Smartcard Web Server Enabler Architecture

Approved Version 1.2 – 05 Mar 2013

Open Mobile Alliance

OMA-AD-Smartcard_Web_Server–V1_2-20130305-A

Contents

Figures

1. Scope (Informative)

The Smart Card 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

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

[ETSI TR 102 216] "TR 102 216 Technical Report Smart Cards; Vocabulary for Smart Card Platform specifications", v3.0.0, European Telecommunications Standards Institute (ETSI), URL: http://www.etsi.org

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

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

[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

[OMA Push] "OMA Push", Open Mobile Alliance™, OMA-Push-V2_2, URL: http://www.openmobilealliance.org/

[OMA SIP Push] "OMA SIP Push", Open Mobile Alliance™, OMA-SIP-Push-V1_0, URL: http://www.openmobilealliance.org/

[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

[SCWS-RD] "SCWS Requirements", Open Mobile Alliance™, OMA-RD-Smartcard_Web_Server-V1_2, URL: http://www.openmobilealliance.org/

[TLS 1.0] "Security Transport Protocol", RFC 2246, January 1999, URL: http://www.ietf.org/rfc/rfc2246.txt

[TLS 1.1] "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006, URL: http://www.ietf.org/rfc/rfc4346.txt

[TLS 1.2] "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008, URL: http://www.ietf.org/rfc/rfc5246.txt

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_4_1, URL: http://www.openmobilealliance.org/

[OMA-DICT] "OMA Dictionary", Open Mobile Alliance™, OMA-Dictionary-V2_7, URL: http://www.openmobilealliance.org/

[SCWS WID] Smartcard web server work item (WID 0196)

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 [**Error! Reference source not found.**].

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 TS 102 223].
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.
Smart Card	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). In addition, the Smart Card refers to the smart card definition of [ETSI TR 102 216].
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 ETSI standard [ETSI TS 102 221]. It is a platform to resident applications (e.g. USIM, CSIM or ISIM).
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

ACP	Access Control Policy
APDU	Application Protocol Data Units

IP	Internet Protocol
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 (Informative)

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 Smart Card (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.

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.

4.1 Version 1.0

The Smart Card Web Server 1.0 defines all the main requirements of an HTTP server implemented in a Smart Card, allowing an HTTP client running in the terminal (e.g. the browser) to access resources stored in the Smart Card. The content delivered by the SCWS can be static resources but also be generated by a Smart Card application. The SCWS 1.0 also defines the remote administration of the Smart Card Web Server by an authorized entity.

4.2 Version 1.1

The Smart Card Web Server 1.1 enabler is a set of optimisations of the Smart Card Web Server 1.0 enabler and therefore does not introduce any new requirement or any change into the architecture. This enabler therefore refers to the requirements and architecture documents of the Smart Card Web Server 1.0 enabler.

4.2.1 Version 1.1.1

The Smart Card Web Server 1.1.1 enabler provides corrections and clarifications on the Smart Card Web Server 1.1.

4.3 Version 1.2

The Smart Card Web Server 1.2 enabler introduces the references to latest versions of TLS (i.e. [TLS 1.1] and [TLS 1.2]) and a new requirement confirming that another removable web server operating in the same terminal can be accessed.

The Smart Card Web Server 1.2 enabler also provides clarification on the implementation of the Smart Card Web Server when using TCP/IP transport Protocol.

The Smart Card Web Server 1.2 enabler introduces the notion of Granted Memory associated to a card administration agent. It allows restricting the amount of content associated to an authorized entity administrating the SCWS content.

The Smart Card Web Server 1.2 enabler introduces also the possibility to send the Push message, which triggers the Remote Administration Session, over SIP as an alternative to the Push message sent over the formatted SMS.

With the increase of opened device operating systems, the Smart Card Web Server 1.2 sets the implementation of the Access Control Policy (ACP) mechanism as mandatory for the device.

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 server 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 Smart Card-device protocol. The current proposal for the local access URL (from within the device) to the SCWS is to use the loopback IP address (also named "localhost") with two TCP port numbers to be assigned for this purpose:

- 3516 for HTTP
- 4116 for HTTPS

The architecture should be open to allow the choice of several Smart Card-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).

