DReaM: Secure End-to-End Interoperable DRM

Vishy Swaminathan,
Tom Jacobs,
Gerard Fernando

Sun Microsystems Inc.,
www.openmediacommons.org

TS-3326

Agenda

Introduction and Background to DReaM
Usage Models
Architecture
Content Protection Technologies
Benefit of Java Technologies in DReaM

Agenda

Introduction and Background to DReaM

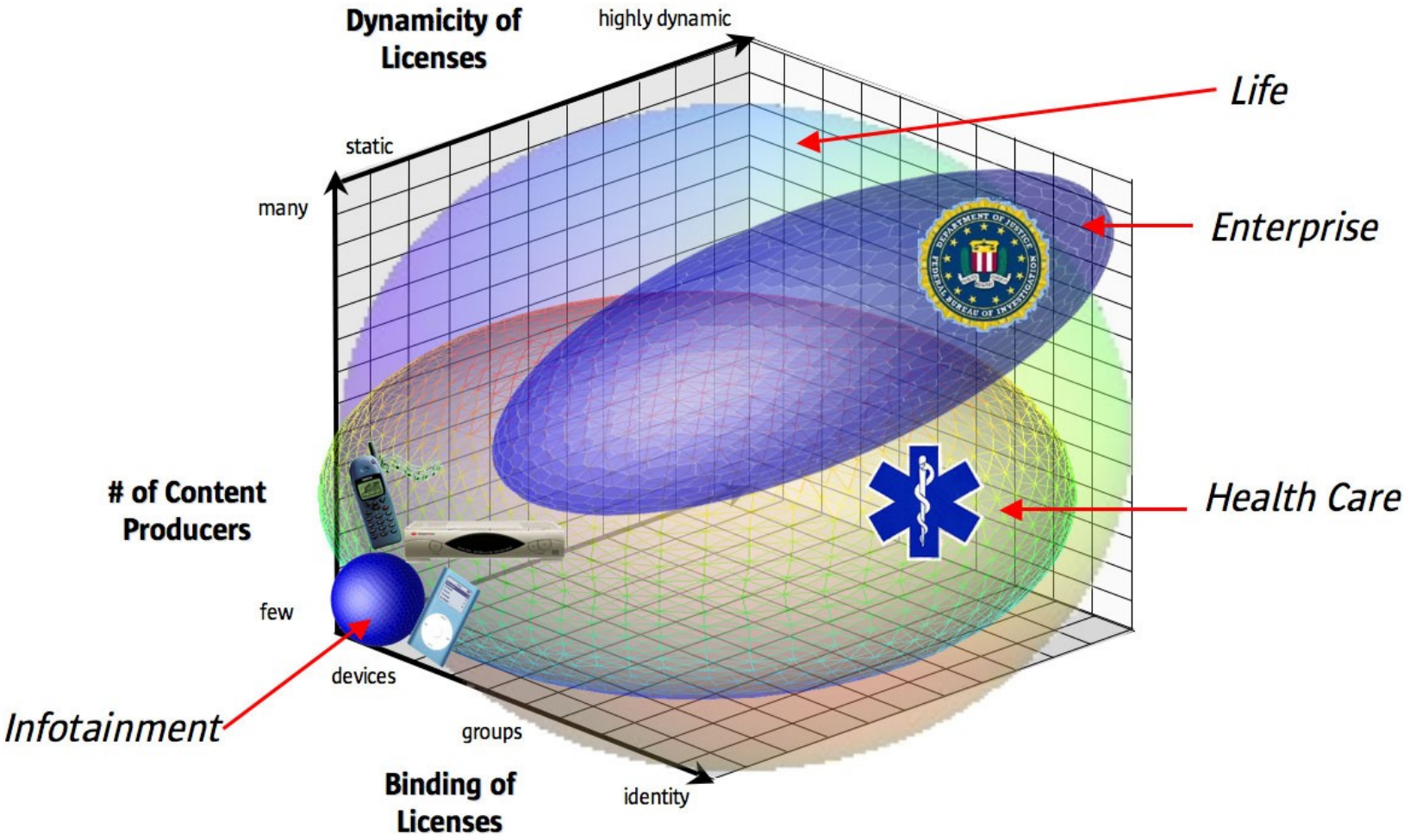
Usage Models

Architecture

Content Protection Technologies

Benefit of Java Technologies in DReaM

