Vorlesung “Multimedia Content Management“

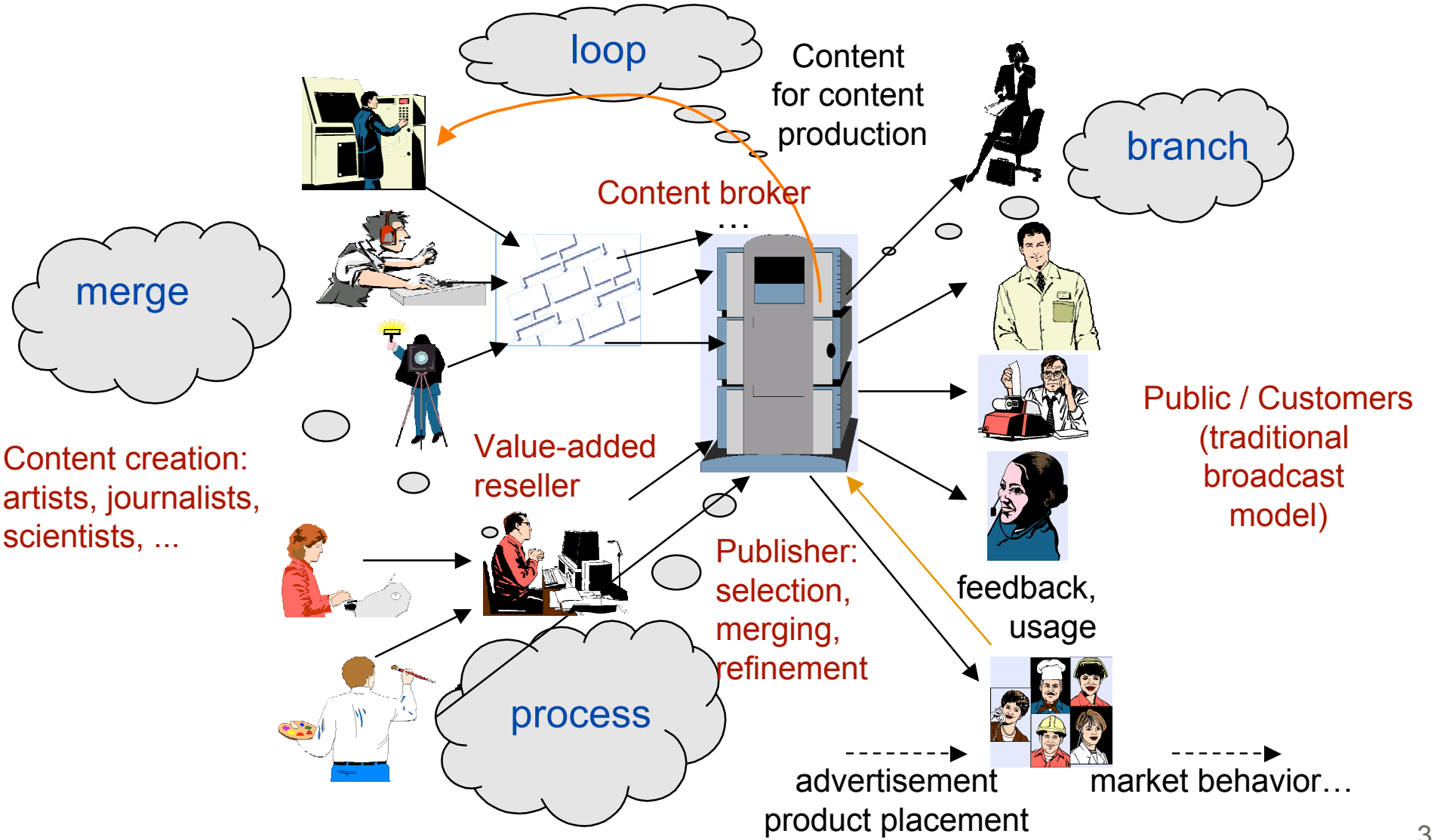
Prof. Ralf Möller, TUHH, Arbeitsbereich STS

- Topics of previous lecture
  - Organization, prerequisites
  - Perception media, multimedia
  - Business models: content commerce
  - Application examples
  - Requirements for MMCMSs
- Topics of this lecture
  - Requirements for MMCMSs continued
  - A software architecture for MMCMSs

Requirements for MMCMSs

- What is required for a MultiMedia Content Management System in order to support the above-mentioned applications?
- Different aspects:
  - Typical use cases
    - How do the various actors access the MMCMS ?
  - Functional specification
    - Identification of and requirements for individual externally visible functions
  - Technical requirements
    - Requirements imposed by the technical environment
    - Dimensioning

Content Commerce: The Content Economy (revisited)



Requirements for MMCMSs / Overview

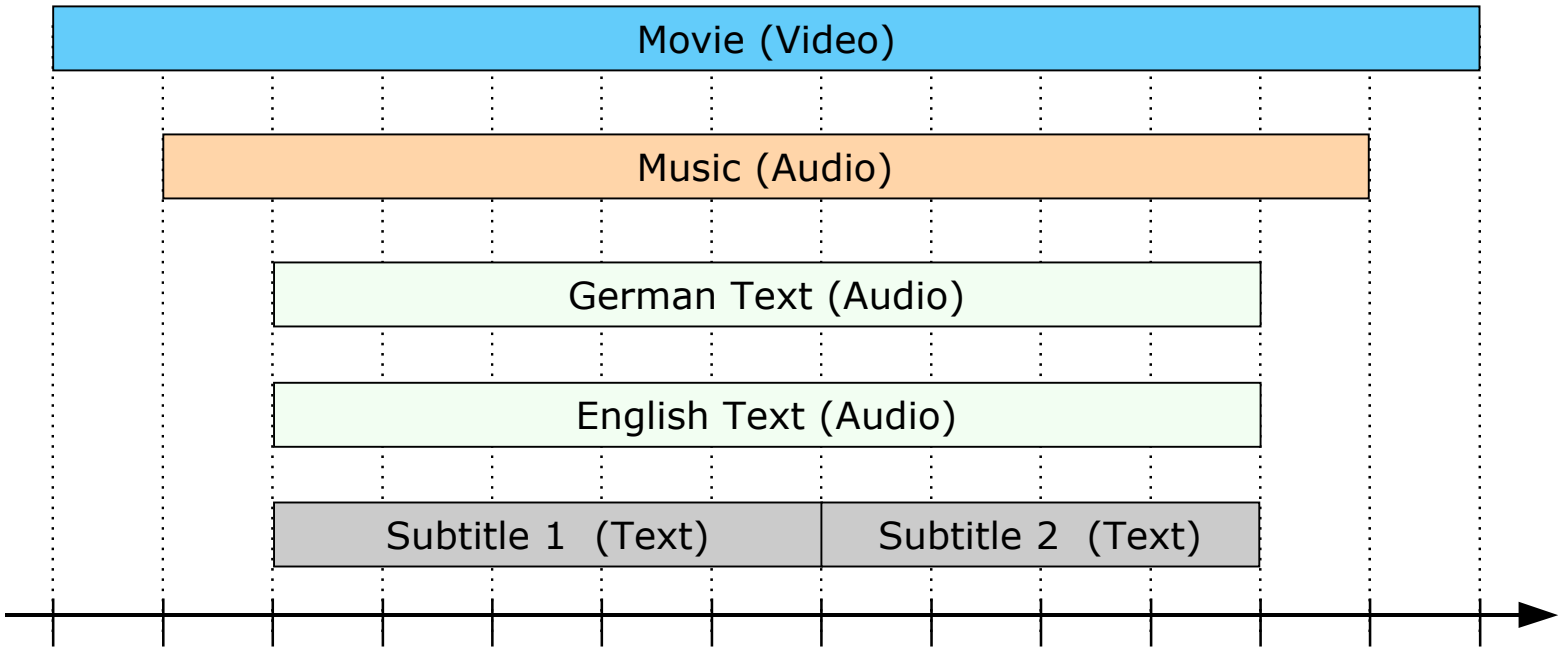
1. **Overview**
2. **Import**
3. **Processing**
4. Export
5. Feedback
6. Technical Requirements

Export – “play out”

- The media content is presented to the user. For time-dependent media objects, this has to happen in real time, i.e., the presentation time is aligned with the passing of time recorded in the media object. Several media objects may be presented together; their spatial and temporal position relative to each other has to be specified and represented on the user’s computer.
  - Embedding content in static parts: The presentation may be assembled from templates and rules for filling the templates with content.
  - Transforming content: The content may have to be transformed prior to export, in order to match the user’s transmission media or output media (bandwidth / size / memory restrictions).
  - In the presentation of time-dependent media data, the user may be willing to sacrifice quality for timeliness
    - Automatic negotiation of Quality of Service (QoS)

Play Out – Synchronization of Different Media

Different Media Bound to a Common Time Line



Apers, Blanken and Houtsma (Eds): Multimedia Databases in Perspective, page 46, fig. 3.3

Play Out - Quality of Service Parameters

- How close are we to the intended presentation?
  - Average delay (system reaction time)
  - Speed ratio (originally intended / presentation rate)
  - Utilization (media data used for presentation / available, e.g. color depth)
  - Jitter (deviation from intended synchrony at some point in time)
  - Skew (accumulated deviation from synchrony)
  - Reliability (avg. frequency of errors)

- Rating of QoS parameters should be based on user's perceived loss of quality.

Play Out – Tightness of Sync. Constraints

- synchronization requirements for different media (max. skew)

Media		Mode, Application	QoS
video	animation	correlated	120 ms
	audio	lip synch.	80 ms
	image	overlay	240 ms
		non overlay	500 ms
	text	overlay	240 ms
		non overlay	500 ms
audio	animation	event correlation (e.g. dancing)	80 ms
	audio	tightly coupled (stereo)	11 □
		loosely coupled (dialog mode with various participants)	120 ms
		loosely coupled (e.g. background music)	500 ms
	image	tightly coupled (e.g. music with notes)	5 ms
		loosely coupled (e.g. slide show)	500 ms
	text	text annotation	240 ms
	pointer	audio relates to showed item	- 500 ms + 750 ms

Apers, Blanken and Houtsma (Eds): Multimedia Databases in Perspective, page 50, fig. 3.4

Export - Hypertext

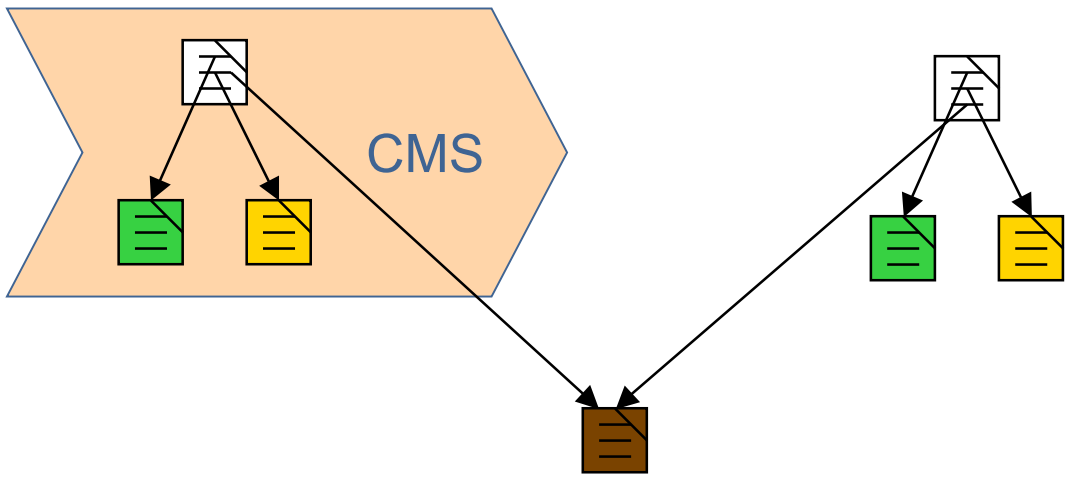
- When exporting a presentation consisting of several distinct documents (e.g. pages), it should be easily possible for the user to navigate inside and between exported documents, either through dedicated navigation elements or through hyper links.
- Links between contents of the MMCMS have to be translated into links between documents in the notation of the created presentation.
- Dead links (to documents not existing any more / not yet exported) should be recognized and avoided (*link management*).
- A link notation has to cover three aspects:
  - Appearance and activation of the link itself
    - The link text, providing some information about the link target
    - How does the user follow the link
  - Description of the link's target
    - Resolved to resource location and accessed when the user follows the link (accesses the link target)
  - Referenced location within the target resource

Absolute and Relative Links

Documents exported together will usually contain links to each other.

When a user requests a preview, links in preview resources should point to preview resources; in the actually exported resources, these links must point to exported resources.

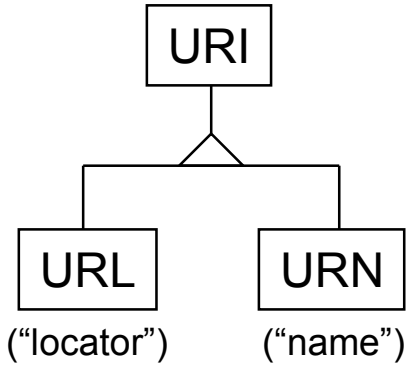
When a set of documents is exported to another system, links have to be relative to the set of exported documents, or refer absolutely to source or target system documents.



- Relative link:
  Location of the target depends on location of the source (context)
- Absolute link:
  Location of the target is independent of location of the source (context)

Links in the WWW: Uniform Resource Identifiers

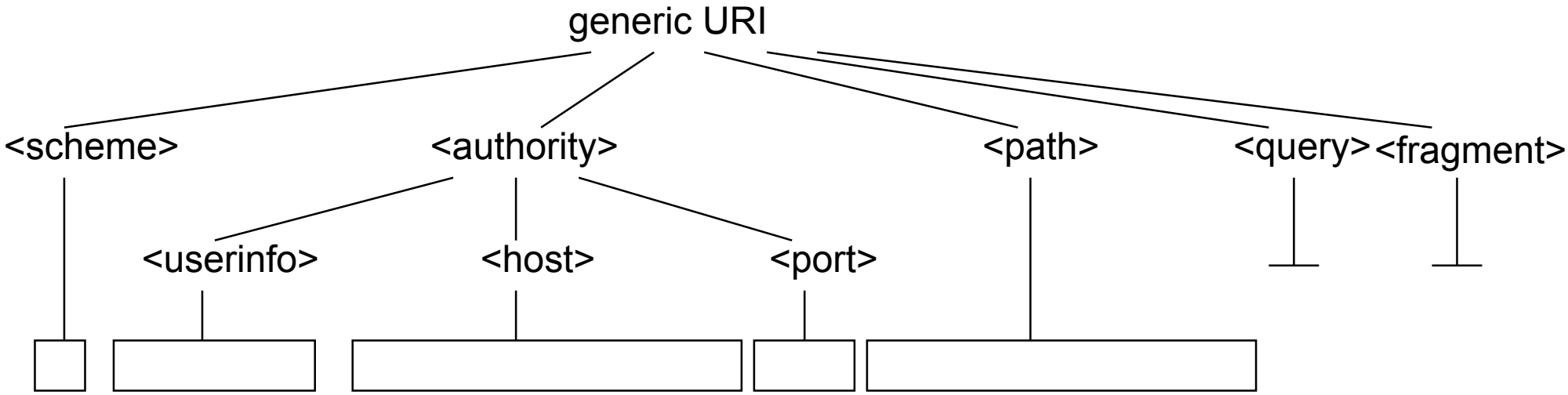
- URI: *Uniform Resource Identifier* (general term)
- URN: *Uniform Resource Name*
  - Target need not exist / be accessible now
  - may be resolved to:
- URL: *Uniform Resource Locator*



- A *generic URI* consists of
  - Scheme - e.g. http, ftp, rtp,…
  - Authority - host name, port number, optionally user information
  - Path - usually hierarchical, absolute or relative, segments separated by slash (/)
  - Query string (optional)
  - Fragment - position within resource

Syntax of Uniform Resource Identifiers

- Human-readable string representation



- ftp://ax.wienberg@ajc01.sts.tu-harburg.de:30001/coremusic/installation/

- Query string: ?<name1>=<value1>&<name2>=<value2>
- Fragment: #<anchor>

Absolute and Relative URIs

- Absolute URI
  - Contains all required information
- `http://www.sts.tu-harburg.de/teaching/ss-01/MMDB/index.html#slides`
- Relative URI
  - Is relative to some base URI
    - Base URI defaults to containing document's URI, but may be explicitly set
    - E.g. HTML header: <base href="http://ajc01:30000/coremusic/generator/sid=3b578f34/">
  - omitted parts (scheme, authority) are filled in from base URI
  - may use relative path:

    `../../ss-02/MMDB/session5.html`

    resolved to
    `http://www.sts.tu-harburg.de/teaching/ss-02/MMDB/session5.html`

Example: Links in XML (1/2)

- Between entities: Attribute value is the URI of the target resource
  - “Simple links” according to W3C’s XLink standard
  - e.g. in HTML: <A HREF=“…”>
- Locally inside an entity: XML ID and IDREF attributes
  - In the DTD, XML attributes can be declared as defining an identifier (attribute type ID) or as referencing an identifier (attribute type IDREF):
    <!ATTLIST person id ID #REQUIRED>
    <!ATTLIST person father IDREF #IMPLIED>
    <!ATTLIST person mother IDREF #IMPLIED>
  - No two elements in an XML entity may have the same ID
  - The value of each IDREF attribute must be the ID of an element in the same XML entity

    <person id=“**m123**”><name>thomas</name><gender>m</gender></person>
    <person id=“**m642**”><name>katia</name><gender>f</gender></person>
    <person id=“m456” father=“***m123***” mother=“***m642***”>
    <name>klaus</name><gender>m</gender>
    </person>

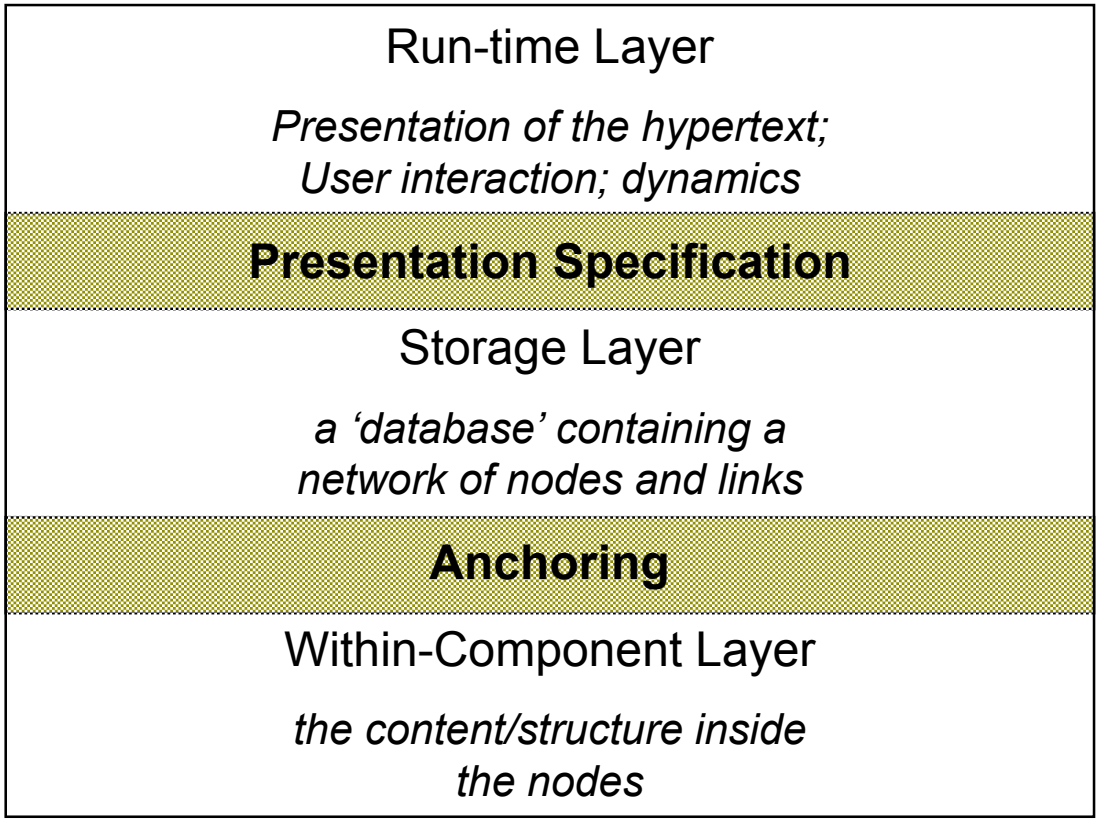
Example: Links in XML (2/2)

- Reference to part of another document using URI *fragment* (part after #)
- XPointer (W3C standard):
  Declarative link by content using path expressions (XPath)
  - http://www.genealogy.de/mann-family.xml#xpointer(person[name==“thomas”])
- HTML uses declared anchors:
  - Declaring the anchor
    <A NAME=“myanchor”>
  - Referencing the anchor
    <A HREF=“doc.html#myanchor”>

Export - Hypermedia

- Combination of Hypertext and Multimedia:
  - Links may be parts of arbitrary media
  - Source (the *linking* part) and destination (the *linked* part)
    - Spatial location
      (in rendered text, 2d within image, 3d within Virtual Reality)
      - Coordinates / area, shape; may be relative to other parts
      - Source: e.g. Visual feedback that traversal would be / was initiated ("onMouseOver")
      - Target: when followed, scroll/zoom to visible area, highlight, …
    - Temporal location in time dependent media object (e.g. audio)
      - Timeline position / range; may be relative to other events
      - Source: freeze or continue source presentation when activated?
      - Target: When followed, skip to beginning in target
    - …or both (in a video and animation)

Dexter Layers for Hypertext



Halasz, Schwartz: The Dexter hypertext reference model. CACM, 37(2):30-39, Feb. 1994

Export – Hypermedia (Implications)

- System needs to know about links (for link management)
  - Links (source) may be buried inside media object
    - Example: Shockwave Flash animation
  - Needs function for extracting links from a media object
  - Needs function for replacing / filling in links during export
    - Example Shockwave Flash: replace with Macromedia Generator, or pass parameters
- Options:
  - System knows about media structure
    - Only play out the requested part of the media object
  - System treats “anchors” as opaque
    - Passed on to media specific modules in the server
    - Or interpreted by the user’s presentation software

Export into another MMCMS

- Content is exported from a MMCMS to another one.
- The content may have to be transformed from the MMCMS's internal representation into a neutral exchange format.
- It should be possible to export documents without loss of information relative to the internal structures (e.g., for archiving purposes). Sometimes, restricted quality is desired because of commercial or bandwidth reasons.
- In contrast to play out, timing is not an issue.
- Record the exact set of versions of documents exported to a customer
  - for recommendation / personalization
  - for context of feedback
  - for accounting purposes
- Before export, authorization / payment should be checked.

Export – Push Model

- A client MMCMS (or user) registers interest in (subscribes) a certain class of content. Regularly, or whenever that kind of content becomes available, a set of documents / document updates is sent to the registered client. In this context, export is an ongoing process.
  - “news feed” from news agency (value-added reseller) to a newspaper (publisher)
  - Email newsletter from publisher to user

Requirements for MMCMSs / Overview

1. **Overview**
2. **Import**
3. **Processing**
4. **Export**
5. Feedback
6. Technical Requirements

Feedback

- Flow of information from the consumer to the publisher
  - Against the direction of the arrow
- Usually not represented by media objects
  - Usage information
    - Which content has been accessed, by whom, in which order?
  - Explicit feedback
    - User rating
- Relate to original content
  - Media objects transformed into delivered multimedia document
  - Version referred to by user
- May trigger additional iteration of internal workflow

Feedback – User profiling and tracking

- A user enters the web site and visits a number of pages which contain contents from several objects stored in the MMCMS.
  - If the user logs in using a nickname or has a cookie, this visit can be correlated to previous visits, *tracking* the user.
  - The user may also voluntarily provide *profile* information about himself/herself (e.g. preferences, interests, age, contact address)
  - In addition, the user may provide explicit *rating* of content
  - The user may invest into the site by configuring a customized presentation (*My…*)
- The publisher gains insights about the correlation between
  - Market segment and
  - Content
- This information can be exploited for *personalization*: targeted advertising, and for *recommending* related content and products of interest for the user, thereby improving the attractiveness of the site for the user (*customer relations*).

User Profiling and Tracking (Some Implications)

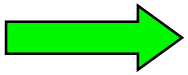
- User Identification, Authentication, Session Management
- Dynamic construction of presentation depending on user profile
  - Scalability => intelligent reuse and caching!
- Recommendation: may require content classification
- Single sign-on:
  - user has to identify him/herself only once to the compound system
  - all components (e.g. community, commerce) share a common user base

Feedback – User Created Content

- Users may provide extended ratings of publisher-provided content (e.g. articles) or external content (e.g. movies) in the form of reviews. Because of the loose coupling between distributed users and the publisher, user created content will usually have a flat structure (web forms, plain text).
- Content created by users may again be subject to review (e.g., for a forum there always exists a master, …

Requirements for MMCMSs / Overview

1. **Overview**
2. **Import**
3. **Processing**
4. **Export**
5. **Feedback**
6. Technical Requirements



Technical Requirements

- Good content and process models are not enough:
- Implemented system must have operational and technical *qualities on strategic level*:
  - Guarantee high **availability**
    - when needed – may differ for various external actors
  - Allow for **scalability** in various dimensions
    - data volume, number of users,…
  - Provide **openness**
    - Integrate into existing / future IT landscape
  - Offer **long-term solution extensibility** (years / decades)
    - Anticipate future demand
    - Handle entire content lifecycle
  - Meet **middleware and hardware restrictions** (economic resource usage)
    - Available/affordable storage capacity, transmission bandwidth, cpu power,…

Availability and Performance

- Web site
  - 24x7
  - Scalability (unforeseeable user base)
  - Target of attacks
- High availability of production tools for editorial staff
  - Early morning to late night
  - Readers and editors may be globally distributed

- Incremental update of web site
- Staging
- Replication
  - Push service towards replicas

Low-Level Integrity (Transactions)

- Content is the mission-critical resource
  - As always: backup & recovery strategy (prepare for catastrophe)
  - Need to make sure the content base is not corrupted (at technical level)
  - For single operations, use *transactions*
    - proven database concept
    - **A**CID – Atomicity
      - Update is executed either completely or not at all
    - A**C**ID – Consistency
      - Low-level check whether the resulting state makes sense
    - AC**I**D – Isolation
      - Outcome is independent of concurrent operations
    - ACI**D** – Durability
      - A committed transaction will not get lost, even if server crashes

High Data Volume for Storage and Transmission

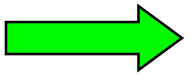
- Persistent secondary storage management
- Stream-based data processing (data may not entirely fit on the client)
- Sharing of data between applications (ditributed servers, multicast)

Media Type	Sample Format	Data Volume	Transfer Rate
Text	ASCII	1 MB / 500 pages	2KB/page
B/W Image	G3/4-Fax	32 MB / 500 images	64 KB/page
Color Image	GIF, TIFF; JPEG	1.6 GB / 500 images 0.2 GB / 500 images	3.2 MB/image 0.4 MB/image
CD-music	CD-DA	52.8 MB / 5 minutes	176 KB/sec.
Consumer video	PAL	6.6 GB / 5 minutes	22 MB/sec.
High quality video	HDTV	33 GB / 5 minutes	110 MB/sec.
Speech	m-law, linear; ADPCM; MPEG audio	2.4 MB / 5 minutes; 0.6 MB / 5 minutes; 0.2 MB / 5 minutes	8 KB/sec.

Apers, Blanken and Houtsma (Eds): Multimedia Databases in Perspective, page 39, fig. 3.1

Requirements for MMCMSs / Summary

1. **Overview**
2. **Import**
3. **Processing**
4. **Export**
5. **Feedback**
6. **Technical Requirements**

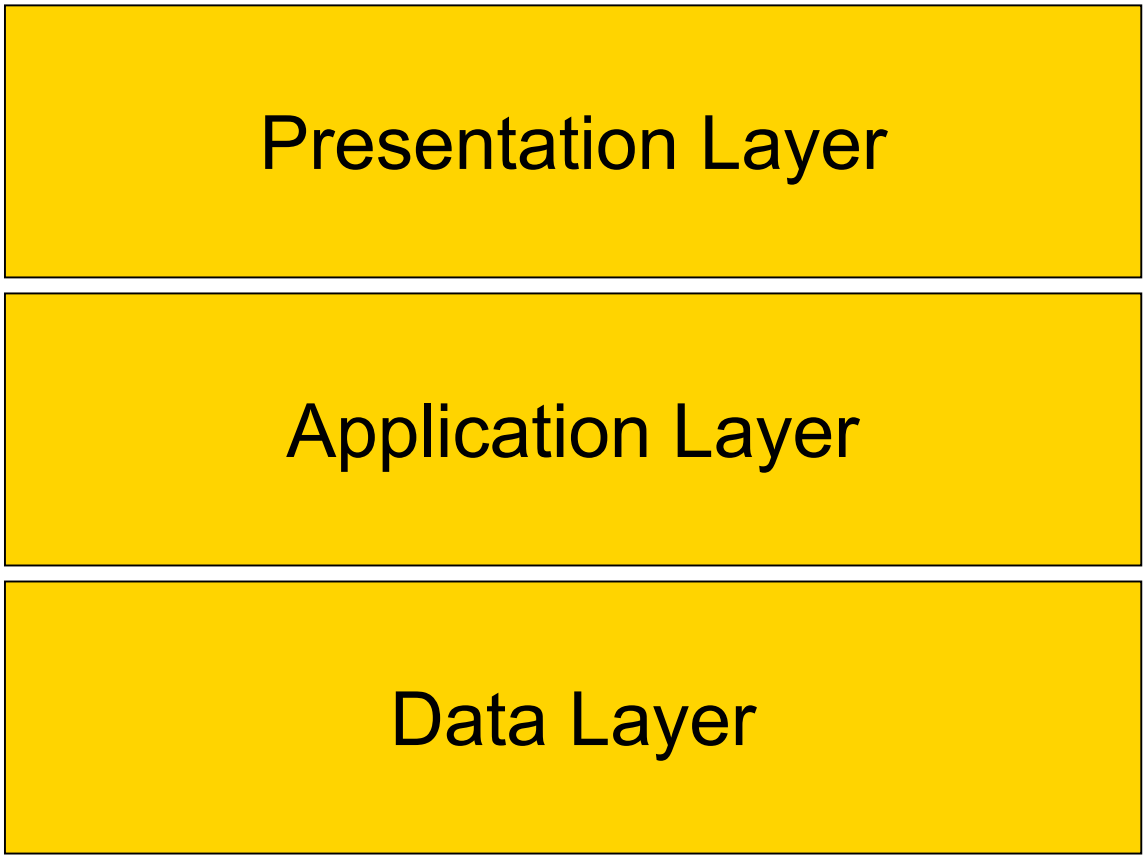


Architecture

Overall Architecture

- Compose System from components considering
  - Functional requirements
  - Non-functional requirements
  - Distribution
  - Communication

The Well-known 3-tier Architecture

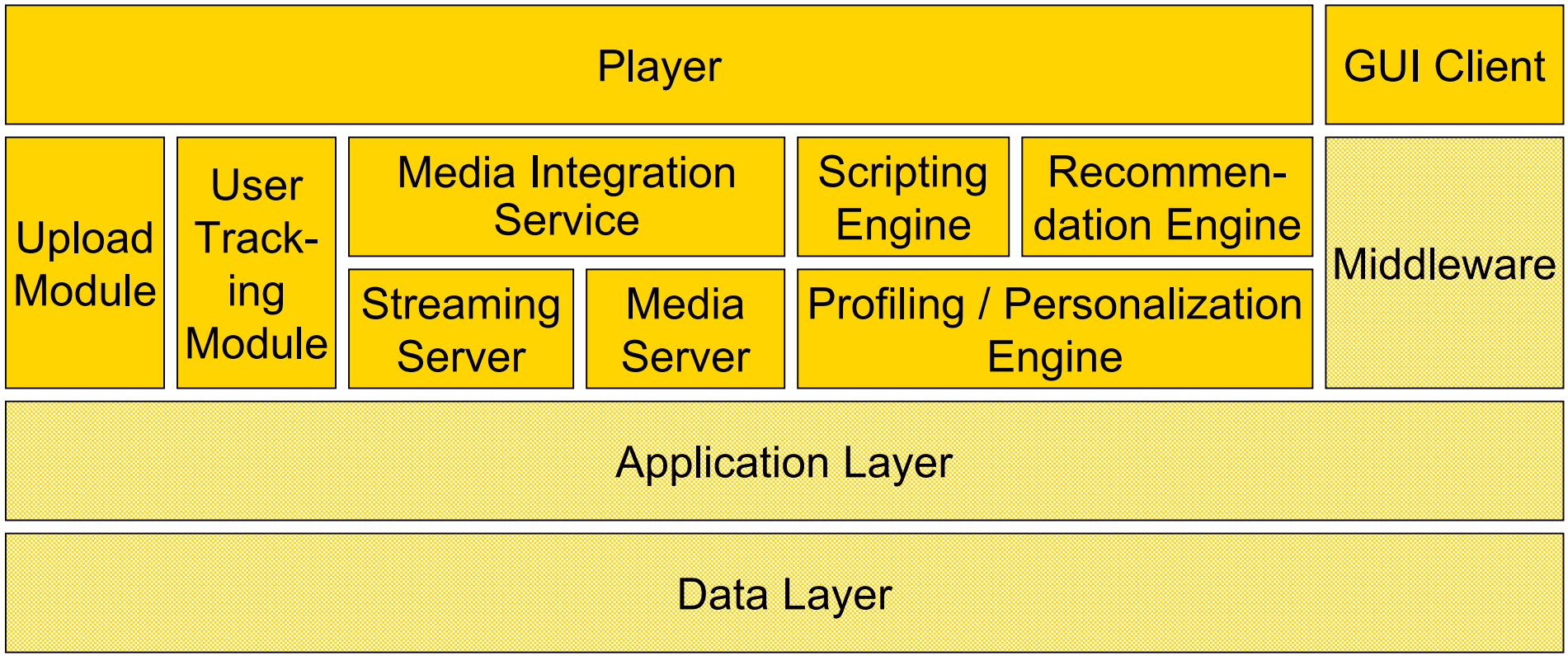


client software;
services for clients

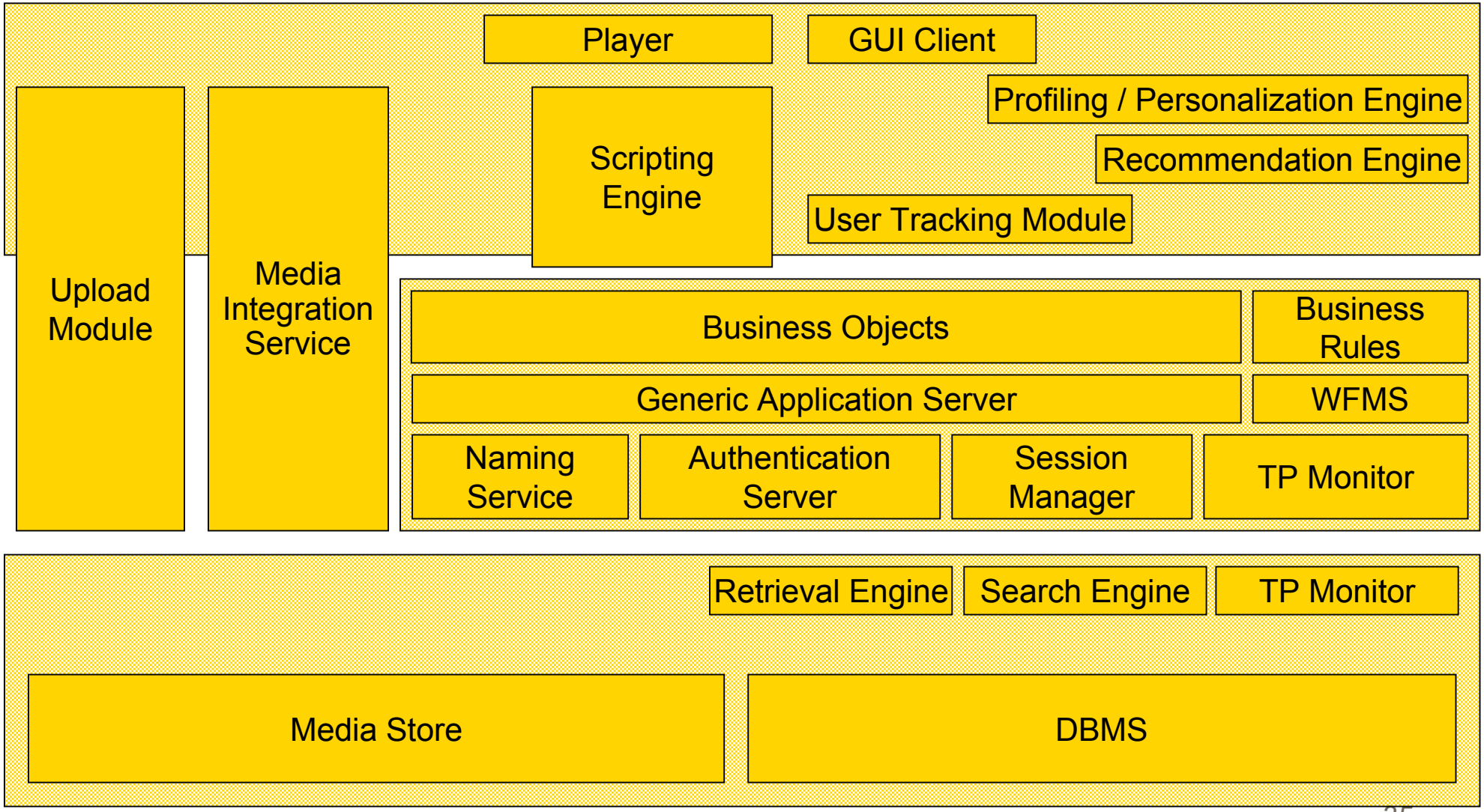
common
services

persistence and
“data-related”
services

Components Diagram: Presentation Layer



MMCMS Architecture Diagram



WFMS

- Two definitions by the Workflow Management Coalition (WfMC):

Definition: *Workflow*
The computerized facilitation or automation of a business process, in whole or in part

Definition: *Workflow Management System*
A system that completely defines, manages and executes „workflows“ through the execution of software whose order of execution is driven by a computer representation of the workflow logic

- Functional areas:
  - Build time: defining and modeling workflow processes and constituent activities
  - Run-time: control & management, sequencing enactment of constituent activities
  - Run-time user interaction: managing work lists (“to do items”)