If the Smart Card implements a TCP/IP stack and has its own IP address, this IP address can be dynamically allocated, it is recommended to use the name "localuicc" to address the Smart Card. In this case, the default HTTP ports are used: 80 for HTTP and 443 for HTTP over TLS.

A SCWS Remote Administration protocol is also defined in the SCWS enabler to provide an efficient administration protocol used by a Remote Administration Server to update the content of the SCWS. This SCWS Remote Administration protocol is based on the HTTP/HTTPS protocol, the HTTP client being implemented by the SCWS Administration Agent in the Smart Card and the HTTP server being implemented by the Remote Administration Server. HTTP over TLS (HTTPS) is used to secure the exchange. The SCWS administrative commands are encapsulated in this Remote Administration protocol.

When the SCWS Administration Server wants to initiate a Remote Administration Session, it first has to send a Push message to the SCWS Administration Agent in the Smart Card, then the SCWS Administration Agent opens the Remote Administration Session. This Push message can be sent either using the usual formatted SMS or using the OMA SIP Push Enabler.

5.1 Dependencies

- OMA Browser
- HTTP/1.1 ([HTTP/1.1])
- TLS ([TLS 1.0], [TLS 1.1], [TLS 1.2])
- HTTPS ([HTTPS], [HTTP over TLS])
- OMA Push Enabler ([OMA Push])
- OMA SIP Push Enabler ([OMA SIP Push]).

5.2 Architectural Diagram

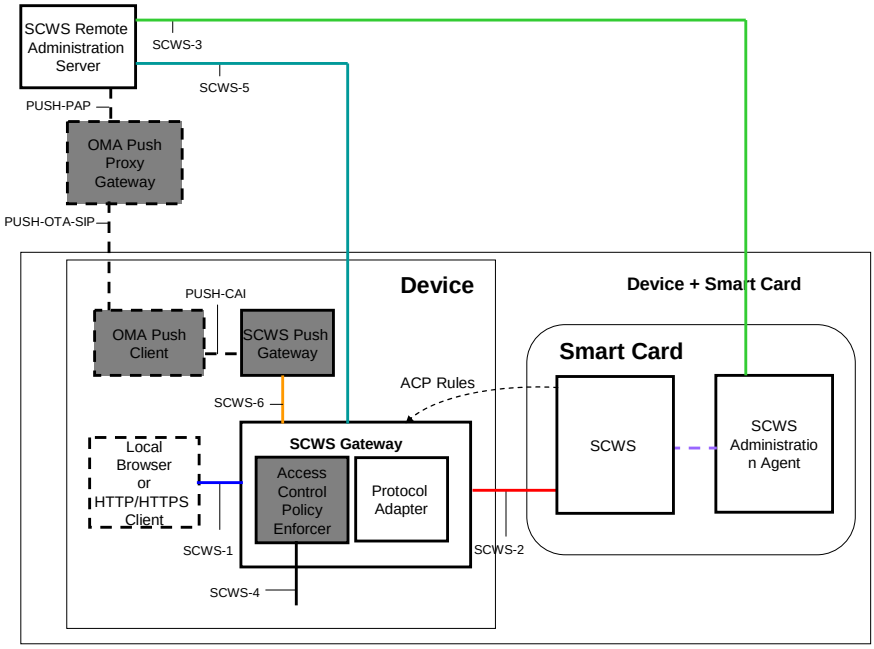

Figure 1: SCWS Connectivity Architectural Model

5.3 Functional Components and Interfaces/reference points definition

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 doesn't have to translate the TCP/IP packets.

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 Remote Administration Server

Name: SCWS Remote Administration Server

Description: Remote server 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.

5.3.1.6 SCWS Administration Agent

Name: SCWS Administration Agent

Description: Agent in the Smart Card in charge of the management of the protocol defined for the SCWS remote administration.

Responsibility: This is the agent in the Smart Card implementing the SCWS Remote Administration protocol and forwarding the administrative commands to the SCWS.

5.3.1.7 OMA Push Proxy Gateway

Name: OMA Push Proxy Gateway

Description: Push Proxy Gateway as defined in [OMA Push]

Responsibility: The proxy gateway that provides OMA push proxy services.

