THIS SPECIFICATION OR OTHER WORK PRODUCT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT SHALL BE MADE ENTIRELY AT THE IMPLEMENTER’S OWN RISK, AND NEITHER THE COMPANY, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER DIRECTLY OR INDIRECTLY ARISING FROM THE IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT.
### Tables

Table 1-1: Normative References.................................................................6
Table 1-2: Terminology and Definitions......................................................7
Table 1-3: Abbreviations and Notations .....................................................8
Table 1-4: Revision History ......................................................................9
Table 3-1: Specification to protocol versions mapping..............................16
1 Introduction

This document describes a simple mechanism enabling a system to discover the application protocols that are supported by a second system.

This mechanism is intended to be used when two system actors are discovering each other for the first time, or when an actor wants to publish updateable protocols capabilities (new supported protocols, or changes in existing protocols).

This document acts as a generic specification for a "system protocol discovery" mechanism. It is applicable whatever the application protocols that might be discovered.

In particular, it specifies the discovery of the so-called Mobile Messaging protocol that is defined in GlobalPlatform System – Messaging Specification for Management of Mobile-NFC Services [GP SM]. Other documents may define the discovery for other contexts.

Organization

Chapter 2 provides an introduction to the System Protocol Discovery Mechanism.

Chapter 3 defines the generic RESTful mechanism for system protocol discovery.

Chapter 4 defines the usage of this mechanism in the Mobile Messaging ecosystem.

Annex A presents the JSON Schema of the response body of the REST end point.

1.1 Audience

The target audiences for this specification are the system integrators, the system or component developers or architects, or the security experts of any entity involved in the systems part of a product or a solution.

In particular, it fully applies to the mobile-NFC ecosystem actors, including service providers, mobile network operators, and Trusted Service Manager solution vendors participating in GlobalPlatform smart card implementations, and developing infrastructure components and support systems.

1.2 IPR Disclaimer

Attention is drawn to the possibility that some of the elements of this GlobalPlatform specification or other work product may be the subject of IPR held by GlobalPlatform members or others. For additional information regarding any such IPR that have been brought to the attention of GlobalPlatform, please visit https://www.globalplatform.org/specificationsipdisclaimers.asp. GlobalPlatform shall not be held responsible for identifying any or all such IPR, and takes no position concerning the possible existence or the evidence, validity, or scope of any such IPR.
1.3 References

Table 1-1: Normative References

<table>
<thead>
<tr>
<th>Standard / Specification</th>
<th>Description</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPC_SPE_034</td>
<td>GlobalPlatform Card Specification v2.2.1</td>
<td>[GPCS]</td>
</tr>
<tr>
<td>GPC_SPE_007</td>
<td>GlobalPlatform Card – Confidential Card Content Management – Card Specification v2.2 – Amendment A</td>
<td>[GP Amd A]</td>
</tr>
<tr>
<td>GPC_GUI_010</td>
<td>GlobalPlatform Card – UICC Configuration v1.0.1</td>
<td>[GP UICC Config]</td>
</tr>
<tr>
<td></td>
<td>Configuration requirements for implementing GlobalPlatform Specifications on the UICC platform</td>
<td></td>
</tr>
<tr>
<td>GPC_GUI_049</td>
<td>GlobalPlatform Card – Secure Element Configuration v1.0</td>
<td>[GP SE Config]</td>
</tr>
<tr>
<td></td>
<td>Configuration requirements for implementing GlobalPlatform Specifications for Secure Elements</td>
<td></td>
</tr>
<tr>
<td>GPC_SPE_095</td>
<td>GlobalPlatform Card – Digital Letter of Approval</td>
<td>[GP DLOA]</td>
</tr>
<tr>
<td>GPD_SPE_027</td>
<td>GlobalPlatform Device – TEE Administration Framework</td>
<td>[GP TAF]</td>
</tr>
<tr>
<td>RFC 1778</td>
<td>The String Representation of Standard Attribute Syntaxes</td>
<td>[RFC 1778]</td>
</tr>
<tr>
<td>RFC 2119</td>
<td>Keys words for use in RFCs to Indicate Requirement Levels</td>
<td>[RFC 2119]</td>
</tr>
<tr>
<td>RFC 2616</td>
<td>Hypertext Transfer Protocol - HTTP/1.1</td>
<td>[RFC 2616]</td>
</tr>
<tr>
<td>RFS 4122</td>
<td>A Universally Unique IDentifier (UUID) URN Namespace</td>
<td>[RFC 4122]</td>
</tr>
<tr>
<td>RFC 4627</td>
<td>The application/json Media Type for JavaScript Object Notation (JSON)</td>
<td>[RFC 4627]</td>
</tr>
<tr>
<td>ISO/IEC 8825-1</td>
<td>Information technology — ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)</td>
<td>[ISO 8825-1]</td>
</tr>
<tr>
<td>ISO/IEC 8859-1</td>
<td>Information technology -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1</td>
<td>[ISO 8859-1]</td>
</tr>
<tr>
<td>IETF JSON Schema Draft 04</td>
<td>JSON Schema: core definitions and terminology Draft 04</td>
<td>[JSON Schema]</td>
</tr>
</tbody>
</table>

1.4 Terminology and Definitions

1.4.1 Key Words

The key words “MUST”, “MUST NOT”, “SHALL”, “SHALL NOT”, “REQUIRED”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document indicate normative statements and are to be interpreted as described in [RFC 2119].
1.4.2 Mandatory, Conditional, Optional

Parameters specified in this document may be set as Mandatory, Optional, or Conditional. The following definitions apply to these terms:

- Mandatory (M): Means that an entry must be supplied
- Conditional (C): Means that the usage of an entry is dependent upon a particular condition.
- Optional (O): Means that an entry can be supplied, but is not required to be supplied.

