



The Language for Digital Rights™

MPEG, Pattaya, Thailand

*Michael Miron, Thomas DeMartini,
Xin Wang, Brad Gandee*

*ContentGuard, Inc.
December 2001*

Discussion Overview

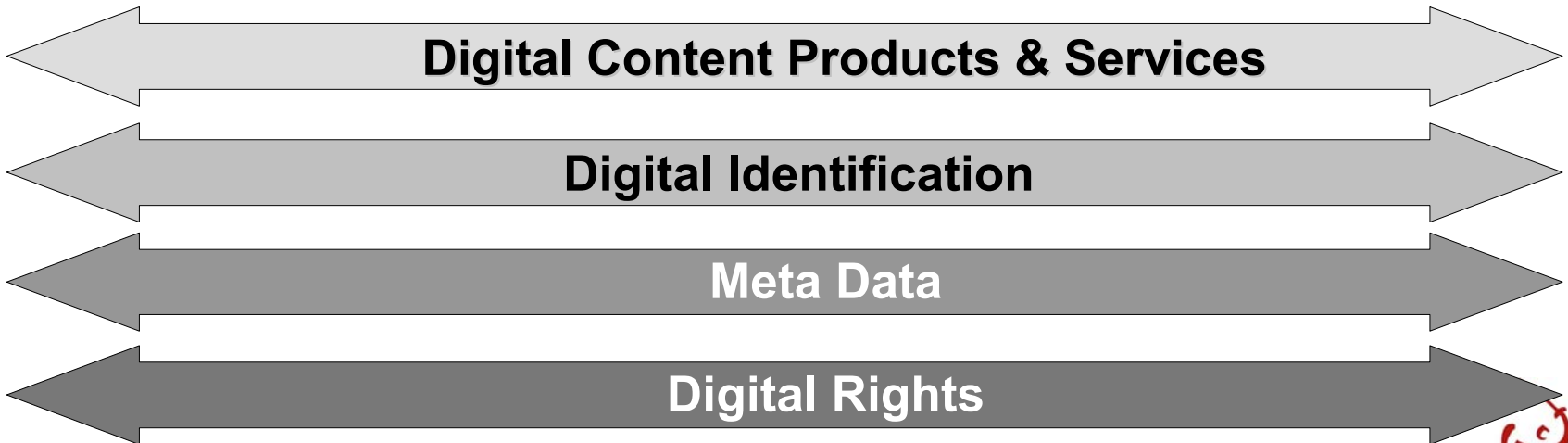
- ① Overview, XrML origins
ContentGuard philosophy,
MPEG submission
roadmap
- ① XrML structure, features,
meeting MPEG
requirements, use cases
- ① Deployment support (tools),
governance, extensions

Michael Miron

Thomas DeMartini
& Xin Wang

Brad Gandee

Digital Supply Chain



Philosophy Underpinning XrML

- ① *Single language across all media types, platforms, formats, resources, products & services to facilitate interoperability*
- ① *Application/domain agnostic structure*
- ① *Comprehensive to express wide variety of business models*
- ① *Application to all phases of life cycle*
- ① *Extensible to allow adaptability and minimize future cost of change*
- ① *Ease of implementation and deployment*

XrML Evolution

XrML 2.0 (11/01) ContentGuard

- ⊙ Support for More Business Models
- ⊙ Enhanced security, flexibility & extensibility

XrML 1.2 (11/01) ContentGuard

- ⊙ Final Maintenance Release of 1.X

XrML 1.03 (8/00) ContentGuard

- ⊙ Enhancements added to increase flexibility

XrML 1.0 (4/00) ContentGuard

- ⊙ Conversion to XML based language
- ⊙ Additional Extensions

DPRL 2.0 ('97-'99) Xerox

- ⊙ Enables specification of rights (fees, terms, and conditions) for digital works

DPRL 1.0 ('94-'96) Xerox

- ⊙ Focus on machine enforceable rights

ContentGuard Patent Policy

- ⊙ ContentGuard holds fundamental early patents
- ⊙ Claims cover the use of any Rights Language
- ⊙ ContentGuard is licensing XrML implementations on RAND basis

ContentGuard

Standards Activity

1. Propose XrML to any organization requiring a Digital Rights Language

- ⊙ MPEG
- ⊙ TV Anytime
- ⊙ OeBF / EBX
- ⊙ DVB
- ⊙ W3C
- ⊙ OASIS
- ⊙ IDRM / IRTF
- ⊙ PRISM
- ⊙ SMPTE DCinema
- ⊙ ICE
- ⊙ cIDf
- ⊙ ISMA
- ⊙ etc.,..