5.3.1.8 OMA Push Client

Name: OMA Push Client

Description: Push Client in the device as defined in [OMA Push]

Responsibility: Client in the device that receives the OMA Push messages sent by the OMA Push Proxy Gateway

5.3.1.9 SCWS Push Gateway

Name: SCWS push gateway

Description: Gateway in the device that uses the services of the OMA Push Client to receive the OMA Push message. This gateway forwards the Push message to the SCWS Gateway using the HTTP protocol.

Responsibility: Gateway in the device registered to the OMA Push Client that receives the OMA Push messages over SIP and forwards them to SCWS Gateway.

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 local transport protocol e.g. either TCP over the Bearer Independent Protocol or TCP over IP

5.3.2.3 SCWS-3:

Name: SCWS 3

Description: Interface between the Smart Card and a Remote Administration Server.

Entities in this enabler that will use the interface or protocol: Smart Card and the Remote Administration Server.

Interface: SCWS Remote Administration Protocols over HTTPs

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 [5.5].

5.3.2.5 SCWS-5:

Name: SCWS 5

Description: Interface between a Remote Administration Server and the device to send the Push message (e.g. Push SMS) used to trigger a SCWS administration session.

Entities in this enabler that will use the interface or protocol: The Remote Administration Server and the device.

Interface: Existing OTA protocol to send formatted SMS to the Smart Card

5.3.2.6 SCWS-6:

Name: SCWS 6

Description: Interface between the SCWS Push Gateway and the SCWS Gateway

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

Interface: HTTP

5.3.2.7 PUSH-PAP

Name: PUSH-PAP

Description: Push Access Protocol (PAP) interface as defined in [OMA Push]. This interface which the OMA Push Proxy Gateway exposes OMA Push services to the Push Initiator, in our case to the SCWS Remote Administration Server

Entities in this enabler that will use the interface or protocol: SCWS Remote Administration Server

Interface: HTTP

5.3.2.8 PUSH-OTA-SIP

Name: PUSH-OTA-SIP

Description: Push Over-The-Air (OTA) interface implemented over SIP as defined in [OMA SIP Push]. It is the interface via which the OMA Push Proxy Gateway and OMA Push Client interact using the Push-OTA protocol over SIP.

Entities in this enabler that will use the interface or protocol: None

Interface: OTA Protocol defined by [OMA SIP Push]

5.3.2.9 PUSH-CAI

Name: PUSH-CAI

Description: Push Client-Application Interface (CAI) as defined in [OMA Push]. It is the interface via which the OMA Push Client exposes OMA Push services to Push-enabled applications, in our case to the SCWS Push Gateway.

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

Interface: The device operating system interfaces depending on the deploying platform

5.4 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.

The protocol used for the SCWS remote administration shall implement the following security requirements: integrity, authentication, confidentiality and anti-replay. All these requirements can be achieved by using HTTP over TLS (HTTPs) protocol.

5.4.1 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”).

5.5 Access Control Policy

A complementary 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 is especially useful 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, SHALL retrieve from the Smart Card. An ACP Enforcer SHALL 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 SHALL 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
OMA-AD-Smartcard_Web_Server-V1_2-20130305-A	05 Mar 2013	Status changed to Approved by TP TP Ref # OMA-TP-2013-0078-INP_SCWS_V1.2_ERP_for_Final_Approval

Appendix B. Flows (informative)

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

B.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 Smart 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.

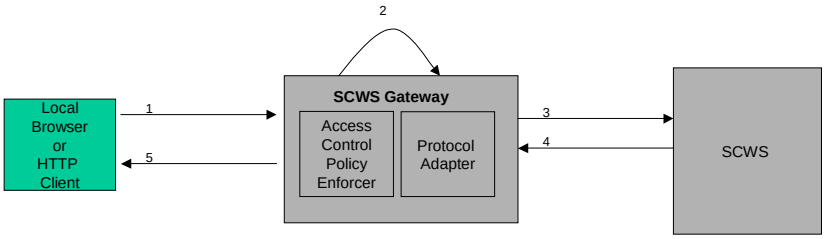

Figure 2: Local client connection

B.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 Smart 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.

