ETSI TS 103 544-16 V1.3.0 (2017-10)

TECHNICAL SPECIFICATION

Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 16: Application Developer Certificates

CAUTION

The present document has been submitted to ETSI as a PAS produced by CCC and approved by the ETSI Technical Committee Intelligent Transport Systems (ITS).

CCC is owner of the copyright of the document CCC-TS-044 and/or had all relevant rights and had assigned said rights to ETSI on an "as is basis". Consequently, to the fullest extent permitted by law, ETSI disclaims all warranties whether express, implied, statutory or otherwise including but not limited to merchantability, non-infringement of any intellectual property rights of third parties. No warranty is given about the accuracy and the completeness of the content of the present document.

Reference

DTS/ITS-88-16

Keywords

interface, ITS, PAS, smartphone

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Intelligent Transport Systems (ITS).

The present document is part 16 of a multi-part deliverable. Full details of the entire series can be found in part 1 [i.1].

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document is part of the MirrorLink® specification which specifies an interface for enabling remote user interaction of a mobile device via another device. The present document is written having a vehicle head-unit to interact with the mobile device in mind, but it will similarly apply for other devices, which provide a colour display, audio input/output and user input mechanisms.

MirrorLink provides the ability to run certified applications on MirrorLink server devices that can be launched from the MirrorLink Client device. In order to improve safety and ensure a quality user experience, an application certification program is implemented that will control which applications can be used with MirrorLink in drive on in non-drive situations. Application developers will be able to use specific application development certificates, which simplifies the development of applications on the one side, but which will be usable only on a small set of MirrorLink Server devices - as well as a potentially restricted set of MirrorLink Client devices.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

[1] IETF RFC 3281: "An Internet Attribute Certificate Profile for Authorization", April 2002, http://www.ietf.org/rfc/rfc3281.txt.

[2] IETF RFC 2459: "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", January 1999, http://www.ietf.org/rfc/rfc2459.txt .

[3] IETF RFC 2560: "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", June 1999, http://tools.ietf.org/html/rfc2560 .

[4] ETSI TS 103 544-9 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink® ; Part 9: UPnP Application Server Service".

[5] ETSI TS 103 544-14 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 14: Application Certificates" .

[6] ETSI TS 103 544-10 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 10: UPnP Client Profile Service" .

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TS 103 544-1 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 1: Connectivity".

3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ACMS	Application Certification Management System
BT	Bluetooth
ML	MirrorLink
OCSP	Online Certificate Status Protocol
RFB	Remote Framebuffer
UPnP	Universal Plug and Play
USB	Universal Serial Bus
VNC	Virtual Network Computing

4 Developer Application Concept

MirrorLink provides the ability to run certified applications on MirrorLink server devices that can be launched from the MirrorLink client device. In order to improve safety and ensure a quality user experience, an application certification program is implemented that will control which applications can be used with MirrorLink in drive on in non-drive situations. Application developers will be able to use specific application development certificates, which simplifies the development of applications on the one side, but which will be usable only on a small set of MirrorLink Server devices - as well as a potentially restricted set of MirrorLink Client devices.

Each application under development, which can be uniquely identified by a platform specific application identifier (App ID), will come with an Application Development Certificate (App Dev Certificate), which contains the App ID; necessary application information, provided to the MirrorLink Client (App Info); and the Developer ID (Dev ID). The Application Development Certificate is self-signed by either the application developer or the MirrorLink Server's software development kit.

The MirrorLink Server will use the information from the App Development Certificate to validate the MirrorLink Application, and to link it to the Developer Identifier Certificate (Dev ID Certificate). The Dev ID Certificate contains a unique Developer Identifier (Dev ID), and one or more Server Device Identifiers (Server Device IDs) for which the Dev ID Certificate is valid. An optional list of Client Device Identifiers (Client Device IDs) defines a black list of client devices, for which the Dev ID Certificate is not valid.

As shown in Figure 1, the App Dev and the Dev ID Certificates are stored on the MirrorLink Server Device. It is the responsibility of the MirrorLink Server to check, whether the Dev ID Certificate has not been revoked and whether it is valid for the MirrorLink Server and Client combination. In case the App Dev Certificate is valid, the corresponding MirrorLink application will be presented to the MirrorLink Client Device as an application coming with a certificate distributed by CCC.