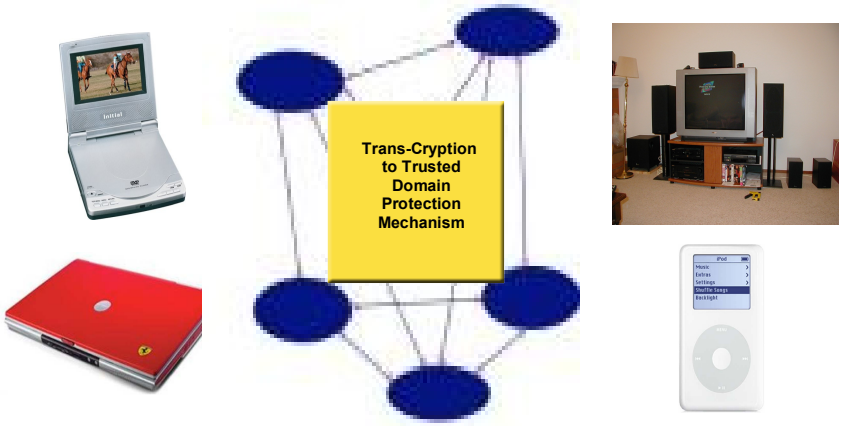
Extended View of DRM



Infotainment/Content Model

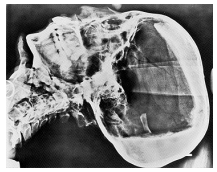
Trusted Client Devices

Trusted P2P Networks



Business Applications

Healthcare and Medical


Military



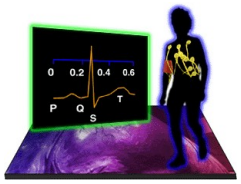

Business and Education


Finance



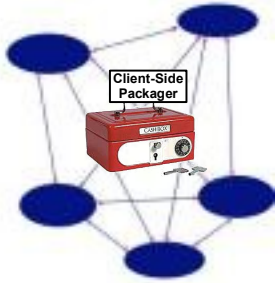
Life Model

Personal Information

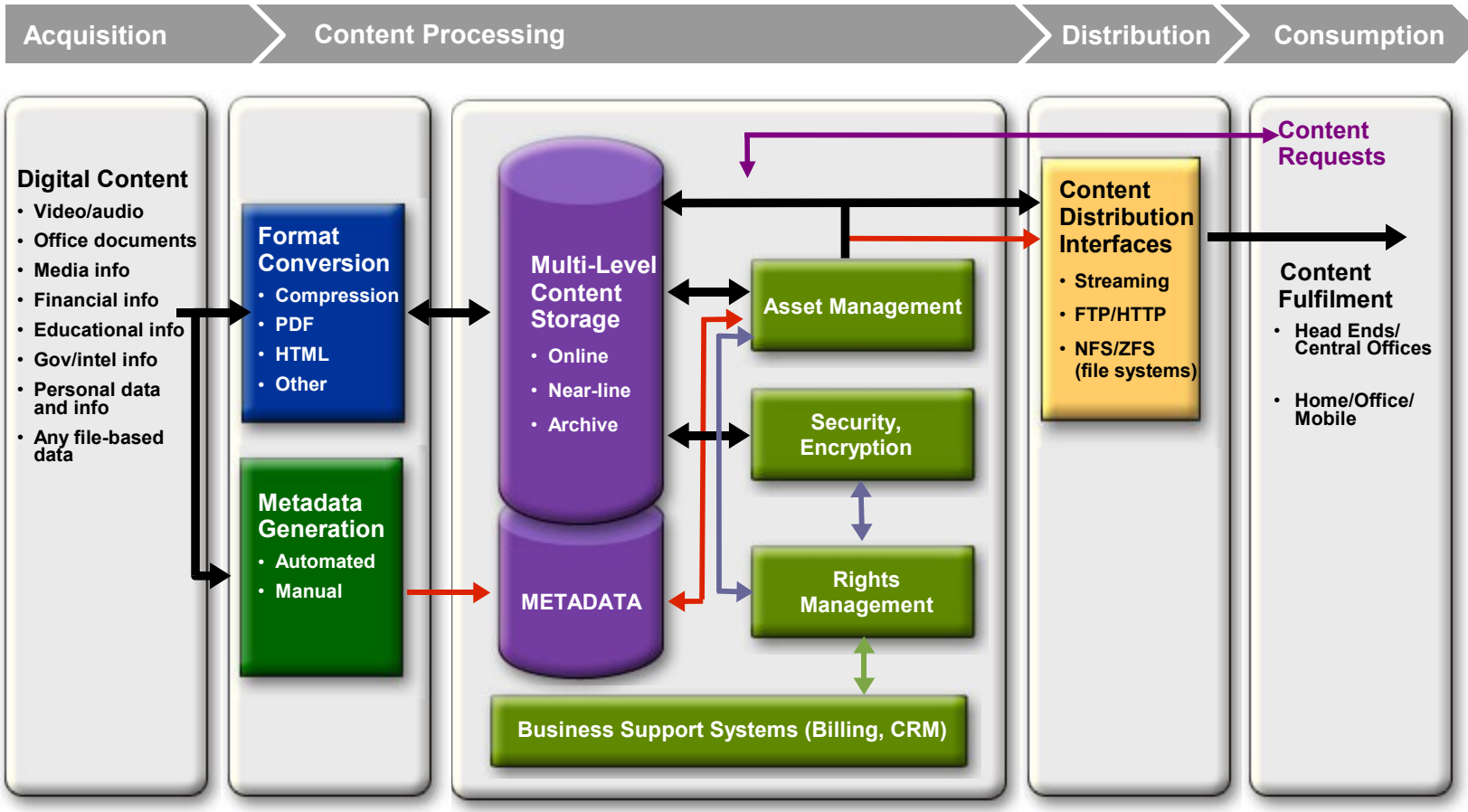


Personally Identifiable Info

Personally Generated Information/Content



