浙江大学计算机学院
数字媒体与网络技术

Digital Asset Management
数字媒体资源管理

3. Multimedia Database System

任课老师：张宏鑫
2020-10-13

Introduce your teams
and projects
(next lesson)

多媒体数据库系统的挑战

- 多媒体数据的独特数据特性
  - 单个条目数据规模和吞吐大
  - 多通道数据需要同步
  - 连续媒体数据需以流媒体形式提取
  - 对于不同客户的QoS需求
  - 相似性比较和搜索困难

Outline

1. MM content organization

2. MM database system architecture

3. MM system service model

4. Multimedia Data Storage

5. Multimedia application

浙江大学计算机学院
数字媒体与网络技术

3.1. Multimedia Content Organization

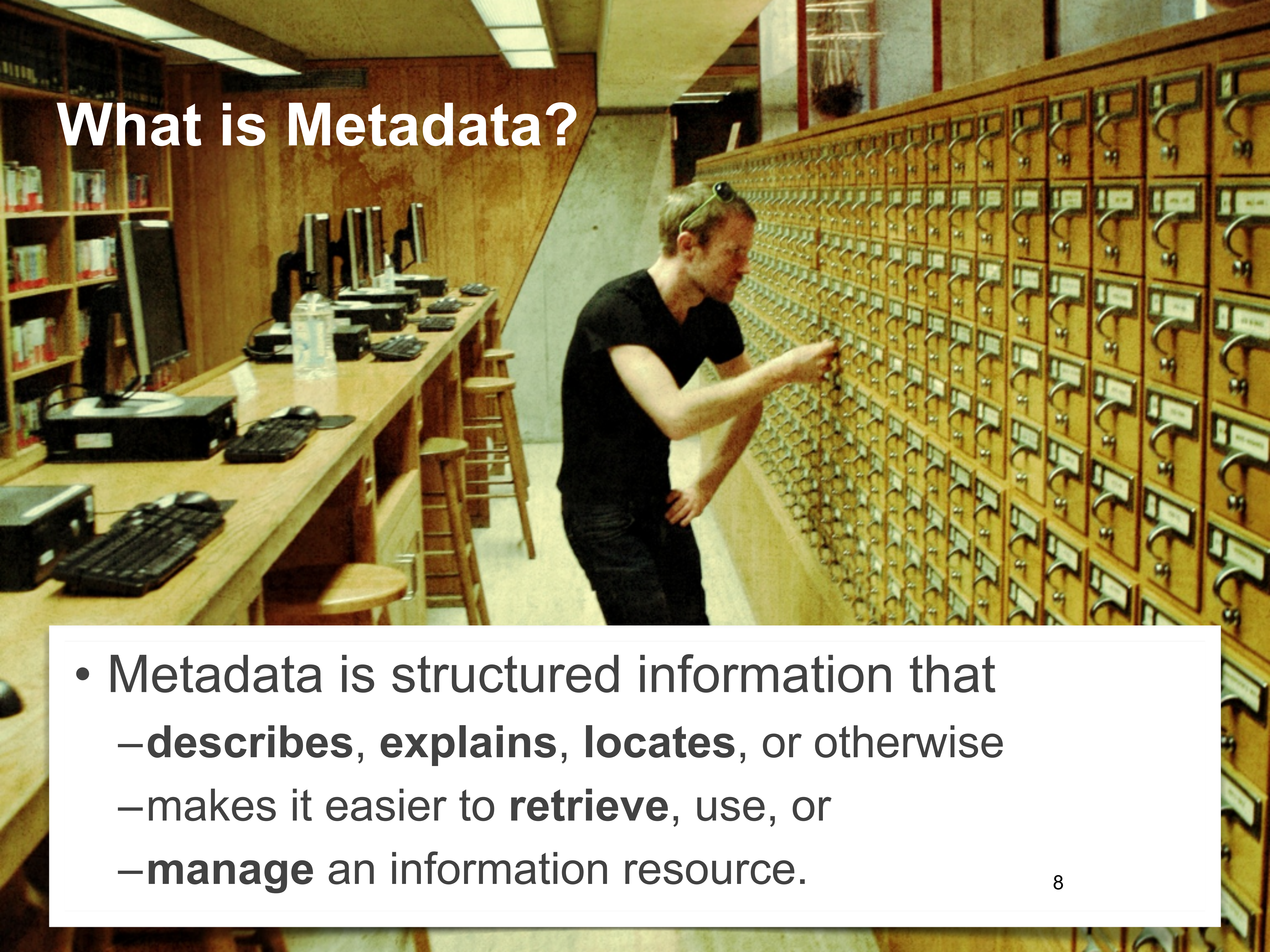
Metadata Model Organization

- Content-dependent Metadata
- Content-descriptive Metadata
- Content-independent Metadata

Metadata Model

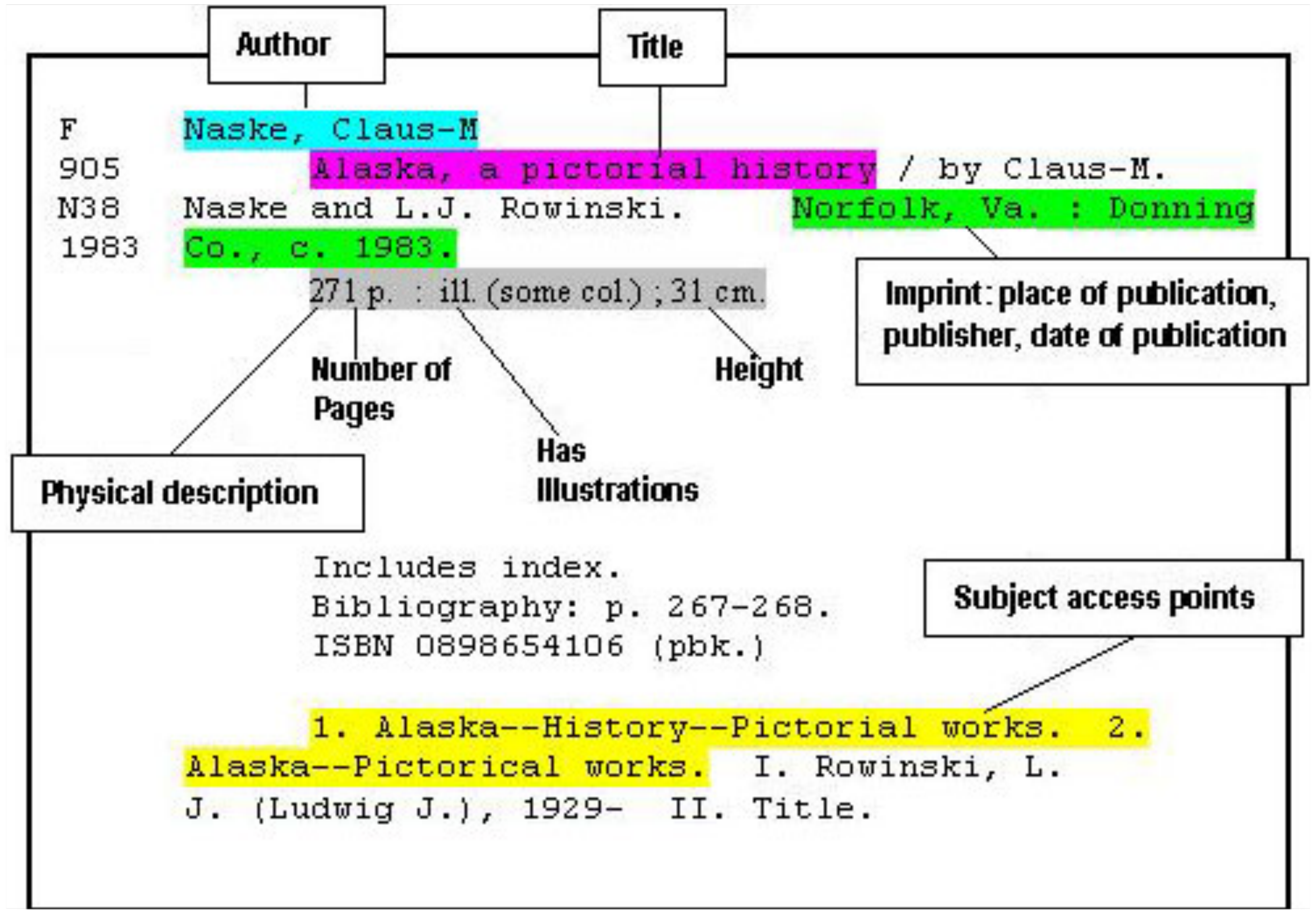
- Metadata => **data about data**
  - –forms an essential part of any database
    - providing **descriptive data** about each stored object, and
    - is the key to **organizing** and **managing** **data** objects
  - –critical for describing essential aspects of content:
    - main topics, author, language, publication, etc.
    - events, scenes, objects, times, places, etc.
    - rights, packaging, access control, content adaptation, …

What is Metadata?



- Metadata is structured information that
  - **describes**, **explains**, **locates**, or otherwise
  - makes it easier to **retrieve**, use, or
  - **manage** an information resource.

What is Metadata?



```
F       Naske, Claus-M
905            Alaska, a pictorial history / by Claus-M.
N38     Naske and L.J. Rowinski.      Norfolk, Va. : Donning
1983    Co., c. 1983.
               271 p. : ill. (some col.) ; 31 cm.

               Includes index.
               Bibliography: p. 267-268.
               ISBN 0898654106 (pbk.)

               1. Alaska--History--Pictorial works.  2.
        Alaska--Pictorical works.  I. Rowinski, L.
        J. (Ludwig J.), 1929-  II. Title.
```

Metadata Model

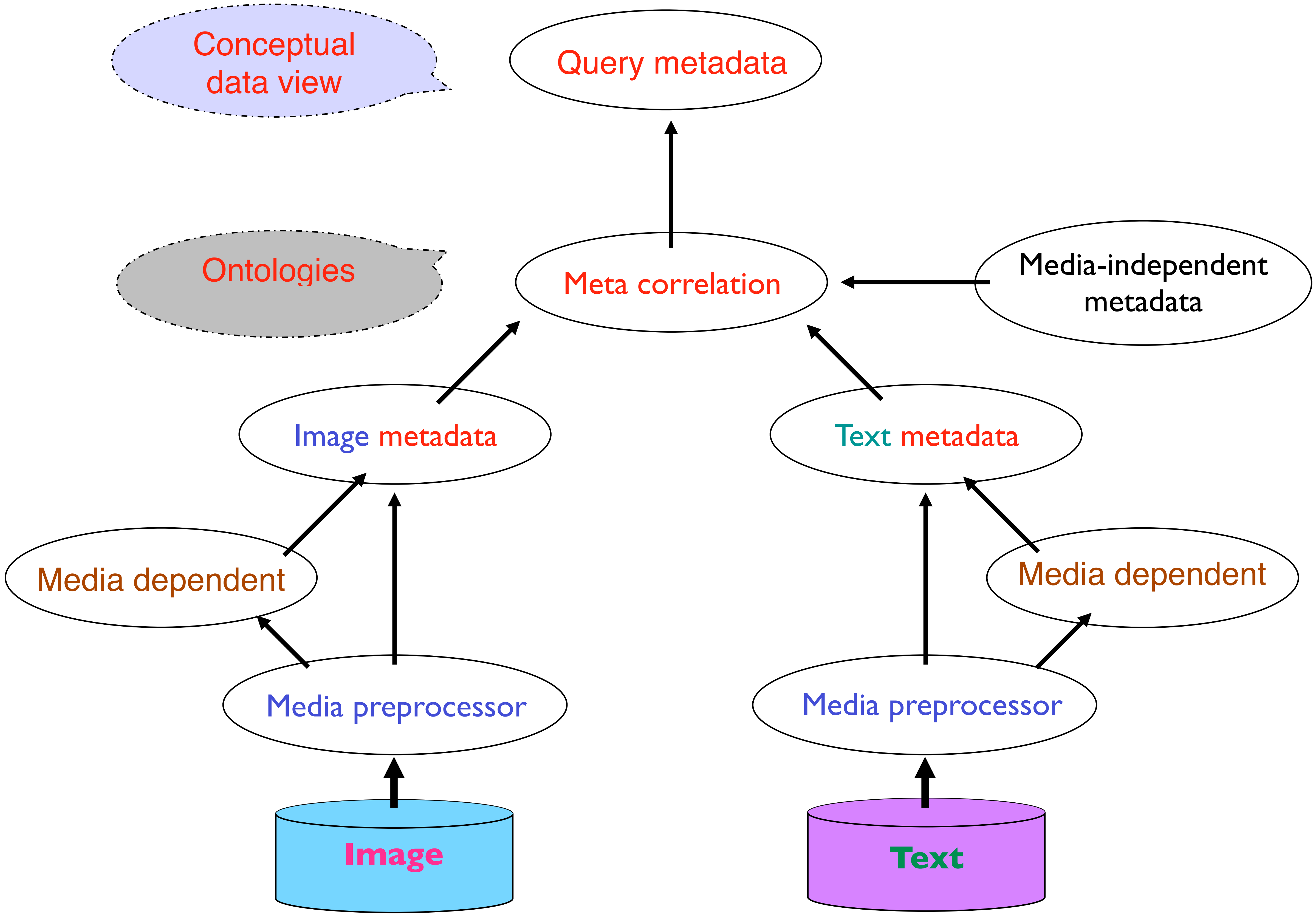
- **Purposes** of metadata:
  - **Administrative**
    - managing and administrating the data collection process
  - **Descriptive**
    - describing and identifying for retrieval purpose, creating indices
  - **Preservation**
    - managing data refreshing and migration
  - **Technical**
    - formats, compression, scaling, encryption, authentication and security
  - **Usage**
    - users, their level and type of use, user tracking, versioning (e.g., a high resolution version and corresponding thumbnail).

Metadata Model

Conformity （一致性） with **open metadata standard** will be a vital:

- Faster design and implementation
- Interoperability (互操作性） with broad field of competitive standards-based tools and systems
- Leveraging of rich set of standards-based technologies for critical functions
  - e.g., content extraction, advanced search, and personalization

The "role" of metadata in query processing:



Classifying Metadata

Classification of metadata can be:

1. Specific to the **media involved**
2. Specific to the **processing**
3. **Content** specific metadata

Image object	Text object	Video
Image capture	title	time based
Image storage	author	play rate
Caption	abstract	camera motion
Genre		camera lighting
Period	Full text indices	
Subjects		
Photographer		
IP rights		
Texture		

$\Bigg\}$ **Sample Metadata**

Metadata Classification

Metadata can be classified as:

- **Content dependent** (e.g., face features; used in CBR)
- **Content-descriptive** (used in TBR)
  1. Domain-independent metadata: independent of the application or subject topic
  2. Domain-dependent metadata: specific to the application area
- **Content-independent** (e.g., photographer's name; used in ABR)

- ABR: Auction-based Retrieval (基于竞价的检索，较少见)
- CBR: Content-based Retrieval
- TBR: Text-based Retrieval

Metadata Classification

Media	Content independent	Content descriptive	Content dependent
Text	status, location, date of update components	keywords, formats, categories, language	subtopic boundary word image spotting
speech	start, end time location confidence of word recognition	speakers	speech recognition speaker recognition prosodic cues change of meaning
Image	creator title date	keywords, formats	feature selection image features (e.g., histogram, segmentation)
Video	product title data distributor	camera shot action distance close-up	shot boundary frame features (e.g., histogram, motion lighting level, height)

Domain-dependent Metadata

- Standards for domain-specific metadata
  - **Digital geospatial metadata**
    - US Geographic Data Committee
    - http://www.fgdc.gov/metadata/metahome.html
  - **Environmental data (UDK)**
    - the European Environmental Catalog
  - **Product data exchange (PDES)**
    - an ANSI standard for the exchange of product model data
  - **Rich Site Summary (RSS)**
    - a lightweight XML vocabulary for describing websites, ideal for news syndication
  - **Medical information (HL7)**
    - provides specification for hospital records and medical information management
    - accredited by ANSI

More about RSS

- Rich Site Summary
- originally RDF Site Summary
- often called Really Simple Syndication

- a family of standard web feed formats to publish frequently updated information:
  - blog entries, news headlines, audio, video

More about RSS

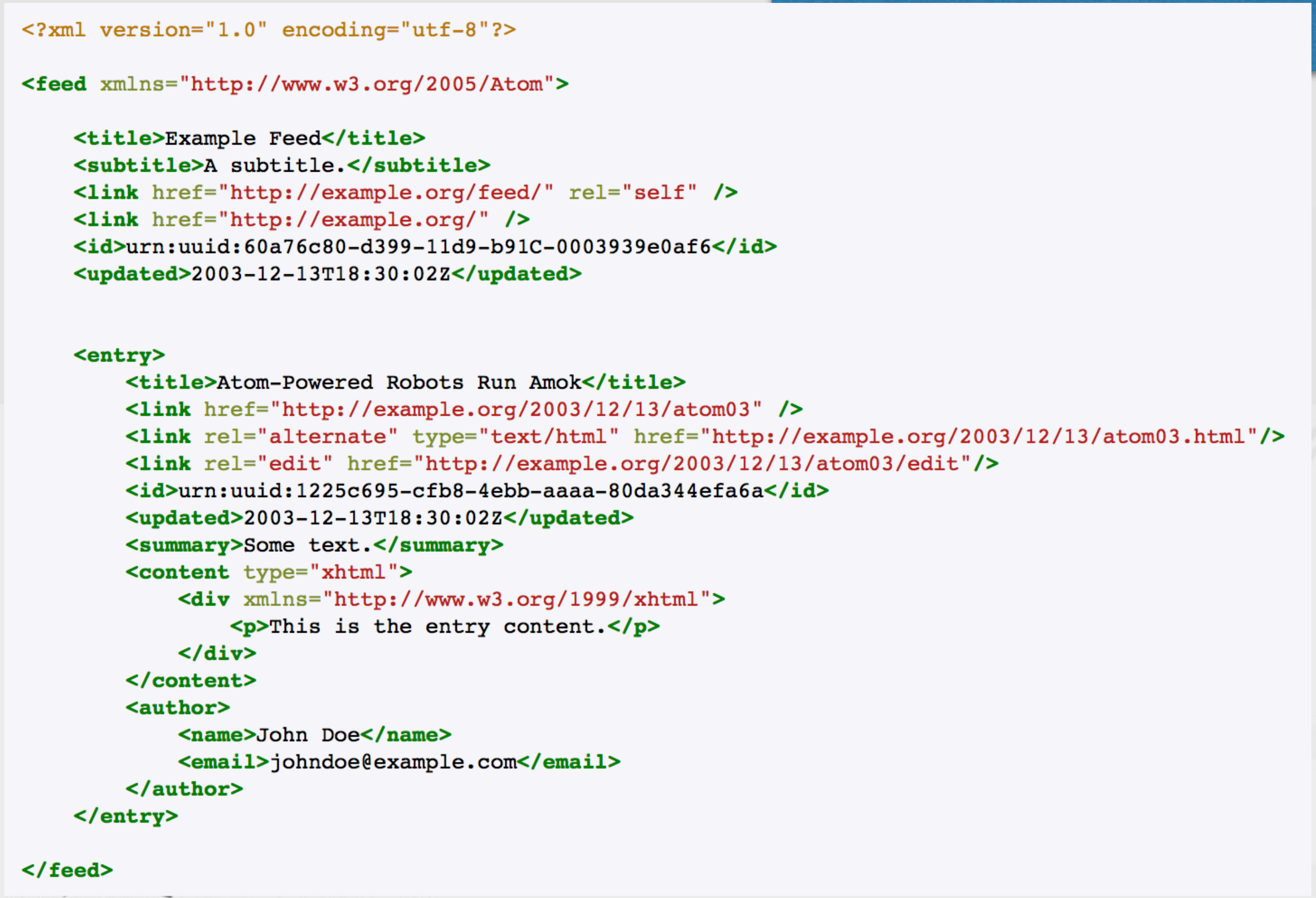
```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
 <title>RSS Title</title>
 <description>This is an example of an RSS feed</description>
 <link>http://www.example.com/main.html</link>
 <lastBuildDate>Mon, 06 Sep 2010 00:01:00 +0000 </lastBuildDate>
 <pubDate>Sun, 06 Sep 2009 16:20:00 +0000</pubDate>
 <ttl>1800</ttl>

 <item>
  <title>Example entry</title>
  <description>Here is some text containing an interesting description.</description>
  <link>http://www.example.com/blog/post/1</link>
  <guid isPermaLink="true">7bd204c6-1655-4c27-aeee-53f933c5395f</guid>
  <pubDate>Sun, 06 Sep 2009 16:20:00 +0000</pubDate>
 </item>

</channel>
</rss>
```

Alternative solution: ATOM

Atom Syndication Format
RFC 4287 (December 2005)
Atom Publishing Protocol
RFC 5023 (October 2007)



```
<?xml version="1.0" encoding="utf-8"?>

<feed xmlns="http://www.w3.org/2005/Atom">

    <title>Example Feed</title>
    <subtitle>A subtitle.</subtitle>
    <link href="http://example.org/feed/" rel="self" />
    <link href="http://example.org/" />
    <id>urn:uuid:60a76c80-d399-11d9-b91C-0003939e0af6</id>
    <updated>2003-12-13T18:30:02Z</updated>


    <entry>
        <title>Atom-Powered Robots Run Amok</title>
        <link href="http://example.org/2003/12/13/atom03" />
        <link rel="alternate" type="text/html" href="http://example.org/2003/12/13/atom03.html"/>
        <link rel="edit" href="http://example.org/2003/12/13/atom03/edit"/>
        <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
        <updated>2003-12-13T18:30:02Z</updated>
        <summary>Some text.</summary>
        <content type="xhtml">
            <div xmlns="http://www.w3.org/1999/xhtml">
                <p>This is the entry content.</p>
            </div>
        </content>
        <author>
            <name>John Doe</name>
            <email>johndoe@example.com</email>
        </author>
    </entry>

</feed>
```

Domain-independent Metadata Standards

- ISO/IEC 11179 (http://metadata-standards.org/11179/)
  - Intended to provide:
    - conceptual framework,
    - logical explanations of the processes for an organization to describe data semantics consistently, and
    - the exchange of data and metadata across organizational units
  - The standard divides data elements into 3 parts:
    - **Object class** – the thing the data describes (e.g., person, airplane)
    - **Property** – a peculiarity that describes/distinguishes objects
    - **Representation** – the allowed values and other information

Domain-independent Metadata Standards

- ISO/IEC 11179

Attribute	Description
Name	the label assigned to the **data element (d.e.)**
Id	the unique identifier assigned to the d.e.
Version	the version of the d.e. (e.g., 1.1 for Dublin Core)
Registration Authority	the entity authorized to register the d.e.
Language	the language in which the d.e. is specified (e.g., English)
Definition	a statement representing the d.e. concept and nature
Obligation	indicates if the d.e. is required to be not null
Data type	indicates the data type that can be represented in d.e.
Maximum Occurrence	indicates any limit to the repeatability of the d.e.
Comment	a remark concerning the application of the d.e.

Domain-independent Metadata Standards

- The Dublin Core Metadata set
  http://purl.org/metadata/dublin_core
  - Originally for resource description records of **online libraries** over Internet
  - version 1.1
    - broaden to other media with a link to the ISO/IEC 11179 standard
      - Each Dublin Core element is defined using a set of 10 attributes from the ISO/IEC 11179
      - Six of them are common to all the Dublin Core element (3-5, 7-9)
    - 15 metadata elements (the Dublin Core) has been proposed
      - which are suggested to be the minimum number of metadata elements to support retrieval of a document-like object (DLO) in a networked environment

The Dublin Core Metadata set

ID	Core element	Semantics
1	Subject	topic addressed by the work
2	Title	the name of the object
3	Creator	entity responsible for the intellectual content
4	Publisher	the agency making the object available
5	Description	an account of the content of the resource
6	Contributor	an entity making contributions to the resource content
7	Date	associated with an event in the life cycle of the resource
8	Resource type	the nature/genre of the resource content
9	Format	physical/digital manifestation of the resource; format of the file (e.g., postscript)
10	Id	unique identifier
11	Relation	a reference to a related resource
12	Source	a ref. to a resource from which the current resource is derived
13	Language	language of the intellectual content
14	Coverage	extent/scope of the resource content; typically include location, period
15	Rights	Information about rights held in and over the resource

Domain-independent Metadata Standards

- Resource Description Framework (RDF)
  - Being developed by the W3C as a foundation for processing metadata
  - Allows multiple metadata schemes to be read by human and parsed by machines
  - Specific objectives include:
    - **Resource discovery** – to provide better search engine capabilities
    - **Cataloging** – for describing the content and relationships available through intelligent software agents
    - **Content rating** – describing collection of pages that represent a single logical "document"
    - **IP rights** – describing the intellectual property of web pages
    - **Privacy preferences and policies** – for users and website
    - **Digital signatures** – to create a "web of trust" for e-commerce, collaboration, and other applications

Resource Description Framework (RDF)

- The formal model of the **RDF framework**:
  - *Resources* (set).
  - *Literals* (set).
  - a subset of resources called *Properties*
  - There is a set called *Statements*, each element of which is a triple of form <**pred, sub, obj**>, where
    - **pred** is a property,
    - **sub** is a resource (member of *Resources*)
    - **obj** is either a resource or a literal
- The preferred language for writing RDF schemas is XML

XML

- Defined by the WWW Consortium (W3C)
- Originally intended as a document markup language not a database language
  - Documents have tags giving extra information about sections of the document
    - <title> XML </title> <slide> Introduction …</slide>
    - <?xml … ?> (document declaration)
    - <!-- definition of elements --> (comments)
  - Derived from SGML (Standard Generalized Markup Language), but simpler to use than SGML
  - **Extensible**, unlike HTML
    - Users can add new tags, and *separately* specify how the tag should be handled for display

XML

– The ability to specify new tags, and to create nested tag structures made XML a great way to exchange **data**, not just documents.
  - Much of the use of XML has been in data exchange applications, not as a replacement for HTML
– Tags make data (relatively) self-documenting

```
<bank>
   <account>
      <account-number> A-101     </account-number>
      <branch-name>      Downtown </branch-name>
      <balance>              500        </balance>
   </account>
    <depositor>
      <account-number> A-101    </account-number>
      <customer-name> Johnson </customer-name>
   </depositor>
</bank>
```

Structure of XML

- **Tag**: label for a section of data
- **Element**: section of data beginning with **<tagname>** and ending with matching **</tagname>**
- Elements must be properly ***nested***
  - ✓Proper nesting

    <account> … <balance> …. </balance> </account>

  - ➡Improper nesting

    <account> … <balance> …. </account> </balance>

  - Formally: every start tag must have a unique matching end tag, that is in the context of the same parent element.
- Every document must have a single top-level element

Structure of XML

– Mixture of text with sub-elements is legal in XML

- Example:

```
<account>
    This account is seldom used any more.
     <account-number> A-102</account-number>
     <branch-name> Perryridge</branch-name>
     <balance>400 </balance>
</account>
```

- Useful for document markup, but discouraged for data representation

Attributes

- Elements can have **attributes**

```
<account acct-type = “checking” >
      <account-number> A-102 </account-number>
        <branch-name> Perryridge </branch-name>
        <balance> 400 </balance>
 </account>
```

- Attributes are specified by *name*=*value* pairs inside the starting tag of an element
- An element may have several attributes, but each attribute name can only occur once

```
<account  acct-type = “checking”  monthly-fee=“5”>
```

Attributes vs. Subelements

– Distinction between subelement and attribute
  - In the context of documents
    - attributes: are part of markup
    - subelements: contents are part of the basic document contents
- Some information can be represented in two ways
  - **<account account-number = “A-101”> …. </account>**
  - **<account>**
    <account-number>A-101</account-number> …
    </account>

attribute

subelement

- Suggestion: use attributes for identifiers of elements, and use subelements for contents

More on XML Syntax

– Elements without subelements or text content can be abbreviated by ending the start tag with a /> and deleting the end tag
  - <account number=“A-101” branch=“Perryridge” balance=“200“ />
– To store string data that may contain tags, without the tags being interpreted as subelements, use CDATA as below
  - <![CDATA[<account> … </account>]]>

    Here, <account> and </account> are treated as just strings

Namespaces

– XML data has to be exchanged between organizations
– Same tag name may have different meaning in different organizations, causing confusion on exchanged documents
– Specifying a unique string as an element name avoids confusion
– Avoid using long unique names all over document by using XML Namespaces

```
<bank Xmlns:FB='http://www.FirstBank.com'>
  …
   <FB:branch>
              <FB:branchname>Downtown</FB:branchname>
        <FB:branchcity>   Brooklyn   </FB:branchcity>
   </FB:branch>
  …
</bank>
```

XML Document Schema

- Database schemas constrain
  - what information can be stored, and
  - the data types of stored values
- not necessary in a XML document
- very important for XML data exchange
  - Otherwise, a site cannot automatically interpret data received from another site
- **Two mechanisms** for specifying XML schema
  - Document Type Definition (**DTD**)
  - **XML Schema**

XML Document Schema

- The type of an XML document can be specified using a DTD
- DTD constraints structure of XML data
  - What elements can occur
  - What attributes can/must an element have
  - What subelements can/must occur inside each element, and how many times.
- DTD does not constrain data types
  - All values represented as strings in XML
- DTD syntax
  - <!ELEMENT element (subelements-specification) >
  - <!ATTLIST element (attributes) >

Element Specification in DTD

– Subelements can be specified as
  - names of elements, or
  - #PCDATA (parsed character data), i.e., character strings
  - EMPTY (no subelements) or ANY (anything can be a subelement)

– Example

```
<! ELEMENT depositor (customer-name  account-number)>
<! ELEMENT customer-name (#PCDATA)>
<! ELEMENT account-number (#PCDATA)>
```

– Subelement specification may have regular expressions

```
<!ELEMENT bank ( ( account | customer | depositor)+)>
```

– Notation:
  - » “|” - alternatives
  - » “+” - 1 or more occurrences
  - » “*” - 0 or more occurrences

IDs and IDREFs

- An element can have at most one attribute of type ID
- The **ID attribute value** of each element in an XML document must be **distinct**
  - Thus the ID attribute value is an object identifier

- An attribute of type IDREF must contain the ID value of an element in the same document
- An attribute of type IDREFS contains a set of (0 or more) ID values.
- Each ID value must contain the ID value of an element in the same document

Bank DTD with ID and IDREF attribute types

```
<!DOCTYPE bank-2[
    <!ELEMENT account (branch, balance)>
    <!ATTLIST account
            account-number ID       # REQUIRED
            owners           IDREFS # REQUIRED>
    <!ELEMENT customer(customer-name, customer-street,
                                          customer-city)>
    <!ATTLIST customer
            customer-id    ID       # REQUIRED
            accounts       IDREFS # REQUIRED>
    … declarations for branch, balance, customer-name,
                          customer-street and customer-city
]>
```

XML data with ID and IDREF attributes

```
<bank-2>
        <account account-number="A-401" owners="C100 C102">
                <branch-name> Downtown </branch-name>
                <balance>          500 </balance>
        </account>
        <customer customer-id="C100" accounts="A-401">
                <customer-name>Joe          </customer-name>
                <customer-street> Monroe   </customer-street>
                <customer-city>      Madison</customer-city>
        </customer>
        <customer customer-id="C102" accounts="A-401 A-402">
                <customer-name> Mary       </customer-name>
                <customer-street> Erin         </customer-street>
                <customer-city>      Newark </customer-city>
        </customer>
</bank-2>
```

Limitations of DTDs

– No typing of text elements and attributes
  - All values are strings, no integers, reals, etc.

– Difficult to specify unordered sets of subelements
  - Order is usually irrelevant in databases
  - (A | B)* allows specification of an unordered set, but
    - Cannot ensure that each of A and B occurs only once

– IDs and IDREFs are untyped
  - The *owners* attribute of an account may contain a reference to another account, which is meaningless
    - *owners* attribute should ideally be constrained to refer to customer elements

Dublin Core Data in XML

- http://dublincore.org/documents/dc-xml-guidelines/

```
<?xml version="1.0"?>
<metadata
  xmlns="http://example.org/myapp/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://example.org/myapp/ http://example.org/myapp/schema.xsd"
  xmlns:dc="http://purl.org/dc/elements/1.1/">

  <dc:title>
    UKOLN
  </dc:title>
  <dc:description>
    UKOLN is a national focus of expertise in digital information
    management. It provides policy, research and awareness services
    to the UK library, information and cultural heritage communities.
    UKOLN is based at the University of Bath.
  </dc:description>
  <dc:publisher>
    UKOLN, University of Bath
  </dc:publisher>
  <dc:identifier>
    http://www.ukoln.ac.uk/
  </dc:identifier>
</metadata>
```

Alternative solution: metadata.JSON

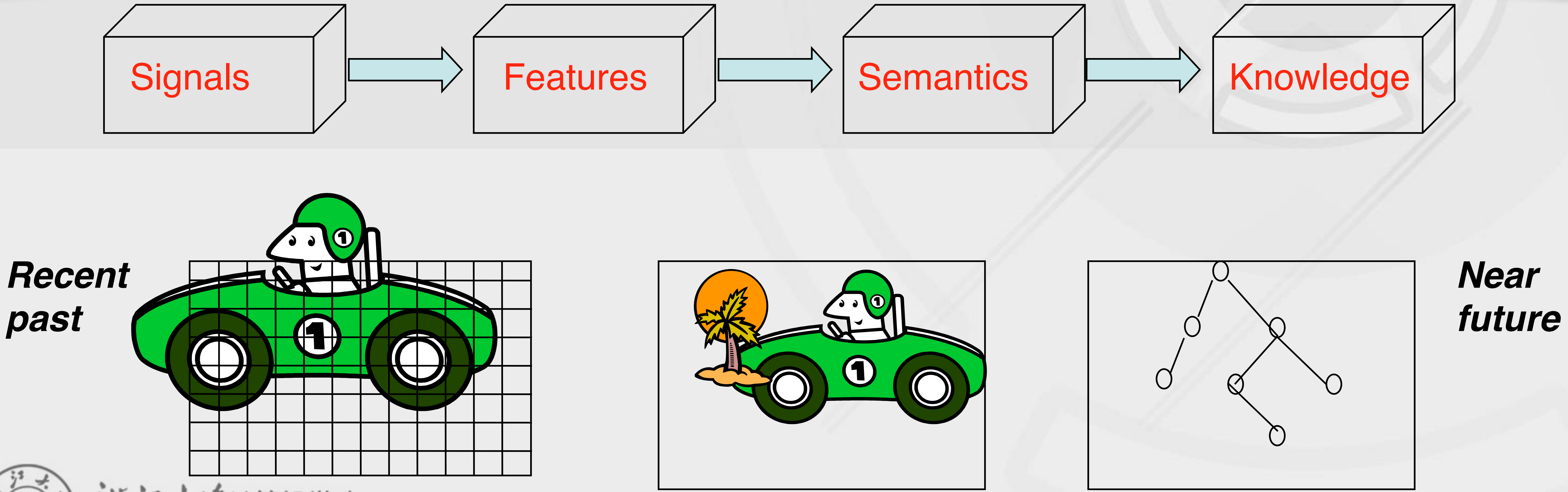
Generic metadata example

```
{
  // (rest of the document object)
  "metadata": {
    "field1__double": [ 1.0 ], // (single atomic type)
    "field2": [ "1", "2", "3", "4" ], // (array of atomic types)
    "field3": [ { "type": "simple" } ], // (single simple object)
    "field4": [ { "type": { "nested": true } } ], // (single nested object)
    "field3": [ { "type": "simple", "index": 1 }, { "type": { "nested": true }, "index": 2 } ], // (array of objects)
    // etc
  }
}
```

浙江大学计算机学院
数字媒体与网络技术

Domain-dependent Metadata Standards

- MPEG series
  - Moving Picture Experts Group (MPEG) since 1998
  - responsible for developing standards of the coded representation of moving pictures and associated audio



Domain-dependent Metadata Standards

Applications			
MPEG-1,-2,-4 Video storage Broadband, streaming video delivery	MPEG-4,-7 CBR Multimedia filtering Content adaptation	MPEG-7 Semantic-based retrieval and filtering Intelligent media services (iTV)	MPEG-21 Multimedia framework e-Commerce
Problems and Innovations			
Compression coding communications	Similarity search object- & feature- based coding	Modeling & classifying, personalization, summarization	Media mining, decision support

MPEG-1,-2 ,

MPEG-4 ,

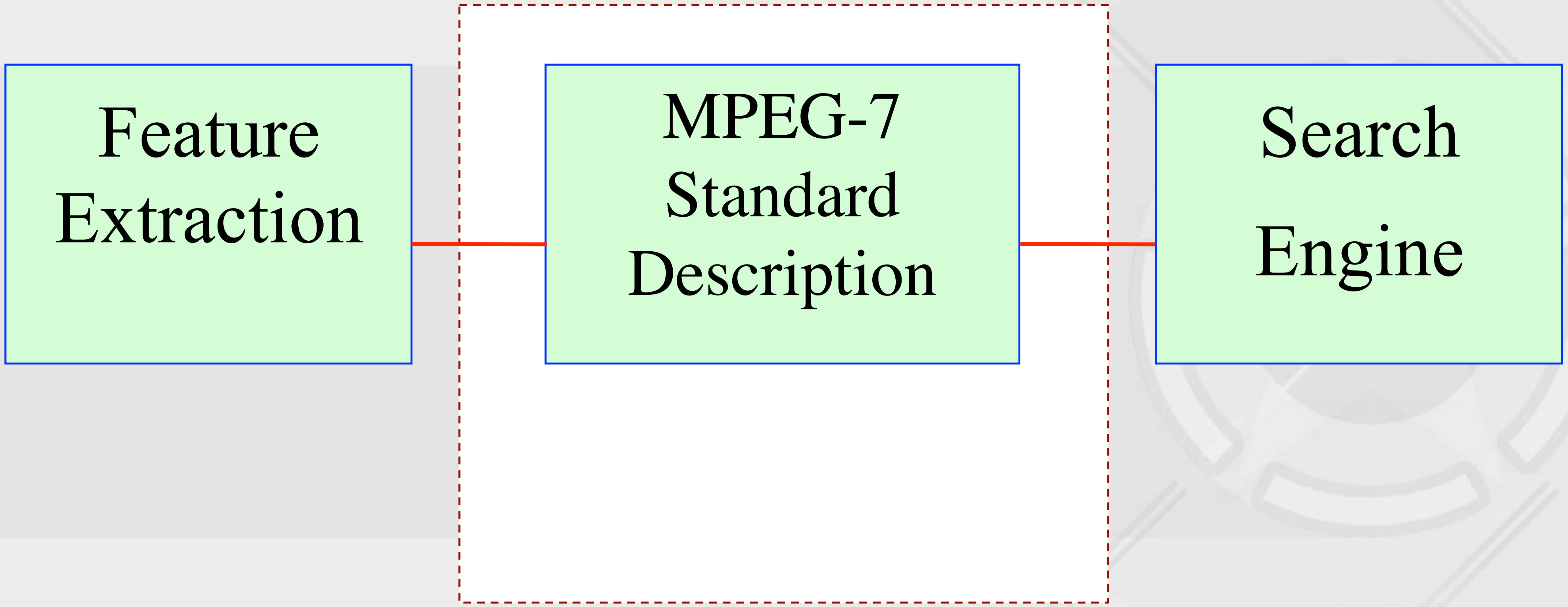
MPEG-7 ,

MPEG-21 ,

MPEG-7

- ***Multimedia Content Description Interface***
  - Representation of information **about** the content
    - still pictures, graphics, 3D models, audio, speech, video & their combination
  - Goal:
    - to **support efficient search** for multimedia content using **standardized descriptions**
    - desirable to use textual information for the descriptions

Domain-independent Metadata Standards

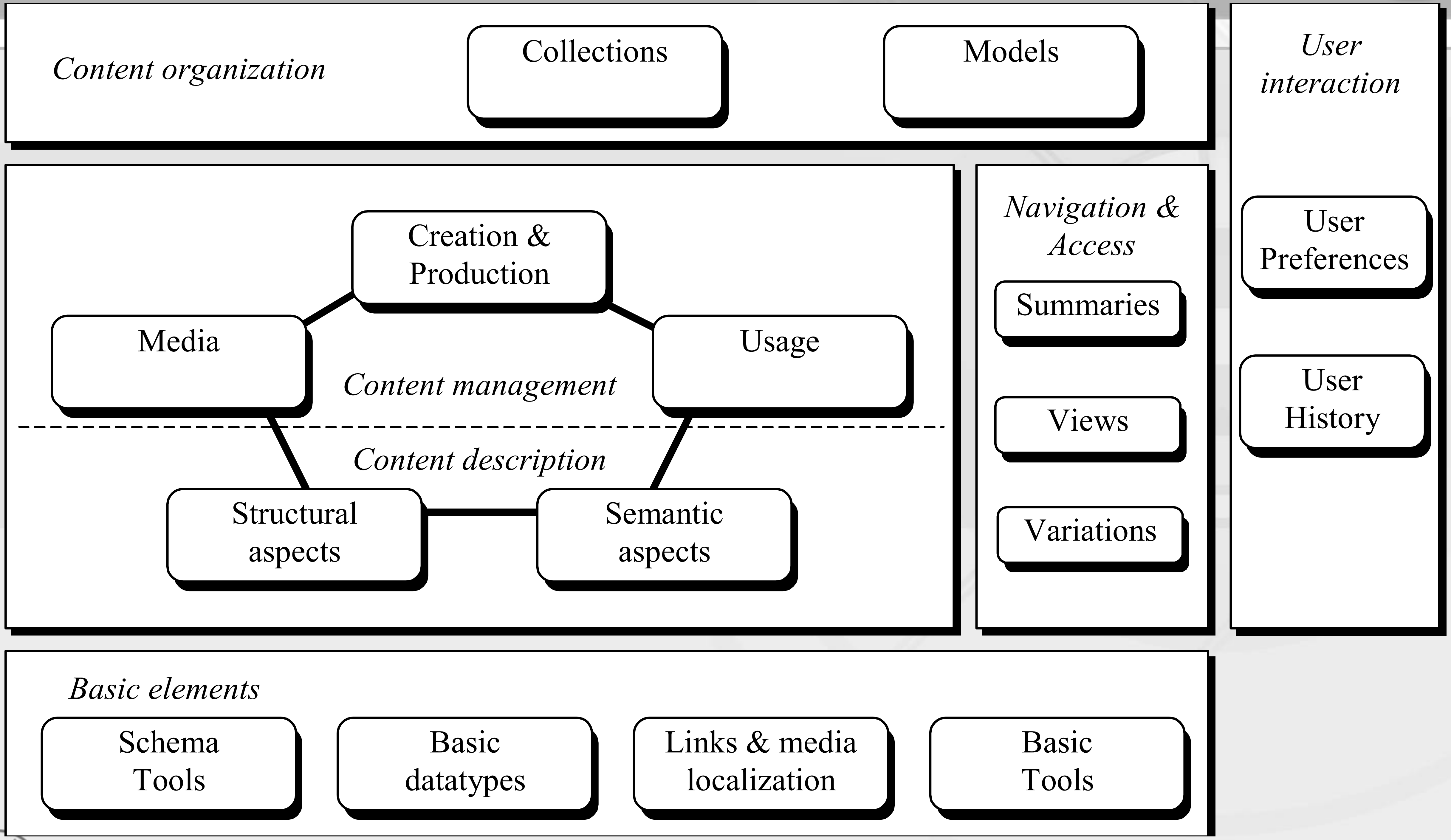


Scope of MPEG-7

MPEG-7

Set of description tools	Functionality
Media	Description of the storage media: typical features include the storage format, the encoding of the multimedia content, the identification of the media. Note that several instances of storage media for the same multimedia content can be described.
Creation & Production	Meta information describing the creation and production of the content: typical features include title, creator, classification, purpose of the creation, etc. This information is most of the time author generated since it cannot be extracted from the content.
Usage	Meta information related to the usage of the content: typical features involve rights holders, access right, publication, and financial information. This information may very likely be subject to change during the lifetime of the multimedia content.
Structural aspects	Description of the multimedia content from the viewpoint of its structure: the description is structured around segments that represent physical spatial, temporal or spatial-temporal components of the multimedia content. Each segment may be described by signal-based features (color, texture, shape, motion, and audio features) and some elementary semantic information.
Semantic aspects	Description of the multimedia content from the viewpoint of its semantic and conceptual notions. It relies on the notions of objects, events, abstract notions and their relationship.

MPEG-7



Content organization
Collections
Models
User interaction
Creation & Production
Media
Usage
Content management
Content description
Structural aspects
Semantic aspects
Navigation & Access
Summaries
Views
Variations
User Preferences
User History
Basic elements
Schema Tools
Basic datatypes
Links & media localization
Basic Tools

MPEG-7 Standard Elements

- **Descriptors** (Ds)
  - describe features, attributes, or groups of attributes of MM content
- **Description Schemes** (DSs)
  - a DS specifies the structure and semantics of the components (which may be other DSs, Ds, or datatypes)
- **Datatypes**
- **Classification Schemes** (CS):
  - lists of defined terms and meanings
- **System Tools**
- **Extensibility**
  - e.g., new DS’s and D’s; registration authority for CS

Outline

1. MM content organization

2. MM database system architecture

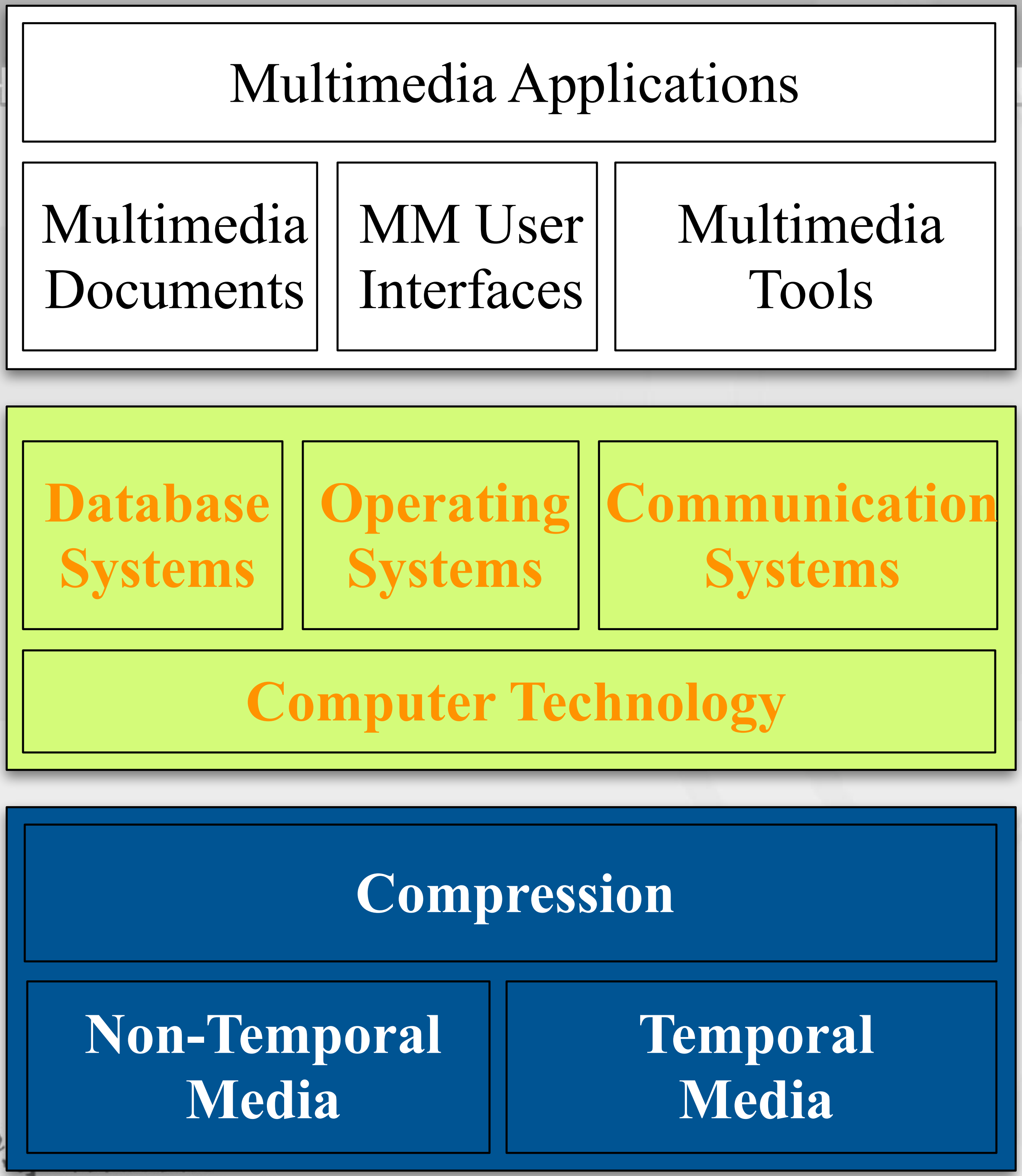
3. MM system service model

4. Multimedia Data Storage

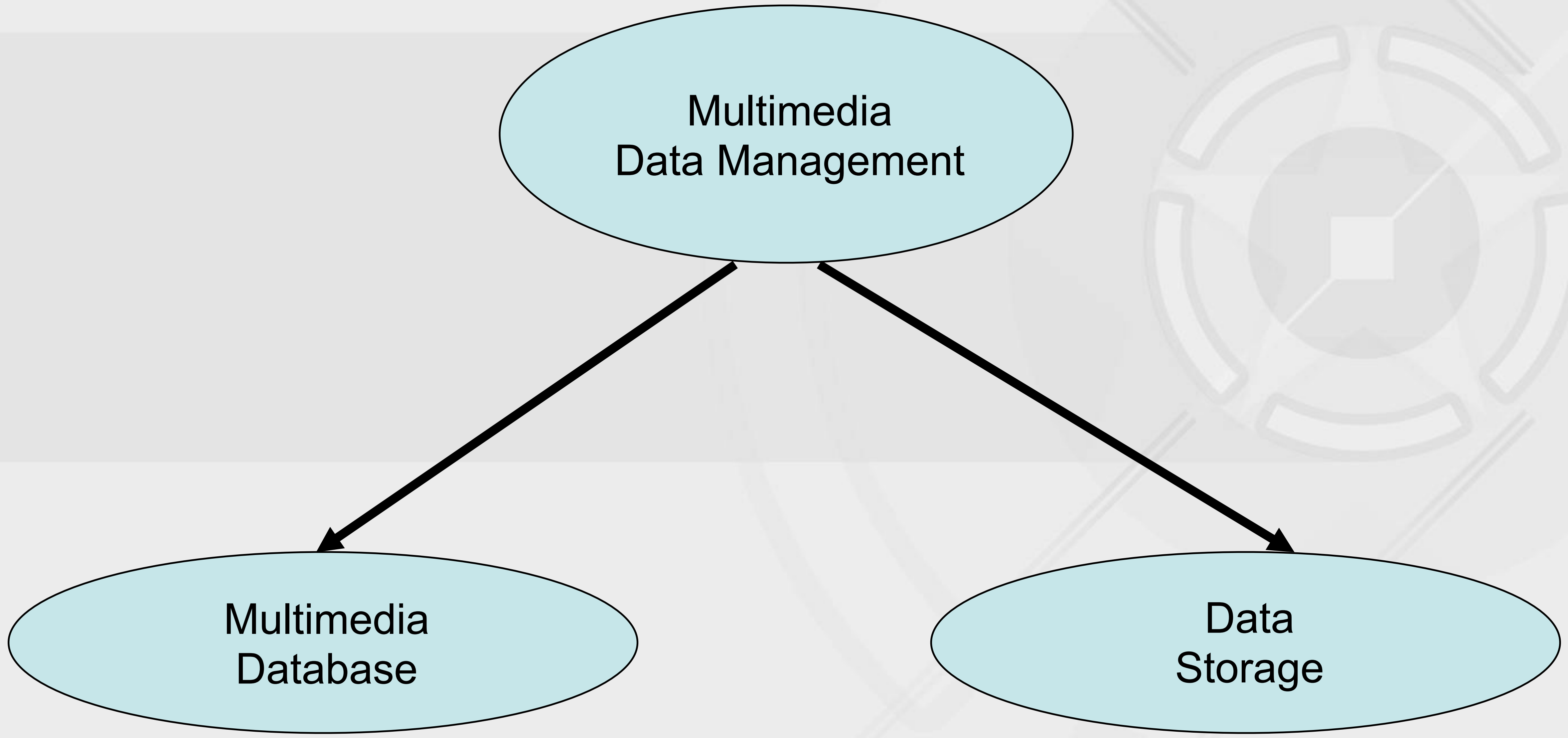
5. Multimedia application

3.2 Multimedia Database System Architecture

Multimedia Architecture



Multimedia Database System

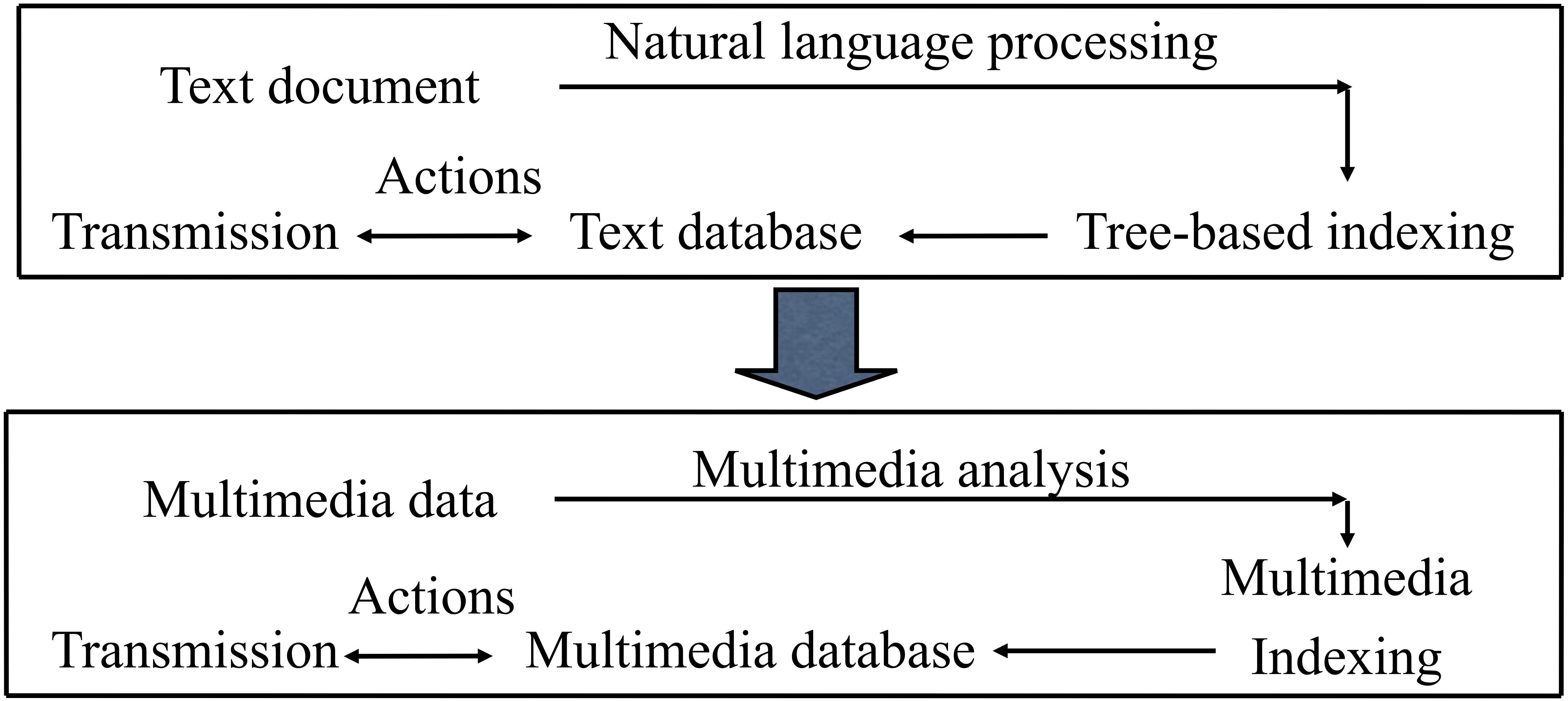


Multimedia Database System

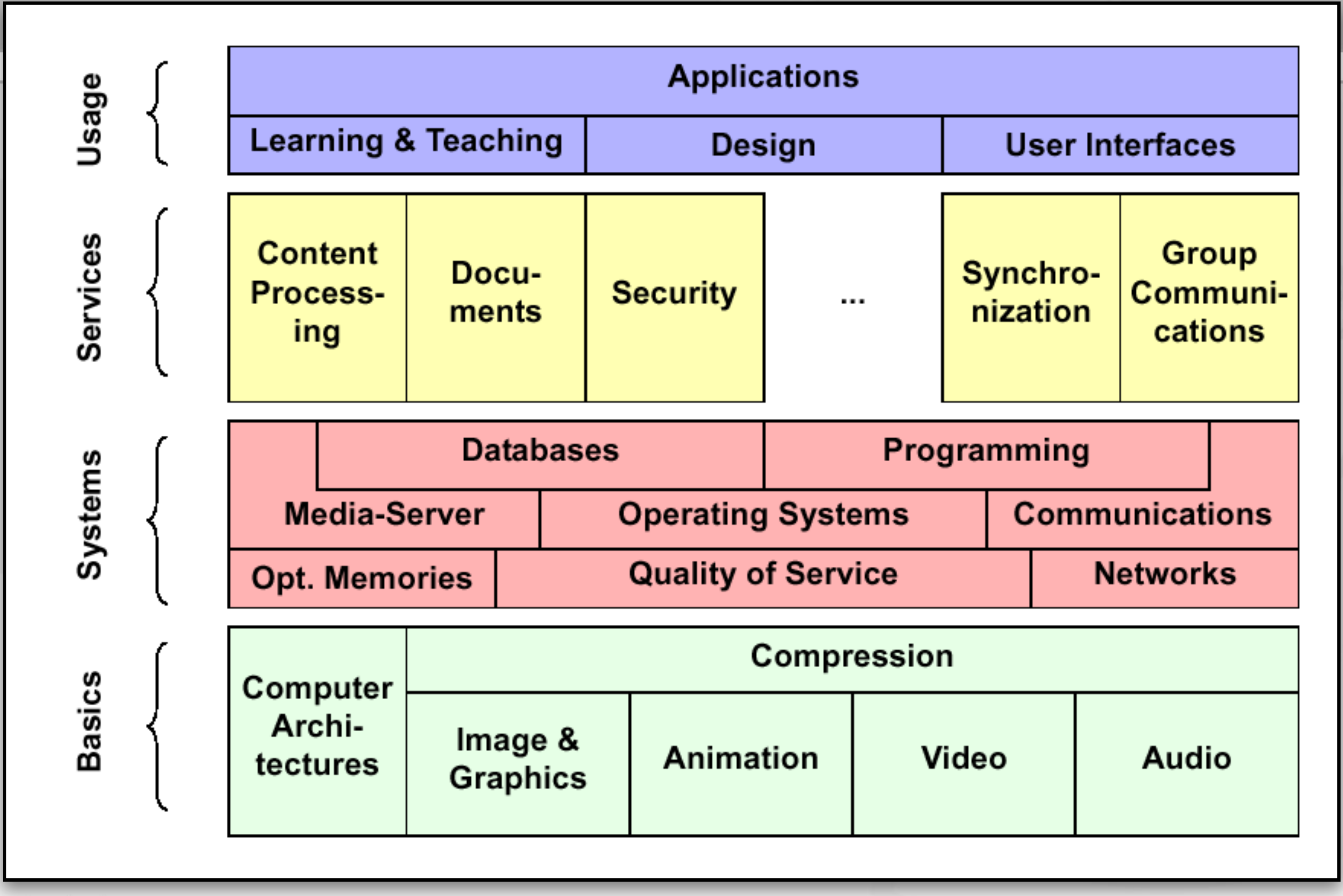
- **Multimedia database** v.s. **text database**
  - **Temporal data**: Requires temporal modeling
  - **Huge amount of data**: Compression helps get around this.
  - Data is **not easily indicative** of the information
  - Requires a lot of **pre-processing** in order to store data efficiently:
    - PCA, feature extraction and segmentation
  - **Novel Query mechanisms**
  - **Hypermedia**: The ability to interactively move around in the data.

How to Build Multimedia Database Systems?

How to build text database? *Yahoo, Google*



Scope



Usage
Applications
Learning & Teaching
Design
User Interfaces
Services
Content Process-ing
Docu-ments
Security
...
Synchro-nization
Group Communi-cations
Systems
Databases
Programming
Media-Server
Operating Systems
Communications
Opt. Memories
Quality of Service
Networks
Basics
Computer Archi-tectures
Compression
Image & Graphics
Animation
Video
Audio

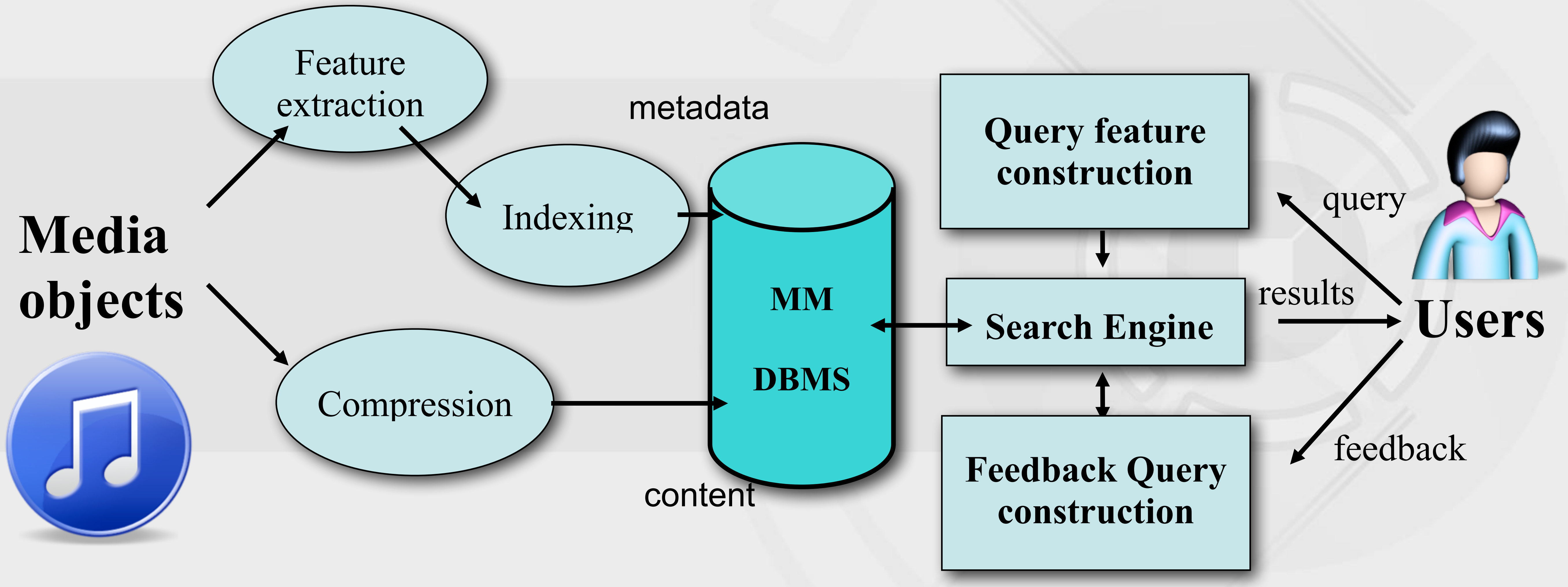
A Reference Architecture for MMDB System

- **Considerations:**
  - **Real time aspects/constraints impose strong demands on the systems**
    - Simultaneous presentation of multimedia objects may cause performance problems.
  - **Data Sharing**
    - Due to the possibly very large multimedia data, traditional replicated data technique may not be applicable, hence data sharing is essential
  - **Multiple Client/ Multiple Server Architecture**

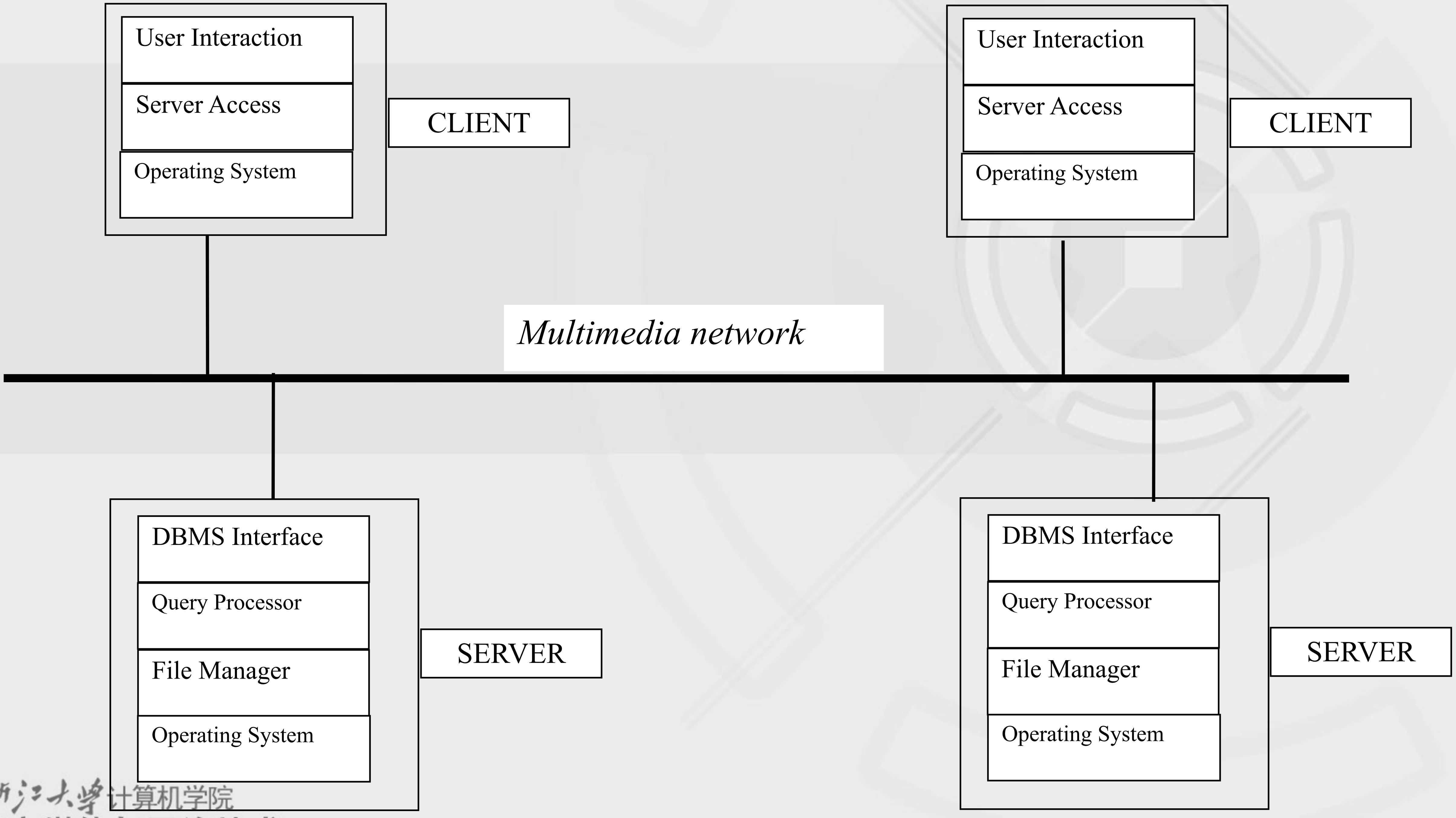
A Reference Architecture for MMDB System

- **Considerations:**
  - **Real time aspects/constraints**
  - **Data Sharing**
  - **Multiple Client/ Multiple Server Architecture**
    - Many multimedia applications work with data that are stored on remote sites (e.g, VOD, tele-learning), which suggests for client / server architecture.
    - A **client** consists of **three** layers…
      - **User Interaction** – takes care of input and output of multimedia data
      - **Server Access** – allows searching of servers by the client
      - **Operating System** – not a real part of the MMDBS
    - A **server** consists of **four** layers:
      - **DBMS Interface**
      - **Query Processor**
      - **File Manager**
      - **Operating System**

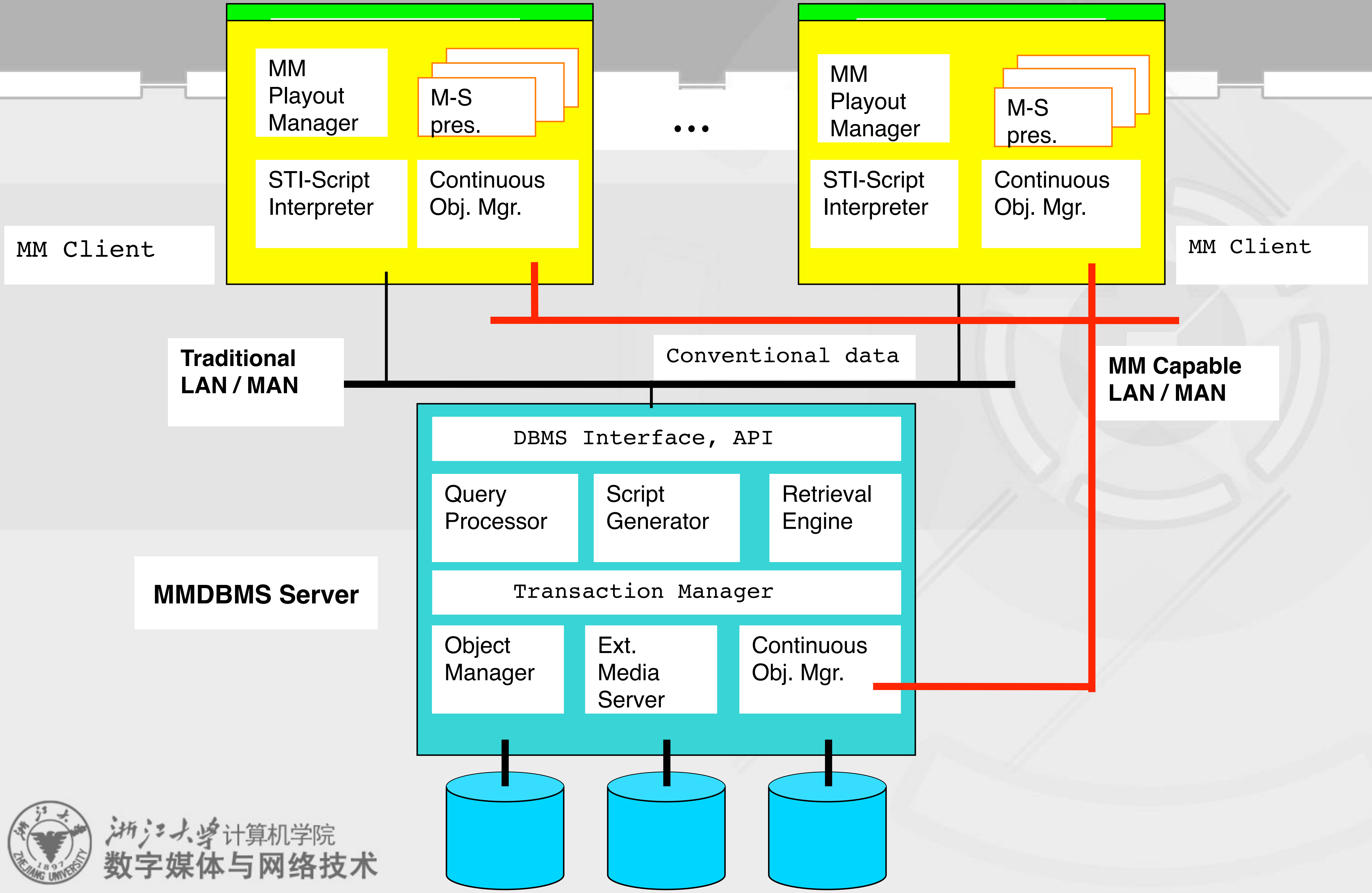
A Generic Architecture of MMDBMS



MMDB Reference Architecture: "Simplified View"



Detailed View of MMDB Architecture

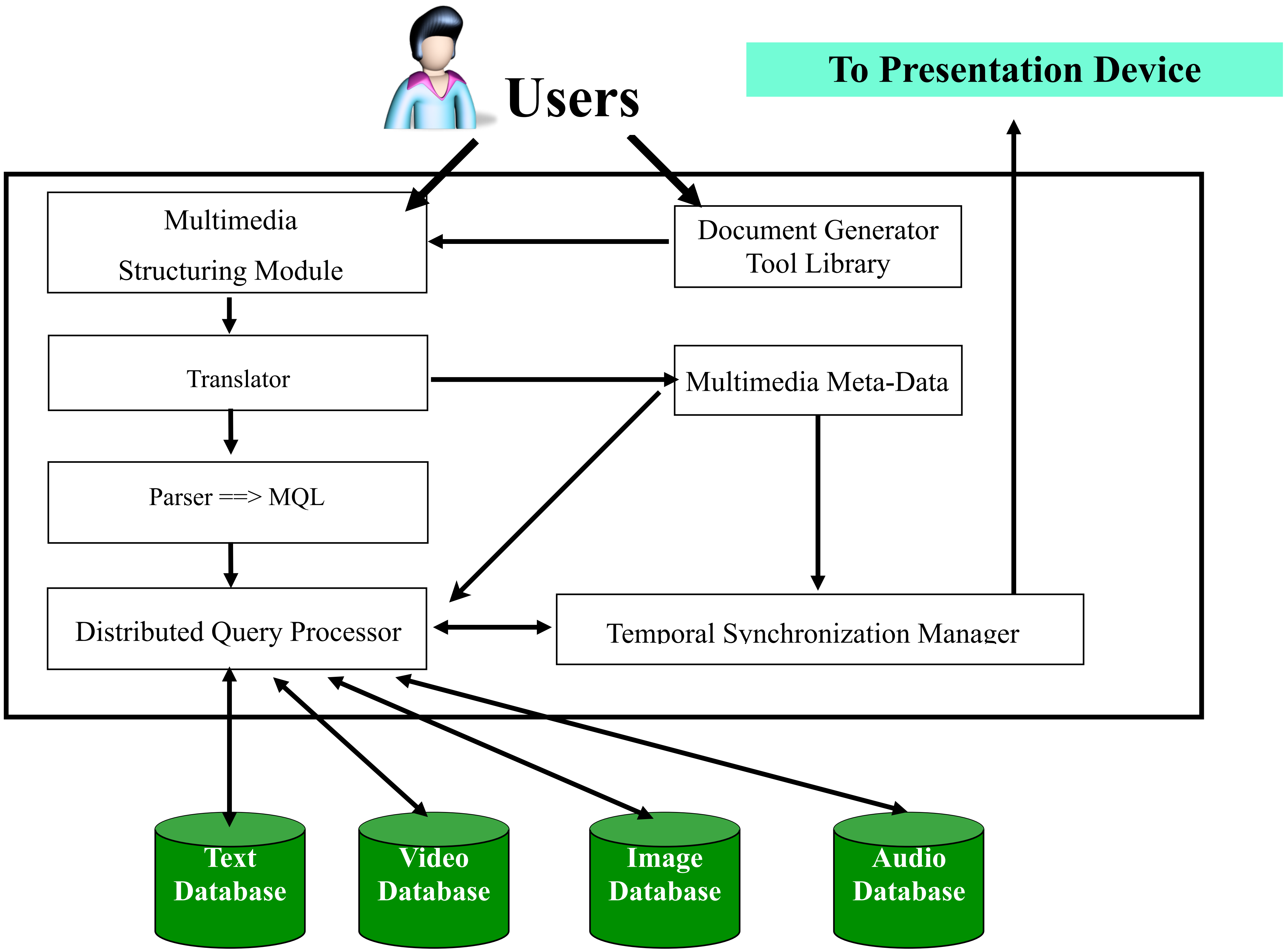


MMDBMS Development

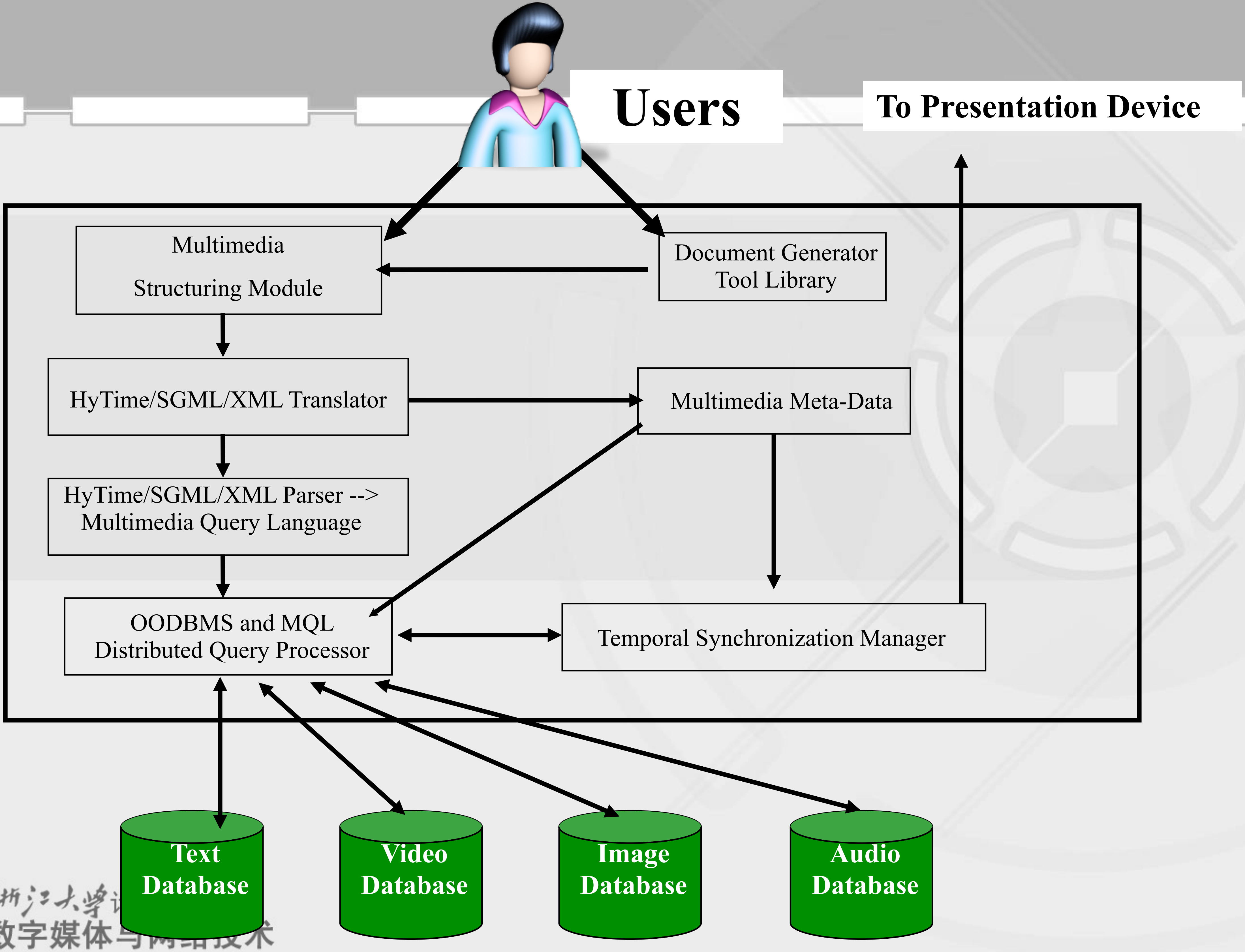
Major steps in developing MMDBMS

1. **Media acquisition:**
   - collect media data from various sources, such as WWW, CD, TV, etc.
2. **Media processing:**
   - extract media representations and their features, including noise filtering, rending, etc.
3. **Media storage:**
   - store the data and their features in the system based on application requirement.
4. **Media organization:**
   - organize the features for retrieval. i.e., indexing the features with effective structures.
5. **Media query processing:**
   - Accommodated with indexing structure, efficient search algorithm with similarity function should be designed.

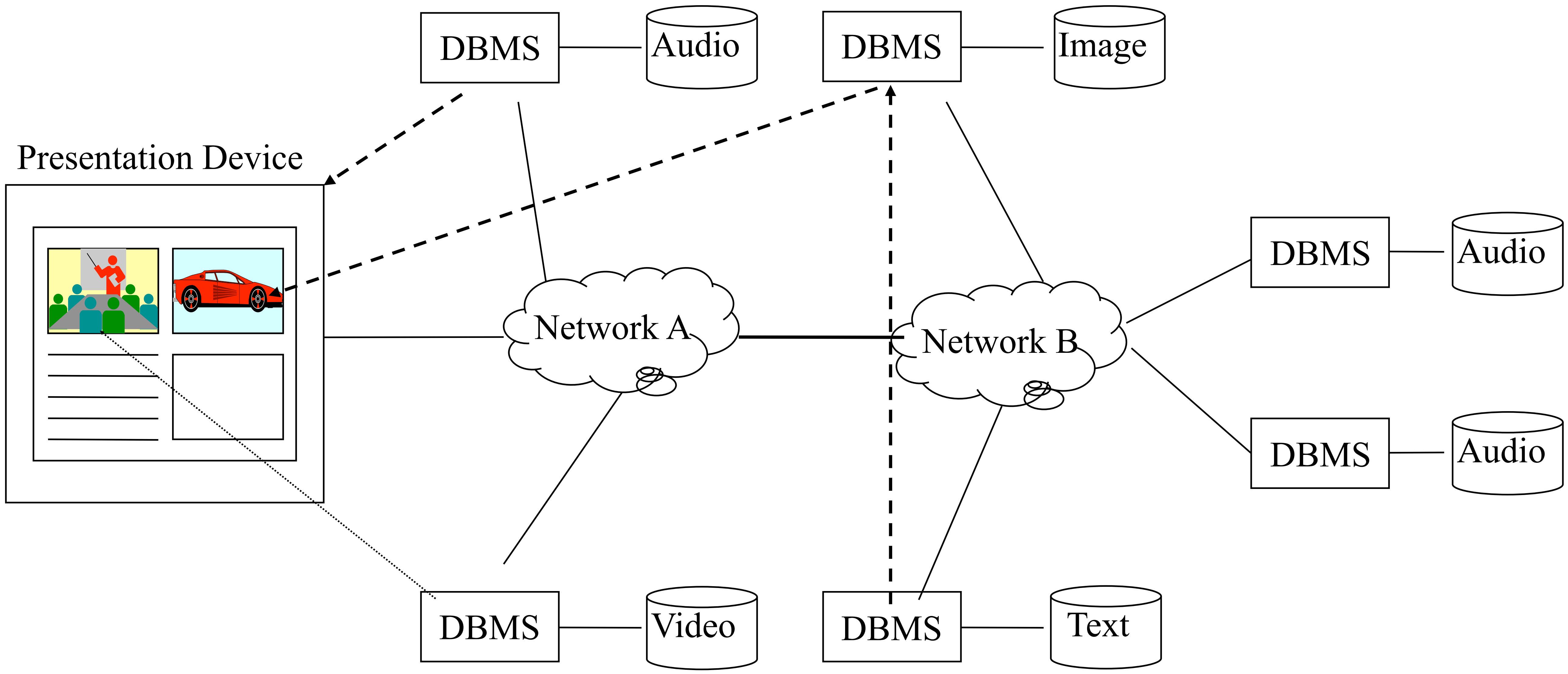
Software Architecture of MMDBMS



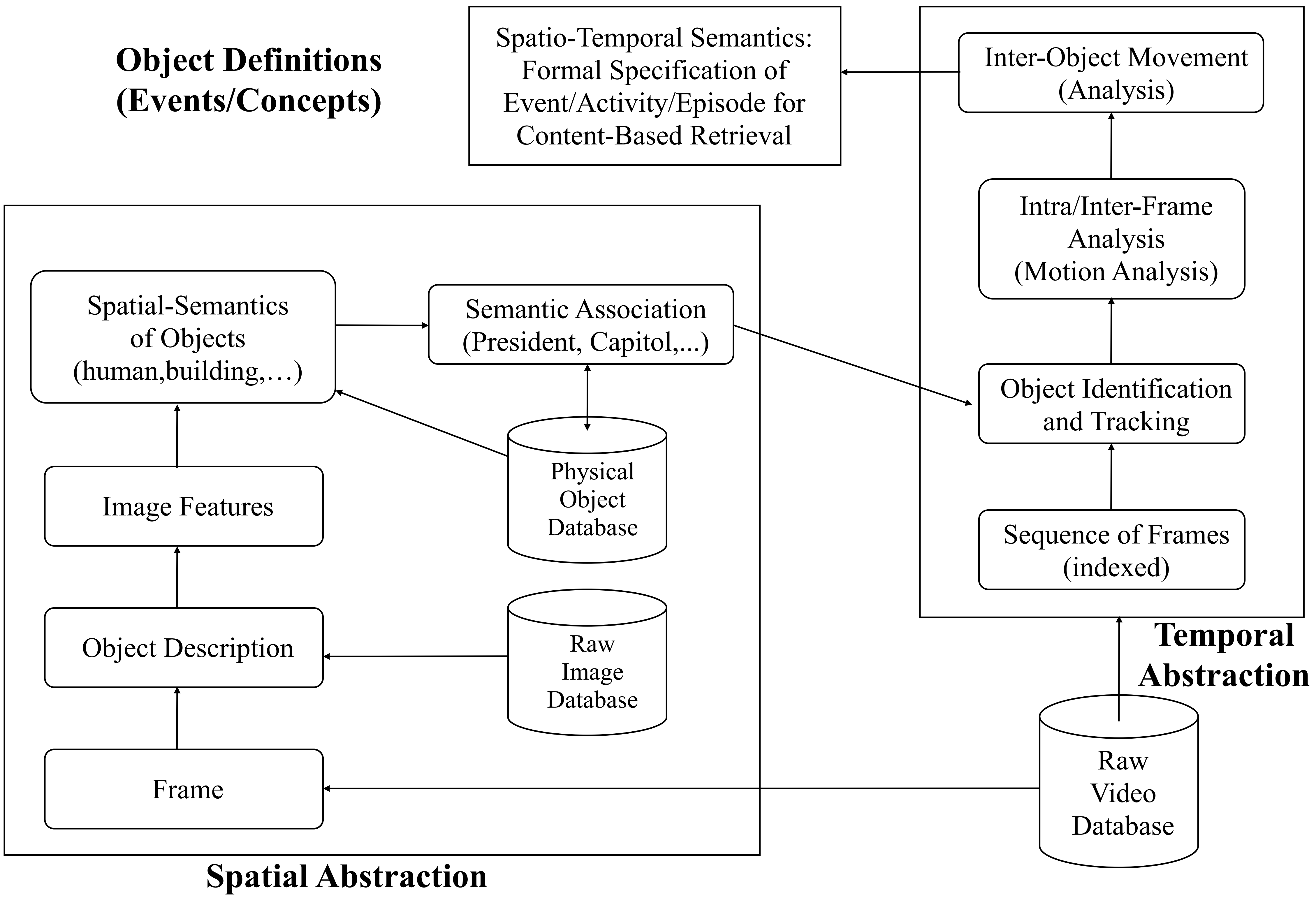
Software Architecture of MMDBMS


Users
To Presentation Device
Multimedia Structuring Module
Document Generator Tool Library
HyTime/SGML/XML Translator
Multimedia Meta-Data
HyTime/SGML/XML Parser --> Multimedia Query Language
OODBMS and MQL Distributed Query Processor
Temporal Synchronization Manager
Text Database
Video Database
Image Database
Audio Database

