SMPTE STANDARD

D-Cinema Operations – Facility List Message Exchange Protocol

Table of Contents

3 Barker Avenue., White Plains, NY 10601
(914) 761-1100

Approved July 14, 2017

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices, and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual. This SMPTE Engineering Document was prepared by Technology Committee 21DC.

Intellectual Property

At the time of publication no notice had been received by SMPTE claiming patent rights essential to the implementation of this Engineering Document. However, attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

Introduction

This section is entirely informative and does not form an integral part of this Engineering Document.

The Facility List Message eXchange protocol (FLM-x) is a client-server protocol for publishing FLM documents (see SMPTE ST 430-16) on a network. FLM-x is based on HTTP (see IETF RFC 7230) and uses REST principles (see Fielding, 2000).

FLM-x is intended to be used by any entity that wishes to host FLM documents, including D-Cinema integrators, exhibitors, service providers, etc. FLM data is constantly changing (as devices are swapped out and new screens become digital), and FLM-x lets clients stay up-to-date with this ever-changing data with minimal network and load requirements.

An FLM-x server makes available at a given URI (the SiteList URI) a SiteList document that lists all available FLM documents offered by the FLM-x server. For instance, assuming an FLM-x server makes a SiteList available at the URI "`https://example.org/FLM/`", an FLM-x client would retrieve the SiteList document using the HTTP GET method (see Section 6.1), which would return, for example:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="/static/sitelist-to-xhtml.xsl" type="text/xsl"?>
<SiteList xmlns="http://www.smpte-ra.org/ns/430-15/2017/SiteList"
       xmlns:xlink="http://www.w3.org/1999/xlink">
  <Originator>https://example.org/FLM/</Originator>
  <SystemName>malcoy</SystemName>
  <DateTimeCreated>2010-04-16T11:26:05-07:00</DateTimeCreated>
  <FacilityList>
  <Facility id="tag:example.org,2015:facilities/882345" modified="2010-04-16T10:55:17-07:00"
                     xlink:href="882345" xlink:type="simple"/>
  <Facility id="tag:example.org,2015:facilities/100883" modified="2010-04-12T16:12:32-07:00"
                     xlink:href="100883" xlink:type="simple"/>
```

```
  <Facility id="tag:example.org,2015:facilities/562999" modified="2010-04-10T04:32:01-07:00"
                      xlink:href="562999" xlink:type="simple"/>
  </FacilityList>
</SiteList>
```

Each of the Facility elements within a SiteList document corresponds to a single D-Cinema facility, and thus a single FLM document. The FLM documents are not contained within the SiteList document itself, and but instead accessed through an FLM URI, which is generated using an Xlink `href` attribute included with each Facility. This allows the SiteList document to remain small even if it contains large number of facilities.

For example, a client interested in retrieving the FLM for the facility with identifier `"tag:example.org,2015:facilities/562999"` would use the HTTP GET method on the following FLM URI (see Section 7.1):