DRM and Digital Asset Workflow



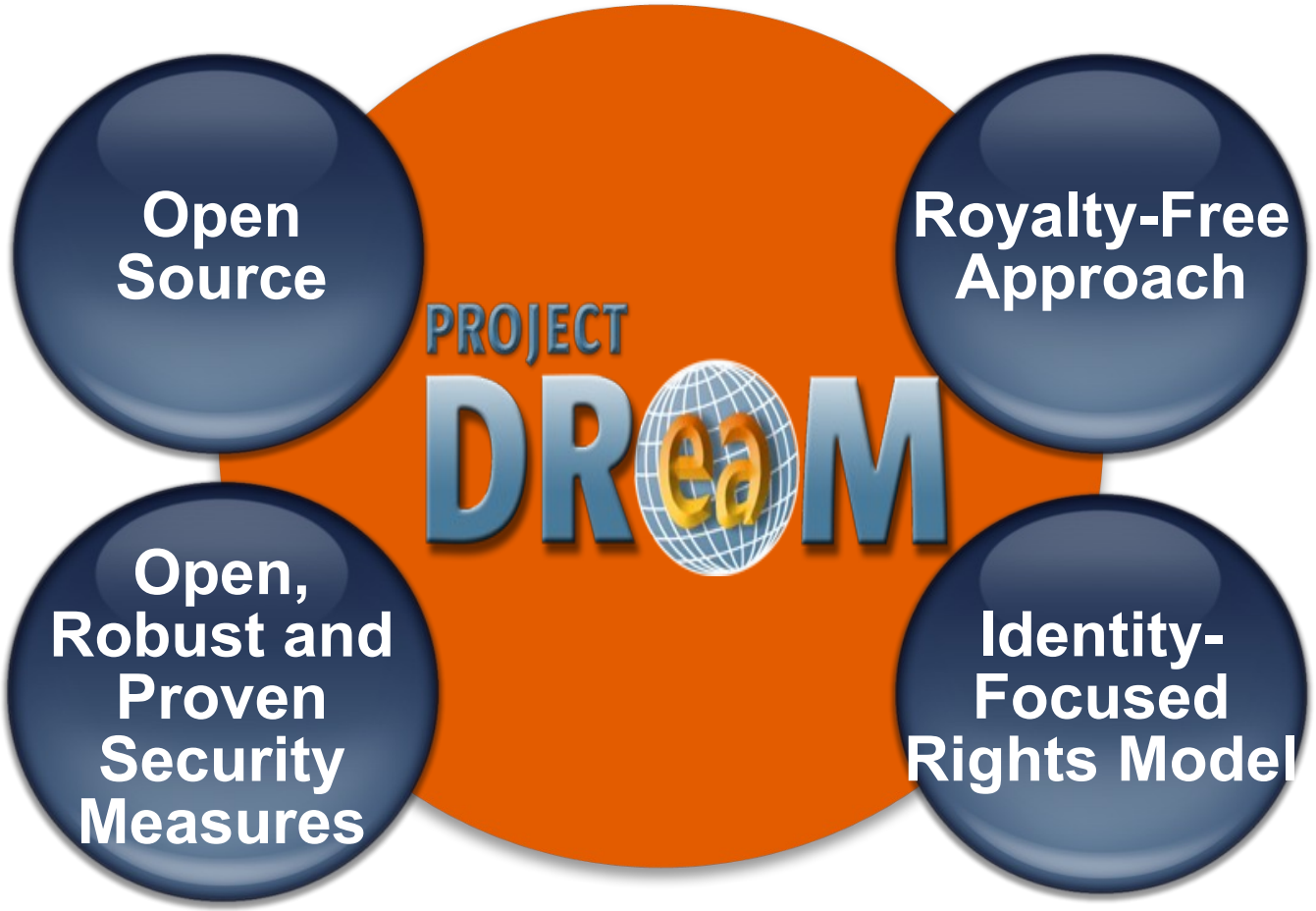
How Open Media Commons Works

- Enable a better model of DRM systems
- Remove barriers to innovation and use

- Open specifications, open source
- Royalty-free design premise

- Drive industry towards interoperability
- Demonstrate leadership in new applications
- Bring together a diverse spectrum of opinions

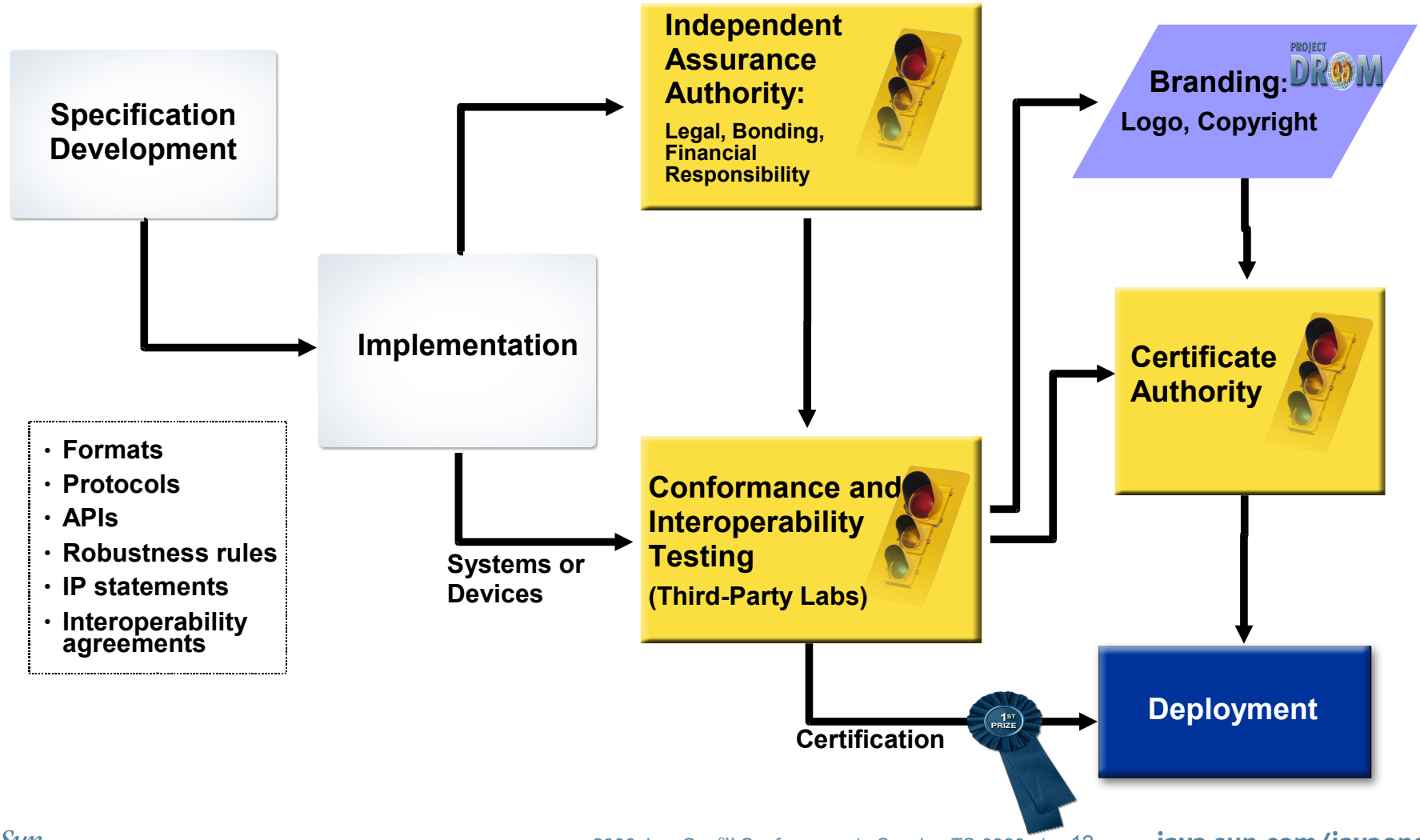
Why Open Media Commons and Project DReaM Is Needed