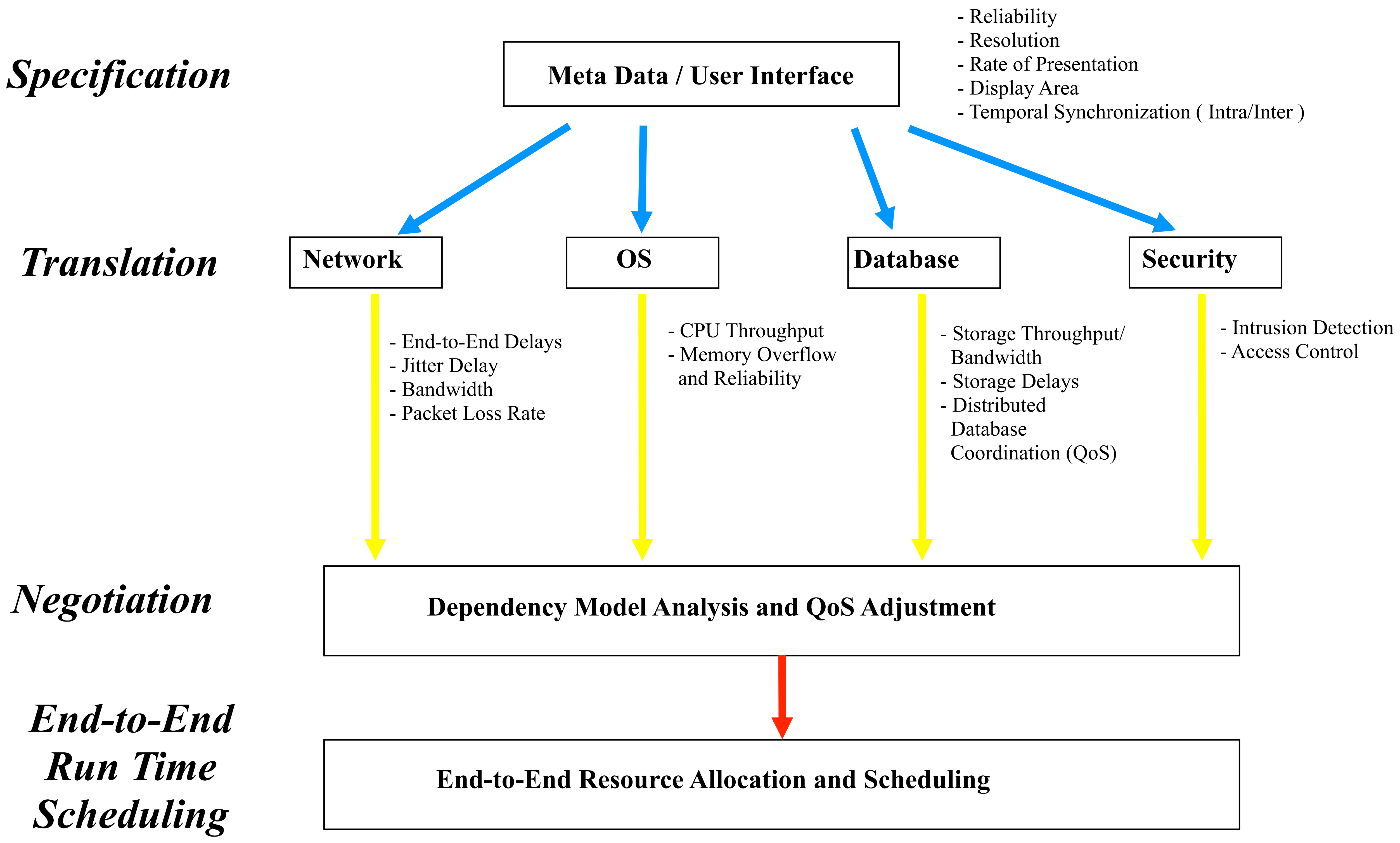
Distributed Multimedia Database Systems



An Architecture for Video Database System

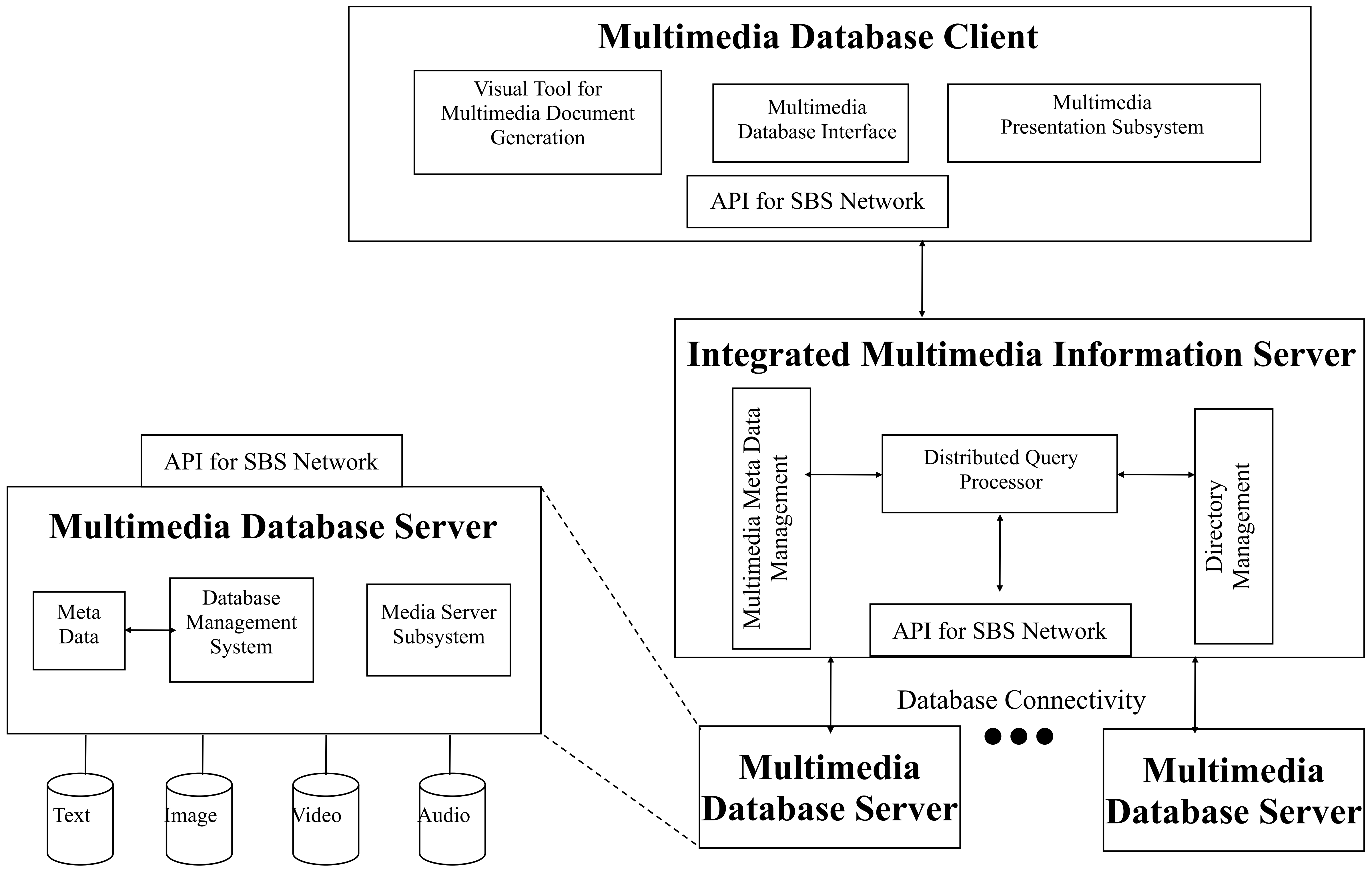

Object Definitions (Events/Concepts)
Spatio-Temporal Semantics: Formal Specification of Event/Activity/Episode for Content-Based Retrieval
Inter-Object Movement (Analysis)
Intra/Inter-Frame Analysis (Motion Analysis)
Object Identification and Tracking
Sequence of Frames (indexed)
Temporal Abstraction
Spatial-Semantics of Objects (human,building,…)
Semantic Association (President, Capitol,...)
Physical Object Database
Image Features
Object Description
Raw Image Database
Frame
Raw Video Database
Spatial Abstraction

End-to-End QoP / QoS Management



Specification
Meta Data / User Interface
- Reliability
- Resolution
- Rate of Presentation
- Display Area
- Temporal Synchronization (Intra/Inter)
Translation
Network
OS
Database
Security
- End-to-End Delays
- Jitter Delay
- Bandwidth
- Packet Loss Rate
- CPU Throughput
- Memory Overflow and Reliability
- Storage Throughput/ Bandwidth
- Storage Delays
- Distributed Database Coordination (QoS)
- Intrusion Detection
- Access Control
Negotiation
Dependency Model Analysis and QoS Adjustment
End-to-End Run Time Scheduling
End-to-End Resource Allocation and Scheduling

Architecture of a Distributed Multimedia Database Management


Multimedia Database Client
Visual Tool for Multimedia Document Generation
Multimedia Database Interface
Multimedia Presentation Subsystem
API for SBS Network
Integrated Multimedia Information Server
Multimedia Meta Data Management
Distributed Query Processor
Directory Management
API for SBS Network
Database Connectivity
Multimedia Database Server
Multimedia Database Server
API for SBS Network
Multimedia Database Server
Meta Data
Database Management System
Media Server Subsystem
Text
Image
Video
Audio

Overview of the System

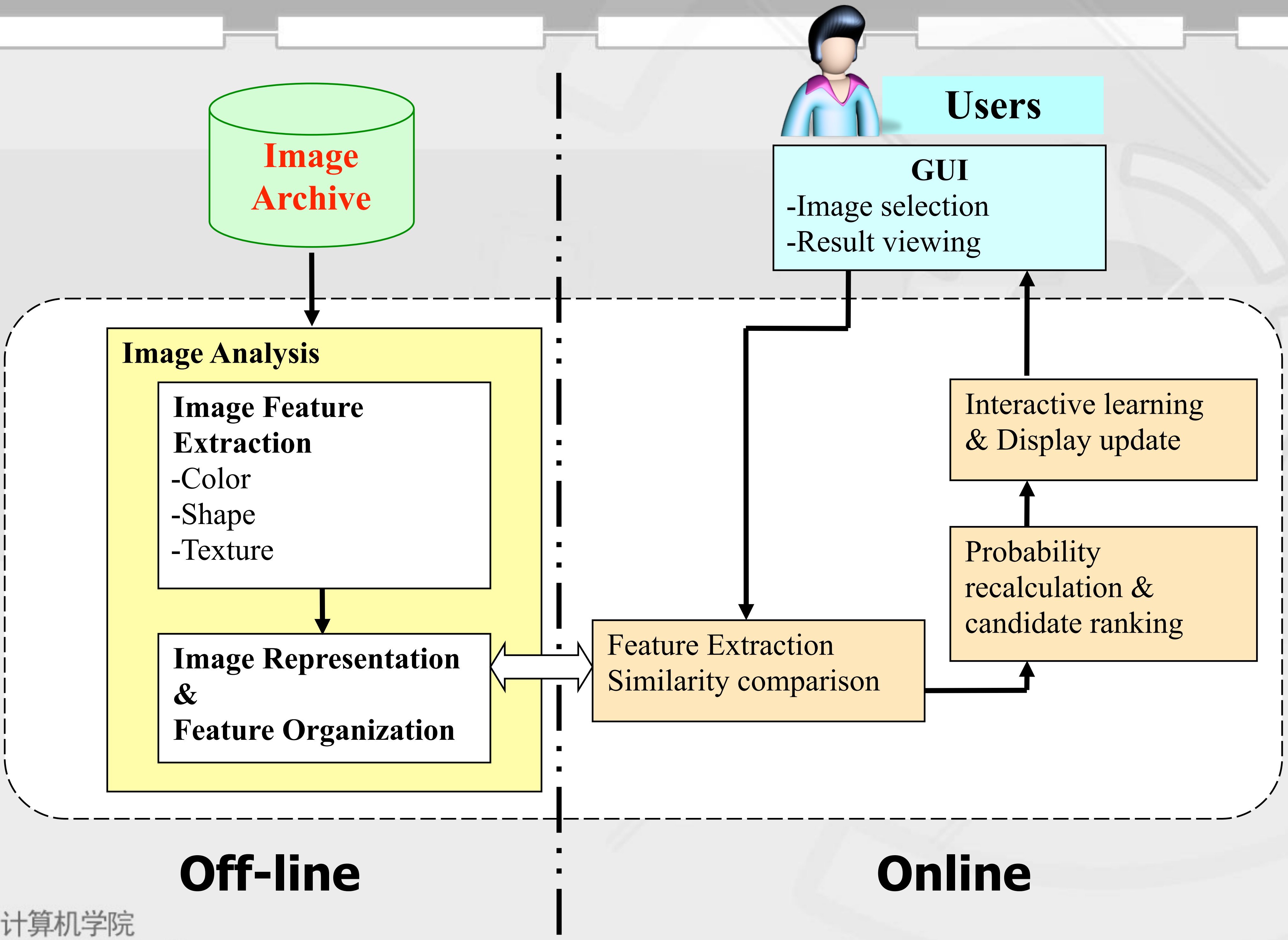


Image Archive
Users
GUI
-Image selection
-Result viewing
Image Analysis
Image Feature Extraction
-Color
-Shape
-Texture
Image Representation & Feature Organization
Feature Extraction
Similarity comparison
Probability recalculation & candidate ranking
Interactive learning & Display update
Off-line
Online