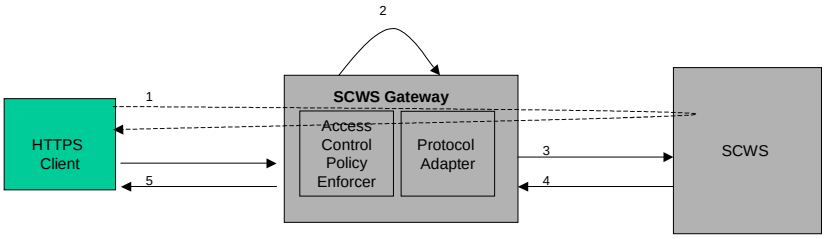

Figure 3: HTTPS client connection

B.3 Administration messages flow

1. Open the HTTPS secure session with the SCWS Remote Administration Server
2. HTTP request sent to the SCWS Remote Administration Server
3. HTTP response including the SCWS administrative commands
4. HTTP request including the result of the previous administrative commands.
5. HTTP response (final)
6. Close the HTTPS session

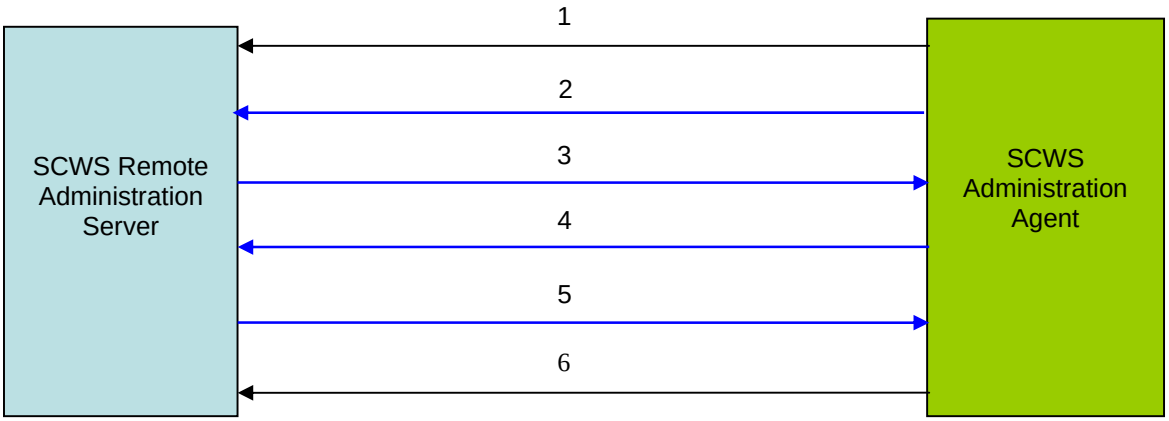

Figure 4: SCWS Remote Administration connection

Appendix C. URL description (normative)

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.

C.1 IP Address

C.1.1 Local

If the Smart Card does not have its own IP address and does not directly support TCP/IP, the SCWS Gateway in the device is used as a protocol adapter to forward the request to the Smart Card using the Bearer Independent Protocol. In this case, 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.

If the Smart Card has its own IP address and directly supports TCP/IP and the device supports direct IP addressing of the Smart Card, then the IP address of the Smart Card can be used. However as this IP address can be allocated dynamically, the UICC name "localuicc" will be used to address the Smart Card.

C.2 Port Number

Each protocol (e.g. HTTP or HTTPS) will use its own port number into the host device.

If the Smart Card does not have its own IP address and does not directly support TCP/IP:

- HTTP will be addressed using the TCP port number 3516
- HTTPS will be addressed using the TCP port number 4116

If the Smart Card has its own IP address and directly supports TCP/IP:

- HTTP will be addressed using the TCP port number 80
- HTTPS will be addressed using the TCP port number 443

C.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://127.0.0.1:3516/pub/files/foobar.xhtml

https://127.0.0.1:4116/pub/files/foobar.xhtml
```

C.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://127.0.0.1:3516/cgi/SSO?account=username&otherparam=123

http://127.0.0.1:3516/cgi/display?df=7F01&ef=3F01&record=01&offset=50&length=10

http://127.0.0.1:3516/cgi/update?df=7F01&ef=3F01&record=01&offset=50&length=3&value
='abc'
```