Digital Rights Management

- Rights management will be applicable to broad spectrum of content, devices, users and industry types
- Open source and DRM is **not** an oxymoron!
  - PKI, C&C, SOC, etc.
- Failed standards process
  - When RAND is not reasonable
- Open source + royalty-free = better
  - Open Media Commons (OMC)

The Process Ahead



Agenda

Introduction and Background to DReaM

Usage Models

Architecture

Content Protection Technologies

Benefit of Java Technologies in DReaM

Common Example Scenarios

- Ringtones (sale, forward locking)
- Commercial Music (subscription with timeout, n-copy burn to CD)
- Garage Bands (free to distribute, free to share, no encryption)
- Documents (view on screen, no print, no modify, no save)
- Education (free to use with attribution)
- Public domain (free to use as long as derivative works also freely available)

DReaM Scenarios

Ubiquitous and User Friendly

- Network identity based rights model
- DReaM client on all your devices
  - Can be multiple independent implementations
- Rights assert-able on any trusted device
- Device specific rights automatically revoke without re-assertion
- New devices easily added, old/lost/stolen devices automatically revoke over time

DReaM Scenarios (Cont.)

Healthcare—Emergency Room

- Patient records securely stored and managed
- Physicians have individual network identity as well as changing “role-based” identity
- Treating physician can “acquire” access rights to meet medical emergency needs
- DRM system would track physician access and roles relative to patient records for audit
- Hospital staff unable to acquire rights without sufficient identity and authentication

DReaM Scenarios (Cont.)

Intelligence

- Intelligence data (reports, photos, audio, video, etc) securely protected and managed
- Intelligence officers have network individual identity as well as changing “role-based” identity
- Officers can “acquire” access rights to meet changing security needs which can be remotely managed
- DRM system would track access and roles relative to data records for audit and alert
- Staff not entitled to acquire rights without sufficient identity and authentication

DReaM Scenarios (Cont.)

Enterprises

- Corporate data (sales forecasts, financial data, customer records, new product info, etc) securely protected and managed
- Employees have individual network identity as well as changing “role-based” identity
- Employee can “acquire” access rights to meet changing job needs via remotely management
- Staff unable to acquire rights without sufficient identity and authentication
- Rights easily withdrawn remotely with automatic revocation or network connectivity

DReaM Scenarios (Cont.)

Fair Use Proposal

- First, content (copyright) owner must agree to this licensing model
- Licensor allows/creates a secure licensing service for approved use licensing that can be subpoenaed
- Users register themselves and trusted devices(s) with the secure licensing service and agree to identify themselves to the service. They could be subpoenaed by a court order if user is suspected of violating agreement
- User request license
- Service issues license to user on device with fair use license and terms (quotation, parody, criticism, comment, research, teaching, etc.)