1.4.3 Other Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>Any electronic equipment into which an NFC Secure Element can be plugged, and that provides a capability for a server to reach the SE through an Over The Air (OTA) or Over The Internet (OTI) link. A mobile phone is a good example of such a device.</td>
</tr>
<tr>
<td>Discovering actor</td>
<td>In this specification, the actor that wants to discover the application protocols that are supported by the SPDM Provider actor.</td>
</tr>
<tr>
<td>REST</td>
<td>REST is a software architecture style consisting of guidelines and best practices for creating scalable web services. Unlike SOAP-based web services, there is no &quot;official&quot; standard for RESTful web APIs. This is because REST is an architectural style, unlike SOAP, which is a protocol. Even though REST is not a standard per se, most RESTful implementations make use of standards like HTTP, URI, JSON, XML, etc.</td>
</tr>
<tr>
<td>RESTful</td>
<td>Said of a system that follows the REST software architecture style.</td>
</tr>
<tr>
<td>Secure Component</td>
<td>GlobalPlatform generic terminology to represent either a Secure Element or a Trusted Execution Environment.</td>
</tr>
<tr>
<td>Secure Element</td>
<td>A tamper resistant component which is used in a device to provide the security, confidentiality, and multiple application environment required to support various business models. May exist in any form factor, such as embedded SE, SIM, UICC, smartSD, smart microSD, etc.</td>
</tr>
<tr>
<td>Secure Element Issuer</td>
<td>Holds the ultimate responsibility for the GlobalPlatform card. Responsible for developing the card product profile, choosing the platform and application technologies, and designing the card layout. Usually holds a particular Security Domain in the SE: the Issuer Security Domain (ISD).</td>
</tr>
<tr>
<td>Security Domain Manager</td>
<td>Directly or indirectly owns a Security Domain; is able to perform Card Content Management actions and/or owns keys in the Security Domain.</td>
</tr>
<tr>
<td>Security Domain Manager URL</td>
<td>The URL that can be obtained by sending a GET DATA command with tag '5F50' to a Security Domain of a Secure Element, as described in GPCS Amendment A [GP Amd A].</td>
</tr>
</tbody>
</table>
### Term | Definition
--- | ---
System Protocol Discovery Mechanism Provider actor (SPDM Provider actor) | In this specification, the actor that provides System Protocol Discovery Mechanism capabilities to Discovering actors.
System Protocol Discovery Mechanism URL | The complete URL, made available by the SPDM Provider actor to Discovering actors, and used for the System Protocol Discovery Mechanism’s HTTP REST request defined in this specification.
Trusted Execution Environment | Secure area of the main processor of a smart phone (or any connected device including tablets, set-top boxes and televisions). It guarantees code and data loaded inside to be protected with respect to confidentiality and integrity.

### 1.5 Abbreviations and Notations

<table>
<thead>
<tr>
<th>Abbreviation / Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BER</td>
<td>Basic Encoding Rule</td>
</tr>
<tr>
<td>CIN</td>
<td>Card Identification Number</td>
</tr>
<tr>
<td>CUD</td>
<td>Card Unique Data</td>
</tr>
<tr>
<td>DLOA</td>
<td>Digital Letter Of Approval</td>
</tr>
<tr>
<td>ESE</td>
<td>Embedded Secure Element</td>
</tr>
<tr>
<td>GPCS</td>
<td>GlobalPlatform Card Specification</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
</tr>
<tr>
<td>IIN</td>
<td>Issuer Identification Number</td>
</tr>
<tr>
<td>ISD</td>
<td>Issuer Security Domain</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>NFC</td>
<td>Near Field Communication</td>
</tr>
<tr>
<td>OID</td>
<td>Object Identifier</td>
</tr>
<tr>
<td>OTA</td>
<td>Over The Air</td>
</tr>
<tr>
<td>OTI</td>
<td>Over The Internet</td>
</tr>
<tr>
<td>REST</td>
<td>REpresentational State Transfer</td>
</tr>
<tr>
<td>SDM</td>
<td>Security Domain Manager</td>
</tr>
<tr>
<td>SE</td>
<td>Secure Element</td>
</tr>
<tr>
<td>SMC</td>
<td>Secure Memory Card</td>
</tr>
<tr>
<td>SOA</td>
<td>Service-Oriented Architecture</td>
</tr>
<tr>
<td>TEE</td>
<td>Trusted Execution Environment</td>
</tr>
<tr>
<td>Abbreviation / Notation</td>
<td>Meaning</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>UICC</td>
<td>Universal Integrated Circuit Card</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
</tbody>
</table>

### 1.6 Revision History

#### Table 1-4: Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 2014</td>
<td>“Version 1.0 Draft”</td>
<td>Committee Review</td>
</tr>
<tr>
<td>October 2014</td>
<td>“Version 1.0 Draft 6”</td>
<td>Member Review</td>
</tr>
<tr>
<td>March 2015</td>
<td>“Version 1.0 Draft 9”</td>
<td>Public Review</td>
</tr>
<tr>
<td>TBD</td>
<td>1.0</td>
<td>Initial release</td>
</tr>
</tbody>
</table>
2 Introduction to the Mechanism

This chapter provides an introduction to the System Protocol Discovery Mechanism.

2.1 Context of Appearance

The combination of Secure Elements with various communication technologies presents significant business opportunities when used in mobile phones. For example, the Near Field Communication (NFC) technology opens opportunities for applications such as payment, transport ticketing, loyalty, physical access control, and many other exciting new services.

To support this fast evolving business environment, several actors, in addition to Mobile Network Operators (MNOs), will become involved in the ecosystem, such as service providers (e.g. banks, transit authority, etc.) and Secure Element providers (e.g. micro-SD (µSD) issuers).

To fulfill their individual roles, these business actors need to communicate with each other and exchange messages in a reliable and interoperable way: for a bank (i.e. a service provider) to request the deployment of a payment application to an actor having an Over-The-Air (OTA) capability; for a Mobile Network Operator to notify the service provider of the various life cycle events of the end-user mobile environment (device lost, etc.), etc.

Such interfaces have been defined by GlobalPlatform in [GP SM]. This specification has standardized functions, messages, and Web Service endpoints that should be used in-between those actors to perform service management.

