System Specification

Version 2.0.1r1 28 January 2015

System Specification Version 2.0.1r1

Notice:

As of the date of publication, this document is a release candidate specification subject to DECE Member review and final adoption by vote of the Management Committee of DECE in accordance with the DECE LLC Operating Agreement. Unless there is notice to the contrary, this specification will become an adopted “Ecosystem Specification” on 5 February 2015.

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Digital Entertainment Content Ecosystem (DECE) LLC (“DECE”) and its members disclaim all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein. This document is subject to change under applicable license provisions, if any.

Implementation of this specification requires a license from DECE.

Contact Information:

Licensing and contract inquiries and requests should be addressed to us at:
http://www.uvvu.com/uv-for-business

Contents

Figures and Tables

1 Introduction

1.1 Scope

1.2 Document Organization

This document describes a new digital content ecosystem designed to allow users to purchase digital media from multiple retailers, sharing their purchases with all members of their household, and enabling seamless playing of the media on all devices in their household.

Section 1 Introduces the organization of this document, and describes its notations and conventions. It includes a glossary of terms, and lists references uses throughout the document.

Section 2 Provides an overview of the Ecosystem.

Section 3 Provides an informational overview of the DECE Architecture and its Roles.

Section 4 Describes the key Ecosystem entities, known as Roles, defining the Coordinator, Retailer, Download Service Provider, Locker Access Service Provider, DECE Device, and Access Portal Roles.

Section 5 Defines the structure of the identifiers used throughout the Ecosystem, their syntax, and which entity serves as their naming authority.

Section 6 Introduces a Node, which is an instance of a Role, and serves as a trust boundary with a unique, certified identity for mutually authenticating and securely communicating with other nodes in the Ecosystem. It also introduces a Security Token which is used for secure delegation of User authorization, and describes the end to end message security.

Section 7 Describes DECE Accounts, Users, Domains, and Rights Locker operations including creation, deletion, and joining Devices to Domains.

Section 8 Introduces the Common File Format used to contain instances of Content.

Section 9 Describes how a Content Provider creates a Container and publishes it to the Ecosystem.

Section 10 Outlines how a Retailer sells Rights to Content and updates the Rights Locker.

Section 11	Shows how Containers are downloaded to Devices.
Section 12	Describes how Content is then Licensed for playback and how the Rights Locker interacts with native DRM systems.
Section 13	Discusses how Content is played on a Device, including Streaming Content from a Locker Access Service Provider.
Section 14	Outlines the support for Discrete Media Rights.
Section 15	Contains details on Superdistribution including Container initialization and License Acquisition.
Appendices	Tables with the current DECE Ecosystem parameters and DRM identifiers.

1.3 Document Notation and Conventions

1.3.1 Notations

The following terms are used to specify conformance elements of this specification. These are adopted from the ISO/IEC Directives, Part 2, Annex H [ISO-P2H].

SHALL and SHALL NOT indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

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.

MAY and NEED NOT indicate a course of action permissible within the limits of the document.

Terms defined to have a specific meaning within this specification will be capitalized, e.g. “Track”, and should be interpreted with their general meaning if not capitalized. Normative key words are written in all caps, e.g. “SHALL”.

1.3.2 Sequence Diagram Conventions

Sequence diagrams loosely conform to the OMG UML 2.0 [UML] conventions.

Usage new to UML 2.0:

Use of iteration frames, especially REF to reference repeated sequences packaged into a shared drawing, and LOOP to illustrate simple iterations with guards denoting the iteration range.

Non-conforming Usage:

Use of double headed arrows to denote a sequence of messages and responses grouped together for simplicity.

Messages and responses colored in red denote messages and responses which are out of the scope of the DECE and are included for illustrative purposes.

1.4 Definitions

Access Portal

A Node implemented by a DECE-licensed entity to act as an intermediary communicating with the Coordinator on one side and applications controlled by the entity on the other side.

Account or DECE Account

The collection managed by the Coordinator of all DECE data relevant to a single household (Devices, Domains, Users, Rights Tokens, Rights Locker, etc.).

Adaptive Streaming

Continuous internet download and playback of media Segments while automatically adjusting media bit rate (and possibly encoded picture size) by selecting from alternative Media Segments available from Web servers in order to dynamically adapt bitrate to network throughput without interrupting playback.

Approved Discrete Media Fulfillment Method (ADMFM)

The use of a format and content protection system in a manner approved by DECE for fulfilling a Discrete Media Right.

Approved DRM

A DRM system that has passed the DECE DRM approval process. A list of the Approved DRMs is contained in Appendix B.

Approved Stream Protection Technology
A DRM system or other content protection technology approved by Content Providers for Streaming. A list of the Approved Stream Protection Technologies is contained in Appendix C.

Asset
A component of Content in abstract form (see Logical Asset) or concrete form (see Physical Asset).

Browser
Browser is used in these specifications as shorthand for *web browser*, which is an end-user software application for retrieving, presenting, and traversing information resources on the World Wide Web. A W3C “user agent.”

Compliance Verification
The process for a DECE Role to carry out all defined requirements for the Compliance Verification Process (CVP).

Client Implementer
An organization that implements the Device Role.

Common File Format (CFF)
The standard DECE Content delivery file format, encoded in one of the approved Media Profiles and packaged (encoded and encrypted) as defined by the DECE Common Container & Media Format Specification.

Common File Format Timed Text (CFF-TT)
Common File Format Timed Text, as defined in [DMedia]. CFF-TT tracks are referred to as subtitle tracks.

Common Streaming
A framework for delivering stream segments streams conforming with the Common Streaming Format (CSF) definition from a LASP to Devices where that Device can play those stream segments.

Common Streaming Format (CSF)
Common Streaming Format (CSF) refers to any Common File Format (CFF) specializations or profiles that are specifically designated for streaming. CSF files CFF files..

Connected Device
A DECE Device that communicates directly and autonomously with the Device Portal.

Consent

Permission from a User for a policy or policies to be applied to the User or to the User’s Account.

Container

Shorthand for Digital CFF Container (DCC).

Content

A movie, television show, music video, or other media work made available in the Ecosystem. The term Content used informally may include Assets. (In [FRBR], a “work.”)

Content Key

A cryptographic key used to decrypt portions of DCC. See Keyset.

Content Provider

A DECE-licensed entity that publishes Content to the Ecosystem.

Coordinator

The central entity controlled by the DECE LLC that facilitates interoperability across Ecosystem services, and stores and manages Accounts.

DASH

Dynamic Adaptive Streaming over HTTP, as specified by [DASH].

DECE

Digital Entertainment Content Ecosystem.

DECE CFF Container (DCC)

Synonym for Digital CFF Container.

DECE Device or Device

A DECE-licensed hardware or software implementation of the Device Specification incorporating one or more Licensed Applications and one or more DRM Clients on a single physical device. The term DECE Device and Device are used synonymously. This is distinguished from a “physical device” used to refer to hardware regardless of whether or not it implements a DECE Device.

Device Join

Adding or “registering” a DECE Device to DECE Domain associated with a single DECE Account, using DRM-specific domain join functions provided at the Coordinator and initiated from the Device.

Device Leave
Removing or “unregistering” a DECE Device from a DECE Domain associated with a single DECE Account, using a DRM-specific domain leave function initiated at the Device.

Device Portal
A programmatic Web services interface made available by the Coordinator Role that exposes a subset of the Coordinator API to DECE Devices.

Device Proxy
A service operated by a Client Implementer to implement some functions of the Device Role. The interface between the Device and the Device Proxy is not mandated by DECE.

Digital CFF Container (DCC)
An instance of Content published in the Common File Format.

Discrete Media
Standalone physical media (e.g., an optical disc or memory device) containing Content bound to the media using an Approved Discrete Media Fulfillment Method and playable on non-DECE devices.

Discrete Media Client
An application that fulfills Discrete Media Rights by recording Content to Discrete Media using an Approved Discrete Media Fulfillment Method.

Discrete Media Content
An instance of a Physical Asset bound to standalone media (such as an optical disc or memory device) in an approved format playable on non-DECE devices.

Discrete Media Right
A Right specific to Discrete Media. That is, permission for a User to obtain Content as Discrete Media.

Domain or DECE Domain
A defined and identifiable group of DECE Devices associated with a single Account across which that Account’s Content can be played. A DECE Domain may span one or more DRM Domains.

Download Manager
Software that downloads DCCs from DSPs using DECE-defined protocols.

Download Manifest
A data structure providing information a Download Manager needs to obtain DCCs associated with a Right. That is, a list of files, download locations, and related information provided by a DSP or Retailer.

Download Service Provider (DSP)
A DECE-licensed service responsible for fulfilling Rights on behalf of a Retailer by delivering DCCs and DRM licenses.

DRM
Digital Rights Management.

DRM Client
An implementation of a DECE-approved DRM that can decrypt DCCs using the Keyset carried in the DRM license and enforce usage rules according to a DRM license and/or policy.

DRM Domain
The set of Devices in a DECE Domain that share the same DRM.

DRM Domain Credential
The object used by a DRM to bind Devices and DRM Licenses to a DRM Domain. Details of the identity and cryptographic methods used are specific to each DRM.

DRM License
An object or policy issued by a DRM License Manager allowing a DRM Client to decrypt a Container.

Dynamic LASP (DLASP)
LASP service that authenticates a User on a session-by-session basis.

Ecosystem
The manifestation of the DECE architecture, as defined by the DECE specifications and implemented by DECE participants.

Experience
Defines the relationships between Presentations, Applications and other data associated with Content.

Experience Media Application
A Media Application that defines an Experience. An Experience Media Application provides enough information to create a user interface that allows a User to Navigate the Content associated with that Experience.

Fulfill To deliver Physical Assets associated with an Account’s Right at the behest of a User in that Account.

ISO 1) The ISO Base Media File format (“ISO container” or “ISO media file”) as used in the DECE Common Container & Media Format Specification. 2) The ISO 9660 file format for storing the contents of an optical disc (“DVD ISO image” or “DVD ISO”). 3) The International Organization for Standardization, which defined both file formats above.

Keyset The set of all Content Keys needed to decrypt playable elements of a DCC.

Late Binding The combination of separately delivered content (e.g. audio, video, or subtitles) into a single, synchronous presentation.

Late Bound Common Streaming A mix of Late Binding and Common Streaming, a Device plays a combination of tracks played from a DCC and tracks streamed using Common Streaming.

LASP (Locker Access Streaming Provider) A DECE-licensed service provider that Streams Physical Assets associated with an Account’s Right to a LASP Client.

LASP Client Hardware and/or software that renders a Stream under control of a LASP and conforms to the DECE output control policies in the LASP Compliance Rules.

LASP Session A period of time during which an authenticated User or Account may receive a stream from a LASP.

License Manager A DRM service operated by a DSP that issues and manages DRM Licenses.

Licensed Application The software in a DECE Device, other than the DRM Client, that performs DECE functions.

Linked LASP (LLASP) LASP service to a LASP Client that is persistently bound by the LASP to an Account.

Logical Asset

An abstract instance of Content, independent of the manifestation such as encoding or packaging. (In [FRBR], an “expression.”)

Media Application

An application (presentation control program) associated with Content. A Media Application may take forms ranging from simple play lists, declarative data, and markup languages; to procedural language programs that are interpreted by players or virtual machines, or compiled to binary to run on specific processors.

Media Player or DECE Media Player

A device or software application that decodes and presents Content from a DCC. A Media Player is a class of a Licensed Application.

Media Presentation

As defined in [DASH], “collection of data that establishes a bounded or unbounded presentation of media content.” A Media Presentation closely corresponds with a Common File Format DCC, although a Media Presentation can also be an abstract representation of media components that play together through a DMP or Streaming.

Media Presentation Description (MPD)

As defined in [DASH], “formalized description for a Media Presentation for the purpose of providing a streaming service.” Note: The formalized description is an XML schema and an instance document of that schema is referred to as an MPD.

Media Profile or Profile

Requirements and constraints such as resolution and subtitle format for Content in the Common File Format. Current Media Profiles are SD and HD.

Metadata

Data that describes Content, including Logical Assets and Physical Assets.

Node

An instance of a Role. A Node is assigned a unique certified identity (a certificate) by DECE, creating a trust boundary used to mutually authenticate and secure communication between the Node and the Coordinator.

Outbound File Transfer

Copying or moving a Digital Asset from a Device so that it can potentially be delivered to another DECE Device.

Parental Control

See Ratings Enforcement.

Parental Control Information or Parental Controls

Coordinator-managed settings to restrict a User’s access to Content and visibility of Content. Compare to Ratings Enforcement.

Persistent Account-bound Mode

A mode of LASP operation intended for use with a LASP Client shared with members of an Account. Compare to Single-session User Mode and Persistent User-bound Mode.

Persistent User-bound Mode

A mode of LASP operation intended for use with LASP Clients in personal devices. Compare to Single-session User Mode and Persistent Account-bound Mode.

Physical Asset

A specific manifestation of an Asset for a single Media Profile, such as a DCC. (In [FRBR], a “manifestation.”)

Playback Device

A DECE Device, a LASP Client, or the physical device containing a LASP Client.

Policy

1) Rules for operating in the Ecosystem. 2) A data structure in the Coordinator used to specify an allowable action or configuration.

Profile

See Media Profile.

Ratings

Subjective classifications of suitability of Content for particular audiences. Ratings may include reasons, which are attributes of a given rating, such as adult language or violence.

Ratings Enforcement

Limiting access to Content or Content listings by applying parental control settings to Content Ratings. The Coordinator does Ratings Enforcement by comparing Parental Control Information for a User to Ratings in Content Metadata. Devices and Linked LASPs may do Ratings Enforcement by comparing Device-specific or service-specific settings or Coordinator Parental Control Information to Ratings in DCCs or Coordinator Metadata or other Ratings sources. Compare to Parental Control Information.

Ratings System

A set of Ratings, typically defined by a ratings body.

Retail Account

An account maintained by a Retailer for facilitating purchases. A Retail Account may be bound to a DECE User.

Retailer

A DECE-licensed entity operating a consumer-facing storefront that sells Rights.

Right

A collection of allowed usages of one Profile of a Logical Asset (a particular piece of Content) associated with an Account. Rights may relate to whether the Content can be downloaded, streamed, or otherwise processed.

Rights Locker

Coordinator functionality that manages a collection of Rights Tokens, uniquely associated with an Account.

Rights Token

An object managed by the Coordinator representing a Right.

Role

A DECE entity that implements a specific set of functionality and both exposes and invokes a defined collection of interfaces. Roles are Coordinator (including the Device Portal), Portal (Web Portal), Access Portal, Content Provider, Retailer, DSP, LASP, Device, and Customer Support.

Security Token

An object for exchanging authentication and authorization data between the Coordinator and a Node. Delegation Security Tokens (often simply called Security Tokens) are primarily for User and Account authentication and intrinsically identify which Coordinator services the Node is authorized to use on behalf of the User or Account. Delegation Security Tokens can be constructed for transient authenticated sessions or for persistent delegation when linking a User to a Node. Federation Security Tokens allow for remote authentication (e.g., authenticated links from a Node to the Web Portal) and distributed identity (e.g., providing a User ID to a Node for Content access). Different from User Credential.

Single-session User Mode

A mode of LASP operation where a short-term session is established, requiring User authentication. Compare to Persistent Account-bound Mode and Persistent User-bound Mode.

Stream or Streaming

Transmitted Content, protected by an Approved Stream Protection Technology, that is not persistently stored on the receiving LASP Client except for the purposes of buffering, including for instant start of playback, and to enable trick-play.

Superdistribution

Any means of distributing DCCs in advance of the recipient obtaining a Right to the Content. This includes preloading DCCs on media or DECE Devices, sharing DCCs on download services or peer to peer networks, and copying a DCC from one DECE Device to another DECE Device in a different Account. Before Superdistributed Content can be accessed (decrypted), a User must obtain the associated Right from a Retailer.

Tethered Device

A DECE Device that consists of a component that communicates with the Coordinator and other DECE Roles (typically on a general purpose computer) and a separate part (typically containing the DECE Media Player) that connects with the component.

Trust Authority

A trusted entity, usually the Coordinator, that issues digital certificates for use by Nodes and other entities licensed by DECE.

Unverified Device Leave Removing or “unregistering” a Device from an Account at the Coordinator without performing a DRM-specific leave function from the Device.

User or DECE User A person with a User Credential that is a member of an Account.

User Access Level A set of privileges specifying allowed behaviors of a User.

User Credential A unique assertion of User identity (a username) secured by a password. Different from Security Token.

Web Portal An interactive HTML application made available by DECE, independent of any particular Retailer or LASP, giving Users direct access via a Web Browser to functions such as Account settings, User management, Rights Locker viewing, and Device management.

1.5 References

1.5.1 DECE References

The following versions of documents SHALL comprise the version 2.0.1 DECE “Ecosystem Specifications”:

[DSystem]	System Specification, Version 2.0.1; fmdece-2.0.xsd
[DCoord]	Coordinator API Specification, Version 1.2; coordinator-1.1.1.xsd
[DDiscrete]	Discrete Media Specification, Version 2.0.1
[DPublisher]	Content Publishing Specification, Version 2.0
[DDevice]	Device Specification, Version 2.0
[DMeta]	*For DCoord:* Content Metadata Specification, Version 1.1.1; mddece-1.1.xsd *For all other documents:* Content Metadata Specification, Version 2.0.1; mddece-2.0.1.xsd

[DCMeta]	*For DCoord:* Common Metadata Specification, Version 2.1c; md-v2.1.xsd *For all other documents:* Common Metadata Specification, Version 2.2; md-v2.2.xsd
[DCMetaCR]	Common Metadata Ratings Specification, Version 2.2.2
[DMedia]	Common File Format & Media Formats Specification, Version 2.0.1; cff-tt-1.1.zip
[DSecMech]	Message Security Mechanisms Specification, Version 1.1.1
[DGeo]	Geography Policies Specification, Version 1.2

[DKeyDelivery]	Keyset Delivery Format Specification, Version 2.0.1;
	keydelivery-2.0.1.xsd;
	keydelivery_redefine-2.0.1.xsd

[DDMP]	Media Package Specification, Version 2.0.1
[DStream]	Common Streaming Protocol Specification, Version 2.0r1
[DCManifest]	Common Media Manifest Metadata, Version 1.1a; manifest-v1.2.xsd

Some specifications are more concerned with certain Roles than others, and the following table summarizes which specifications are most applicable to each Role. However, the table below is provided for convenience only; it does not in any way limit Role implementers' obligations to comply with all requirements applicable to them regardless of which specification contains those requirements and whether those specifications are indicated as "most applicable" by the table below. (See Section 4 for details on all the DECE Roles.)

	Content Provider	Retailer	DSP	LASP	Device
DSystem	●	●	●	●	●
DCoord	●	●	●	●	●
DSecMech	●	●	●	●	●
DMeta	●	●	●	●	●
DDiscrete	●	●	●		●
DMedia	●		●	●	●
DDevice				●	●
DPublisher	●		●	●	●
DGeo	●	●		●	
DKeyDelivery	●		●	●	
DDMP	●		●	●	●

DStream	●			●	●

Specification and Roles Table

1.5.2 External References

[DASH] ISO/IEC 23009-1:2011, “Dynamic Adaptive Streaming over HTTP”

[EIDR] EIDR ID Format v1.02 http://eidr.org/documents/EIDR_ID_Format_v1.02_Jan2012.pdf

[EVCert] Guidelines for the Issuance and Management of Extended Validation Certificates http://www.cabforum.org/Guidelines_v1_2.pdf

[FRBR] IFLA Study Group on Functional Requirements for Bibliographic Records http://www.ifla.org/en/publications/functional-requirements-for-bibliographic-records

[HTTP] Hypertext Transfer Protocol – HTTP/1.1 (RFC 2616) http://www.ietf.org/rfc/rfc2616.txt

[HTTP Auth] HTTP Authentication (RFC 2617) http://www.ietf.org/rfc/rfc2617.txt

[ISAN] ISO 15706-2:2007, "Information and documentation -- International Standard Audiovisual Number (ISAN) -- Part 2: Version identifier

[ISO-P2H] ISO/IEC Directives, Part 2, Annex H http://www.iec.ch/tiss/iec/Directives-part2-Ed5.pdf

[RFC3986] “Internationalized Domain Names in Applications (IDNA): Protocol”, J. Klensin, August 2010. http://www.ietf.org/rfc/rfc5891.txt

[RFC5891] “Uniform Resource Identifier (URI): Generic Syntax” T. Berners-Lee, R. Fielding and L. Masinter, January 2005. http://www.ietf.org/rfc/rfc3986.txt

[SAML] Security Assertion Markup Language Version 2.0 http://saml.xml.org/saml-specifications

[SMPTE2053] SMPTE ST 2053:2011, *Media Package for Storage, Distribution and Playback of Multimedia File Sets and Internet Resources*, July 13, 2011.

[TLS] The Transport Layer Security (TLS) Protocol, Version 1.2 (RFC 5246) http://tools.ietf.org/html/rfc5246

[UML] Object Management Group (OMG) Unified Modeling Language (UML) http://www.omg.org/spec/UML/2.0/

[URI] Uniform Resource Identifier (URI): Generic Syntax (RFC 3986). http://tools.ietf.org/html/rfc3986 and Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations (RFC 3305) http://tools.ietf.org/html/rfc3305

[X.509] Internet X.509 Public Key Infrastructure Certificate and CRL Profile (RFC 2459) http://www.ietf.org/rfc/rfc2459.txt

1.6 XML Change Management

XML schemas necessarily change as systems evolve. In particular, DECE Nodes and Devices will encounter situations where XML documents are authored against newer revisions of defining XML schemas. It is necessary that Nodes and Devices handle this situation gracefully by extracting needed information from the document while ignoring extraneous changes to the document.

In DECE specifications, schema updates are designed to support backwards compatibility. For example, element and attributes can be added, but required elements are not removed; or more generally ordinality of elements and attributes can be widened but not narrowed. Values are not changed in either syntax or semantics.

Given these rules for encoding, Nodes and Devices also follow rules to support backwards compatibility.

An XML document is considered compatible if its structure does not preclude the extraction of data from the document. For example, a document with additional elements and attributes do not preclude schema parsing and data extraction.

For all uses of compatible XML documents other than use of the Coordinator API, the following requirements apply:

- Nodes and Devices SHALL NOT reject compatible XML documents, even if they fail schema validation.
- Nodes and Devices SHALL extract data from compatible XML documents.
- Nodes and Devices MAY ignore elements and attributes whose presence is not allowed in the specification and schema versions against which that Node or Device was built. For example, if the original schema allows one instance and three instances are found, the 2nd and 3rd instance may be ignored.

The Coordinator API is exempt as it always uses well defined namespaces that map to version-based URLs corresponding to released API versions.

2 DECE Overview

2.1 Background

Today's consumer of audio and video media has, over many decades, grown used to a simple yet effective method of acquiring content that ultimately results in the purchase of some form of physical media such as CDs, DVDs and now Blu-ray Disks. Consumers have come to expect convenience and flexibility with the CD and DVD purchase and usage experience. In particular, consumers can choose among several retailers and make the decision on where to make their purchase based on price, choice, convenience, affinity, and the like. Competition creates a robust ecosystem that is beneficial to the consumer, retailer, distributor, rights holder, and device manufacturers. Furthermore consumers know that content purchased at any retailer will play on any CD or DVD player. The consumer knows that the content they purchased is theirs and they are free to take it with them and enjoy it wherever they like. This is based on the trust consumers have placed in the DVD and CD brands, the underlying technologies and the industry's success at educating consumers that "it will just work".

With the wide spread availability and penetration of high-speed broadband, and the movement towards devices with direct IP connectivity, that physical media in general, and optical media specifically, may soon be outdated. As we move from a world of DVDs and CDs to a world where content can be purchased and enjoyed directly from the comfort of your living room or personal media player follows that consumers will continue to expect the flexibility and convenience of the DVD experience as described above. They will expect the usage model they have grown accustomed to in the physical world will work for content they will purchase in the digital world.

The reality is that to date this has not been the case. Existing digital content solutions are closed ecosystems, resulting in a market of numerous non-interoperable silos. Each silo has a different set of usage rules enforced by a single Digital Rights Management (DRM) solution and each is linked to a single retail portal selling a limited set of content. Content licensing in these silos is usually bound to a single or very limited set of devices, as defined by the specific usage rules for each silo, limiting how and when consumers can enjoy the content they have purchased. These "silo" ecosystems are neither flexible nor convenient and fall short when it comes to the expectations of consumers. Ultimately, this results in a fragmented market that gives little incentive for consumers to shift to purchasing content online.

In one scenario consumers will simply fail to adopt online content acquisition in sufficient quantity to be fiscally viable, and continue to purchase content on physical media. In the worst case, consumers may use of illegal file sharing networks to gain access to the content they want on any or all devices they own. Apple has achieved a degree of success with its iPod + iTunes, but this has primarily been for music not video. Aside from Apple, the increasing trend is to deliver music DRM-free in MP3 format. For music, the unprotected MP3 format provides the flexibility and convenience associated with traditional CDs. However, the music industry's delay in defining a convenient legal electronic ecosystem has contributed to widespread piracy and financial disaster for the industry. The task at hand is to define and implement a convenient, flexible ecosystem for digital content, particularly high-value studio film content that meets consumer expectations for convenience and choice, and presents a better experience than today's physical delivery systems or piracy.

2.2 New Ecosystem

This new Ecosystem must benefit all participants.

- **The consumer** – The Ecosystem must allow consumers to seamlessly experience any digital content from any retailer across many devices.
- **The retailer** – The Ecosystem must not constrain the ability of retailers to compete in the market place.
- **The device manufacturer** – The device manufacturer must be able to easily implement and innovate on a range of competitive devices that can compete in the marketplace
- **The content owner** – The Ecosystem must ensure the security of the content owner's intellectual property.

It may seem like a daunting set of requirements, however, frameworks and technologies do exist today that can be used to create an ecosystem that can address them. At a minimum, the solution must address several important areas.

- There must exist a single well branded Ecosystem and associated usage model that is shared and enforced across all Ecosystem participants.
- It must leverage a single universal media format, playable on a large class of devices.
- It must allow for the use of multiple Digital Rights Management (DRM) technologies that are able to enforce the usage model. This will ensure that content can be rendered on a wide range of systems and devices.
- Media formats and DRM systems should be generally invisible to the consumer: a consumer should only be concerned with the title and the quality level (profile) of his purchase but should be unaware of the technical details of media formats and protection systems.

- A record of consumer purchases is maintained in the cloud by the Ecosystem, easing consumer management and availability.
- In order to ensure true interoperability, a single architectural framework must exist that will enable consumers to easily purchase and access content they own from a diverse set of content retailers on a wide-ranging set of devices, while still allowing competition and innovation in the marketplace.

3 DECE Architecture (Informative)

The Digital Entertainment Content Ecosystem (DECE or the “Ecosystem”) has been designed to provide the consumer with the best possible digital content experience. In effect the Ecosystem is *user centric*, allowing the consumer to purchase, play and share digital content as they have grown accustomed in doing with physical media. Three major concepts form the foundation of the Ecosystem:

1. Users are able to purchase Content from multiple Retailers.
2. Multiple Users representing a household can be aggregated (grouped) into a single Account, enabling the sharing of Content between them.
3. Any User that is a member of the Account can acquire and play Content across set of devices associated with the Account.

In order to realize the concepts described above, the Ecosystem defines a set of entities that have well specified relationships and behavior. The entity at the center of the Ecosystem is the DECE Account. The DECE Account in turn manages three additional entities that are instrumental in enforcing the Ecosystem usage rules: The Rights Locker, Domain and a set of Users.

A Rights Locker stores all proofs of purchases, also known as Rights Tokens, for content purchased by any User associated with the Account. Rights Tokens are DRM-independent representations of the rights associated with an instance of purchased Content. All Users associated with the Account have access to the Rights Tokens in the Account’s Rights Locker including those that were purchases by other Users associated with the Account. A DECE Domain represents a group of DECE Devices and native DRM domain information. Each DRM-enabled Device associated with the Account is registered and joins the Domain. For each Device specific metadata such as DRM supported and video/audio capabilities is stored and made available via the architecture when necessary. In addition the Domain manages the collection of native DRM information – one for each Ecosystem-approved DRM – associated with the Account. This collection of DRM information is managed by a native DRM Client, and is represented to the Ecosystem with a DRM Domain Credential. This set of native DRM Domain Credentials forms a logical domain that enables the core DRM interoperability mechanism of the Ecosystem.

An Account is uniquely associated with a set of DECE Users. Each User is uniquely identified by the Ecosystem and Users authenticate themselves to the Coordinator via a User Credential. Retailers continue to manage their own retail accounts and login credentials as they do today, however in order to purchase Content a User must give a Retailer access to the DECE Account by authenticating to the Coordinator and optionally linking the Account to the Retailer. The Ecosystem makes use of a DECE User‘s identity to enable several key features, including access to streaming content for devices that are not a member of the Domain and parental control functionality. In addition the User is assigned one of three permission levels. Details of these concepts are further defined in Section 7.2.2.

The diagram below depicts these entities and relationships.

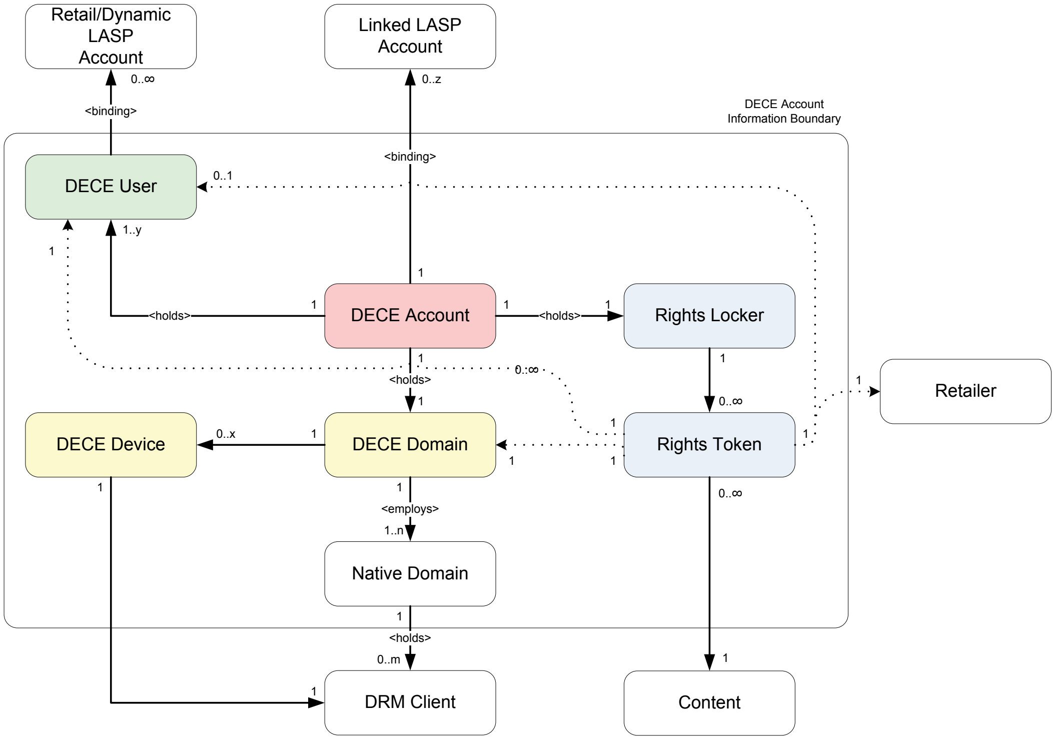


Figure 1 – Entity – Relationship Diagram

Entities within the DECE Boundary are managed by the Coordinator where entities outside of this boundary are managed by other service providers in the Ecosystem.

3.1 DECE Roles Overview

One of the underlying goals of the Ecosystem is to minimize the impact to the existing processes and procedures Content Owners and Retailers use to obtain, package, deliver, and license Content they sell to consumers. The DECE architecture is designed as a coordination layer on top of the existing retail content service offerings. Retail content service offerings will continue to obtain, package, deliver, and license Content to their customers pretty much as they do today.

In order to support new Ecosystem functionality the Retailers must augment their infrastructure to now support multiple domain-based DRM’s. In addition Retailers must now communicate with a global and central Ecosystem run service, known as the Coordinator, which enables the interoperability across Retailers, Devices and Users.

The architecture defines a set of Roles and their relations. The following diagram depicts these Roles and defines the high level architecture for the Ecosystem.

