

DReaM-MMI: Alternate Mechanism to Deliver Rights



Vishy Swaminathan
Santa Clara, California
March 15-16, 2006

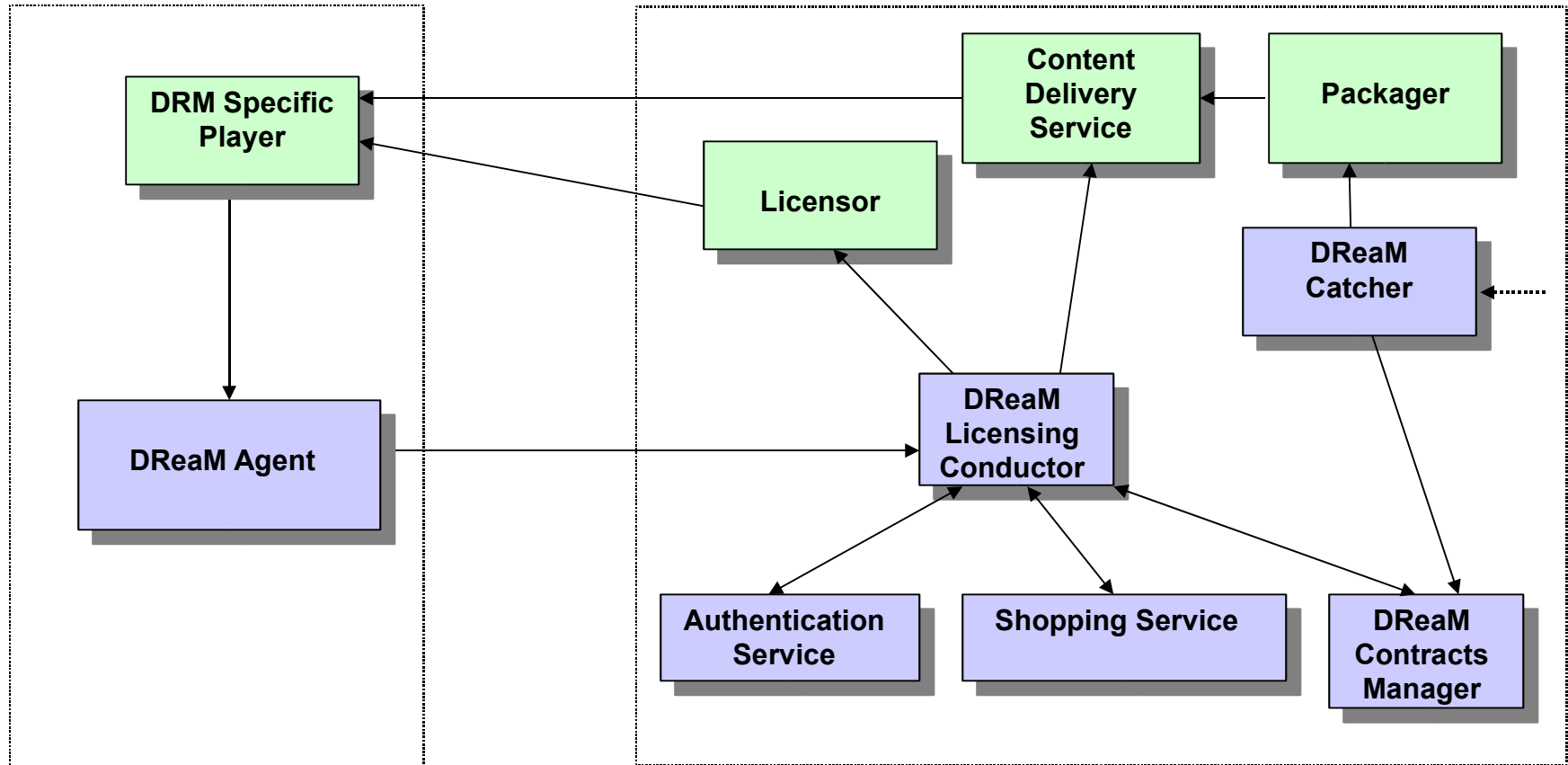
Agenda

- Introduction
- Scope and Requirements
- Design Premise
- Key Features
- Technical Details
- Summary