However, as [GP SM] evolves from one version to another (adding new features or enhancing existing ones), some actors might support several versions of this protocol. As deployments grow, actors might also need to adapt their systems and sometime move the locations of their Web Service endpoints. Finally, as the payment, ticketing, access, loyalty, etc., use cases are increasingly adopted, actors will need to dynamically discover the other party and go beyond business pre-agreements, for example to deploy services in a roaming situation.

For all those reasons, the System Protocol Discovery Mechanism has been created to enable two actors to:

- Identify each other,
- Share their working environment, i.e. which SE they would like to work on,
- Share the protocols that may be used for further dialog (optionally getting detailed information related to those protocols: endpoint addresses, versions, etc.).

This mechanism is protocol agnostic: it can be used to discover any type of protocol.

However, this specification also defines the usage of this mechanism for the mobile messaging specified in [GP SM]. Other documents may define the usage of this discovery mechanism for other contexts.
2.2 Example of use case related to SE discovery

Despite its generic applicability to any ecosystem and any protocol to be discovered, the System Protocol Discovery Mechanism has been defined with the following use case in mind: a Device application of a Service Provider is newly deployed on a Device. Through this application, the Service Provider wants to discover who are the actors that are responsible for the Secure Element hosted in the Device, in order to request the deployment of a mobile-NFC service in this mobile environment.

In more details:

- Thanks to the capability defined in the GlobalPlatform Card Specification [GPCS], the Device application of the Service Provider is able to uniquely identify the Secure Element, by querying the Issuer Security Domain (ISD) for the Issuer Identification Number (IIN) and the Card Identification Number (CIN).

- Thanks to the capability defined in GPCS Amendment A [GP Amd A], the Device application of the Service Provider is also able to query for the Security Domain Manager URL of the ISD.

- Once this information is sent back to the Service Provider backend system, the Service Provider (called the Discovering actor in this context) can use the Security Domain Manager URL to send a specific request that complies with the System Protocol Discovery Mechanism specification. In particular, the Security Domain Manager URL is complemented with the IIN and the CIN of the Secure Element and with the identity of the Discovering actor in order to build a System Protocol Discovery Mechanism URL (SPDM URL).

- When receiving a request on this SPDM URL, the ISD Security Domain Manager (called the SPDM Provider actor in this context) returns its identity and the list of supported application protocols.

- Based on the protocol information returned, the Service Provider can then initiate an application communication that corresponds to the business actions he would like to perform through the ISD. For example, request the creation of a Supplementary Security Domain that could host the Service Provider's mobile-NFC service (this application communication is out of scope of the System Protocol Discovery Mechanism).

Note that usage of the System Protocol Discovery Mechanism is not restricted to this specific use case. It can apply to other use cases that do not necessary imply a Service Provider or a Device application, the ISD Security Domain Manager, or that is not linked to the management of a mobile-NFC service. It can also apply to other Secure Components such as the Trusted Execution Environment (TEE).
3 System Protocol Discovery Mechanism, Generic Definition

3.1 Mechanism Definition

In the following sections, Discovering actor refers to the actor that wants to discover the application protocols that are supported by a SPDM Provider actor.

The System Protocol Discovery Mechanism consists of a simple HTTP REST call sent by the Discovering actor to an URL (called the System Protocol Discovery Mechanism URL) made available by the SPDM Provider actor. Version 1.1 of the HTTP protocol [RFC 2616] SHALL be used.

3.1.1 HTTP REST Request

The SPDM Provider SHALL provide the following REST end point:

HTTP Command:
GET

Resource URL:
spdm/protocols

Parameters:
- version (Mandatory): represents the version of the System Protocol Discovery Mechanism that is used by the Discovering actor in this HTTP REST request.
  By convention, the version set by the Discovering actor SHALL be the highest version that is supported by this actor.
- requesterId (Mandatory): represents the identifier of the Discovering actor (the requester). No specific format is defined at this stage of the document for such an identifier. Section 4 defines the format for some specific contexts, but other documents may define the format for other contexts.
- secureComponentType (Mandatory): the type of Secure Component. Value could be either "SE" for a Secure Element or "TEE" for a Trusted Execution Environment.
- secureComponentId (Mandatory): the identifier of the Secure Component.
  - For a Secure Element, it SHALL be the hexadecimal string representation of the Card Unique Data of the Secure Element (for example "564644453030303047544f303030303031303289345600000000000004F". Note that the Card Unique Data is the concatenation of the Issuer Identification Number (IIN) and the Card Identification Number (CIN) of the Secure Element.
  - For the Trusted Execution Environment, it SHALL be the UUID string representation of the TEE device ID as defined in [RFC 4122] (for example: "f81d4fae-7dec-11d0-a765-00a0c91e6bf6").
The query parameters MAY be provided in any order.

Other query parameters MAY be present. If so, they SHOULD be ignored, unless a specific meaning to those parameters (out of scope of this specification) has been agreed by business agreement between parties.

**Prototype:**

The prototype of the HTTP REST request is consequently the following:

```
spdm/protocols?version=<version>&requesterId=<Discovering actor Id>
&secureComponentType={SE|TEE}&secureComponentId=<SC Id>
```

### 3.1.2 HTTP REST Response

#### 3.1.2.1 Redirection

If the SPDM Provider actor wants to redirect the incoming HTTP REST request to another HTTP endpoint, then the SPDM Provider actor SHALL return an appropriate "HTTP 3xx" redirection code together with a new URL in the Location HTTP response header.

The body of the HTTP response SHALL be empty.

Such a standard redirection mechanism indicates to the Discovering actor that it SHOULD send a similar HTTP REST request to the URL specified in the Location HTTP response header. According to the HTTP specification [RFC 2616], several redirection codes may be used, such as, for example:

- 301 Moved Permanently
- 302 Found
- 303 See Other
- 307 Temporary Redirect

The Service Protocol Discovery Mechanism specification does not mandate nor recommend the particular usage of any one of those codes.