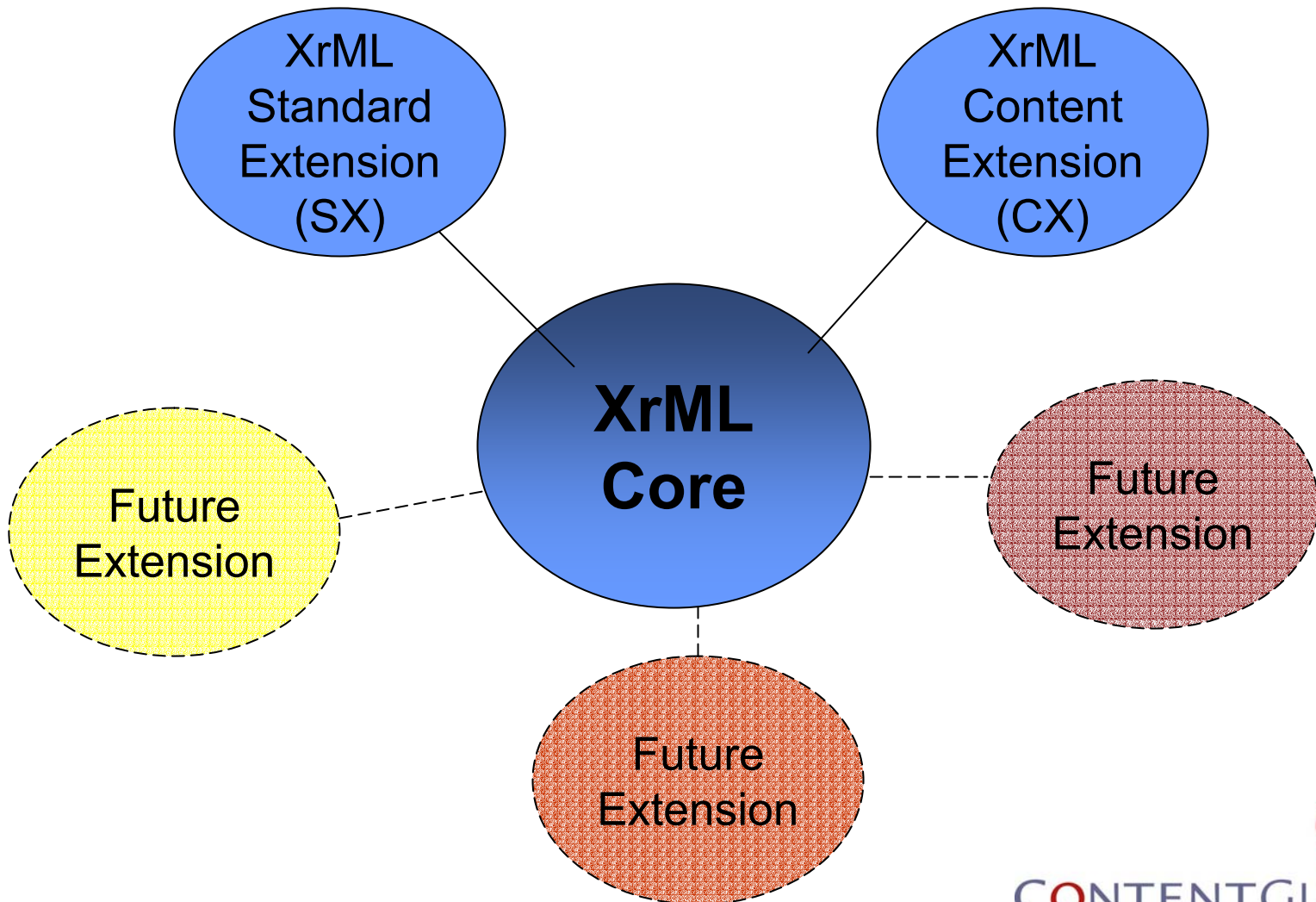
2. Turn Governance over to Single Standards Body