OMC's Project DReaM

- Supports Open Media Commons (OMC) vision of open source and royalty-free intellectual property
- Leverage Liberty Alliance to promote Identity & Role based licensing models, rather than device based licensing only
- Addresses both Conditional Access (CA) and Digital Rights Management (DRM) system models
- Addresses customer immediate multi-system interoperability requirements
- Independent of any specific content type
- A horizontal solution platform capable of supporting: Business, Content and Life vertical markets
- Exploration of RMS policy options beyond encryption (watermarking & tracking)

DReaM Components



DReaM Client Side

DReaM Service Side

Major Components of DReaM

- Content Catcher/Ingest (**DReaM Catcher**)
 - Import digital content
 - Import business rules related to content
- Content Packaging (**DReaM Packager**)
 - Content encrypted (AES encryption)
 - Key(s) for encrypted data are produced and then posted to Secure License Server (**DReaM Licensor**) Database
 - Protected content is posted to content delivery server
- Content Rights Server/Repository (**DReaM Contracts Manager**)
 - Repository of business rules
 - Business rules are interpreted here
- Content Distribution Server
 - Support for distribution of protected data
 - Optionally, protected keys are delivered “in-band” with protected content
- License Service
 - a) Primary point-of-contact with clients (**DReaM Licensing Conductor**)
 - High transaction rate
 - Application Server likely
 - b) Communicates with **DReaM Contracts Manager** to generate "offer" of price and terms
 - c) Authentication (user, device, SIM, Java Card, Liberty single sign-on, etc) (Sun Access Manager)
 - d) Payment & Fulfillment (e-Commerce) (BSS, OSS-J)
 - e) Generates license (**DReaM Licensor**)
 - Data keys protected for transport using (eg. ECC)
 - f) Deliver license (**DReaM Licensor**)
 - Communicates with Content Distribution Server for delivery of "in-band" license
 - "Out-of-band" licenses are delivered to users directly.
- Client side (**DReaM Agent**)
 - Ability to negotiate for licenses
 - Supports CA and DRM models
 - Secure key repository
- DRM specific player

3 Key Elements of DReaM

- Digital Rights Management Service (DReaM-MMI)
 - > Ability to manage rights for any type of content in various usage models
- Conditional Access (DReaM-CAS)
 - > Ability to deliver timeline dependent content to multiple consumers (IPTV, telemetry, surveillance)
 - > Includes content protection and distribution technologies
- Disintermediation (DReaM intraOPERAbility Framework)
 - > Separation of back-office services from players/consumption
 - > Authentication, Billing, Entitlement, Rights Management, Subscriber Services, Portal, Content Delivery (sometime proprietary)
 - > Interoperability with existing content protection technologies
 - > Client-side Player/Consumption, Licensing, Packaging
 - > Emphasis on Identity based licensing over device licensing

3 Key Elements of DReaM

- Digital Rights Management Service (DReaM-MMI)
 - > Ability to manage rights for any type of content in various usage models
- Conditional Access (DReaM-CAS)
 - > Ability to deliver timeline dependent content to multiple consumers (IPTV, telemetry, surveillance)
 - > Includes content protection and distribution technologies
- Disintermediation (DReaM intraOPERAbility Framework)
 - > Separation of back-office services from players/consumption
 - > Authentication, Billing, Entitlement, Rights Management, Subscriber Services, Portal, Content Delivery (sometime proprietary)
 - > Interoperability with existing content protection technologies
 - > Client-side Player/Consumption, Licensing, Packaging
 - > Emphasis on Identity based licensing over device licensing

DReaM-MMI

An alternate method for expressing and controlling rights using the Mother-May-I paradigm

DReaM-MMI : Requirements

- Mechanism to
 - > Request and obtain rights
 - > Release unused rights
 - > Aggregation of requests and Release
- Mobility – rights accessible for any client
- Support impulse buying
- Occasionally Disconnected Modes
- Support Tethered devices

DReaM-MMI – Design premise

- Rights are stored on the network and accessible to any network-ed client.
- Rights are Identity-based.
- Fine-grain rights released to the clients
- All clients are network-ed directly or through a proxy.

DReaM-MMI : Key features

- MMI protocol
 - > Extremely Simple
 - > Request/Response based
 - > Request and Release messages are granted or denied.
 - > Additional hints defined for optimization of future requests
- Bindings for https defined – other transports to follow later.
- Multiple profiles defined for different domains – extensible.