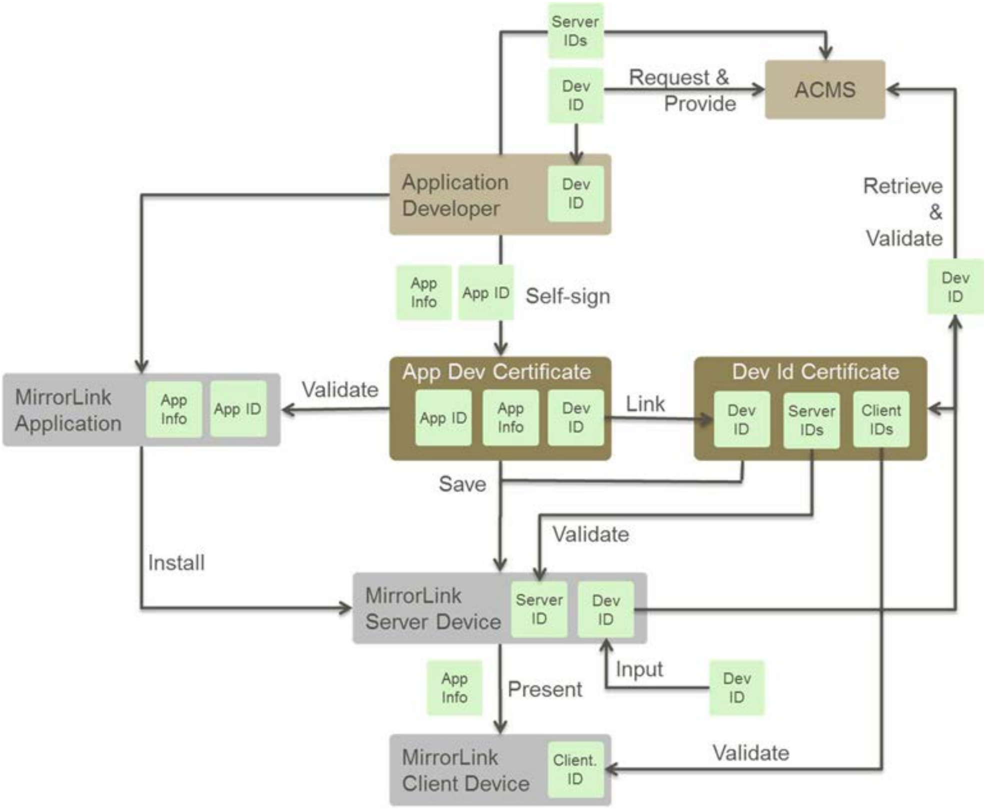

Figure 1: Application Developer Certification Architecture (MirrorLink Server View)

A MirrorLink Client will not see the difference from any regular non-development version, besides a different signing entity name and an additional X.509 v3 extension.

Support for development applications as described above may be restricted to specific MirrorLink Server Developer devices; those shall be made available to application developers. Therefore, a regular MirrorLink Server device may NOT be able to run development applications as certified applications.

5 Application Developer Certificate Structure

5.1 Application Development Certificate

5.1.1 General

MirrorLink Application Development Certificates shall be a public key X.509 version 3 certificate as specified in [1].

The certificate is a self-signed certificate. The signing authority shall not set an expiration date of longer than 1 month from the date of signing.

Application Development Certificate shall use 2048-bit RSA keys with SHA-256 or SHA-512 signature algorithms.

5.1.2 Extension Header

The X.509 extension header shall have the following format:

```
X509v3 extensions:
CCC-MirrorLink-Developer-Id Extension:
        extnId:    1.3.6.1.4.1.41577.3.1
        critical:  no
        extnValue: DER:OCTET STRING
   CCC-MirrorLink Extension:
        extnId:    1.3.6.1.4.1.41577.2.1
        critical:  no
        extnValue: DER:<DER encoded XML, as specified below>
```

5.1.3 Extension Values

5.1.3.1 CCC-MirrorLink-Developer-Id

Developer Id, as provided from the Application Certification Management System (ACMS), shall be formatted as a character string of up to 40 alphanumeric characters (‘a'-‘z', ‘0'-‘9').