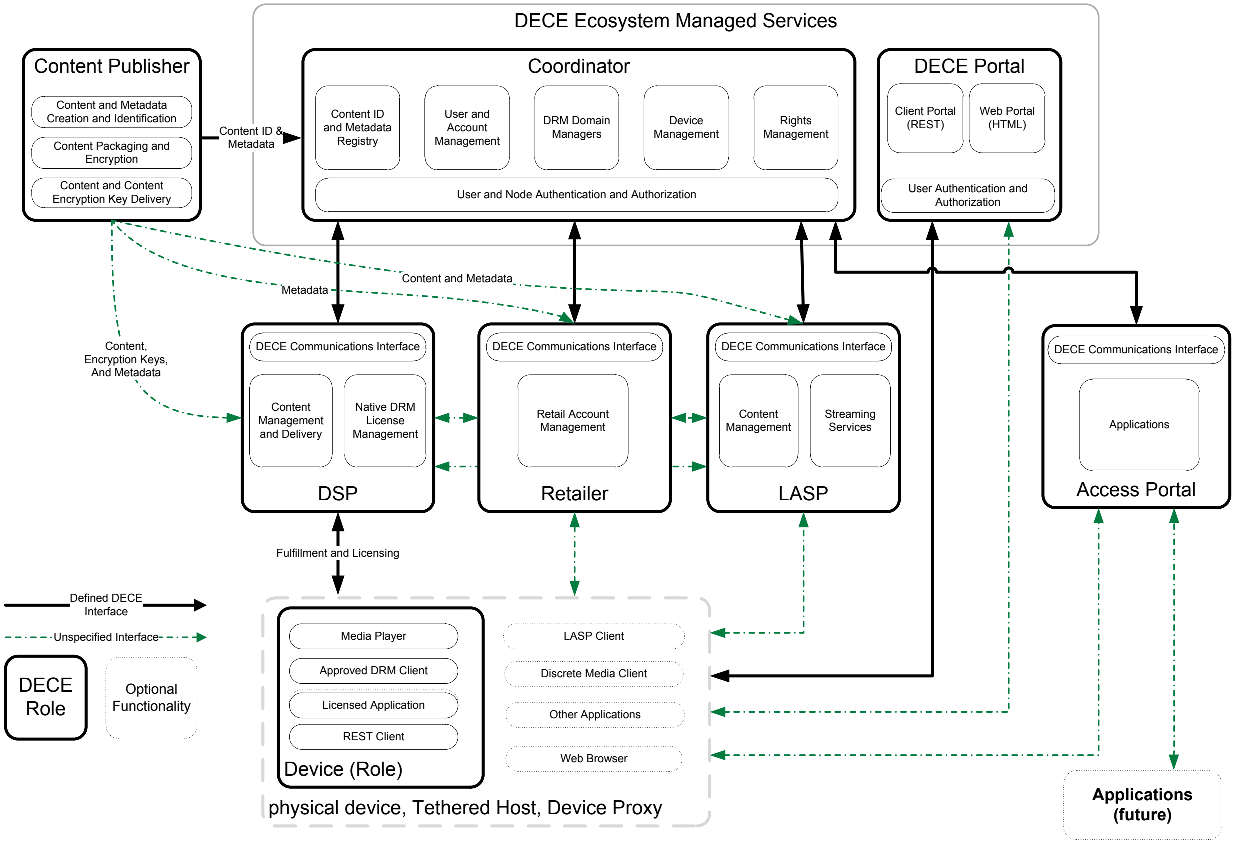

DECE Ecosystem Managed Services
Content Publisher
Content and Metadata Creation and Identification
Content Packaging and Encryption
Content and Content Encryption Key Delivery
Content ID & Metadata
Coordinator
Content ID and Metadata Registry
User and Account Management
DRM Domain Managers
Device Management
Rights Management
User and Node Authentication and Authorization
DECE Portal
Client Portal (REST)
Web Portal (HTML)
User Authentication and Authorization
Content and Metadata
Metadata
Content, Encryption Keys, And Metadata
DECE Communications Interface
Content Management and Delivery
Native DRM License Management
DSP
DECE Communications Interface
Retail Account Management
Retailer
DECE Communications Interface
Content Management
Streaming Services
LASP
DECE Communications Interface
Applications
Access Portal
Fulfillment and Licensing
Defined DECE Interface
Unspecified Interface
DECE Role
Optional Functionality
Media Player
Approved DRM Client
Licensed Application
REST Client
Device (Role)
LASP Client
Discrete Media Client
Other Applications
Web Browser
physical device, Tethered Host, Device Proxy
Applications (future)

Figure 2 – Ecosystem High Level Architecture

4 Roles

A *Role* is an entity that implements a specific set of Ecosystem functionality and both exposes and invokes a defined collection of interfaces. This section briefly describes each of the Roles that exist in the Ecosystem. Only companies with a valid license agreement with the DECE LLC may create instances of a Role in accordance with the assigned obligations of the Role.

4.1 The Coordinator Role

The Coordinator is a central entity operated on behalf of the DECE LLC that facilitates interoperability across Ecosystem services and stores/manages the Account. The Coordinator operates at a known Internet address.

The Coordinator Role enables interoperability between each of the other Roles in the Ecosystem. It manages the Ecosystem data and is responsible for enforcing the Ecosystem parameters globally. Communication with the Coordinator occurs using either a set of DECE-defined web service API's or via the Web Portal (a Coordinator-hosted consumer-facing user interface. It is important to note that the Coordinator does not manage, deliver, or license Content. This functionality is handled by the Retailer and the Retailer's DSP Role, defined in Section 4.2 and Section 4.3 respectively. The Coordinator provides *authorization* for content delivery, domain management, and license issuance whereas the DSP *manages, delivers, and licenses* content.

The functionality of the Coordinator role is split into several modules.

4.1.1 User/Account Management

As described earlier, the Coordinator is responsible for managing all of the DECE Accounts. Each Account contains one or more Users which are authenticated to the Ecosystem by a User ID and password.

Each User is associated with a set of attributes including standard fields such as first name, last name, email address, and the like. The User is assigned a single permission level, which is used to control access to Ecosystem data and services and an optional parental control setting, which is used to manage access to Content.

See Section 7.1 for further details on Accounts, and Section 7.2 for Users.

4.1.2 Domain/Device Management

The DECE Domain represents a group of Devices uniquely associated with a single Account. Each DRM-enabled device associated with the Account is registered and joins the Domain. The Domain manages the set of native DRM information associated with each Account. In effect, this set of native DRM information represents a “logical domain” that enables the core DRM interoperability mechanism of the Ecosystem.

The Coordinator runs domain management services for all of the Approved DRMs, coordinating the individual native DRM domains into the global DECE Domain. How it does this is described in Section 7.3.

Users can manage their Devices via their Retailer or LASP, and also directly via the Coordinator. Users can add new Devices to their Domain, remove existing Devices from their Domain, view the list of all Devices associated with their Domain and view, and update metadata associated with each Device.

4.1.3 Rights Management (Rights Locker)

The Rights Locker stores all proofs of purchases (excluding pricing information), also known as Rights Tokens, for content purchased by any User associated with the Account. Rights Tokens are DRM-independent representations of the rights associated with an instance of purchased Content. All Users associated with the Account have access to the Rights Tokens in the Accounts Rights Locker including those that were purchases by other Users. Other information about the User’s rights to Content is managed by the Rights Token, including the profile level of the content and an indication if the User has fulfilled the Discrete Media Right. Although Rights Tokens do not exist outside of the context of the Ecosystem, they are accessed, managed and manipulated via the web services interfaces exposed by the Coordinator role. Rights Tokens are used by LASPs, Retailers, and DSPs to authorize content acquisition and native DRM licensing.

4.1.4 Content ID and Metadata Registry

Content is made available for sale within the Ecosystem via Content Providers. To bootstrap this process Content Providers communicate the unique identifier and a small subset of descriptive and technical metadata, such as title and rating, to a Content Registry managed by the Coordinator. (See Section 9.1.2 for additional details.)

4.1.5 Device Portal

The Coordinator Role makes available a programmatic web services interface (referred to as the *Device Portal*) that exposes a subset of Coordinator functionality to Devices. The functionality of this web service interface includes joining (and leaving) the Device to the Account Domain, the ability to access the contents of the User’s Rights Locker, and the initiation of Container download (re-acquisition) based on those rights. See [DCoord] Appendix A for a complete list of the APIs supported by the Device Portal.

4.2 Retailer Role

The Retailer Role provides the customer-facing storefront service and sells Ecosystem-specific content to consumers. This typically includes providing the storefront and e-commerce functionality, managing the User's retail account and providing payment capabilities. When a Retailer sells DECE Content the Retailer Role is responsible for notifying the Coordinator of the details of the content sold to the User. The Retailer creates a unique Rights Token object that is passed to the Coordinator via a web service call for inclusion in the User's Rights Locker. This Rights Token can then be referenced for future interactions with the Ecosystem.

In addition to the Retailer specific requirements throughout this document, the following requirements are also normative.

The Retailer SHALL conform to protocols defined in [DCoord].

The Retailer SHALL authenticate with the Coordinator as described in [DCoord] Section 2.3 and [DSecMech].

Retailers SHALL ensure all DECE Rights obtained through them are licensable across all DECE Approved DRM's.

Note that a Retailer is not obligated to make its store front operational on every Device. But it is still responsible for every Device to be able to fulfill and license any rights sold through them for all Approved DRMs.

It is expected that Retailers will either build DSP Role functionality into their existing infrastructure themselves or partner with one or more service providers that will provide DSP functionality on their behalf. Interfaces between the Retailer and DSP are not defined by the DECE Specifications. A Retailer may use multiple DSPs serving different DRMs in order to satisfy the requirement that a Retailer support all the Approved DRMs.

The Retailer SHALL update a User's Rights Locker by creating a Rights Token as described in Section 10.1.1 when a User purchases a Right.

Retailers SHALL ensure all DECE Rights obtained through them can be fulfilled as described in Section 11.1.

A Retailer SHALL access the DECE Account with a Security Token as described in Section 7.1.2.

A Retailer MAY bind the Retailer account to the DECE Account as described in Section 7.1.2.

A Retailer SHALL NOT persistently store User Credentials (DECE Username and password) obtained solely for the purpose of creating a new User.

Accessing a DECE Account via a Security Token enables the `LockerViewAllConsent` policy, granting the Retailer access to an Account's entire Rights Locker regardless of the Retailer who originally sold the Right to the Content. See Section 7.1.2.2 and [DCoord] Section 5.5.

If the `LockerViewAllConsent` policy is enabled, when a Retailer displays an Account's Rights Locker the Retailer SHALL show all Rights Tokens, or provide a means to show all Rights Tokens, in an Account's Rights Locker regardless of whether the Right was sold by the Retailer.

The Retailer or its DSP SHALL write the Base Location to the Container as described in Section 8.3.2.2.

The Retailer or its DSP MAY write the Base Purl Location to the Container as described in Section 8.3.2.23.

A Retailer cannot Stream Content using its Retailer Node. In order to Stream Content, a Retailer must also be a LASP and Stream Content via its LASP Node. See Section 6.5.2 for information on how a Retailer that is also a LASP can support sharing a Security Token across Nodes.

4.3 The Download Service Provider (DSP) Role

The Retailer is obligated for delivery of Content to the Users through the DSP Role. The Retailer has the option to support this Role directly by building on top of existing backend infrastructure or to use third party or parties to meet their obligations. The DSPs responsibilities in the Ecosystem are threefold:

- The DSP is responsible for the local management of the latest copies of the native DRM Domain Credentials associated with each Domain. These DRM Domain Credentials are received from the Coordinator (i.e., the authoritative source) and made available to the local DRM License Managers.

- The DSP is responsible for setting up and managing License Managers for one or more of the Approved DRMs. They are responsible for issuing DRM Licenses with the correct keys required to decrypt Content associated with Rights Tokens in the Account. The use of the DRM Domain Credentials shared and received from the Coordinator enables multiple DSP's to issue a domain-based license to any of the Devices associated with the Domain.

- The DSP is responsible for the delivery of the encrypted Container. How the DSP receives the encrypted Container and associated metadata from the Content Provider is out of scope of DECE.

Note: There is no requirement for a single DSP to support all the DECE Approved DRMs. However, the Retailer Role does have the obligation to provide support for all the Approved DRMs either through a single DSP or through relationships with multiple DSPs.

The DSP SHALL conform to protocols defined in [DCoord].

The DSP SHALL authenticate with the Coordinator as described in [DCoord] Section 2.3 and [DSecMech].

The DSP SHALL support HTTP/1.1 [HTTP] and TLS 1.2 [TLS].

The DSP SHALL support HTTP/1.1 byte-range requests and SHALL send the “Accept-Ranges: bytes” header field for Fulfillment services.

A DSP SHALL NOT Stream Content. Only a LASP can Stream Content.

4.4 Locker Access Streaming Provider Role (LASP) Role

A *Locker Access Streaming Provider* (LASP) is defined as a streaming media service provider that participates in the Ecosystem and complies with DECE Policies to stream Content to devices. These devices may consist of user devices as well as devices operated by a service/system operator, e.g., Set Top Box, cellular phone, and general purpose computer.

Providing streaming services is an important capability of the Ecosystem because it allows Users flexible, remote, and real-time access to their purchased content. A LASP participates in the Ecosystem by allowing DECE Users to access their Rights Locker in order to authorize the LASP to stream their content to a LASP Client. As part of the Ecosystem, a LASP operates under a bilateral licensing agreement with Content Providers to acquire Content and provide this service. Content Providers have the option to grant streaming rights without the need for a bilateral agreement.

The Coordinator protocols required for a LASP to stream Content to a LASP Client are described in Section 13.2.

There are two categories of LASP services defined as *Linked* and *Dynamic*. A Linked LASP (LLASP) service streams to LASP Clients that are authenticated and persistently bound to a DECE Account. A Dynamic LASP (DLASP) service authenticates and is bound to a DECE Account at the User level.

There are three modes of LASP operation, each defined by different requirements. A LASP can service different LASP Clients using different modes.

Single-session User Mode

This mode implements a Dynamic LASP service for short-term, User-authenticated Streaming sessions, and is intended for transient environments such as web browsers, Internet cafes, and hospitality industry installations (providing Content access from hotel rooms, airplanes, etc.). Frequent re-authentication is required. See section 4.4.2.1 for details.

Persistent User-bound Mode

This mode implements a Dynamic LASP service for LASP Clients persistently associated with a single User, such as apps on a tablet or smartphone. Requirements are placed on LASP Clients serviced in this mode and limits are placed on the corresponding User-level Account bindings that a LASP can put in place. See section 4.4.2.1 for details.

Persistent Account-bound Mode

This mode implements a Linked LASP service for LASP Clients that may be used by multiple Users in an Account, and are therefore strongly associated with the Account, such as cable set-top boxes and applications on smart TVs. Requirements are placed on LASP Clients serviced in this mode and limits are placed on the corresponding Account-level bindings that a LASP can put in place. See section 4.4.3.1 for details.

4.4.1 General LASP Requirements

A LASP SHALL only Stream Content to a LASP Client.

A LASP SHALL conform to protocols defined in [DCoord].

A LASP SHALL authenticate with the Coordinator as described in [DCoord] Section 2.3 and [DSecMech].

A LASP SHALL access the DECE Account with a Security Token as described in Section 7.1.2.

A LASP MAY bind the LASP account to the DECE Account as described in Section 7.1.2.

A LASP SHALL NOT persistently store User Credentials (DECE User name and password) obtained solely for the purpose of creating a new User.

The protocol a LASP uses to stream Content to a device is out of the scope of the DECE. See Section 4.4.5 for requirements about protecting Streams.

A LASP can access an Account's entire Rights Locker regardless of the Retailer who originally sold the Right to the Content. See the `LockerViewAllConsent` policy in [DCoord] Section 5.5.

A LASP SHALL respect session stream limits. The number of simultaneous streams allowed per Account is limited. The LASP_SESSION_LIMIT parameter in Section 16 defines the current limit set by DECE policy. The Coordinator enforces this limit as described in Section 13.2.2.

Prior to streaming Content to a User, the LASP SHALL ensure the Rights Locker contains a Rights Token allowing the User to stream that Content. See the `CanStream` element in [DCoord] Section 7.2.5.

A LASP SHALL terminate all active Sessions upon unbinding from an Account.

A LASP MAY use the Digital CFF Container for streaming, or it MAY use an alternate format.

A LASP can only Stream Content. That is, a LASP is not permitted to sell Rights to Content or to provide fulfillment services.

A LASP SHALL provide DECE Account Management functions in accordance with the LASP Compliance Rules. The LASP MAY either refer the user to the DECE Web Portal, or provide an interface using the Coordinator APIs ([DCoord] Section 13), except in Persistent User-Bound Mode, where a custom interface must be provided (see Section 4.4.2.2).

A LASP providing an Account Management interface using Coordinator APIs SHALL at minimum allow setting the User's e-mail address and password. Account Management consent (ManageAccountConsent + ManageUserConsent) is not required for this functionality, but in the case where such consent has not been given to the LASP, the LASP SHALL provide a set-only feature and SHALL NOT display the existing email address and password.

Note: Until a future release of the Coordinator, LASP ability to set these fields may be blocked in the absence of Account Management consent, in which case the latter requirement above does not apply.

4.4.2 Dynamic LASP

A Dynamic LASP is a LASP service that streams Content to a LASP Client to an authenticated User. Authorization to stream content from a Dynamic LASP is obtained by authenticating the User on a session-by-session basis for Single-session User Mode or by binding to the Account at the User level for Persistent User-bound Mode. An example of Dynamic LASP streaming would be the streaming of Content to a PC from an online streaming service or streaming of Content to a hotel room TV. Dynamic LASPs determine what Content may be streamed to a User by ensuring that the User is a member of the corresponding Account associated with the Rights Token.

The Coordinator ensures a User has at least the Standard-Access permission level to create a Dynamic LASP session. See Section 7.2.2 for details on User Access Levels.

The Coordinator uses the User's Parental Control Information to filter the Rights Locker view and to restrict Streaming.

4.4.2.1 LASP Requirements for Single-session User Mode

A LASP choosing to service a LASP Client in Single-session User Mode SHALL meet all of the following requirements whenever it is servicing such LASP Client.

The LASP SHALL interface with the Coordinator using the Dynamic LASP (DLASP) subrole.

The LASP SHALL only bind to the DECE Account at the User Level. (Note that Account binding is optional for this mode.)

Before Streaming can begin, the LASP SHALL require the User to authenticate directly to the Coordinator using their User Credential or indirectly to the Coordinator through the LASP using their LASP credential, according to DYNAMIC_LASP_AUTHENTICATION_DURATION (see Section 16). In other words, if it has been more than DYNAMIC_LASP_AUTHENTICATION_DURATION since a previous User authentication or if there was no previous User authentication, the LASP SHALL require the User to authenticate using one of the two specified methods.

The LASP SHALL make reasonable efforts, where possible, to prevent credential caching at user agents and to force re-authentication to occur at user agents. For example, random field names in HTML forms may prevent Browsers from storing passwords.

4.4.2.2 LASP Requirements for Persistent User-bound Mode

A LASP choosing to service a LASP Client in Persistent User-bound Mode SHALL meet all of the following requirements whenever it is servicing such LASP Client.

The LASP Client, including user interface and authentication methods, SHALL be under the control of the LASP.

The LASP Client SHALL be persistently tied to an account at the LASP service.

The LASP SHALL interface with the Coordinator using the Dynamic LASP (DLASP) subrole.

The LASP SHALL only bind to the DECE Account at the User Level.

Each LASP Account bound for use in this mode SHALL only be bound to Users within a single DECE Account. That is, binding from a single LASP Account using the Dynamic LASP subrole can occur multiple times but only to a single DECE Account.

The LASP SHALL bind to a maximum of DYNAMIC_LASP_PERSISTENT ACCOUNT_LIMIT DECE Users at a time (see Section 16). The LASP SHALL enforce this limit (as opposed to Persistent Account-bound Mode, where the Coordinator enforces a similar limit).

The LASP SHALL provide DECE Account Management functions, at minimum the ability for the User to set email address and password, as described in section 4.4.1. Such functions SHALL be accessible directly from every LASP Client serviced in Persistent User-bound Mode and SHALL NOT require separate or additional authentication other than what is required for Streaming.

If the Coordinator signals to the LASP via an error message that authentication is required, the LASP SHALL require the User to authenticate directly to the Coordinator using their User Credential or indirectly to the Coordinator through the LASP using their LASP credential.

4.4.3 Linked LASP

A Linked LASP service is for LASP Clients that are persistently bound and provisioned to a single DECE Account versus a User, as Linked LASP services are not associated with a particular User but to a household Account. Because the linkage is to an Account versus a User, and because provisioned devices often don’t have interfaces conducive to logging in with a username and password, it is not practical to require a User to authenticate on a session-by-session basis. Examples of LASP Clients supported with Linked LASP service would be a mobile phone with a built-in app associated with the mobile carrier service or a cable set-top box Streaming over a proprietary cable conditional access system. A Linked LASP service is used for Persistent User-bound Mode.

The Coordinator ensures that a User has at least the Standard-Access permission level to bind their Account to a Linked LASP service or to delete a binding. See Section 7.2.2 for details on User Access Levels.

4.4.3.1 LASP Requirements for Persistent Account-bound Mode

A LASP choosing to service a LASP Client in Persistent Account-bound Mode SHALL meet all of the following requirements whenever it is servicing such LASP Client.

The LASP Client, including user interface and authentication methods, SHALL be under the control of the LASP.

The LASP Client SHALL be persistently tied to an account at the LASP service.

The LASP SHALL interface with the Coordinator using the Linked LASP (LLASP) subrole.

The LASP SHALL only bind at the Account Level.

Each LASP Account bound for use with a Linked LASP service SHALL only be bound to a single DECE Account. That is, binding from a single LASP Account using the Linked LASP subrole can only be to a single DECE Account.

The LASP SHALL bind to a maximum of LINK_LASP_ACCOUNT_LIMIT DECE Accounts at a time (see Section 16). The Coordinator enforces this limit.

A Linked LASP is limited in how often it can rebind to an Account it had been previously bound to. The LINK_LASP_ACCOUNT_FLIPPING_LIMIT parameter in Section 16 defines this maximum frequency. The Coordinator enforces this limit.

The Linked LASP SHALL offer Ratings Enforcement as specified in the LASP Compliance Rules.

Ratings enforcement support is completely provided by what the LASP implements in the service and in the LASP Client. How it does it is out of the scope of DECE. The Coordinator returns all Rights in the Rights Locker for the Account to the Linked LASP.

4.4.4 LASP Authorization

Content Providers may choose to make some of their Content available for Streaming without requiring bilateral agreements as long as requirements of the LASP agreements are met. The Content Provider indicates which Content can be Streamed in this manner by setting the `AssentStreamAllowed` element in the `LogicalAsset` resource for the Content's ALID and Media Profile in the Coordinator. See [DCoord] Section 6.5.

A LASP MAY Stream Content whose ALID has a true `AssentStreamAllowed` element. A LASP using this feature is sometimes called a Checkbox LASP.

4.4.5 Stream Protection Technologies

A LASP SHALL protect a Stream in one of the following ways:

- By using an Approved DRM (listed in Appendix B, Section 17)
- By using an Approved Stream Protection Technology (listed in Appendix C, Section 18)
- Or by using a content protection technology in accordance with the bilateral agreements between the LASP and the Content Provider.

4.5 DECE Portal Role (Web Portal)

Consumers of DECE content are able to interact with the Ecosystem via the DECE Portal Role. This role makes available an interactive web application (referred to as the *Web Portal*) for the DECE consumer brand and gives Users direct access to Account settings such as a view of their Rights, management of Users in their household account and the ability to add and remove Devices via the use of standard web browsers.

The DECE Portal Role is separate from the Coordinator role to enable, if desired, an entity or organization other than the Coordinator operator to build and manage the consumer facing user experience. Over time, multiple Web Portal Roles may exist, running perhaps in parallel, to enable multiple user experiences that cater to different to environments – ranging from rich interactive environments based on Flash or Silverlight to simple no-frills user experiences built for constrained mobile devices connected to low-bandwidth high-latency networks. The Web Portal Role leverages the same DECE defined B2B interfaces used by other Roles in the Ecosystem such as a Retailer, LASP or DSP. However in order to provide the best experience for the consumer this Role may also use interfaces not available to other Roles.

Access to all of the functionality provided by this Role is based on authentication of the User via their DECE User Credentials.

4.6 Content Provider Role

The Content Provider Role is the authoritative source for all DECE Content and is implemented and run by the various content owner or their partners. The Content Provider Role is responsible for:

- Content and Content Metadata creation and Identification,
- Encoding and encryption of Content into a Digital CFF Container,
- Delivery of Containers, Content Metadata and Content Encryption Key(s).

Once the Content Provider completes the Content Publishing process, as defined in [DPublisher] it is available for use by Retailers, DSP's and LASPs. As shown in Figure 2, while the [DPublisher] will define the behavior required of the Content Provider, including how content is created, encoded, encrypted, and what data will be communicated to various DECE Roles, it will only normatively define how content metadata and identifiers are conveyed between the Content Provider and Coordinator. How data is communicated to other Roles in the Ecosystem will not be defined by the DECE Ecosystem.

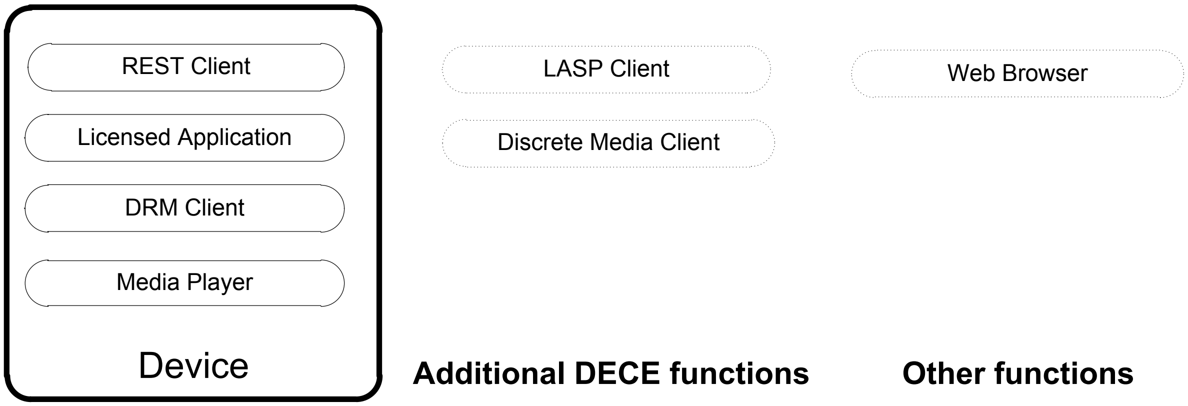
4.7 Device Role

The Device Role defines functionality for a consumer product that contains one or more DECE-approved Licensed Applications and one or more Approved DRM Clients in a single DECE Domain, which collectively complies with applicable specifications. (See Section 4.7.3 for more information about Approved DRM Clients.)

A DECE Device (or Device) is a hardware or software product or combination of products that implement a Device Role. DECE Devices include a DRM Client and a Licensed Application.

Devices in the Ecosystem must be a member of at most one DECE Account. To join a DECE Account, a Device must support one of the Approved DRMs (Section 17) and thus must have an installed DRM Client. Devices must also support the DECE media format defined in [DMedia].

The following diagram illustrates a DECE Device. As shown, it contains a REST Client, Licensed Application and a DRM Client and a Media Player. The REST Client communicates with the Device Portal. The Media Player interacts with the DRM Client, the Licensed Application or both. Although not part of the Device Role, LASP Clients and Discrete Media Clients may coexist with the Licensed Application. A Web Browser can complement a Licensed Application, as can an Access Portal client (see Section 4.7.2 and Section 4.8 for information about an Access Portal). Content is downloaded using a Download Manager (part of a Licensed Application) and optionally a Browser or another application out of scope of DECE.



4.7.1 DECE Device

A DECE Device's Licensed Applications and DRM Clients may be in only one DECE Domain. A Licensed Application and its DRM Clients may be on only one DECE Device. Note that in many cases, the Coordinator recognizes that Licensed Applications and a common DRM Client are on a single physical device and counts them as only one DECE Device with respect to Account limits. In other cases, in the perspective of DECE, Licensed Applications with multiple DRM Clients are seen as multiple DECE Devices even when on the same physical device. For example, on a mobile device, apps using distinct DRM Clients would each count as one Media Client with respect to Account Limits.

Currently a DECE Device supports:

- A single Licensed Application and a single DRM Client
- Multiple Licensed Applications and a single DRM Client

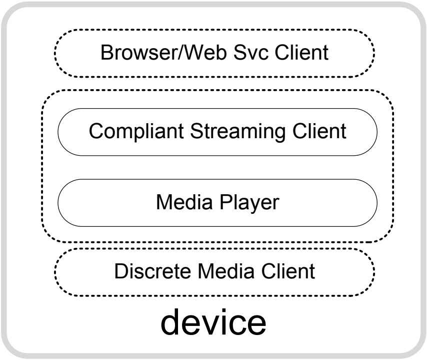
Note that some DRM systems support the concept of a “DRM platform” on a physical device, where a single native DRM domain is shared across applications on the physical device. In some cases, each application may have its own instance of a DRM client linked into the application; however, for the purposes of DECE, the DRM system will identify the DRM platform as a single DRM Client Role with a single DRM Client ID to the Coordinator, and DECE considers this to be a case of multiple Licensed Applications sharing a single DRM Client.

Multiple applications accessing multiple Approved DRM systems are currently treated as multiple DECE Device instances. A physical device with multiple Approved DRMs is supported by the Ecosystem as multiple DECE Devices. This restriction may be eliminated in a future release.

The term *DECE Device* or *Device* is used to refer to an entity that complies with applicable DECE requirements, legal, business and technical.

As the DECE Device contains the Licensed Application, DRM Client and other components, unless otherwise stated, requirements that address the DECE Device are not specifically directed to any particular component. That is, the requirement may be satisfied by the Licensed Application, DRM Client or any other component that is part of that DECE Device.

The term *device* may be used to refer to both DECE Devices and consumer products that do not meet DECE’s definition of a DECE Device. The following figure illustrates a device with functions applicable to DECE, yet not including the necessary functionality to be a DECE Device as the “Media Player” is not a Licensed Application, and it does not have an Approved DRM Client.



4.7.2 Connected Devices, Tethered Devices and Proxied Devices

DECE allows the Device Role to be distributed across hardware and onto servers with certain constraints. DECE Devices that have an Internet connection (not necessarily always available) and support the DECE communications protocols necessary to perform all Device interactions with DECE servers are called *Connected DECE Devices*.

Other DECE Devices that depend on another device under the control of the User (often a general purpose computer) to communicate with the Coordinator through the Device Portal, are called *Tethered DECE Devices*. This terminology is in reference to their tethering to another device via a local connection, for example using a USB cable. The general purpose computer or any other device to which a DECE Device is tethered is called a *Tethered Host*.

DECE Devices may use a server under the management of the Client Implementer to communicate with the Device Portal. The server is called a *Device Proxy*. These DECE Devices are said to be *Proxied Devices*.

Unless specifically referring to a “Connected” or “Tethered” Device, DECE specifications use the term *DECE Device* to refer to the Device Role functionality regardless of where it is implemented; that is, the physical device, the Tethered Host, or the Device Proxy.

A Device Proxy MAY request and temporarily store User Credentials.

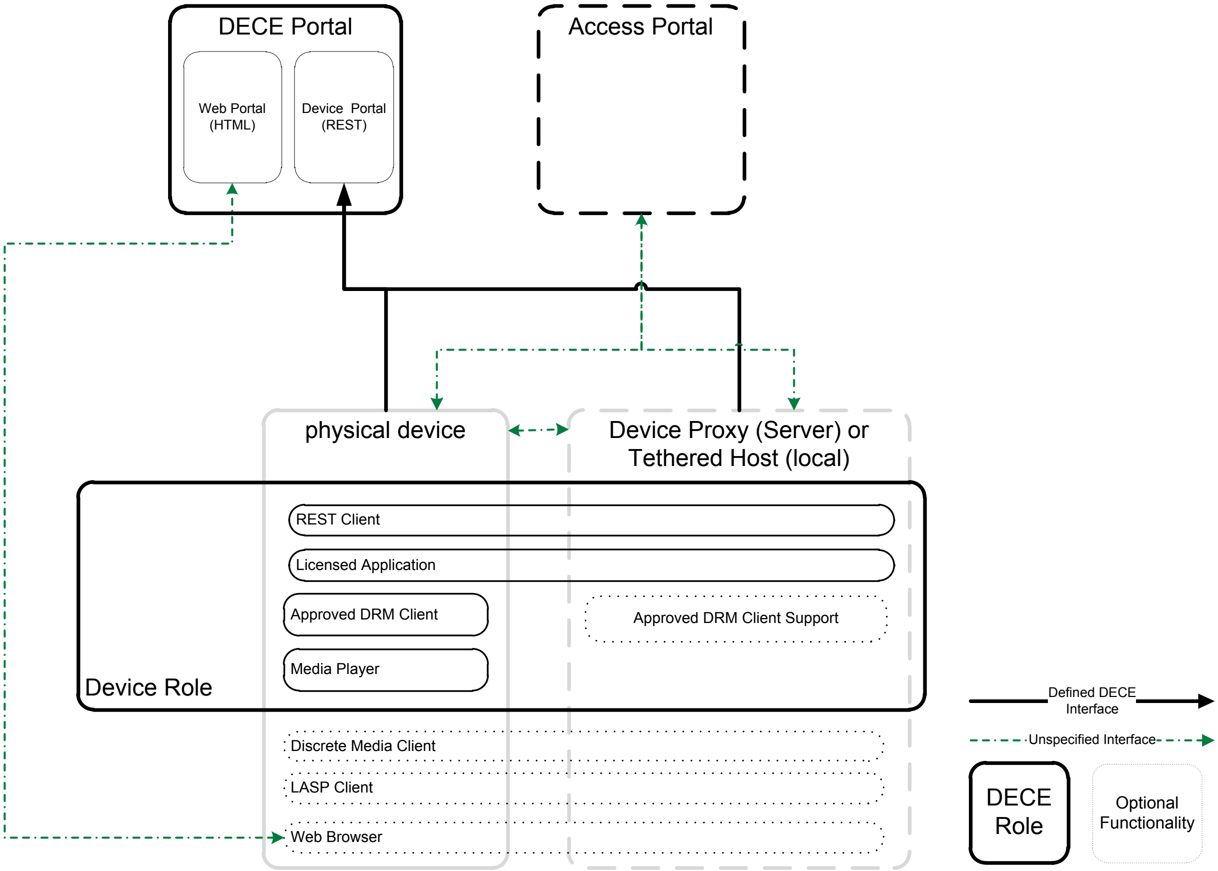
A Device Proxy MAY access the Coordinator on behalf of a User of a Proxied Device to obtain and store a User Security Token from the Coordinator.

How a Device Proxy joins a DECE Device to a DECE Domain is described in Section 7.3.3.1.3.

How a Device Proxy causes a DECE Device to leave a DECE Domain is described in Section 7.3.4.1.

DECE Devices may provide additional functionality such as Account Management using a server under the management of the Client Implementer that communicates with the Coordinator. The server is called an Access Portal (see Section 4.8).