```
https://example.org/FLM/562999
```

The response from such a request would be the FLM document itself, assuming it exists.

By design, FLM-x clients do not retrieve FLM documents based on modification or creation date, but instead retrieve the entire SiteList document, which contains the last modification date for individual FLM documents. Note that FLM-x clients can avoid repeatedly retrieving the entire SiteList document by caching the `Last-Modified` HTTP response header and using that date in an `If-Modified-Since` HTTP request header.

A benefit of basing the FLM-x protocol on REST principles and using XML documents is that it functions naturally with web browsers. For example, if a FLM URI is typed into a browser, that FLM will be displayed. Since the XML document is served with a `"Content-Type: application/xml"` HTTP header, most modern browsers will allow the user to explore the XML tree interactively.

An FLM-x server can make the data even more human-friendly by supplying XSL style sheets, e.g. for SiteList, FLM, or Error documents, to transform the XML document into human-friendly XHTML.

1 Scope

This document specifies a protocol to efficiently publish, retrieve, synchronize and submit aggregate Extended Facility List Message (FLM) instances over HTTP.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; then formal languages; then figures; and then any other language forms.

3 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this engineering document. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this engineering document are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax

IETF RFC 2617, HTTP Authentication: Basic and Digest Access Authentication

IETF RFC 5246, The Transport Layer Security (TLS) Protocol, Version 1.2

IETF RFC 7230, Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing

IETF RFC 7231, Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content

IETF RFC 7232, Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests

IETF RFC 7234, Hypertext Transfer Protocol (HTTP/1.1): Caching

SMPTE ST 430-16:2017, D-Cinema Operations – Extended Facility List Message

World Wide Web Consortium (W3C) (2010, May 6). XML Linking Language (XLink) Version 1.1

4 Terms and Definitions

For the purposes of this document, the following terms and definitions apply.

4.1 FLM

Extended Facility List Message, as defined in SMPTE ST 430-16.

4.2 FLM-x protocol

The Facility List Message eXchange protocol (FLM-x) specified in this document.

4.3 FLM-x server

A FLM-x server is an entity that implements at least one of the requests specified in this document.

4.4 FLM-x client

A FLM-x client is an entity that performs at least one of the requests specified in this document.

5 General

5.1 HTTP

An FLM-x server shall support the HTTP 1.1 protocol as specified in IETF RFC 7230, IETF RFC 7231, IETF RFC 7232 and IETF RFC 7234

A persistent connection, as specified in Section 6.3 of IETF RFC 7230, should be maintained.

NOTE: Section 3.3.2 of RFC 7230 specifies the various means by which the length of a message can be transmitted.

5.2 TLS

An FLM-x Server may use TLS 1.2 as specified in IETF RFC 5246 and updated by IETF RFC 5746, IETF RFC 6176, IETF RFC 7465, IETF RFC 7507 and IETF RFC 7568.

Other versions of TLS, e.g. TLSv1.1, TLSv1.0, etc..., shall not be used.

5.3 XML Schema Definitions

Table 1. XML Schema root element definition.

```
<xs:schema
  elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.smpte-ra.org/ns/430-15/2017/SiteList"
  xmlns:tns="http://www.smpte-ra.org/ns/430-15/2017/SiteList"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  attributeFormDefault="unqualified">

  <xs:import namespace="http://www.w3.org/1999/xlink"/>
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
<!-- schema definitions found in this document -->
</xs:schema>
```

This specification uses XML schema definitions as specified in W3C XML Schema Part 1: Structures to define XML structures. These schema definitions shall belong to the XML Schema document whose root element is listed in Table 1.

In the event of a conflict between XML schema definitions and the prose, the prose shall take precedence.

5.4 XML Document Encoding

XML documents included in HTTP requests and responses specified in this document shall use UTF-8 character encoding as specified in [XML].

As a result, the `Content-Type` header for requests and responses containing a single XML document shall be `"application/xml; charset=UTF-8"`.

5.5 Site List URI

A Site List URI is an absolute URI against which one of the requests specified in Section 6 may be performed.

NOTE: The means by which FLM-x clients are informed of the existence of a SiteList URI is not specified in this document.

5.6 FLM URI

A FLM URI is an absolute URI against which one of the requests specified in Section 7 may be performed.

5.7 Authentication

An FLM-x server may grant different access rights to different users. For example, one user could be allowed read/write access, while another could only make read requests.

If different access rights are granted to different users, an FLM-x server shall use one of the following methods to authenticate a user:

- **Basic Authentication Scheme as specified in RFC 2617**. If this method is used, then TLS 1.2, as specified in Section 5.2 of this document, shall be used. The user is uniquely identified by the `userid` parameter of the Basic Authentication credentials.
- **Client Certificate as specified in RFC 5246**. The user is uniquely identified by the client certificate provided by the FLM-x client when establishing the TLS session.

5.8 HTTP Client Errors

The body of responses with a status code in the `4xx` class shall consist of a single XML document whose root element is the `Error` element specified in Table 2.

Table 2. Error element and ErrorType type definitions.