Agenda

Introduction and Background to DReaM

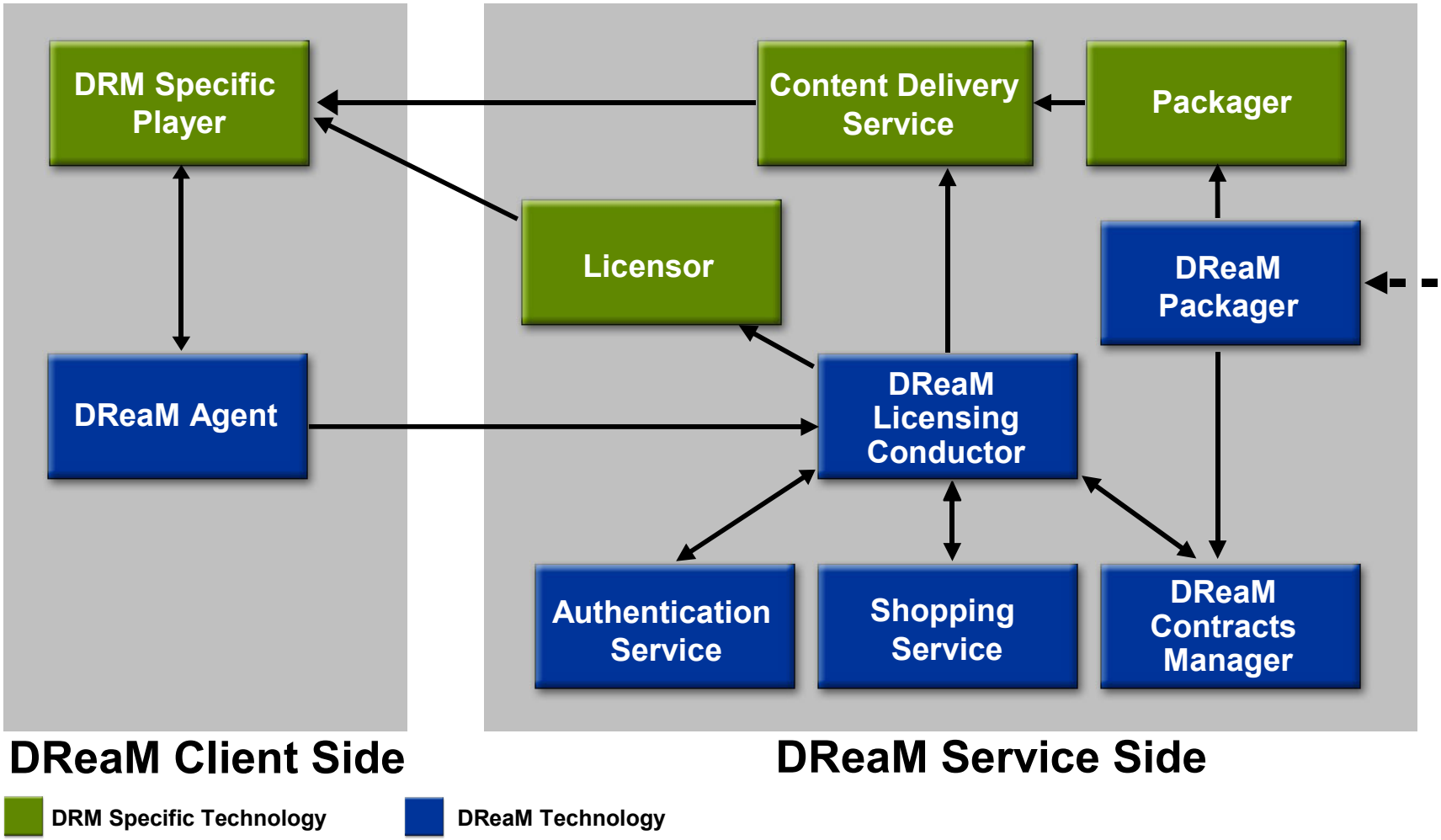
Usage Models

Architecture

Content Protection Technologies

Benefit of Java Technologies in DReaM

DReaM Components



Three Key Elements of DReaM

- Disintermediation (DReaM-D15N)
  - Separation of back-office services from players/consumption
  - Interoperability with existing content protection technologies
  - Emphasis on Identity based licensing over device licensing
- 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)

Disintermediation (D15N) Concepts

- Enables distribution of content across multiple access networks
- Doesn't replace existing DRM/CAS systems, rather it abstracts key function and fully co-exists
- Proxy mechanism needed on devices to redirect to disintermediation server
- Content Usage Rights (CURs) reside on the DReaM Contracts Manager and are superset of the DRM specific CURs which are delivered in a license
- Authentication step requires network connection—usage rights can be exploited when unconnected

D15N—Benefits

- Service providers
  - Ability to retain rights to user/usage data
  - Choose their own authentication solution independent of DRM technology
  - Support existing/legacy devices with own DRM systems which have licensable SDK interfaces
    - Mobile phones, PCs, CE devices
- Content owners
  - Greatest opportunity to reach heterogeneous world of devices owned by users

How Does D15N Work?

1. Content packaged with D15N information
2. Users request rights for content which gets redirected to D15N server (through user's client D15N proxy)
3. Proxy client installed on devices/systems
4. D15N server processes requests
   a) D15N service authenticates user and evaluates content rights, existing rights acknowledged or purchase transacted
   b) D15N service signals license server to deliver license to user
   c) License server then delivers license to client
5. Client receives license key and can now consume content

D15N can be employed with existing protection solutions which allow:

- License server redirection
- Authentication independent of DRM vendor (ex. Liberty, Passport)
- Examples: Microsoft RMS, Secure Digital Container (so far...)

Agenda

Introduction and Background to DReaM

Usage Models

Architecture

Content Protection Technologies

Benefit of Java Technologies in DReaM

DReaM-CAS

Fully specified conditional access system for protecting digital TV streams

DReaM-CAS

- Conditional access profile for MPEG-2 TS
- Strong crypto: AES-128 (content) and RSA (keys)
- Fully specified structures:
  - EMM (Entitlement Management Messages)
  - ECM (Entitlement Control Messages)
- End-to-end prototype implementation open sourced at dream.dev.java.net

DReaM-CAS: Requirements

- Encrypted (video) data—is sent to all clients that request it like in broadcast/multicast scenarios
- No complicated rights negotiations only access to content viewing is controlled—STBs and DVRs supported
- Pay-per-view and video-on-demand with trick-play (fast-forward, fast-reverse, etc.)
- Imperceptible to the user

DReaM-CAS: Key Features

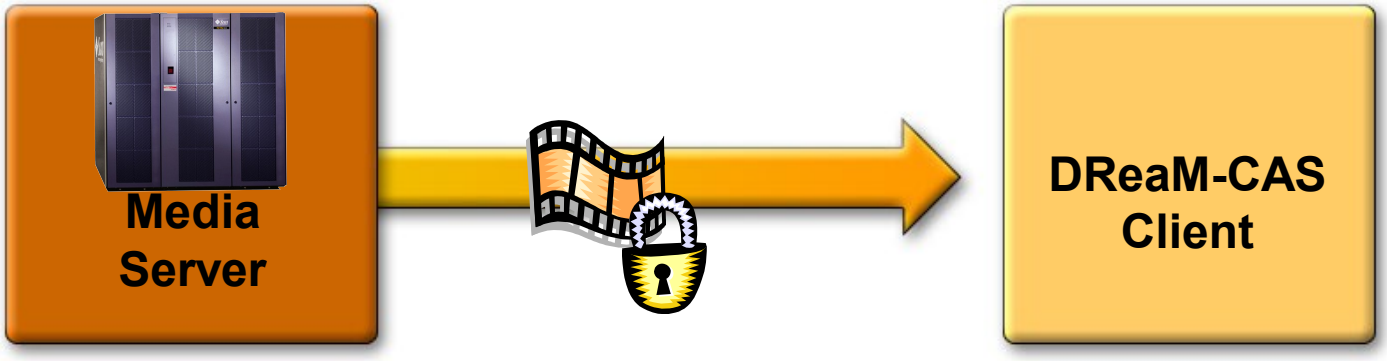
- Symmetric encryption for encrypting
  - Data stream and
  - Stream keys used to encrypt data
  - Open encryption standards (AES 128)
  - In-band
- Asymmetric encryption for encrypting access keys used to encrypt stream keys
  - Public key/private ikey mechanisms
  - RSA asymmetric encryption
  - Out-of-band—https, PKI etc.