As the Service Protocol Discovery Mechanism is intended to be used with this new URL, the Location URL of the HTTP response header SHALL comply with the format specified in section 3.1.1 (and with the format specified in section 4.6.1 when applied to [GP SM]). It SHALL contain the same values for the query parameters as the one present in the URL initially requested. The Location URL of the HTTP response header MAY also contain other query parameters added during the redirection.
3.1.2.2 Standard Response

In any other case, the SPDM Provider actor SHALL return an "HTTP 200 OK" code, and the HTTP response body SHALL contain a data structure that respects the following JSON format [RFC 4627]. The JSON Schema representation of this format is provided in section A.1.

```json
{
  "version": "<version>",
  "supportedVersions": ["<version #1>", "version #2"],
  "providerId": "<SPDM Provider Id>",

  "supportedProtocols": [
    {
      "protocolId": "<Protocol #1 Id>",
      "protocolVersion": "<Protocol #1 Version>",
      "protocolInfo": {
        <Protocol #1 Info>
      }
    },
    {
      "protocolId": "<Protocol #N Id>",
      "protocolVersion": "<Protocol #N Version>",
      "protocolInfo": {
        <Protocol #N Info>
      }
    }
  ],

  "referAlso": [
    {
      "additionalProviderURL": "<Additional SPDM Provider #1 URL>"
    },
    {
      "additionalProviderURL": "<Additional SPDM Provider #N URL>"
    }
  ]
}
```

Where:

- The **version** parameter (Mandatory) represents the version of the System Protocol Discovery Mechanism protocol that is used by the SPDM Provider actor in this HTTP response.
- The **supportedVersions** parameter (Conditional) provides the list of versions of the System Protocol Discovery Mechanism protocol that are supported by the SPDM Provider actor. Presence of this field is detailed in section 0.
- The **providerId** parameter (Mandatory) represents the identifier of the SPDM Provider actor. No specific format is defined at this stage of the document for such an identifier. Section 4 defines the format for some specific contexts, but other documents may define the format for other contexts.
- The supportedProtocols parameter (Optional) is a list representing the various protocols that are supported by the SPDM Provider actor. Each item of this list represents a single protocol in a specific version. The items contain the following parameters:
  - The protocolId parameter (Mandatory) represents the identifier of the application protocol that might be used by the Discovering actor to dialog with the SPDM Provider actor. No specific format is defined at this stage of the document for such an identifier. Section 4 defines the format for some specific contexts, but other documents may define the format for other contexts.
  - The protocolVersion parameter (Mandatory) represents the specific version of this application protocol. No specific format is defined at this stage of the document for such an identifier. Section 4 defines the format for some specific contexts, but other documents may define the format for other contexts.
  - The protocolInfo parameter (Optional) represents protocol-specific information that MAY be needed by the Discovering actor to dialog with the SPDM Provider actor, using this protocol. If several versions of the same protocol (same protocolId value) are supported by the SPDM Provider actor, then each of the supported versions SHALL be specified in the supportedProtocols list, as a separate item.
- The referAlso parameter (Optional) is a list of additional actors that may be contacted through the Service Protocol Discovery Mechanism. Those actors may be partners or delegates (i.e. acting on behalf) of the SPDM Provider actor. The Service Protocol Discovery Mechanism MAY then be used to contact one or several of those actors to discover their capabilities, and then involve them in the business use cases if required.
  Each item of this list represents a single actor. The items contain the following parameters:
  - The additionalProviderURL parameter (Mandatory) represents the URL of an additional actor to which the Service Protocol Discovery Mechanism could be applied. This URL SHALL comply with the format specified in section 3.1.1 (and section 4.6.1 when applied to [GP SM]). It SHALL contain the same values for the query parameters as the one present in the URL initially requested, but MAY also contain other query parameters added by the SPDM Provider actor.
    The referAlso parameter differs from the redirection codes usage described in section 3.1.2.1 in that redirection simply provides a different endpoint of the same SPDM Provider actor whereas the referAlso parameter provides endpoints of other actors that are partners or act on behalf of the SPDM Provider actor.

The supportedProtocols parameter MAY be alone in the HTTP response body (no referAlso parameter). The supported protocols of the SPDM Provider actor are declared to the Discovering actor and it is the Discovering actor’s responsibility to directly contact the SPDM Provider actor using one of the discovered protocols.

The referAlso parameter MAY be alone in the HTTP response body (no supportedProtocols parameter). If so, the Discovering actor SHOULD contact one or several of the other actors to get more knowledge on the ecosystem.

Both the supportedProtocols and the referAlso parameters MAY be present in the HTTP response body. If so, the supported protocols of the SPDM Provider actor are declared to the Discovering actor but the Discovering actor MAY contact the other actors to get more knowledge on the ecosystem, if required.

If neither the supportedProtocols nor the referAlso parameters are provided, then the Discovering actor has no specific information on the SPDM Provider actor’s protocols or on the related actors.
3.1.2.3 Error Management

In various cases, the SPDM Provider actor MAY NOT be able or MAY NOT want to provide detailed information to the Discovering actor. This is for example the case:

- If some mandatory parameters are missing in the HTTP REST request,
- If the SPDM Provider actor is not managing the Secure Component identified in the HTTP REST request,
- If the Discovering actor is not trusted by the SPDM Provider actor.

If so, the SPDM Provider actor MAY provide a valid System Protocol Discovery Mechanism response (see section 0) that contains an empty supportedProtocols list (Editor's note: Should we use the HTTP codes instead, e.g.: 400 Bad Request?).

The SPDM Provider actor MAY also choose NOT to reply to the HTTP REST request.

3.1.3 Security

No specific security mechanism is mandated by this specification. However, the security to be applied to the SPDM mechanism fully depends on the scheme (http or https) of the URL used for the HTTP REST request.

Refer to section 0, and in particular to section 3.2.4 for the description of this URL, and for the security associated to the HTTP REST request.

3.1.4 Version management

3.1.4.1 Version numbering

A version is a string following the "<major>.<minor>.<maintenance>" format, where <major>, <minor> and <maintenance> are integer values.