```
<xs:element name="Error" type="tns:ErrorType"/>
<xs:simpleType name="Token">
<xs:restriction base="xs:string">
  <xs:pattern value="[_0-9a-zA-Z]+"/>
</xs:restriction>
</xs:simpleType>

<xs:complexType name="ErrorType">
<xs:sequence>
  <xs:element name="Token" type="tns:Token"/>
  <xs:element name="Message" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

NOTE: Responses that use the `Error` element specify the value of the `Token` element but leave the value of the `Message` element to the discretion of the implementation, which, for instance, can tailor it to specific locales.

5.9 Common HTTP Errors

5.9.1 406 Not Acceptable

This response shall be returned if the `Accept` header of a request does not include the content type of the corresponding FLM-X server response.

The `Token` element value shall be `"ContentTypeNotSupported"`.

EXAMPLE: This section applies if the header `"Accept: application/json"` is received when making a Get FLM request (Section 7.1), which returns content with content-type equal to `"application/xml; charset=UTF-8"`.

5.9.2 415 Unsupported Media Type

This response shall be returned if the `Content-Type` header of a request does not match the content type of the body returned by the FLM-x server.

The `Token` element value shall be `"ContentTypeNotSupported"`.

EXAMPLE: This section applies if the header `"Content-Type: application/json"` is received when making a Add or Update FLM request (Section 7.2), which expects content with content-type equal to `"application/xml; charset=UTF-8"`.

5.9.3 405 Method Not Allowed

This response shall be returned if a request URI is not recognized.

The `Token` element shall be `"MethodNotAllowed"`.

5.9.4 403 Forbidden

This response shall be returned if a user is not authorized to make a request as defined in Section 5.9.4.

The `Token` shall be "`UserNotAuthorized`".

6 SiteList Requests

6.1 Get SiteList

6.1.1 Description

This request retrieves all facilities associated with a SiteList URI.

EXAMPLE: The following HTTP request to authority `"example.org"` retrieves all facilities associated with the SiteList URI `"https://example.org/blah/foo"`:

```
GET /blah/foo HTTP/1.1
```

6.1.2 ETag

To minimize the impact of repeated polling by FLM-x clients, an FLM-x Server should implement the entity-tag feature specified in RFC 7232.

6.1.3 Request

6.1.3.1 Method and Target

`GET {targetSiteList}`

6.1.3.2 URI Parameters

`{targetSiteList}` `origin-form` extracted from the SiteList URI, as specified in IETF RFC 7230.

6.1.3.3 Body

No body shall be present

6.1.4 Reponses

6.1.4.1 200 OK

The body shall be an XML document whose root is a `SiteList` element as specified in Table 3.

Table 3. SiteList element and SiteListType type definitions.

```
<xs:element name="SiteList" type="tns:SiteListType"/>
<xs:complexType name="SiteListType">
  <xs:sequence>
    <xs:element name="Originator" type="xs:anyURI" />
    <xs:element name="SystemName" type="xs:string" />
    <xs:element name="DateTimeCreated" type="xs:dateTime" />
    <xs:element name="FacilityList" type="tns:FacilityListType">
    <xs:unique name="facility-id">
      <xs:selector xpath="tns:Facility" />
      <xs:field xpath="@id" />
    </xs:unique>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

The `Originator` element shall be the absolute URI from which the `SiteList` element was retrieved.

EXAMPLE: `<Originator>https://noname/flm</Originator>`

The `SystemName` element shall be a human-readable description of the server from which the `SiteList` element was retrieved.

EXAMPLE: `<SystemName>Issuer Foo</SystemName>`

The `DateTimeCreated` element shall be the date and time of creation of the `SiteList` element. It is intended strictly for display to a user.

The `FacilityListType` shall be as specified in Table 4.

Table 4. FacilityListType type definition.