DReaM-MMI : Message

MMIMessage =
MMIRequest | MMIResponse

DReaM-MMI : Request

**MMIRequest =
MMIMessageType
IdentitySegment
[DeviceSegment]
RightsSegment
[SignatureSegment]**



DReaM-MMI : Request

**IdentitySegment =
AuthServiceID [AuthTkn]**

DReaM-MMI : Request

**IdentitySegment =
AuthServiceID [AuthTkn]**

**DeviceSegment =
[LocationId] [#DeviceId]**

**SignatureSegment = SigAlg
Signature**


DReaM-MMI : Rights Segment

**RightsSegment = ProfileId
1*MMIRightsRequestElement**



DReaM-MMI : Rights Element

```
MMIRightsRequestElement =  
  ReqElemId ( ( 1#ContentId  
  #ServiceId ) | 1#ServiceId ) )  
  1#VerbElement
```



At least one of
ContentId or
ServiceId

DReaM-MMI : Verb Element

VerbElement =
VerbElemId Verb [Count]
[Duration | Period]
[#VerbSpecificArgs]



DReaM-MMI : Response

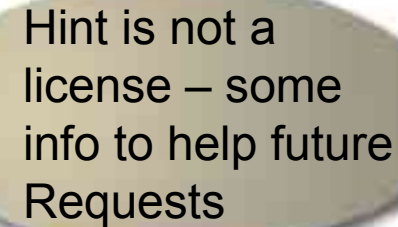
```
MMIRightsResponse =  
1#Status  
1#MMIRightsResponseElement  
[ RequestHashSegment ]  
[ ResponseId ]  
SignatureSegment
```

DReaM-MMI : Response Element

```
MMIRightsResponseElement =  
  ReqElemId Notification  
  [1#Hint] [Keys]  
  [1#RightsErrorStatus]
```

DReaM-MMI : Hints

```
Hint = HintIndexNum Label  
[1#ContentId]  
[1#VerbElement]
```



Hint is not a
license – some
info to help future
Requests

```
Label = ("CanDo"  
"CannotDo")
```

|

DReaM-MMI : Protocol Details

- MMIMessage = MMIRequest | MMIResponse)
- MMIRequest = MMIMessageType IdentitySegment
[DeviceSegment] RightsSegment
[SignatureSegment]
- IdentitySegment = AuthServiceID [AuthTkn]
- DeviceSegment = [LocationId] [#DeviceId]
- RightsSegment = ProfileId
1*MMIRightsRequestElement
- MMIRightsRequestElement = ReqElemId
((1#ContentId #ServiceId) | (1#ServiceId))
1#VerbElement

DReaM-MMI : Protocol Details

- `VerbElement = VerbElemId Verb [Count]
[Duration | Period] [#VerbSpecificArgs]`
- `SignatureSegment = SigAlg Signature`
- `MMIRightsResponse = 1#Status
1#<MMIRightsResponseElement>
[RequestHashSegment] [ResponseId]
SignatureSegment`
- `MMIRightsResponseElement = ReqElemId
Notification [1#Hint] [Keys]
[1#RightsErrorStatus]`
- `RightsErrorStatus = RightsCode`
- `Hint = HintIndexNum Label [1#ContentId]
[1#VerbElement]`
- `Label = ("CanDo" | "CannotDo")`

DReaM-MMI (Request)

- A sample URL doing an MMI request using HTTP GET would look like this:

```
http://greatcontent.org/myService?MMIVersion=1.0\  
&MMIMessageType=MMIRightsRequest\  
&Identity.UserId=Doc+Viewer&Identity.RoleId=LeadDocViewer\  
&Device.DeviceId=123456abc\  
&Rights.ProfileId=org.omc.dream.profiles.media\  
&Rights.ReqElem.Id=23\  
&Rights.23.ContentId=113%2C114%2C115\  
&Rights.23.Count=1&Rights.23.Verb=PLAY%2CRECORD
```

Here an MMI Rights Request for content ids 113, 114 and 115 is being sent out. Permission is requested to PLAY and RECORD the content once (Count=1).