The version field SHOULD match the version of the System Protocol Discovery Mechanism specification itself. There is one exception to this rule though, in case of a maintenance update of the System Protocol Discovery Mechanism specification: if the protocol itself is untouched in the maintenance update (no changes in the JSON or the HTTP REST request) and the logic related to the protocol remains unchanged, the version will remain the same for two different releases of the System Protocol Discovery Mechanism specification. This might be the case for example if the updated specification only contains clarifications.

As soon as there is a change to the actual protocol, the version of the protocol will also be updated.

The mapping between available specifications and protocols looks like this:

<table>
<thead>
<tr>
<th>Specification version</th>
<th>Protocol version</th>
<th>version JSON field content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1.0.0</td>
<td>Version 1.0.0</td>
<td>&quot;1.0.0&quot;</td>
</tr>
</tbody>
</table>
3.1.4.2 Version negotiation

Since the Discovering actor and the SPDM Provider actor might support different versions of the System Protocol Discovery Mechanism, it’s necessary to perform version negotiation to agree on which commonly supported version to use during communication.

- If the SPDM Provider actor supports the version number presented by the Discovering actor in the HTTP REST request, then the SPDM Provider actor SHALL return a standard response (according to section 0) with the same version number in the version field. The supportedVersions field SHALL however NOT be present.

- If the SPDM Provider actor does not support the exact version that is indicated by the Discovering actor in the HTTP REST request (see section 3.1.1), then the response of the SPDM Provider actor depends on the supported versions:
  - If the SPDM Provider actor supports only versions of the System Protocol Discovery Mechanism that are strictly greater than the one provided by the Discovering actor in the HTTP REST request (see section 3.1.1), then the SPDM Provider actor SHALL return an "HTTP 501 Not Implemented" code as HTTP response. The body of the HTTP response SHALL be empty.
  - Otherwise, the SPDM Provider actor SHALL return a standard response (according to section 0) with the following constraints:
    - The version field SHALL contain the highest version that is supported by the SPDM Provider actor, but that is lower than the version provided by the Discovering actor. In other words, the version number returned by the SPDM Provider actor SHALL be as high as possible, but SHALL NOT be greater than the one provided in the HTTP REST request.
    - The supportedVersions field SHALL contain the full list of versions that are supported by the SPDM Provider actor.
    - The other fields of the HTTP response body SHALL be provided, following the System Protocol Discovery Mechanism version set by the SPDM Provider actor in the version field.

If the Discovering actor does not support this version, then it is the Discovering actor responsibility to send another HTTP REST request with a version number that is supported by both actors.

3.1.4.3 Examples

Below are examples of responses addressing the various examples of SPDM version negotiation.

Assuming that the Discovering actor supports both the version 1.0.0, 1.1.0 and 1.2.0, then the HTTP REST request SHALL be using the highest version number:

```
spdm/protocols?version=1.2.0&requesterId=1.2.3.4&secureComponentType=SEE&secureComponentId=010203040506
```

- If the SPDM Provider supports this version 1.2.0, the response SHALL be:

  ```
  {
    "version": "1.2.0",
    "providerId": "5.6.7.8",
    "supportedProtocols": [...]
  }
  ```

  With the rest of the response being compliant with version 1.2.0 of this specification.
• If the SPDM Provider only supports version 1.3.0 and 2.0.0, the response SHALL be a "HTTP 501 Not Implemented", with an empty body.

• If the SPDM Provider only supports version 1.0.0, 1.0.1 and 1.1.0, the response SHALL be using the highest version number supported by the SPDM Provider:

```json
{
    "version": "1.1.0",
    "supportedVersions": ["1.0.0", "1.0.1", "1.1.0"],
    "providerId": "5.6.7.8",
    "supportedProtocols": [...]  
}
```

With the rest of the response being compliant with version 1.1.0 of this specification.

As the Discovering actor also supports version 1.1.0, then the Discovering actor can understand the response provided by the SPDM Provider.

In case the Discovering actor would not have supported version 1.1.0, then he MAY have sent again a HTTP REST request with a version number that matches one of the versions supported by the SPDM Provider: for example version 1.0.0.
3.2 Building the SPDM URLs

The full URL corresponding to the System Protocol Discovery Mechanism REST request is named the System Protocol Discovery Mechanism URL (or SPDM URL).

Following the definition of the SPDM HTTP REST request (see section 3.1.1), this SPDM URL contains information that are linked to the Secure Component (for example, the identifier of the Secure Element or of the TEE) and information on the Discovering actor itself (for example, the Discovering actor identifier).

The SPDM URL is consequently built by this Discovering actor using:

- Information that the Discovering actor will know (its identifier, the SPDM version; the type of Secure Component) or ask to the Secure Component (the Secure Component identifier),
- And an URL that correspond to the SPDM Provider domain. This URL is called the Base URL.

3.2.1 Format of Base URLs

It is RECOMMENDED that Base URLs respect the following format:

```
{http|https}://<Actor Domain and Path>[?<additional query parameters>]
```

Where:

- The URL scheme is either http or https (see section 3.2.4).
- The `<Actor Domain and Path>` is the location domain and path of the owner of the URL, to be used as the domain and path for the SPDM URL.
- Additional query parameters MAY (Optional) be provided in the `<additional query parameters>` part of the Base URL, each individual query parameter respecting the following format "<tag>=<value>", and being separated by a '" character.
- The entire Base URL only includes displayable characters (alphabetic, numerical and special) or space in the range from '0x20' to '0x7E' of the ASCII encoding as defined in [ISO 8859-1].

3.2.2 Getting Base URLs from Secure Components

Base URLs MAY be obtained by any mean by the Discovering actor. However, the Secure Components themselves may store such Base URLs which may then be used by a Discovering actor to build the SPDM URL. This is the case, for example, for the Security Domain Manager URL (see section 4.4) or for the DLOA Registrar URL (refer to [GP DLOA]) in Secure Elements, or for the Security Domain Authority URL (see section 4.5) in Trusted Execution Environments, and that may serve as Base URLs.