```
<xs:complexType name="FacilityListType">
  <xs:sequence>
    <xs:element name="Facility" type="tns:FacilityType" maxOccurs="unbounded"
                minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

Each `Facility` element shall correspond to a unique site.

The `FacilityType` shall be specified as specified in Table 5.

Table 5. FacilityType type definition.

```
<xs:complexType name="FacilityType">
  <xs:complexContent>
```

```
    <xs:restriction base="xs:anyType">
    <xs:attribute name="id" type="xs:anyURI" use="required" />
    <xs:attribute name="modified" type="xs:dateTime" use="required" />
    <xs:attribute ref="xlink:href" use="required" />
    <xs:attribute ref="xlink:type" use="required" />
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

The `id` attribute shall be the equal to the `FacilityID`, as defined in ST 430-16, of the FLM.

The `xlink:href` attribute shall be a URI, per the W3C XLink 1.1 specification.

The `xlink:type` attribute shall be equal to `"simple"` per the W3C Xlink 1.1 specification.

The `modified` attribute shall indicate when the FLM was last modified.

The absolute URI resulting from resolving (as specified in Section 5 of IETF RFC 3986) the `xlink:href` attribute against the absolute URI from which the `SiteList` element was retrieved shall be a FLM URI, as defined in Section 5.6.

Clients shall never attempt to construct FLM URIs other than as described above.

EXAMPLE: For example, to retrieve the FLM corresponding to the following `Facility` element:

```
<Facility id="xyz:882345" modified="2010-04-16T10:55:17-07:00"
                     xlink:href="https://example.org/FLM/xyz%3A882345" xlink:type="simple"/>
```

the following HTTP request would performed:

```
GET /FLM/xyz%3A882345 HTTP/1.1
```

6.1.4.2 301 Moved Permanently

This response shall be returned if the SiteList URI is no longer valid and has been replaced by another Site List URI.

The response shall contain a `Location` header containing the replacement SiteList URI.

7 FLM Requests

7.1 Get FLM

7.1.1 Description

This request retrieves the FLM for a single facility.

7.1.2 Request

7.1.2.1 Method and Target

```
GET {targetFLM}
```

7.1.2.2 URI Parameters

`{targetFLM}` `origin-form` extracted from the FLM URI, as specified in IETF RFC 7230.

7.1.2.3 Body

No body shall be present.

7.1.3 Reponses

7.1.3.1 200 OK

If the an FLM document is associated with `{targetFLM}`, the body shall consist of the FLM document.

The HTTP `Last-Modified` header shall correspond to the `modified` attribute of the matching `Facility` element in the `SiteList`.

EXAMPLE: The `Last-Modified` header associated with

```
<Facility id="432949" modified="2014-06-30T18:21:30-07:00" xlink:href="xyz%3A432949"
                      xlink:type="simple"/>
```

is