Roadmap of XrML Submission

- ⊙ *A model that was used in developing the submission*
- ⊙ *XrML 2.0 Specification with the following parts:*
 - *Part I: Primer*
 - *Part II: XrML Core Schema*
 - *Part III: Standard Extension Schema*
 - *Part IV: Content Extension Schema*
 - *Part V: Appendices*
- ⊙ *XrML 2.0 Example Use Cases*
- ⊙ *XrML 2.0 Response to MPEG REL/RDD Requirements*
- ⊙ *XrML 2.0: Interaction with Elements of MPEG Multimedia Framework*
- ⊙ *Comments on the appropriateness of the requirements*
- ⊙ *Other relevant standards and specifications*
- ⊙ *XrML SDK implementation*

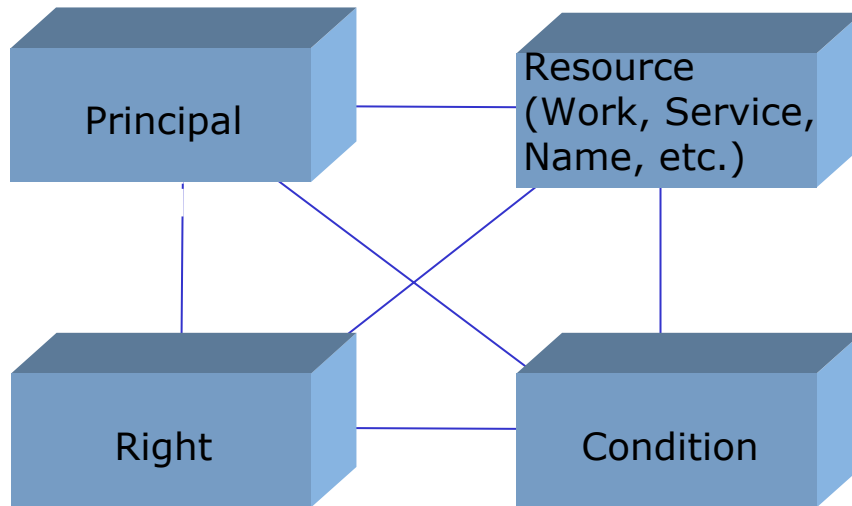
XrML Submission Documents m7640

XrML Extensibility Architecture



XrML Basic Features

◎ Four Key Components



◎ Granting Mechanisms

- Grant
- License

Use Case - Basic

- ⊙ *Alice can Play When the Thistle Blooms for three weeks starting on November 15th, 2001, at 4:03:02 in the morning. (Use Cases Section 3.1).*

Business Models Supported in XrML

- ⊙ Unlimited usage
- ⊙ Flat fee sale
- ⊙ Pay per view
- ⊙ Preview
- ⊙ Promotion
- ⊙ Subscription/Membership
- ⊙ Transfer
- ⊙ Gifting
- ⊙ Library loan
- ⊙ Site/volume license
- ⊙ Rent
- ⊙ Multi-tier models
- ⊙ Territory restricted
- ⊙ Component based model
- ⊙ User types based model
- ⊙ Payment to multiple Rights Holders
- ⊙ Superdistribution
- ⊙ Composite content
- ⊙ Personal lending

XrML Advanced Features

◎ Mechanisms for Enhanced Expressiveness

- Variables (via ForAll and XmlPatternAbstract)
- Rights Grouping (via GrantGroup)
- Delegation (via DelegationControl)
- Meta Rights (via Issue, Obtain, and Revoke)
- “Attribute” Certificates (via PossessProperty)