The provisioning of those Base URLs in the Secure Components, as well as the way to retrieve them from the Secure Components are out of scope of this specification.

The encoding of the Base URL when stored and then delivered by the Secure Component is however RECOMMENDED to be in ASCII format, coded on one byte and left-justified (see [ISO 8859-1]). As an example, the Base URL "https://myCompany.com/myPath" will be encoded as follows:

"68747470733A2F2F6D79436F6D70617468"
3.2.3 Creating a SPDM URLs from a Base URLs

When building a SPDM URL from a Base URL, the Base URL scheme, actor domain and path and additional query parameters SHALL be used without any modification. The Discovering actor SHALL then insert the required information (the spdm/protocols path and the version, requesterId, secureComponentType and secureComponentId query parameters) so that the SPDM URL complies with the HTTP REST request format defined in section 3.1.1.

In case the Base URL contains additional query parameters, then the Discovering actor SHALL also manage the location of the '?' character so that when building the SPDM URL, the domain and path part of the Base URL be separated from the additional query parameters part of this Base URL.

Assuming that the Base URL is:

{http|https}://<Actor Domain and Path>[?<additional query parameters>]

then the format of the SPDM URL built by the Discovering actor SHALL be:

{http|https}://<Actor Domain and Path>/spdm/protocols?version=<version>
&requesterId=<Discovering actor Id>&secureComponentType={SE|TEE}
&secureComponentId=<SC Id>[&<additional query parameters>]

3.2.4 Security of SPDM URLs

The scheme of the Base URL defines the security that SHALL be applied to the System Protocol Discovery Mechanism:

- If the scheme of the Base URL is "http": TLS security is NOT MANDATED between the Discovering actor and the SPDM Provider actor.

- If the scheme of the Base URL is "https": TLS security, with server authentication, SHALL be set up between the Discovering actor and the SPDM Provider actor prior to send the HTTP REST request. TLS in mutual authentication is NOT MANDATED.

  SPDM Provider implementations SHOULD set the minimum protocol version of TLS to v1.2. Servers SHOULD NOT accept TLS v1.1 or any older protocol versions.

The SPDM Provider actor MAY support one or both URL schemes.

Upon business agreement, the Discovering actor or the SPDM Provider actor MAY impose a complementary security mechanism to be set up between the two actors for the System Protocol Discovery Mechanism.

The security mechanisms that may be potentially required for any further application dialog between the Discovering actor and the SPDM Provider actor are out of scope of this specification, and are independent of the security mechanism that may have been set up for the System Protocol Discovery Mechanism.
4 Usage in Specific Contexts

The following sections specify the particular rules and usages that apply to the System Protocol Discovery Mechanism in some specific contexts. Other documents may define rules and usages for other contexts.

4.1 Usage for GlobalPlatform System

This section specifies the particular rules and usages that apply to the System Protocol Discovery Mechanism when used in the context of the GlobalPlatform System.

Identifiers:

The requesterId query parameter of the HTTP REST request SHALL be an Object Identifier (OID) in a BER encoding format (as specified in section 8.19, “Encoding of an object identifier value” of [ISO 8825-1], without the Tag and Length bytes) or in the numerical dot notation format (as specified in section 2.15, “Object identifier” of [RFC 1778]).

The providerId parameter of the HTTP response SHALL be an Object Identifier (OID) in a BER encoding format (as specified in section 8.19, “Encoding of an object identifier value” of [ISO 8825-1], without the Tag and Length bytes) or in the numerical dot notation format (as specified in section 2.15, “Object identifier” of [RFC 1778]).

For example, the GlobalPlatform OID is "2A864886FC6B" in the BER format, and is "1.2.840.114283" in the numerical dot notation.

OIDs of companies are usually registered to the Internet Assigned Numbers Authority (IANA) organization.

4.2 Usage for Secure Element Issuers

This section specifies the particular rules and usages that apply to Secure Element Issuers for the usage of the System Protocol Discovery Mechanism.

As described in the GPCS UICC Configuration [GP UICC Config] and in the GPCS Secure Element Configuration [GP SE Config], the Issuer Security Domain SHALL be personalized so that it provides the IIN and CIN information as part of the answer to a GET DATA command with tags '42' and '45'.

This will enable the Discovering actor to query those data from the Secure Element, and to set the Card Unique Data (in its hexadecimal string representation) in the secureComponentId parameter of the HTTP REST request (together with the "SE" value to the secureComponentType parameter). The Card Unique Data is the concatenation of the Issuer Identification Number (IIN) and the Card Identification Number (CIN).

Note that the CUD information, or CIN and IIN separately, MAY be obtained by other means.
4.3 Usage for Trusted Execution Environment Issuers

This section specifies the particular rules and usages that apply to Trusted Execution Environment Issuers for the usage of the System Protocol Discovery Mechanism.

As described in the GPD TEE Administration Framework [GP TAF], the TEE SHALL be personalized with a device ID (gpd.tee.deviceID). Any client application can retrieve this device ID using the unprivileged audit operation Retrieve TEE Definition as defined in [GP TAF].

This will enable the Discovering actor to query this data from the TEE, and to set it (in its string representation according to [RFC 4122]) in the secureComponentId parameter of the HTTP REST request (together with the "TEE" value to the secureComponentType parameter).

Note that for TEE not supporting the [GP TAF] specification, the ability to get the device ID directly from the TEE is implementation dependent. TEE device ID MAY then be obtained by other means.

Moreover, support of a unique identifier such as the device ID for TEE not supporting the [GP TAF] specification is also implementation dependent. As a consequence, support of the SPDM mechanism for TEE not supporting [GP TAF] is also implementation dependent.

4.4 Usage for (Supplementary) Security Domains Managers in Secure Elements

This section specifies the particular rules and usages that apply to Security Domain Managers of Secure Elements for the usage of the System Protocol Discovery Mechanism.