The following illustration shows how functionality can be distributed:



4.7.3 Approved DRM Client

A DRM Client is a native DRM agent—it handles all functions related to the Digital Rights Management function of the DECE Device. Decryption and policy enforcement is provided by the DRM Client. DECE uniquely and securely identifies each DRM Client.

DECE has approved several DRM systems for use in DECE Devices. Each of these is referred to as an “Approved DRM” (see Section 17).

An *Approved DRM Client* (referred to as a *DRM Client*) uses an Approved DRM. Functions of a DRM Client includes domain management, key management, license management, content decryption, and anything else required to make DRM encrypted content available to the Media Player in a decodable form.

4.7.4 HD and SD Devices

Not all Devices can play all Media Profiles. The Media Profiles are: HD (high definition) and SD (standard definition).

A DECE Device is an ‘HD Device’ or ‘SD Device’.

An HD Device is a DECE Device capable of playing HD and SD profile Containers.

An SD Device is a DECE Device capable of playing SD profile Containers, but not HD Containers.

4.8 Access Portal Role

An Access Portal is an application or service that provides User access to DECE functions such as User and Account management, Device management, and so on, similar to the access that may be provided by a Retailer, LASP, or Web Portal.

A Client Implementer may additionally take the Role of an Access Portal provider to implement applications integrated with a DECE Device to enable Coordinator functions beyond the basic Device Portal functions such as Locker view.

The interface between the Access Portal and applications that connect to it is not specified by DECE.

5 URN Structure & Identifiers

DECE Universal Resource Names (URN) structure used for identifiers and other purposes SHALL conform to RFC 3986 and RFC 3305 [URI]. It SHALL use the "dece" namespace identifier (NID). The basic structure for a DECE URN is:

```
<DECEURN> ::= "urn:dece:" <type> ":" <type-dependent>
```

- <type> is the type of identifier. The value of this token SHALL be one of the tokens defined in the table below. Note that some type values have the literal prefix, "type:".
- <type-dependent> is a string that is defined by the DECE specification provision that defines the <type>.

DECE URNs are case insensitive except where the definition of a particular <type> specifies case requirements for the <type-dependent> portion of the DECE URN.

When using URN's in a URL (e.g. HTTP requests), clients SHALL ensure the URL conforms to the encoding provisions of [URI].

Type	Defined In
accountid	[DCoord] Section 13.1.1.1
alid	Section 5.5.1.1
apid	Section 5.5.1.2
applicationid	Section
bid	Section 5.5.3
cid	Section 5.5.2
container	[DMedia] Section 6.4.2
dmpid	Section 5.5.5
domainid	Section 5.4.2
drm	Section 5.4.1
drmclientid	Section 5.4.3
errorid	[DCoord] Appendix B
licappid	Section 5.4.4
org	Section 5.2
presentationid	Section 5.5.4
protocolversion	[DCoord] Appendix C
pskc	[DKeyDelivery] Section 4

Type	Defined In
rightslockerid	[DCoord] Section 13.3
rightstokenid	[DCoord] Section 11.2.2
role	[DCoord] Section 2.3.3
streamid	[DCoord] Section 11.1.1.2
type:**discretemediaformat**	[DCoord] Section 16.2.5
type:**false**	[DCoord] Section 17.8
type:**geoloc**	[DCoord] Section 17.8.2
type:**geoprofile**	[DGeo] Section 2.4
type:Media**Profile**	[DCoord] Section 6.5.3
type:**policy**	[DCoord] Section 5.4.1
type:**rating**	[DCoord] Section 5.5.5.3
type:**state**	[DCoord] Section 16.2.3
type:**status**	[DCoord] Section 17.2.1
type:**tokentype**	[DSecMech] Section 5
type:**transaction**	[DCoord] Section 17.9
type:**true**	[DCoord] Section 17.8
type:**unknown**	[DCoord] Section 17.8
type:**viewfilter**	[DCoord] Section 3.15
userid	[DCoord] Section 14.1.2.3

DECE requires the use of multiple types of identifiers. In most cases, the only requirement for identifiers is that they be unique within the Ecosystem. That is, two objects exchanged by DECE components using DECE interfaces will only use the same ID if they refer to the same entity. IDs often must be persistent. That is, the identified entity will always be referred to by the same identifier.

5.1 DECE Identifier Structure

5.1.1 Internal Coordinator Managed/Assigned Identifiers

Identifiers of this type are assigned by the Coordinator and represent a unique entity/resource within the Ecosystem. These identifiers are used to build the Path value defined for each interface.

5.1.2 Ecosystem Assigned Identifiers

These identifiers are manually assigned by DECE. That is, DECE administrative personnel explicitly assign them in accordance with rules here and with DECE policies. DRM and Profile Identifiers will be assigned based on which DRM and profile are approved for use in the Ecosystem. Retail, LASP and DSP identifiers uniquely identify organizations who have executed the corresponding license agreements.

5.1.3 Content Identifiers

A Content Identifier is a DECE URN with a <type> of one of the following:

- "alid" (An Asset Logical Identifier. See Section 5.5.1.1)
- "apid" (An Asset Physical Identifier. See Section 5.5.1.2)
- "bid" (A Bundle Identifier. See Section 5.5.3)
- "cid" (A Content Identifier. See Section 5.5.2)
- "presentationid" (A Media Presentation Identifier. See Section 5.5.4)
- 'dmpid' (A DECE Media Package Identifier. See Section 5.5.4)
- applicationid

Content Identifiers must be unique throughout the Ecosystem.

The basic structure for a Content Identifier is:

```
<DECEID> ::= <DECEURN> ":" <scheme> ":" <SSID>
```

- <scheme> is either a DECE recognized naming scheme (e.g., "ISAN") or "org" non-standard naming. These are specific to ID type and are therefore discussed in sections addressing IDs of each type.
- <SSID> (scheme specific ID) is a string that corresponds with IDs in scheme <scheme>. For example, if the scheme is "ISAN" then the <SSID> would be an ISAN number.

There is a special case where <scheme> is "org", allowing organizations to use their own identifier scheme. See Section 5.1.3.2.

All Content Identifiers are subject to the following requirements:

- The Content Identifier together SHALL NOT refer to more than one Asset.

- The <scheme> SHALL NOT contain a colon (":") character.
- The <scheme> MAY be any unique name of an identifier standard, with the corresponding SSID being the unique asset identifier defined by that scheme.
- The <SSID> MAY contain a single colon (":") character. Other URN reserved characters SHALL be escape encoded.

If the <scheme> is one of the schemes listed in the "Scheme" column of Table 3, then the <SSID> SHALL be constructed to conform to the requirements listed in the corresponding "Section" column of the same table.

Scheme	Expected value for <SSID>	Section
EIDR-S	Entertainment Identifier Registry [EIDR]. EIDR-S is a shortened EIDR that does not include the "10.5240/" prefix.	5.1.3.3
EIDR-X	Entertainment Identifier Registry [EIDR]. EIDR-X is a shortened EIDR that does not include the "10.5240/" prefix, along with an additional extension.	5.1.3.4
org	<SSID> begins with the Organization Name of the assigning organization and follows with a string of characters that provides a unique identifier.	5.1.3.2

Table 3 – Content Identifier Schemes with Normative Requirements

Table 4 shows other commonly used schemes for Content Identifiers and their SSID.

Scheme	Expected value for <SSID>
AMG	AMG
DOI	Digital Object Identifier, http://www.doi.org
EIDR	Entertainment Identifier Registry; see section 2.4 in [EIDR]. Since EIDR uses a forward slash ("/") character in the SSID, it is recommended that the "eidr-s" or "eidr-x" scheme be used instead.
EIDRS	The use of the "eidrs" scheme is deprecated and should not be used. Use "eidr-s" or "eidr-x" instead.
File	Indicates that the identifier that follows is a local file name.
grid	A Global Release identifier for a music video; exactly 18 alphanumeric characters
IMDB	IMDB
ISAN	An <ISAN> element, as specified in ISO15706-2 Annex D.
ISBN	An ISBN, ISO 2108, http://www.isbn-international.org
ISMN	Printed music, ISO 10957, http://ismn-international.org/
ISRC	Master recordings, ISO 3901, http://www.ifpi.org/content/section_resources/isrc.html

Scheme	Expected value for <SSID>
ISSN	Serials. ISO 3297:1998.
ISTC	Textual works. ISO 21047
ISWC	Musical Works, http://www.cisac.org
MUZE	Muze
TRIB	Tribune
TVG	TV Guide
URI	A URI; this allows compatibility with TVAnytime and MPEG-21
UUID	A UUID in the form 8-4-4-4-12

Table 4 – Content Identifier SSIDs

Some sample identifiers are:

Organization ID	urn:dece:org:org:dece:mycompany
Content ALID	urn:dece:alid:ISAN:000000018947000000000000
Content ALID	urn:dece:alid:org:mystudio:12345abcdef

5.1.3.1 Content Identifier SSID Canonicalization

If a Content Identifier scheme of Table 3, with the exception of <UID> of "org", incorporates external registry identifiers, and the syntax rules of that registry allow optional characters that are not considered part of the formal syntax, such identifiers SHALL be canonicalized to their normal form, in accordance with that identifier's canonicalization requirements.

If a Content Identifier scheme of Table 4 incorporates external registry identifiers, and the syntax rules of that registry allow optional characters that are not considered part of the formal syntax, such identifiers SHOULD be canonicalized to their normal form, in accordance with that identifier's canonicalization requirements.

5.1.3.2 "org" Scheme Requirements

The "org" scheme allows organizations recognized by the DECE organization to assign IDs using their own naming conventions. If <scheme> is "org" then:

```
<SSID> ::= <organization>":"<UID>
```

- <organization> SHALL be the Organization Name assigned by DECE to an organization. See Section 5.2.1.

- <UID> is a unique identifier assigned by the organization identified in <organization>. Organizations may use any naming convention as long as it complies with RFC 3986 [URI] syntax.

When DECE assigns identifiers, <organization> is “dece” and an ID would have the form:

```
“urn:dece:”<type>”:org:dece:”<UID>
```

5.1.3.3 EIDR-S Scheme Requirements

EIDR-S is used when an Entertainment Identifier Registry identifier [EIDR] is desired. To avoid the use of the ‘/’ character in a DECE Content Identifier, the EIDR-S scheme is recommended as a shortened version of EIDR that does not include the “10.5240/” prefix.

If the Content Identifier scheme is EIDR-S, the following requirements SHALL apply:

- <scheme> SHALL be “eidr-s”.
- <SSID> SHALL be the canonical form for the DOI suffix defined in [EIDR] section 1.2 with the following stipulations:
    - The DOI prefix and terminating “/” (the “10.5240/” prefix) SHALL NOT be included in the <SSID>.
    - The normalization rules defined in [EIDR] section 1.2 SHALL be followed. For example, the EIDR ID is required to be normalized to upper case, the hyphens in the ID are required, and the check character is required to be included.

Examples of eidr-s identifiers are:

eidr-s ALID	urn:dece:alid:eidr-s:50A5-34E1-4FFF-0BBD-17C9-G
eidr-s ContentID	urn:dece:cid:eidr-s:1E63-2E9A-11AB-FE88-1B89-M

5.1.3.4 EIDR-X Content Identifier Requirements

In order to create an arbitrary number of Content Identifiers for a given EIDR, it is necessary to create an identifier that is the concatenation of an EIDR with additional unique information in the form of an extension.

The recommended solution is the creation of the ‘eidr-x’ scheme. This scheme is an eidr-s identifier concatenated with an alphanumeric <extension>. The identifier creator must ensure the EIDR identifier and extension taken together is a unique identifier for an Asset. The identifier creator must establish a best practice to avoid collisions if multiple identifier creators are using extensions with the same EIDR identifier.

If the Content Identifier scheme is EIDR-X, the following requirements SHALL apply:

- <scheme> SHALL be "eidr-x".
- <SSID> SHALL be the <SSID> defined for the EIDR-S scheme in Section 5.1.3.2, followed by a ":" and an <extension>. The <extension> SHALL be an alphanumeric string (consisting of A-Z, 0-9 characters, case insensitive).

Examples of eidr-x identifiers are:

eidr-x ALID	urn:dece:alid:eidr-x:50A5-34E1-4FFF-0BBD-17C9-G:1
eidr-x ALID	urn:dece:alid:eidr-x:50A5-34E1-4FFF-0BBD-17C9-G:france
eidr-x ALID	urn:dece:alid:eidr-x:50A5-34E1-4FFF-0BBD-17C9-G:123abc

5.1.4 ID Assignment

The following table shows the ID and which entity is responsible for generating the values to assign to an ID. The entity can be the Coordinator, Ecosystem or Content Provider.

Category	ID	<type>	Assignment
Organization/Role			
	Organization Name	N/A	Ecosystem
	OrganizationID	org	Ecosystem
	Role	N/A	Ecosystem
User/Account			
	AccountID	accountid	Coordinator
	UserID	userid	Coordinator
	RightsLockerID	rightslockerid	Coordinator
	RightsTokenID	rightstokenid	Coordinator
	StreamID	streamid	Coordinator
DRM/Device/Domain			
	DomainID	domainid	Coordinator
	DRMClientID	drmclientid	Coordinator
Content			
	AssetLogicalID	alid	Content Provider
	AssetPhysicalID	apid	Content Provider

Category	ID	<type>	Assignment
	ContentID	cid	Content Provider
	BundleID	bid	Content Provider, Retailer

Table 5 – Identifier Type and Assignment

5.2 Organization Identifiers

This section describes identifiers associated with Organizations and Roles.

5.2.1 Organization Names

Organizations are identified uniquely by an *Organization Name* which is assigned by DECE as part of an organization entering the Ecosystem.

Organization Names are two or more characters up to a maximum of 63 characters. Since Organization Names can also be used as part of an internet domain name (see Section 8.3.3 for an example), they are limited to only using upper and lowercase letters and decimal digits as defined by [URI]. Graphic symbols normally allowed by [URI] including hyphen, period, underscore, and tilde and percent-encoded data octets are SHALL NOT be used for an Organization Name. For example a space cannot be added such as: “my%20company”. As with all DECE identifiers, Organization Names are case insensitive.

For example, “mycompany” and “best4you” are examples of Organization Names.

Organization Names are used along with “org:” for other types of identifiers and in Role IDs as well. For example:

ALID	urn:dece:alid:org:mycompany:abcdefg
Retailer Role ID	urn:dece:retailer:mycompany

5.2.2 Organization IDs

An Organization ID is of the form:

```
“urn:dece:org:org:dece:”<organization>
```

- <organization> is the Organization Name as defined in Section 5.2.1.

Note that <type> is “org”, the <scheme> is “org” denoting a private naming authority as described in Section 5.1, and the <SSID> is “dece:<organization>” as DECE is the only valid naming authoring for Organization IDs at this time.

Organization ID	urn:dece:org:org:dece:MYCOMPANY

5.3 User and Account-related Identifiers

All these IDs are assigned by the Coordinator. <type> shall be in conformance with Table 5 – Identifier Type and Assignment above. The <SSID> of these IDs is at the discretion of the Coordinator. They must be unique throughout the Ecosystem.

5.4 Device and DRM Identifiers

5.4.1 DRM Name and DRM ID

A DRM name is a DECE assigned name for each DRM as defined in Appendix B (Section 17).

A DRM ID is of the form:

```
“urn:dece:drm:”<DRM name>”:”<DRM version>
```

- `<DRM name>` is from Table 29 – Approved DRM List in Section 17.
- <`DRM version`> is an identifier representing a specific system version of an Approved DRM implementation from Table 29 – Approved DRM List in Section 17.

5.4.2 DomainID

A DomainID is the Coordinator identifier used to identify a domain within a given DRM. More specifically, there is a one to one correlation between the DRM Domain ID and the DRM Domain Credential. The DomainID is referred to as the DRM Domain ID in this document (see Section 7.3.2).

DomainIDs are of the form:

```
“urn:dece:domainid:”<DRM name>”:”<DRM-specific Domain ID>
```

- <DRM name> is a DRM Name from Table 29 – Approved DRM List in Section 17.
- <DRM-specific Domain ID> is a UTF-8 string created by the Coordinator to identify the DRM domain. The syntax of the string varies on a per-DRM basis.

5.4.3 DRMClientID

DRMClientIDs identify a DRM Client within the Ecosystem. These are globally unique.

DRMClientIDs are of the form:

```
"urn:dece:drmclientid:"<DRM name>":"<DRM-specific DRM Client ID>
```

- `<DRM name>` is a DRM Name
- `<DRM-specific DRM Client ID>` is a UTF-8 encoded string whose form is specific to the DRM, and is assigned by the DRM system to uniquely identify the DRM Client within the DECE ecosystem.

5.4.4 LicAppID

LicAppIDs identify a Licensed Applications within the Ecosystem. These are globally unique.

LicAppIDs are of the form:

```
"urn:dece:licappid:"<LicApp-specific LicApp ID>
```

- < LicApp-specific LicApp ID> is an identifier assigned by the Coordinator representing a specific Licensed Application instance.

5.5 Content Identifiers

Content Identifiers are assigned by Content Providers, independent of the Coordinator. However, they must be globally unique within the Ecosystem. The following scheme provides flexibility in naming while maintaining uniqueness.

5.5.1 Asset Identifiers

DECE maintains several types of asset identifiers:

- An Asset Logical Identifier (ALID) denotes an abstract representation of a content item. An ALID is referred to in a Rights Token, indicating the media object for which rights have been obtained. Each ALID must have at least one Media Profile.
- Asset Physical Identifier (APID) refers to a physical entity (i.e., a Digital CFF Container) for a single Media Profile that is associated with a logical asset. The APID is structured to be included in the container. An APID is sufficient identification for a DRM system to determine a license.

The following describes the current assumptions for relationships between ALIDs, APIDs and file names. If the assumptions change, the naming rules may also change

- An ALID is referred to in a Rights Token as the media object for which rights have been obtained.
- The actual Right is an ALID/profile pair.
- An ALID explicitly refers to one or more physical assets. That is, ALIDs map to one or more APIDs.
- A physical asset contains only one Media Profile. That is, an APID maps to only one Media Profile.
- An ALID is retrievable from an APID for the purpose of rights verification.

5.5.1.1 ALID

Syntax:

```
“urn:dece:alid:”<scheme>“:”<SSID>
```

The following restrictions apply to the <scheme> and <SSID> part of an ALID:

- An ALID scheme may not contain the colon character
- An ALID SSID may contain a single colon character. Other URN reserved characters must be escape encoded.
- An ALID scheme may be any unique name of an identifier standard, as long as it does not contain a colon, with the SSID being the unique asset identifier defined by that scheme. See Section 5.1.3.

5.5.1.2 APID

Syntax:

```
“urn:dece:apid:”<ALID scheme>“:”<APID SSID>
```

Each APID is associated with one or more ALIDs, although typically there will be a single ALID associated with a single APID. The APID-to-ALID mapping information described in [DCoord] section 6.2 allows associated ALID(s) to be retrieved for a given APID. An APID is constrained this as follows:

- Each APID is globally unique
- <ALID scheme> matches the <scheme> from the associated ALID

- <APID SSID> may contain a single colon character. Other URN reserved characters must be escape encoded.
- The scheme of the <APID SSID> is the same as <ALID scheme>, and the SSID is in accordance with Section 5.1.3.

For example:

ALID (org)	`urn:dece:alid:org:mycompany:abcdefg`
invalid APID (org)	`urn:dece:apid:org:mycompany:abcdefg:100` (extra colon: scheme = “org”, SSID = “mycompany:abcdefg:100” but can only contain one colon)
ALID (ISAN)	`urn:dece:alid:isan:000000018947000000000000`
APID (ISAN)	`urn:dece:apid:isan:000000018947000000000000:a203`
ALID (EIDR-S)	`urn:dece:alid:eidr-s:1E63-2E9A-11AB-FE88-1B89-M`
CID (EIDR-X)	`urn:dece:cid:eidr-x:C854-F52D-B0CF-1AE4-391A-7:EST`

5.5.2 ContentID

Syntax:

```
“urn:dece:cid:”<scheme>”:”<SSID>
```

A ContentID points to Coordinator-required metadata. Each ALID must have an associated ContentID. ContentIDs are not necessarily associated with an ALID. ContentIDs may refer to items such as shows or seasons, even if there is no single asset for that entity.

The <scheme> and <SSID> for ContentIDs are described in Section 5.1.3.

For example:

ContentID	`urn:dece:cid:eidr-s:1E63-2E9A-11AB-FE88-1B89-M`

5.5.3 Bundle Identifiers

Syntax:

```
“urn:dece:bid:”<scheme>”:”<SSID>
```

- <scheme> is “org:”<organization>
- <organization> is the Organization Name as defined in Section 5.2.1.

A Bundle defines and describes an arbitrary group of logical assets sold together. When posted with a Rights Token as part of the `SoldAs` element, the Bundle indicates the context of the sale, specifically the set of ALIDs sold in the Retail transaction. Bundles may be created by Content Providers or Retailers.

A Bundle's structure and APIs are defined in [DCoord] Section 6.3. Guidelines on structuring bundles can be found in [DPublisher] Section 7.5.

There are no standard identifiers for bundles: the scheme type of a bundle must be "org".

Example:

BID	urn:dece:bid:org:mycompany:1234abc567

5.5.4 Media Presentation ID

Syntax:

```
"urn:dece:presentationid:"<scheme>":"<SSID>
```

Each Media Presentation ID (PresentationID) identifies a Media Presentation. The concept of a Media Presentation is derived from [DASH] "collection of data that establishes a bounded or unbounded presentation of media content." A Media Presentation closely corresponds closely with a Common File Format DCC, so the construct of a Presentation ID is identical to an APID that PresentationIDs have a <type> of "presentationid" where APIDs have a <type> of "apid".

A PresentationID is constrained as follows:

- Each PresentationID is globally unique
- <scheme> should be EIDR-S or EIDR-X

5.5.5 DECE Media Package (DMP) ID

Syntax:

```
"urn:dece:dmpid:"<scheme>":"<SSID>
```

Each DECE Media Package (DMPID) identifies an Original DMP (DMP) and any instantiation of that ODMP including the defined components. A DMPID is constrained as follows:

- Each DMPID is globally unique

5.5.6 Media Application ID

Syntax:

```
“urn:dece:applicationid:”<scheme>”:”<SSID>
```

Each Media Application ID (PresentationID) identifies a Media Application. DECE does not currently support any Media Application technology. However, the DMP structure allows the inclusion of Media Applications. If included, they must be properly identified with an ApplicationID.

An ApplicationID is constrained as follows:

- Each ApplicationID is globally unique

5.6 Role Identifiers

The naming for DECE Roles is as follows:

```
“urn:dece:role:”<role>[“:customersupport”]
```

The <role> element corresponds to a DECE defined role as indicated in the table below:

Role	<role>	[:customersupport] allowed**
Content Provider	contentprovider	Yes
Coordinator	coordinator	Yes
Customer Support	customersupport	No
DECE Device	device	Yes
DECE Portal	portal	Yes
DRM Domain Manager	drmdomainmanager	No
DSP	dsp	Yes
Dynamic LASP*	lasp:dynamic	Yes
Linked LASP*	lasp:linked	Yes
Access Portal	accessportal	Yes
Retailer	retailer	Yes
User	user	No

Table 6 – Role Identifiers

*Note that there is only one Role for a LASP. A LASP can operate as a Dynamic LASP or a Linked LASP service in different modes as described in Section 4.4; the Coordinator treats these LASP services as sub-roles, and requires separate Nodes and Role Identifiers for the two services.

**The column labeled “[:customersupport] allowed” indicates whether the optional sub-role for customer support can be added to a Role identifier. For example: “urn:dece:role:retailer:customersupport” is legal, while “urn:dece:role:account:customersupport” is not. The Coordinator treats customer support as a separate Node with separate API permissions. See [DCoord] Section 1.8 for more information.

Example Role Identifier:

Retailer	urn:dece:role:retailer

6 Nodes and Communication

Now that we have defined the Roles in the Ecosystem, we must define how Roles securely communicate with the Coordinator and occasionally with each other. To enable this, the concept of a Node is introduced. A *Node* is a trust boundary that is assigned a unique, certified identity (e.g., a certificate) by a trust authority. This certified identity is used to mutually authenticate and secure the communication to other nodes in the Ecosystem.

A Node is identified by Fully Qualified Domain Name (FQDN) that is present in the associated Node certificate.

A Node can only be associated with one Role. If an Organization provides multiple Roles such as a combined Retailer and DSP, each of its Roles requires separate Nodes with unique certificates.

A Role can have multiple Nodes associated with it. Each Node fully represents the Organization in that Role. This is typically used when an Organization has distinct divisions or has contracted other companies to provide services on its behalf. Node operations can then be separately identified, and individual Node identities can be revoked if needed.

Note that Licensed Applications and Devices are not Nodes. Devices, specifically the Licensed Applications within them, communicate with the Coordinator through the Device Portal, which has a different security model. See sections 4.1.5 and 6.4.

The Coordinator Role is always asserted by a single Node run by the DECE organization.

6.1 Node Communication to the Coordinator

A single interaction between a DECE Node and the Coordinator Node consists of a synchronous messaging round-trip (one request and one response) between a requesting node and a responding node that exposes a DECE-defined web service interface. All messages pass through a secure communications layer designed to protect and deliver each message.

Nodes may also communicate with other Nodes, such as required by Security Token delegation and federation. See [DSecMech] for requirements on how the communication must be secured.

As shown in Figure 7, the application layer functionality provided by the node, together with the secure communication layer components, comprise a Node. Nodes in DECE rely on standard networking infrastructure for delivery of messages; the DECE layers simply add DECE specific trust and security properties.

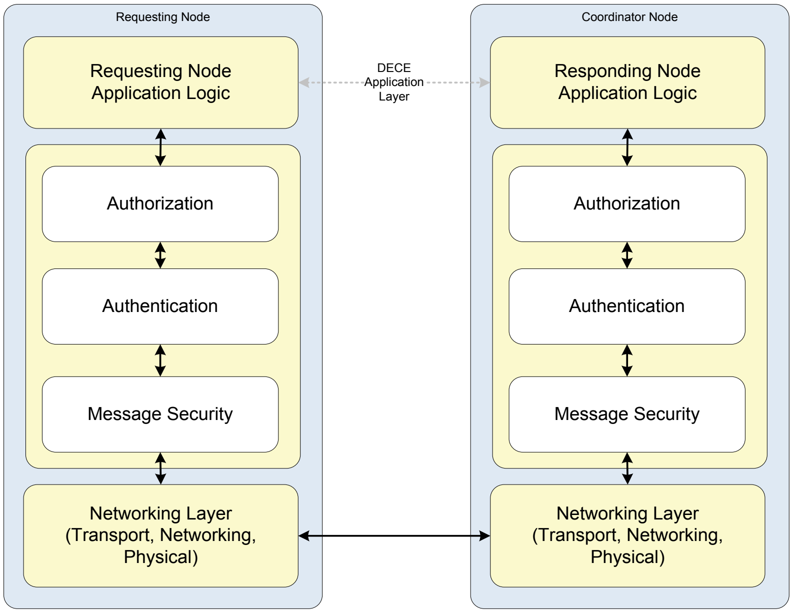

Requesting Node
Coordinator Node
Requesting Node Application Logic
DECE Application Layer
Responding Node Application Logic
Authorization
Authentication
Message Security
Authorization
Authentication
Message Security
Networking Layer (Transport, Networking, Physical)
Networking Layer (Transport, Networking, Physical)

Figure 7 – Node Messaging Diagram

6.2 Secure Communications Layer

This section describes the various components of the DECE defined secure communications layer and how they are used together to properly control access to DECE functions and data. Industry standard security technologies are defined to enable authentication, authorization and overall end to end message security.

6.2.1 Node Authentication

Node authentication is accomplished via the use of Internet profiled X.509 digital certificate [X509] that identify the domain name and organization of the Node. These TLS [TLS] certificates will be provided during Node licensing by the Coordinator Role.

Nodes authenticate to the Coordinator via mutual TLS authentication mechanisms. The Coordinator matches the certificate subject as a licensed Node. These certificates are provided to the coordinator prior to activating the Node to the coordinator. Nodes requiring Consumer interactions (e.g. Browsers) must use Extended Validation Certificates [EVCert].

An organization that operates multiple Roles must utilize unique certificates for each Node it operates.

See [DCoord] Section 2.3 for Node Authentication normative requirements.

6.2.2 Node Authorization

Node authorization is enabled by the Coordinator maintaining access permissions mapped to Roles. A Node is authorized to belong to exactly one given Role based on a license agreement with the DECE LLC.

The Coordinator checks the Node's Role against the allowed Roles for a given API call (see [DCoord] Appendix A).

See [DCoord] Section 2.3 for Node Authentication normative requirements.

6.3 User Authentication and Authorization

6.3.1 User Authentication

DECE Users (described in Section 7.2) are identified by a User Credential (a unique username and password pair managed by the Coordinator).

User passwords may only be changed by the User directly interacting with the Coordinator or an Account Management Node. The Coordinator does not require passwords to be changed periodically.

See [DCoord] Section 2.1 and [DSecMech] for normative requirements on User authentication and username and password restrictions.

6.3.2 User Authorization

Once properly authenticated DECE Users are authorized to access DECE data and services based on two authorization attributes:

1. Their access level as defined in Section 7.2.2; and
2. Their parental control settings as described in Section 7.2.6.

See [DCoord] Section 2.4 for more details on User authorization.

6.4 DECE Device Communication

Devices are an exception to the formal definition of a DECE Node, yet still interact with the Ecosystem similar to how a Node would. While a Node as defined in Section 6 is associated with a unique certified identity within the Ecosystem, Devices are not uniquely identified by DECE directly and do not have Node certificates.

Devices must open a secure TLS connection to the Device Portal, but instead of doing Node Authentication via a certificate, the Device authenticates the User as described in Section 6.3.

6.5 Security Token

There are many scenarios where a DECE Node, such as a Retailer or LASP, is interacting with the Coordinator on behalf of a User. In order to properly control access to user data while providing a simple yet secure experience for the User, authorization will be explicitly delegated by the User to the Node using a Delegation Security Token.

A *Security Token* is an XML object used for exchanging authentication and authorization data between an *identity provider* (such as the Coordinator) and a *service provider*[1] (the consumer of assertions such as a Retailer, DSP, or a LASP).

The Coordinator can support multiple Security Token profiles, but as of this specification the only Security Token profile in use is the Security Assertion Markup Language (SAML) version 2.0 [SAML], which is an XML-based framework developed by the Organization for the Advancement of Structured Information Standards (OASIS). It allows security information relating to a subject to be shared among service providers in a platform-independent way. SAML uses the public-key infrastructure (PKI) based model to establish trust, and supports WS-Security for securing web services messages.

Security Tokens are a central mechanism for authenticating and authorizing a User in the Ecosystem. Security Tokens:

- Provide a secure cross-vendor and platform-independent Single Sign-On (SSO). A User accessing the Ecosystem through a Retailer, DSP, LASP, or a Device need only use their personal credentials once to login, after which a Security Token is returned allowing the service provider to continue to operate on behalf of the User as long as the token remains valid. With User consent, a service provider can bind their account to the DECE Account via the Security Token. See Section 7.1.2 for details on Account binding.
- Improve privacy: User information such as the User ID and Account ID are mapped into per-Node unique identifiers. The actual values are never directly stored in a Security Token, so that different Nodes will use different identifiers to refer to the same entity. This mapping is

[1] Note that while a Device is not typically thought of as a service provider in the web sense of the word, it does provide services to the User. While an *agent* may be a more suitable word in the context of DECE, SAML uses the terms Identity Provider (IdP) and Service Provider (SP), and this document conforms to SAML terminology to help a reader understand the SAML specifications.

transparent to the service provider as all Coordinator APIs expecting user or account identifiers take the per-Node values.

- Allow delegation: A service provider such as a Retailer needs to conditionally allow other service providers, such as a DSP, to operate on the User's behalf. A Delegation Security Token allows constrained delegation, where specifically authorized Nodes can act in a limited but transparent fashion on behalf of the User. For example, a long-term Security Token resulting from binding a Retailer's account to a User's DECE Account (see Section 7.1.2) allows the Retailer Node or its DSP Nodes to use the Security Token to access Rights Tokens in the User's Rights Locker.
- Allow federation: A service provider can rely on the identity provided by the DECE ecosystem in addition to or in place of establishing its own identity for the user. Federation Security Tokens support single sign-on and authenticated handoff between Nodes. For example, LASPs can give Users the option to sign in to their LASP services using a single set of DECE Credentials instead of separate credentials specific to each LASP, or LASPS can provide an account management link that takes Users directly to their profile pages at a Web Portal without requiring them to sign in at the Web Portal.
- Have a specified validity period allowing for a Security Token to have a limited duration.
- Support revocation as either the service provider or the identity provider can terminate the Security Token. For example, a User can terminate a relationship with a lost Device without having to change their password or other User Credential.

6.5.1 Establishing a Security Context

Most of the Coordinator API calls require a Delegation Security Token to be passed in the HTTP headers in order to establish a security context for the call.

The Delegation Security Token can be obtained by a variety of mechanisms described in [DSecMech]. For example, a User can login via HTTP Basic Auth [HTTP Auth] to the Coordinator to establish the security context, and the Coordinator will return the Delegation Security Token in the HTTP Response.

A Device can use the `SecurityTokenExchange` API [DSecMech] to supply a User Credential and obtain a Delegation Security Token.

Once the Delegation Security Token is obtained, it is included in the HTTP header in subsequent calls to the Coordinator. A Security Token is long-lived or short-lived (session-based), and can be stored in a Device as long as it is treated as securely as a User Credential.

In order to reduce the need for frequent explicit User authentication, Users may bind their Retailer, LASP, or Access Portal accounts to their DECE Account, allowing the service to store a long-lived Delegation Security Token for access on the User's behalf as specified by User and Account consent policies without requiring the User to re-enter their User Credentials each time. See Section 7.1.2 for information on Account binding, Section 7.1.6 for Account consent policies, and Section 7.2.3 for User consent policies.