DReaM-CAS: Design Premise

- Two-way IP network connection available
- Certificates or equivalent technology for public-keys of clients
- Http or https connection available
- Standard public-key, private-key mechanism for protecting access keys
- Access keys protect stream keys used for protecting data
- Protected stream keys are embedded with protected data

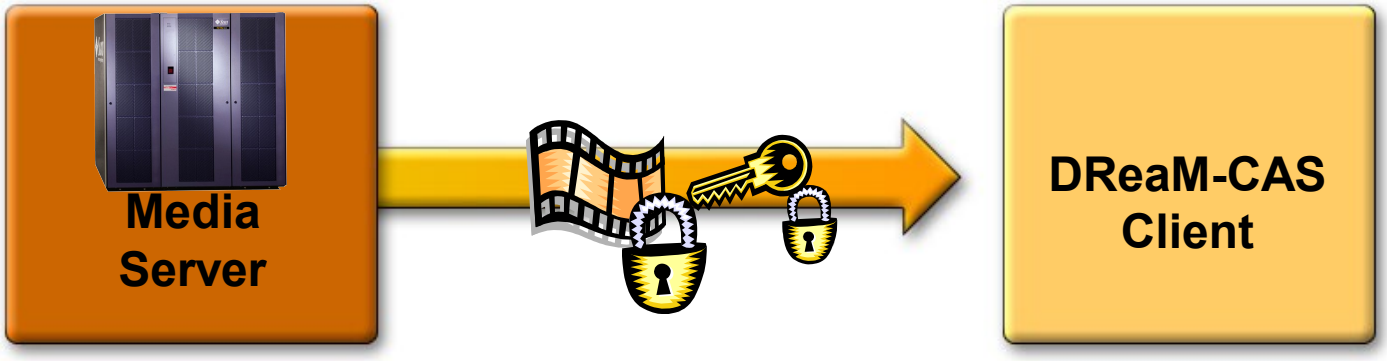
DReaM-CAS: Details

$$\text{Content}_{\text{prot}} = \text{SymEncrypt}(\text{Content}, k_s)$$



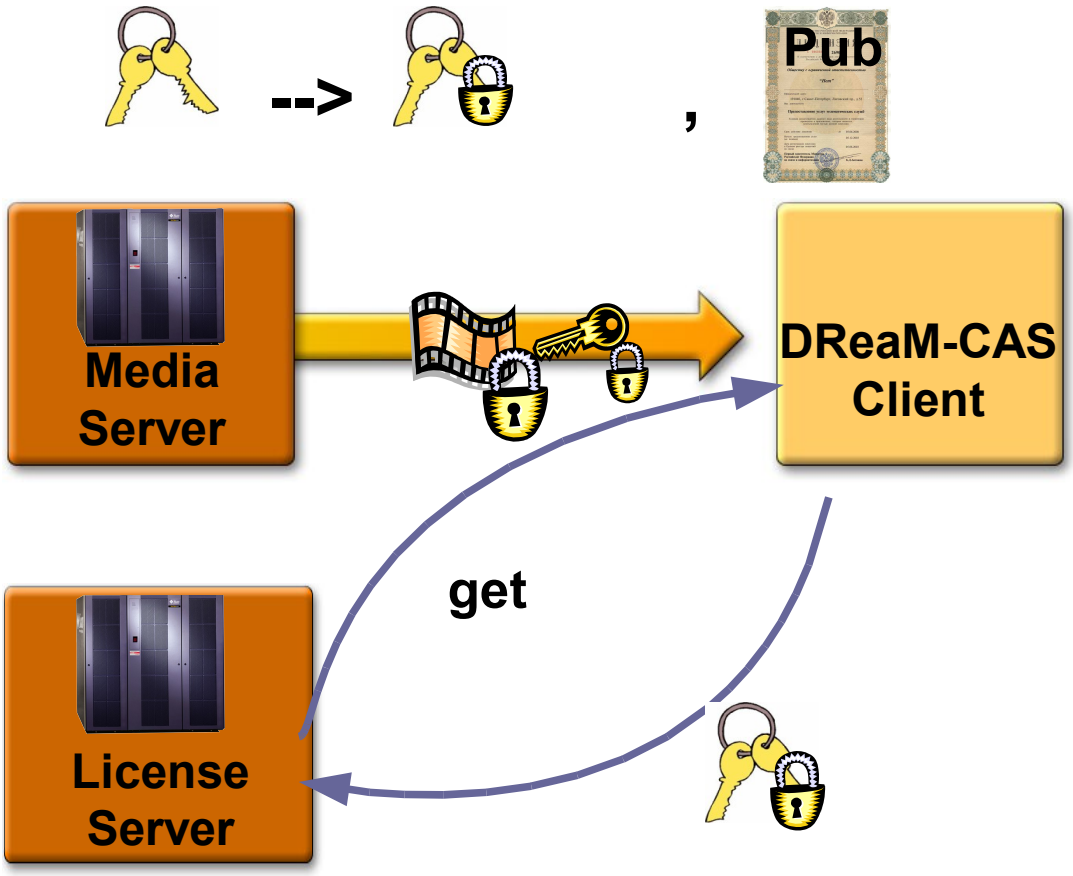
DReaM-CAS: Details

$$K' = \text{SymEncrypt}(k_s, k_a)$$



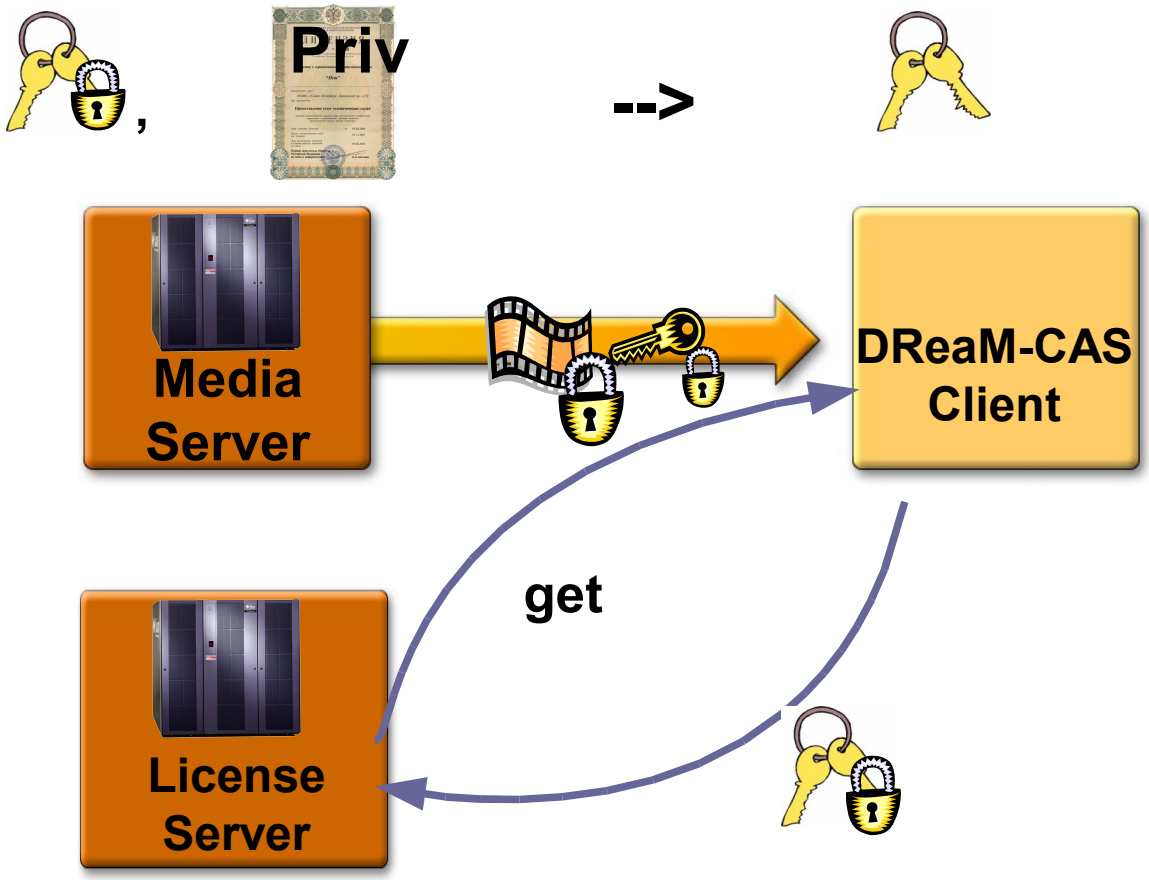
DReaM-CAS: Details

$$T_e = AsymEncrypt(k_a, k_{pub})$$



DReaM-CAS: Details

$$k_a = AsymDecrypt(T_e, k_{priv})$$

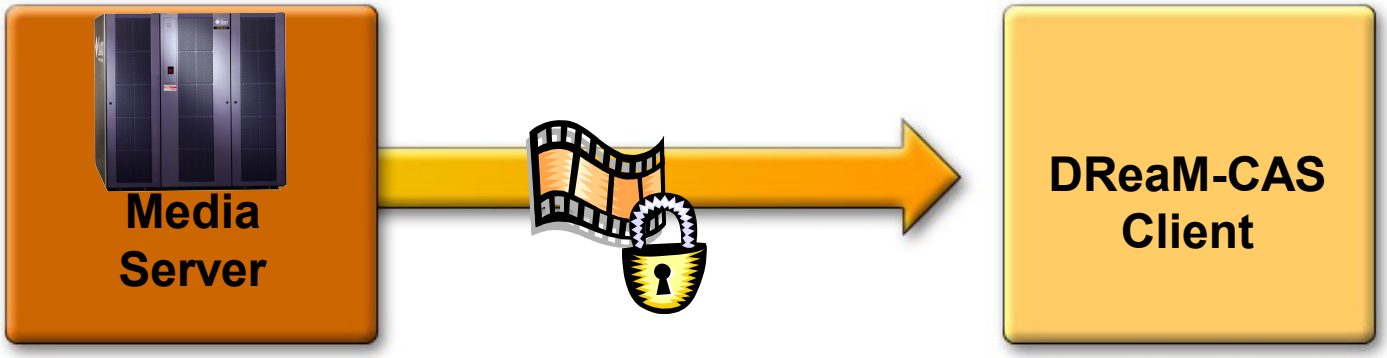


DReaM-CAS: Details

$$k_s = \text{SymDecrypt}(K', k_a)$$