Use Case – Multi-tier Distribution

- ⊙ *A library can let anyone (from some group) play any song (from some set) for \$1. (Use Cases Section 3.3.a).*
 - *\$1 for user to play*
 - *\$1 for library to let one person play one song*
 - *\$1 for library to let one person play any song*
 - *\$1 for library to let anyone play one song*
 - *\$1 for library to let anyone play any song*

- ⊙ *Features Used:*
 - *ForAll, Variables*
 - *Mathematically sound definition, machine semantic-interpretable*
 - *Multi-tier ready*

Use Case – Subscription

- ⊙ *Any subscriber can view A Book of James. (Use Case Section 4.6.c+b).*
- ⊙ *Features:*
 - *“Attribute” Certificates (via PossessProperty)*
 - *Notion of equality and pattern matching are built into the core and are applicable to all extensions*

Use Case – Superdistribution

- ◎ *Alice can play Investing 101: An Online Lecture Video and can access an online stockquote service for three weeks.*
- ◎ *Anyone else can view the video provided they ask the university that produced it first and it approves.*
- ◎ *Otherwise, they can purchase the course and stock quote service from the distributor for \$59.99.*

- ◎ *Features:*
 - *GrantGroup*
 - *Non-content Resource Types (e.g. Web Service)*

XrML Meets All MPEG Requirements

⊙ Highlights

- *2.1.1 support of multiple usage & business models*
- *2.1.6 extensibility*
- *2.1.8 expressiveness*
- *2.1.18 well-defined semantics*
- *2.1.20 sequencing*
- *2.2.1 digital item description*
- *2.2.10 lifecycle of digital items*
- *2.3.8 revocation of issued permissions*
- *2.4.1 usage conditions*

Req. 2.1.6 Extensibility

- ◎ *XML Schema extension points in XrML include*
 - *Principal*
 - *Right*
 - *Resource*
 - *Condition*
 - *Issuer*
 - *License/any ##other*
 - *DigitalWork*
 - *Metadata*
 - *ServiceReference*
- ◎ *Leverage other existing standards such as for metadata and ID*
- ◎ *Interoperability with INDECS-2 RDD (see the mapping doc)*
- ◎ *Extensions using only equality and pattern matching do not force applications to be upgraded*

Req. 2.1.8 Expressiveness

- ⊙ *Certification of rights ownership by self and other parties*
- ⊙ *Publishing and distribution agreements (rights to edit, aggregate and distribute content, rights to issue rights, and royalty distribution)*
- ⊙ *Specification of any principal, right, resource and condition via variable usage and pattern matching*
- ⊙ *Rights grouping for different purposes*
- ⊙ *Usage state management (initialization, update, querying, sharing, and transferring) for content lifecycle and enabling robust IPMP applications*

Req 2.1.9: Well Defined Semantics

- ⊙ *XrML elements have mathematically precise semantics*
- ⊙ *Unambiguous expressions make XrML machine interpretable and actionable*

Req. 2.1.20 Sequencing

⊙ *Fulfillment*

- *play an ad/legal notice before play a movie (“trackQuery” on state of exercise count of playing ad: > 0)*
- *print after play same content (“trackQuery” on exercise count of play: > 0)*

⊙ *Non-Fulfillment*

- *Transfer if not played – transfer only “not used” content (“trackQuery” on exercise count of play: = 0)*

Req. 2.2.1 Digital Item Description

- ◎ *Secure or non-secure reference to external metadata*
- ◎ *In-line inclusion of metadata in XML & binary forms*

- `<cx:digitalWork>`
- `<cx:metadata>`
 - `<xml>`
 - + `<mpeg21:DIDL`
 - `xmlns:RDF="http://www.w3.org/1999/02/22-rdf-syntax-ns#"`
 - `xmlns:dc="http://purl.org/dc/elements/1.1/">`
 - `</xml>`
 - `</cx:metadata>`
- `</cx:digitalWork>`

- ◎ *Specification of any other mechanism to reference and include metadata*

Req. 2.2.10 Life Cycle of Digital Items

- ◎ *Creation*
 - *Rights to edit, aggregate, and export content*
 - *Rights to issue rights to distribute & use content*
- ◎ *Distribution*
 - *Rights to distribute content & issue and delegate rights*
 - *Rights to repackage content*
- ◎ *Consumption*
 - *Rights to use, transfer and file-manage content*
 - *Authentication of content and its metadata*
- ◎ *End-to-end*
 - *Manage distribution and usage state information*
 - *Rights to revoke issued licenses*
 - *Enable building trust relationship and license verification chains*