```
Last-Modified: Fri, 23 Apr 2010 01:00:30 GMT
```

NOTE: The `datetime` format of the `Last-Modified` HTTP header differs from the `datetime` format found in the `SiteList` and other D-Cinema XML documents. The former is always expressed in UTC, whereas the latter can include a time zone.

The FLM document returned by the FLM-x server may include XML processing instructions.

EXAMPLE: An XSLT stylesheet can be included in an FLM to facilitate viewing, e.g. `<?xml-stylesheet href="/static/flm-to-xhtml.xsl" type="text/xsl"?>`.

7.1.3.2 301 Moved Permanently

This response shall be returned if the FLM URI is no longer valid and has been replaced by another FLM URI.

The response shall contain a `Location` header containing the replacement FLM URI.

7.1.3.3 410 Gone

This response shall be returned if the FLM is not available and its whereabouts are unknown.

The Token element value shall be `"NoSuchFLM"`.

7.2 Add or Update Facility

7.2.1 Description

This request allows the FLM associated with a facility to be created or modified.

7.2.2 Request

7.2.2.1 Method and Target

`POST {targetFLM}`

7.2.2.2 URI Parameters

`{targetFLM}` `request-target` in `origin-form` as specified in IETF RFC 7230.

7.2.2.3 Body

The body shall consist of a single FLM document as specified in SMPTE ST 430-16.

The FLM-x server should strip any XML Processing Instructions included in the body of the request, since these PI can contain ephemeral information and/or security implications.

7.2.3 Reponses

7.2.3.1 201 Created

If there is no existing FLM document associated with `{targetFLM}`, the FLM-x server shall associate `{targetFLM}` with the FLM provided in the request body and an HTTP status of `201 Created` shall be returned.

7.2.3.2 204 No Content

If there is an existing FLM document associated with `{targetFLM}`, that existing FLM document shall be replaced with the FLM document provided in the request body and an HTTP status of `204 No Content` shall be returned.

7.2.3.3 411 Length Required

This response shall be returned if the client omitted the `Content-Length` header on the request.

The `Token` shall be `"MissingContentLength"`.

7.2.3.4 400 Bad Request

This response shall be returned:

- if the FLM in the request body contains an error, in which case the `Token` shall be `"MalformedXML"`; or
- if the FLM-x server believes that the FLM in the request body duplicates an existing FLM , in which case the `Token` shall be `"DuplicateViolation"`.

NOTE: The criteria by which an FLM-x server determines whether an FLM duplicates an existing one is not specified herein, and left to each implementation.

7.3 Delete Facility

7.3.1 Description

This request deletes an FLM.

EXAMPLE: To delete the FLM associated with FLM URI "`https://example.org/FLM/xyz%3A882345`", the corresponding request is made to the "`example.org`" authority:

```
DELETE /FLM/xyz%3A882345 HTTP/1.1
```

7.3.2 Request

7.3.2.1 Method and Target

`DELETE {targetFLM}`

7.3.2.2 URI Parameters

`{targetFLM}` `request-target` in `origin-form` as specified in IETF RFC 7230.

7.3.2.3 Body

No body shall be present.

7.3.3 Reponses

7.3.3.1 204 No Content

This response shall be returned if the FLM is successfully deleted.

7.3.3.2 410 Gone

This response shall be returned if the FLM is not available and its whereabouts are unknown.

The `Token` element value shall be "`NoSuchFLM`".

8 FLM-x Client Behavior

8.1 Get SiteList

When making Get SiteList (Section 6.1) requests, an FLM-x client should:

- make successive requests no less than 120 seconds apart; and
- support the entity-tag feature field as specified in RFC 7232.

8.2 Get FLM

An FLM-x client should make a Get Facility request for a Facility listed in a SiteList only if one the following conditions is met:

- the FLM-x client has never performed a Get Facility request on the Facility, as identified by the `id` attribute of the `Facility` element; or
- the Facility has been modified since the last Get Facility request on the Facility performed by the FLM-x client, based on the `modified` element of the `Facility` element.

8.3 Deleted FLM

An FLM-x client should detect deleted Facilities by identifying Facility elements, as identified by their `id` attributes, missing from a previously retrieved SiteList.

Annex A Security Considerations (informative)

A.1 Authorization

The FLM-x server can implement an access control system to enforce which operations a user is permitted to perform:

- Role-based, where the role of a user making requests to an FLM-x server is established by the authentication described in Section 5.7.
- Maintain separate permissions on each existing FLM-x record identifying which users (or role held by a user) may perform the following operations: read, update and delete.
- Maintain a separate permission on the site identifying which users (or role held by a user) can retrieve a Facility List.
- Maintain separate permissions for each user (or role held by a user) granting the ability to:
    - Create a new record owned by the user (or role)
    - Create a new record owned by another user (or role)
    - Maintain an "Administrator" role allowed to perform all operations
    - Grant each user the minimum privileges needed to perform their tasks

The mechanism used to support this model is left to the implementer. Possibilities include tagging each facility record with an owner, or having separate URIs to access facilities based on their owners.

It is recommended that users of FLM-x server take the necessary steps to formalize such rules to ensure that data integrity is maintained.

A.2 TLS Certificates

TLS certificates need to be carefully managed to ensure the integrity of the system. It is recommended that:

- certificate be issued & signed by a trusted Certificate Authority; and
- certificate are confirmed to be valid, e.g. not be expired and not revoked.

Transport Layer Protection Cheat Sheet (see bibliography) provides guidelines on the use of TLS certificates.

A.3 Logging

It is recommended that FLM-x server maintain a log all significant client-server transactions, including:

- Authentication (successful and failed)
- Retrieval of FLM records
- Creation of FLM records
- Modification of FLM records