5.1.3.2 CCC-MirrorLink Extension Value

The DER encoded XML of the application information, as specified in [5].

The Signing Entity Name of application development certificates shall be "DEVELOPER".

5.2 Developer Identification Certificate

5.2.1 General

The MirrorLink Dev ID Certificate shall be a public key X.509 version 3 certificate as specified in [1].

The certificate shall be signed by the CCC's Root Certificate. A hierarchy of certification authorities (CAs) may be used for Dev ID certificates. In case intermediate CAs are used, the entire certificate chain up to the root CA shall be provided to the MirrorLink Server together with the Dev ID certificate. Any intermediate certificate shall not have an expiration date of more than 1 year from the date of signing.

The Intermediate certificate, which is signed by the CCC root CA, shall have a Common Name (CN) in the issuer information, identical to "ACMS CA"; otherwise the certificate shall not be accepted. A valid example issuer information is given below:

```
Issuer: O=Car Connectivity Consortium, CN=ACMS CA
```

Any intermediate certificate shall use 4096-bit RSA keys with SHA-512 signature algorithms.

5.2.2 Extension Header

The X.509 extension header shall have the following format:

```
X509v3 extensions:
    CCC-MirrorLink-Developer-Id Extension:
        extnId:    1.3.6.1.4.1.41577.3.1
        critical:  no
        extnValue: DER:OCTET STRING
    CCC-MirrorLink-Developer-Server-Ids Extension:
        extnId:    1.3.6.1.4.1.41577.3.2
        critical:  no
        extnValue: DER:OCTET STRING
    CCC-MirrorLink-Client-Manufacturer-Ids Extension:
        extnId:    1.3.6.1.4.1.41577.3.3
        critical:  no
        extnValue: DER:OCTET STRING
```

5.2.3 Extension Values

5.2.3.1 CCC-MirrorLink-Developer-Id

Developer Id, as provided from the Application Certification Management System (ACMS), shall be formatted as a character string of up to 40 alphanumeric characters (‘a'-‘z', ‘0'-‘9').

5.2.3.2 CCC-MirrorLink-Developer-Server-Ids

A comma-delimited list of Server Ids, for which the Dev ID certificate is valid; each entry shall be formatted as a string (UTF-8).

Server IDs are the IMEI/IMEISV number or version 5 UUID derived from an equivalent unique identifier of the MirrorLink Server devices on which development applications can be used, as defined in the platform specific specification.

5.2.3.3 CCC-MirrorLink-Client-Manufacturer-Ids

Comma-separated list of MirrorLink Client manufacturer ids, for which the Dev ID certificate is not valid (black list). Each entry shall be formatted as a string (UTF-8).

Each list entry represents a manufacturer name, and shall match the manufacturer name (as provided from the UPnP Client Profile service [6]) or the *AppCertFilter*'s entity name (as used in the UPnP Application Server service [4]).

5.3 Root Certificate

The signing certification authority's Root Certificate, a hash of it or a hash of its public key shall be stored in the MirrorLink Server. Access to the certificate's public key shall be read-only.

Expiration date of the root certificate shall be 20 years from the date of signing.

Root certificate shall use 4096-bit RSA keys with SHA-512 signature algorithms. The root certificate shall be identical to the DAP root certificate.

6 Developer Identification Certificate Life Cycle

6.1 Certificate Retrieval and Validation

6.1.1 Certificate Retrieval

The MirrorLink Server shall use HTTP-GET to obtain the MirrorLink Dev ID certificate from the Application Certification Management System using the following URL:

http://acms.carconnectivity.org:80

The following GET command shall be used to obtain the application certificate:

```
GET /obtainDeveloperCertificate.html?
certificateVersion=1.0&
developerID=<Developer Identifier>&
serverID=<Server Identifier>
HTTP/1.1<CR><LF>
Host: acms.carconnectivity.org:80<CR><LF>
<CR><LF>
```

The provided *serverID* shall uniquely identify a particular MirrorLink Server device. The MirrorLink Server shall use the IMEI/IMEISV number (or equivalent unique identifier) of the MirrorLink Server device for *serverID*. Devices without an IMEI/IMEISV number shall not be used for Application development at this time.