Req. 2.3.8 Revocation of Issued Permissions

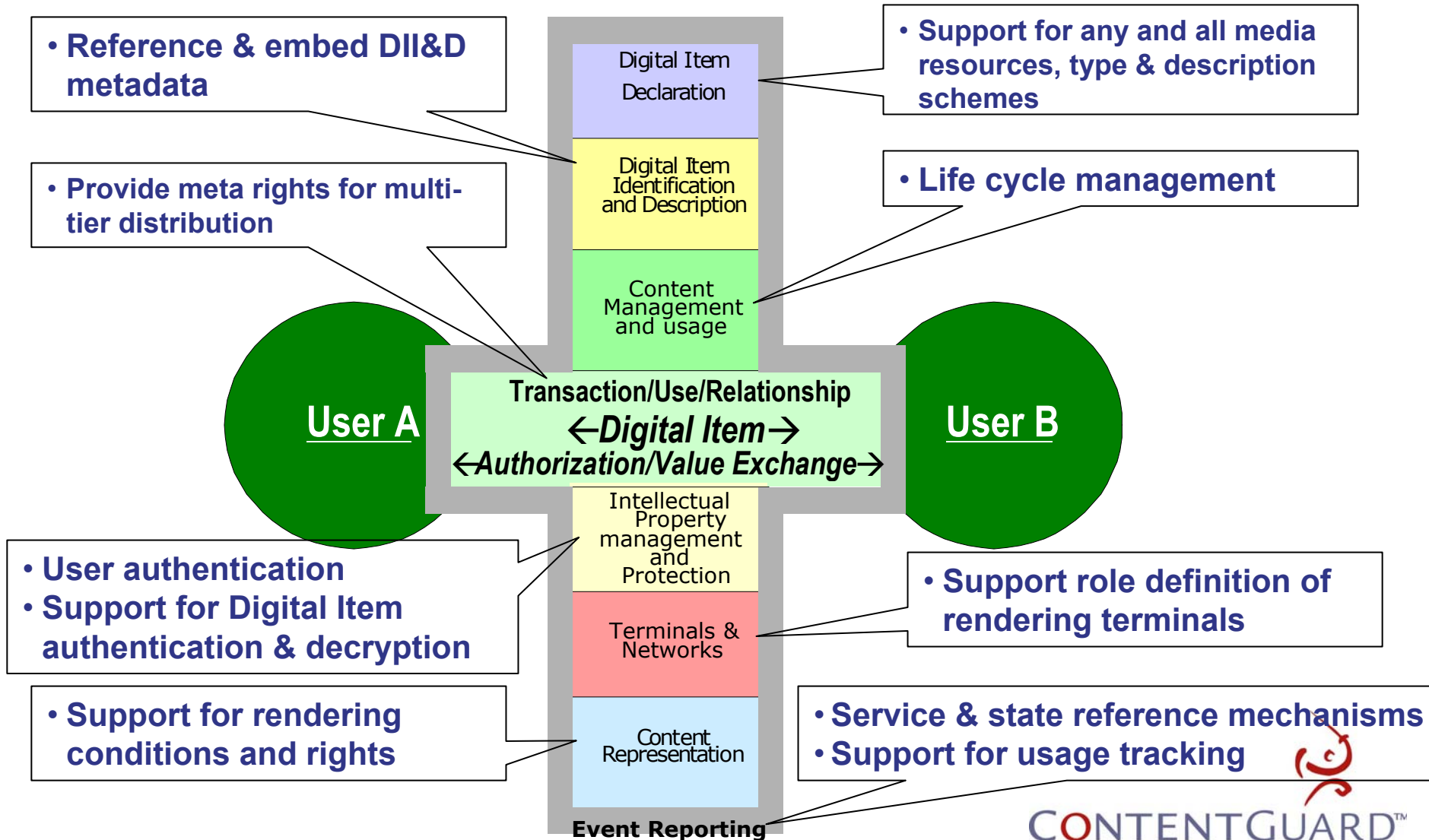
- ⊙ *Explicit right to revoke issued licenses*
 - *Revocation is on their signatures, as anyone can create unsigned licenses*
- ⊙ *Service based mechanisms to check revocation status*
 - *“revocationMechanism”*
- ⊙ *Condition to enforce checking of revocation status*
 - *“revocationFreshness”*