Similarly, Devices may store Delegation Security Tokens for access to the Rights Locker of an Account without requiring the User to login with their DECE User Credentials each time they use the Device. This is not the same as binding an Account, as the UserLinkConsent policy does not apply to Devices, but it achieves similar persistent access. See Section 7.3.3.

6.5.2 Using Security Tokens Across Multiple Nodes

While organizations supporting multiple Roles must use a separate Node per Role, a Security Token can be shared across Nodes to support a multiple Role login. For example a Retailer that is also a LASP can bind their Retailer account to a User's DECE Account, and then use the same long-lived Security Token enabled by the bind operation to authenticate the User from their LASP Node.

This is possible since Security Tokens may specify a set of Nodes, identified by NodeID, any which of which are authorized to use the same Security Token in Coordinator protocol messages. The SAML Token Profile defined in [DSecMech] uses the Audience element of the SAML Assertion to indicate what Nodes are authorized to use the Assertion. This allows an implementation which operates multiple Roles (and therefore multiple Nodes) within the Ecosystem to share the same Assertion.

6.5.3 User-level vs. Account-level Security Tokens

A Security Token always represents a particular User, and contains both the Account and User identifiers. Depending on the Coordinator API and the Role of the requesting Node, the Coordinator may interpret the Security Token at an "Account level" or a "User level" depending upon the context.

However, for simplicity, the Ecosystem specifications sometimes refer to an "Account-level" or "User-level" Security Token. This is a convention to mean that the Security Token is issued to a Node that will use the Security Token to access Coordinator APIs at the appropriate Account or User level.

6.6 Single Sign-on using Federation Security Tokens

In addition to employing Security Tokens for delegation, the Web Portal supports Federation Security Tokens specified in [DSecMech] that enable User single sign-on from eligible Nodes to the Web Portal.

Nodes which can establish local authentication sessions with a User may assert the Users identity towards the Web Portal.

Nodes that choose to assert a User's identity to the Web Portal must already possess a Delegation Security Token, and are required to make reference to that delegation token in their Federation Security Token. Other constraints and processing rules are provided in [DSecMech].

6.7 End-To-End Message Security

End-to-end message confidentiality and integrity functions are provided by the use of TLS [TLS].

Intra-Node communication is based on mutually authenticated TLS using Node certificates plus the addition of the Node's Role Assertion. The requesting Node asserts its identity and the responding Node verifies that (a) the identity is asserted by a mutually trusted naming authority, (b) that the roles asserted in the authorization layer were asserted about the Node identified, and (c) that the communication provably originates from the Node asserting its identity.

All communications between the DECE User and the DECE Portal role is protected by server-side TLS authentication and HTTP Basic Authentication of the user.

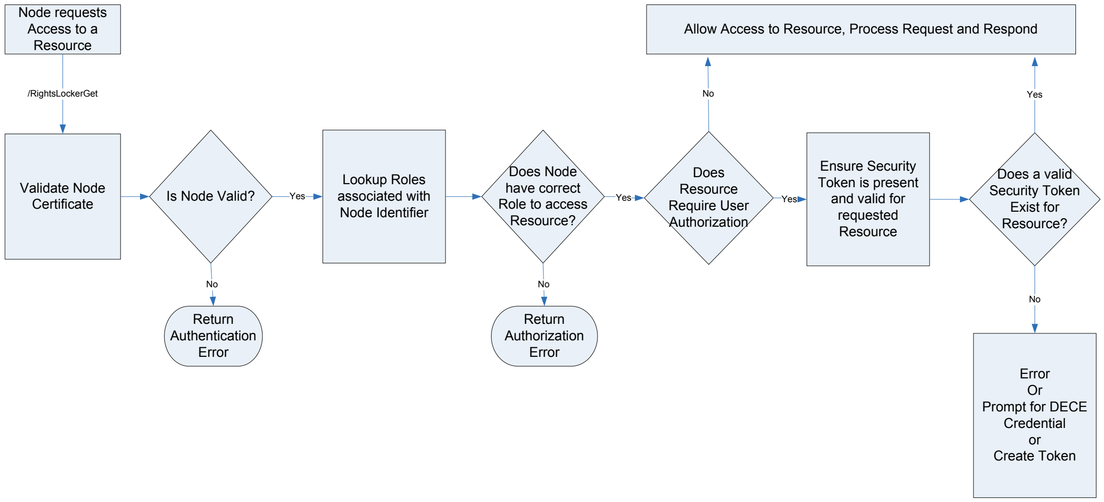

Node requests Access to a Resource
/RightsLockerGet
Validate Node Certificate
Is Node Valid?
Yes
No
Return Authentication Error
Lookup Roles associated with Node Identifier
Does Node have correct Role to access Resource?
Yes
No
Return Authorization Error
Does Resource Require User Authorization
No
Yes
Allow Access to Resource, Process Request and Respond
Ensure Security Token is present and valid for requested Resource
Does a valid Security Token Exist for Resource?
Yes
No
Error Or Prompt for DECE Credential or Create Token

Figure 8 – Authentication (AuthN) and Authorization (AuthZ) Flow

7 Account and Rights Management

7.1 The Account

The Account lies at the center of all DECE-defined entities. Each Account is associated with exactly one Domain, one Rights Locker, and a set of Users.

7.1.1 Account Creation

DECE Accounts can be created via a DECE Web Portal interface or interfaces maintained by Retailers, LASPs, or Access Portals using the `AccountCreate` API [DCoord] Section 13.1.1.

In the simple case, a user prepares to create an account by browsing to an account creation page provided by a Web Portal or a Node. The page will present a form requesting the first User's information such as Username, Password, Contact info, etc. (See Section 7.2 for details on Users.) When the form is posted, the Web Portal or other Node creates the Account with the `AccountCreate` and `UserCreate` Coordinator APIs [DCoord] Section 14.1.2.

The Coordinator creates a new Account, DECE Domain, and an empty Rights Locker. It also creates the first User in the account with Full-Access rights using the user information from the form. The Security Token for the created User is returned.

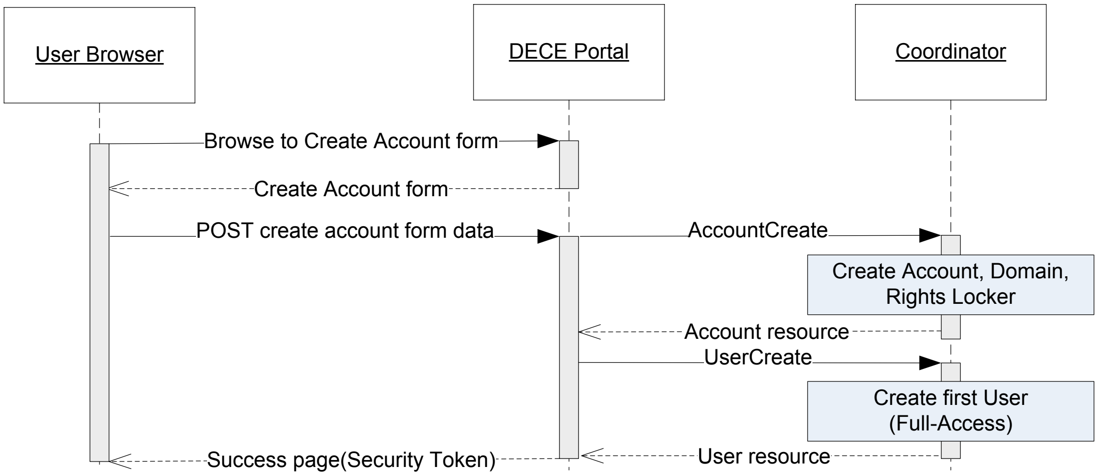

User Browser
DECE Portal
Coordinator
Browse to Create Account form
Create Account form
POST create account form data
AccountCreate
Create Account, Domain, Rights Locker
Account resource
UserCreate
Create first User (Full-Access)
Success page(Security Token)
User resource

Figure 9 – Account Creation

A Retailer or LASP can combine Account creation with Account binding as described below.

7.1.2 Account Access and Binding

Access to Account and User information in the Coordinator by a Node on behalf of a User is granted by the User logging in through an UltraViolet authentication interface and obtaining a Delegation Security Token. Access can also be provided to a Node as part of the process when the Node creates a new User. Access allows the Node to read and in some cases write information to the User and Account resources and associated resources, including the Rights Locker and Devices in the Account's Domain. This allows Nodes to display Content in a Rights Locker, Retailers to provide Content purchase, LASPs to Stream Content, Nodes to provide Account Management (i.e., changing user settings and policies such as username, password, email address, and Parental Control Settings) with proper User consent, and so on.

Account binding (often called account linking) is the process of granting a Node persistent access on behalf of a User without subsequent explicit Coordinator logins. (See the [DCoord] Section 12 on Node to Account Delegation for more information on Account binding.)

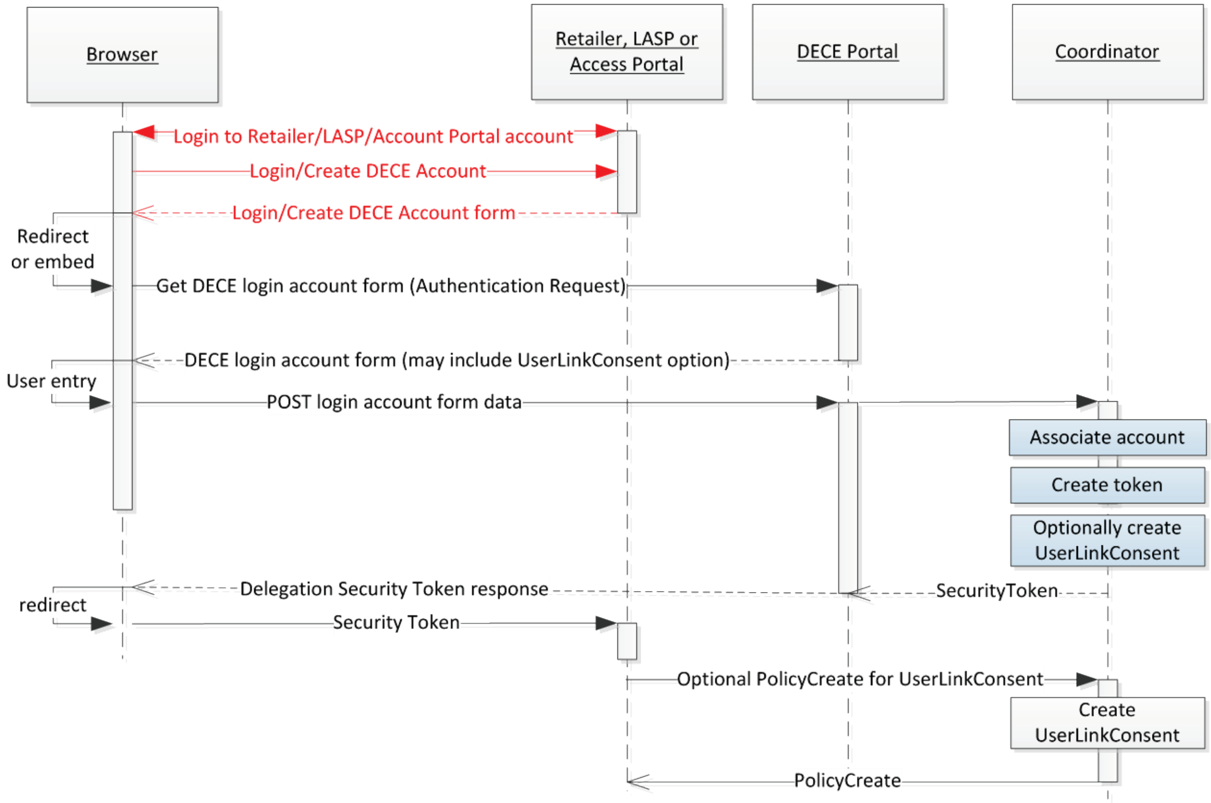
Note that Account binding is a convenience to the User and is not required prior to performing Coordinator functions. For example, a Retailer can allow a User to purchase Content without requiring a bound DECE Account. In this example the User's Browser would log the User into the Coordinator to obtain a short-term Security Token for the User's DECE Account.

The Coordinator keeps track of what Nodes an Account is bound to.

Security Tokens are described in Section 6.5.

7.1.2.1 General Account Binding Flow

The workflows for binding a Retailer, Linked LASP, a Dynamic LASP, and an Access Portal are the same. They differ in how the Coordinator records the binding, the type of Security Token that is returned, and its duration.


Browser
Retailer, LASP or Access Portal
DECE Portal
Coordinator
Login to Retailer/LASP/Account Portal account
Login/Create DECE Account
Login/Create DECE Account form
Redirect or embed
Get DECE login account form (Authentication Request)
DECE login account form (may include UserLinkConsent option)
User entry
POST login account form data
Associate account
Create token
Optionally create UserLinkConsent
Delegation Security Token response
SecurityToken
redirect
Security Token
Optional PolicyCreate for UserLinkConsent
Create UserLinkConsent
PolicyCreate

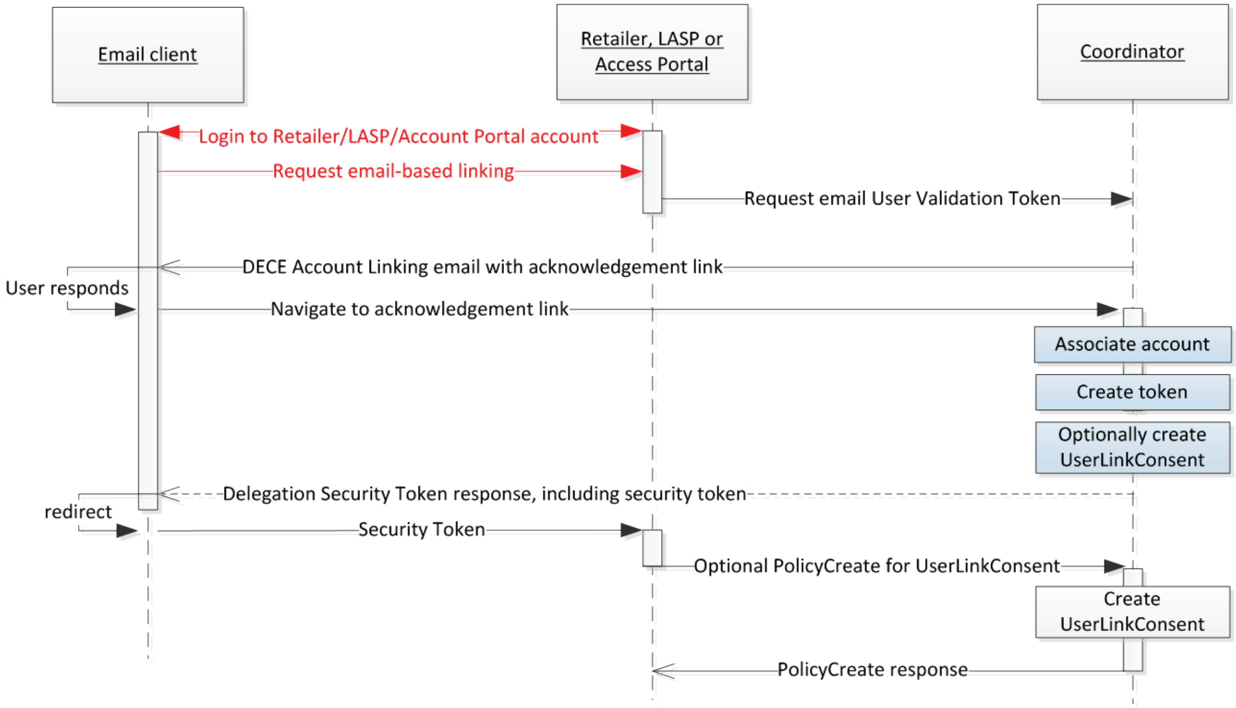

Email client
Retailer, LASP or Access Portal
Coordinator
Login to Retailer/LASP/Account Portal account
Request email-based linking
Request email User Validation Token
DECE Account Linking email with acknowledgement link
User responds
Navigate to acknowledgement link
Associate account
Create token
Optionally create UserLinkConsent
Delegation Security Token response, including security token
redirect
Security Token
Optional PolicyCreate for UserLinkConsent
Create UserLinkConsent
PolicyCreate response

Figure 10 – DECE Account Binding

First, the User must browse to a Retailer, LASP, or Access Portal and navigate to a page to login to their DECE Account. The Node's login page redirects to the DECE authentication page or embeds in an HTML "iframe" the DECE authentication page by sending an authentication request message to the Coordinator (see section 4.3 and the relevant Token Profile section(s) of [DSecMech]).

The DECE authentication interface allows the User to enter their User Credentials to log into their existing Account. The POST of the form data causes the Coordinator to return a Delegation Security Token via a redirect to the Node's page. The authentication UI includes an option such as "Link my account" for the User to consent to account binding. If the user agrees, the Coordinator binds the Account to the Node by setting a UserLinkConsent policy on the Account (see [DCoord] Section 5).

Alternatively, the Node may separately collect user consent for account linking, before or after invoking the DECE authentication interface. In the case where link consent is not collected directly by the Coordinator, the Node receives a short-lived Delegation Security Token, which provides access to the User's account but does not bind the Account to the Node. If the User agrees to the Node's request to link to the User's Account, the Node may then set a UserLinkConsent policy on the Account, using the short-lived Delegation Security Token, thus binding the Account to the Node. The Node should then call the Security Token Service (see [DSecMech] section 8) to receive a new, long-lived Delegation Security Token.

As an alternative to invoking the DECE authentication interface, a Node may use email-based linking. This allows the Node to fully control the user interface, which may be necessary in an environment that does not support HTML as required by the DECE authentication interface. In this case the Node uses the DelegationTokenRequest variation of the UserValidationTokenCreate API to cause an email message to be sent to the User, presenting the Node's request to link to the User's Account. If the user responds, the Coordinator sets the UserLinkConsent policy, binding the Account to the Node, and returns a long-lived Delegation Security Token to the Node. See [DCoord] Section 14.1.6 for details.

For new users without an existing UltraViolet Account, a Node may create a User (and an Account, if creating the first User) and bind to the Account of newly created User. The Node uses the Coordinator APIs to create the Account, if necessary, and then create a User in the Account. The Node calls the Security Token Service (see [DSecMech] section 8) to exchange the User Credentials provided during User create for a short-term Delegation Security Token. As part of the User creation process, the Node may request consent from the User to link accounts. If the User agrees, the Node may then set a UserLinkConsent policy on the Account, using the short-term Delegation Security Token, thus binding the Account to the Node. The Node should then call the Security Token Service (see [DSecMech] section 8) to receive a new, long-term Delegation Security Token.

The details of how the Coordinator does the binding and the characteristics of the Delegation Security Token differ depending on the Node's Role as a Retailer, Dynamic LASP, Linked LASP, or Access Portal.

A User may delete an Account binding at any time (see 7.1.3).

7.1.2.2 Retail Account Binding

A Retail account is bound to a Security Token at the User level.

No special User permission level is required to bind their Retail account to their DECE Account.

7.1.2.3 Dynamic LASP Account Binding

A Dynamic LASP account is bound to a Security Token at the User level. (Note that the Security Token always contains User and Account identifying information, but the Coordinator uses the User information for a Dynamic LASP Node while it uses the Account information for a Linked LASP Node.)

A Dynamic LASP may establish short-term access to a User's Account for session-based streaming, such as on a Web Browser in an Internet cafe, or it may bind to the Account at the User level for persistent access, such from a streaming app. See 4.4.2 for details of different LASP modes.

The Dynamic LASP MAY capture a User Credential for the purpose of User Creation. A LASP SHALL NOT store such User Credential.

Section 4.4.2 defines a Dynamic LASP including the normative requirements for re-authentication.

7.1.2.4 Linked LASP Account Binding

A Linked LASP account is bound to a Security Token at the Account level. (Note that the Security Token always contains User and Account identifying information, but the Coordinator uses the User information for a Dynamic LASP Node while it uses the Account information for a Linked LASP Node.)

Only an SAU or FAU can bind to a Linked LASP.

Section 4.4.3 defines a Linked LASP and includes normative requirements for Account binding limits.

7.1.3 Deleting Account Binding

Deleting an Account binding removes the association between the DECE Account and the bound Node in the Coordinator. An Account binding is removed by deleting the UserLinkConsent policy for the bound Node. The Coordinator revokes all associated Delegation Security Tokens (see [DSecMech] Section 4.3.2), and the Node may receive a disconnect or logout message from the Coordinator.

A Linked LASP SHALL ensure the User has Full-Access or Standard-Access Privileges on the Account to disassociate a Linked LASP account. See Section 7.2.2 for details on User Access Levels.

A LASP SHALL remove all Account-specific and User-specific identification information when deleting an Account binding including Security Tokens.

Upon disassociation of a Linked LASP Account from an Account, all active Linked LASP Sessions SHALL be terminated.

A User may revoke consent for Account binding at any time, at the Web Portal or an Account Management Node, which results in deletion of the UserLinkConsent policy for the Node and revocation of the Delegation Security Token issued to the Node.

In order to regain access to an Account after unbinding, the Node must repeat the authentication process described in 7.1.2.1.

7.1.4 Account Deletion

Accounts can be deleted via the DECE Web Portal interface or an interface provided by a Node with Account Management consent. See [DCoord] Section 13.1.3.

Deleting an Account sets the status of all the Account and related elements to “deleted”, effectively making the Account inaccessible. The Account is not physically deleted for a limited duration and retains the previously purchased Rights in the Rights Locker in case the account is later restored, such as by a Customer Support intervention. Subsequent calls to the Coordinator such as for purchases, Rights Locker gets, fulfillment, license acquisition etc. return an error. See [DCoord] Section 13.1.3 for details.

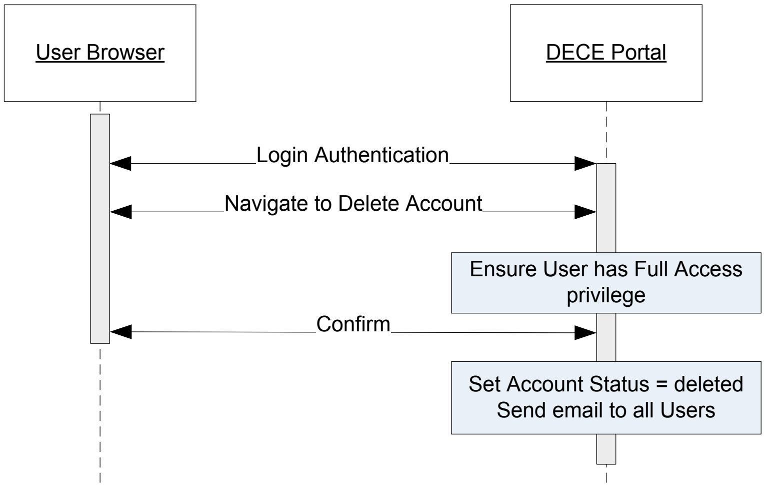

User Browser
DECE Portal
Login Authentication
Navigate to Delete Account
Ensure User has Full Access privilege
Confirm
Set Account Status = deleted
Send email to all Users

Figure 11 – Account Deletion

7.1.5 Account Limits

The Coordinator enforces certain limits:

- The ACCOUNT_USER_LIMIT parameter specifies the maximum number of Users in a DECE Account.
- The DOMAIN_DEVICE_LIMIT parameter specifies the maximum number of DECE Devices that can be joined to a DECE Account.
- The LASP_SESSION_LIMIT parameter specifies a small limit on the number of concurrent streams via a LASP.

The values of these parameters are determined by DECE policies, and are subject to change. There are other limits as well beyond the key ones highlighted above. The Appendix in Section 16 lists the current limits.

7.1.6 Account Consent Policies

[DCoord] Section 5.5.1 describes the complete list of Consent policies applicable to Accounts. Key Account-level consent policies (and the corresponding Coordinator policy name) include:

- **Purchase History** (`LockerViewAllConsent`). Permission for the identified Retailer or LASP to view all Rights Tokens in the Account's Rights Locker, with limited information from Rights Tokens from other Retailers.
- **Manage Account** (`ManageAccountConsent`). Permission for the identified Retailer or LASP to provide an interface to make changes at the Account level (add/delete Users, view/ rename/remove Devices, rename Account, etc., subject to User Access Control). ([DCoord] Section 5.5.1.2).
- **Allow Users to Consent to User Management** (`EnableManageUserConsent`). Permission for all Users in the Account to set their User Management policy (see Section 7.2.3).
- **Allow Users to Consent to User Marketing** (`EnableUserDataUsageConsent`). Permission for all Users in the Account to set their User Marketing policy (see Section 7.2.3).

Only Full-Access Users (Section 7.2.2) can change the above consent policies.

Consent policies and who is permitted to consent are subject to local law and the age of the User.

7.2 Users

An Account has a set of Users, enabling the Content to be shared between Users within the Account. The set of Users in an Account typically represents a family.

A User can only be associated with a single Account, and is identified by a unique Username.

7.2.1 User Data

Field Name	Description
UserID	Unique identifier generated by the Coordinator.
Username	User’s username, part of their Credentials for authentication.
Password	User’s password, part of their Credentials for authentication. Randomly generated by the Coordinator if not supplied.
GivenName	User’s first name or names.
Surname	User’s last name.
PrimaryEmail	The primary email account.
Country	Postal address Country.
DateOfBirth	The full date (year, month, day) of the User’s birth.

Table 12 – Required User data collected by the Coordinator (informative)

Table 12 shows the minimum required User data collected by the Coordinator for informative purposes. The full details are described in the `UserData-type` defined in [DCoord] Section 14.2.

Note that many regions have privacy laws governing the collection of personal information from users, especially children. A Retailer SHALL conform to all applicable privacy regulations for their region.

7.2.2 User Access Levels

A User is associated with a single User Access Level.

The Ecosystem defines the following three User Access Levels. These MAY be modified by applicable Geography-based Policies [DGeo]:

Nodes SHALL NOT set the Access Level of a User to BAU. (BAU is no longer supported, however the definition is retained here as informative detail.)

- Basic-Access User (BAU):
    - Can be any age.

  - Can obtain Content from a Retailer; that is, add a Rights Token to the Account's Rights Locker.
  - Can bind or unbind one or more of their Retailer accounts with the Account.
  - Can view the Account's Rights Locker and download Content.
  - Can consume the Discrete Media Right.
  - Can view the list of Devices joined to the Account.
  - Can view the list of Users in the Account.
  - Can view their Parental Control Information (see [DCoord] Section 5.5.5).
  - Can view their User Access Level.
  - Can set their User information: username, password, display name, e-mail address, and alternate e-mail address.
  - Can initiate a LASP Session from any of their Account bound Linked LASP accounts.

- Standard-Access User (SAU):
  - Inherits all Basic-Access User privileges.
  - Can create or remove a BAU or SAU in the Account.
  - Can join a Device to the Account.
  - Can perform an unverified removal of a Device in the Account.
  - Can bind or unbind the Account with a Linked LASP account.
  - Can bind or unbind one or more of their Dynamic LASP accounts with the Account.
  - Can initiate a LASP Session from any of their bound Dynamic LASP accounts.
- Full-Access User (FAU):
  - Inherits all Standard-Access User privileges.
  - Can delete the Account.
  - Can set the Account name.
  - Can add or remove an FAU, SAU, and BAU to the Account.
  - Can set the User Access Level for each User in their Account.
  - Can set the Parental Control Information for each User in the Account.
  - The initial User created when the Account is created is granted Full-Access User privileges.

Function	BAU	SAU	FAU
Delete Account			●
View the list of Users in their Account	●	●	●
Create FAU			●
Create SAU		●	●
Create BAU		●	●
Remove FAU			●
Remove SAU		●	●
Remove BAU		●	●
Remove themself		●	●
Promote / Demote Access Level for all Users			●
View/Set/Change Parental Control Information for all Users			●
View their own Parental Control Information	●	●	●
Join Device		●	●
Unverified Remove Device		●	●
View the Devices in the Account	●	●	●
Purchase Content (add Token to Rights Locker)	●	●	●
Bind/Unbind their Retail accounts	●	●	●
Unbind Retail accounts for all Users			●
Initiate an authenticated Dynamic LASP Session		●	●
Bind/Unbind their Dynamic LASP accounts		●	●
Unbind Dynamic LASP accounts for all Users			●
Bind/Unbind Linked LASP accounts with Account		●	●
View Rights Locker	●	●	●
Download Content	●	●	●
Consume Discrete Media Right	●	●	●

Table 13 – User Access Level Permissions

7.2.3 User Consent Policies

Consent policies and who is permitted to consent are subject to local law and the age of the User. See [DGeo].

[DCoord] Section 5.5.2 describes the complete list of Consent policies applicable to Users. Key User-level consent policies (and the corresponding Coordinator policy name) include:

- **Bind Account** (`UserLinkConsent`). BAU, SAU, or FAU. Permission for the identified Retailer or LASP to receive a long-lived Security Token to communicate with the Coordinator on the User or Account's behalf.

- **User Management** (ManageUserConsent). BAU, SAU, or FAU. Permission for the identified Account Management Node to update information about or delete the specified User. This can only be applied if the EnableManageUserConsent Account level policy had previously been set.
- **User Marketing** (UserDataUsageConsent). BAU, SAU or FAU. Permission for the identified Retailer or LASP to use information in the Coordinator about that User, such as e-mail, for marketing purposes.

7.2.4 Adding Users

Users can use the DECE Web Portal interface to add new Users to their Account. Only a User with Standard-Access or better (see Section 7.2.2) may add or remove Users from their Account.

Retailers and LASPs MAY use the UserCreate Coordinator API [DCoord] Section 14.1.2 to allow a User who has already bound their retail or LASP account to their DECE Account to add new Users to the Account.

7.2.5 Deleting Users

Users can be deleted via the DECE Web Portal interface or an interface provided by a Node with Account Management consent (see [DCoord] Section 14.1.5).

Deleting a User flags them as deleted, rather than completely removed for a limited duration to provide an audit trail and to allow Customer Support to correct improperly deleted Users.

A deleted User cannot log into the Account, and any previously issued User-level Security Tokens will be denied access.

The Coordinator will not allow the deletion of the last User of the Account. It will otherwise allow the invoking User to delete themself.

7.2.6 Parental Controls and Rating Enforcement

Parental controls are settings used to restrict access to Content and visibility of Content. *Ratings enforcement* is the application of parent control settings to Content ratings. The Coordinator associates DECE parental control attributes with Users for filtering Locker views based on Content ratings. DECE Devices may have their own parental control settings for ratings enforcement when Content is played on the Device. Retailers and LASPS may have their own parental control settings for controlling purchases, locker viewing, and streaming. Parental control systems and ratings enforcement methods by DECE Devices and by Retailers and LASPs are out of scope.

A User is also associated with parental control attributes for zero or more ratings systems. These attributes allow parents and/or guardians to control what Rights Tokens the User may or may not see. For example a User in the US with a parental control setting of “PG13” will only be able to see content with a rating of PG-13 or lower. Content with a rating above PG-13 will not be displayed. If a User has no parental control attributes or if there is no corresponding rating for the Content, then the Rights Tokens are not filtered.

Parental controls are applied by the Coordinator to filter Locker views in the Web Portal and to filter Rights Tokens passed through the Coordinator API. However, parental control filtering only applies at the User level. If Rights Tokens are requested by a Node with an Account-level security context, then the Node is responsible for any necessary ratings enforcement using its own system. If the Node has a User level Security Token it may retrieve a User’s parental control settings from the Coordinator for use in setting its own parental controls.

Rating systems are associated with regions. For example, the Motion Picture Association of America (MPAA) rating system is used in the US for movies, the TV Parental Guidelines rating system is used in the US for TV shows, and the British Board of Film Classification (BBFC) rating system is used for movies in the UK. DECE does not map between rating systems. If there is a parental control setting for one system but the Content is only rated in another system, this is equivalent to no parental control setting and no Content rating. DECE has three all-region parental control settings to handle these cases: one to indicate if unrated content is blocked, one to indicate if adult content is blocked, and one to indicate if explicit music lyrics are blocked. Likewise, the special adult rating applies to Content independent of rating systems. See section 5 of [DCoord].

Retailers should ensure that Users can’t view and purchase inappropriate content, but the Coordinator also checks when a Rights Token is added to an Account and will return an error if the Content rating exceeds the parental control setting, but only in the case where there is a matching region in both the User’s parental control settings and the ratings of the Content.

7.3 The Domain

In general, a digital rights domain is a group of devices belonging to a user or household that can share the same DRM licenses. The concept of a device domain is supported by the latest versions of most major DRMs. In a non-domain-based DRM scheme, licenses are bound to an identifier and cryptographic key previously provisioned in each device. As such, content protected by this license can only be accessed on a single device. If access is required on another device a new license must be issued, usually at an additional cost to the consumer.

In a domain-based DRM scheme, licenses are bound to a domain identifier represented by a cryptographic key. This domain key is shared between a set of devices owned by a consumer within the domain. This provisioning process is handled by DRM specific (e.g., native) domain manager interfaces and messages. Once the domain key is available on all devices of the same DRM, licenses can then be bound to the domain key, instead of the device directly, allowing for protected content to be accessed on all devices within the domain without the need reacquire a new license.

A DECE Domain expands the domain concept described above from a single DRM to multiple DRMs to allow interoperability between DRM systems. In this scenario we define a DECE Domain as a logical domain that is *authorized* by the Ecosystem and *enforced* through one or more native DRM domains.

7.3.1 Coordination of Domain Information

The Domain management function in DECE is managed by the Coordinator and per-DRM components called the DRM Domain Managers. The integration between a DRM Domain Manager and the Coordinator is a custom integration between the entities and is not specified by DECE.

Per-DRM License Managers are operated by DSPs. They need Account-specific Domain information to issue licenses for DECE Devices in that Account. The information is called *DRM Domain Credentials*, and is stored in the Coordinator for use by the DSP if needed.

The DRM Domain Credential is a binary object that is passed between the Domain Manager and the License Manager. This object is opaque to the Coordinator and DSPs and is passed through without inspection. The DRM Domain Credential is used to communicate information necessary for licensing from a domain manager to a License Manager. Some DRMs pass domain information without using the Coordinator.

At stated previously the coordination of domain information across Ecosystem entities enables the concept of the “interoperable domain.” This is accomplished by sharing the native DRM Domain Credentials for each Account from the Coordinator to the DSP’s.

An overview of the steps required to create a Domain through issuing a domain-based license are:

1. **DECE Domain creation**: The DECE Account is created, which also creates the DECE Domain. The Coordinator creates a DECE Domain ID as needed prior to licensing to be the global unique cross-DRM identifier for the unified domain, and a DRM Domain ID per native DRM domain. See Section 7.3.2.

2. **DRM Domain initialization**: The DECE Domain is associated with each Approved DRM native domain as needed prior to a DECE Device joining a domain or Content being licensed. The Coordinator binds the DECE Domain with a native domain by calling a native DRM Domain

Manager, passing it the DRM Domain ID, and receiving a DRM Domain Credential for the newly created native DRM domain. See Section 7.3.2.

3. **Device Joining**: Before Content can be played on a DECE Device, the DECE Device is added to the domain. This is done by doing a Device Join, which requests the Coordinator to add the DECE Device to the DECE Domain. The Coordinator interacts with the native DRM Domain Manager to add the DECE Device to the native DRM domain. See Section 7.3.3.
4. **Content Licensing**: When a DECE Device plays back purchased Content, the DECE Device must obtain a native DRM license from the DSP (the DSP could supply the license in the Container, or the license can be acquired from the DSP during playback). The DSP creates a native DRM domain-based license using the DRM Domain Credential associated with the User's Account by the DRM Domain Manager. See Section 12.

Once content has been licensed by a native DRM, the native DRM system manages the licensed playback. How licensing works when Content is moved or shared across DECE Devices is covered in Section 12.

7.3.2 Domain Creation

As the Coordinator has access to the domain management functionality for all Ecosystem-approved DRM's, it is responsible for the initial creation of all of the native DRM Domain Credentials. This initialization step may happen when a new DECE Account is created as described in Section 7.1.1 or it can be deferred by the Coordinator until a DECE Device joins the Domain or Content is licensed. The initialization of these credentials creates the DECE Domain associated with the Account which can then be communicated to the DSP's as necessary.

Each Approved DRM has a *DRM Domain Manager* module or service available to the Coordinator. These are collectively called *DRM Domain Managers* in Figure 2. The API between the Coordinator and the DRM Domain Manager differs based upon the needs of each Approved DRM, and is a custom integration between the Coordinator and the Approved DRM.

The DECE Domain is initialized by the Coordinator creating a unique DECE Domain ID to identify the Account-wide Domain, and a unique DRM Domain ID per Approved DRM. The DRM Domain ID is specific to the native DRM system or even potentially the DRM version and is distinct from the DECE Domain ID. The former is for identifying a domain during DRM Join operations, while the latter can be used for global identification of the virtual DECE Domain.

Prior to licensing or a Device Join, the Coordinator calls a DRM Domain Manager native API to create the native domain, passing in the DRM Domain ID, and receiving the native DRM Domain Credential. In some cases, this is a cryptographic key representing the DRM’s native domain, but its contents are opaque to the Coordinator and it can be any DRM-specific binary object.

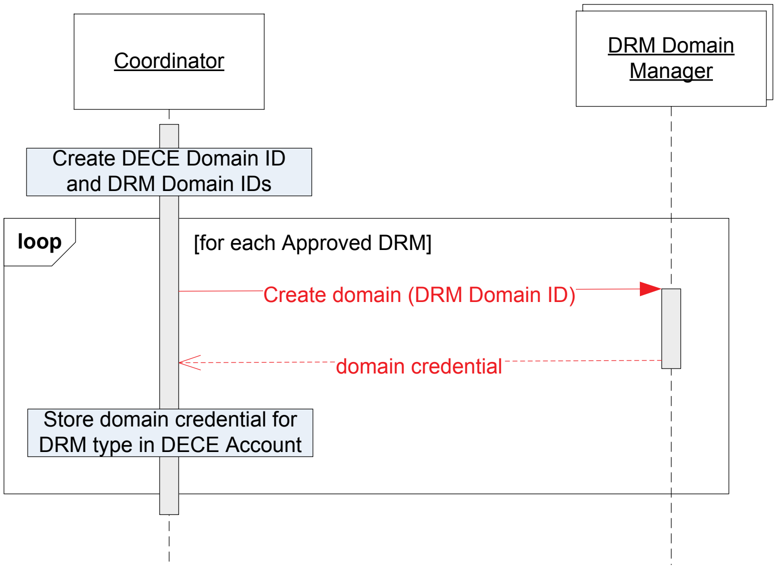

Coordinator
DRM Domain Manager
Create DECE Domain ID and DRM Domain IDs
loop
[for each Approved DRM]
Create domain (DRM Domain ID)
domain credential
Store domain credential for DRM type in DECE Account

Figure 14 – DECE Domain Creation

Note that the Coordinator stores the DRM Domain Credential associated with the DRM ID. The DRM ID includes the DRM version (see Section 5.4.1) so the DRM Domain Credential is per-DRM version. This is desirable as a DRM Domain Credential may change as DRM systems are updated.

7.3.3 Device Join

Adding a DECE Device to a group of devices in an Account that can share DRM licenses (a digital rights domain) is called a *Device Join*. Outside of streaming content from a LASP, a DECE Device must join the DECE Domain in order to play purchased Content.

The Coordinator enforces that a DECE Domain has a maximum of DOMAIN_DEVICE_LIMIT Devices joined at any time (see Section 16).

The DRM Clients in a DECE Device can only be joined to one DECE Domain. (Note that a physical device can be treated as multiple DECE Devices; this is necessary for devices supporting multiple DRMs in situations where the Coordinator cannot definitively determine that multiple DRM Clients are on the same physical device.) However, the DRM Client on the DECE Device may be bound to other native DRM domains. This means that joining a DECE Domain will not impact any preexisting non-DECE content already licensed to the Device.

A Device Join has two primary functions:

1. To bind the DECE Device to the User's Account, thus tying the Device to their DECE Domain and eliminating the constant need to log into their Account in order to use the Ecosystem.
2. To join the DRM Client on the Device into its native DRM domain. This enables Approved DRMs to share their native licenses among devices in a household.

In order to initiate a join, a *Join Trigger* must be obtained from the native DRM Domain Manager by the DECE Device. The Join Trigger is an opaque binary object containing DRM specific information used by the DRM Client to connect to its DRM Domain Manager and join the devices. There are a variety of ways to initiate a join to obtain the Join Trigger, but once the Device has the Join Trigger the actual join process is the same.

7.3.3.1 Initiating a Device Join

In order to initiate a join, a User logs into their DECE Account so that the Ecosystem can bind the Device to the Account and obtain the Join Trigger the DRM Client needs to perform its native DRM join.

The Coordinator ensures the User is authorized to join a Device and has Standard-Access authorization or greater.

As some Devices are not network connected, or do not have a full keyboard, there are a number of ways to log in and initiate a join:

- Device Standalone Join: A DECE Device with the ability to easily enter usernames and passwords and with Internet access can directly use the Device Portal APIs permitting the User to use an interface on the Device to directly log into their Account and start the join. Tethered DECE Devices can also use this method from an application on the Tethered Host.
- Web Portal Initiated Join: The User can use a Browser to access the DECE Portal web site to obtain a simple numeric code to be entered on the DECE Device to initiate the join. This is useful for DECE Devices with limited data entry, such as without a convenient full keyboard.
- Proxy Join: Allows DECE Devices that communicate to a Device Proxy to have the Proxy operate on the Device's behalf to initiate the join.

7.3.3.1.1Device Standalone Join

In a Standalone Join, the Licensed Application on the DECE Device initializes the join by using Device Portal APIs to the Coordinator (Device Portal).

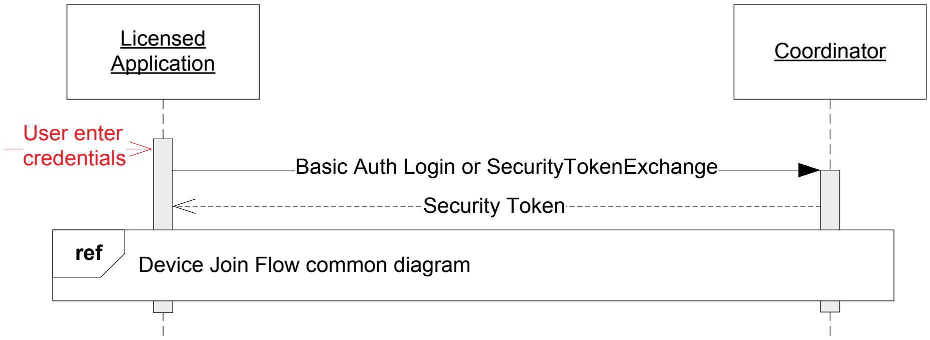

Licensed Application
Coordinator
User enter credentials
Basic Auth Login or SecurityTokenExchange
Security Token
ref
Device Join Flow common diagram

Figure 15 – Device Standalone Join Initiation

Initializing the join is straight forward: the User enters their credentials on the DECE Device, which then authenticates the User with the Device Portal by using an HTTP Basic Auth login [HTTP Auth] or via the `SecurityTokenExchange` API [DSecMech].

Once the communication with the Device Portal has been established, the DECE Device uses the standard Device Join Flow described below in Section 7.3.3.2.

7.3.3.1.2Node-Initiated Join

A Node-initiated Device Join simplifies joining a DECE Device with limited keyboard capabilities by allowing a User to use a Browser on another device (such as a general purpose computer) to log into the DECE Web Portal or any other Node with providing this feature as part of Account Management functionality in order to get a small numeric *Domain Join Code* which can then be entered into the Licensed Application on the DECE Device.

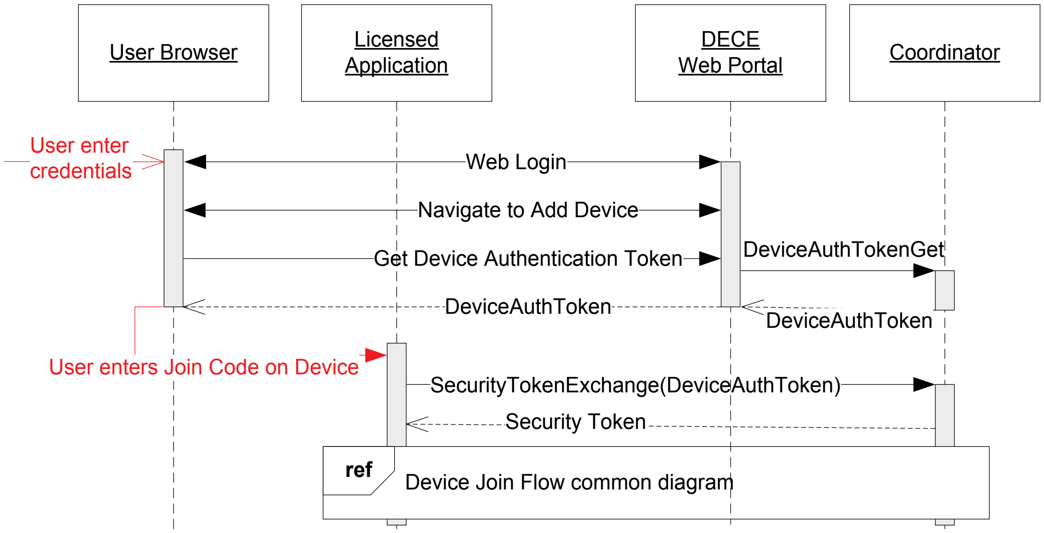


Figure 16 – Node Join Initiation

The User logs into the DECE Web Portal via a Browser on another device with a full keyboard, such as a general purpose computer, or connects to a Node-provided user interface. The User requests a numeric Domain Join Code. The User notes the code, and switches to the DECE Device.

On the DECE Device, they enter the numeric Domain Join Code into the Licensed Application. This allows the User to be logged into the Coordinator via the `SecurityTokenExchange` API [DSecMech] to allow the standard Device Join Flow described in Section 7.3.3.2 to complete. How a DECE Device uses a Domain Join Code is describe in [DDevice] Section 4.1.1.2.

The Domain Join Code is valid for a limited duration. If the code expires before the User succeeds in joining a Device, they can log into the DECE Web Portal or an appropriate Node and obtain a new Domain Join Code.

7.3.3.1.3 Point of Sale (POS) Join and Assisted Join

Point of Sale Join (POS Join) allows a Node to support Device Join without the User authenticating at the Device or entering additional data such a Join Code. POS Join is designed to help Devices be easily Joined to an Account after purchase, potentially at the store or via the retail website before the User even starts to use the Device.

POS Join can also be used for Assisted Join, where a Retailer, LASP, or Access Portal is also a Client Implementer or has a relationship with a Client Implementer and is able to manage Device Join on behalf of a User. This is useful in situations where the User is already authenticated at the Node, since the User can simply agree to Join an associated Device without needing to re-enter their User Credential.

POS Join requires that the Node have a unique code (DeviceUniqueString) that the Device also has. The details of how this is communicated between the Node and Device are out of scope of DECE.

The Node must also have a Security Token for a User in the Account to which the Device is to be Joined. This may be obtained at any time before the POS Join.

The Node posts the DeviceUniqueString to the Coordinator on behalf of the User, initiating the POS Join process. Subsequently the Licensed Application passes the same DeviceUniqueString in a Join Trigger request so that the Coordinator can match the Licensed Application to the appropriate Account and complete the Device Join without additional Account information.

Details are provided in [DDevice] Section 4.1.1.4 and [DCoord] Section 9.1.6.

Nodes SHOULD obtain consent from a User before performing a POS Join.

7.3.3.1.4Device Proxy Initiated Join

A Device Proxy (introduced in Section4.7.2) allows Client Implementers to operate a proxy for the DECE Device to facilitate the Device Join operation.

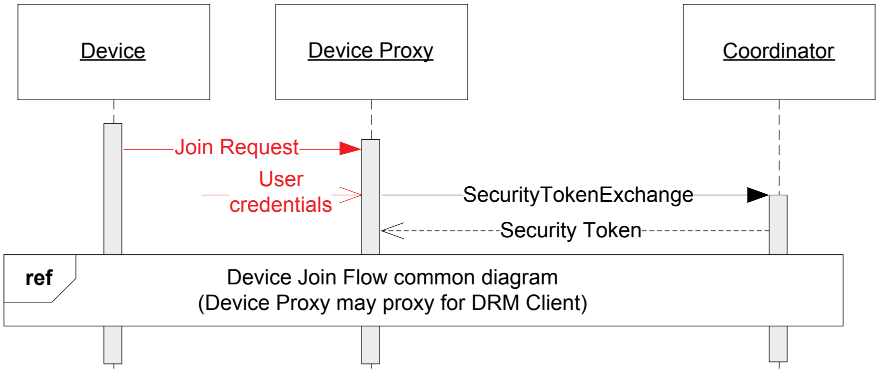


Figure 17 – Device Proxy Join Initiation

A Device Proxy MAY temporarily store User Credentials for authentication with the Coordinator in accordance with [DSecMech]. This is to enable a Device Proxy to allow a User to enter their User Credentials so that the Device Proxy can authenticate with the Coordinator on behalf of a Device.

The Device Proxy can proxy for the Proxied Device to initiate a Device Join. It can obtain the User Credentials from the User, the Device, or use previously stored credentials.

Once the Device Proxy is authenticated and has received a Security Token from the Coordinator Login, it can operate on behalf of the User and Proxied Device during the common Device Join Flow described in

Section 7.3.3.2. In Figure 18, functions performed by the DECE Device or the DRM Client may be partially implemented by a Device Proxy service (not shown).

The Device Proxy MAY proxy for the DRM Client. This assumes that the implementation is consistent with the Approved DRM license and does not violate the compliance and robustness rules of the Approved DRM.

If the Device Proxy proxies for the DRM Client, it may use proprietary protocols allowing it to provide some or all of the functions of the DRM Client in Figure 18.

A Device Proxy MAY do device attestation on behalf of a Proxied Device.

If a Device Proxy does device attestation (see Section 7.3.3.3) on behalf of a Proxied Device, the Device Proxy SHALL ensure the DECE Device conforms to DECE requirements.

7.3.3.2 Device Join Flow

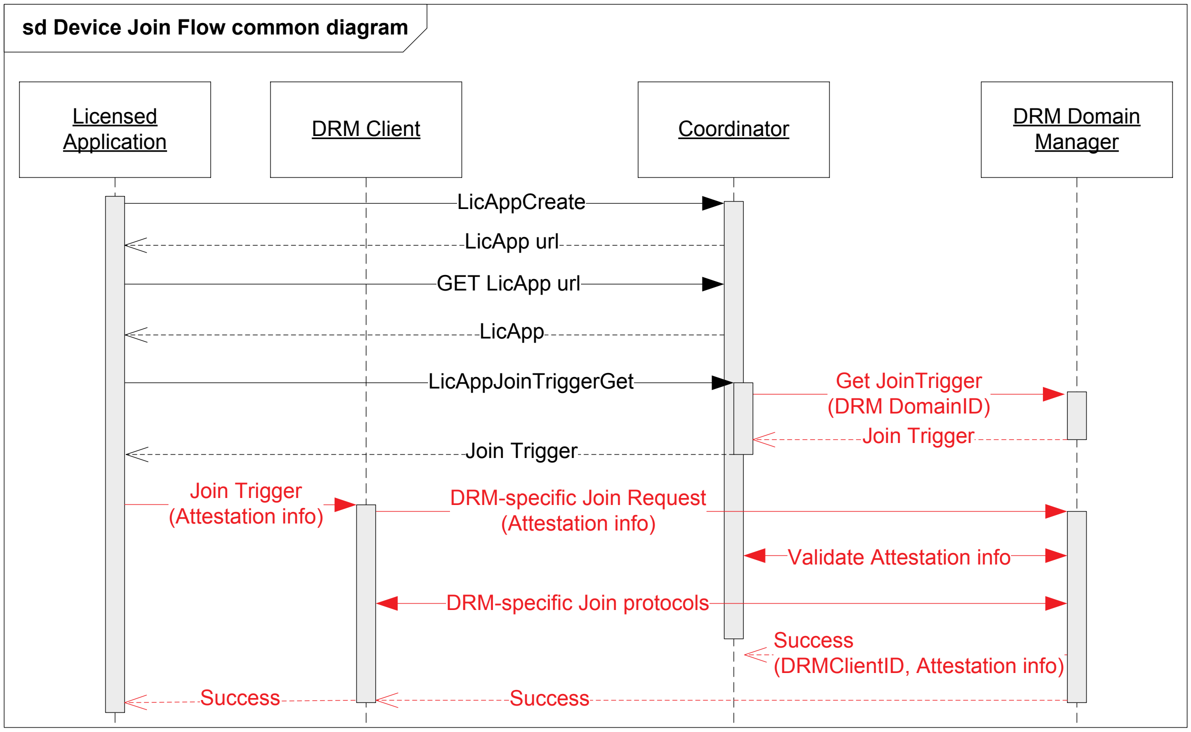

sd Device Join Flow common diagram
Licensed Application
DRM Client
Coordinator
DRM Domain Manager
LicAppCreate
LicApp url
GET LicApp url
LicApp
LicAppJoinTriggerGet
Get JoinTrigger (DRM DomainID)
Join Trigger
Join Trigger
Join Trigger (Attestation info)
DRM-specific Join Request (Attestation info)
Validate Attestation info
DRM-specific Join protocols
Success (DRMClientID, Attestation info)
Success
Success

Figure 18 – Device Join Flow

1. The Licensed Application authenticates the User to the Coordinator via a variety of methods as described in the preceding sections, and calls `LicAppCreate` to create a Licensed Application (`LicApp`) resource in the Coordinator. On success, `LicAppCreate` returns a HTTP 201

response with the URL of a LicApp resource populated with the LicAppID, DeviceID, DomainID and other data used for subsequent calls. The Device should then retrieve the LicApp resource via a GET.

2. The Licensed Application does a LicAppJoinTriggerGet query to get a Join Trigger to initiate the DRM domain join. The Coordinator checks the Account limits for the number of devices (DOMAIN_DEVICE_LIMIT) and to ensure the Device has not been joined too often (DEVICE_DOMAIN_FLIPPING_LIMIT). The limits are listed in Section 16.

3. The Coordinator calls the native DRM Domain Manager to request the Join Trigger, identifying the Domain being joined with the DRM Domain ID created during Account Creation. This API is specific to the DRM Domain Manager and is out of the scope of the DECE.

4. The Domain Manager returns the DRM-specific Join Trigger binary object to the Coordinator (a DRMClientTrigger). The Coordinator returns the Join Trigger to the Device in response to the LicAppJoinTriggerGet.

5. The Device calls the DRM Client's proprietary Device Join interface, passing in the Join Trigger and attestation data. This API is out of the scope of the DECE. (Attestation is described in Section 7.3.3.3.)

6. The DRM Client and the DRM Domain Manager use their own proprietary protocols out of the scope of the DECE to do the native DRM Device Join and convey the attestation data to the Coordinator for validation.

7. When the join is successful, the DRM Domain Manager returns the DRMClientID to the Coordinator to be associated with the Account.

8. Once joined, a Device may store the Account-level Security Token returned by the Coordinator from the login in step (1) to reduce the need for subsequent use of User Credentials.

7.3.3.3 Device Attestation

Attestation is an assertion of compliance between the Client Implementer and the DRM.

DECE Devices have the means to identify themselves to the Ecosystem for the following purposes:

- Prevent Non-Compliant Devices from joining to keep consumers from mistakenly adding a non-compliant Device with a compliant DRM
- Ensure only licensed Client Implementers can function in the Ecosystem

- Ensure only compliant and logoed devices can function in the Ecosystem

DECE provides each Client Implementer with a Client Implementer unique ID. The Client Implementer provides DECE with model and, where necessary, Licensed Application identification strings. As part of Device Join, the DECE Device passes the Client Implementer, model and application identification through the DRM to the Coordinator who verifies that these strings correspond with an approved product. Products not on the list are not allowed to Join. Use of a Client Implementer, model and application requires a Device Role license, and is an assertion of compliance between a Client Implementer and DECE.

DRM Clients also verify through DRM-specific mechanisms Licensed Applications to disallow other applications from playing protected DECE Content.

7.3.3.4 Multiple DRM Clients or Licensed Applications on a physical device

Where multiple DRM Clients and Licensed Applications in the same Domain exist on a single physical device, it is desirable that they be counted as a single DECE Device with respect to Account Device limits. When the Coordinator can consolidate DRM Clients and Licensed Applications, it does.

Licensed Applications are prohibited from attempting to Join multiple Domains. DRM Clients are required to make use of available methods to avoid Joining multiple Accounts supported on a single physical device.

7.3.4 Device Leave

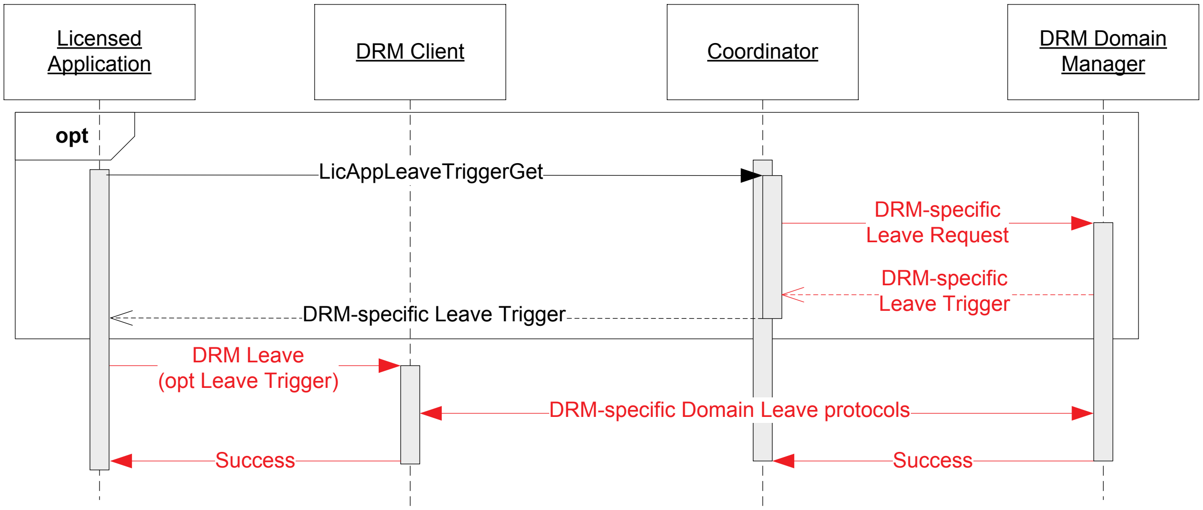

Licensed Application
DRM Client
Coordinator
DRM Domain Manager
opt
LicAppLeaveTriggerGet
DRM-specific Leave Request
DRM-specific Leave Trigger
DRM-specific Leave Trigger
DRM Leave (opt Leave Trigger)
DRM-specific Domain Leave protocols
Success
Success

Figure 19 – Device Leave

Any User may initiate a Device Leave; no special User Access Level is required.

A Device may use the LicAppLeaveTriggerGet API (see [DCoord] Section 9.2) if the DRM Client needs information to locate the DRM Domain Manager or to obtain Domain identifiers if they were not stored as part of the Join. Or the DRM Client domain leave APIs can be called directly if no Leave Trigger is required.

Performing a Device Leave causes the Coordinator to delete the Licensed Application related resources created during the Device Join operation (see Section 7.3.3.2), and decrements the count of Devices joined to the Account for the purpose of enforcing the DOMAIN_DEVICE_LIMIT.

When a DECE Device leaves a domain, the Device (and Licensed Application) must delete all Account-specific and User-specific identification information including Security Tokens.

After a Device is removed from the Account (the DECE Domain), any future attempts to play or license Content using a Right in the DECE Account Rights Locker will fail until the Device rejoins the DECE Domain. The Coordinator enforces the DEVICE_DOMAIN_FLIPPING_LIMIT when a Device is rejoined as described in Section 7.3.3.2. Note that simply leaving an Account and rejoining the same Account without an intervening join to a different Account does not count against the DEVICE_DOMAIN_FLIPPING_LIMIT so that accidental leaves are not penalized.

Since any cached DRM licenses on the DECE Device are inherently tied to the native DRM domain, when a DECE Device leaves a domain the DRM Client ensures that any cached licenses can no longer be used to play Content.

7.3.4.1 Device Proxy Initiated Leave

A Device Proxy (introduced in Section 4.7.2) allows Client Implementers to operate a proxy for the DECE Device to facilitate the Device Leave operation. The Device Proxy can proxy for the DECE Device to initiate a Device Leave.

The Device Proxy acts on behalf of the Device by authenticating with the Coordinator (using the Security Token returned from the Device Join operation, see Section 7.3.3.1.3) and initiating the Device Leave operation by calling the LicAppLeaveTriggerGet query to obtain a DRM-specific Leave Trigger.

The interface between the DECE Device and the Device Proxy is not specified by DECE, but the Device Proxy SHALL obtain the DRM-specific Leave Trigger from the Coordinator and pass it to the DRM Client to cause the Device to leave the DRM Domain.

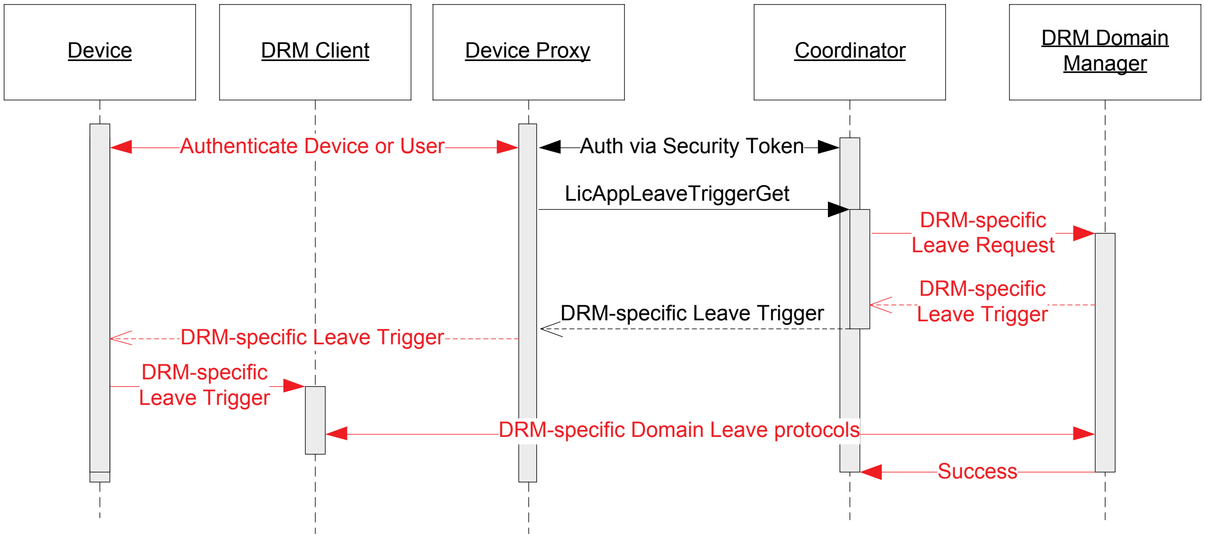

Device
DRM Client
Device Proxy
Coordinator
DRM Domain Manager
Authenticate Device or User
Auth via Security Token
LicAppLeaveTriggerGet
DRM-specific Leave Request
DRM-specific Leave Trigger
DRM-specific Leave Trigger
DRM-specific Leave Trigger
DRM-specific Leave Trigger
DRM-specific Domain Leave protocols
Success

Figure 20 – Device Proxy Initiated Device Leave (illustrative)

The Device Proxy MAY proxy for the DRM Client. This assumes that the implementation is consistent with the Approved DRM license and does not violate the compliance and robustness rules of the Approved DRM.

If the Device Proxy proxies for the DRM Client, it may use proprietary protocols allowing it to provide some or all of the functions of the DRM Client in Figure 20.

7.3.4.2 Unverified Device Leave

It is not always possible to communicate with a Device and have the Device initiate a Device Leave. A Device may have been lost, reinitialized, or damaged. Users need to be able to remove a missing Device from their Account in order to prevent future licensing or fulfillment operations from occurring, and to decrease the number of Devices counted against the Account's DOMAIN_DEVICE_LIMIT. (See Section 16 for descriptions of Account limits.)

The DECE Portal (Web Portal) allows a Standard-Access or Full-Access User to remove a DECE Device from the Account's Domain without cooperation from the Device. The Coordinator allows this to happen infrequently by enforcing the UNVERIFIED_DEVICE_REPLACEMENT_LIMIT when a Device that was removed via an Unverified Device Leave is rejoined to the Account.

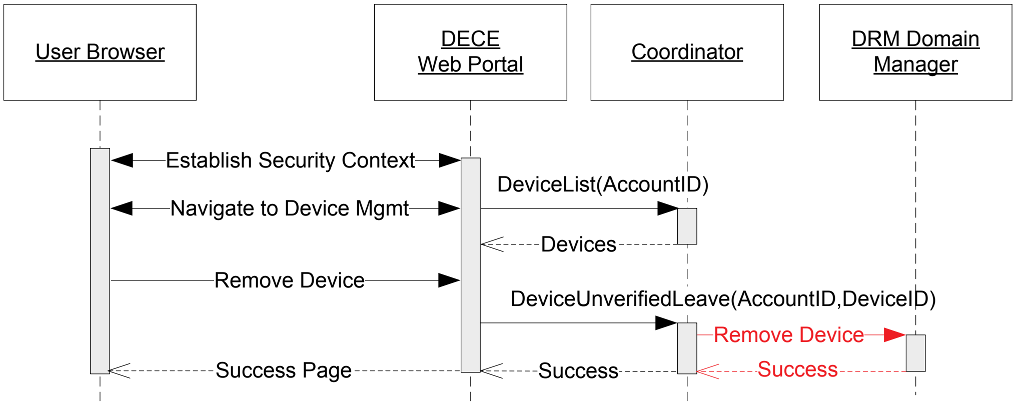


Figure 21 – Forced Device Leave

Since not all DRM systems can revoke a Domain Credential from a Device, especially if the Device is disconnected from any network, a Device which was forcibly removed from an Account may still be able to play Content using previously cached licenses. However, any future licensing action will fail. Whether a DRM system supports revocation of licenses is out of the scope of the DECE.

7.3.5 Device Move

From a DECE perspective, moving a DECE Device from one Account to another is a Device Leave followed by a Device Join using the workflows previously discussed.

Note that Content previously purchased on the original Account will no longer be playable on the Device once it has moved to another Account. Splitting an Account and moving Content from one Account to another is not currently supported by the Ecosystem.

7.4 The Rights Locker

This section describes the concept of the Rights Locker and Rights Tokens.

As previously described in Section 4.1.3, the Coordinator maintains the Rights Locker for a DECE Account. The Rights Locker stores all proofs of purchases in the form of Rights Tokens for content purchased by any User associated with the Account.

7.4.1 Rights Token Overview

A *Rights Token* is a DRM-independent representation of the rights associated with an instance of purchased Content. Other information about the User's rights to Content is managed by the Rights Token, including which Media Profiles were purchased, and whether the Content may be copied to

Discrete Media. Although Rights Tokens do not exist outside of the context of the Ecosystem, they are accessed, managed and manipulated via the web services interfaces exposed by the Coordinator role.