The MirrorLink Server shall retrieve the Dev ID Certificate before it can use any self-signed application development certificates. The MirrorLink Server shall not retrieve the Dev ID Certificate, unless the device is going to be used for MirrorLink application development.

The ACMS's HTTP Server shall return the Dev ID certificate and the entire chain of intermediate certificates, Base 64 encoded. Blank lines separate the certificates, starting from the certificate signed directly by the CCC root CA.

Otherwise it shall provide one of the following error codes:

Table 1: Certificate Retrieval Error Codes

HTTP Error Code	CCC Error Code	Description
1xx	N/A	MirrorLink Server shall handle the HTTP response in compliance with the HTTP protocol (implementation specific).
200	N/A	MirrorLink Server shall validate the received application certificate, in accordance with clause 0.
2xx	N/A	MirrorLink Server shall handle the HTTP response in compliance with the HTTP protocol (implementation specific).
3xx	N/A	MirrorLink Server shall handle the HTTP response in compliance with the HTTP protocol (implementation specific).
400	N/A	Bad request - The request cannot be fulfilled due to bad syntax (e.g. missing, empty or wrongly formatted parameter). The MirrorLink Server shall not retry the request.
4xx	N/A	MirrorLink Server shall not retry the request
500	800	No certificate available for the given parameter The MirrorLink Server should retry the request.
500	801	Certification Database currently offline The MirrorLink Server shall retry between 1h and 24h after the last HTTP-Get attempt.
500	8xx	Reserved for future use The MirrorLink Server should retry the request.
500	900	Certificate has been revoked. The MirrorLink Server shall not retry the request.
500	9xx	Reserved for future use The MirrorLink Server shall not retry the request.
500	xxx	Reserved for future use The MirrorLink Server should retry the request.
5xx	N/A	The MirrorLink Server should retry the request.

If the ACMS HTTP Server returns with an error response other than 200 (Ok response), the MirrorLink Server shall consider the Dev ID certificate as not being available and any development application linked to the developer ID shall be considered a MirrorLink aware-application only. The MirrorLink Server should retry the HTTP-Get request, unless otherwise stated above. If the MirrorLink Server retries the request, it shall retry between 50 % and 100 % of the query period since the last HTTP-Get request. If no automatic retry is provided, the MirrorLink Server shall provide a manual retry option.

NOTE: There are no time constraints in case of manual retry.

6.1.2 Certificate Validation

The validation of Dev ID certificates is following the steps below:

1) Validate the Dev ID certificate and trust chain

2) Validate the MirrorLink Server identifier is in the list of Server IDs, as given in the CCC-MirrorLink-Developer-Server-Id X.509 extension.

3) Validate the MirrorLink Client's manufacturer identifier is not within the black list of certified manufacturer identifiers, as given in the CCC-MirrorLink-Developer-Manufacturer -Ids X.509 extension, as specified in clause 5.2.3.3.

The MirrorLink Server shall execute all certificate validation steps at MirrorLink connection setup, prior to including any development application into any certified application listing.

The MirrorLink Server shall not retry to download a new Dev ID certificate in case any of the following validation steps failed:

- Validation of the trust chain failed
- Validation of the certificate's signature failed
- Validation of the server identifier failed

In case the validation of MirrorLink Client manufacturer failed, the Dev ID certificate shall not include any development application into any certified application listing. It shall include the Dev ID certificate into the regular OCSP checks. Any change to the blacklisted client manufacturer list is provided via a OCSP certificate update, as defined in clause 6.2.4.

In case a MirrorLink Server has multiple Dev ID certificates, with different developer IDs, installed, it shall make any development application available as certified applications, containing the respective validated developer ID.

The MirrorLink Server shall retry to download a new Dev ID certificate between 50 % and 100 % of the query period after the last HTTP-Get attempt in case the following validation steps failed:

- Dev ID certificate is expired

If any of the steps fail, all development applications, linked to the developer ID, shall be considered to be non-certified and the MirrorLink Server shall not add them to the certified application list (*A_ARG_TYPE_CertifiedAppList*).

Applications, which failed validation, may be included in the regular application list (*A_ARG_TYPE_AppList*). In that case, the applications shall not have a trust level of `"Application Certificate"`.

6.1.3 Testing Considerations

For Certification Validation testing purposes during MirrorLink device certification, the MirrorLink Server shall accept the CTS root certificate to validate application certificates distributed by the ACMS. This Test Mode shall not be accessible in production devices.

6.2 Certificate Revocation Checks

6.2.1 Revocation Protocol

The MirrorLink Server shall use the Online Certificate Status Protocol (OCSP) [3] to verify the status of Dev ID certificate. The URI, where the MirrorLink Server shall ask for the certificate status, shall be available from the *AuthorityInfoAccess* (AIA) field, as defined in [2], in the certificate.

The MirrorLink Server shall include a Nonce extension, with a random nonce value, into the OCSP request to prevent any replay attack. OCSP responses shall be signed, with the signature algorithm and key of the issuing certificate. The signature algorithm shall be RSA with at least 2 048 bits with at least SHA-256.

The MirrorLink Server shall use OCSP over HTTP to send and receive OCSP requests and responses. Their formatting is specified in Appendix A of [3].

The MirrorLink Server shall take the following actions for the respective application certificate, in case the *ocspResponseStatus* has a value, indicated below:

- tryLater: Should retry
- internalError: Should retry
- malformedRequest: Shall not send any further OCSP requests
- sigRequired: Shall not send any further OCSP requests
- unauthorized: Shall not send any further OCSP requests

The MirrorLink Server shall take the following actions for the respective application certificate, in case the *ocspResponseStatus* is `successful` and the *certStatus* has a value, indicated below:

- unknown: Shall not send any further OCSP requests;
- good: See clause 6.2.2;
- revoked: See clauses 6.2.3 and 6.2.4.

The MirrorLink Server should retry the OCSP request, in case OCSP response fails validation at least one of the following checks:

- Validation of the certificate trust chain
- Validation of the response signature
- Validation that the nonce value matched the one from the OCSP request

The MirrorLink Server should retry the OCSP request, unless otherwise stated above. If the MirrorLink Server retries the request, it shall retry between 50 % and 100 % of the query period since the last OCSP request. If no automatic retry is provided, the MirrorLink Server shall provide a manual retry option.

NOTE: There are no time constraints in case of manual retry.

6.2.2 Certificate Valid

The MirrorLink Server shall consider a developer ID certificate to be valid, if:

- The OCSP *certStatus* is `"good"`.

6.2.3 Certificate Revoked

The MirrorLink Server shall consider the developer ID certificate to be revoked if:

- The OCSP *certStatus* is `"revoked"` and
- The ACMS returns the HTTP-Get response with Error Code 500/900, when requesting a new certificate.

The MirrorLink Server shall not send any further HTTP-Get and OCSP request for a revoked developer ID certificate.

6.2.4 Certificate Updated

A developer ID certificate shall be updated in the following cases:

1) Current Dev ID certificate expired
2) Server or Client identifier fields in the certificate changed within the ACMS

The MirrorLink Server shall consider the developer ID certificate as to be updated if:

- the OCSP *certStatus* is `"revoked";` and
- the ACMS returns the HTTP-Get response with Error Code 200, when requesting a new certificate.

The retrieved updated application certificate shall be validated (in accordance with clause 6.1.2, including an OCPS request for the updated developer ID certificate.

6.2.5 Testing Consideration

For OCSP testing purposes during MirrorLink device certification, the MirrorLink Server shall accept the CTS root certificate to validate responses from the ACMS. This Test Mode shall not be accessible in production devices.

6.3 Query and Grace Periods

6.3.1 Query Period

The MirrorLink Server shall verify from the ACMS, whether the developer ID certificate has been revoked. Validation shall happen within 50 % and 100 % of the query period, as defined in [5].

Developer ID certificates shall use the same query period as regular application certificates. If the query period is set to 0, the Developer ID shall be checked on MirrorLink connection setup and at least every 24 h, while the MirrorLink connection lasts.

Failure to receive a revocation list update, after the query period, shall invalidate the Dev ID certificate. The MirrorLink Server shall remove any development application immediately from the certified application listing, if the query period expires during a MirrorLink connection. The MirrorLink Server may still provide access to the application certificate via the *appCertificateURL* entry in the UPnP application listing.

6.3.2 Grace Period

The MirrorLink Server shall not allow for any Grace Period for Developer ID certificates.

7 Application Development Certificate Life Cycle

7.1 Certificate Retrieval and Validation

7.1.1 Certificate Retrieval

The MirrorLink Server manufacturer shall provide a mechanism to create and self-sign an application development certificate. The application development certificate shall include the developer identifier from the Dev ID Certificate. To facilitate rapid application development cycles, the application identifier should be set to a dummy value, e.g. `"0000"`. The MirrorLink Server shall ignore the provided application identifier in the self-signed application development certificate.

The MirrorLink Server shall not access the ACMS, using an HTTP-Get request, to retrieve any application development certificate.

7.1.2 Certificate Validation

The validation of Application Development certificates is following the steps below:

1) Validate the application development certificate

2) Validate the developer identifier is identical to the validated one in the Developer ID certificate

NOTE: The application identifier will not be validated, as it is expected to be ignored.

The MirrorLink Server shall execute all certificate validation steps at MirrorLink connection setup, prior including any development application into any certified application listing.

In case the certificate has been validated, the MirrorLink Server shall treat the application as if the certificate has been signed by the CCC, i.e.

- The MirrorLink Server shall apply all rules specified for CCC certified applications in [5].
- The MirrorLink Server shall replace the Entity Name `"DEVELOPER"` by `"CCC"` within any provided *A_ARG_TYPE_AppCertificateInfo* structure.
- The MirrorLink Server shall include them into any response, when the MirrorLink Client is looking for `"CCC"` certified applications.

If any of the steps fail, the development application shall be considered non-certified and the MirrorLink Server Shall not add it to the certified application list (*A_ARG_TYPE_CertifiedAppList*).

Applications, which failed validation, may be included in the regular application list (*A_ARG_TYPE_AppList*). In that case, the application shall not have a trust level of `"Application Certificate"`.

7.1.3 Certificate Update

An application development certificates shall be updated in the following cases:

1) Current application development certificate expired

2) New developer identifier

The MirrorLink Server manufacturer shall provide a mechanism to update a self-signed application development certificate.

On update of an application development certificate, the MirrorLink Server shall immediately validate the application development certificate.

7.2 Certificate Revocation Checks

Application development certificates are not subject to any Revocation Protocol.

The MirrorLink Server shall not use OCSP requests to check with the ACMS the revocation status of any application development certificate.

Annex A (informative): OCSP Request & Response Example

An example OCSP request is given below.

```
OCSP Request Data:
    Version: 1 (0x0)
    Requestor List:
        Certificate ID:
          Hash Algorithm: sha1
          Issuer Name Hash: 8809099A55F562E1EA13273360FFB7F50B76F508
          Issuer Key Hash: BF37183EB53B43AD1D7237E59CE2FC4DF96C7BFC
          Serial Number: 6D
    Request Extensions:
        OCSP Nonce:
            041035DA009D2912E3CEC403D34B319228D9
```

An example OCSP response is given below, which includes the query and grace periods update.