Req. 2.4.1 Usage Conditions

- ⊙ *Temporal*
 - *validityTime, validityIntervalFloat, validityTimeMetered, validityTimePeriodic*
- ⊙ *Fee*
 - *paymentFlat, paymentMetered, paymentInterval, paymentPerUse, markup*
- ⊙ *Exercise Limit*
- ⊙ *Territory*
 - *location (ISO3166 country & region code), domain (URI)*
- ⊙ *Previously issued grant and validly held prerequisite right*
 - *existsRight, prerequisiteRight*
- ⊙ *Exercise count of a (different) right: trackQuery*
- ⊙ *Revocation checking: revocationFreshness*
- ⊙ *Dynamic condition checking: seekApproval*

XrML in MPEG-21 Framework

Examples



Governance

Active discussions with two major International Standards Organizations as home for XrML

- *Long term governance*
- *Management of the extension process*
- *Active Liaison to other Content/Media Specific Standards Bodies*

Implementation & Deployment Support

- ◎ *XrML SDK released to aid developers of Content Applications that :*
 - *Provide Labeling of Content with Rights*
 - *Enable Distribution of Digital Content*
 - *Enable Use of Digital Works according to Assigned Rights*
- ◎ *SDK Documentation includes*
 - *Installation Guide*
 - *User's Guide*
 - *API Programmer's Guide*
 - *Guide to use and create XrML Templates*

Extension Creation

- ◎ “Interactions” (from another MPEG submission)
 - Accept: “User must view and agree with the textual information”
`<xsd:element name="accept" substitutionGroup="o-ex:requiremetElement"/>`
 - Register: “User must register their details with a service provider”
`<xsd:element name="register" substitutionGroup="o-ex:requiremetElement"/>`
 - Example:
`<register>
 <context>
 <service>http:example.com/registerhere</service>
 </context>
</register>`

Example Extension to XrML

```
<xsd:schema targetNamespace="http://www.example.org/interact"
  xmlns:i="http://www.example.org/interact" xmlns:r="http://www.xrml.org/schema/2001/11/xrml2core"
  xmlns:dsig="http://www.w3.org/2000/09/xmlsig#" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:import namespace="http://www.xrml.org/schema/2001/11/xrml2core"/>

  <xsd:element name="accept" type="i:Accept" substitutionGroup="r:condition"></xsd:element>
  <xsd:element name="register" type="i:Register" substitutionGroup="r:condition"></xsd:element>
  <xsd:complexType name="Accept"><xsd:complexContent>
    <xsd:extension base="r:Condition">
      <xsd:sequence>
        <xsd:element name="statement" type="r:LinguisticString" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension></xsd:complexContent></xsd:complexType>
  <xsd:complexType name="Register"><xsd:complexContent>
    <xsd:extension base="r:Condition">
      <xsd:sequence>
        <xsd:element name="registerServiceReference" type="r:ServiceReference"/>
      </xsd:sequence>
    </xsd:extension></xsd:complexContent></xsd:complexType>
</xsd:schema>
```

XrML 2.0 Highlights

- ◎ **Mathematical Precision** – *no ambiguity*
- ◎ **Expressiveness** – *advanced business models, life-cycle management, usage state tracking, pattern matching*
- ◎ **Well defined core and extensions architecture**
 - *Compact: Use of only those terms needed*
 - *Applications based on equality & pattern matching enable extensions without the need to upgrade*
- ◎ **Comprehensive Security**
 - *Entity authentication (Users, software, hardware, Digital Items, etc.)*
 - *Integrity and confidentiality of rights expressions*
- ◎ **Up-to-date Standards and Technologies**