibility: This is the HTTP client used to connect the SCWS using an HTTP layer.

5.3.1.4 HTTPS Client

Name: HTTPS client

Description: local HTTPS client (see "local browser or HTTPS client" in the architecture diagram).

Responsibility: This is the HTTPS client used to connect the SCWS using a TLS layer.

5.3.1.5 SCWS administration application

Name: SCWS administration application

Description: Remote application providing administration mechanisms for SCWS.

Responsibility: This is the administration platform providing content that the SCWS administrator wants to install and manage in the SCWS.

5.3.2 Interfaces and Protocols

5.3.2.1 SCWS-1:

Name: SCWS 1

Description: Interface between browser, HTTP/HTTPS client and the SCWS Gateway for sending/receiving HTTP or HTTPS requests and responses.

Entities in this enabler that will use the interface or protocol: HTTP or HTTPS client.

Protocol: TCP/IP

5.3.2.2 SCWS-2:

Name: SCWS 2

Description: Interface between SCWS Gateway and the SCWS.

Entities in this enabler that will use the interface or protocol: SCWS Gateway and the SCWS.

Interface: SCWS transport protocol

5.3.2.3 SCWS-3:

Name: SCWS 3

Description: Interface between the smart card and a remote administration application.

Entities in this enabler that will use the interface or protocol: Smart card and the remote administration application.

Interface: Existing OTA protocols

5.3.2.4 SCWS-4:

Name: SCWS 4

Description: This is an I2 type interface, meaning that it depends on external capabilities. This is the Interface between the ACP Enforcer and the device operating system.

Entities in this enabler that will use the interface or protocol: ACP Enforcer and the device operating system.

Interface: The device operating system interfaces depending on the deploying platform. More information is in [5.6].

5.4 Flows (informative)

The purpose of this section is to describe the high-level data flows between the architectural entities described in the architectural diagram.

5.4.1 HTTP Messages Flow

1. Client application generates an HTTP request.
2. The HTTP request is sent through the Device Network Stack to the SCWS Gateway where the ACP Enforcer applies filtering as specified in the filtering rules read from the smart card. If the SCWS port is blocked for this client then the message cannot reach the SCWS.
3. The SCWS Gateway sends the message to the card over the local transport protocol.
4. The SCWS parses the request and prepares the data to send in response to the browser.
5. The SCWS Gateway sends the message back to the Client application.

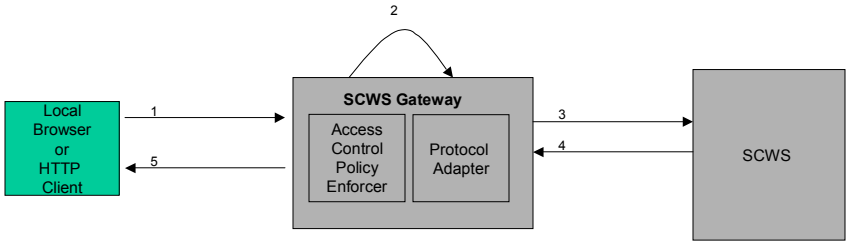

Figure 2: Local client connection

5.4.2 HTTPS Messages Flow

1. Client application initiate an HTTPS secure session with the SCWS if not already negotiated: The TLS packets are sent through the Device Network Stack to the SCWS Gateway where the ACP Enforcer applies filtering as specified

in the filtering rules read from the smart card. If the SCWS port is blocked for this client then the message cannot reach the SCWS (client will timeout).

2. If a TLS session is successfully established the client application sends the HTTP requests over the secure channel. The HTTP request protected with TLS is sent through Device Network Stack to the SCWS Gateway.
3. SCWS gateway sends the Secured message to the card over the local transport protocol.
4. The SCWS parses the secure message and prepares the data to send in response to the client application. The Secured response is sent to the SCWS Gateway over the local transport protocol.
5. The SCWS Gateway sends the secured message to the client application.

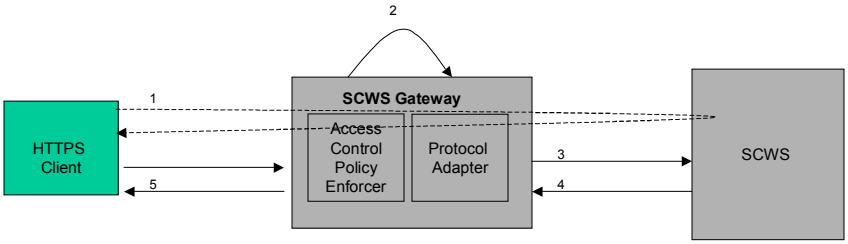

Figure 3: HTTPS client connection

5.4.3 Administration messages flow

1. Use existing smart card OTA protocols to administrate the SCWS.
2.

5.5 Security considerations

The SCWS is a web server, running within the smart card, to which local HTTP applications in the device can connect. The security considerations are the same as with any remote server that the user can browse with the handset Web browser. The SCWS shall implement HTTP and HTTPS and thus provide the same level of authentication, confidentiality and integrity as provided by other Web servers.

5.5.1 User authentication

If the smart card resource requires an access condition, which has not been fulfilled, the SCWS will provide means to enable this security condition as defined for local terminal-application APDU protocols (e.g. may perform a request to the user in

order to ask for a PIN). It is proposed to rely on RFC2617 for the authentication (e.g. “basic access authentication” or “digest”).

5.6 Access Control Policy

A complementary and optional security feature is the implementation of an additional access control to the SCWS within the device itself. It is called the ACP Enforcer (Access Control Policy Enforcer) and is aimed to provide an internal firewall for handset applications. It provides mainly a protection against denial of service attacks on the SCWS. The ACP Enforcer may be needed in devices that allow the user to download and install applications in the device itself (e.g. open OS phones). One use case is the download and installation of a malicious application in the handset that will try to block the access to the SCWS or ask the user for his passwords in order to access private information in the SCWS.

The Access Control Policy (ACP) is a data object that the device, implementing an ACP Enforcer, can retrieve from the smart card. An ACP Enforcer can be implemented by devices that implement a trusted execution environment (as defined by external standardisation fora). The ACP data object defines the following possible internal filtering rules:

- Allow access to the SCWS to applications that are trusted by the handset manufacturer
- Allow access to the SCWS to all trusted applications in the handset
- Allow access to the SCWS to some trusted applications in the handset that are identified by the hash of the signing certificate
- Allow access to all applications

A “Trusted Application” is an application that is signed and wherein the signature can be verified by a Trusted Certificate within the device or a Trusted Certificate that is retrieved by the device from the smart card.

The Access Control Policy Enforcer can enforce access restrictions to the SCWS by blocking access to the relevant TCP ports for certain local applications within the device.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-AD-Smartcard_Web_Server-V1_0	20 Jan 2005	n/a	First version
	10 Jun 2005		Integrating comments from the joint meeting between SEC-SCT and BAC-MAE (see document OMA-SCT-2005-0038-minutes-ioint-meeting-BAC-MAE)
	03 Aug 2005		Application of Architecture Document Template Removal of all text that should belong to a technical specification
	06 Oct 2005		Removal of internal views and concentrating on external interfaces
	23 Nov 2005		Removal of remote browsing and introduction of a new administration interface
	15 Dec 2005		Updates to the ACP section and the addition of a new I-2 interface
	29 Mar 2006		Remove use cases and requirement according to review report (ADRR)
	04 Feb 2007		Clean-up for TP approval
Candidate Version OMA-AD-Smartcard_Web_Server-V1_0	09 Feb 2007		Status changed to Candidate by TP TP ref # OMA-TP-2007-0078R05-INP_SCWS_V1_0_ERP_and_ETR_for_candidate_approval Editorial clean-up prior to publication.

Appendix B. URL description

The proposed SCWS URL will take the form:

```
http://<IPaddress>:<port>/<path>?<searchpart>

https://<IPaddress>:<port>/<path>?<searchpart>
```

according to [RFC1738]. The optional <searchpart> is a sequence of one or more <name>=<value> pairs separated by a '&' character.

The SCWS SHALL support URLs with a length of at least 1024 characters.

B.1 IP Address

B.1.1 Local

When connected from a local client, the loopback IP Address 127.0.0.1 will be used. This address is also named "localhost" on some systems. However, only the IP address 127.0.0.1 should be used in the URL and not any mnemonic name.

B.2 Port Number

Each protocol (e.g. HTTP or HTTPS) will use its own port number into the host device and will be transferred to the SCWS using the APDU transport.

As HTTP and HTTPS protocols already have TCP port numbers reserved and it must remain possible for the hosting device to run its own HTTP and HTTPS services, an offset will be added to the usual port number.

This offset needs to be decided taking into account the already reserved port numbers. For example the offset 20000 provides the way to address the port 80 for HTTP and the port 443 for HTTPS without collision with any other protocol.

HTTP will be addressed using the TCP port number 20080.

HTTPS will be addressed using the TCP port number 20443.

B.3 Sample URL to get static content

It is possible to address any resource accessible with the SCWS. This resource can be an xHTML file.

As an example, a file called "foobar.xhtml" in directory "pub/files" corresponds to this URL:

```
http://127.0.0.1:20080/pub/files/foobar.xhtml

https://127.0.0.1:20443/pub/files/foobar.xhtml
```

B.4 Sample URL to get dynamic content trough an application

Applications in the smart card are identified in the URL can be triggered by the SCWS. An application performs a specific task and may dynamically create content and return it to the client. Parameters for the application can be passed in the URL. By convention the parameters start with the '?' character and are being formatted as a series of name=value pairs, separated by the '&' character. The SCWS forwards the parameters to the addressed application.

Example:

The following URLs include parameters which are specific for the addressed applications:

http://127.0.0.1:20080/cgi/SSO?account=username&otherparam=123

http://127.0.0.1:20080/cgi/display?df=7F01&ef=3F01&record=01&offset=50&length=10

http://127.0.0.1:20080/cgi/update?df=7F01&ef=3F01&record=01&offset=50&length=3&value='abc'