DReaM-CAS: Details

$$\text{Content}_{\text{prot}} = \text{SymEncrypt}(\text{Content}, k_s)$$



DReaM-MMI

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

DReaM-MMI: Requirements

- Mechanism to
  - Request and obtain rights
  - Release unused rights
  - Aggregation of requests
- Mobility—rights accessible for any client
- Impulse buying
- Occasionally disconnected modes
- Support tethered devices

DReaM-MMI: Design Premise

- Rights are stored on the network and accessible to any networked client
- Rights are identity-based
- Fine-grain rights released to the clients
- All clients are networked 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 protocols 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

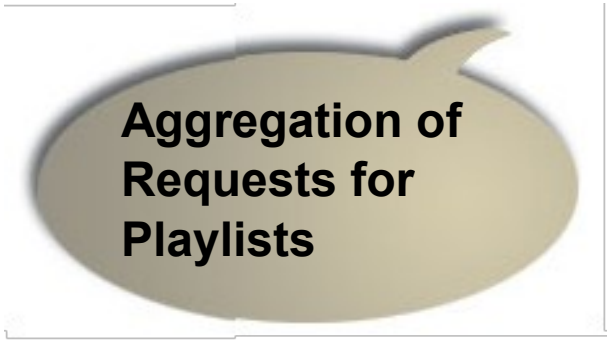
```
IdentitySegment =
  AuthServiceID [AuthTkn]

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
```