NOTE: The phrase < ... > indicates that the content has been shortened for readability purpose.

```
OCSP Response Data:
    OCSP Response Status: successful (0x0)
    Response Type: Basic OCSP Response
    Version: 1 (0x0)
    Responder Id: BF37183EB53B43AD1D7237E59CE2FC4DF96C7BFC
    Produced At: May 16 13:28:35 2013 GMT
    Responses:
    Certificate ID:
      Hash Algorithm: sha1
      Issuer Name Hash: 8809099A55F562E1EA13273360FFB7F50B76F508
      Issuer Key Hash: BF37183EB53B43AD1D7237E59CE2FC4DF96C7BFC
      Serial Number: 69
    Cert Status: good
    This Update: May 16 13:28:35 2013 GMT
    Response Extensions:
        1.3.6.1.4.1.41577.1.1:
            24
        1.3.6.1.4.1.41577.1.3:
            22
        OCSP Nonce:
            04101F0696B93BB03B5E84955AA32E16535F
        1.3.6.1.4.1.41577.1.2:
            12
    Signature Algorithm: sha512WithRSAEncryption
         5b:75:04:e2:40:12:fa:ea:85:67:c0:75:29:2b:b0:04:9a:8a:
         < ... >
         aa:de:96:58:4a:14:e3:6e:cc:28:92:f3:a9:cc:13:8e:f5:a7:
         62:00:51:b5:8d:53:1f:1e
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 12034049345340335056 (0xa701881ed68863d0)
    Signature Algorithm: sha512WithRSAEncryption
        Issuer: CN=CCC Root CA, O=Car Connectivity Consortium
        Validity
            Not Before: Apr 25 09:34:45 2013 GMT
            Not After : Oct 17 09:34:45 2032 GMT
        Subject: O=Car Connectivity Consortium, CN=ACMS CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (4096 bit)
                Modulus:
                    00:a3:8e:31:a8:dc:43:51:78:f8:c6:c8:a9:12:22:
                    < ... >
                    7e:e4:36:a8:01:51:ed:c7:4d:a3:9d:e8:62:9f:36:
                    03:10:25
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                BF:37:18:3E:B5:3B:43:AD:1D:72:37:E5:9C:E2:FC:4D:
                F9:6C:7B:FC
            X509v3 Authority Key Identifier:
                keyid:52:7C:16:40:94:8A:E4:D7:BA:01:24:72:AB:1E:95:E3:
```

```
                1A:12:0C:C3
                DirName:/CN=CCC Root CA/O=Car Connectivity Consortium
                serial:E3:EE:B1:5C:85:7B:63:B6
            X509v3 Basic Constraints:
                CA:TRUE
            X509v3 Key Usage:
                Certificate Sign, CRL Sign
    Signature Algorithm: sha512WithRSAEncryption
         1c:a1:c6:a2:ed:89:5d:19:ee:f1:07:1c:eb:c0:92:7e:d1:25:
         < ... >
         86:5b:a3:cc:45:1d:0a:4e:6f:ae:50:9e:80:a2:32:8f:7c:8d:
         cc:ed:75:81:63:be:83:31
-----BEGIN CERTIFICATE-----
MIIFozCCA4ugAwIBAgIJAKcBiB7WiGPQMA0GCSqGSIb3DQEBDQUAMDwxFDASBgNV
< ... >
EJaXdG/6JqHvY0sYyorzqjiPk/ww7sL+f0Nowu6GW6PMRR0KTm+uUJ6AojKPfI3M
7XWBY76DMQ==
-----END CERTIFICATE-----
Response verify OK
devCert.crt: good
    This Update: May 16 13:28:35 2013 GMT
```

Annex B (informative): Application Developer Certificate Example

An example Application Certificate is given below.

NOTE: The phrase `< ... >` indicates that the content has been shortened for readability purpose.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 105 (0x69)
    Signature Algorithm: sha512WithRSAEncryption
        Issuer: O=Car Connectivity Consortium, CN=ACMS CA
        Validity
            Not Before: May 16 00:29:28 2013 GMT
            Not After : Jul 23 00:29:28 2023 GMT
        Subject: CN=16542e60939048fba856ccd034b07f8b0506e249
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d3:72:b9:cf:61:78:91:a5:b2:69:84:f0:77:34:
                    < ... >
                    4d:2e:49:ab:4b:50:c2:83:06:41:4f:6c:72:24:87:
                    97:b5
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
            1.3.6.1.4.1.41577.3.1:
                16542e60939048fba856ccd034b07f8b0506e249
            1.3.6.1.4.1.41577.3.2:
                1234,12345,123456,1,2,3,4,5,6,7,8,9,10,1234,
                < ... >
                34,12345,123456,1,2,3,4,5,6,7,8,9,10
            1.3.6.1.4.1.41577.3.3:
                EMPTY
    Signature Algorithm: sha512WithRSAEncryption
         01:da:0b:01:b9:1d:79:60:17:c1:e5:9e:97:00:29:d8:09:c4:
         < ... >
         ce:a7:b6:02:c4:c2:11:8c:16:3a:b5:ed:33:13:0f:bd:c4:bb:
         79:c4:b2:90:f0:e9:88:db
```

Annex C (informative): Authors and Contributors

The following people have contributed to the present document:

Rapporteur:	Dr. Jörg Brakensiek, E-Qualus (for Car Connectivity Consortium LLC)
Other contributors:	Ed Pichon, E-Qualus (for Car Connectivity Consortium LLC)

History

Document history		
V1.3.0	October 2017	Publication