A Rights Token contains (among other information, see [DCoord] Section 7.2):

Element	Description
ALID	The Asset Logical ID for the asset.
ContentID	The Content ID for the metadata corresponding with the ALID.
Profile	A list of the Media Profiles included in the Right. Currently SD (Standard Definition) and HD (High Definition) are supported.
CanDownload	Per profile, whether the Containers can be downloaded
CanStream	Per profile, whether the content can be streamed
DiscreteMediaRightsRemaining	Per profile, whether the content can be exported to discrete media
SoldAs	Purchase information when multiple assets are purchased together. See 10.1.1.3.
PurchaseInfo	Retailer information about the purchase. See 10.1.1.4.
FulfillmentWebLoc	Locations of web pages or direct HTTP links to DCCs for downloading Content. See 11.1.2.
FulfillmentManifestLoc	Locations of manifest files for device downloading. See 11.1.3.
StreamWebLoc	Locations of web resources to Stream the Content. See 10.1.1.5.
LicenseAcqBaseLoc	Location used for calculating a licensing address. See 12.2.2.

Table 22 – Rights Token Elements

See [DDiscrete] for additional information in the Rights Token controlling Discrete Media exports.

7.4.2 Adding Rights

A Rights Token is added to the Rights Locker by a Retailer when a Right is purchased by a User. Section 10 describes the purchase process, and describes how a Retailer adds a Right Token to the Rights Locker for the DECE Account associated with the purchasing User.

7.4.3 Viewing the Rights Locker

All Users associated with the Account have access to the Rights Tokens in the Account’s Rights Locker including those that were purchases by other Users, subject to Ratings Enforcement by the Coordinator as described below.

The Coordinator provides a Web Portal user interface for a User to manage and view their Rights Locker.

The Coordinator also provides a web service programmatic interface for use by a Retailer, DSP, LASP, or DECE Device. The APIs for managing Rights Tokens and the Rights Locker are described in [DCoord] Section 7.

When an Account is bound, the User can consent to a Node (such as a Retailer, LASP, or Access Portal) having full view of the Rights in the Rights Locker. See the `LockerViewAllConsent` policy in [DCoord] Section 5.5.1.

If the `LockerViewAllConsent` policy is not true, the Coordinator will filter the Rights Locker view to exclude Rights Tokens issued by other Retailers. Once the `LockerViewAllConsent` policy is set to true, the Retailer will be able to see and display in their user interface Rights Tokens from any Retailer.

A Standard-Access or Full-Access User can also opt-in to a Node having Rights Locker data access such as for using Rights Locker data for purchase recommendations. See the `UserDataUsageConsent` policy in [DCoord] Section 5.5.1.

A Rights Locker view may be refined by the Coordinator to apply Parental Control filtering for Nodes with a User-level Security Token. Section 7.2.6 describes parental control and ratings enforcement, and is described in detail in [DCoord] Section 5.5.

7.4.4 Authorizing Access to Content and License Issuance

Prior to licensing access to Content, a DSP SHALL ensure that there exists a corresponding Rights Token in the Account's Rights Locker as described in Section 12.4.

Similarly, a LASP SHALL ensure a Rights Token allowing streaming exists prior to streaming Content as described in Section 13.2.

7.4.5 Rights Availability Windows and Recalling APIDs

Content Providers may occasionally need to specify time periods where fulfilling, licensing, streaming and using Discrete Media Rights to Content may be restricted. The time period for restricted access is referred to as a *Window* or a *holdback* in DECE documents. As these restriction Windows are for an entire Content as represented by an ALID, the Window is not expressed in the Rights Token but rather in a separate `LogicalAsset` resource (a *Logical to Digital Asset Mapping*) in the Coordinator (see [DCoord] Section 6.5).

The type of restrictions an ALID (for all or select Profiles) may be subject to include:

- APIDs for the ALID may be Recalled (revoked) or Replaced.
- Downloads (fulfillment) may be restricted for certain regions and time periods.
- Licensing may be similarly restricted.
- Streaming may be similarly restricted.
- Discrete Media Rights fulfillment may be similarly restricted.

Any such restrictions are stored in the `LogicalAsset`, which can be updated at any time by the Content Provider as described in [DPublisher].

The `LogicalAsset` applies to an ALID and Media Profile. The `LogicalAsset` stores the ALID, Media Profile, Content ID, Discrete Media Fulfillment methods, an `AssentStreamAllowed` flag indicating if streaming is enabled for a LASP without the need for a license agreement with a Content Provider (see Section 4.4.4), one or more `DigitalAssetGroup` elements within one or more `AssetFulfillmentGroup` elements, and optional `AssetRestriction` elements.

7.4.5.1 Recalled and Replaced APIDs

The `AssetFulfillmentGroup` is explained in [DCoord] Section 6.5.2. It contains a set of `DigitalAssetGroup` elements indicating the active APIDs for the ALID, and also listing the Replaced APIDs and Recalled APIDs. Before an APID can be used, this collection must be checked to determine if the APID is valid. See [DCoord] Section 6.5.2.4.

The `LogicalAsset` is checked prior to fulfillment to see if an APID has been recalled or replaced (Section 11.1.6), and the `LogicalAsset` may be checked for holdback information prior to licensing (Section 12.4.1). This is done by obtaining the `LogicalAsset` for the given `ALID` and Media Profile. The `DigitalAssetGroup` element indicates whether the `APID` in the `LogicalAsset` is in an `ActiveAPID` element (e.g. has not been replaced or recalled).

7.4.5.2 Asset Restriction Windows

Holdback information stored in the `LogicalAsset` is used in Locker views provided by the Web Portal Role and other Roles to indicate to Users that Content may not be available in a certain region and time period.

The `AssetRestriction` element in the `LogicalAsset` indicates constraints on Container Fulfillment, Container Licensing, Streaming, and Discrete Media Fulfillment for a time range in a region. See [DCoord] Section 6.5.2.6.

Note that the AssetWindow construct from earlier versions of the specifications is replaced by AssetRestriction.

Section 11.1.6 defines the conditions under which the DSP checks the `LogicalAsset` before Fulfilling a Container. Section 12.4.1 defines the conditions under which the DSP checks the `LogicalAsset` before Licensing a Container. Section 13.2 defines the conditions under which the LASP checks the `LogicalAsset` before Streaming Content.

7.4.6 Coordinating Rights

As the Ecosystem enables multiple retailers to sell content, the coordination of rights is another essential Ecosystem concept. Rights Tokens represent a purchase of content from any Retailer by a particular User associated with a specific Account. These rights are made available to any Users associated with the Account and can be downloaded and licensed on any device in the Accounts Domain.

8 Common File Format Container and DECE Media Package

8.1 Overview

DECE Content is encoded into the Common File Format (CFF) and is packaged in a Digital CFF Container (DCC, or simply referred to as a Container in this document). The Common File Format is designed to:

- Play across multiple devices
- Work with multiple DRMs
- Support progressive (segmented) download
- Contain information for licensing and purchasing
- Contain metadata describing the Content
- Hold DRM licenses in the Container for ease of transporting Containers within a Domain

The Common File Format and Digital CFF Container are described in detail in [DMedia].

DCCs can be packaged in DECE Media Packages (DMPs) defined in [DDMP].

The CFF supports the use of video elementary streams encoded in the AVC format (H.264) with some additional requirements and constraints. A wide range of audio coding technologies are supported, including several based on MPEG-4 AAC as well as Dolby® and DTS™ formats. Graphics and text-based subtitles are supported. The CFF also supports a common fragmentation structure enabling fast searching and trick modes as well as streaming. See [DMedia] for details on the video, audio and subtitle tracks encoding.

The CFF specifies a standard encryption scheme and key mapping that can be used with multiple Approved DRM systems. Standard encryption algorithms are specified for regular, opaque sample data, and for AVC video data with sub-sample level headers exposed to enable reformatting of video streams without decryption. See [DMedia] for details on track encryption and DRM support.

Protected DECE files contain a set of metadata, minimally including descriptive metadata (e.g., title), identifying metadata (e.g., DECE content identifier), parental control metadata (to be defined), license resolution metadata (License Manager URLs), and one or more pointers to more complete metadata resources.

DECE content may also be made available for a limited number of exports to Discrete Media (e.g. a DVD or secure memory device), and may also be consumed in streaming mode through authorized streaming services, referred to as LASPs (see Section 4.4).

For Discrete Media exports, the Coordinator keeps track of the number of exports to ensure that the maximum number of allowed exports is not exceeded. See [DDiscrete] and [DCoord] for more information about Discrete Media Rights.

8.2 Media Profiles

Audio-visual content in the Ecosystem are classified in a limited number of *Media Profiles*, very similar to MPEG profiles, where each Media Profile specifies a set of constraints on encoding formats, bitrates, number and type of audio-visual channels, aspect ratio, and more. Each Media Profile is targeted to a specific class of devices, trying to match the computational and rendering resources of the device class, while at the same time providing an optimal user experience. Currently two Media Profiles have been defined:

- a standard definition (SD) profile and
- a high definition (HD) profile.

Each profile defines both DCCs that contain combinations of video, audio and subtitle tracks; and DCCs that contain a single audio, video or subtitle track.

8.3 DECE Metadata

DECE Metadata is described in [DMeta]. How it is stored in the Container is described in [DMedia].

There are different types of Metadata stored in the Container:

- Physical Asset Information: consisting of the Asset Physical Identifier (APID) and Media Profile information, along with additional information used for licensing (Base Location) and to assist in locating a Retailer for Superdistributed Containers (Base PURL Location).
- Required Metadata: mandatory metadata describing the Content in the Container, including a ContentID and basic Movie and track information including Ratings, Images, title, run length, publisher, release year, etc.
- Optional Metadata: additional metadata that may be included to further describe the content.

8.3.1 Asset Physical Identifier (APID)

The Asset Physical Identifier (APID) defined in Section 5.5.1 is stored in the DCC in the Asset Information Box (‘`ainf`’) along with the Media Profile and Media Profile version. See [DMedia] Section 2.2.4.

The APID is stored in the Container when the Container is created by the Content Provider.

8.3.2 Base Location

The *Base Location* is information provided by the Retailer to locate the License Managers. The Base Location is an Internet domain name that is used to construct fully qualified domain names for licensing and downloading Content as described below.

The Base Location is stored in the Base Location Box (‘`bloc`’) in the DCC as defined in [DMedia] Section E.1.1 or in the BaseLocations part of a DMP as defined [DDMP], Section 4.2.5.

The Base Location SHALL be encoded as a string of bytes as defined by Internationalized Domain Names constraints [RFC5891], followed by null bytes (0x00) to a length of 256 bytes.

A Base Location is constructed as:

```
BaseLocation ::= [<retailersub>”.”]<retailerID> “.” <decedomain>
```

Where

- <`decedomain`> is the fully qualified domain name for the DECE licensing organization
- <`retailerID`> is a name assigned to the licensed retailer by the DECE
- <`retailersub`> are additional optional subdomain names a retailer can freely use at their discretion

For example: craigstore.decellc.org or mexico.craigstore.decellc.com

8.3.2.1 Reading the Base Location

The Base Location may not always be set, or it may be invalid. In this case, licensing and download URLs can be obtained from the Coordinator as described in Section 12.2.2.

8.3.2.2 Setting the Base Location

For a DCC not part of a DMP, the Retailer or DSP SHALL write the Base Location to the Container. How the DSP does this is outside the scope of the DECE. The DSP can do this when Content received from a Content Provider is added to their system, or it can be updated later during Content fulfillment.

For a DMP, the Retailer or DSP SHALL create a BaseLocations document for each Presentation.

For a DMP, the Retailer or DSP SHALL either include the Baselocations Parts in the downloaded DMP, or include these Parts as required downloads for the DMP (as indicated by forceDownload in the Manifest, as per Section 11. In other words, BaseLocations will always be included in the DMP after the initial download process has completed.

The Retailer SHALL ensure the DNS zone for the Base Location is set to resolve to the correct Retailer web server.

If a purchase changes the Base Location, such as by the User selecting a different Retailer, the DECE Device shall replace the existing Base Location with the new Base Location in the Container or DMP. This is necessary because the Base Location is used for licensing and an incorrect Base Location will cause unnecessary redirects as part of the licensing process. This requirement is defined in the DECE Device Specification [DDevice] Section 5.2.3.

8.3.3 Purchase URL (PURL)

A *Purchase URL* provides a location where a Right may be purchased via a web browser. There is no implicit guarantee that the Right can be purchased (e.g., the Retailer may have stopped selling that content), but there is a guarantee that if the Right is purchased, the Container with the Base Purl Location will be licensable under that Right.

The Container or DMP may optionally include a Base Purl Location that can be used to create a Purchase URL. The Base Purl Location is stored in the Base Location Box (`‘bloc’`) in the DCC as per [DMedia] Section 2.2.3 or in the BaseLocations Part within a DMP as per [DDMP] 4.3.5. This is primarily useful when Content is superdistributed or copied outside of a DECE Domain, requiring a Right to be purchased before the Content can be used.

Although not specified by DECE, a DECE Device may use other methods to locate a Retailer, including use of third party services, or having a pre-existing relationship with one or more DECE Retailers.

The Base Purl Location is optional. If it is not supplied the Retailer does not support constructing Purchase Locations. Otherwise the purchase internet domain is constructed by combining the `BasePurlLocation` with a hardcoded DECE internet domain, as in:

```
PurchaseUrl ::= "http://purchase." <basePurlLocation> "." <decedomain>
"/index.html?apid=" <APID>
```

Where

- <basePurlLocation> is the Retailer's Organization Name (see Section 5.2.1) stored in the BasePurlLocation element in the File Metadata box in the Container. If no Base Purl Location is defined, this field SHALL be filled with null bytes (0x00). If defined, the Base Purl Location SHALL be encoded as a string of bytes as defined by [RFC3986], followed by null bytes (0x00) to a length of 256 bytes.
- <decedomain> is the fully qualified domain name for the DECE licensing organization.
- <APID> is the APID from the Container. See Section 8.3.1.

For example:
http://purchase.xyzstore.decellc.com/index.html?apid=urn:dece:apid:ISAN:1209123091029:a203

8.3.3.1 Reading the Base Purl Location

If a Device attempts to license a DCC and determines that the Domain does not have the Right to play the Content, the Device MAY use the BasePurlLocation element to construct a Purchase URL and direct the user to the Retailer website to potentially acquire the Right. See [DDevice] Section 5.2.1.

8.3.3.2 Setting the Base Purl Location

The Retailer (or Content Provider or DSP on behalf of the Retailer) MAY write the Base Purl Location to the Container. How this is done is outside the scope of the DECE.

If the Retailer writes the Base Purl Location, the Retailer SHALL use its Organization Name (Section 5.2.1) as the value of the BasePurlLocation element.

8.3.4 License Acquisition Location

The *License Acquisition Location* (LALOC) is a fully-qualified domain name (FQDN) for the License Manager responsible for licensing the Content for a particular DSP and DRM. It is derived from the Base Location stored in the DCC or from the Rights Token LicenseAcqBaseLoc, and is not directly stored itself.

Assuming a Base Location, the License Acquisition Location (LALOC) is constructed as follows:

```
LALOC ::=  <DRM name> "-license." <BaseLocation>
```

Where

- <DRM name> is from Table 29 – Approved DRM List in Section 17.
- <BaseLocation> is the Base Location from the Container (Section 8.3.2) or from the LicenseAcqBaseLoc element in the Rights Token (Section 12.2.2).

For example: playready-license.xyzstore.decellc.com

9 Content Publishing

The figure below provides an overview of the Ecosystem publishing flow. Many parts of this flow are out-of-scope for DECE, but are included to provide a relatively complete view of information flow and linkages within the Ecosystem. The accompanying text provides a narrative description of the key activities within the publishing flow, offering context for the publishing requirements enumerated in the next section.

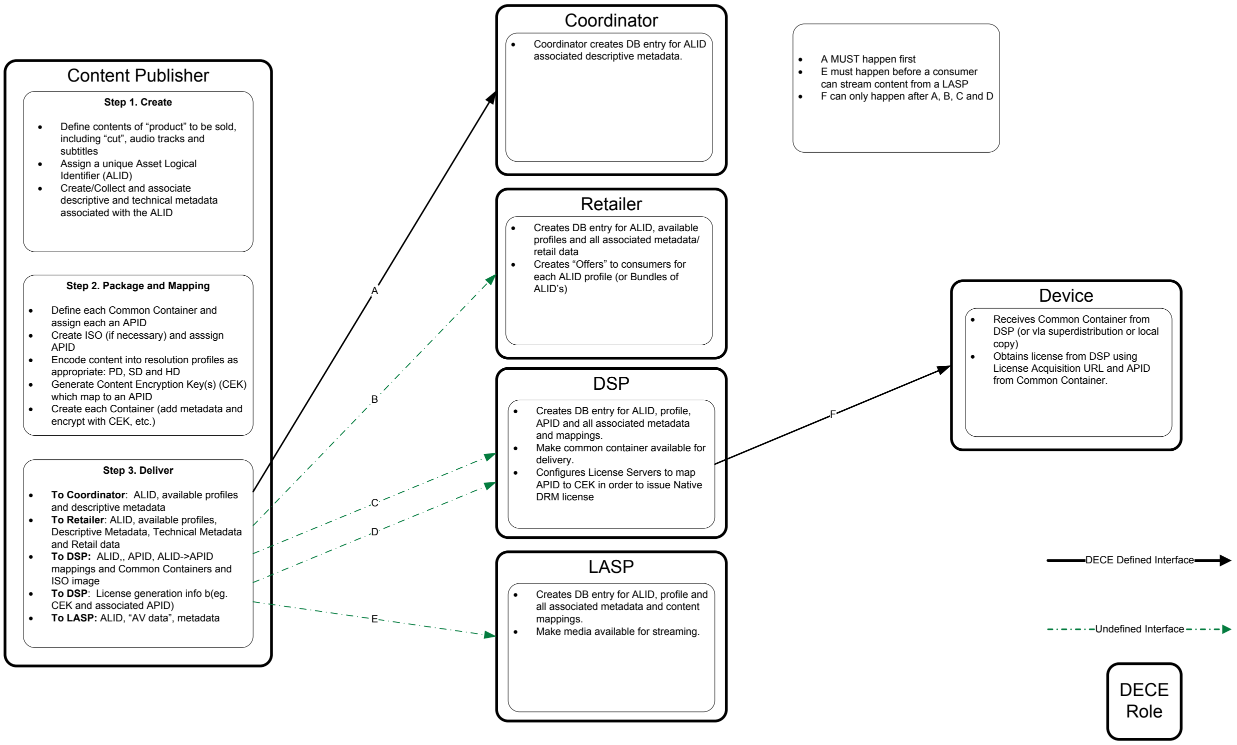

Content Publisher
Step 1. Create
Define contents of "product" to be sold, including "cut", audio tracks and subtitles
Assign a unique Asset Logical Identifier (ALID)
Create/Collect and associate descriptive and technical metadata associated with the ALID
Step 2. Package and Mapping
Define each Common Container and assign each an APID
Create ISO (if necessary) and asssign APID
Encode content into resolution profiles as appropriate: PD, SD and HD
Generate Content Encryption Key(s) (CEK) which map to an APID
Create each Container (add metadata and encrypt with CEK, etc.)
Step 3. Deliver
To Coordinator: ALID, available profiles and descriptive metadata
To Retailer: ALID, available profiles, Descriptive Metadata, Technical Metadata and Retail data
To DSP: ALID,, APID, ALID->APID mappings and Common Containers and ISO image
To DSP: License generation info b(eg. CEK and associated APID)
To LASP: ALID, "AV data", metadata
A
B
C
D
E
F
Coordinator
Coordinator creates DB entry for ALID associated descriptive metadata.
Retailer
Creates DB entry for ALID, available profiles and all associated metadata/ retail data
Creates "Offers" to consumers for each ALID profile (or Bundles of ALID's)
DSP
Creates DB entry for ALID, profile, APID and all associated metadata and mappings.
Make common container available for delivery.
Configures License Servers to map APID to CEK in order to issue Native DRM license
LASP
Creates DB entry for ALID, profile and all associated metadata and content mappings.
Make media available for streaming.
A MUST happen first
E must happen before a consumer can stream content from a LASP
F can only happen after A, B, C and D
Device
Receives Common Container from DSP (or vla superdistribution or local copy)
Obtains license from DSP using License Acquisition URL and APID from Common Container.
DECE Defined Interface
Undefined Interface
DECE Role

Figure 23 – DECE High Level Content Publishing Architecture

9.1 Content Provider

The starting point for the DECE publishing flow is when the Content Provider is ready to make a DECE product available for sale and fulfillment.

9.1.1 Product Creation

Product Creation involves defining what will be sold (logical assets), how it will be fulfilled (containers) and how it will be described (metadata). It is important to have a relatively broad view of the product; for example, think not just of an episode, but consider it as part of a season, and in turn part of a show. Consider how other assets, such as DVD extras, will be included in the product. This definition needs to be detailed to include which video, audio and subtitle tracks will be provided. DECE Content Publishing [DPublisher] provides guidance on product structuring.

Generally, the first step is to identify which Rights will be sold and in what combination. This closely aligns with the physical assets (Containers) that the User gets when purchasing the product. The Rights definition also includes which Media Profiles (i.e., HD and SD) will be offered.

The next step is to detail the product definition. This includes defining the specific Profiles, Rights and Containers, and the mapping between Rights/Profiles and Containers. Containers need to be defined to the track level (video, audio, and subtitle). It is necessary to determine track assignment, coding parameters, encryption key structure and so forth.

The Content Providers and Retailers may collaborate on any aspect of Product Creation, although that is outside of DECE scope.

9.1.2 Metadata

It must be determined which Metadata will be prepared for the product, including metadata associated with the Right (Basic Metadata), metadata describing parent objects (also Basic Metadata) and associated with the Containers (Container Metadata.) Each metadata element has a globally unique ContentID.

There is also metadata associated with product structures, in particular Bundles. Bundles describe groupings of products not otherwise described by the metadata structure. This allows products to consist of collections of works constructed for marketing purposes (e.g., all movies with a particular actor).

Metadata is described in detail in [DMeta].

9.1.3 Content Preparation for a DSP

Once defined, the product must be built. Although this section describes Container construction in a particular order, as long as a Container is valid, it need not be constructed in this order.

First the video, audio, and subtitles must be gathered and encoded and built into Containers in accordance with DECE Media Format Specification [DMedia]. Discrete Media must also be constructed if required for the profiles to be offered.

Most DECE Containers contain encrypted tracks, protected by Digital Rights Management. The key structure must be defined, Content Encryption Keys (CEKs) generated and content appropriately encrypted in accordance with the [DMedia]. Keys must be managed securely.

Identifiers must be created for the product. This includes Asset Logical IDs (ALIDs) for the Right, ContentIDs for metadata, and Asset Physical IDs (APIDs) for Containers. The requirements on these identifiers are that they conform to the identifier encoding rules in this specification, and they are globally unique. Encoding rules allows Content Providers to use standard ID schemes, such as [ISAN], or house IDs while creating container(s).

Containers contain Required Metadata and may contain Container Optional Metadata as defined in [DMeta] Section 4 and Content Publishing [DPublisher] Section 4.2. How the metadata is stored in the DCC is described in [DMedia], Section 2.3.3. Appropriate metadata is generated and inserted into the Container. If optional metadata is included, it should cover the Basic Metadata for the media and Digital Asset Metadata for each track. That is, the overall work should be described as well as each track. There are provisions for including multiple languages for Content Providers to use as appropriate for their products.

If Discrete Media Rights are supported, the Discrete Media packages must be prepared and encrypted as described in [DDiscrete].

When DMPs are to be used, the Content Provider prepares Original DECE Media Packages (ODMP) and additional components to be added to the ODMP. Note that Containers are essential elements of a DMP, but are not necessarily present in the ODMP.

When Content Provider intends to offer Content through Late Bound Common Streaming, the Content Provider can create CSF tracks suitable for Common Streaming in conjunction with the downloadable DECE Container tracks. Note that this does not preclude the DSP from creating its own Late Bound Common Streaming tracks.

9.1.4 Content Preparation for a LASP

The format of content published to LASPs is not defined by DECE, it is important that the appropriate media packages are prepared for conveyance to LASPs. These media packages may be Digital CFF Containers, although alternatives are also acceptable.

When Content is to be offered through Common Streaming, the Content Provider can create CSF tracks suitable for Common Streaming. Note that this does not preclude the DSP from creating its own Common Streaming tracks.

9.1.5 Delivery

Once everything is prepared, it must be delivered.

9.1.5.1 Delivery to Coordinator

The Content Provider delivers information to the Coordinator, typically using the API interface defined in [DCoord] Section 6. Published information includes basic metadata, for both Assets being offered (Logical Assets) as well as parent information (e.g., seasons and shows); physical metadata for each Container, mappings between Logical Assets and Metadata (ALID to ContentID), mappings for fulfillment (ALID to one or more APIDs) and any Content Provider defined Bundles. Logical to Digital Asset Mapping also includes policies, such as Licensing and Fulfillment Windows, if any (see Section 7.4.5).

[DCoord] Section 6 describes the Coordinator data structures and APIs for publishing metadata, the Logical to Digital Asset Mapping Table, and creating Bundles.

9.1.5.2 Delivery to Retailer

Although out of scope of DECE specification, it is assumed that Content Providers will make the ALID, available profiles, metadata, bundle information as well as business rules available to Retailers.

9.1.5.3 Delivery to DSP

Also out of scope for DECE specification is the delivery mechanism to DSPs. But the DSP must receive the Containers for fulfillment, along with the corresponding ALID, APID, and the Contents Encryption Key (CEK) and any other information needed to generate licenses.

DSPs need to securely handle and manage the CEKs in accordance with the DSP agreements.

9.1.5.4 Delivery to LASP

LASPs must receive the ALID, media and other information necessary to stream content in a form that the LASP can use to stream media which is out of scope for DECE specification. This may be in the form of Containers or some other format such as mezzanine files.

9.1.6 Product Update

Products may change over time, either for marketing reasons or because of a need to correct an anomaly in the product.

It is the responsibility of the Content Provider to distributed updates to appropriate destinations, including the Coordinator, Retailers, DSPs and LASPs.

Metadata may be updated, but it must include a revision to allow 3rd parties to determine which version is the most recent (`UpdateNum` element).

Bundles should not be updated. Bundles contain information about how a product was offered and sold. If a bundle changes, it may cause confusion and support issues with Users. Content Providers should create new bundles (new BundleIDs) to correct bundle issues.

Containers may be updated if necessary. They must be distributed to DSPs and LASPs. DECE supports replacing Containers with improved Containers. The Content Provider may determine whether downloads and/or licensing on the old Container is still allowed. There is also a means to halt distribution of a Container (e.g., if it is found to violates a parental control restriction). These Containers may not be downloaded or licensed, and are considered 'recalled'. Content Providers may specify region and time based download and licensing policies to implement holdbacks and other contractual restrictions. These are handled through the Logical to Digital Asset Mapping Table (Section 7.4.5)

9.2 Retailer and DSP Content Preparation

Once the Retailer has the necessary information and appropriate agreements, it may proceed with selling the product.

DECE allows, although business agreements may not, the Retailer to further define the product. Retailers can group Logical Assets together into Bundles. Bundle construction is the same as for Content Providers and must be posted to the Coordinator.

Even without Bundles, Retailers can sell multiple assets together, such as offering an entire season consisting of all individual episodes. In many of these grouping, the metadata already defines the grouping structure so there is no need to create a Bundle.

Although the process of selling is discussed elsewhere in this specification (Section 10), it is worth noting that the Retailer posts relevant grouping information into the Rights Token (i.e., the `SoldAs` element). If the asset was sold as part of a bundle, the BundleID is posted. If it was sold as part of a grouping covered by metadata, the list of ContentIDs associated with that group are included in the Rights Token. This allows the User to later reconstruct how the Rights were obtained.

The Retailer or DSP (which one is outside DECE scope to define) must modify the Container to facilitate licensing. In particular, they must include the appropriate Base Location (see Section 8.3.1) information into the Container prior to download, allowing the Device to direct to the appropriate License Manager. The Retailer or DSP may also include Purchase Location (see `basePurlLocation`, Section 8.3.3) used by a DECE Device to construct a Purchase URL facilitating purchase of superdistributed or shared Containers.

DSPs may insert licenses as part of the download process to make Content playable when it arrives at the Device, without an additional licensing step.

Retailers and DSPs must keep information current, particularly which Containers should be offered for download and licensed. This information should arrive from the Content Provider, but the DSP must also keep track of ALID to APID mappings to ensure replaced and recalled Containers are handled correctly.

9.3 LASP

LASP are not directly involved in publishing other than as recipients of metadata and media.

10 Purchasing Content

The DECE does not specify how a User selects a Retailer or how the Retailer enables a User to browse and purchase Content. Content purchased from any DECE Retailer will play on any DECE Device with the appropriate Profile.

Once a piece of Content is purchased, DECE specifies how the purchased Rights are coordinated across DECE Devices and Approved DRMs and how Ecosystem limits such as number of concurrent streams are maintained for the DECE Account.

10.1 Coordinating Purchased Rights

Once a Right to Content is purchased, a Retailer must update the Coordinator to add the purchased Rights into the Rights Locker in the User's DECE Account.

A Retailer SHALL call `RightsTokenCreate` to the Coordinator with a fully formed `RightsToken` as described in the DECE Coordinator Interface Specification [DCoord] Section 7.1.2 and Section 7.2.

This creates a Rights Token in the User's Rights Locker granting rights (such as download, streaming, and Discrete Media export) to various Media Profiles (e.g. HD or SD) of a piece of Content specified by an ALID and ContentID or to a BundleID. It also includes information about the purchase transaction, and other information described in the `RightsTokenCreate` API.

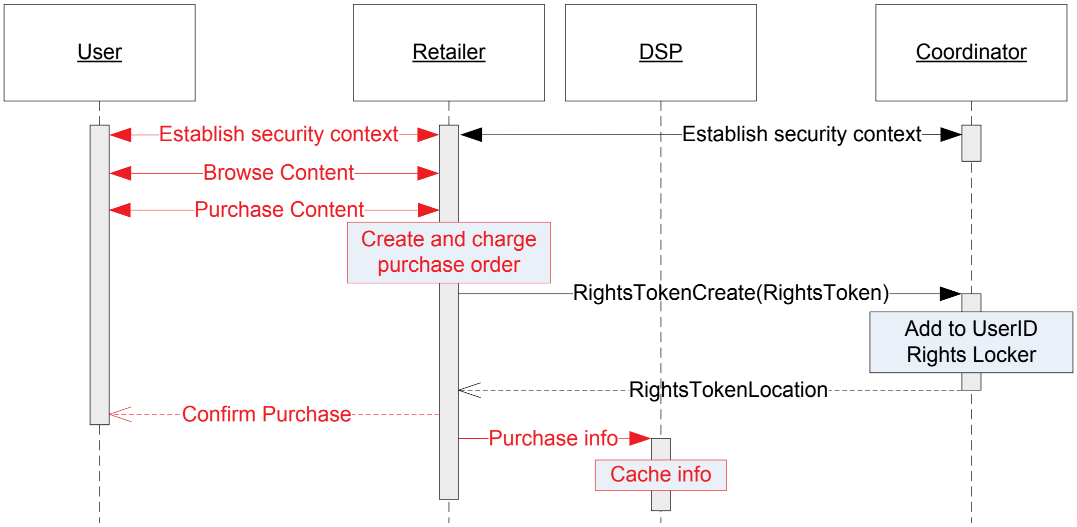

User
Retailer
DSP
Coordinator
Establish security context
Establish security context
Browse Content
Purchase Content
Create and charge purchase order
RightsTokenCreate(RightsToken)
Add to UserID Rights Locker
RightsTokenLocation
Confirm Purchase
Purchase info
Cache info

Figure 24 – Purchasing Content

10.1.1Creating the Rights Token

The Retailer must create a Rights Token that describes the Right purchased and the context of the purchase. In this context, the term ‘purchase’ is used broadly to cover any action that leads to the acquisition of a Right.

The Retailer SHALL create Rights Tokens in accordance with DECE Policies. For example, the Rights Token must include all required Media Profiles.

The Retailer SHALL create Rights Tokens in accordance with the terms of the purchase. That is, the content of the Rights Token accurately reflects aspects of the purchase the asset purchased, rights acquired, the context of the purchase, and parties involved in the purchase.

10.1.1.1 Rights Identification

The `ALID` element of the Rights Token defines which asset is added to the Account. The Retailer SHALL populate the `ALID` element with the Asset Logical ID for the asset being added to the Rights Locker.

The `RightsProfiles` element defines the Rights are around each Media Profile. The Retailer SHALL create a `PurchaseProfile` element for each Media Profile associated with the purchase. In accordance with DECE Policies, subject to change, the sub-elements are set up as follows:

- `DiscreteMediaRightsRemaining` element is included if exporting to Discrete Media is supported
- `CanDownload` set to ‘true’
- `CanStream` set to ‘true’

Note that the Rights Token is structured to support future rental Use Cases. However, these are not supported at this time.

10.1.1.2 Metadata Reference

The `ContentID` element SHALL be set to the ContentID corresponding with the ALID.

10.1.1.3 Metadata regarding Sale

The `SoldAs` element is used to describe the context of the sale.

If a right is sold alone, that is a single ALID is the only asset sold in the transaction, `SoldAs` will typically be absent.

The Retailer SHALL include the SoldAs element when more than one asset is purchased together. Note that this is important to support views of the Rights Locker, and for Customer Support.

If present, the SoldAs element SHALL include either one or more ContentID elements or a BundleID element.

As described in DECE Content Publishing [DPublisher], Section 7, structure of content can either be defined in metadata or in a Compound Object. In metadata-structured content, such as episodes of a season, a sequence of ContentID elements will fully describe the grouping. When a product is structured as a Compound Object, a BundleID element best describes the grouping.

If Rights are sold in a structure not covered by metadata or an existing Bundle, the Retailer SHOULD create a Bundle as defined in DECE Coordinator Interface Specification [DCoord], Section 6.

When viewing a Rights Locker, it can be helpful to see a description of a grouping; for example, "Show XYZ, Season 2." The Retailer MAY include a DisplayName in the SoldAs element. The Retailer is expected to include this element if they determine it will improve readability.

10.1.1.4 Purchase Info

The Retailer SHALL populate the PurchaseInfo element.

The PurchaseInfo element is populated as follows:

- RetailerID SHALL be the Retailer's RetailerID
- RetailerTransaction SHALL include a string that allows the Retailer to associate the Rights Token with an internal transaction. Note that this supports customer support.
- PurchaseAccount SHALL be the AccountID for the DECE account for which the Right was originally purchased. The AccountID can be obtained from the Security Token.
- PurchaseUser SHALL be the UserID (obtainable from the Security Token) for the User who purchased the Right. PurchaseTime SHALL include UTC date and time of the transaction.

Note that fields in PurchaseInfo are not modified if a Rights Token is moved to another Account. Therefore, over time, certain information such as PurchaseAccount will not necessarily align with the DECE Account.

10.1.1.5 Fulfillment and Licensing Locations

Part of the Rights Token created by the Retailer includes Internet locations used for licensing and downloading Content. These locations are specific to the DSP, and can be set by the DSP on behalf of the Retailer since the Retailer's Security Token enables it to be shared with a DSP.

If fulfillment is required, the Retailer SHALL provide a mechanism to allow the purchased Content to be downloaded.

If fulfillment is required, the Retailer SHALL provide one or more `FulfillmentWebLoc` elements for each Media Profile included in the Right. The `FulfillmentWebLoc` is a URL to a fulfillment web page or a DCC. How the `FulfillmentWebLoc` is used is described in Section 11.1.2. More than one `FulfillmentWebLoc` may be specified with the same `MediaProfile` attribute along with an associated `Preference` indicating a preferred order as defined in [DCoord] Section 7.2.8 and 7.2.9.

The Retailer MAY use a distinct `FulfillmentWebLoc` URL per Media Profile, or the Retailer MAY use the same `FulfillmentWebLoc` URL for all Media Profiles. Using the same URL allows the Retailer to let the User select the desired profile on a common fulfillment web page.

The Retailer MAY also include additional information in the `FulfillmentWebLoc` URL (e.g. in the URL query string) to allow the Retailer to implement access control as described in Section 11.1.4.8.

If fulfillment is required, the Retailer SHALL provide one or more `FulfillmentManifestLoc` elements. The `FulfillmentManifestLoc` is a URL to a network location where a media manifest can be obtained. The manifest file is defined in Section 11.1.4.5. Use of this field is explained in Section 11.1.3.

The Retailer SHALL ensure that a valid manifest file is present at the `FulfillmentManifestLoc` endpoint.

The Retailer SHALL ensure that the manifest file provides the location of at least one Container for each applicable Media Profile, and that a valid Container is present at the provided location(s).

If fulfillment is required, the Retailer SHALL provide one `LicenseAcqBaseLoc` element. The `LicenseAcqBaseLoc` element contains the Base Location used to calculate the License Acquisition URL. Section 8.3.2 describes how to create the Base Location.

The `StreamWebLoc` element contains a URL and is designed to help the User find Streaming access provided by the Retailer from which they acquired the Content. Multiple instances of this element can be provided. `StreamWebLoc` is primarily designed for Browsers, although applications may use the

URL too. It is assumed that applications would use file type to determine which `StreamWebLoc` instance to use.

If Streaming is required, the Retailer SHALL provide one or more `StreamWebLoc` elements for the Rights Token or one or more StreamWebLoc elements for each Media Profile.

LASPs may support Streaming from a Browser. That is, the User can select a link and Streaming can begin; perhaps by launching an application. Alternatively, LASPs may Stream only from applications not accessible directly from a browser (e.g., an embedded application in a device).

In cases where Streaming is accessible from a Browser, the Retailer SHALL provide a `StreamWebLoc` that links directly or closely to a location where the User can Stream.

The Retailer SHALL NOT provide a `StreamWebLoc` that links to the top level of a web site unless Streaming is directly available from the top level.

In cases where Streaming is not accessible from a Browser, the Retailer SHALL provide a `StreamWebLoc` that links to a web page containing instructions on how to Stream the Content.

10.1.2 Updating the DSP to Enable Licensing

The Retailer MAY have a mechanism to inform its DSPs of the purchase enabling the DSP to license the purchased Content without requiring a call to the Coordinator to check the Rights Token. This communication is out of DECE scope.

The DSP MAY create a license when notified by the Retailer of a purchase, or it MAY defer license creation until License Acquisition as described in Section 12.

10.2 Purchasing Superdistributed or Copied Content

While the DECE does not specify how to locate a Retailer in general, it does provide a mechanism for a Retailer or Content Provider to place a suggested Retailer into a Digital CFF Container. Then if a User has a copy of the Container they have an easy way to locate a preferred Retailer able to sell Rights to the Content.

This can happen when Content is Superdistributed (see Section 15), or simply copied or shared between friends. In any of these cases, the User will not have a license to view the Content, and the native DRM system would not recognize any licenses stored in the Container as valid as they would not be keyed to the User’s DRM domain.

To ease purchasing rights to a Container already in the User's possession, a Retailer or Content Provider (operating in conjunction with a Retailer) can store a Purchasing Location in the Container. Section 8.3.3 describes how the Purchasing Location in the Container can be used to construct a Purchase URL, which a DECE Device may use to locate a Retailer able to sell Rights to the Content.

There is no implicit guarantee that the Right can be purchased. For example, the Retailer may have stopped selling that content. But there is a guarantee that if the Right is purchased, the Container with the Purchasing Location will be licensable under that Right.

Other methods may be used to locate a Retailer. A DECE Device may use third party services, or have a pre-existing relationship with one or more DECE Retailers.

11 Content Fulfillment

DECE requires Retailers to make an Account's Content available to all DECE Devices joined to the Account. To ensure that all DECE Devices can acquire Content "out of the box," there is minimum required functionality for all DSP download servers and all DECE Devices. Retailers, DSPs, and DECE Devices are free to implement additional or alternative download features as long as the minimum functionality remains available. (For example, download managers may implement P2P transport, job scheduling, bandwidth throttling, multithreaded downloads, and so on.) Alternative download mechanisms are out of scope of DECE.

DECE supports several methods of delivering Containers (Content packaged into a DCC) to DECE Devices and incorporating those Containers into the DECE Device's storage. Fulfillment is the term used to describe the process of delivering Content in the form of Containers to the DECE Device.

Fulfillment includes:

- Downloading Containers directly by a DECE Device
- Downloading a Discrete Media package using a Discrete Media Client
- Using a proxy such as a personal computer or media server to download and copy a Container to a DECE Device.
- "Superdistributing" Content by preinstalling or copying a Container onto a DECE Device or media (see Section 15).

Fulfillment may be initiated through a Device, a Retailer, the Web Portal, or any Node that can get the fulfillment information from a Rights Locker query. Details of how the download is initiated are left to the Device or Node. Download may be done one file at a time using standard HTTP mechanisms ("Web download") or by a Download Manager using the DECE download manifest mechanism ("Manifest download").

11.1 Container Download

11.1.1 Download Locations Provided in the Coordinator

One or more fulfillment locations may be obtained from the Coordinator via the `RightsTokenGet` query. See [DCoord] Section 7.1.4.

The relevant elements of the Rights Token are `FulfillmentWebLoc` and the `FulfillmentManifestLoc`. At least one of each will exist, and there may be more than one. These location elements each contain a URL associated with a Media Profile and optionally an element called Preference defined as an integer. Preference defines an ordering.

DECE Devices and other download implementations SHOULD use the URLs with the following precedence:

1. URLs with lower numbers Preference are used before URLs with higher number Preference
2. URLs with Preference are used before URLs without Preference
3. Two or more URLs with the same Preference may be used in any order
4. Two or more URLs without Preference may be used in any order

The fulfillment locations are specified in the Rights Token when it is created when Content is purchased as described in Section 10.1.1.

11.1.2 Web-initiated Download from a Fulfillment Web Page

A Web-initiated download is done by directing a Web Browser to a fulfillment URL provided by a Retailer or DSP to download a DCC for a given Media Profile. The URL is stored in the Rights Token by the Retailer in the FulfillmentWebLoc element with the desired MediaProfile attribute (see Section 10.1.1.5). A Retailer may also direct a Web Browser to a fulfillment web page, typically after Content is purchased.

The FulfillmentWebLoc can be a direct URL to the DCC for the specified Media Profile or a URL to a fulfillment web page containing links for downloading one or more Containers. A fulfillment web page may have links to individual files for HTTP download using the download feature of the browser, or may point to Fulfillment Manifest files for use by a Download Manager if one is available (see Section 11.1.4.5 below).

There is a separate FulfillmentWebLoc element with a MediaProfile attribute for each Media Profile in the Right. While this can be used to point to an individual file or fulfillment web page for a given profile, the same URL can be used for multiple Media Profiles if a Retailer prefers to have a web page containing download options for several or all Media Profiles.

Individual Digital CFF Containers use the video/vnd.dece.mp4 MIME type (see [DCIF] Section 2.1), which may be recognized by the Web Browser to launch a player or may simply be downloaded.

It is recommended that the fulfillment web page provide a description for each link so that that User can choose the appropriate Container(s) to download for the desired Media Profile (e.g. SD or HD).

Containers may be collected into a single file, such as a zip file. The details of packaging into a single file by the DSP and un-packaging by the User are out of scope of DECE.

11.1.3 Download Manager Download using a Fulfillment Manifest

A Fulfillment Manifest is provided by the DSP to reference one or more Digital CFF Containers for a Download Manager to selectively download. DECE does not define how a download manager works, but does define the Fulfillment Manifest structure and the HTTP download mechanism that SHALL be supported by all DSPs for use by a DECE-compliant download manager.

Section 11.1.6 below discusses the DSP's responsibility to ensure a Digital CFF Container is not subject to fulfillment restrictions before allowing a download to be initiated.

`FulfillmentManifestLoc`, the URL to a Fulfillment Manifest, is obtained from the Coordinator via a `RightsTokenGet` query or from a link. The URL references a Fulfillment Manifest resource retrieved with HTTP GET. The Fulfillment Manifest is an XML structure defined by `FulfillmentManifest-type`. XML schema documentation conventions are the same as the Coordinator Interface Specification [DCoord].

The download manager retrieves the Fulfillment Manifest from the provided location, chooses which assets to download, and uses the URLs provided to connect to an HTTP server to download the assets. The download manager MAY interact with the User and list the available assets for the User to choose from, or MAY select the assets automatically based on User preferences (or a combination of both). The download manager may use the APID in the Manifest to retrieve information about each downloadable Container, such as audio language, from the Coordinator.

The Manifest can include information to support DECE Media Package (DMP) downloading. When downloading the DMP, the download manager first downloads the Original DMP (ODMP). Information in the DMP instructs the download manager what other assets are required for download, and which are optional. The download manager downloads the required assets. The download manager MAY interact with the User and list the available optional assets for the User to choose from, or MAY select the optional assets automatically based on User preferences (or a combination of both).

The Manifest can include information to support updates. When an APID is replaced (including recalls where the asset is replaced), the replaced APIDs are listed, indicating to the download manager that an asset has been replaced.

A DMP is updated by revising parts of that DMP while maintaining the same DMPID. Content Providers publish updates only when the Manifest update mechanism supports those changes; otherwise a new DMP is created, identified by a new DMP ID.

When a DMP has been updated, the DSP SHALL include a properly constructed DMP element in the Manifest as follows reflecting the current state of the DMP; and also including indication of obsolete DMP Parts. The specific mechanisms for signaling obsolete Parts is provided later in this section.

The Manifest supports Common Streaming through the StreamingInfo element. This element provides information to provide Common Streaming information corresponding with an audio, video or subtitle track that is part of a DCC.

DSPs SHALL support the HTTP/1.1 GET and RANGE GET commands [HTTP], with or without TLS [TLS], for download of files referenced in the Fulfillment Manifest. Download Managers MAY use GET or RANGE GET, with or without TLS, to download the files. Download Managers SHOULD support continuation of downloads that were interrupted.

11.1.4DMP Download and Update Process

This section defines the process for downloading and uploading DMPs.

Schema documentation conventions used this section follows conventions defined in [DMeta].

11.1.4.1 Download Process

DMP Download requirements are designed to support the following basic use cases:

- Download of a complete DMP that corresponds with a multi-track DCC. The Device downloads all tracks into a complete DMP that would correspond with a multi-track DCC.
- Download a subset of tracks into a DMP with the option of downloading some or all of any remaining tracks later. In this case, the ODMP is unchanged so the only download required is the new tracks.
- The ODMP has changed for any of a variety of reasons, such as track recall/replace, DMP part replacement (e.g., metadata) or the addition of new tracks. The Device would determine the changes, make the necessary updates, and download the additional elements as it determines appropriate. Note that a Device might discover an ODMP with a track recall change through a licensing failure.

The process always begins with an Initial Download. This includes the ODMP, other essential components, tracks indicated by the ‘forceDownload’ mechanism and any optional tracks the Device decides to download.

- Additional tracks can be downloaded following the original download in a process called Subsequent Download. In any instance where the ODMP might have been updated, it is necessary to check for ODMP updates prior to performing such a download. Note that certain licensing failures can indicate an ODMP update, so the Device might check for updates in this condition.

The process for Identifying Items that Require Download is as follows:

- DCC Parts to download are listed in DMPItem/ODMP/Presentation/APID. If APID/@forceDownload='true' the DCC with DCCResource/APID matching Presentation/APID must be downloaded.

- Application Parts to download are listed in DMPItem/ODMP/Application/@ApplicationID. If Application/@forceDownload='true' the Application with DMPItem/AppResourceList/AppResource/@ApplicationID matching Application/ApplicationID must be downloaded. The Application consists of items referenced in all instances of the associated DMPItem/AppResourceList/AppResource.

- Other Parts to download are listed in DMPItem/OCMP/OtherPart, Presentation/DependentPart and Application/DependentPart. If OtherPart/@forceDownload='true' the OtherPart matching OtherResource/@LocalSource must be downloaded. If DependentPart/@forceDownload='true' the DependentPart matching OtherResource/@LocalSource must be downloaded.

- If two Parts have the same LocalSource, it need not be downloaded more than once. This could occur, for example, if Presentations share DCCs or image Parts.

DCCResource, AppResource/AppItem, OtherResource and OtherPart all have elements to support download. These elements are:

- LocationURL is the Part download link.

- Hash is the Part hash

- Length is the Part length

- LocalSource is the local filename. Some filenames are constrained as specified in [DDMP], Section 4.1.3.

11.1.4.2 Initial Download

The process for initial downloading of a DMP is as follows

- Retrieve Manifest from DSP (as described above)
- Identify ODMP from Manifest's DMP element
- Download ODMP from DSP using LocationURL
- Identify Items that Require Download (as defined above)
- Download (as defined above)
- Validation consists of checking the hash and length of the downloaded file against the Part has and Part length in the Fulfillment Manifest

11.1.4.3 Subsequent Download

Subsequent Download is the capability to download portions of a DMP distinctly from Parts downloaded through the 'forceDownload' mechanism. It could in conjunction with the original download or it could be much later, perhaps years later. Subsequent Download includes the download of optional tracks (i.e., those not signaled as forced through the 'forceDownload' mechanism), track updates required due to a track recall, and other Parts of a DMP as required or desired.

It is not always necessary to download all items when the DMP is downloaded. That is, not all Parts are necessarily marked with @forceDownload='true'. In this case, the Device can download Parts later. Typically, the Device would determine which DCC or Application to download by using metadata information in the DMP, then obtain a new Manifest, locate the matching Part by APID or ApplicationID and download the associated Container or Application Parts

However, the potential exists for a DCC or Application to be replaced or recalled after the DMP was downloaded, outdating the information in the DMP.

The Manifest supports both downloading and updating any DMP Parts. The Manifest also provides information to clean up unused Parts.

11.1.4.3.1DCC and Media Application Subsequent Download

DCC Subsequent Download is downloading a DCC Part of the DMP after the initial DMP Download is complete and, where applicable, adding the DCC to the DMP.

Media Application Subsequent Download is downloading a Media Application Part of the DMP after the initial DMP Download is complete and, where applicable, adding the Media Application to the DMP.

APIDs are identified by looking for Manifest/DMPItem/AppResource elements containing APID elements matching the desired APID. Once matched, the LocationURL can be used to download the Container.

The process for Subsequent Download of a DCC or Media Application is as follows

- Retrieve Manifest from DSP (as described above)
- Determine if any DMP Updates are required. If an update is required, perform a DMP Update and restart the entire process. Note that it is possible that the Updated DMP will not have the desired APID, so the Device might also need to restart whatever process initiated the Subsequent Download.
- If update is not required, proceed
- Identify Parts that Require Download (as defined above)
- Download (as defined above) DCC or Media Application Parts.

11.1.4.4 Update Process

In the time that passes between Initial Download and Subsequent Download, there may be updates to the DMP. For example, when a DCC is been replaced the Table Of Content will be updated to reference an updated Presentation Description referencing the new DCC, so the these Parts along with the DCC must be updated.. The update process uses a combination of the [SMPTE2053] update process based on Version along with additional information provided in the Fulfillment Manifest. The Manifest also contains information the Device can use to remove Parts that are no longer needed.

The Update processes are as follows:

- SMPTE 2053 Part Update Process updates Table Of Contents Document, Presentation Documents and Media Application Documents. This process is as follows:
  - The Device determines an update is necessary by comparing the Version in the DMP's TableOfContent/@Version with the Version in DMPItem/UpdatedTOC/@Version. If DMPItem/UpdatedTOC is not present, no update is necessary.
  - If UpdatedTOC/@Version is greater that TableOfContents/@Version, TableOfContents must be updated with the Part referenced by UpdatedTOC.
  - Presentation Description Documents are then updated in accordance with [SMPTE2053].

- The DCC Update process updates DCCs. This process is as follows:
    - Note that the function of the DCC is to update DCCs that are already in the DMP. Other DCCs referenced in ODMP and Presentation Description Documents can be ignored. Note that ODMP and Presentation Description Documents to not relate old DCCs to new DCCs so it is impractical, although possible, to use these objects for update.
    - For each DCC in the DMP, the APID of that DCC is checked against ObsoleteAPID elements in DCCResource. If a match is found the downloaded DCC is obsolete and its replacement is in referenced in the DCCResource element that includes the match.
    - DCCs are then updated with the referenced DCCs.

Update information structure in the DMP is defined in [DDMP], Section 4.3.1.

If the Device opts to download a DCC or Media Application, then the Device is required to perform the DMP update functions.

11.1.4.5 Fulfillment Manifest File and element

The Fulfillment Manifest is returned as a file containing a `FulfillmentManifest` element.

Element	Attribute	Definition	Value	Card.
FulfillmentManifest		The root level element of a Fulfillment Manifest file.	`fmdece:FulfillmentManifest-type`	

11.1.4.6 FulfillmentManifest-type

This type is not included in the Right Token, but it is referenced by the Rights Token. DMP support items are distinct from non-DMP items to ensure backwards compatibility. Item is not used for DCCs within a DMP; for that, DMPItem/DCCResource is used.

Element	Attribute	Definition	Value	Card.
FulfillmentManifest-type				
ALID		Asset Logical ID fulfilled by this Manifest	md:AssetLogicalID-type	

Element	Attribute	Definition	Value	Card.
Item		Information about a multi-track DCC. This does not apply to DMP-related items.	fmdece:FulfillmentManifestItem-type	0..n
DMPItem		Information about the DMP	fmdece:FulfillmentManifestDMP-type	0..n
StreamingInfo		Information to support Common Streaming of Content. One instance per APID	fmdece:FulfillmentManifestStreamInfo-type	0..n

11.1.4.7 FulfillmentManifestItem-type

Element	Attribute	Definition	Value	Card.
FulfillmentManifestItem-type				
Description		Description of the individual item. This is provided for user interfaces that list individual files	xs:string	1..n
	Language	Language associated with Description. At most one instance of Description may omit Language. If missing, the associated Description is the default.	xs:language	0..1

Element	Attribute	Definition	Value	Card.
MediaProfile		Media Profile (i.e., HD or SD). This allows a manifest to include all required files, including those of lower profile (e.g., SD files for an HD Right). This is required for any item that is associated with a Media Profile (e.g., DCC).	mddece:AssetProfile-type	0..1
APID		Asset Physical ID for the Asset	md:AssetPhysicalID	
LocationURL		URL reference to location(s) of Container. May include access control information. If multiple instances exist, any may be used.	xs:anyURI	1..n
Hash		File hash	xs:string	0..1
	Type	hash type	xs:string 'crc32' 'sha1' 'md5'	
Length		Byte length of the file	xs:integer	
LocalName		Name for file in local file system. This allows the manifest to point to a single location for a Container, yet customize the local file name, possibly for each manifest.	xs:string	0..1

Element	Attribute	Definition	Value	Card.
ObsoleteAPID		APID for the Asset superseded by Asset referenced by the APID element. This includes all APIDs in LogicalAsset/…[RecalledAPID|ReplacedAPID]	mdAssetPhysicalID	0..n

11.1.4.8 FulfillmentManifestDMP-type

This element contains information necessary to download the DMP and the related parts, including Presentations (including DCCs), Applications and other components.

Element	Attribute	Definition	Value	Card.
FulfillmentManifestDMP-type				
Description		Description of the DMP. This is provided for user interfaces	xs:string	1..n
	Language	Language associated with Description. At most one instance of Description may omit Language. If missing, the associated Description is the default.	xs:language	0..1
LocationURL		URL reference to location(s) of ODMP. May include access control information. If multiple instances exist, any may be used.	xs:anyURI	1..n
Length		Byte length of the file	xs:integer	
LocalName		Name for DMP in local file system.	xs:string	0..1
ODMP		Definition of the DMP	mddece:OCMP-type	
DCCResource		Information about DCCs in the DMP	fmdece:FulfillmentManifestDCCResource-type	0..n

Element	Attribute	Definition	Value	Card.
AppResourceList		Information about Media Applications in the DMP	fmdece:FulfillmentManifestAppResourceList-type	0..n
OtherResource		Information about other resource in the DMP. For example, BaseLocations.	fmdece:FulfillmentManifestOtherResource-type	0..n
UpdatedTOC		Provides information about Table Of Content replacements, if any. UpdatedTOC/Type='TableOfContents'	fmdece:FulfillmentManifest2053Part-type	0..1

11.1.4.9 Types used to define DMP Download and Update

There are four primary types of DMP Part that are individually addressed in the Fulfillment Manifest

- DCC Resource – Part of a Presentation, these Resources are DCCs. Each DCCResource element refers to a DCC Resource.
- Application Resources – A Media Application contains one or more Application Resources. AppResourceList is a collection of all Application Resources associated with an ApplicationID. AppResourceList/AppResource refers to an Application Resource.
- Other Resources – Other Resources can be included in a DMP. At the top level is BaseLocations. Other Resources can include DMP Parts or Presentation Parts that are note DCCs, Applications or [SMPTE2053] parts (i.e., TOC, Presentation Description Document and Media Application Document). OtherResource elements refer to other Resources.
- Table of Contents – The [SMPTE2053] TOC is required for a ODMP. However, the TOC can be updated. UpdatedTOC, if present, references the most recent TOC.

For each of these types an element type is defined to provide information on downloading, updating and removing, as appropriate, Resources of that type.

11.1.4.9.1FulfillmentManifestDCCResource-type

This element defines how to download the DCC associated with a given APID.

A DCC is obsolete if its APID matches an instance of DCCResource/ObsoleteAPID.

Element	Attribute	Definition	Value	Card.
FulfillmentManifest DCCResource-type			fmdece:FulfillmentManifestResourceRef-type (by extension)	
APID		APID of the DCC	md:AssetPhysicalID-type	
ObsoleteAPID		APID of DCCs that have been replaced by this DCC	md:AssetPhysicalID-type	0..n

11.1.4.9.2FulfillmentManfiestAppResourceList-type

An Application consists of one or more Application Resources referenced by AppItem elements. Within a given AppResource element, active Application Resources are listed in AppResourceList/AppResource elements. Note that this refers to any Media Application Resource, whether it is referenced by a Presentation or the TOC.

If an Application Resource is obsolete its filename is listed in AppResourceList/ObsoleteLocalSource.

Element	Attribute	Definition	Value	Card.
FulfillmentManifestAppResourceList-type				
	ApplicationID	Application ID for the Application	md:id-type	
AppResource		Information required to download or update an Application Resource.	fmdece:Fulfillment ManifestResourceRef-type (by extension)	1..n
	isPresent	Indicates whether the Application Resource is present in the ODMP.		

Element	Attribute	Definition	Value	Card.
	forceDownload	Indicates whether the download manager must download the Application Resource immediately following ODMP download.		
	overwrite	Indicates whether file of same LocalSource should be overwritten.		
ObsoleteLocalSource		LocalSource of Resources replaced by this Resource.	xs:string	0..n

11.1.4.9.3FulfillmentManfiestOtherResource-type

Other Resources include BaseLocations, Image files and metadata. Some of these files are included in the ODMP, so they only will be included in the Manifest if they have been updated.

Some types of Resources have the own versioning mechanism. If so, the means of determining replacement is defined in [DDMP], Section 4.3.1. Otherwise, Resources are determined to be obsolete if their filename including path matches OtherResource/ObsoleteLocalSource.

Element	Attribute	Definition	Value	Card.
FulfillmentManifest OtherResource-type			fmdece:FulfillmentManifestResourceRef-type (by extension)	
Type		Type of this Resource	fmdece:FulfillmentManifestType-type	
Version		Version of the Resource. The meaning of Version depends on the type of Resource.	xs:string	0..1
ObsoleteLocalSource		LocalSource of Resources replaced by this Resource.	xs:string	0..n

Hash SHALL be included.

The Type element SHALL have one of the following values

- 'Image' – Image file, compatible with [DMeta], Section 3.2.
- 'BaseLocation' – BaseLocations document as per [DMeta], Section 4.4.
- 'Other' – File that is not one of the above.

11.1.4.9.4FulfillmentManifest2053Part-type

Each instance of this element describes a part in the DMP subject to replacement. If @Version is more current than Version in the DMP, this part is more current.

Any [SMPTE2053] Presentation Part or Application Part that is not referenced by TableOfContents is obsolete.

Element	Attribute	Definition	Value	Card.
FulfillmentManifest2053Part-type			fmdece:FulfillmentManifestResourceRef-type (by extension)	
	Version	Version of the file referenced here. Version corresponds with Version definitions in [SMPTE2053]	xs:nonNegativeInteger	
Type		Type of Part (see below)	xs:string	0..1

11.1.4.9.5FulfillmentManifestResourceType-type

This simple type is xs:string and SHALL have one of the following values:

- 'TableOfContents' – [SMPTE2053] Table Of Contents
- 'PresentationDescription' – [SMPTE2053] Presentation Description document (//Presentation)
- 'MetadataMovie' – MetadataMovie document as per [DDMP]
- 'MetadataTail' – MetadataTail document as per [DDMP]
- 'Image' – Image document as per [DDMP]
- 'Other' – File that is not one of the above.

11.1.4.10FulfillmentManifestStreamInfo-type

This element provides streaming information corresponding with a Physical Asset, reference by APID. The principle is that an Asset might be represented by a downloadable DCC associated with that APID and one or more Common Streaming forms associated with that same APID.

As a DCC can contain more than one track, the TrackReference attribute disambiguates which track is described by this element.

Element	Attribute	Definition	Value	Card.
FulfillmentManifestStreamInfo-type				
	PresentationID	The PresentationID for the Media Presentation that can be streamed through this MPD.	md:PhysicalAssetID-type	
MPDLocation		URL representing location of MPD. If multiple instances exist, any may be used.	xs:anyURI	1..n
ExperienceMedia Application		Information to select titles and tracks, and otherwise navigate the streaming experience.	mddece:ExperienceStreamApp-type	0..1
BaseLocations		BaseLocation information required to acquire licenses.	mddece:ContainerBaseLocations-type	0..1

11.1.4.10.1FulfillmentManifestResourceRef-type

This complex type is used to refer to Resources. It is a building block for other types that reference Resources.

Element	Attribute	Definition	Value	Card.
FulfillmentManifest ResourceRef-type				

Element	Attribute	Definition	Value	Card.
LocationURL		URL reference to location(s) of Container. May include access control information. If multiple instances exist, any may be used.	xs:anyURI	1..n
Hash		File hash	md:Hash-type	0..1
Length		Byte length of the file	xs:integer	
LocalSource		Part Name in accordance with [SMPTE2053]		

11.1.5 Access Control

Content protection is provided by the DRM Client, so downloading does not per se require authentication or secure communication. However, Retailers and DSPs will typically wish to provide download services only to Users with a legitimate right to access the content.

Authority to access Content is provided by the Retailer. The `FulfillmentWebLoc`, `FulfillmentManifestLoc`, or `LocationURL` URLs may include user authentication credentials, which should be opaque to the Download Manager or Web Browser. For example, the DSP may check the Rights Token in the Coordinator to ensure that the User has purchased the Content, and then place SAML or other authentication tokens specific to the User in the URLs it generates for the Fulfillment Manifest. Another example approach would be for the DSP to generate single-use or limited-time URLs managed by a CDN.

If the DSP cannot fulfill a request due to an access control violation, the DSP SHALL return an HTTP 403 error response as accordance with Section 11.1.7.

11.1.6 Replaced/Recalled APIDs and Fulfillment Window Restrictions

Content Providers may occasionally need to recall or replace APIDs, or specify time periods where fulfilling Content may be restricted, as described in Section 7.4.5.

The DSP SHALL NOT Fulfill Containers whose APID is listed as Recalled in the associated LogicalAsset with the exception that the RecalledAPID MAY be licensed if the LicensingAllowed attribute is set to 'true'. See [DCoord] Section 6.5.2.4.

For APIDs listed as Replaced, the DSP SHOULD instead Fulfill with a Container whose APID is in the corresponding ActiveAPID element if the DSP possesses the Container.

The DSP NEED NOT observe the constraints in any applicable AssetRestriction when fulfilling a Container. However, note that there might be contractual requirements imposed by Content Providers that require a DSP to comply with fulfillment constraints, which may or may not be the same as the constraints in any applicable AssetRestriction.

If the DSP cannot fulfill a request due to the APID being invalid or due to the ALID being subject to a Fulfillment restriction, the DSP SHALL return a HTTP 403 error response as accordance with Section 11.1.7.

11.1.7 Fulfillment Error Handling

A DSP may not be able to fulfill a download request to a DCC for the following reasons:

- Payment Required: The Retailer has a policy requiring an additional payment to be made to download content once the DECE fulfillment obligations for a free download have been met.
- APID Recalled: The Content Provider has recalled the APID. See Section 7.4.5.
- Fulfillment Restricted: The DSP is unable to fulfill the request due to the ALID being subject to a Download restriction for the relevant Region. See Section 11.1.6.
- Access Control: The Retailer does not permit the download to occur. For example, the Content was not purchased. See Section 11.1.4.8.
- Any other error where the DSP prohibits the request from being fulfilled.

Whenever the DSP prohibits a request from being fulfilled, the DSP SHALL respond to the request with an HTTP 403 Forbidden response ([HTTP] Section 6 and 10.4.4), and the DSP SHALL return a ResponseError-type XML structure conforming to [DCoord] Section 18.1, using values for the ErrorID attribute and the ErrorLink element as described below.

11.1.7.1 Error ID

All of the Error IDs are prefixed with urn:dece:errorid:org:dece:

Error ID	Description	Code
FulfillmentPaymentRequired	A payment is required to download the file.	403
FulfillmentAPIDRecalled	The content file has been recalled.	403
FulfillmentRestricted	Downloads of the content are restricted at this time.	403
FulfillmentInvalidAccess	The retailer does not allow access to the content file.	403
FulfillmentProhibited	Any other error requiring an HTTP 403 response.	403

11.1.7.2 ErrorLink

The `ErrorLink` element of the ResponseError-type ([DCoord] Section 18.1) should be the URL of a web page describing the error.

12 Licensing Content

The first time Content is played on a DECE Device, the DRM Client on the Device must acquire a native DRM license for the Content. The license authorizes the DRM Client to permit playback of the Content, and provides the necessary keys for Content decryption. The process of a DRM Client obtaining a license is called *license acquisition*.

The DECE Device SHALL be joined to a DECE Domain prior to attempting to acquire a license. Device Joining is described in Section 7.3.3.

12.1 License Cached in the Device or Container

When a DECE Device attempts to play Content, the Device first determines if it already has a license for the Content accessible to its DRM Client. How a DRM system does this is out of the scope of the DECE. It may check a local license cache maintained by the DRM system on the device (#1 in Figure 25), or contact its License Manager operated by the DSP if it knows the address (#2 in Figure 25). (How to obtain the address of the License Manager is covered later in Section 12.2.)

If a valid license is not found, the Device must also check for a valid license cached in the Container (#3 in Figure 25). How licenses are stored in the Container is described in [DDevice] Section 7.2.5. DECE requires this to support a user copying a Container to another DECE Device in the same domain via normal file system or other non-DRM enabled operations, and then taking the Device offline before playing the content and acquiring a license.

Note that the user experience of copying a Container to a Device, going offline, and then attempting playback will vary. Offline license acquisition will fail if the License has not been cached in the Container. Even if the Container had been played, if it had been played only by Devices with a different DRM than the target Device, a usable license will not have been cached in the Container.

The DECE Device checks the Container for a valid license prior to license acquisition.

If a license is obtained during license acquisition, the DECE Device will store the license in the Container as described in [DDevice] Section 7.2.5, replacing any older license as needed.