DReaM-MMI (Response)

A sample response to an MMI request:

HTTP/1.1 OK

Content-type: text/plain

Content-length: nnnn

ResponseId=1003

MMIVersion=1.0

Status="RequestOK"

Response.ReqElemId=23

Response.23.Notification=granted

Response.23.Hint.HintIndexNum=1

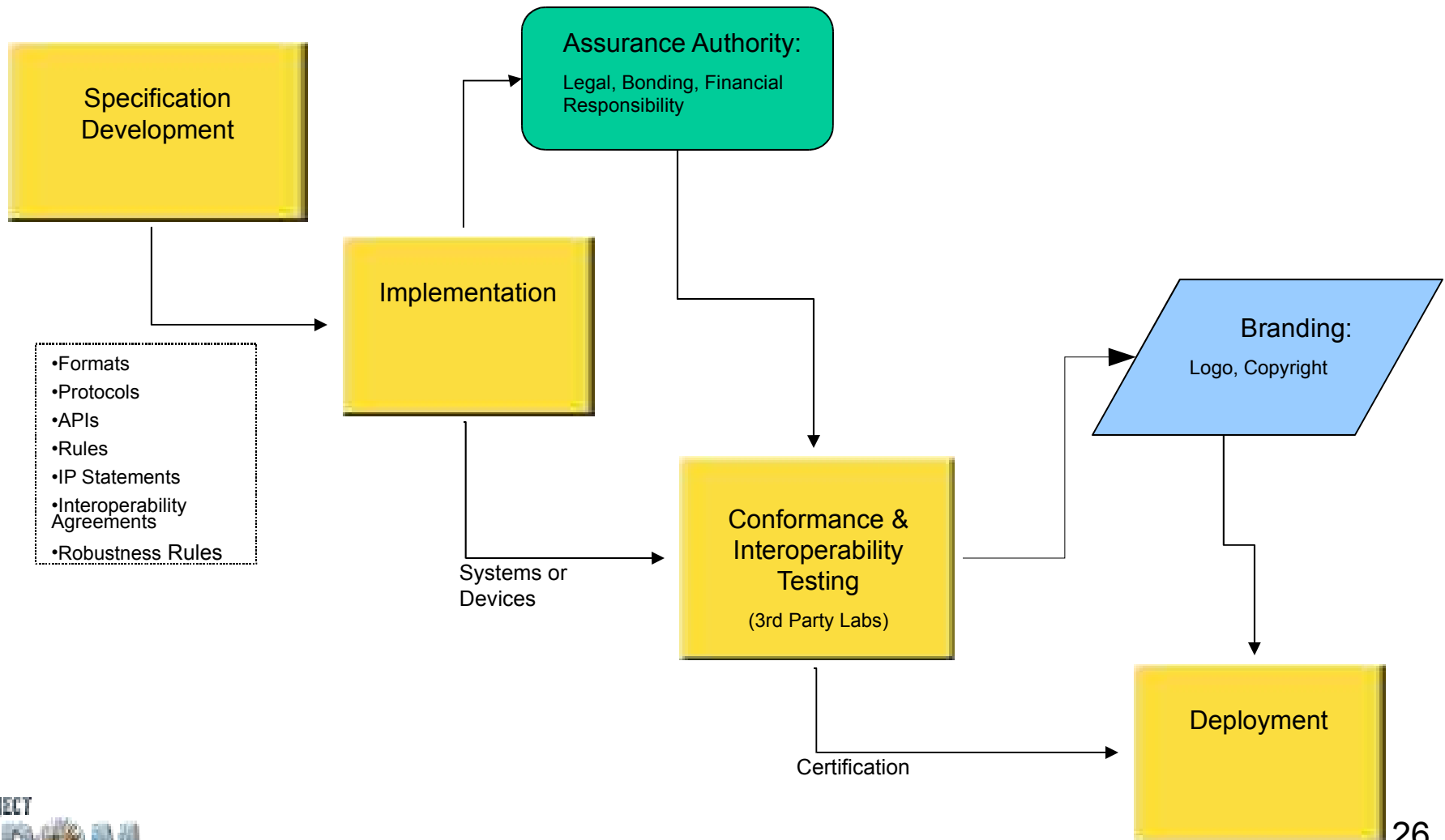
Response.23.Hint.1.Label=Allowed

Response.23.Hint.1.ContentIds=113,114,115

Response.23.Hint.1.Verb=PLAY,RECORD

Response.23.Hint.1.Count=29

The Process Ahead



Summary

- Introduction to DReaM-MMI
- Requirements
- Design Premise
- Key Features
- Details

Q&A

PROJECT
DREAM

An Open, End-to-End
Content Protection
Solution



<http://OpenMediaCommons.org>