As described in [GP Amd A], Security Domain Managers SHALL personalize their Security Domains so that they provide a Security Domain Manager URL as part of the answer to a GET DATA command with tag '5F50'. This is also true for Secure Element Issuers that SHALL personalize the Issuer Security Domain of their SE.

If the Security Domain Manager URL is intended to be used as a Base URL for a Discovering actor to build a SPDM URL, then the recommendations stated in section 0 SHALL be applied to the Security Domain Manager URLs:

- The format of the Security Domain Manager URL as a Base URL,
- The encoding for the storage of this URL in the Secure Element,
- The encoding for the delivery of this URL by the SE,
- And the way to build the SPDM URL

SHALL comply with the format and encoding defined in section 0.

It MAY be acceptable that this Security Domain Manager URL information can be obtained by other means than querying the Secure Element.
4.5 Usage for Security Domains Authorities in Trusted Execution Environments

This section specifies the particular rules and usages that apply to Security Domain Authorities of Trusted Execution Environments for the usage of the System Protocol Discovery Mechanism.

As described in [GP TAF], Authorities responsible for Security Domain MAY personalize their Security Domains so that they provide an Authority URL as part of the answer to the unprivileged audit operation Retrieve SD Definition (in the Authority field) as defined in [GP TAF]. Note that it is required to know the SD UUID in order to use this audit operation.

If the Security Domain Authority URL is intended to be used as a Base URL for a Discovering actor to build a SPDM URL, then the recommendations stated in section 0 on the format of the Security Domain Authority URL as a Base URL, and the way to build the SPDM URL SHALL be applied.

The encoding for the storage of this URL in the TEE and the encoding for the delivery of this URL by the TEE SHALL comply with [GP TAF].

It MAY be acceptable that this Security Domain Authority URL information can be obtained by other means than querying the TEE.

4.6 Usage for "GlobalPlatform System Messaging for Mobile-NFC Services Management" for Secure Elements

The following sections specify the particular rules and usages that apply to the System Protocol Discovery Mechanism when applied to the discovery of the protocol defined in [GP SM], for Secure Elements.

4.6.1 HTTP REST Request

The Discovering actor SHALL:

1. Get the Secure Element identifier, as described in section 4.2,
2. Retrieve the Security Domain Manager URL, as described in section 4.4,
3. Build the System Protocol Discovery Mechanism URL from this Security Domain Manager URL, according to the process described in section 0, and using its requester id as described in section 4.1 and the information obtained during the previous steps,
4. Perform a HTTP REST request on this System Protocol Discovery Mechanism URL, following section 3.1.1.

As an example, the HTTP REST request URL might be:

http://myCompany.com/spdm/protocols?version=1.0.0&requesterId=1.2.3.4&secureComponentType=SEE&secureComponentId=010203040506

Where:

- The retrieved Security Domain Manager URL is:
  http://myCompany.com
• The System Protocol Discovery Mechanism URL being built by the Discovering actor, based on the Security Domain Managers URL, adding the version, requesterId, secureComponentType and secureComponentId query parameters, and complementing the Security Domain Manager URL path with the HTTP REST path "spdm/protocols".

As another example, it could also be:

https://myCompany.com/discovery/spdm/protocols?version=1.0.0
&requesterId=2A0304&secureComponentType=SEE
&secureComponentId=010203040506&additionalParam=42

Where:

• The retrieved Security Domain Manager URL is:
  https://myCompany.com/discovery?additionalParam=42

• The System Protocol Discovery Mechanism URL being built by the Discovering actor, based on the Security Domain Managers URL, adding the version, requesterId, secureComponentType and secureComponentId query parameters, and complementing the Security Domain Manager URL path with the HTTP REST path "spdm/protocols".

4.6.2 HTTP Response

The protocolId parameter SHALL be set to "gps-mm", standing for GlobalPlatform System Mobile Messaging.

The protocolVersion parameter SHALL be set to the version of [GP SM] document described in the "Revision History" section of [GP SM], in the format "<major>.<minor>" or "<major>.<minor>.<maintenance>".

The protocolInfo parameter SHALL contain a functionGroups parameter (Mandatory) that is a list of items, each of them representing a specific function group provided by the Security Domain Manager. Each item of the list SHALL contain the following parameters:

• The functionGroup parameter (Mandatory), representing the name of the function group that is provided by the Security Domain Manager. This name SHALL comply with the Service names defined in the "WSDL Binding" section of [GP SM].

• The endPoint parameter (Mandatory), representing the Web Service endpoint where the functions of the function group can be called, i.e. where to locate the Service in the Web Service terminology.

If the Security Domain Manager supports several versions of [GP SM], then each supported version SHALL be specified as a separate items of the supportedProtocols list. The endpoint corresponding to a specific function group that is present in several supportedProtocols items (i.e. present for different versions of [GP SM]) MAY be identical (Security Domain Manager implementation dependent).
Following usages depicted in sections 4.1, 4.2, and 4.4, here is an example of an HTTP response for the support of version 1.1.2 and version 1.2.0 of some function groups (SEInfo and CAInfo) of [GP SM]. The JSON Schema representation of this format is provided in section A.2.

```json
{
  "version": "1.0.0",
  "providerId": "5.6.7.8",

  "supportedProtocols": [
    {
      "protocolId": "gps-mm",
      "protocolVersion": "1.1.2",
      "protocolInfo": {
        "functionGroups": [
          {
            "functionGroup": "SEInfo",
            "endPoint": "https://myCompany.com/systems-messaging/SEInfo"
          },
          {
            "functionGroup": "CAInfo",
            "endPoint": "https://myCompany.com/systems-messaging/CAInfo_1.1.2"
          }
        ]
      }
    },
    {
      "protocolId": "gps-mm",
      "protocolVersion": "1.2.0",
      "protocolInfo": {
        "functionGroups": [
          {
            "functionGroup": "SEInfo",
            "endPoint": "https://myCompany.com/systems-messaging/SEInfo"
          },
          {
            "functionGroup": "GlobalServiceManagement",
            "endPoint": "https://myCompany.com/systems-messaging/CAInfo_1.2.0"
          }
        ]
      }
    }
  ]
}
```

As another example, here is an example of an HTTP response that references two partners of the Security Domain Manager:

```json
{
  "version": "1.0.0",
  "providerId": "CE0708",

  "referAlso": [
    {
      "additionalProviderURL": "http://additionalActor1/"
    }
  ]
}
```
Both the protocols description and the partner/delegates reference parts can be returned in the same response, as shown in the example below:

```json
{
    "version": "1.0.0",
    "providerId": "5.6.7.8",

    "supportedProtocols": [
        {
            "protocolId": "gps-mm",
            "protocolVersion": "1.1.2",
            "protocolInfo": {
                "functionGroups": [
                    {
                        "functionGroup": "SEInfo",
                        "endPoint": "https://myCompany.com/systems-messaging/SEInfo"
                    }
                ]
            }
        }
    ],

    "referAlso": [
        {
            "additionalProviderURL": "http://additionalActor1/"
        }
    ]
}
```
Annex A

JSON Schema for HTTP Response

The following sections present the JSON Schema of the HTTP response body, according to the JSON Schema: core definitions and terminology Draft 04 [JSON Schema].

A.1 Generic Definition

```json
{
    "title": "System Protocol Discovery Mechanism: JSON Schema of the HTTP response Body",
    "type": "object",
    "$schema": "http://json-schema.org/draft-04/schema#",
    "properties": {
        "version": {
            "type": "string",
            "description": "Version of the System Protocol Discovery Mechanism that is used by the SPDM Provider actor in this HTTP response."
        },
        "supportedVersions": {
            "type": "array",
            "description": "List of version of the System Protocol Discovery Mechanism that are supported by the SPDM Provider actor.",
            "items": {
                "type": "string"
            }
        },
        "providerId": {
            "type": "string",
            "description": "Identifier of the SPDM Provider actor."
        },
        "supportedProtocols": {
            "type": "array",
            "description": "List of the various protocols that are supported by the SPDM Provider actor. Each item of this list represents a single protocol, in a specific version.",
            "items": {
                "type": "object",
                "properties": {
                    "protocolId": {
                        "type": "string",
                        "description": "Identifier of the application protocol that might be used by the Discovering actor to dialog with the SPDM Provider actor."
                    },
                    "protocolVersion": {
                        "type": "string",
                        "description": "The specific version of this application protocol."
                    },
                    "protocolInfo": {
                        "type": "object",
                        "description": "Protocol-specific information that may be needed by the Discovering actor to dialog with the SPDM Provider actor, using this protocol."
                    }
                }
            }
        }
    }
}
```
},
"additionalProperties": false,
"required": [
  "protocolId",
  "protocolVersion"
]
}
],
"referAlso": {
  "type": "array",
  "description": "List of the additional actors that may be contacted through the Service Protocol Discovery Mechanism. Those actors may be partners or delegates (i.e. acting on behalf) of the SPDM Provider actor."
  "items": {
    "type": "object",
    "properties": {
      "additionalProviderURL": {
        "type": "string",
        "description": "URL of the additional actor onto which applying directly the Service Protocol Discovery Mechanism."
      }
    }
  }
},
"required": [
  "additionalProviderURL"
]
}
]
},
"additionalProperties": false,
"required": [
  "version",
  "providerId"
]
}
A.2 Applied to GlobalPlatform System Messaging for Mobile-NFC Services Management for Secure Elements

{
    "title": "System Protocol Discovery Mechanism: JSON Schema of the HTTP response Body",
    "type": "object",
    "$schema": "http://json-schema.org/draft-04/schema#",
    "properties": {
        "version": {
            "type": "string",
            "description": "Version of the System Protocol Discovery Mechanism that is used by the SPDM Provider actor in this HTTP response."
        },
        "supportedVersions": {
            "type": "array",
            "description": "List of version of the System Protocol Discovery Mechanism that are supported by the SPDM Provider actor."
        },
        "providerId": {
            "type": "string",
            "description": "Identifier of the SPDM Provider actor."
        },
        "supportedProtocols": {
            "type": "array",
            "description": "List of the various protocols that are supported by the SPDM Provider actor. Each item of this list represents a single protocol, in a specific version."
        },
        "protocolId": {
            "type": "string",
            "description": "Identifier of the application protocol that might be used by the Discovering actor to dialog with the SPDM Provider actor."
        },
        "protocolVersion": {
            "type": "string",
            "description": "The specific version of this application protocol."
        },
        "protocolInfo": {
            "type": "object",
            "description": "Protocol-specific information that may be needed by the Discovering actor to dialog with the SPDM Provider actor, using this protocol."
        },
        "functionGroups": {
            "type": "array",
            "description": "List of the specific function groups provided by the SPDM Provider actor."
        }
    }
}
"items": {
  "type": "object",
  "properties": {
    "functionGroup": {
      "type": "string",
      "description": "The name of the function group that is provided by the SPDM Provider actor. This name shall comply with the Service names defined in the WSDL Binding section of the System Messaging specification for Mobile-NFC Service Management document."
    },
    "endPoint": {
      "type": "string",
      "description": "The Web Service endpoint where the functions of the function group can be called, i.e. where to locate the Service in the Web Service terminology."
    }
  }
},
"additionalProperties": false,
"required": ["functionGroup", "endPoint"]
},
"additionalProperties": false,
"required": ["functionGroups"]
},
"additionalProperties": false,
"required": ["protocolId", "protocolVersion"]
},
"referAlso": {
  "type": "array",
  "description": "List of the additional actors that may be contacted through the Service Protocol Discovery Mechanism. Those actors may be partners or delegates (i.e. acting on behalf) of the SPDM Provider actor.",
  "items": {
    "type": "object",
    "additionalProviderURL": {
      "type": "string",
      "description": "URL of the additional actor onto which applying directly the Service Protocol Discovery Mechanism."
    }
  }
},
"required": ["additionalProviderURL"]


```json
{}
{
}
,
"additionalProperties": false,
"required": [ 
  "version",
  "providerId"
]
}
```