

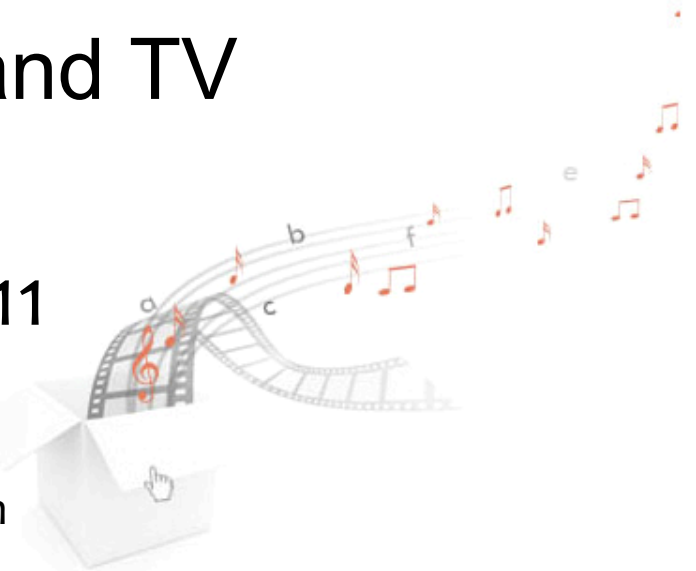
Connected Media Experience Standards Setting Organization

Second W3C Web and TV
Workshop

8-9 February 2011

Gregg Kellogg

gregg@kellogg-assoc.com



What is CME*?

- CME is an open standard for packaging digital content and the associated back-end services that together offer a compelling media experience
- A CME package is a collection of files (encodings), such as MP3 audio files, MP4 video files, JPG images, and more
- A CME package can include online assets and updates that are available to registered users. Registration is easy and users can log in through their existing social media accounts (e.g. Facebook, Twitter, OpenID)
- CME is intended for televisions, computers, mobile devices, and more. Files & data are separate from the user interface, which can be optimized per platform
- The platform supports User Generated Content (UGC) and customization of both the package contents and look & feel
- CME is intended to favor royalty-free IP where possible. Third party licenses may be required for parts of the specification
- More technical details can be found at: <http://www.connectedmediaexperience.org/technicaloverview.html>

Why CME?

Consumers Want

- Deluxe features (lyrics, videos, photos, etc.)
- Aggregated related & relevant data/assets – news, TV appearances, tour info, etc.
- A communal experience with other fans & the chance to connect with stars/artists
- Dynamic content
- No compatibility problems
- Online connectivity and interactivity

Artists & Content Owners Want

- Ability to offer high value packages with deluxe features
- Ability to establish and maintain relationships with fans
- Compelling benefits associated with legitimate content only
- Avoidance of format wars & incompatibility problems
- Support for new business models and release models (e.g. serialized track/content releases)
- Backward compatibility and future proofing

CME is NOT...

- **A DRM** – Nor does it mandate DRM. Content owners can optionally limit online content to registered users, or include DRM content in a CME
- **Proprietary** – CME is available under RAND license to anyone. Where possible, it is based on existing open and free technology
- **Technology specific** – CME has minimum format, metadata, and UI requirements to ensure compatibility. However, any content format or user interface technology supported by the client can be used
- **A new codec** – CME uses familiar codecs, such as MP3 for audio
- **Only for albums** – CME is also intended for singles
- **Music-specific** – CME is designed to work with any type of digital content (e.g. film, television, video games)
- **An opaque monolithic file** – A CME package may expand into a directory
- **Tied to specific usage rules** – CME mandates no content policies
- **For big companies only** – CME is intended for use by indie bands/labels, small retailers, open source players, individuals, and more

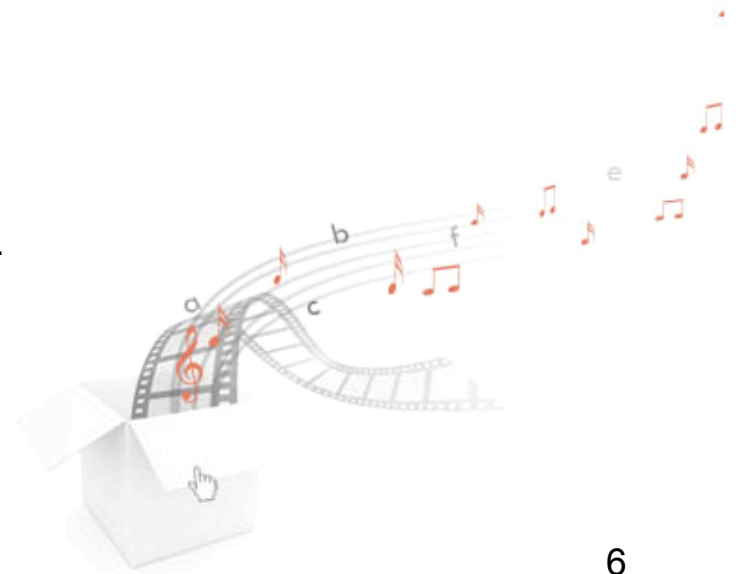
Who is Involved?