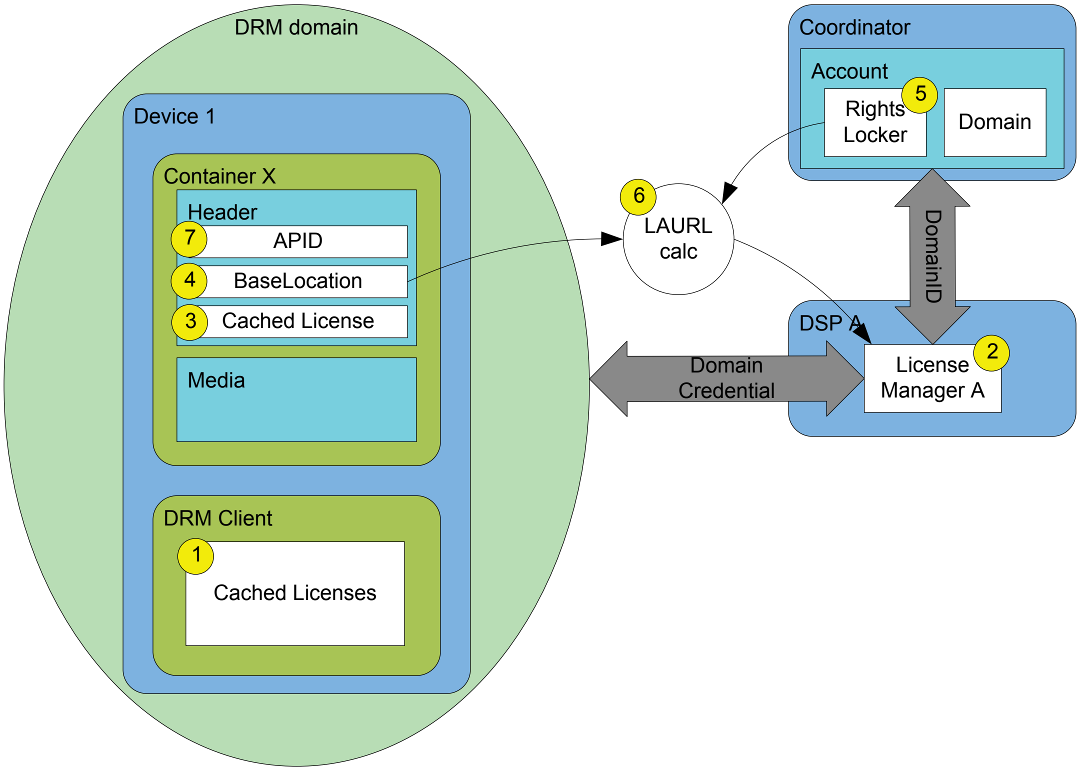


Figure 25 – License Acquisition (simplified)

12.2 Locating a License Manager

If the DRM Client does not have a valid license, it must determine the URL to contact the License Manager authorized to issue licenses for the Right owned by the Account. License Managers are not global; only the License Managers for a DSP operated on behalf of a Retailer who sold the Rights for the Content to the Account is obligated to issue licenses in the User's Domain.

Before DECE, a Retailer would package their content along with a License Acquisition URL used to locate the Retailer's License Manager. In that system, the content file could only be used with one Retailer and one DRM system.

DECE expands this concept to support multiple Retailers and DRM systems. It does this by:

- Providing a Base Location in the Container (#4 in Figure 25) to cache an association to a Retailer which is used to construct a URL to the License Manager.
- Storing the License Manager Location for the Retailer who sold the Right in the Rights Token in the Coordinator. (#5 in Figure 25.)

12.2.1 Base Location in the Container

The Base Location (#4 in Figure 25) is a box in the Container or in a DMP as defined in Section 8.3.1. It contains the Internet domain of the Retailer who sold or distributed the content, which can be used to construct the Retailer's License Acquisition Location (LALOC) as described in Section 8.3.4.

The Base Location is a cache of the Retailer location. It may be empty or otherwise invalid (e.g. pointing to a previous User's Retailer if the Container had been copied).

Normally, the Base Location is maintained by:

- A Retailer authorizes a DSP or Content Provider to set the Base Location when a Container is added to the DSP prior to distribution or fulfillment. This requirement is specified in Section 8.3.2.2.
- A DECE Device updates a Base Location if it was changed by a successful license acquisition. This requirement is specified in the DECE Device Specification [DDevice] Section 5.2.3.

If the License Manager cannot be located via the Base Location, or if it returns an error, then the LALOC is derived from the Coordinator as described next in Section 12.2.2.

The DECE Device (which includes the DRM Client) will attempt to locate the License Manager via the Base Location in the Container prior to obtaining the address from the Coordinator.

12.2.2 License Acquisition Location from the Coordinator

If the License Manager address cannot be determined from the Container, it can be derived from the Coordinator (#5 in Figure 25). When a Retailer sells a right to Content, it must update the Rights Token in the User's Rights Locker as described in Section 10.1. One of the fields in the Rights Token the Retailer must set is the `LicenseAcqBaseLoc` element containing the Base Location used to calculate the address of the appropriate DSP's License Manager. Section 10.1.1.5 describes the Retailer requirement to set this element, and Section 8.3.4 defines how to calculate the License Acquisition Location (LALOC) from the Base Location stored in the element.

The DECE Device Specification [DDevice] 7.2.4 describes how a DECE Device does a `RightsTokenGet` query to the Coordinator to get the Rights Token.

12.3 License Acquisition

The URL to contact the License Manager is constructed from the LALOC, and is called the License Acquisition URL (LAURL). The LALOC contains the hostname portion of the URL, regardless of whether it was calculated from the Container `BaseLocation` or from the Coordinator Rights Token `LicenseAcqBaseLoc` element. The License Acquisition URL is calculated from the LALOC to obtain the full URL of the native DRM License Manager (#6 in Figure 25). License acquisition is initiated via a DRM license trigger retrieved from the DRM License Manager similar to the DRM join and leave triggers retrieved from the Coordinator. The license trigger is retrieved from the License Manager by appending the APID as path parameter to the LALOC as follows:

```
LAURL ::= "http://" <LALOC> "/" <APID> ["?" <DRM specific parameters>]
```

Where

- <`LALOC`> License Acquisition Location (Section 8.3.4).
- <`APID`> Asset Physical Identifier as stored in the DCC (Section 8.3.1)
- `<DRM specific parameters>` DRM system specific parameters may be appended as URL parameters as required by the DRM system.

The `LAURL` MAY be percent-encoded if any parameters contain reserved characters in accordance with [URI]. The DSP SHALL accept both percent-encoded and non-percent-encoded forms in accordance with [URI].

For example: playready-license.xyzstore.uvvu.com/urn:dece:apid:org:mycompany:abcdefg:100?DRMspecificparameter=value

Once a License Acquisition URL is obtained, the DRM Client uses it to connect to its License Manager and retrieve a DRM license trigger via a HTTP GET. The license trigger is executed by the DRM Client to retrieve a DRM license for the APID.

12.4 Issuing a License

If the DRM License Manager doesn't have a valid license for the domain, the DSP must issue a license after determining if the User has rights to the Content.

When a Content Provider distributed Content to a DSP, the Content Provider provided the Containers, ALIDs, APIDs, ALID to APID mapping, and the Content Encryption Keys (CEKs) along with any other information needed to generate licenses. See Section 9.1.5.3.

The DSP MAY use information stored from the Retailer when the User purchased the Content (see Section 10.1.2) to determine what rights the User has for the Content. If the DSP relies upon purchase information from the Retailer, the DSP SHALL only cache the information regarding the User's purchase as specified by the DSP_PURCHASE_INFO_CACHE_LIMIT parameter (see Section 16). If it does not use purchase information from the Retailer, or if the DSP_PURCHASE_INFO_CACHE_LIMIT time has expired, the following requirement for the DSP to do a RightsTokenGet Coordinator query applies.

When licensing occurs for a Container within a DMP, the licensing is still based on the Media Presentation. However, rather than using the PresentationID for that Presentation, the APID of the DCC containing the Primary Video Track (as defined in [DDevice], Section 11.1) is used. This makes licensing of Presentations within a DMP and licensing of multi-track DCCs equivalent. Note that APID within the DMP being licensed might be a multi-track DCCs, or a single track DCCs containing a video track.

The DSP is responsible for ensuring the APID or Presentation ID is valid and the ALID is not subject to Window restricting licensing. See Section 12.4.1 below.

The DSP SHALL do a RightsTokenGet Coordinator query [DCoord] Section 7.1.4 if it cannot otherwise determine if the User has a Right to the Content. This query can be done by Presentation ID, APID, or ALID.

If the User does not have a valid Rights Token for a Right the DSP SHALL NOT create a license for that Right.

If the User has a valid Rights Token, the DSP SHALL create the license by:

- Setting the DRM license fields as required by the Content Provider and DRM for the Media Profile corresponding to the Right.
- Looking up the CEKs for the APID and setting the DRM license key accordingly.

The new license must be returned to the DRM Client, successfully completing the license acquisition.

The DECE Device must update the DRM-specific license in the Container with the new license upon a successful license acquisition. See [DDevice], Section 7.2.5.

12.4.1 Licensing Restriction Windows and Recalled APIDs

Content Providers may occasionally need to recall or replace Physical Assets, or specify time periods where licensing Content may be restricted, as described in Section 7.4.5.

The DSP SHALL NOT License Containers whose APID is listed as Recalled in the associated LogicalAsset if the LicensingAllowed attribute is not set to ‘true’. See [DCoord] Section 6.5.2.4.

A Container can be licensed indirectly by providing a license for a Media Presentation that references the APID. Therefore, Licensing a Presentation that includes a Container is equivalent to Licensing that Container. Furthermore, as licensing is not currently performed against individual audio tracks, so the DSP has no knowledge whether it is licensing a particular audio track. The following requirement roots from this condition.

The DSP SHOULD NOT enforce LicensingAllowed rules when Licensing DCCs with exactly one audio track.

The DSP NEED NOT observe the constraints in any applicable AssetRestriction when licensing a Container. However, note that there might be contractual requirements imposed by Content Providers that require a DSP to comply with licensing constraints, which may or may not be the same as the constraints in any applicable AssetRestriction.

12.5 Examples

12.5.1 Container Copied to DECE Device in same Account with same DRM

If the Container was played on the initial DECE Device, it will have a license cached in the Digital CFF Container associated with the DRM ID.

When the Container is copied to another DECE Device joined to the same Account, if the new DECE Device uses the same DRM the Container should be playable without requiring Internet connectivity. This works because Approved DRMs are domain-based DRMs, and the license stored in the Container will work on all DECE Devices joined to the same domain.

12.5.2 Container Copied to DECE Device in same Account with different DRM

This example assumes the Container was never played on a DECE Device with the same DRM. Otherwise it is the same case as Section 12.5.1.

In this case there will not be a valid license cached on the Device or in the Container. See Section 12.1.

The BaseLocation will be valid as rights to the Container had already been purchased by a User in the same Account, assuming the Container had been previously played previously in the DECE Account or the DSP had set the BaseLocation during fulfillment.

The LALOC will be calculated from the BaseLocation as described in Section 12.2.1, and the Retailer's DSP for the new DRM will be contacted to acquire a license.

If the DSP's License Manager does not already have a license for the Content and DRM domain, it will query the Rights Locker in the Coordinator to obtain the Rights Token, and create a license.

The license will be stored in the Container for the DRM ID allowing the Content to be played.

12.5.3 Container Copied to DECE Device Outside of the Account

In this case any licenses stored in the Container will be invalid. A DRM license is tied to the DRM Domain Credentials of the native DRM, which is in turn tied to the DECE Account that purchased the Content.

In most cases the BaseLocation will be invalid. In this case the DECE Device will query the Coordinator for a Rights Token, which will fail if the new User had not previously purchased the Content.

If the BaseLocation is valid, which could occur if the new User had a valid account with the same Retailer, when the DSP tries to license the Content it will fail when it queries the Coordinator for a Rights Token.

This will require the new User to purchase rights to the Content before it can be played.

13 Playing Content

13.1 Playing from a Digital CFF Container

A DECE Device plays media from a Digital CFF Container as described in DECE Device Specification [DDevice], Section 8.

A Digital CFF Container includes Required Metadata and may include Optional Metadata as described in 8.3. Included in these metadata are descriptions of the content within the Container that can be used for informative purposes (e.g., displaying information about the content) or functionally (e.g., implementing parental controls based on ratings in the Movie Metadata).

Assuming the Container meets the requirement for play, such as it is compatible with the profile of the Device and parental controls are appropriately applied, the content is decrypted and decoded on the Device and presented. Presentation may be on a built-in display, or through an external interface such as HDMI.

During the playback process, the Device and the DRM Client are responsible for protecting the content and the keys associated with decrypting the content. The DRM Client decrypts the Content (described in [DMedia] Section 3) and enforces Output Controls as specified by the DRM Client compliance rules.

Playback may include trick play; that is the ability to perform actions such as fast forward and rewind, depending on the Device’s capabilities.

If a Device has the ability to play a Container while it is being downloaded (Progressive Download) it may do so.

If a Container has more than one audio track, the Device offers the capabilities to select which track is played.

If a Container has one or more subtitle tracks, the Device offers the capability to select a subtitle track.

For situations where a User wishes to play specific tracks but not save them, a Device can offer the option of streaming some tracks while playing other tracks from the Container. This mechanism is called Late Bound Common Streaming as it uses the mechanisms of Common Streaming to obtain track files that are Late Bound with Container tracks during playback. Examples where this might be relevant include playback of a director’s commentary, or alternate video. Note that this creates additional possibilities for interactive Content.

Late Bound Common Streaming uses the Domain model for DRM. That is, the licensing and playback content security are the same for the streamed tracks as it is for the Container-based tracks.

13.2 Streaming from LASP

Before a LASP can stream content, it must first authenticate with the Coordinator. A Linked LASP does this by binding to a DECE Account as described in Section 7.1.2.4, while a Dynamic LASP can either establish short-term or long-term access as described in Section 7.1.2.3. An authentication operation, with an optional Account binding step, is required to get a Security Token from the Coordinator allowing viewing of the Rights Locker and streaming to be managed.

The LASP uses the Coordinator APIs to view the Rights Locker (see [DCoord] Section 7) and provide an interface for the User to select content to stream.

Content Providers may occasionally need to specify regions and time periods where streaming Content may be restricted as described in Section 7.4.5.

If a LASP is authorized to provide a Stream based on `AssentStreamAllowed` being true in the `LogicalAsset` resource in the Coordinator (see [DCoord] Section 6.5), the LASP SHALL NOT Stream Content in a time period or region restricted by any applicable AssetRestriction.

If a LASP is not relying on AssentStreamAllowed being true, the LASP NEED NOT observe the constraints in applicable AssetRestriction elements. However, note that there might be contractual requirements imposed by Content Providers that require a LASP to comply with streaming constraints, which may or may not be the same as the constraints in any applicable AssetRestriction.

The LASP SHALL NOT stream from Containers whose APID is listed as Recalled in the associated `LogicalAsset` except in the case where the LASP has a contractual agreement with the Content Provider allowing otherwise. See [DCoord] Section 6.5.2.4.

Before the LASP can stream the Content, the LASP SHALL ensure the Rights Locker has a valid corresponding Rights Token with the `CanStream` element set to “true” for the Profile to be streamed.

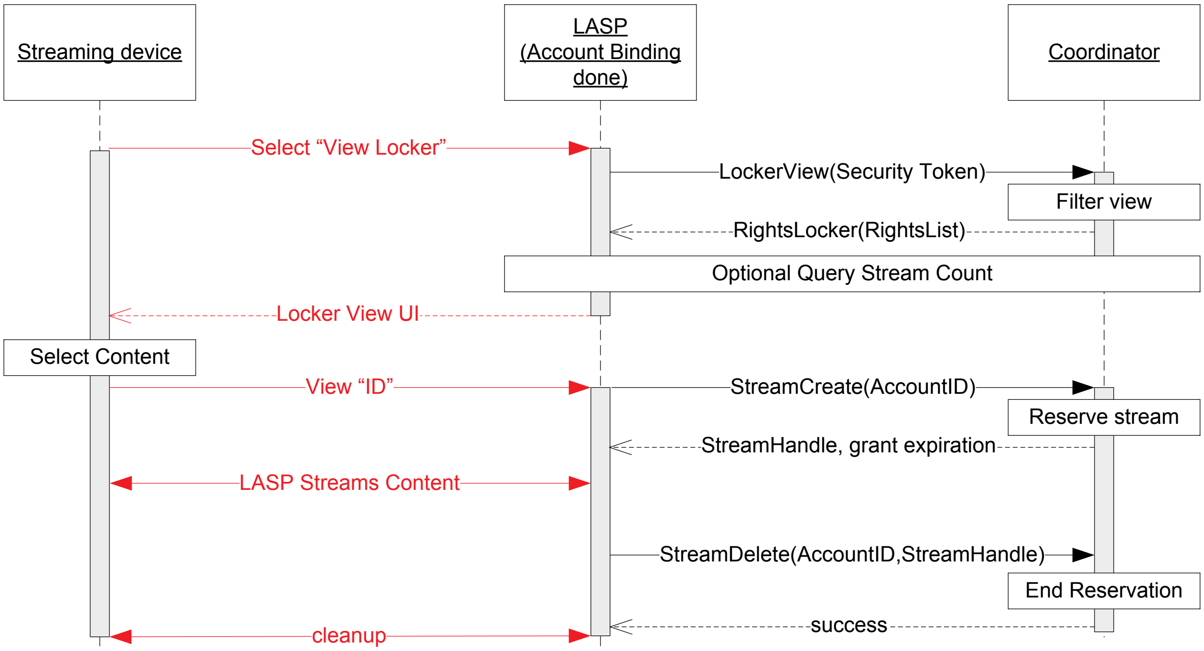


Figure 26 – LASP Streaming Flow

13.2.1View Filtering

A Dynamic LASP is bound to a User (Section 7.1.2.3), which is known to the Coordinator via the Security Token. The Coordinator will filter the User's RightsList to only show Content viewable by the User, meeting any Parental Control requirements.

A Linked LASP is bound to a DECE Account, and does not necessarily know who the User is. (For example, a Linked LASP could be a family television.) All available Rights will be returned in the RightsList for the Account. The streaming device may implement its own Parental Control system, in which case it should filter the RightsList on the device. How the device does this is out of the scope of the DECE.

13.2.2Stream Counts and Reservation

The Coordinator keeps track of how many streams are active for an Account, and enforces a maximum limit. (See LASP_SESSION_LIMIT in Section 16.)

A LASP SHALL adhere to the streaming API specified in the [DCoord] Section 11.

A LASP MAY request a list of active streams for the account using the StreamList Coordinator query. The LASP may display this list to the User to enable them to terminate conflicting streams.

A LASP MAY determine how many streams are available by reading the AvailableStreams attribute of the StreamList Coordinator query. See [DCoord] Section 11.1.2 for more information.

A LASP SHALL POST StreamCreate to the Coordinator before it can stream content. The LASP is not required to wait for a response before starting the stream. See [DCoord] 11.1.1.

StreamCreate updates the stream count for the Account. A stream can only be reserved for a limited amount of time so that reservations will be released if a User stops watching Content without terminating the stream (e.g. leaves the stream paused and turns off the display).

The Stream reservation expiration limit is subject to changes in policy. Streams can be renewed if the time limit is exceeded via the StreamRenew call.

13.2.3 Common Streaming

Common Streaming refers to the ability to generate Common Streaming Format (CSF) in accordance with [DMedia] and stream in accordance with [DStream] and [DDevice], Section x.x.

Informative information on publishing for Common Streaming is included in [DPublisher], Section x.x.

Common Streaming uses industry standards to simplify implementation. Common Streaming is based on Common File Format (CFF) ad defined in [DMedia]. Consequently, playback of CSF is close to CFF. The streaming protocol is based on MPEG DASH [DASH]. Other Common Streaming functions, such as Licensing, are existing DECE mechanisms.

Nodes that support Common Streaming SHALL support Common Streaming requirements in accordance with DECE Specifications. For avoidance of doubt, this does not preclude any other streaming mechanisms, including those that may differ from Common Streaming only slightly. The intent here is to clearly differentiate between implementations that are compliant and those that are not.

Nodes that stream in accordance with Common Streaming SHALL comply with [DStream].

Requirements for Devices can be found in [DDevice].

14 Discrete Media Rights

See [DDiscrete] for information about Discrete Media Rights.

15 Superdistribution

Superdistribution is any means of distributing DCCs in advance of the recipient purchasing a Right to the DCC. This includes preloading DCCs on media or DECE Devices, sharing DCCs on download services or peer to peer networks, and copying a DCC from one DECE Device to another DECE Device in a different Account. Before Superdistributed Content can be accessed (decrypted), a User must obtain the associated Right.

Superdistribution allows and encourages encrypted Containers to be distributed freely while the Content owner retains control over the ability to use and modify the product. Superdistribution is a highly efficient means of distribution because distribution is not impeded by any barriers and anyone can become a distributor. Superdistributed Content generally requires a license that the User must purchase before being able to play the Content.

15.1 Preparing a Container for Superdistribution

If a Content Provider or Retailer desires to Superdistribute a Container, the Content Provider or Retailer SHALL prepare the Container by ensuring the BasePurlLocation in the Container is set to the Organization Name of the preferred Retailer as described in Section 8.3.3.

A Content Provider or Retailer SHALL also set the `BaseLocation` in a Container intended to be Superdistributed as described in Section 8.3.2.

Setting the `BasePurlLocation` enables a User to purchase a Right to the Content from the preferred Retailer who enabled the Superdistribution. However, it does not guarantee that the User or Device will purchase the Right from the preferred Retailer.

15.2 Licensing Superdistributed Content

If the Content Provider chooses to encrypt the Container, it can be freely Superdistributed without concern since the Content cannot be accessed without a User licensing the Content (in order to obtain the key required to decrypt the Container).

15.2.1 Initial Licensing of Superdistributed Content

When a Superdistributed Container is attempted to be played for the first time, the Device will not have a License for the Container and will attempt License Acquisition as described in Section 12 first trying the license acquisition URL derived from `BaseLocation`, and when that fails the Device will do a

RightsTokenGet query to determine the authoritative license acquisition URL. However, as the User has not yet purchased a Right to the Content, License Acquisition will fail when no Rights Token is found.

The Device should then prompt the User to purchase a Right to the Content. It may use the BasePurlLocation to locate the preferred Retailer's web page for the Container's APID, or it may use another Retailer preferred by the User or the Device as described in Section 10.2. The Retailer's API or web interfaces used to purchase Rights are out of DECE scope.

When the User purchases a Right to the Content, the Retailer will update the Coordinator by calling RightsTokenCreate to add a Rights Token to the User's Rights Locker and update the DSP using a private communication as described in Section 10.1.

License Acquisition can then proceed. If the Right was purchased from a different Retailer than specified by the BasePurlLocation, the Device will locate the License Manager from the Rights Locker in the Coordinator as described in Section 12.2.2. Otherwise, the Device will use BaseLocation to create a License Acquisition URL to locate the License Manager as described in Section 12.2.1. As the Right was purchased for the User's Account, License Acquisition should succeed and Content playback should be allowed.

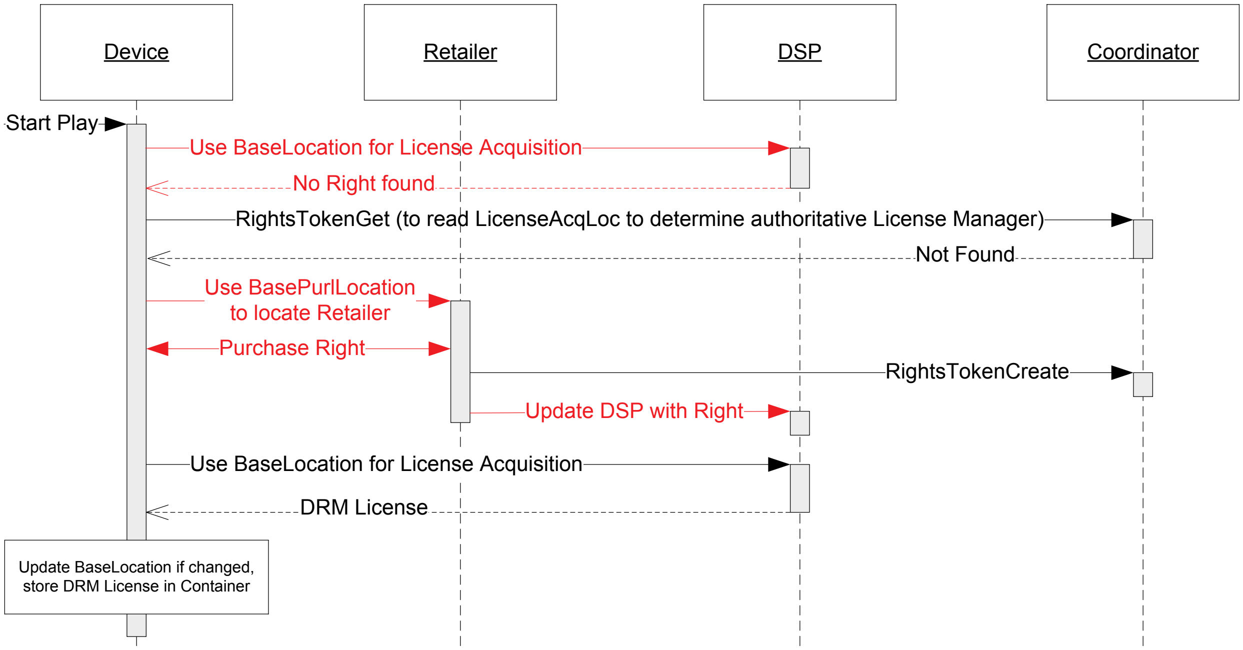

Device
Retailer
DSP
Coordinator
Start Play
Use BaseLocation for License Acquisition
No Right found
RightsTokenGet (to read LicenseAcqLoc to determine authoritative License Manager)
Not Found
Use BasePurlLocation to locate Retailer
Purchase Right
RightsTokenCreate
Update DSP with Right
Use BaseLocation for License Acquisition
DRM License
Update BaseLocation if changed, store DRM License in Container

Figure 27 – Superdistributed Container License Acquisition

Note that Figure 27 is simplified:

- authentication is omitted,

- whether the Device uses a Browser or a web service API to communicate with the Retailer is omitted as it is out of DECE scope,

- it omits calls by the DSP to determine licensing windows (Section 12.4.1) and to verify the Rights Token validity if the information from the Retailer is insufficient,

- the case where the BaseLocation is invalid is not shown during the final License Acquisition; in that case the Device would do a RightsTokenGet query to obtain the LicenseAcqBaseLoc (Section 12.2.2).

15.2.2 Licensing of Copied Content

Once a Container has been played by a User on a Device, it should have the BaseLocation set to the Retailer the Right was obtained from, and a native DRM license for the User's Domain may be stored in the Container as described in Section 12.

If the Container is copied to another Device joined to the same Account Domain (as in another Device in the same household), either the cached license in the Container can be used (as it is valid for a Device in the same Domain) or License Acquisition will succeed as the Right will still be in the Account's Rights Locker, regardless of which native DRM the Device uses.

However, if the Container is copied to a Device that is not joined to the Account Domain, such as to a friend's Device, License Acquisition will fail and a new Right will have to be purchased by the new User. This is because:

- All the native DRM licenses cached in the Container are bound to the specific Domain (actually to the native DRM Domain which is potentially even more restrictive) and the DRM systems will not allow the license to be used to play Content outside of the Domain.

- As the new User is in a different Domain, the License Manager pointed to by the BaseLocation in the Container will not find a Right for the Content in either the License Manager or in the Coordinator's Rights Locker for the User, and will be unable to issue a License.

The result is the same as for the initial Licensing of Superdistributed Content described above in Figure 27. The Device should prompt the User to purchase a Right to the Content using the BasePurlLocation or an alternative preferred Retailer. When a Right is purchased, the new User's Rights Locker will be updated, and License Acquisition will succeed and the Container can be played on the User's Device.

16 Appendix A: Ecosystem Parameters

Parameter	User Limits	Description
ACCOUNT_USER_LIMIT	6	The maximum number of concurrent Users per Account.
DEVICE_DOMAIN_FLIPPING_LIMIT	3 times per 90 days	The maximum number of times a Device is allowed to rejoin a previous Domain after an intervening join to a different Domain.
DISCRETE_MEDIA_LIMIT	1	The maximum number of allowed discrete media allowed per associated Rights Token.
DOMAIN_DEVICE_LIMIT	12	The maximum number of Devices concurrently joined to a Domain.
DSP_PURCHASE_INFO_CACHE_LIMIT	6 hours	The maximum time purchase information can be stored by a DSP for licensing Content without requiring a Coordinator call to verify the Rights Token.
DYNAMIC_LASP_AUTHENTICATION_DURATION	24 hours	The maximum time between user re-authentication to the LASP.
LASP_SESSION_LIMIT	12	The maximum number of concurrent LASP Sessions per Account (i.e., maximum number of concurrent streams for one Account). May be changed by DECE at any time, but will be at least LASP_SESSION_MINIMUM_LIMIT.
LASP_SESSION_MINIMUM_LIMIT	3	The minimum value for LASP_SESSION_LIMIT.
LINK_LASP_ACCOUNT_FLIPPING_LIMIT	2 times per 365 days	The maximum number of times a LLASP account is allowed to re-bind to a previous Account after an intervening bind to a different LLASP.
DYNAMIC_LASP_PERSISTENT_USER_FLIPPING_LIMIT	2 times per 365 days	The maximum number of times a DLASP account is allowed to re-bind to a previous User, for streaming in Persistent User-bound Mode, after an intervening bind to a different LASP. This limit is not currently enforced.
LINK_LASP_ACCOUNT_LIMIT	2	The maximum number of active bindings to any one Account from a LLASP.
DYNAMIC_LASP_PERSISTENT ACCOUNT_LIMIT	1	The maximum number of active bindings to any one User from a DLASP for use in Persistent User-bound Mode.
UNVERIFIED_DEVICE_REPLACEMENT_LIMIT	2 times per 365 days	The maximum number of joins available because of unverified Device removals from a Domain in a defined period.

Table 28 – Ecosystem Parameters

17 Appendix B: Approved DRM List

The following table lists the DRMs that have passed the DECE DRM approval process and are authorized to decrypt Content. The table contains the following columns:

DRM system An informal descriptive name of the DRM.

DRM name The <DRM name> used to identify the DRM when creating DECE DRM Identifiers as defined in Section 5.4.1.

DRM version An identifier representing a specific system version of the DRM used for the <DRM version> of a DRM Identifier as defined in Section 5.4.1.

UUID The DRM universally unique identifier used to identify the DRM in DRM-specific boxes in the CFF. The binary representations of all UUIDs are in big-endian format (the most significant byte is ordered first with the remaining bytes ordered by decreasing degree of significance).

DRM system	DRM name	DRM version	UUID
CMLA	cmlauv	1.0	A2 B5 56 80 6F 43 11 E0 9A 3F 00 02 A5 D5 C5 1B (binary byte array field) "A2B55680-6F43-11E0-9A3F-0002A5D5C51B" (big endian string – Informative)
Microsoft PlayReady	playready	2.0.0	9A 04 F0 79 98 40 42 86 AB 92 E6 5B E0 88 5F 95 (binary byte array field) "79f0049a-4098-8642-ab92-e65be0885f95" (little endian string – Informative)

Table 29 – Approved DRM List

Additional information for implementing Approved DRMs, including contact information, is separately published by DECE, currently at http://uvvuwiki.com .

18 Appendix C: Approved Stream Protection Technology List

As described in Section 4.4.5, a LASP may use any Approved DRM or one of the following Approved Stream Protection Technologies to stream Content. Not all of the Approved DRMs may support streaming, so for completeness the Approved DRMs that do are included in the table below.

Licensing Authority	Technology	Video Format Resolution from [DMedia]	Restrictions
Adobe	Flash Access 2.0	SD, HD	
Cisco/SA	PowerKey	SD, HD	Closed Devices
Marlin	Marlin	SD, HD	
Microsoft	MediaRoom	SD, HD	Closed Devices
Microsoft	PlayReady	SD, HD	
Motorola	MediaCipher	SD, HD	Closed Devices
Motorola	SecureMedia	SD, HD	HD, see 18.1
Nagra	Media ACCESS CLK, ELK	SD, HD	Closed Devices
Nagra	MediaAccess PRM	SD, HD	HD, see 18.1
NDS	VideoGuard	SD, HD	Closed Devices
CMLA	CMLA-OMA DRM	SD, HD	
Rovi	DivX DRM Series 5	SD, HD	HD, see 18.1
Verimatrix	VCAS	SD, HD	HD, see 18.1
WideVine	WideVine Version 4.0	SD, HD	

Notes:

Licensing Authority refers to the manufacturer or other entity responsible for licensing the Stream Protection Technology.

Technology is the name, which may include a version, of the Stream Protection Technology that has been approved.

Media Resolution lists which profiles are approved for streaming by the Stream Protection Technology.

Restrictions are other limitations on the use of the Stream Protection Technology as described in Section 18.1.

18.1 Restrictions

Closed Devices

The stream protection technology is only approved for use to provisioned devices under the control and administration of a system operator and using a conditional access system. These devices must not support unrestricted addition of applications or playback of content.

DivX DRM Series 5

Approved for HD Content only on DivX Certified players that follow the robustness rules detailed in the DivX® Certified Test Kit (CTK) DivX Plus Streaming (DPS) Profile Requirements document version 1.0 section 2.12.

Motorola SecureMedia, Nagra Media PRM and Verimatrix VCAS

HD content only on Closed Devices or LASP Clients featuring secure boot, signed software image, scrambled RAM, platform-based decryption, hardware root of trust, and a secure, unique identity that specifies the device type and version.

If the LASP Client does not meet the requirements above, only SD resolution playback SHALL be allowed.

END