DReaM-MMI: Verb Element

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

DReaM-MMI: Response

```
MMIRightsResponse =
1#Status
1#MMIRightsResponsElement
[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]
```


```
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

Agenda

Introduction and Background to DReaM

Usage Models

Architecture

Content Protection Technologies

Benefit of Java Technologies in DReaM

Benefits of Java Technologies in DReaM

- Supports a wide array of media consumption devices:
  - Mobile phones
  - Set-top boxes (MHP/OCAP/JavaTV)
  - Blu-ray devices
  - Laptops/desktops
- Large developer base (> 3 million)
- Content portability across multiple devices with Java

Benefits of Java Technologies in DReaM (Cont.)

- Market leading wireless platform
  - > 1 billion devices
  - Special technology features, e.g., over-the-air (OTA) provisioning, signed MIDlets
- Standardized interfaces allow using “best of breed” components, avoid vendor lock-in
- Industry leading security features and scalability
- More functionality must be supported directly with Java

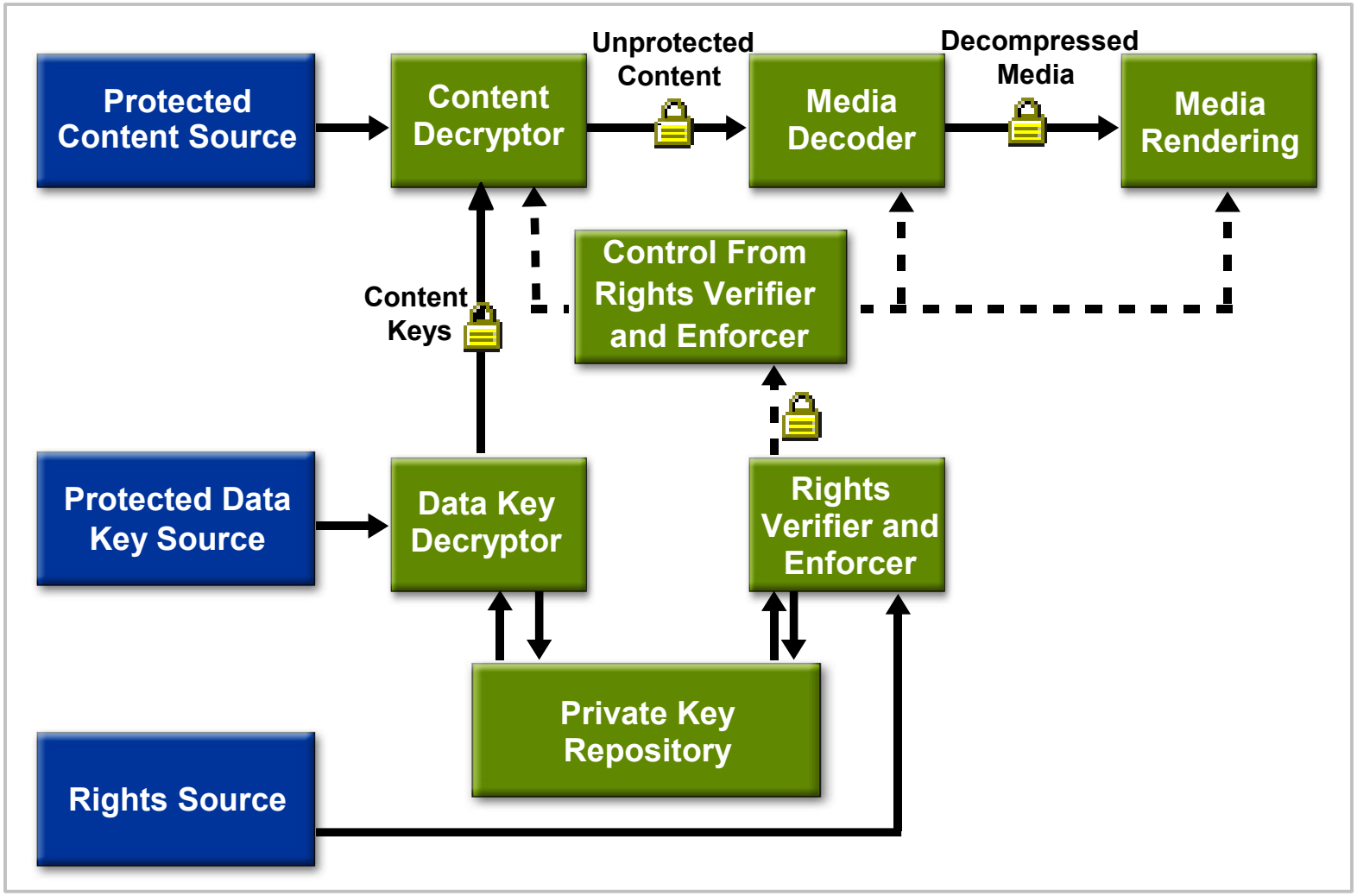
Relevant Java Technologies

- Java ME has support for client robustness
- MIDP 2.0 supports “trusted applications” and “protection domains”
- SATSA (JSR 177) has secure storage, cryptographic operations and secure execution environment
- Relevant JSRs:
  - JSR 37 (MIDP 1.0/JSR 118 (MIDP 2.0)
  - JSR 135 (MM API)/JSR 234
  - JSR 177 (SATS API)
  - JSR 268 (Java Smart Card I/O API)

Requirements for Robust/Secure Client with JME

- In addition to existing Java technologies—what more is required for DReaM client?
- Overview of robustness requirements in the next two slides

DReaM Client: Overview of Robustness Requirements



DReaM Client: Overview of Robustness Requirements

- Secure repository—required for high value data (private keys and rights)
- Secure execution
  - Data key decryptor, rights verifier and enforcer, content decryptor and content processor must be secure (may be implemented as a single secure module)
  - Other secure applications need to be executed to prevent, or detect and quarantine:
    - Unauthorized application execution
    - Display/audio output “scrapers”
    - Unauthorized debugging and probing
- Secure clock—required to ensure rights are not compromised
- Data and control paths, marked with  must be secure

DEMO

Check Out Demo at Booth 902
(Experience the Power of Java)

Specs Available at
http://openmediacommons.org
Code Available at
http://dream.dev.java.net

Q&A

DReaM: Secure End-to-End Interoperable DRM

Vishy Swaminathan,
Tom Jacobs,
Gerard Fernando

Sun Microsystems Inc.,
www.openmediacommons.org

TS-3326