- The Connected Media Experience (CME) standards body is working on a voluntary technical standard for enhanced digital packages intended to redefine how consumers experience media
- CME is currently 23 companies and invited experts, with more members joining every month.
- Members are from a range of industries, including content, consumer electronics, and digital service providers
- The CME members are inviting media, technology, and e-commerce companies to join the standards body

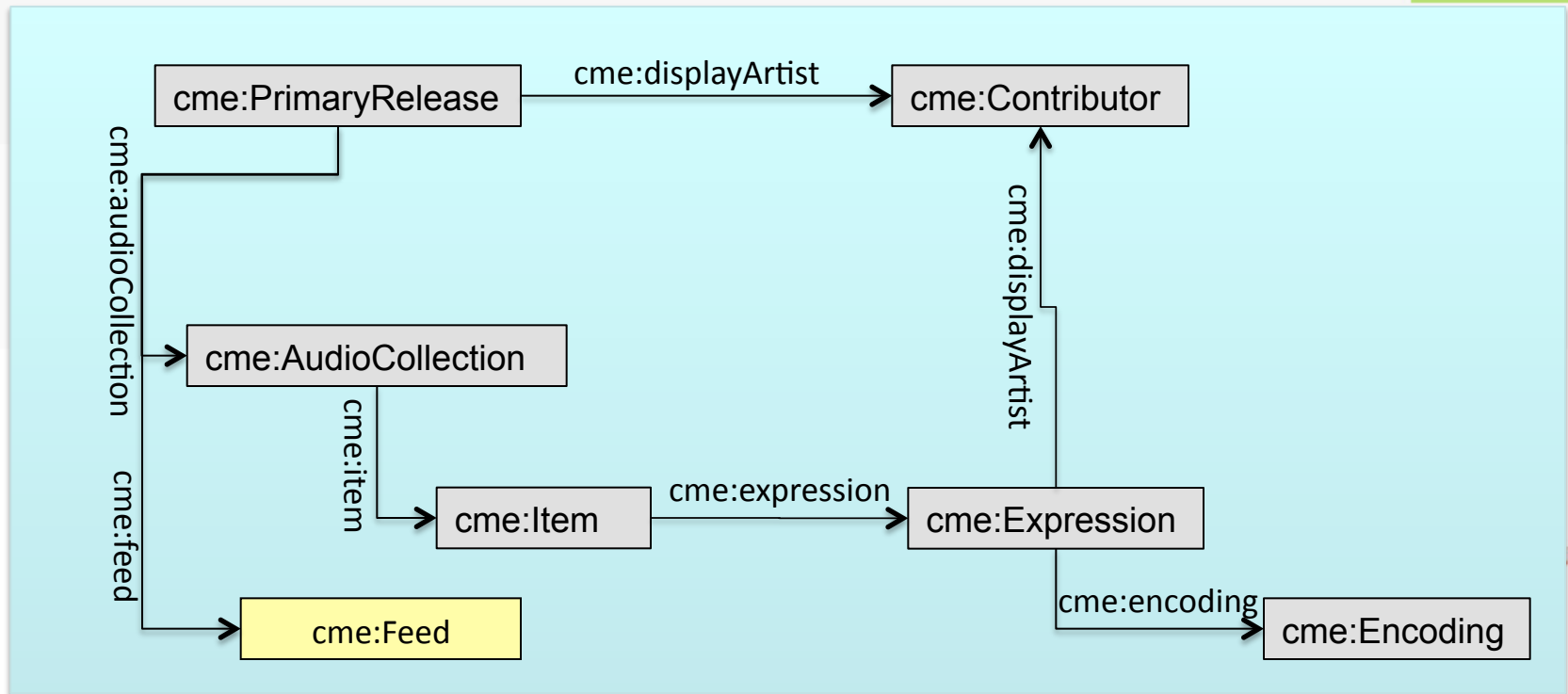


CME Manifest (Notation-3)

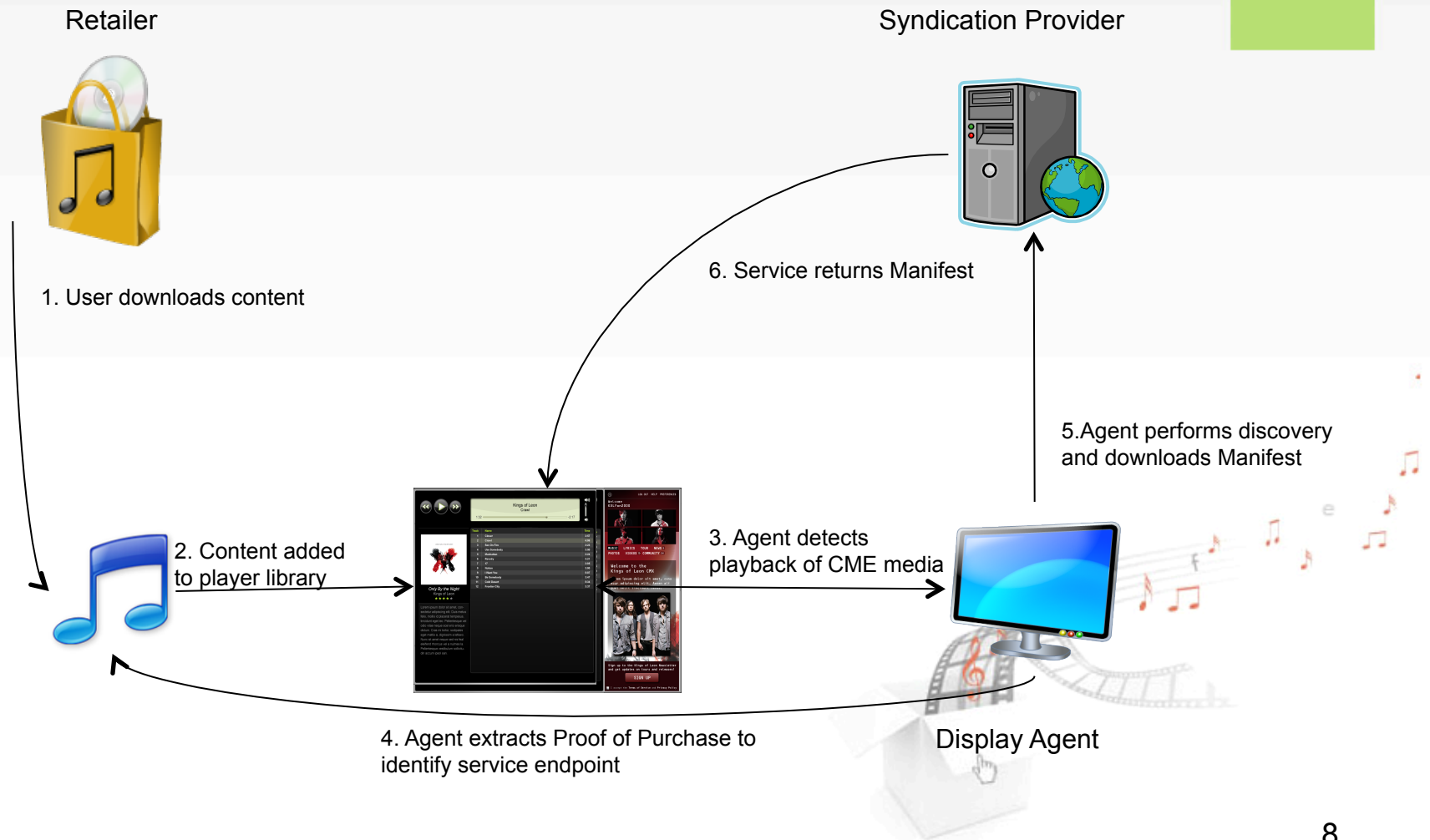
```
@prefix cme: <http://cmx.org/spec/terms> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@base <http://cmx.example.com/releases/Various%20Artists-Happy%20Birthday/"> .
<releases/single> a cme:PrimaryRelease;
  dc:title "Happy Birthday - single"@en-us; cme:displayArtist _:VariousArtists;
  cme:parentalWarning "unspecified"; mo:grid "A1-a1788-aaaaaaaaa-b"^^cme:GRid;
  dc:created "2009-04-30T06:15:51Z";
  cme:displayArtist <contributors/Various%20Artists>;
  cme:audioCollection [ a cme:AudioCollection; dc:title "Songs"@en-us;
    cme:item
      [ a cme:Item; cme:itemNumber="1";
        cme:expression
          [ a cme:Audio; mo:isrc "aa-aaa-11-01789";
            cme:displayArtist _:VariousArtists
            cme:encoding <audio/HappyBirthday.mp3> ] ] ] .
_:VariousArtists a mo:MusicArtist; dc:title="Various Artists"@en-us .
<audio/HappyBirthday.mp3> a cme:Encoding;
  dc:title "Happy Birthday"@en-us;
  dc:format "audio/mpeg".
```



Release Graph



Presentation Workflow



Proof of Purchase (UITS)



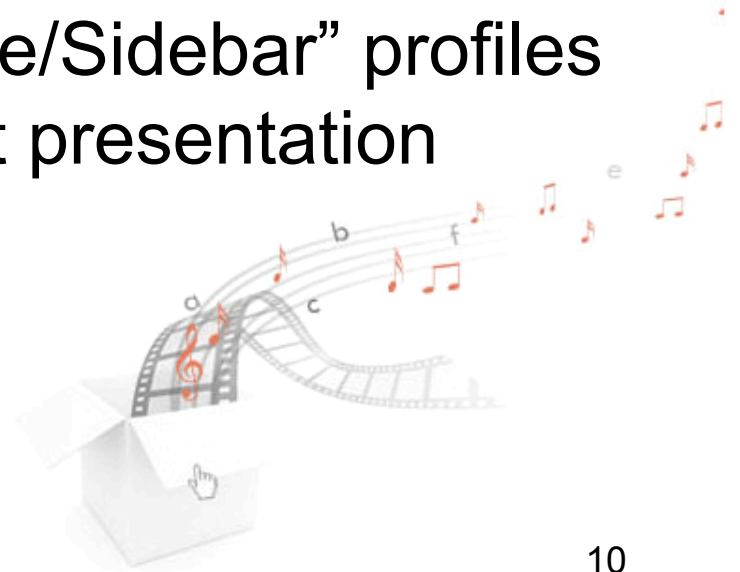
Extract UITS Header from ID3 Tags

Perform discovery on CME URL

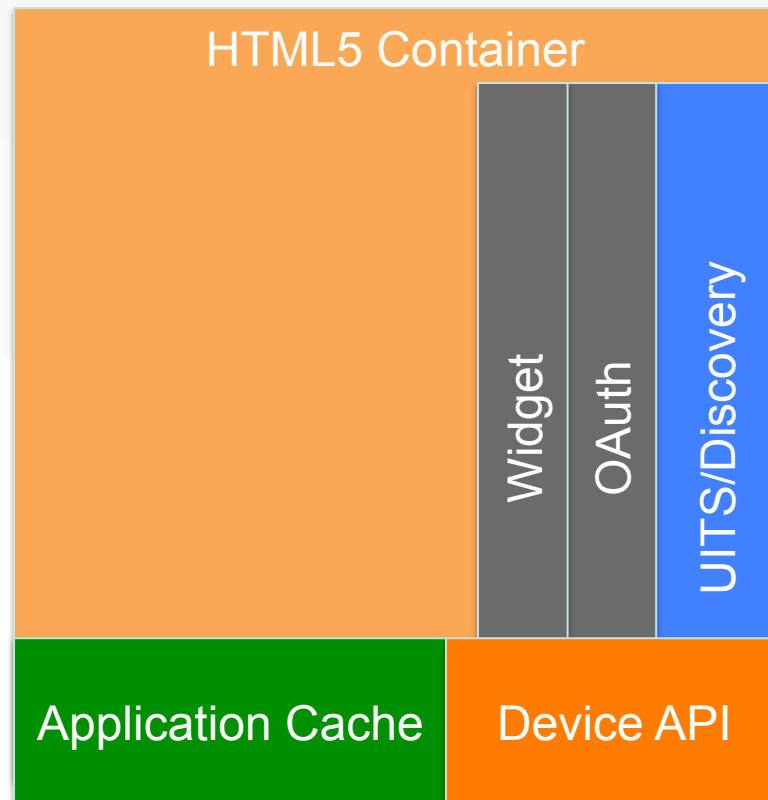
```
<uits:UITS>
  <metadata>
    <nonce>e2ca69945977dc3a</nonce>
    <Distributor>Test</Distributor>
    <Time>2009-04-09T17:45:55Z</Time>
    <ProductID type="GRID">
      A1-aaaaa-aaaaaaaaaaa-b
    </ProductID>
    <AssetID type="ISRC">
      aa-aaa-11-11112
    </AssetID>
    <TID version="1">trans1</TID>
    <UID version="1">user1</UID>
    <Media algorithm="SHA-256">...</Media>
    <URL type="CME">
      http://cme.example.com/releases/KingsOfLeon
    </URL>
  </metadata>
  <signature canonicalization="none"
    keyID="cert.rsa" algorithm="RSA2048">...
</signature>
</uits:UITS>
```

Multi-Platform

- PCs, Mobile, TV, Gaming, Auto, Embedded, ...
- HTML5/CSS3/JS presentation platform
 - “High Definition” and “Mobile/Sidebar” profiles used to identify content, not presentation elements.



Agent Architecture

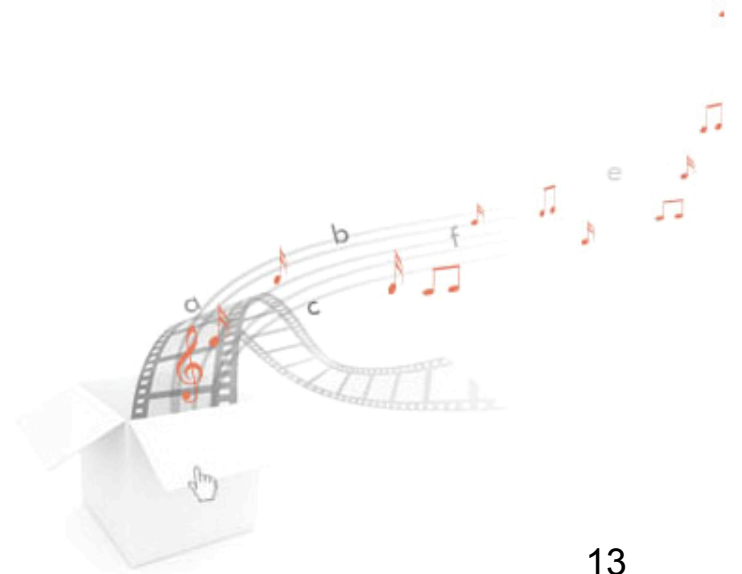


Shared Access

- One user, multiple clients
- Shared Registrations between users
- Buy on PC, play on Mobile
- Buy on Mobile, play on PC
- Assets shared through Media Library
 - API Requirement to access/save into media library



Considerations for W3C



Content Owner Perspective

- Expectation: DVD-like stability
 - runs in 2011 ... still runs in 2021
 - however, new releases need exploit new features
- Limitations:
 - Content must be paid for
 - Limited support lifetime
 - Limited downloadability of extended assets
 - Expectation of lasting value to consumer
 - Transportable between devices

Standards Perspective

- Standards must last for a long time
 - Backwards compatibility
- Desire to not invent, however ...
 - Pace of standards slow (e.g., most HTML-spectrum specs still in WD status!)
 - Adoption not uniform

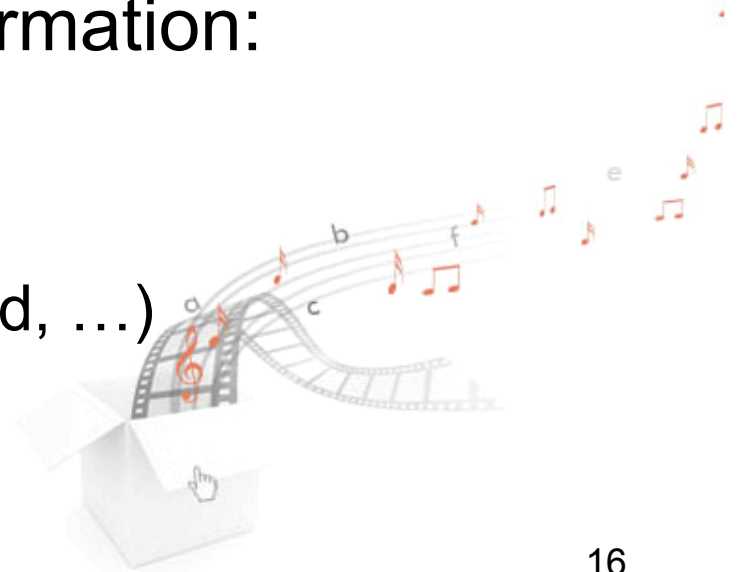
Summary

Calculation of support for currently selected criteria

	IE	Firefox	Safari	Chrome	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
Two versions back	6.0: 9%	3.0: 34%	3.2: 48%	6.0: 85%	10.5: 65%	3.2: 59%			
Previous version	7.0: 13%	3.5: 61%	4.0: 68%	7.0: 87%	10.6: 73%	4.1: 69%			2.1: 56%
Current	8.0: 29%	3.6: 70%	5.0: 81%	8.0: 89%	11.0: 74%	4.2: 71%	5.0: 38%	10.0: 51%	2.2-2.3: 60%
Near Future (early 2011)	8.0: 29%	4.0: 89%	5.0: 81%	9.0: 93%	11.1: 76%				
Future (mid/late 2011)	9.0: 61%	4.0: 89%	6.0: 90%	10.0: 93%	11.1: 76%				

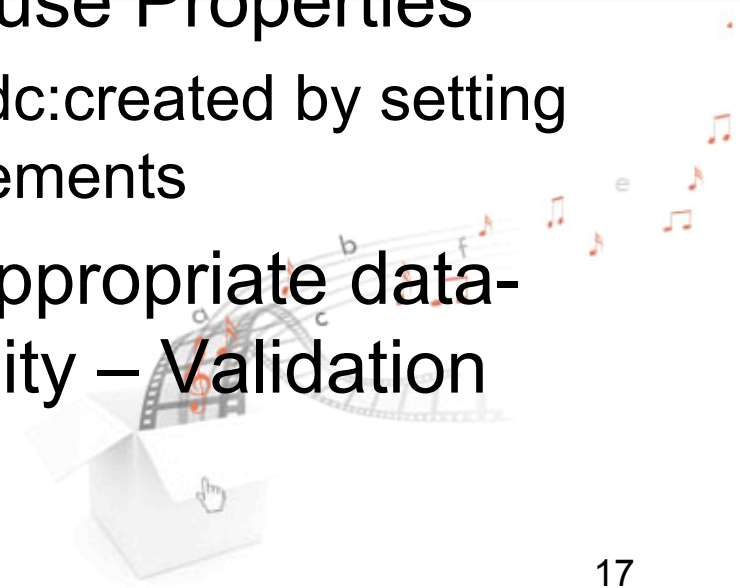
Semantic Requirements

- Need for Metadata
 - Accurately describe contents of release
 - Identify contributors and relationship to tracks
 - Reuse resources between releases
 - Convey interoperability information:
 - downloadsAllowed
 - requiresRegistration
 - unique identifiers (ISRC, GRid, ...)



Semantic Requirements

- Reuse of existing ontologies
 - Media Annotations re-creates basic properties such as *description* and *duration*. Why not use Dublin Core?
 - Advice: create Classes, re-use Properties
 - E.g, CME:Release requires dc:created by setting cardinality and format requirements
 - OWL Restrictions ensure appropriate data-types and property cardinality – Validation




Semantic Requirements

- Collections in RDF are problematic
 - rdf:List not friendly to SPARQL queries
 - rdf:Container (e.g., rdf:Seq) not really semantic!
 - How do derive order without understanding of rdf:_n?
 - RDFa does not automatically create lists!
 - `` – obvious list creation
 - CME uses separate elements for collections
 - Pretty ugly, though!

CME Collection Representation

```
<collections/Songs> a cme:AudioCollection;  
  dc:title "Songs";  
  cme:item  
    [ a cme:Item cme:itemNumber "1"; cme:expression <audio/Bright_Lights> ],  
    [ a cme:Item cme:itemNumber "2"; cme:expression <audio/Push> ] .
```

```
<li about="collections/Songs" typeof="AudioCollection">  
  <h1 property="title">Songs</h1>  
  <ol rel="item">  
    <li typeof="Item">  
      <span property="itemNumber" datatype="xsd:nonNegativeInteger">1</span>  
      <ul rel="expression"><li about="audio/Bright_Lights" typeof="Audio">...</li></ul>  
    </li>  
    <li typeof="Item">  
      <span property="itemNumber" datatype="xsd:nonNegativeInteger">2</span>  
      <ul rel="expression"><li about="audio/Bright_Lights" typeof="Audio">...</li></ul>  
    </li>  
  </ol>  
</li>
```



Semantic Requirements

- Video has greater requirements for Metadata
 - Media Fragments offers a URI handle to describe sub-elements
 - Amount of semantic information expressed in a 2 hour movie may be enormous
 - Consider streaming RDF profile to synchronize semantic state with fragment.
 - SPARQL to service okay, what about embedded/offline?



Video Markup

Scene 42: 42:01:00.00 – 43:01:00.00



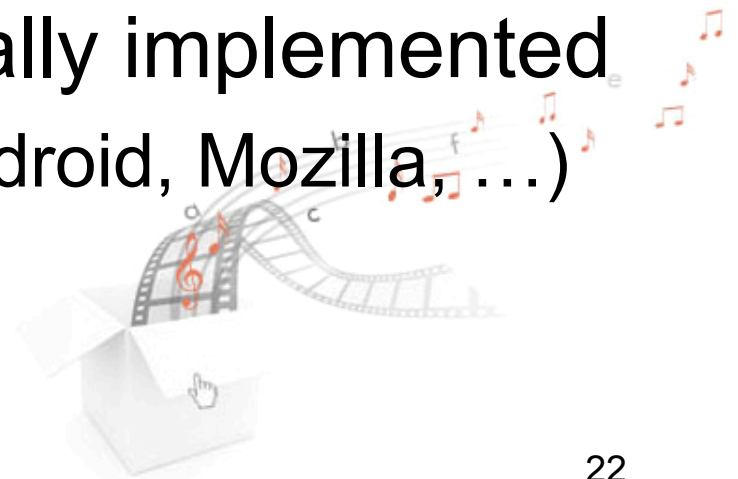
Actor: **Chris Pine**
<http://www.imdb.com/name/nm1517976/>
Actor Timeline
Scenes with Actor

Character: **Spock**
<http://en.wikipedia.org/wiki/Spock>
Character Timeline
Scenes with Character

Object: **Vulcan Ears**
Purchase with 1-click on [Amazon](#)

Other standards issues

- UI Navigation profiles
 - How to adapt a specific presentation to the requirements of a given platform?
- Integration with Media Libraries
 - Gallery API is a start in the right direction
- W3C Widgets not universally implemented
 - Competing “standards” (Android, Mozilla, ...)
 - Incremental updates



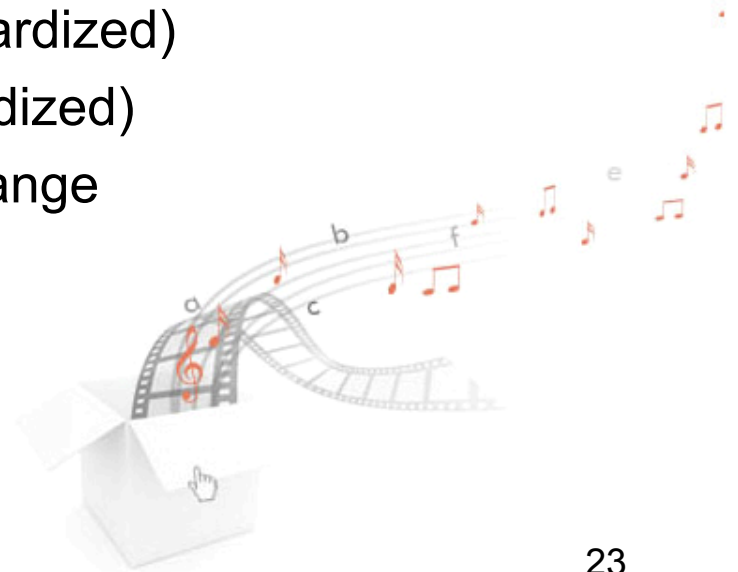
CME Navigation Profile

navigationProfile of type `sequence<DOMString>`, readonly

Communicated through JS API as defined by Agent.

Possible Profile values:

- **mouse** – basic mouse events
- **multi-touch** – `onTouchStart`, ... (not standardized)
- **gesture** – `onGestureStart`, ... (not standardized)
- **arrow** – `onClick`, `onFocus`, `onBlur`, `onChange`
- **keyboard** – `onClick`, `onFocus`, ...
- **drag-drop** – `dragOver`, `dragEnter`, `drop`



Release Properties

cme:Release = mo:Release

cme:PrimaryRelease =
mo:ReleaseEvent

Properties:

cme:collection^{1-*} (cme:Collection)

cme:displayArtist^{1-*} (cme:Contributor)

cme:frontCover^{1-*} (cme:Image)

cme:feed (cme:Feed)

cme:parentalWarning⁰⁻¹

(cme:ParentalWarningType)

cme:presentation^{1-*} (cme:Presentation)

cme:registrationSharing⁰⁻¹

(xsd:nonNegativeInteger)

dc:issued⁰⁻¹ (xsd:dateTime)

dc:modified⁰⁻¹ (xsd:dateTime)

dc:rights (cme:RightsType)

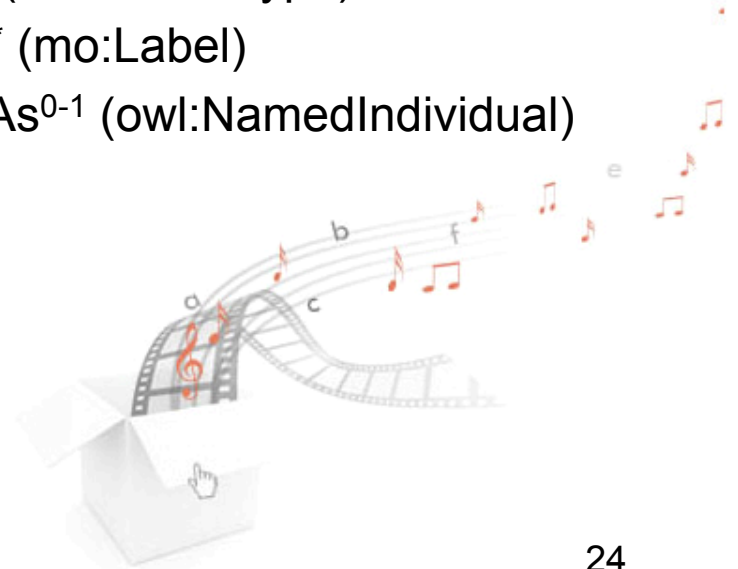
dc:title^{1-*}

mo:genre^{0-*} (mo:Genre)

mo:grid⁰⁻¹ (cme:GRidType)

mo:label^{0-*} (mo:Label)

owl:sameAs⁰⁻¹ (owl:NamedIndividual)



Expression Properties

cme:Expression = frbr:Expression

cme:Image

mo:MusicalExpression

mo:PublishedLyrics

mo:Signal

cme:Audio

cme:Video

cme:Document = foaf:Document

Properties:

cme:collection^{1-*} (cme:Collection)

cme:displayArtist^{1-*} (cme:Contributor)

cme:encoding^{1-*} (cme:Encoding)

cme:frontCover^{1-*} (cme:Image)

cme:feed (cme:Feed)

cme:parentalWarning⁰⁻¹
(cme:ParentalWarningType)

cme:lyrics⁰⁻¹ (mo:PublishedLyrics)

dc:issued⁰⁻¹ (xsd:dateTime)

dc:modified⁰⁻¹ (xsd:dateTime)

dc:rights (cme:RightsType)

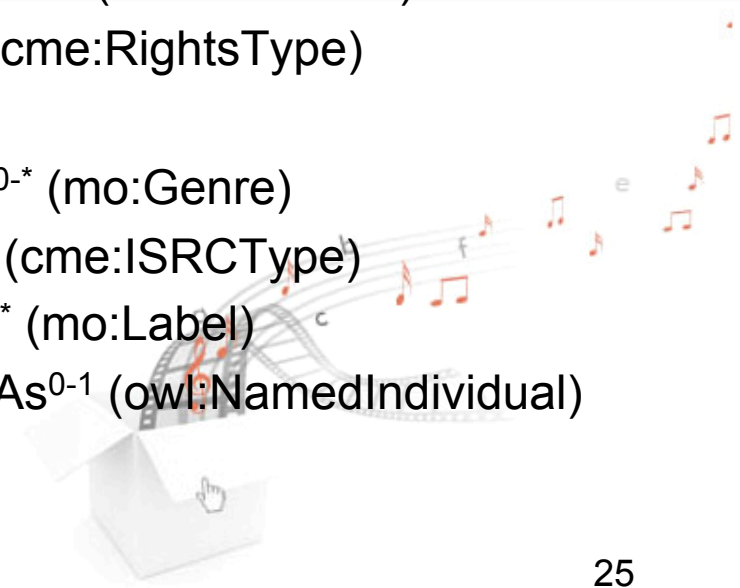
dc:title^{1-*}

mo:genre^{0-*} (mo:Genre)

mo:isrc⁰⁻¹ (cme:ISRCType)

mo:label^{0-*} (mo:Label)

owl:sameAs⁰⁻¹ (owl:NamedIndividual)



Encoding Properties

cme:Encoding = frbr:Manifestation

Properties:

cme:displayProfile^{0-*} (cme:ProfileType)

cme:channels⁰⁻¹ (xsd:integer)

cme:duration⁰⁻¹ (xsd:duration)

cme:height⁰⁻¹ (xsd:nonNegativeInteger)

cme:location⁰⁻¹ (cme:LocationType)

cme:payloadHash⁰⁻¹ (xsd:string)

cme:requiresProofOfPurchase⁰⁻¹
(xsd:boolean)

cme:requiresRegistration⁰⁻¹
(xsd:boolean)

cme:resolution⁰⁻¹ (cme:ResolutionType)

cme:sampleRate⁰⁻¹ (xsd:decimal)

cme:size⁰⁻¹ (xsd:nonNegativeInteger)

cme:width⁰⁻¹ (xsd:nonNegativeInteger)

dc:description^{0-*}

dc:format¹ (dc:FileFormat)

dc:title^{1-*}