It is further recommend that the log be stored in a secure location to prevent modification or removal of log records.

Annex B Operational Considerations (informative)

B.1 FLM-x Server Deployment

This specification places no constraints on the number of servers deployed, the parties deploying them (e.g. Exhibitors, Distributors, Studios), or how tightly they are coupled. A highly idealized representation of some of the relationships between the various roles is shown in Figure 1.

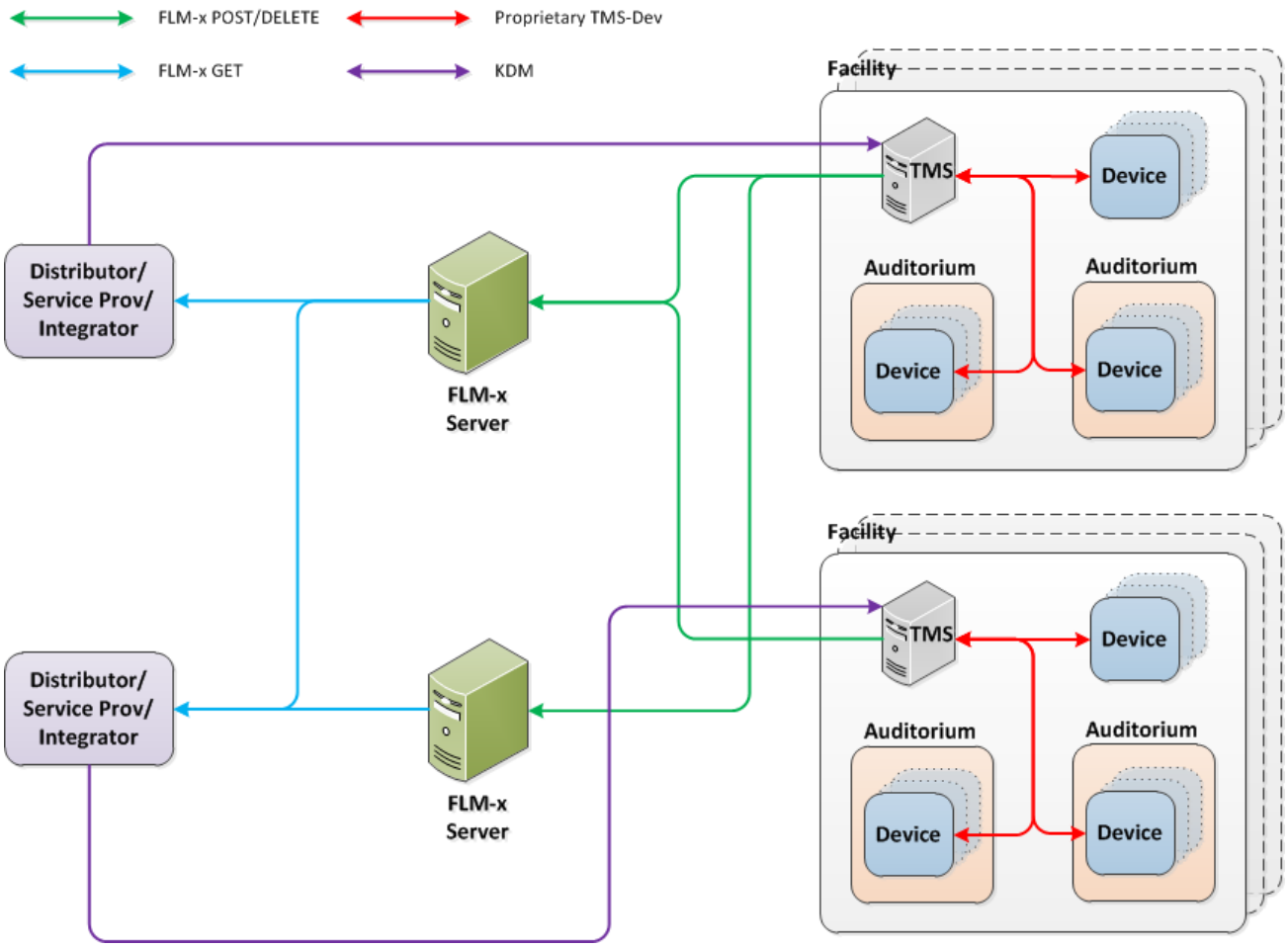

Figure 1: Representative Roles

It is recommended that:

- exhibitors (or agents acting on their behalf) are only able to modify facilities under their control;
- in general, users are only able to obtain information on facilities they have a legitimate need to access, e.g. a distributor creating a DCP (as specified in ST 429-2) or a KDM (as specified in ST 430-1).

B.2 Discovery

This specification does not include a means of determining, a priori, the one or more SiteList URIs providing information about a facility. SiteList URIs can be provided, for instance, when obtaining user authentication credentials for an FLM-x server.

B.3 FLM Creation

Facility records contain many fields (particularly those that are device-related). It is recommended that marshaling these field be automated to the greatest extent possible to prevent transcription errors. This functionality can be logically implemented in theatre management system (TMS), as it would typically be able to access Exhibitors' intranets (to gain access to device information, as well as the Internet (to talk to FLM-x server(s)). At a minimum, it is recommended that the TMS be able to do the following:

- Find and identify devices regardless of manufacturer
- Determine which auditorium a device is in (or if the device is global to the facility)
- Ensure devices are physically collocated
- Collect data needed for Facility records from devices

B.4 FLM Validation

It is recommended that processes exist to continually monitor Facility records to make sure they are valid. This extends beyond syntactic correctness to include checks for common errors such as non-unique Facility ID, expired certificates, etc.

Annex C Method Summary (informative)

Table 6 summarizes the complete set of operations that may be performed within the FLM-x architecture.

Table 6: Method Summary

Operation	HTTP Method	Example URI	Reference
Get a SiteList	GET	https://foo.org/FLM/	Section 6.1
Get an FLM	GET	https://foo.org/FLM/xyz%3A123	Section 7.1
Add or Update an FLM	POST	https://foo.org/FLM/xyz%3A123	Section 7.2
Delete an FLM	DELETE	https://foo.org/FLM/xyz%3A123	Section 7.3
Any other combination			Section 5.9.3

Annex D XML Schema (informative)

This specification is accompanied by the following element, which is an XML schema document as specified in the XML Schema Part 1: Structures.

st430-15a-2017.xsd

This element collects the XML schema definitions defined in this specification. It is informative and, in case of conflict, this specification takes precedence.

Bibliography (informative)

Roy T. Fielding. Architectural styles and the design of network-based software architectures. PhD Thesis, University of California, Irvine, 2000.

SMPTE ST 429-2, D-Cinema Packaging — DCP Operational Constraints

SMPTE ST 430-1, D-Cinema Operations — Key Delivery Message

Transport Layer Protection Cheat Sheet
(https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet)