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
2008-09-17

Outline

1. MM content organization

2. MM database system architecture

3. MM system service model

4. Multimedia Data Storage

5. Multimedia application

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, …

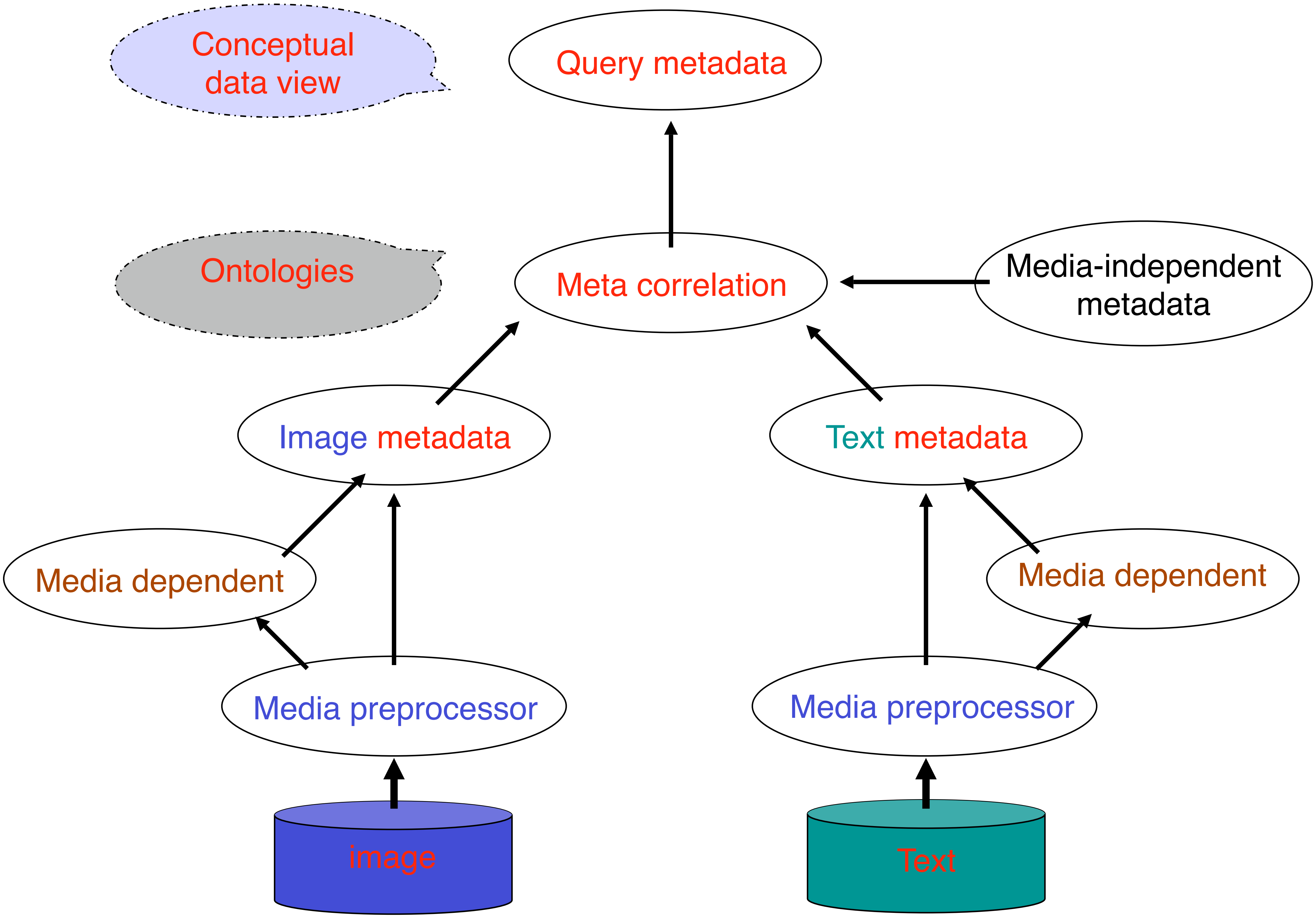
Metadata Model

- **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 with **open metadata standard** will be a vital:
  - Faster design and implementation
  - Interoperability 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		

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)

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

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**:
  - There is a set called *Resources*.
  - There is a set called *Literals*.
  - There is 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

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

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>   **attribute**
  - <account>
    <account-number>A-101</account-number> …   **subelement**
    </account>
- 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

浙江大学计算机学院
数字媒体与网络技术

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

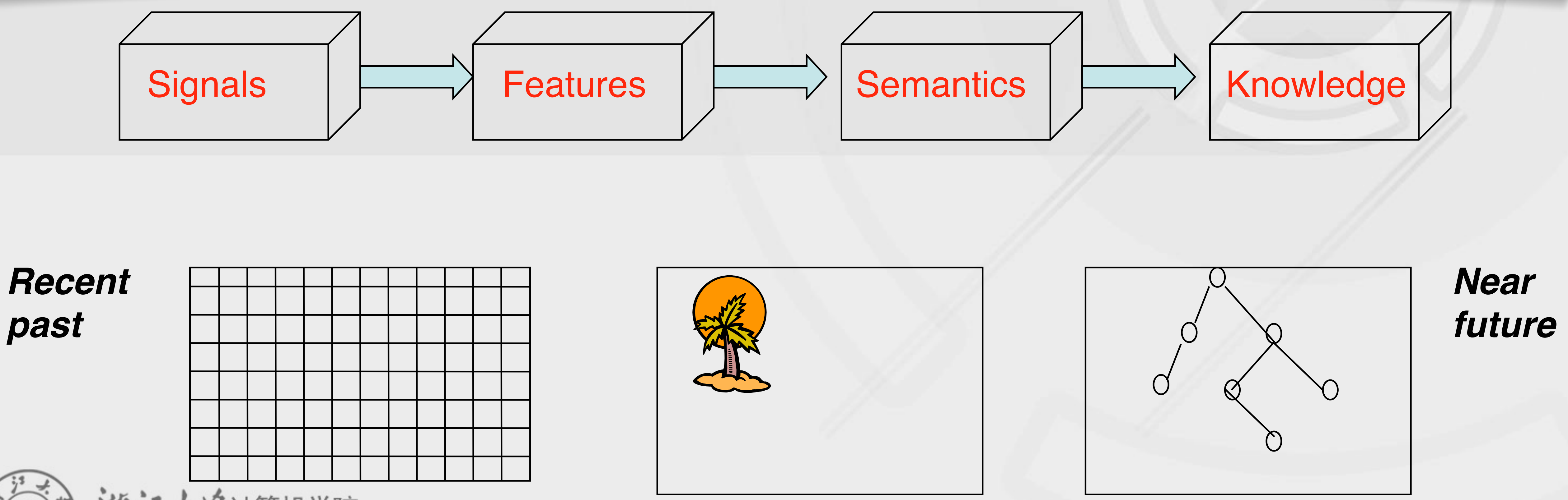
```
<bank-2>
    <account account-number=“A-401” owners=“C100 C102”>
            <branch-name> Downtown </branch-name>
            <balance>       500 </balance>
    </account>
    <customer customer-id=“C100” accounts=“A-401”>
            <customer-name>Joe        </customer-name>
            <customer-street> 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

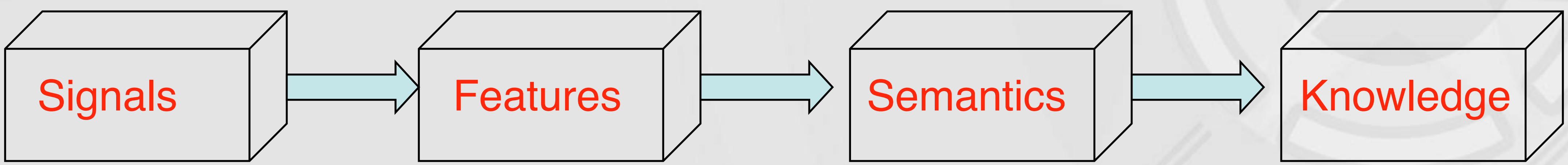
Domain-independent Metadata Standards

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

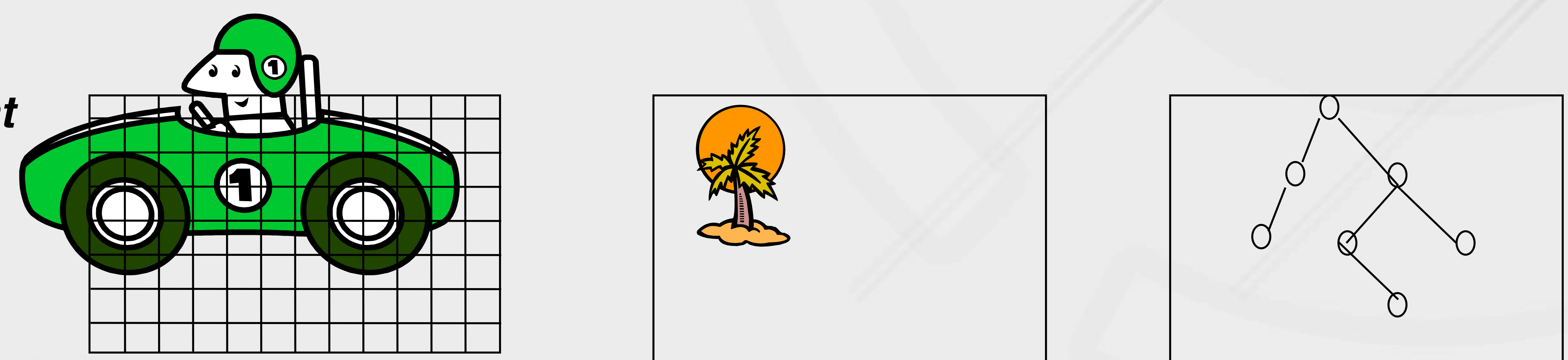


Domain-independent Metadata Standards

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



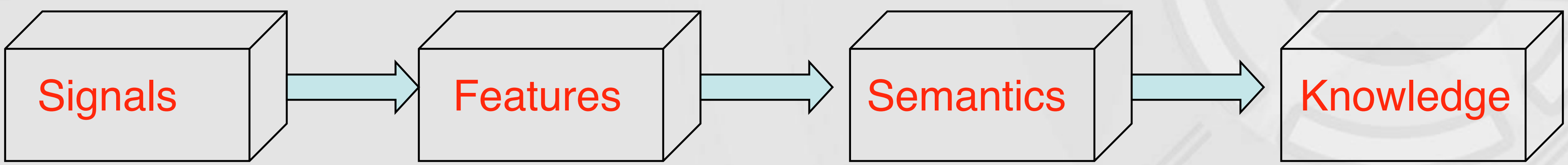
Recent past



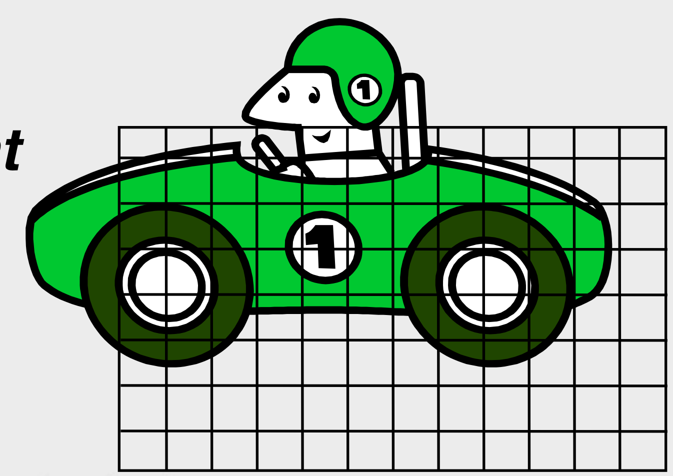
Near future

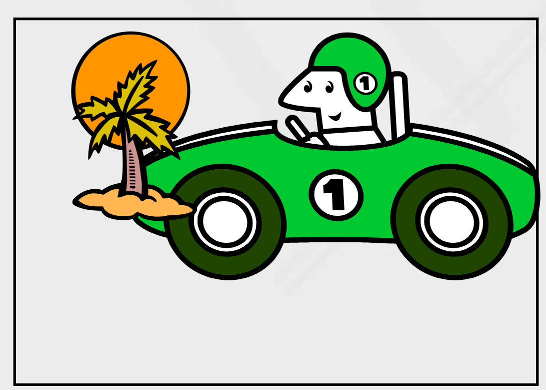
Domain-independent Metadata Standards

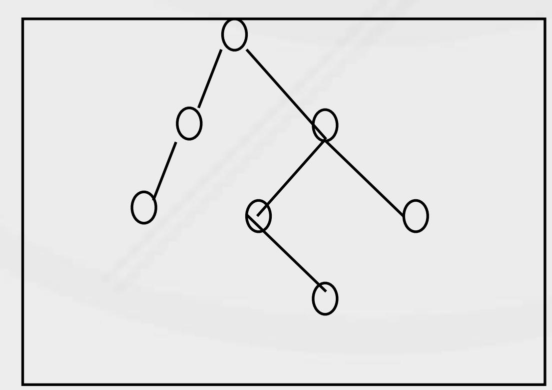
- MPEG series
  - Moving Picture Experts Group (MPEG) since 1998
  - responsible for developing standards of the coded representation of moving pictures and associated audio



Recent past







Near future

Domain-independent Metadata Standards

Applications			
MPEG-1,-2,-4	MPEG-4,-7	MPEG-7	MPEG-21
Video storage **Broadband, streaming video delivery**	**CBR** **Multimedia filtering** **Content adaptation**	**Semantic-based retrieval and filtering** **Intelligent media services (iTV)**	**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 ,

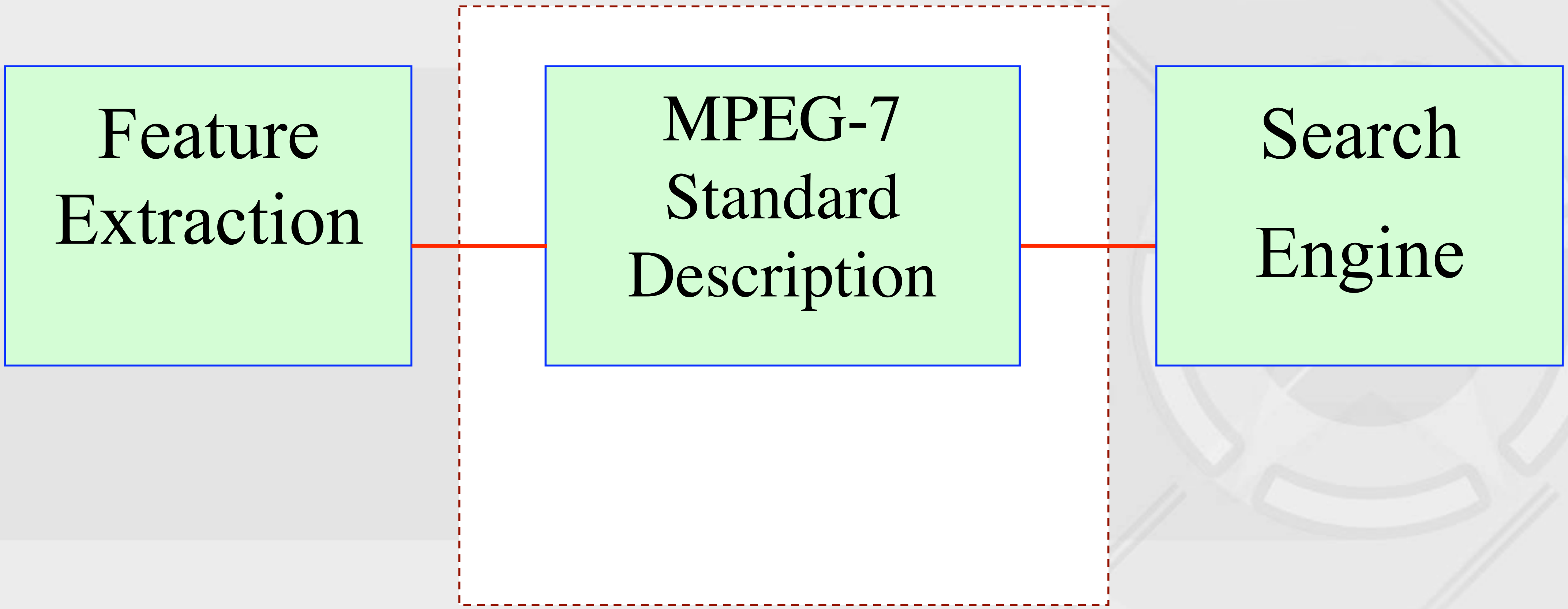
MPEG-21 ,

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

</description>
<xml>
<resource>

Domain-independent Metadata Standards

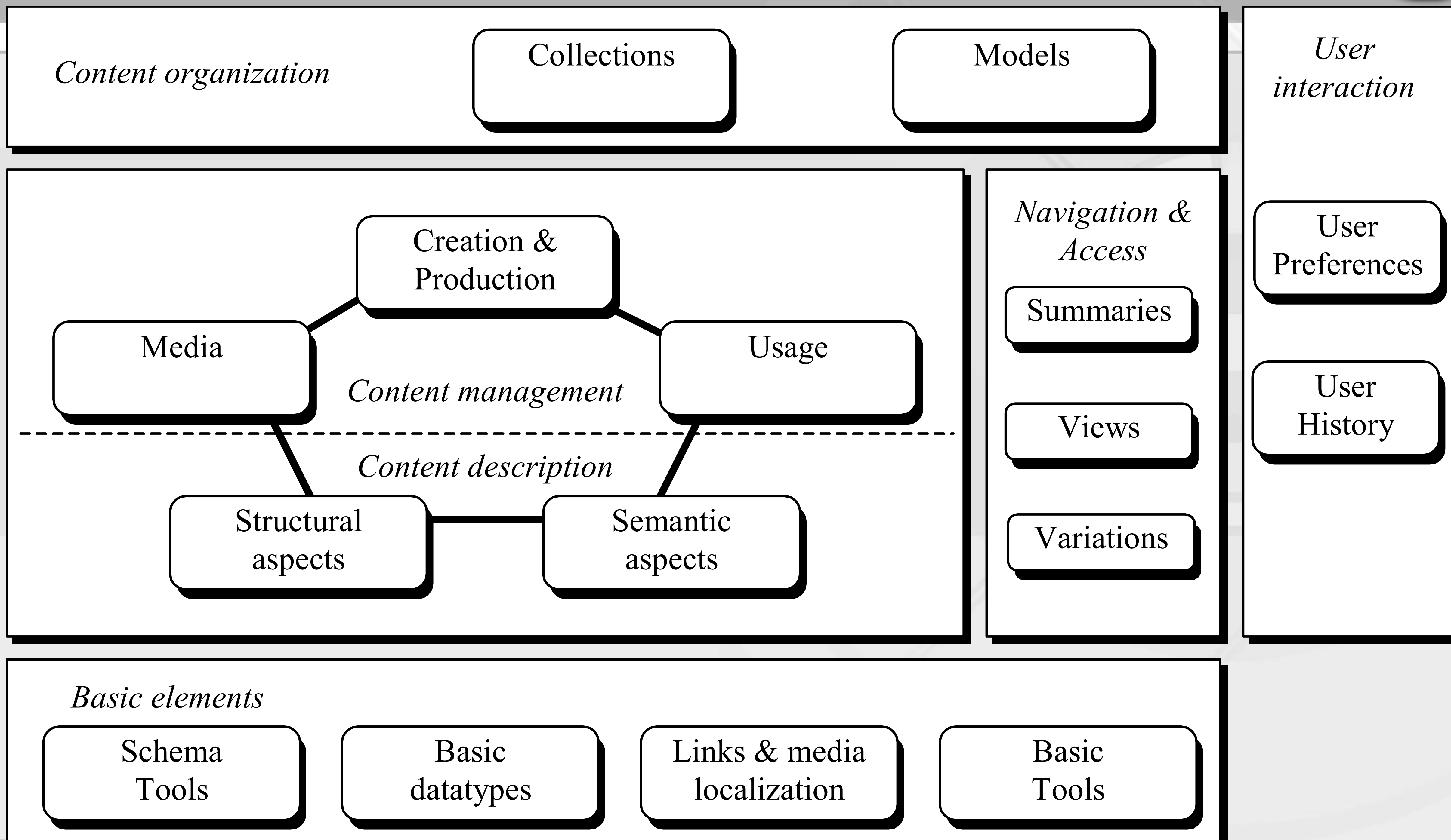


Scope of MPEG-7

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
Navigation & Access
Summaries
Views
Variations
User Preferences
User History
Creation & Production
Media
Usage
Content management
Content description
Structural aspects
Semantic aspects
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

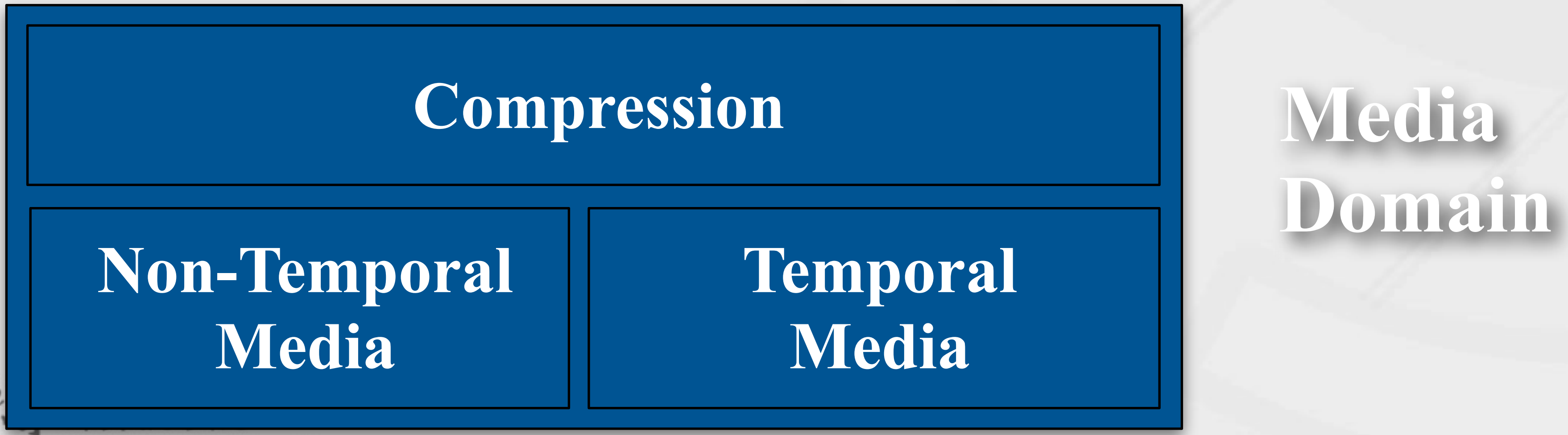
4. Multimedia Data Storage

5. Multimedia application

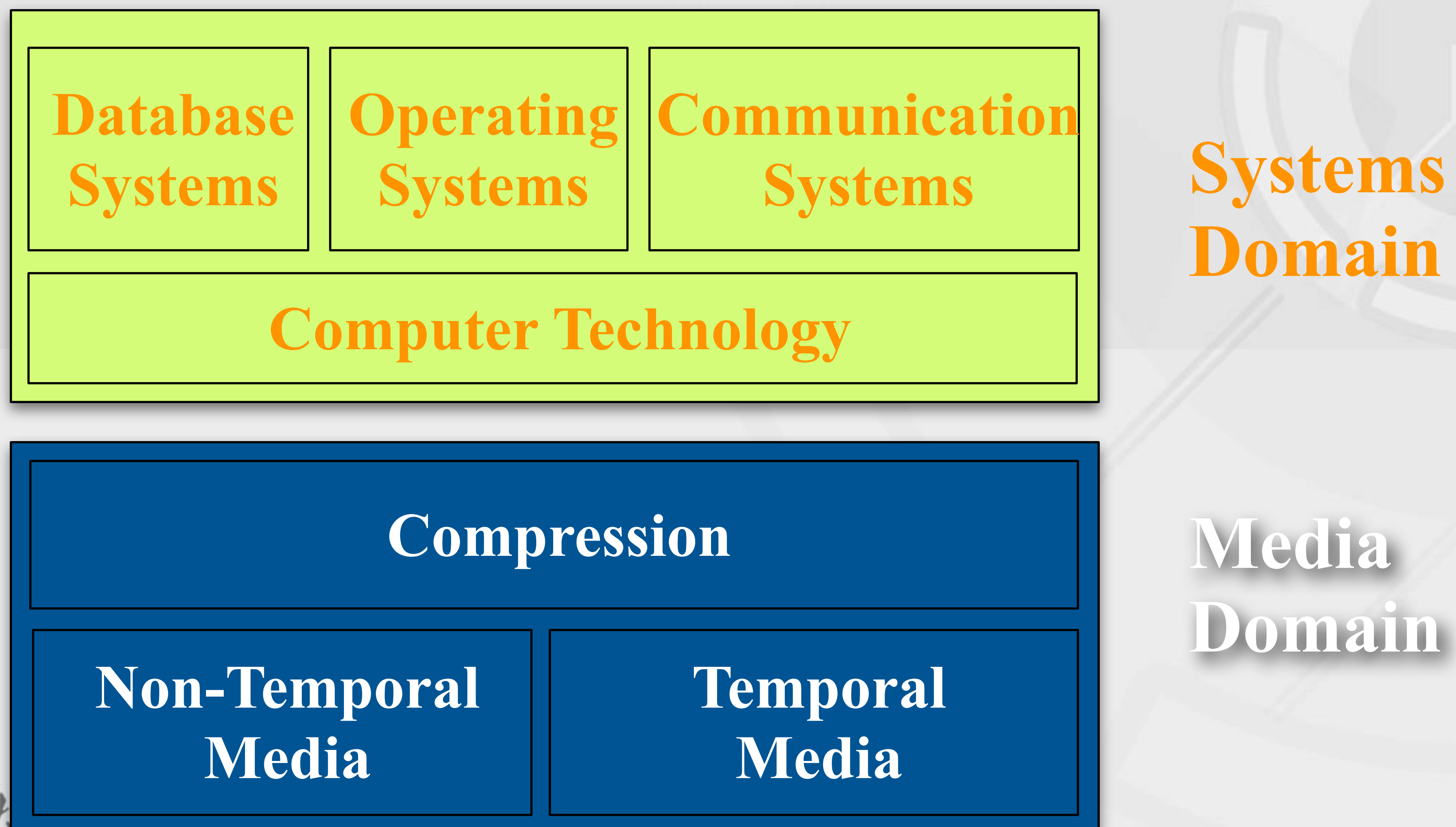
3.2 Multimedia Database System Architecture

Multimedia Architecture

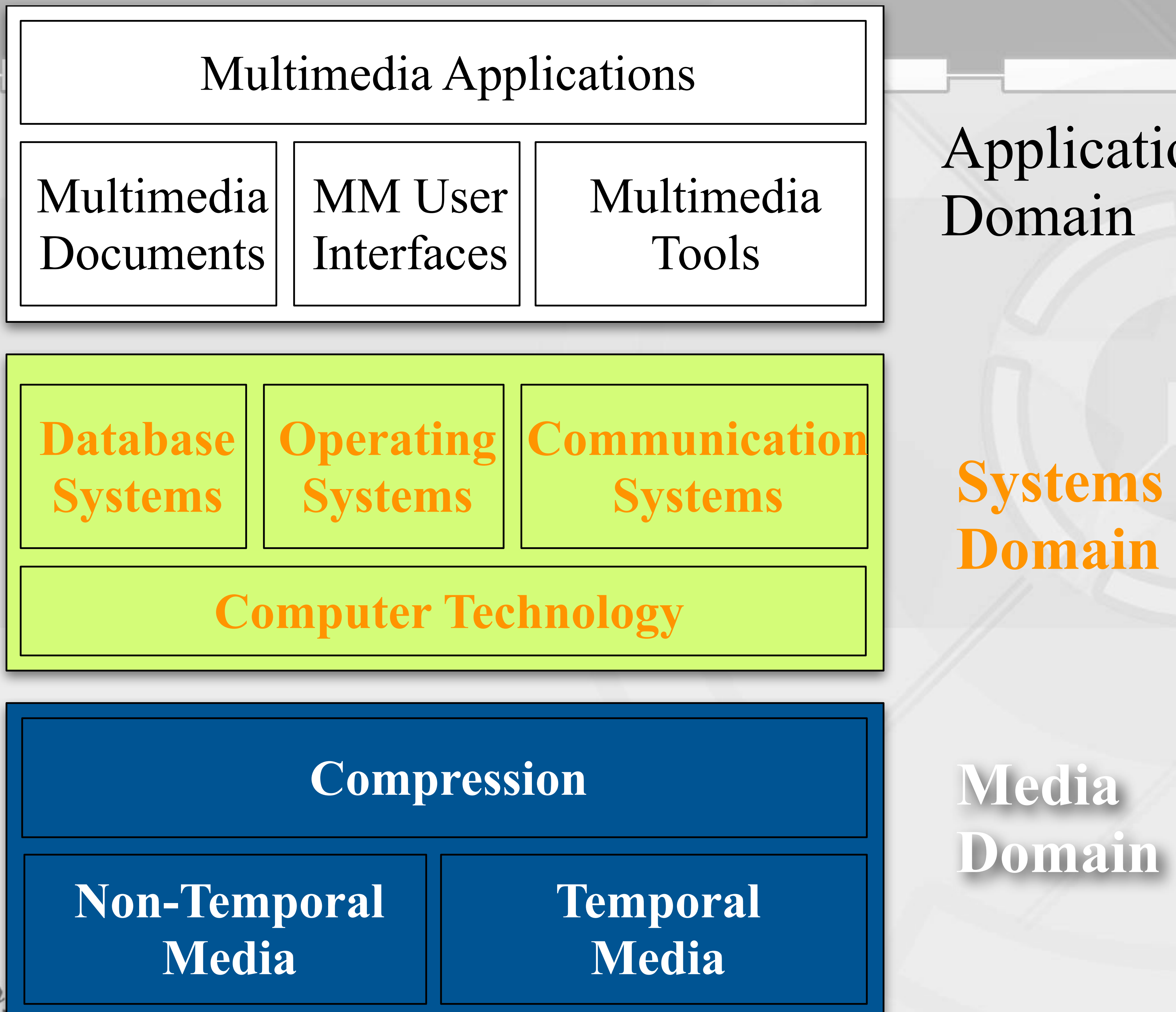
Multimedia Architecture



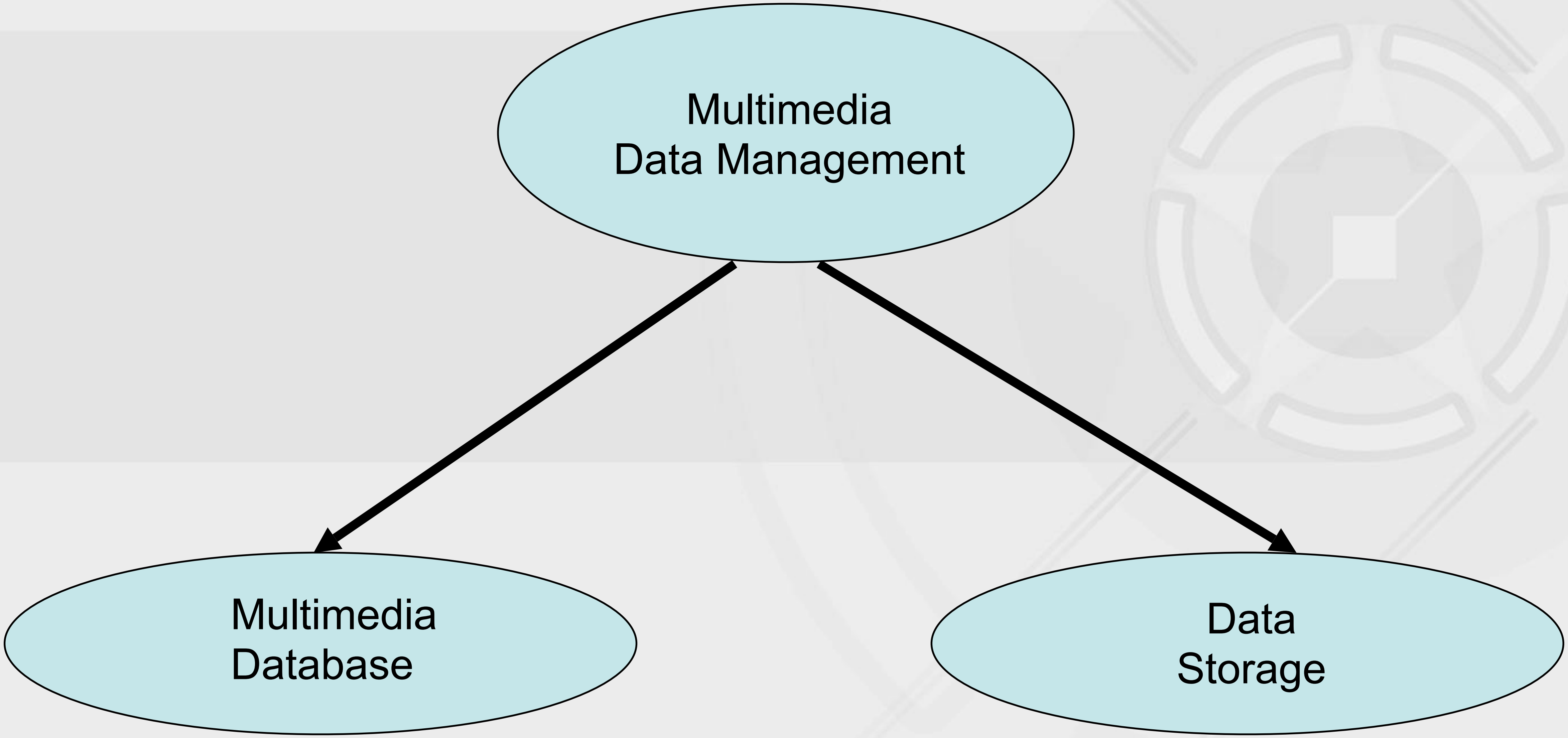
Multimedia Architecture



Multimedia Architecture



Multimedia Database System



Multimedia Database System

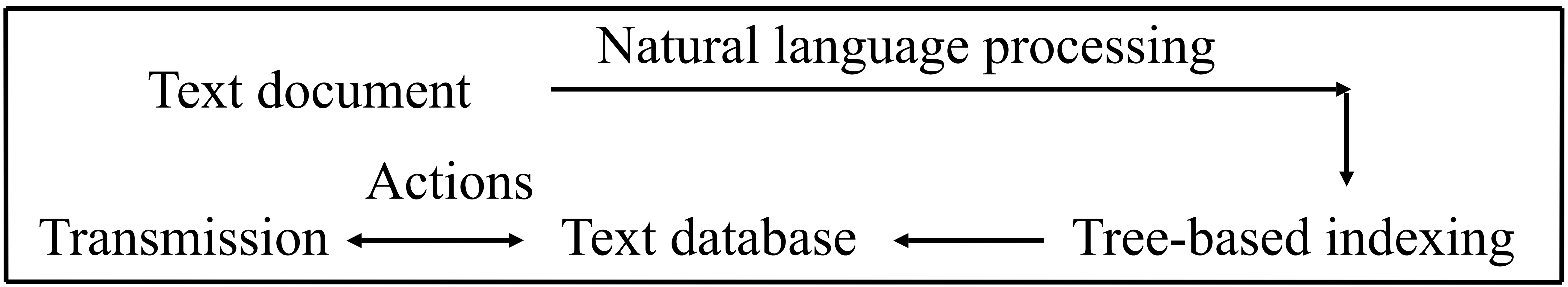
- **Multimedia database** v.s. **text database**
  - **Temporal data**: Requires temporal modeling
  - **Huge amount of data**: Compression helps get around this.
  - Data is **not easily indicative** of the information
  - Requires a lot of **pre-processing** in order to store data efficiently:
    - 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*

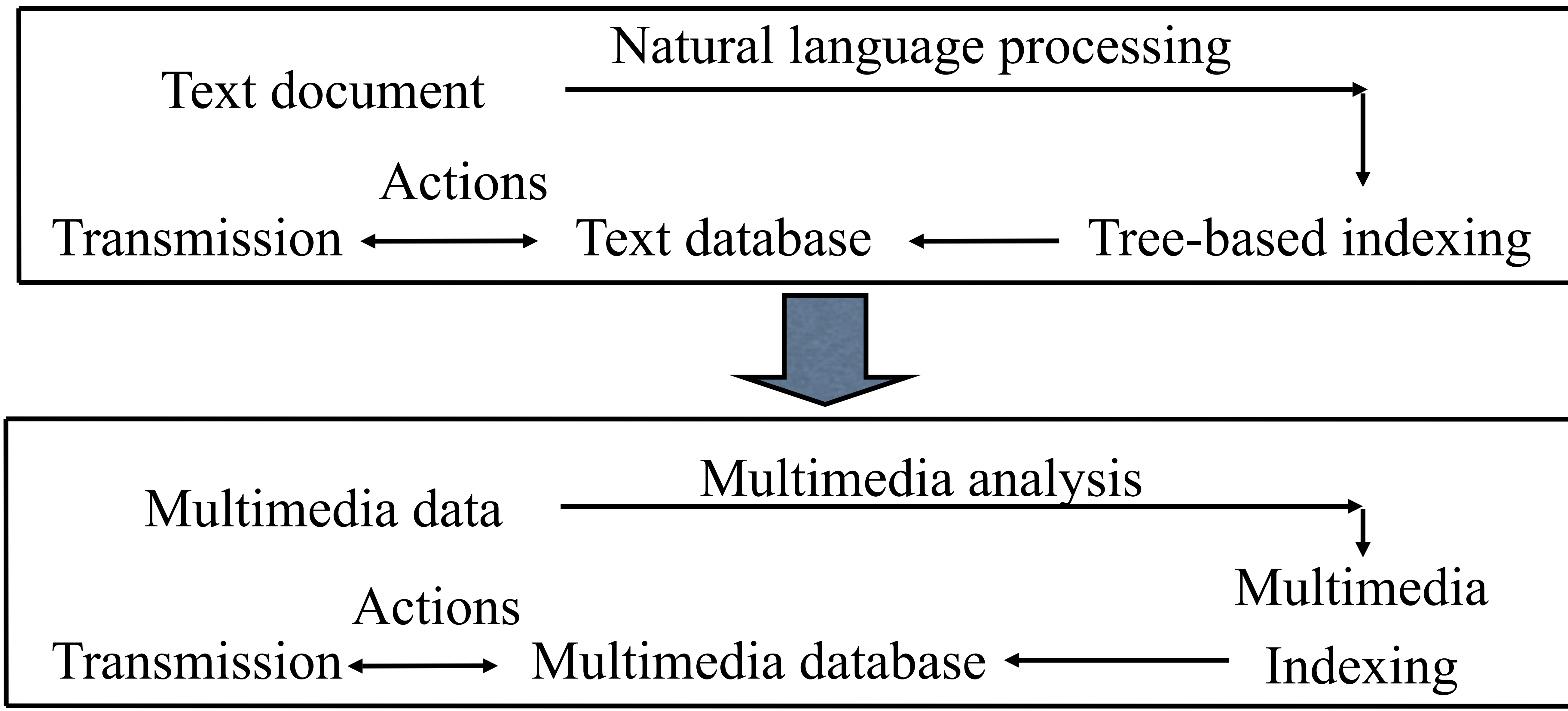
How to Build Multimedia Database Systems?

How to build text database? *Yahoo, Google*

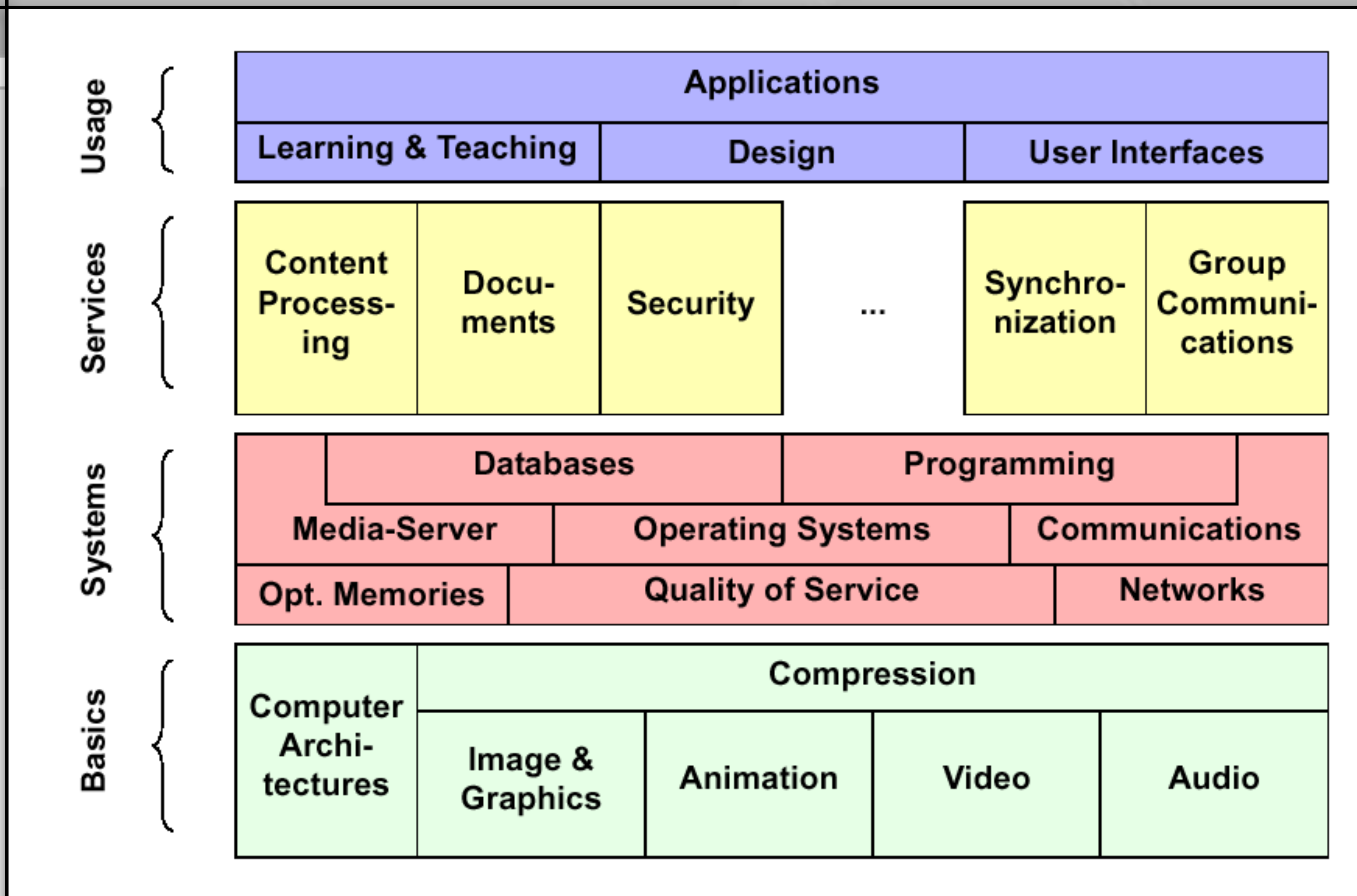


How to Build Multimedia Database Systems?

How to build text database? *Yahoo, Google*

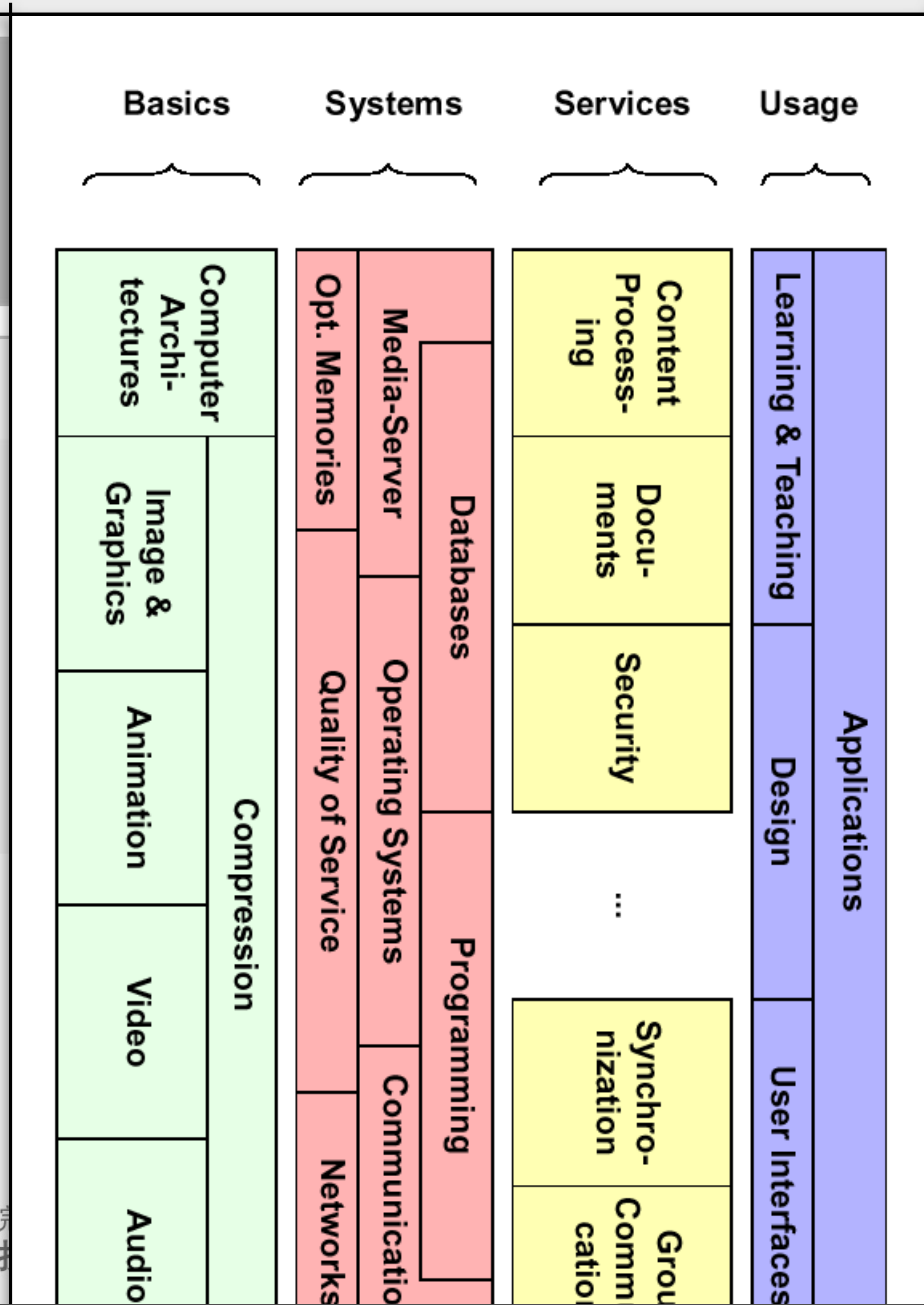


Scope

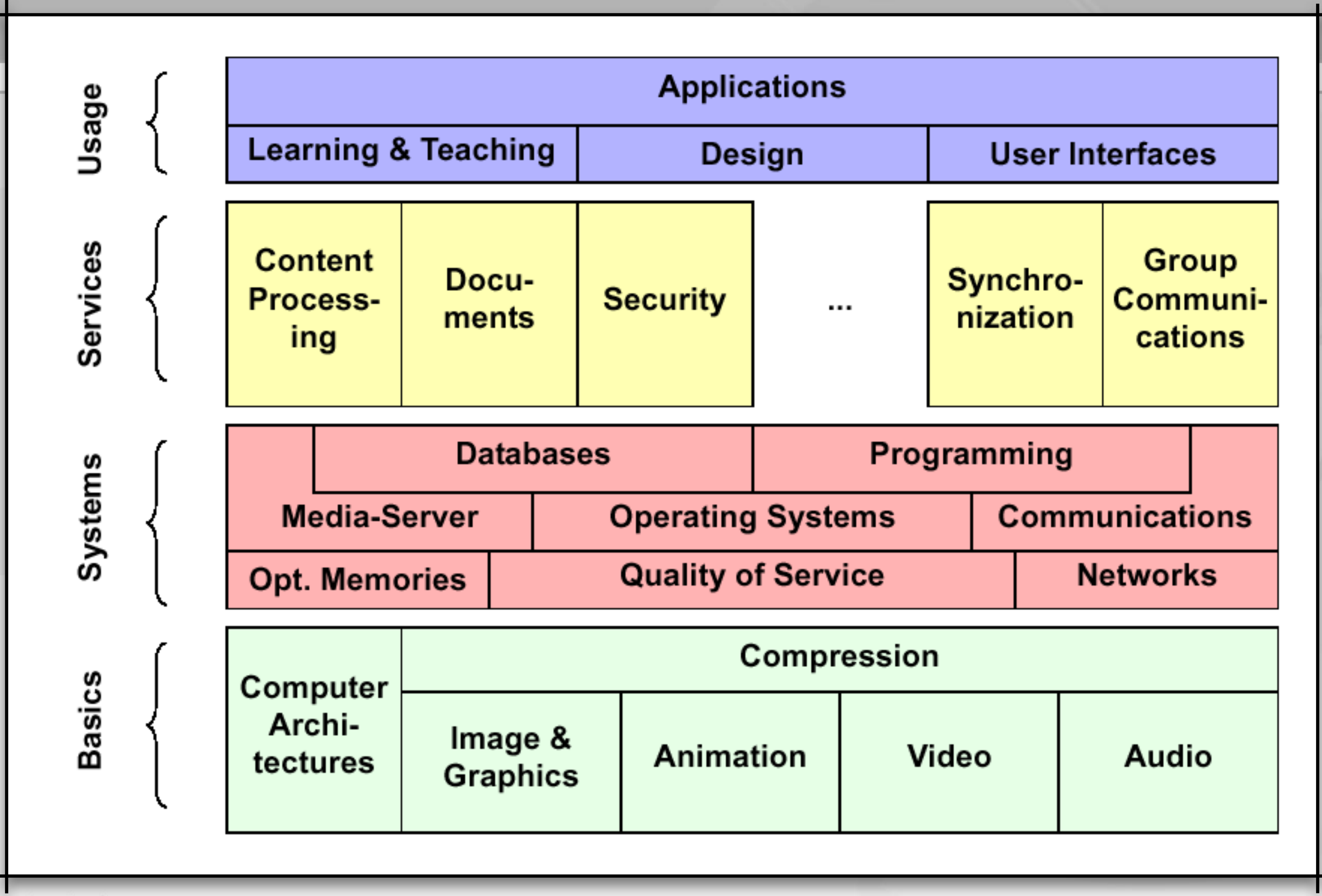


Usage
Applications
Learning & Teaching
Design
User Interfaces
Services
Content Process-ing
Docu-ments
Security
...
Synchro-nization
Group Communi-cations
Systems
Databases
Programming
Media-Server
Operating Systems
Communications
Opt. Memories
Quality of Service
Networks
Basics
Computer Archi-tectures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

Scope


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

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
Compression
Computer Archi-tectures
Image & Graphics
Animation
Video
Audio

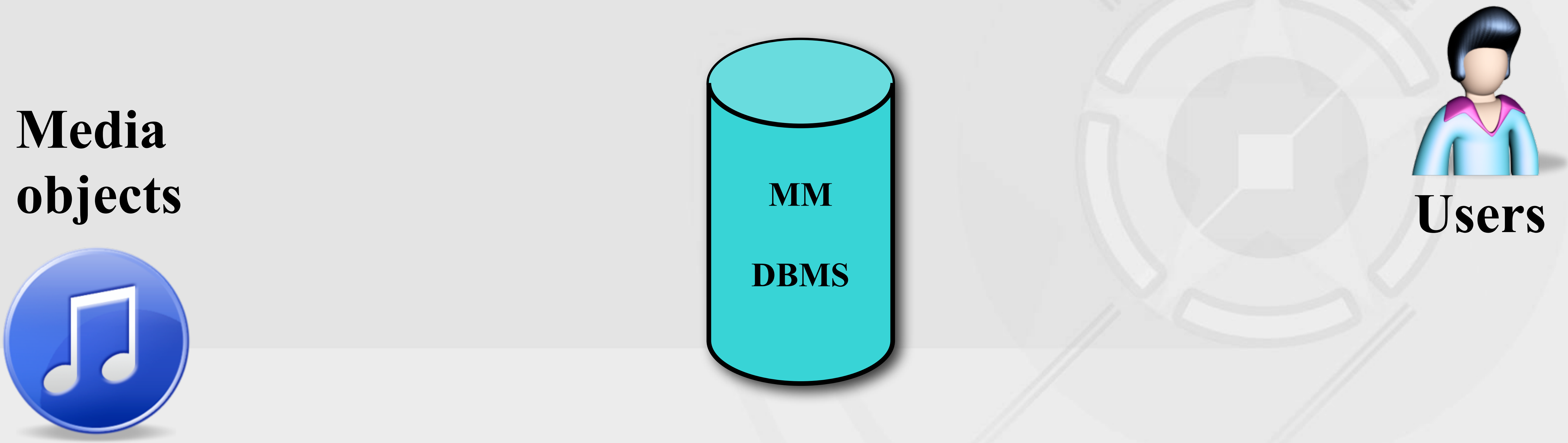
A Reference Architecture for MMDB System

- **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**

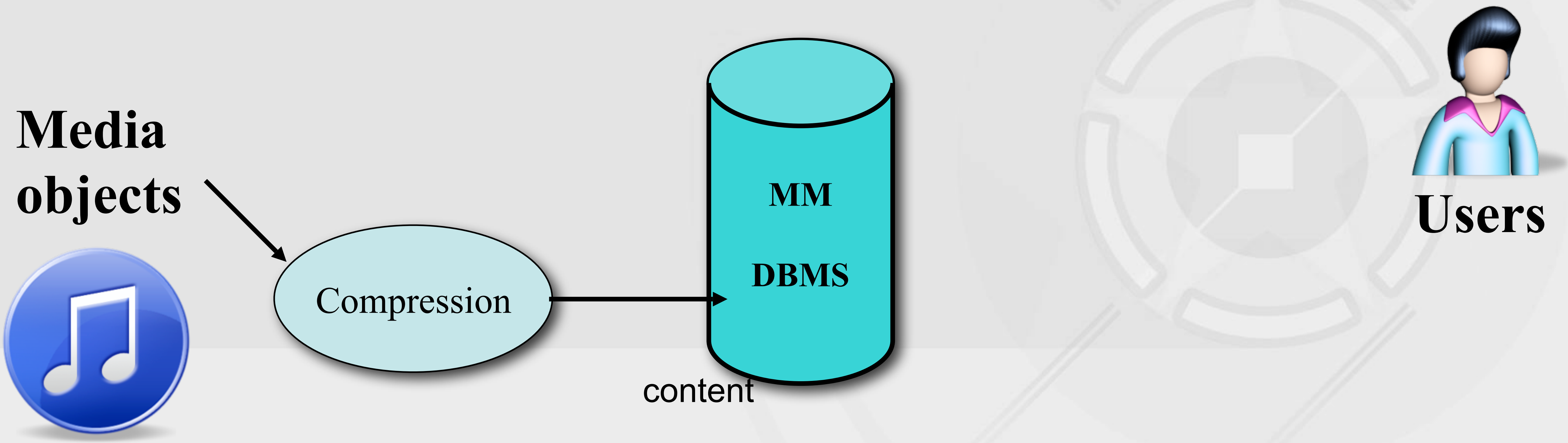
A Reference Architecture for MMDB System

- **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**

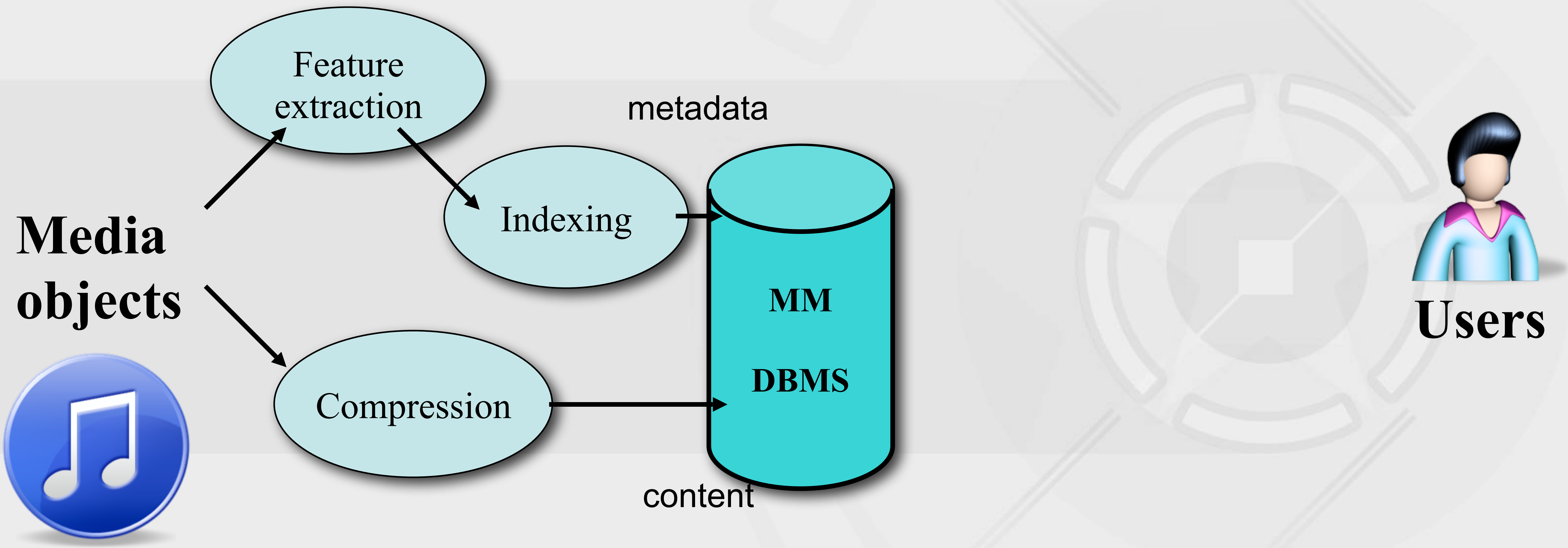
A Generic Architecture of MMDBMS



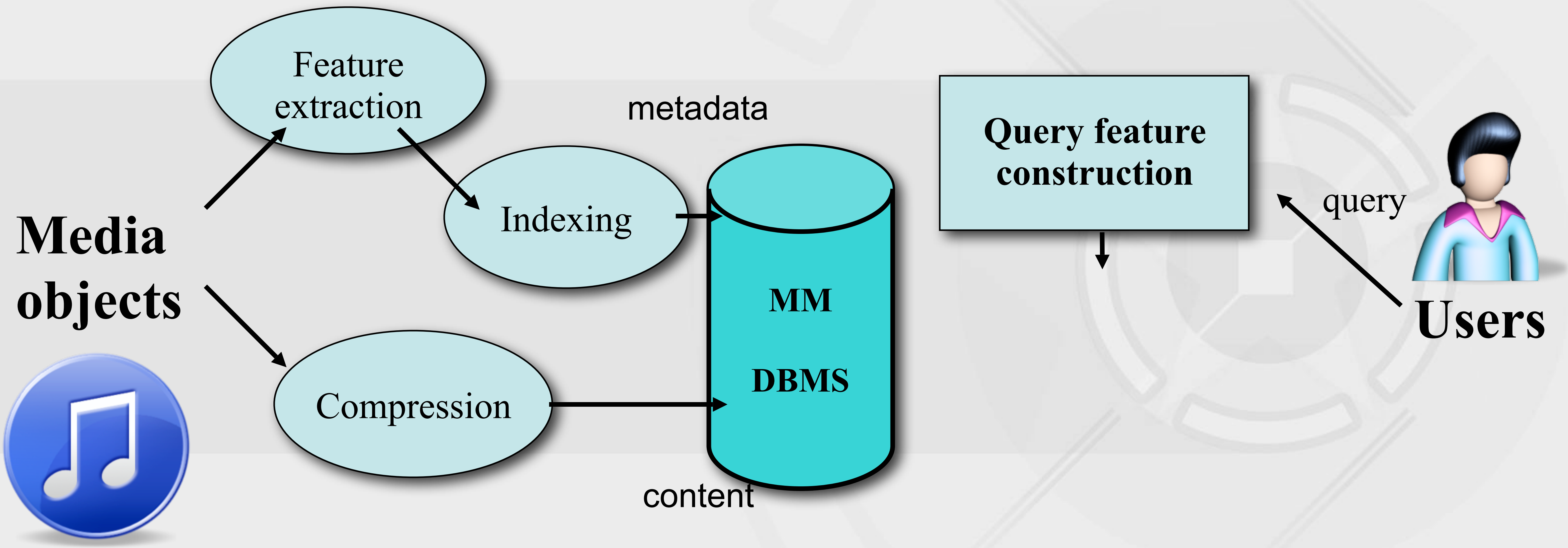
A Generic Architecture of MMDBMS



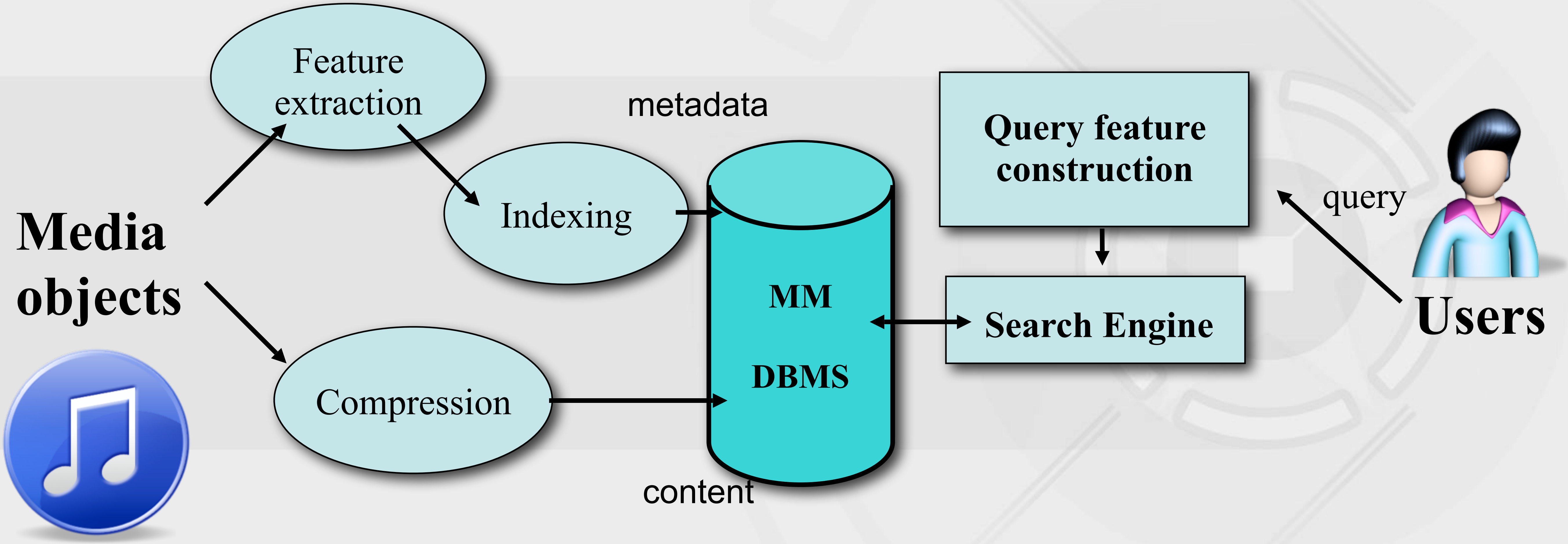
A Generic Architecture of MMDBMS



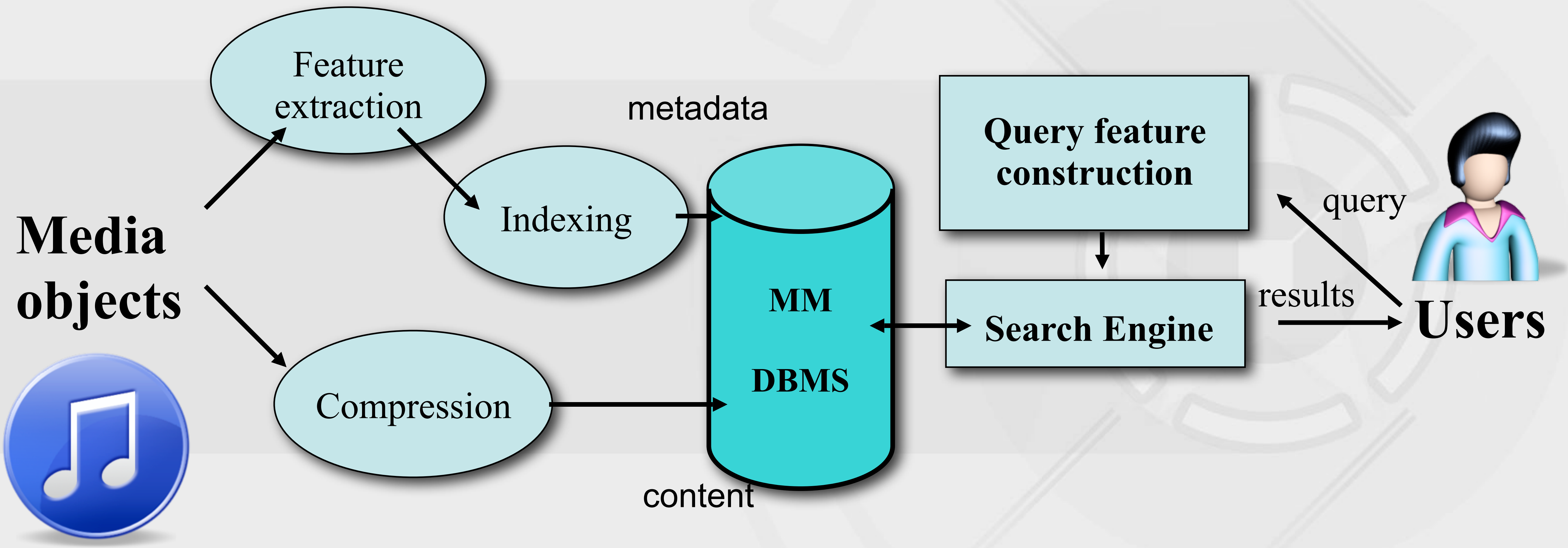
A Generic Architecture of MMDBMS



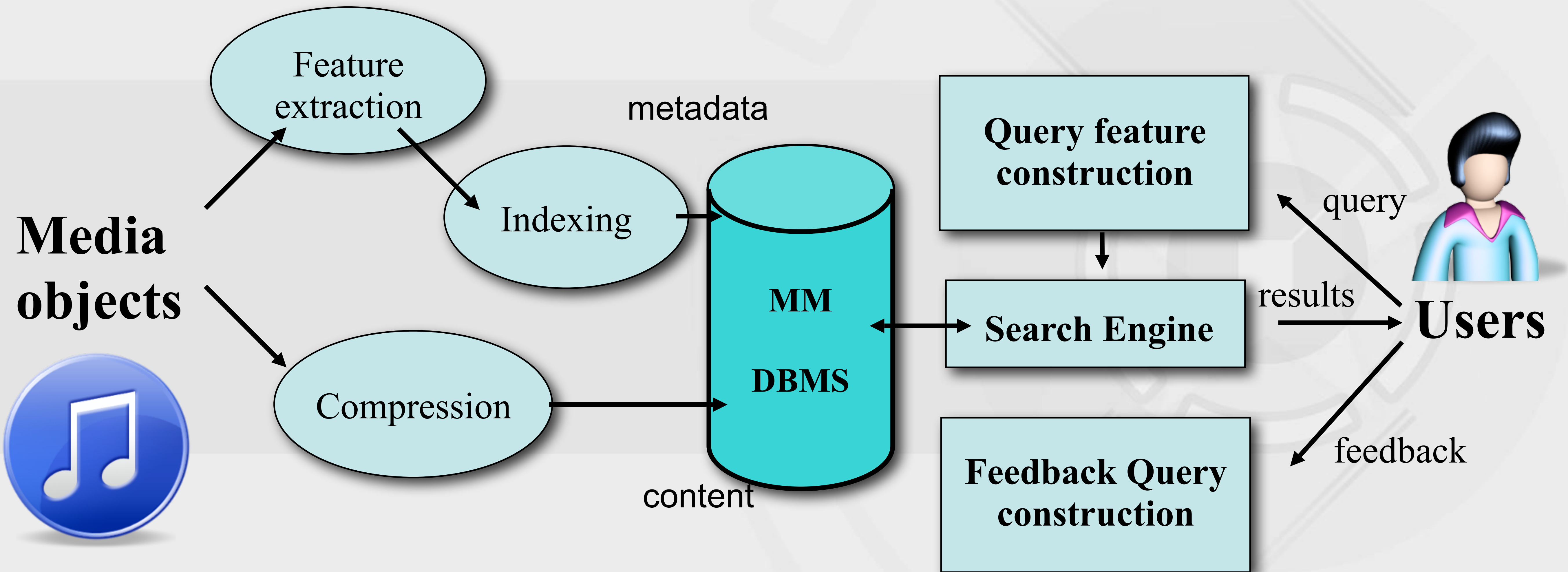
A Generic Architecture of MMDBMS



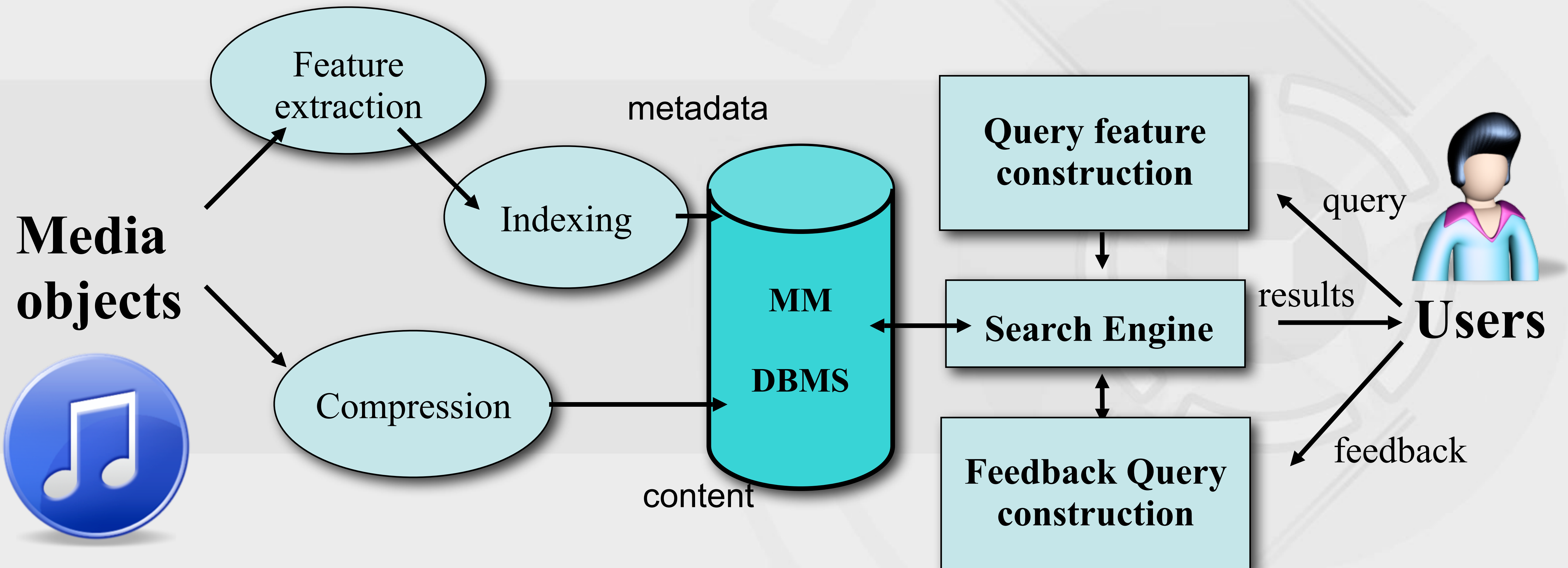
A Generic Architecture of MMDBMS



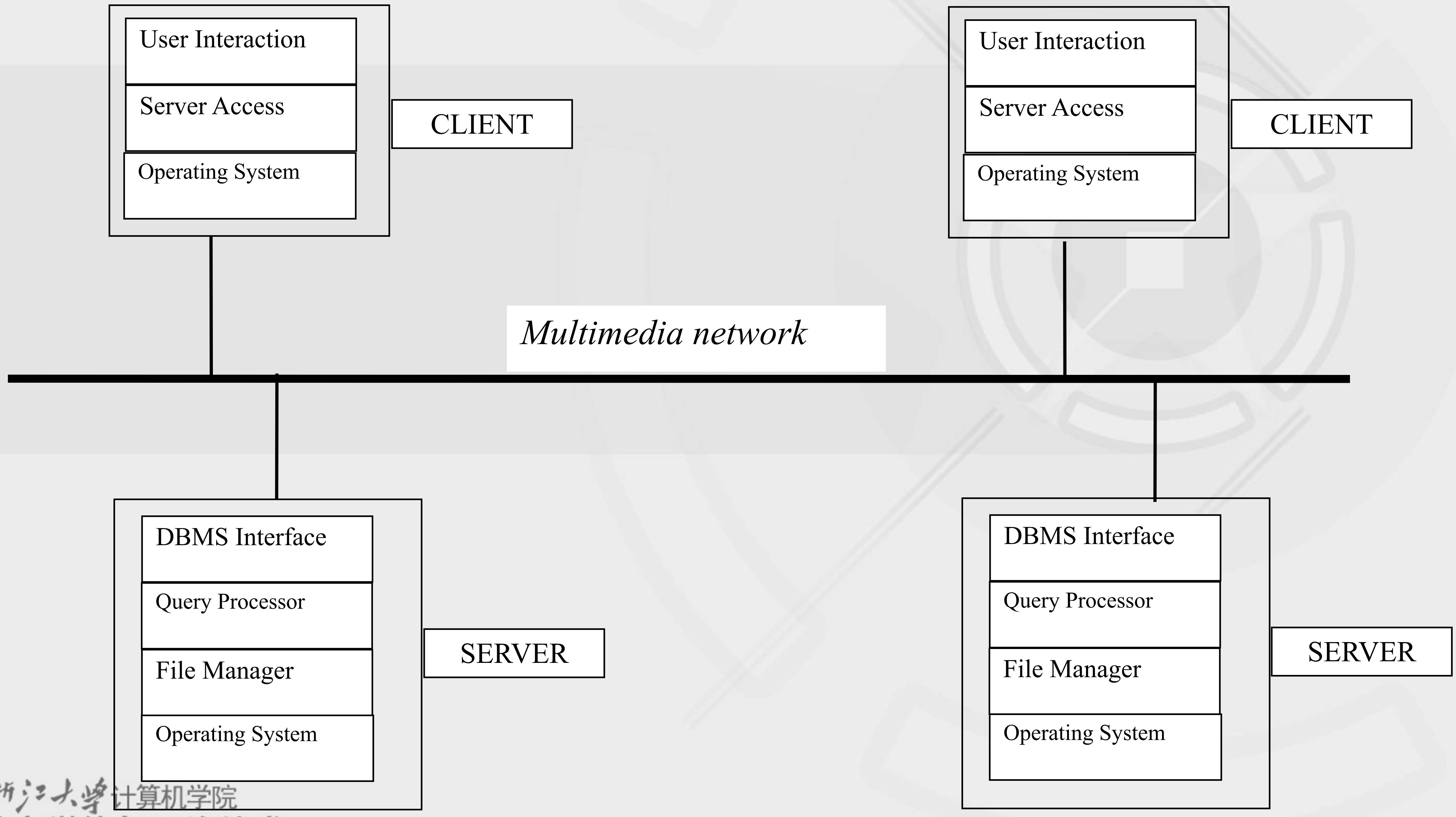
A Generic Architecture of MMDBMS



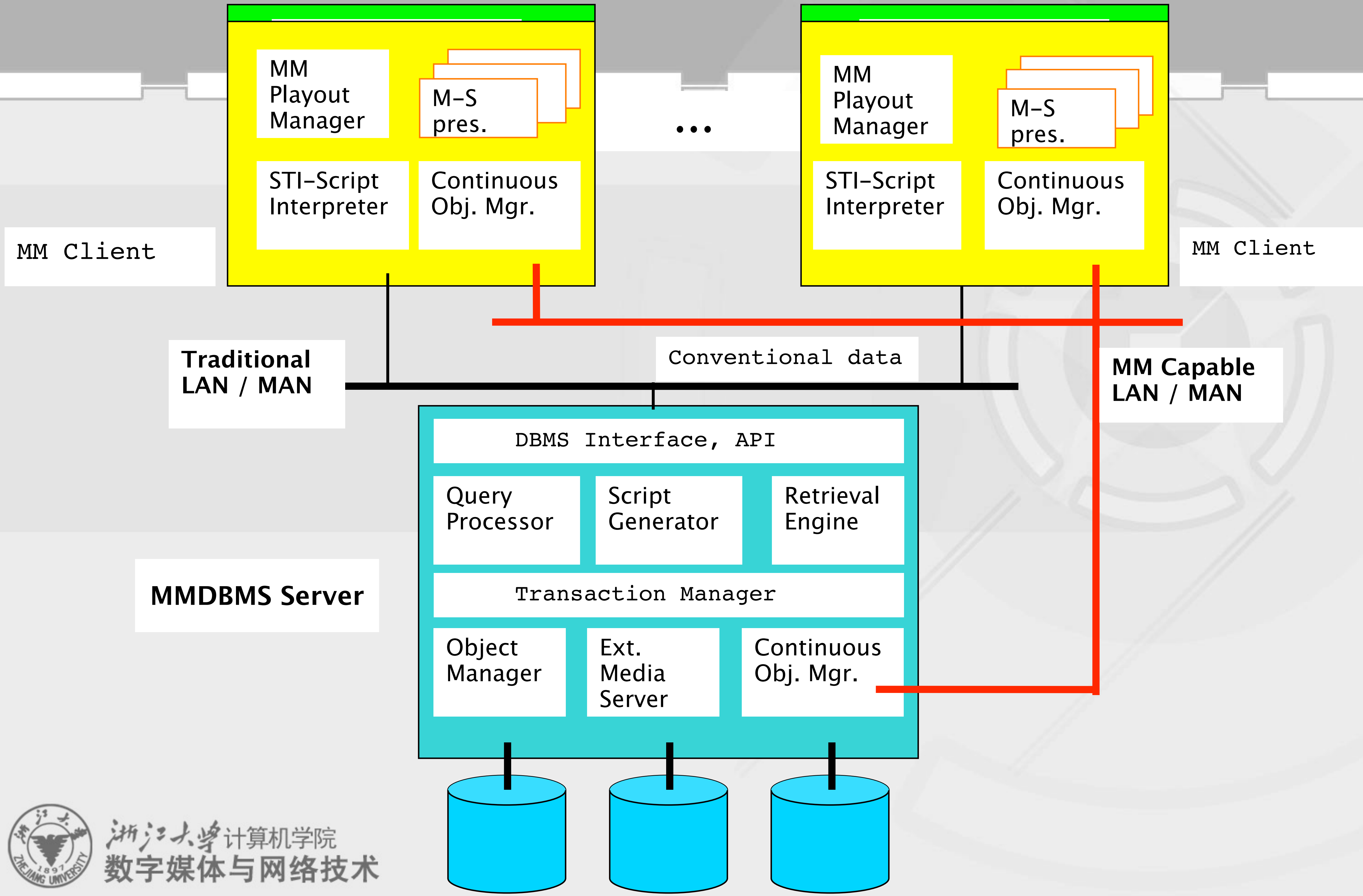
A Generic Architecture of MMDBMS



MMDB Reference Architecture: "Simplified View"



Detailed View of MMDB Architecture



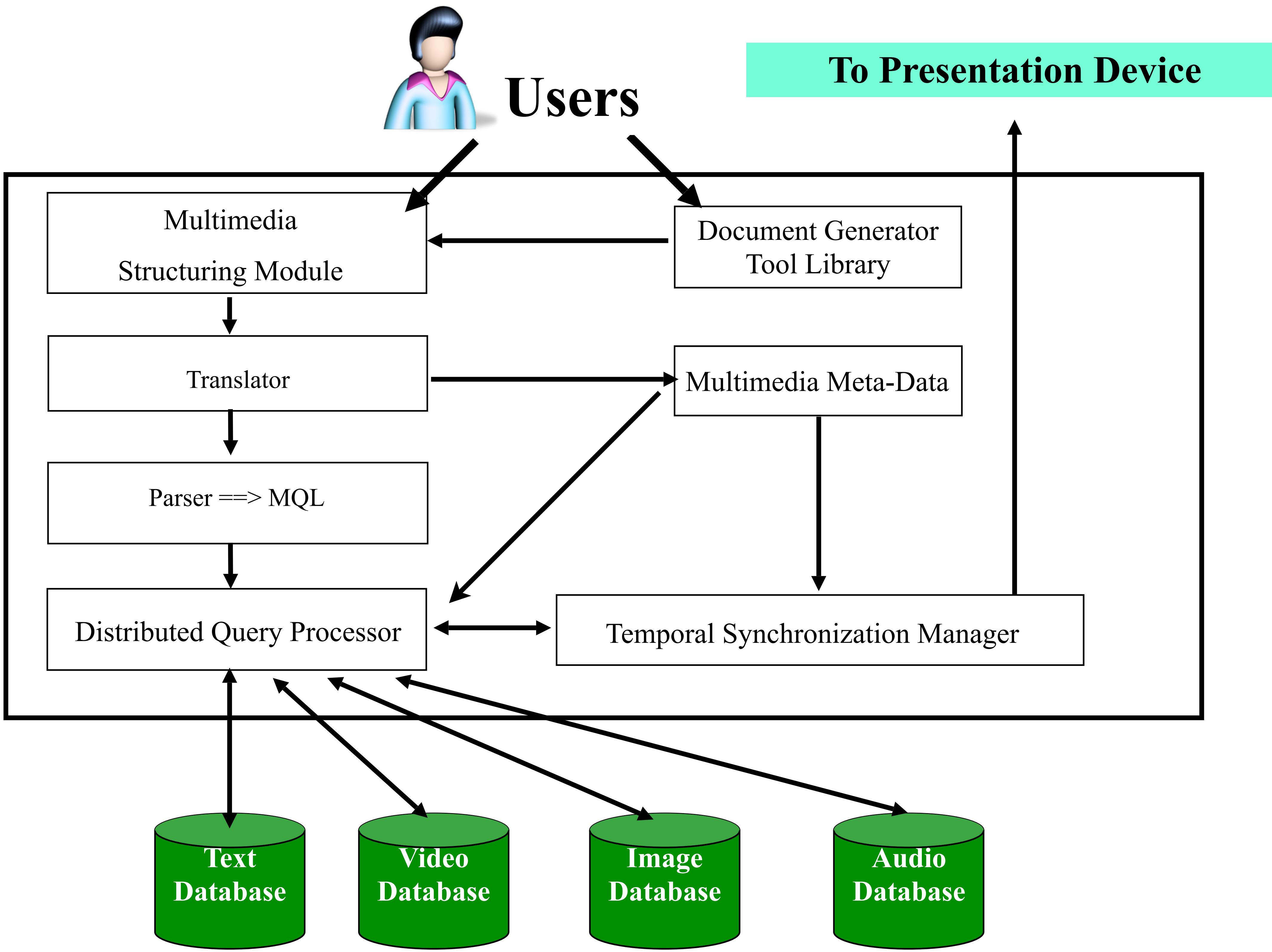
浙江大学计算机学院
数字媒体与网络技术

MMDBMS Development

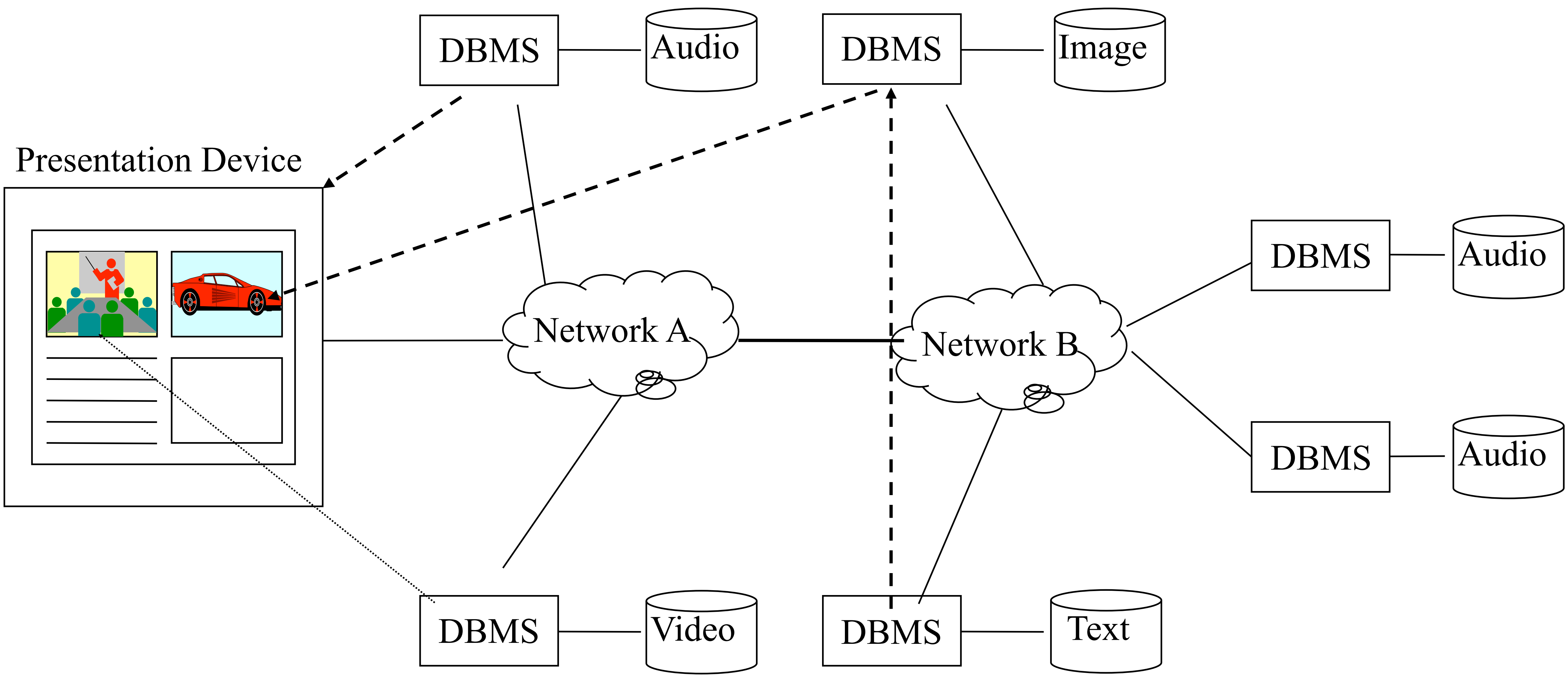
Major steps in developing MMDBMS

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.

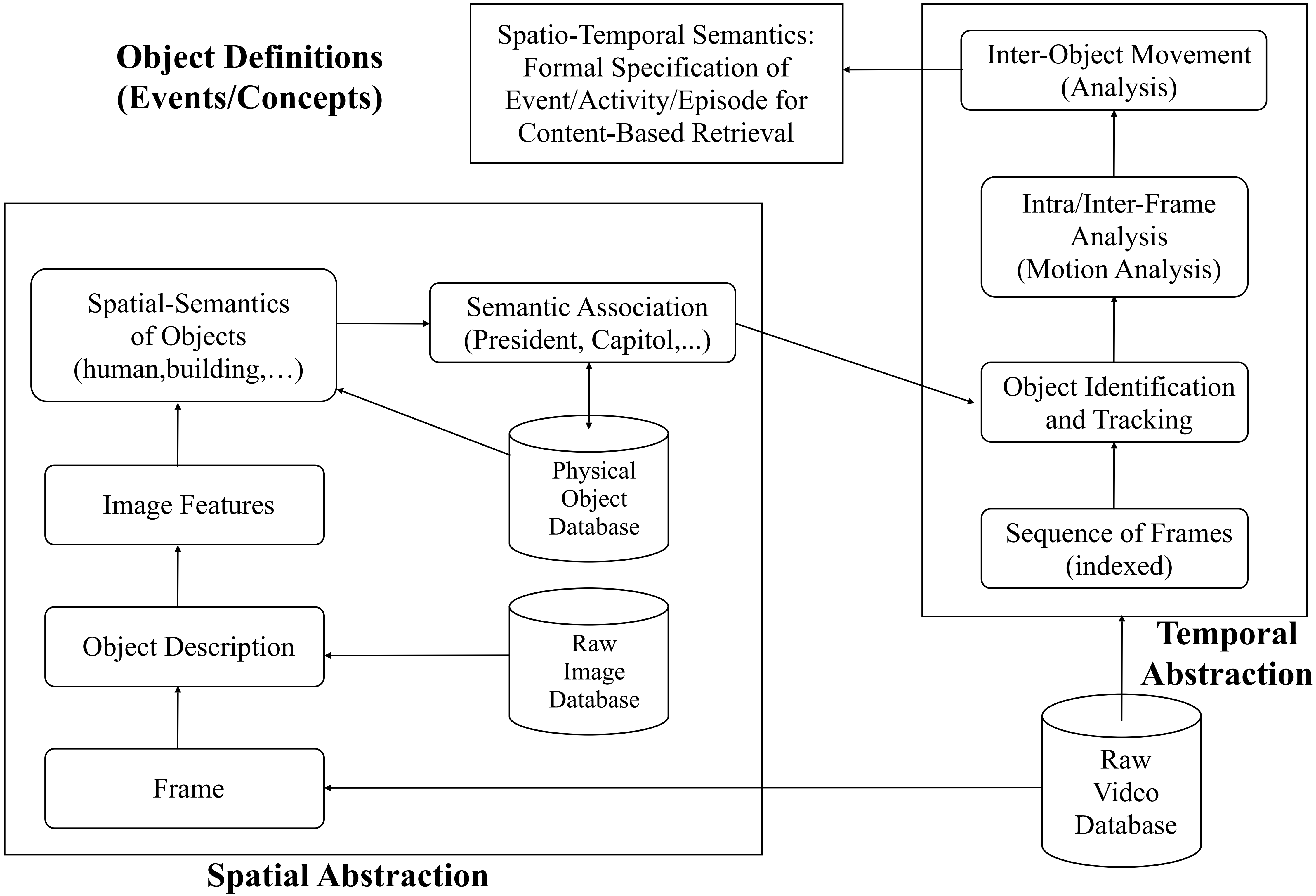
Software Architecture of MMDBMS



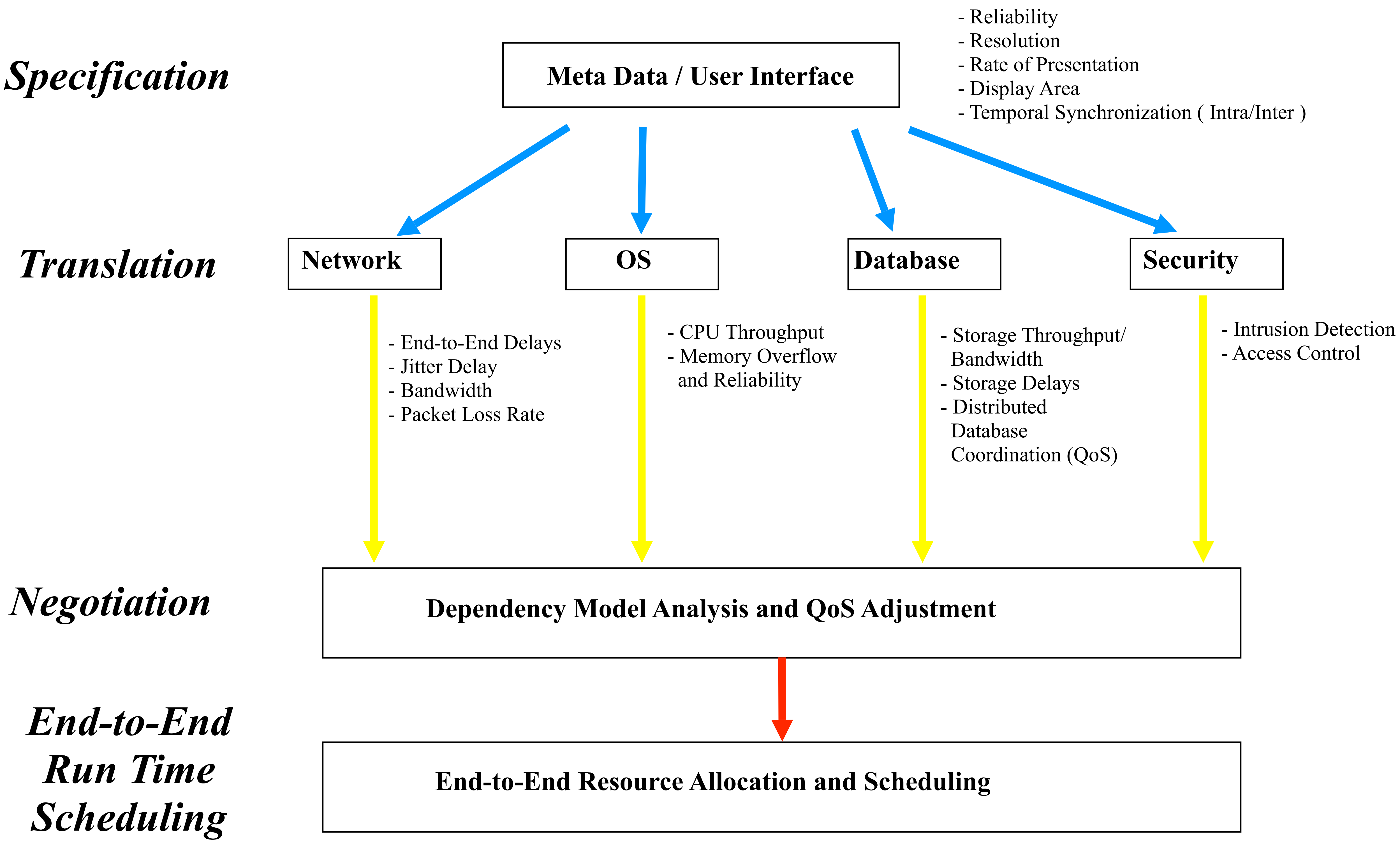
Distributed Multimedia Database Systems



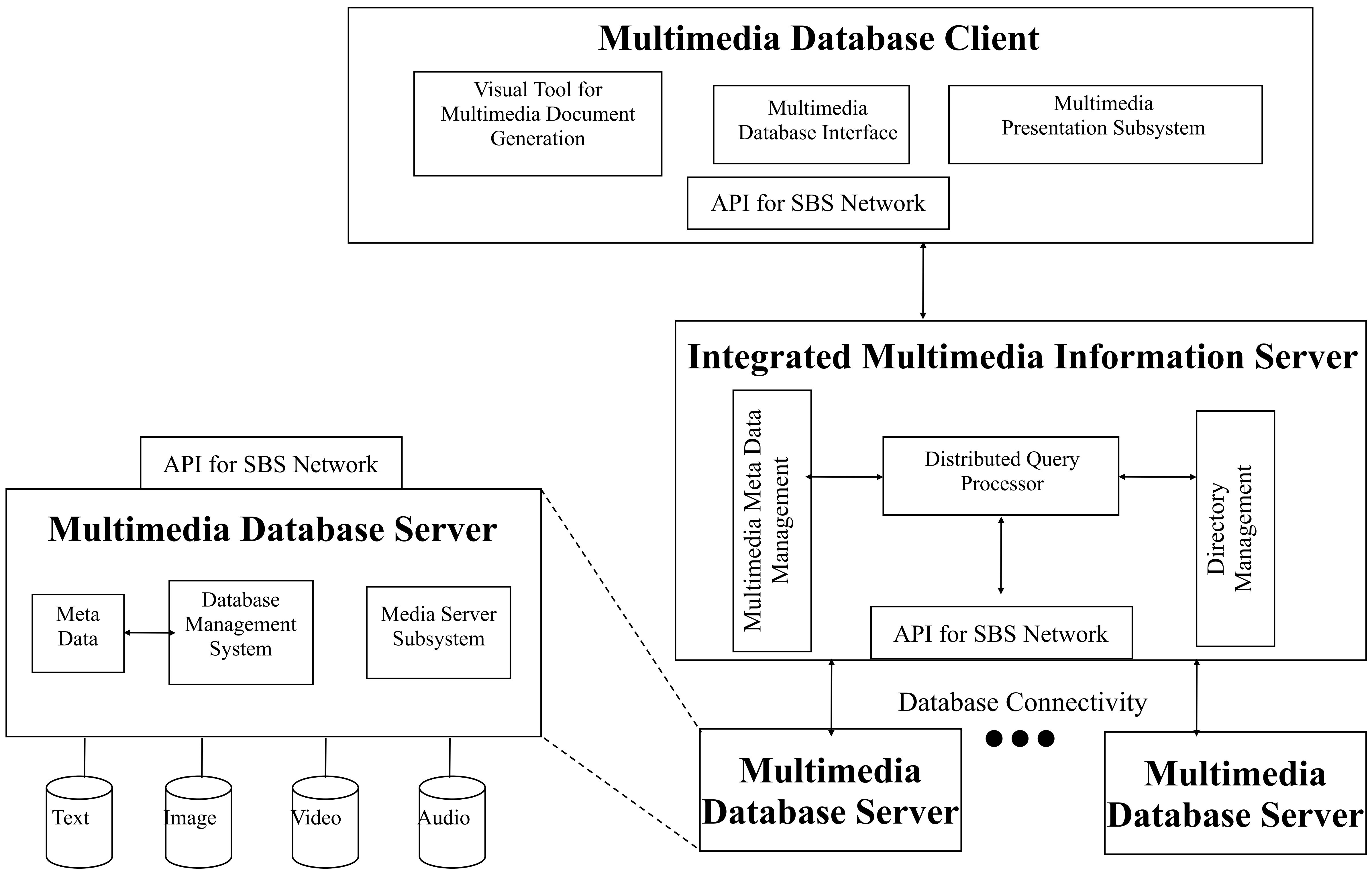
An Architecture for Video Database System



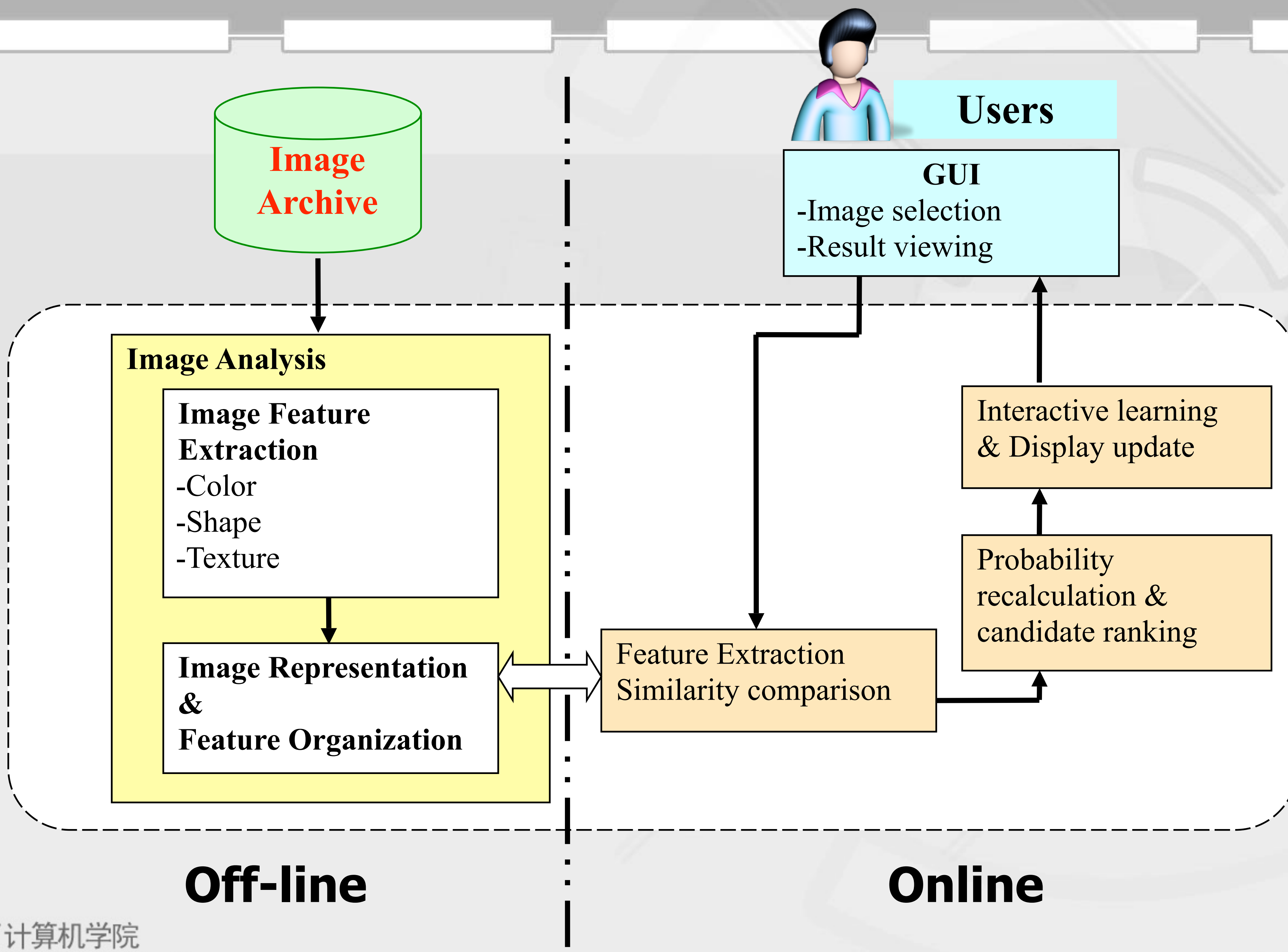
End-to-End QoP / QoS Management



Architecture of a Distributed Multimedia Database Management



Overview of the System



Outline

1. MM content organization

2. MM database system architecture

3. MM system service model

4. Multimedia Data Storage

5. Multimedia application

3.3 Multimedia System Service Model

What is a Media Service/Server?

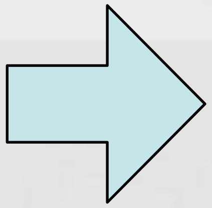
- A scalable storage manager
  - –Allocates multimedia data optimally among disk resources
  - –Performs memory and disk-based I/O optimization
- Supports
  - –real-time and non-real-time clients
  - –presentation of continuous-media data
  - –mixed workloads: schedules the retrieval of blocks
- Performs admission control

Service Models

- Random Access
  - **Maximize the number of clients** that can be served concurrently at any time with a low response time
  - **Minimize latency** (等待时间)
- Enhanced Pay-per-view (EPPV)
  - **Increase the number of clients** that can be serviced concurrently beyond the available disk and memory bandwidth, while guaranteeing a constraint on the response time

Service Models

- Example
  - **Server**



- 50 movies, 100 min. each
- Request rate: 1 movie/min
- Max. capacity: 20 streams

- Random Access Model
  - Case 1: after 20 movies, no more memory left. 21st movie waits for 80 minutes, 22nd movie waits for 81 minutes …
  - Case 2: after 20 movies, more memory can be allocated. 21st movie has to wait (initial latency) till one round of the previous 20 movies each has been served.
- EPPV Model:
  - At any time 20 movies are served, movies are initiated every 5 minutes
  - Streams are distributed uniformly during these 20 minutes

Outline

1. MM content organization

2. MM database system architecture

3. MM system service model

4. Multimedia Data Storage

5. Multimedia application

3.4 Multimedia Data Storage

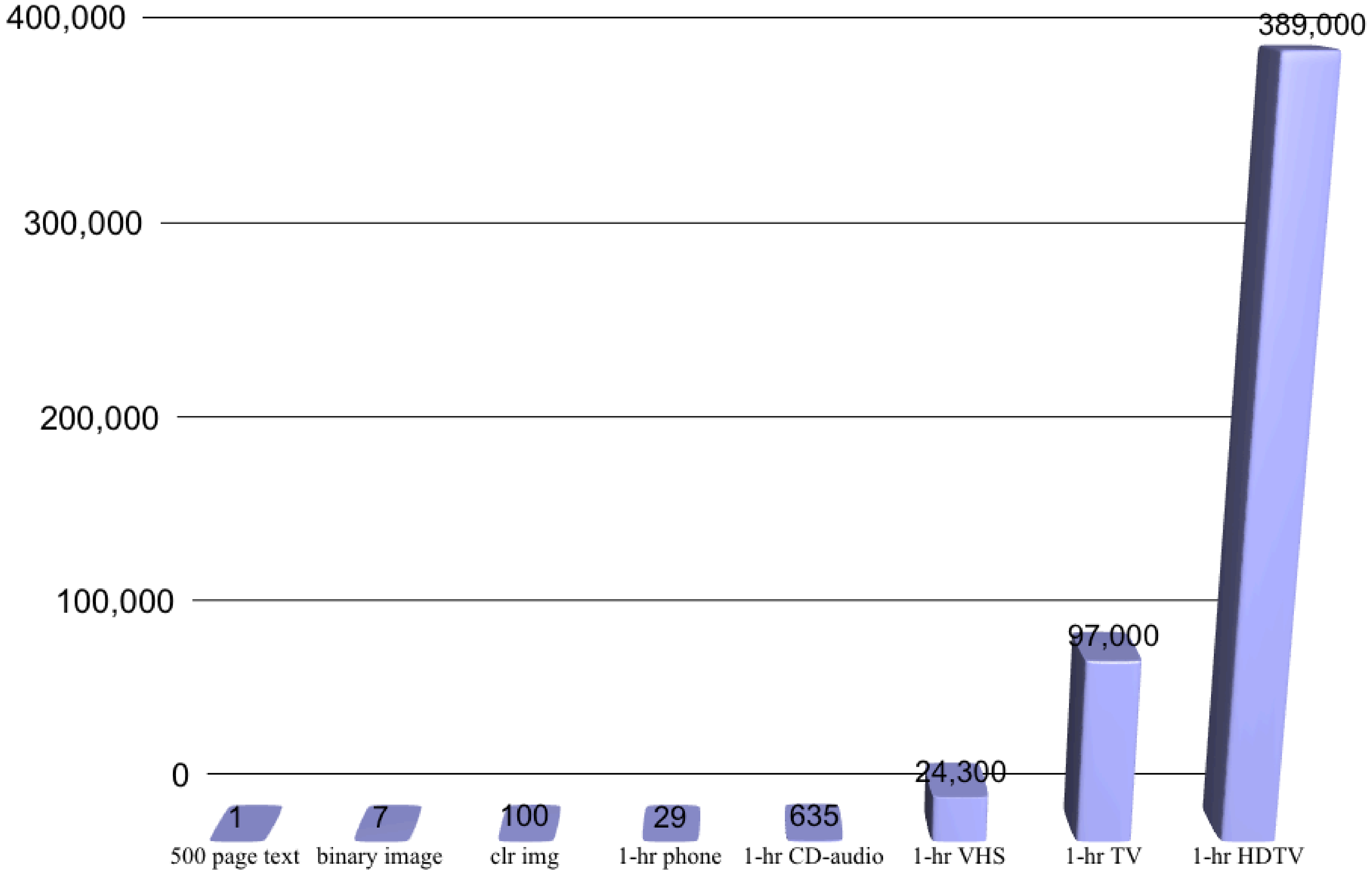
Multimedia Data Storage

- Storage Requirements
- RAID Technology
- Optical Storage Technology

Requirements of MM Information

- Storage and Bandwidth Requirement
  - measured in bytes or Mbytes for storage
  - measured in bits/s or Mbits/s for bandwidth
    - An image 480 x 600 (24 bits per pixel),
      - 864k bytes (without compression).
      - To transmit it within 2 sec => 3.456Mb/s.
    - 1GB Hard-disk
      - 1.5 hr. of CD-audio or
      - 36 seconds of TV quality video
      - require 800s to be transferred (10Mbits/s network).

Storage & Bandwidth Requirements



Delay and Delay Jitter Requirements

- Digital audio and video are **time-dependent** continuous media
- **dynamic media** => achieve **a reasonable quality playback** of audio and video, media samples must be received and played back at regular intervals.
- E.g. audio playback, 8K samples/sec have to be achieved
- **End-to-end delay is the sum of all delays** in all the components of a MM system, disk access, ADC, encoding, host processing, network access & transmission, buffering, decoding, and DAC

  In most conversation type applications, end-to-end delay should be kept below **300ms**

- Delay variation is commonly called **delay jitter**. It should be small enough to achieve smooth playback of continuous media, e.g.,

  < **10ms** for telephone-quality voice and TV-quality video,

  < **1ms** for stereo effect in high quality audio.

Other Requirements

Quest for Semantic Structure

- For alphanumeric information, **computer can search & retrieve alphanumeric items** from a DB or document collection.
- It is **hard to automatically retrieve** digital audio, image, & video as no semantic structure is revealed from the series of sampled values

Spatial-Temporal Relationship Among Related Media

- **Retrieval and transmission of MM data must be coordinated and presented** so that their specified temporal relationship are maintained for presentation
- A synchronization scheme therefore defines **the mechanisms used to achieve the required degree of synchronization**
- Two areas of works: user-oriented and system-oriented synchronization

Other Requirements

Error and Loss Tolerance

- Unlike alphanumeric information, we can tolerate some error or loss in MM
- For voice, we can tolerate a bit error rate of 10^{-2}
- For images and video, we can tolerate a bit rate from 10^{-4} to 10^{-6}.
- Another parameter: **packet loss rate** - a much more stringent requirement

Text v.s. MM Data Requirements

Characteristics	Text-based Data	Multimedia Data
Storage Req.	Small	Large
Data Rate	Low	High
Traffic Pattern	Bursty	Stream-oriented, highly bursty
Error/Reliability Req.	No loss	Some loss
Delay/Latency Req.	None	Low
Temporal Relationship	None	Synchronized Trans.

Quality of Service (QoS)

- To provide a uniform framework to specify and guarantee these diverse requirement, a concept called QoS has been introduced.
- QoS is **a set of requirement, but there is no universally agreed one**.
- QoS is **a contract** negotiated and agreed **among MM applications and MM system** (service provider)
- The QoS requirement is normally specified in two grades: the preferable quality and the acceptable one.
- The QoS guarantee can be in one of three forms: hard or deterministic (fully satisfied), soft or statistic (guaranteed with a certain probability), and best effort (no guarantee at all)
- A lot of research issues are involved and still undergoing!!

File Systems

- The most visible part of an operating system.
- organization of the file system
  - an important factor for the usability and convenience of the operating system.
- Files are stored in secondary storage, so they can be used by different applications.
- In traditional file systems, the information types stored in files are sources, objects, libraries and executables of programs etc.
- In multimedia systems, the stored information also covers digitized video and audio with their related real-time “**read**” and “**write**” demands.
- ===>>> additional requirements in the design and implementation

File Systems

Traditional File Systems

- The main goals of traditional files systems are:
  - to **provide** a comfortable interface for file access to the user
  - to **make** efficient use of storage media
  - to **allow** arbitrary deletion and extension of files

Multimedia File Systems

- the main goal is to provide a **constant and timely retrieval** of data.
- It can be achieved through providing enough buffer for each data stream and the employment of disk scheduling algorithms, especially optimized for real-time storage and retrieval of data.

Multimedia File Systems

- The much greater size of continuous media files and the fact that they will usually be retrieved sequentially are reasons for an ***optimization of the disk layout***
- Continuous media streams predominantly **belong to the *write-once-read-many nature* (ROM?)**, and streams that are recorded at the same time are likely to be played back at the same time.
- Hence, it seems to be reasonable to ***store continuous media data in large data blocks contiguously on disk***.
- Files that are likely **to be retrieved together are grouped together on the disk**.
- With such a disk layout, the buffer requirements and seek times decrease.
- The disadvantage of the continuous approach is **external fragmentation** and **copying overhead** during insertion and deletion.

Data Management & Disk Spanning

Data Management:

- **Command queuing**: allows execution of multiple sequential commands with system CPU intervention. It helps in minimizing head switching and disk rotational latency.
- **Scatter-gather**: scatter is a process whereby data is set for best fit in available block of memory or disk. Gather reassembles data into contiguous blocks on disk or in memory.

Disk Spanning

- Attach **multiple devices** to a **single host adapter**.
- good way to increase storage capacity by adding incremental drives.

RAID Redundant Arrays of Inexpensive Disks

– **By definition RAID has three attributes:**

- a set of disk drives viewed by the user as one or more logical drives
- data is distributed across the set of drives in a pre-defined manner
- redundant capacity or data reconstruction capability is added, in order to recover data in the event of a disk failure

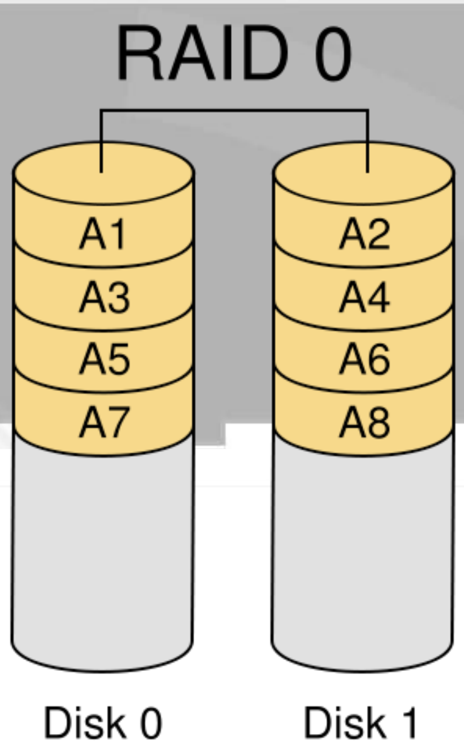
– **Objectives of RAID**

- Hot backup of disk systems (as in mirroring)
- Large volume storage at lower cost
- Higher performance at lower cost
- Ease of data recovery (fault tolerance)
- High MTBF (mean time between failure)

Different Levels of RAID

- **Eight discrete levels of RAID functionality**
  - Level 0 - disk striping
  - Level 1 - disk mirroring
  - Level 2 - bit interleaving and Hamming Error Correction (HEC) parity
  - Level 3 - bit interleaving and XOR parity
  - Level 4 - block interleaving with XOR parity
  - Level 5 - block interleaving with parity distribution
  - Level 6 - Fault tolerant system
  - Level 7 - Heterogeneous system
- Data is spread across the drives in units of **512 bytes** called **segments**. **Multiple segments form a block.**

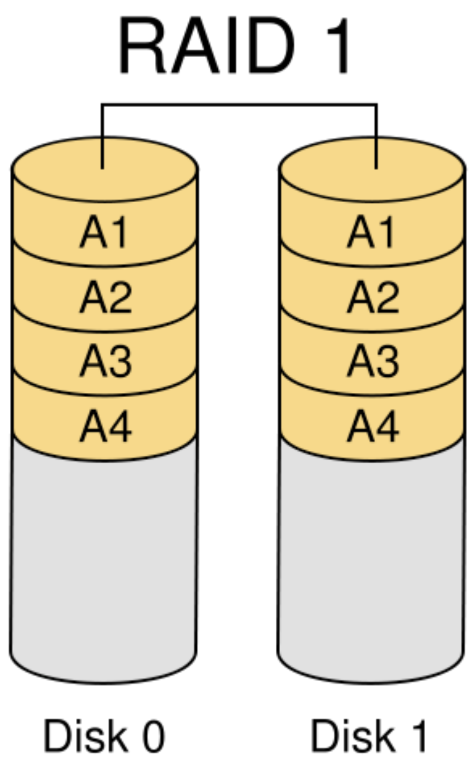
RAID Level 0 - Disk Striping



- To improve performance by overlapping disk reads and writes
- Multiple drives connected to a single disk controller
- Data is striped to spread segments of data across multiple drives in block sizes ranging from 1 to 64 Kbytes
- Disk striping provides a higher transfer rate for write and retrieve block of data
- Typical application: database applications
- Drawbacks:
  - If one drive fails, the whole drive system fails
  - Does not offer any data redundancy, no fault tolerance

RAID Level 1 - Disk Mirroring

- Each main drive has a **mirror drive**
- Two copies of every file will write to two separate drives
  complete redundancy



- Performance:
  * Disk write : take almost twice time
  * Disk read : can be speed up by overlapping seeks
- Typical use:
  * in file servers provides backup in the event of disk failure
- Duplexing:
  * Use two separate controllers
  * The second controller enhances both fault tolerance and performance
  * Separate controllers allow parallel writes and parallel reads

RAID Level 2
- Bit Interleaving and HEC Parity

- Contain arrays of multiple drives connected to a disk array controller.
- Data is written interleaved across multiple drives (often one bit at a time) and multiple check disks are used to detect and correct errors.
- **Hamming error correction (HEC)** code is used for error detection and correction.
- The drive spindles must be **synchronized** as a single I/O operation accesses all drives
- Benefits:
  - High level of data integrity and reliability (error correction feature)
  - Mainly use for **supercomputers** to access large volumes of data with a small number of I/O request.

RAID Level 2

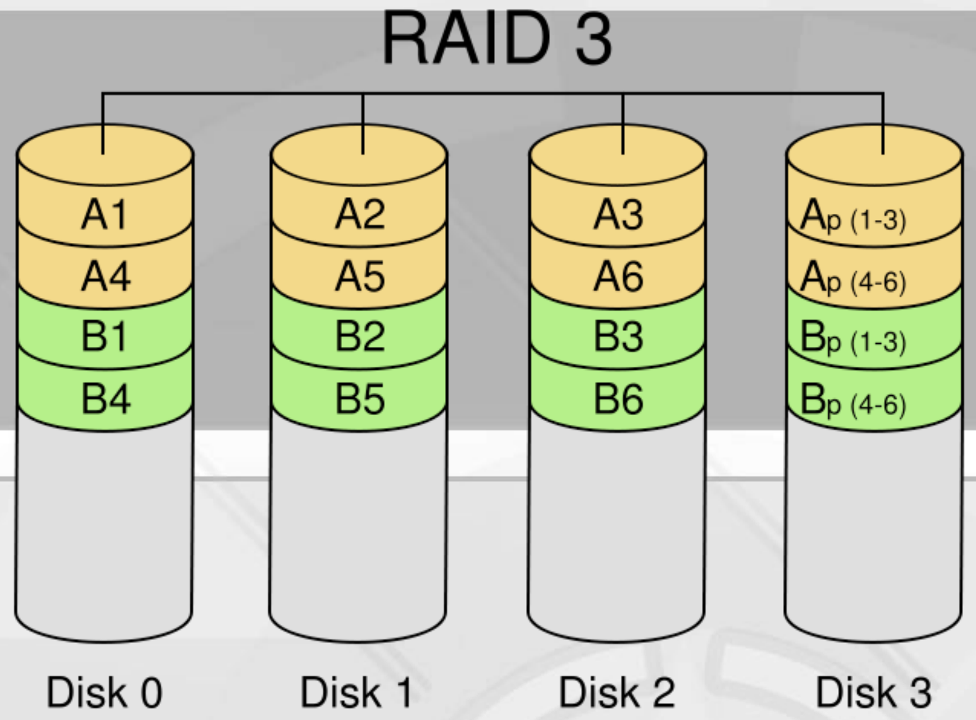
- Bit Interleaving and HEC Parity

Drawbacks:

- **Expensive** - requires **multiple drives** for error detection and correction
- **Error-correcting scheme**: **slow** and **cumbersome**
- Multimedia applications can afford to **lose occasional bit** or there without any significant impact on the system or the display quality.
- Each sector on a drive is associated with sectors on other drives to form a single storage unit, it takes multiple sectors across all data drives to storage even just a few bytes, **resulting in waste of storage**.
- Should not be used for transaction processing where **the data size of each transaction is small**.

RAID Level 3

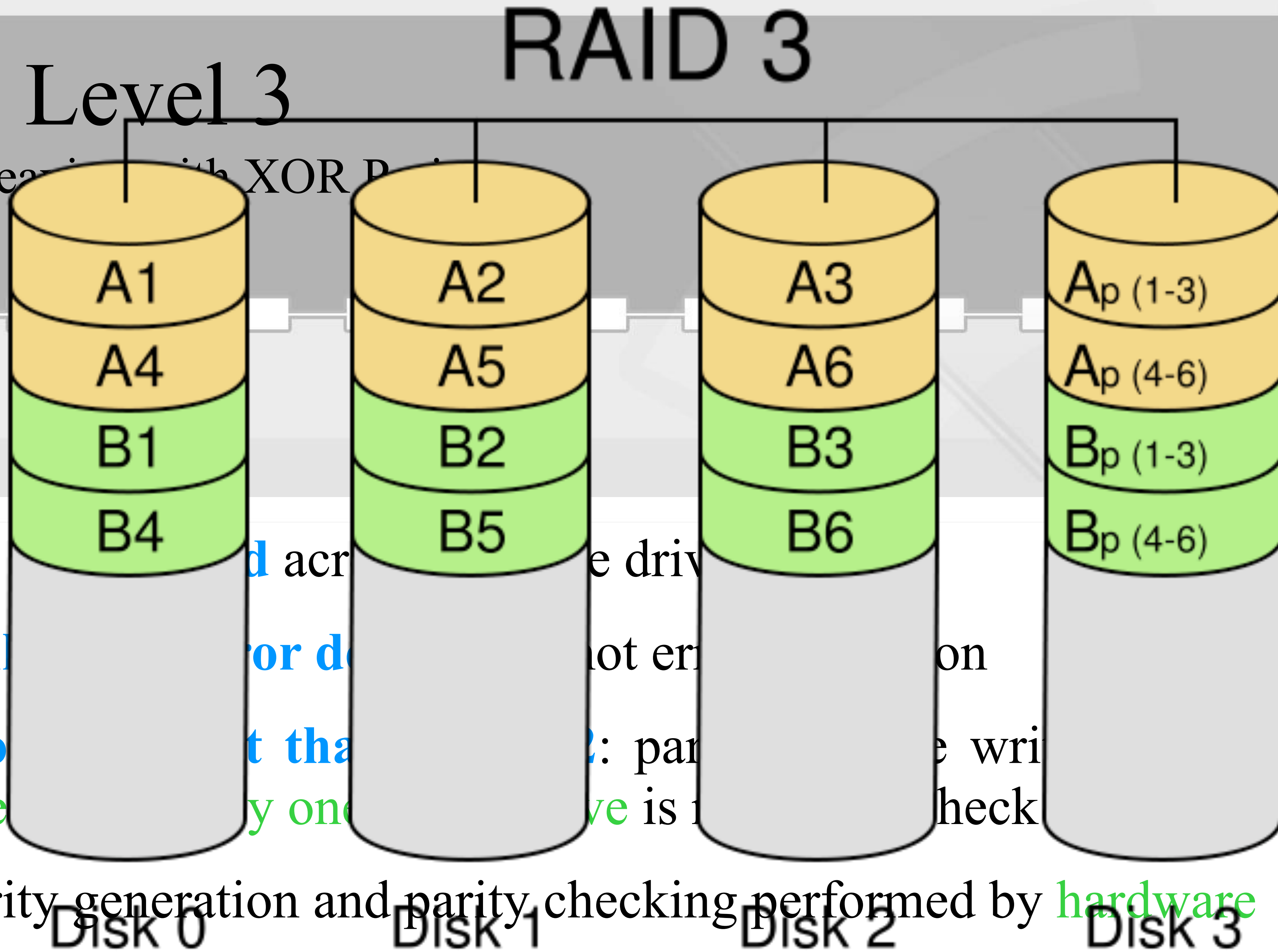
- Bit Interleaving with XOR Parity



- **Bit interleaved** across multiple drives
- **Only offer error detection** - not error correction
- **More efficient than RAID 2**: parity bits are written into the data stream and only one parity drive is needed to check data accuracy.
- Parity generation and parity checking performed by hardware
- **Not suitable for small transaction**
- **Good for supercomputer and data server**: large sequential I/O request

RAID Level 3

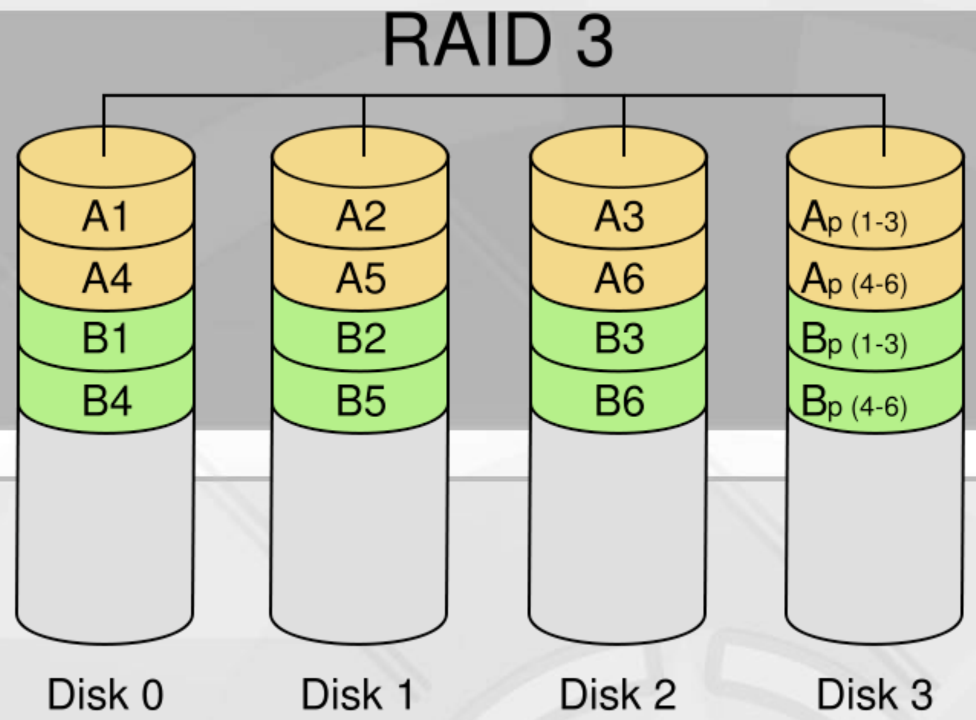
- Bit Interlea... ...th XOR P...



- **Bit** ... **d** acr... e driv...
- **Onl** ... **or d**... ot er... on
- **Mo**... **t tha**... : par... e wri... e data stre... y one ... ve is ... heck... acy.
- Parity generation and parity checking performed by hardware
- **Not suitable for small transaction**
- **Good for supercomputer and data server**: large sequential I/O request

RAID Level 3

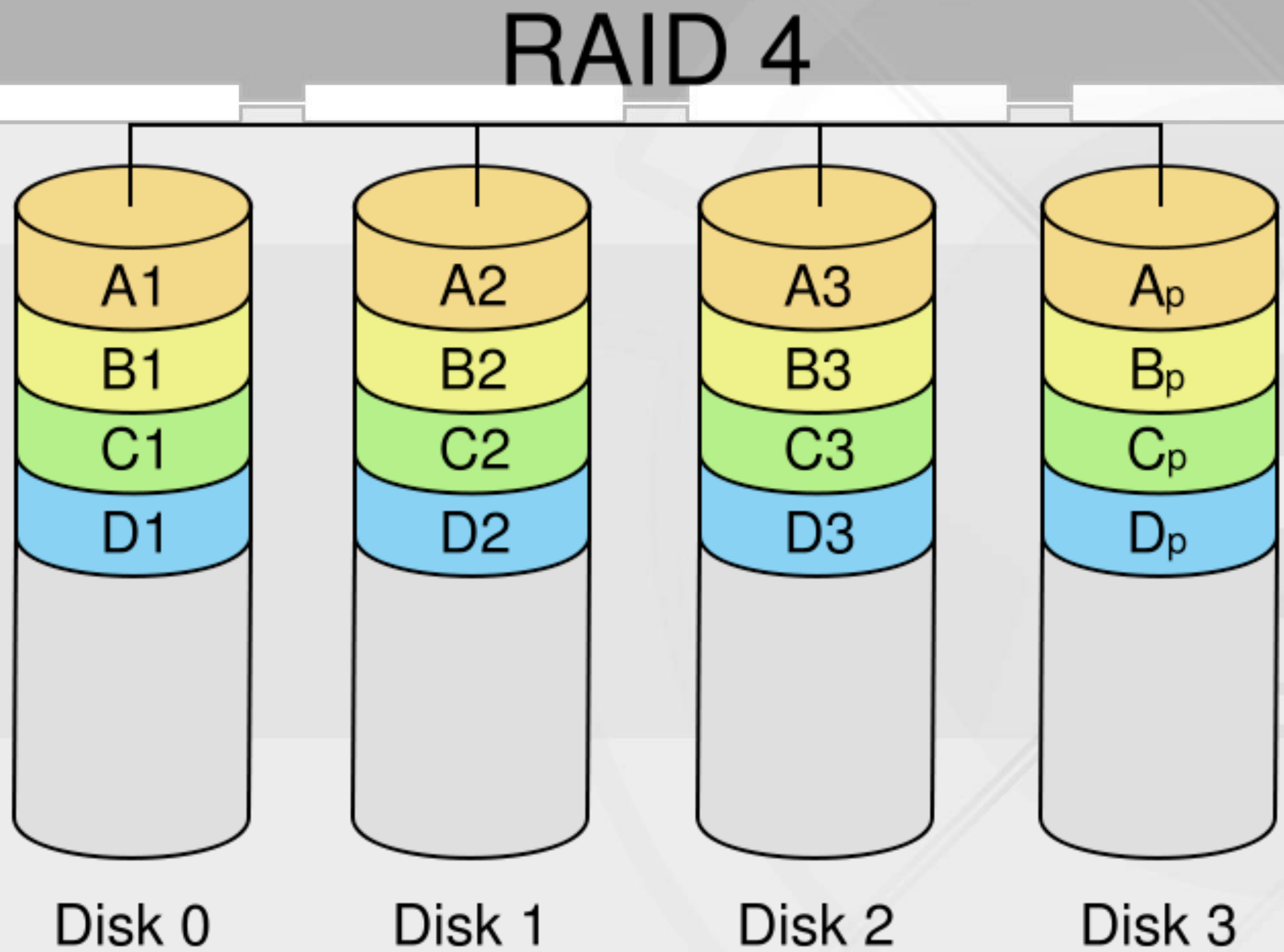
- Bit Interleaving with XOR Parity



- **Bit interleaved** across multiple drives
- **Only offer error detection** - not error correction
- **More efficient than RAID 2**: parity bits are written into the data stream and only one parity drive is needed to check data accuracy.
- Parity generation and parity checking performed by hardware
- **Not suitable for small transaction**
- **Good for supercomputer and data server**: large sequential I/O request

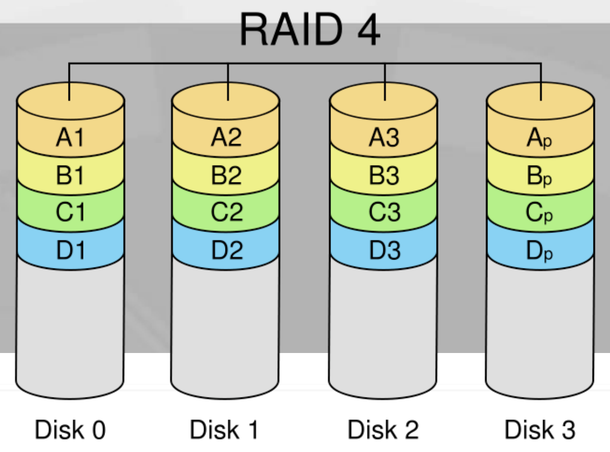
RAID Level 4

- Block Interleaving with XOR Parity



RAID Level 4

- Block Interleaving with XOR Parity

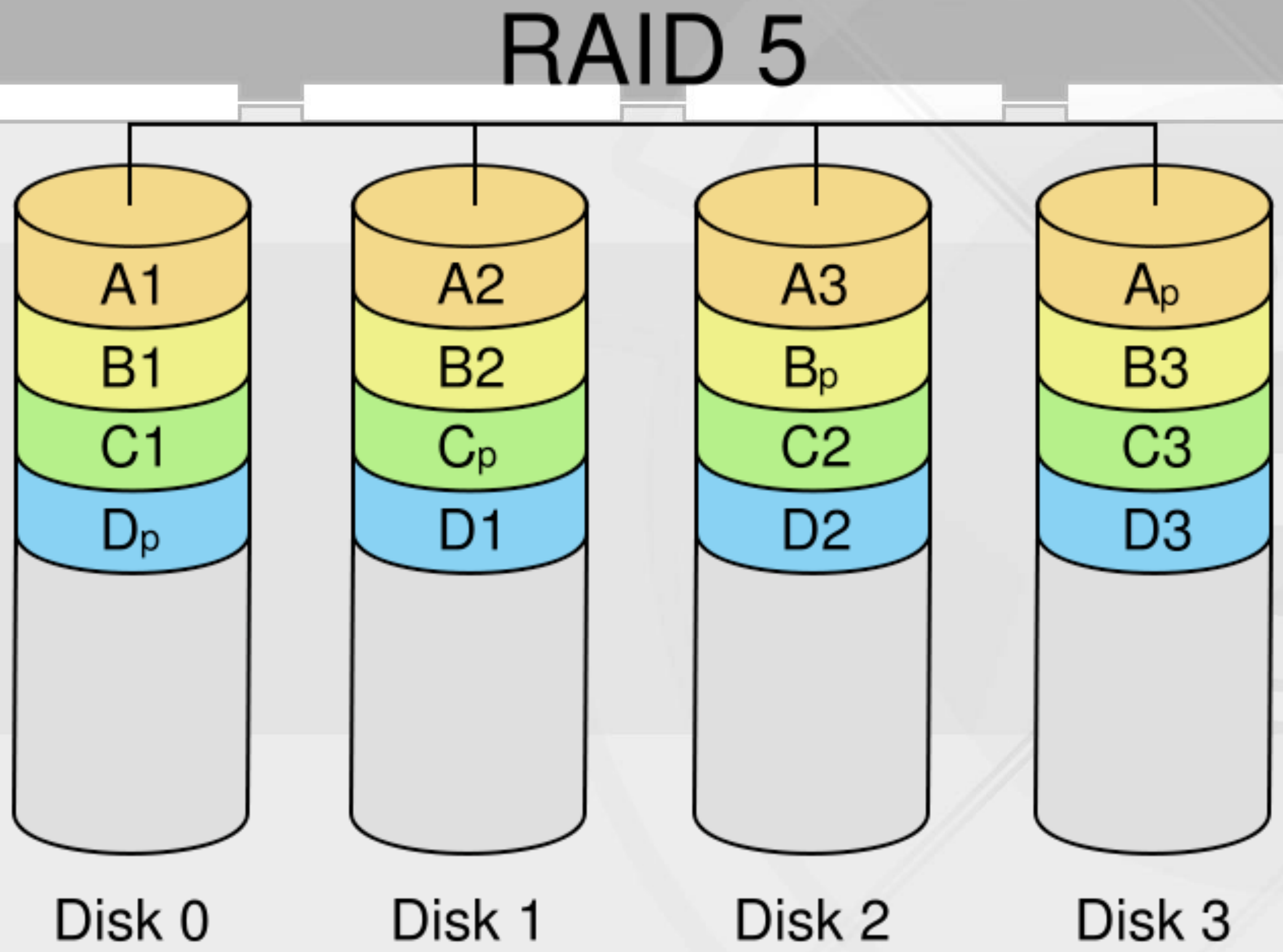


- **Write successive blocks of data on different drives.**
- Data is interleaved at block level.
- RAID 4 access is to individual strips rather than to all disks at once (as in RAID 3); therefore disks operate individually
- Separate I/O requests can be satisfied
- Good for applications that require high I/O request rates but bad for applications that require high data transfer rate
- Bit-by-bit parity is calculated across corresponding strips on each disk
- Parity bits stored in the redundant disk
- Write penalty
  - For every write to a strip, the parity strip must also be recalculated and written, i.e., updated (by an array management software)
  - When an I/O write request of small size is performed, RAID 4 involves a write penalty.

浙江大学计算机学院
数字媒体与网络技术

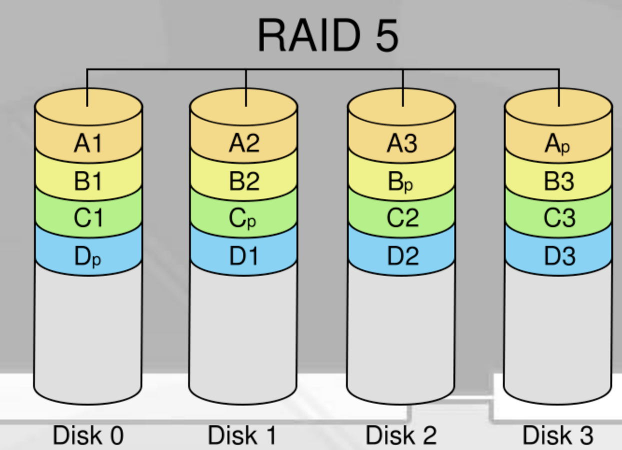
RAID Level 5

- Block Interleaving with Parity Distribution



RAID Level 5

- Block Interleaving with Parity Distribution

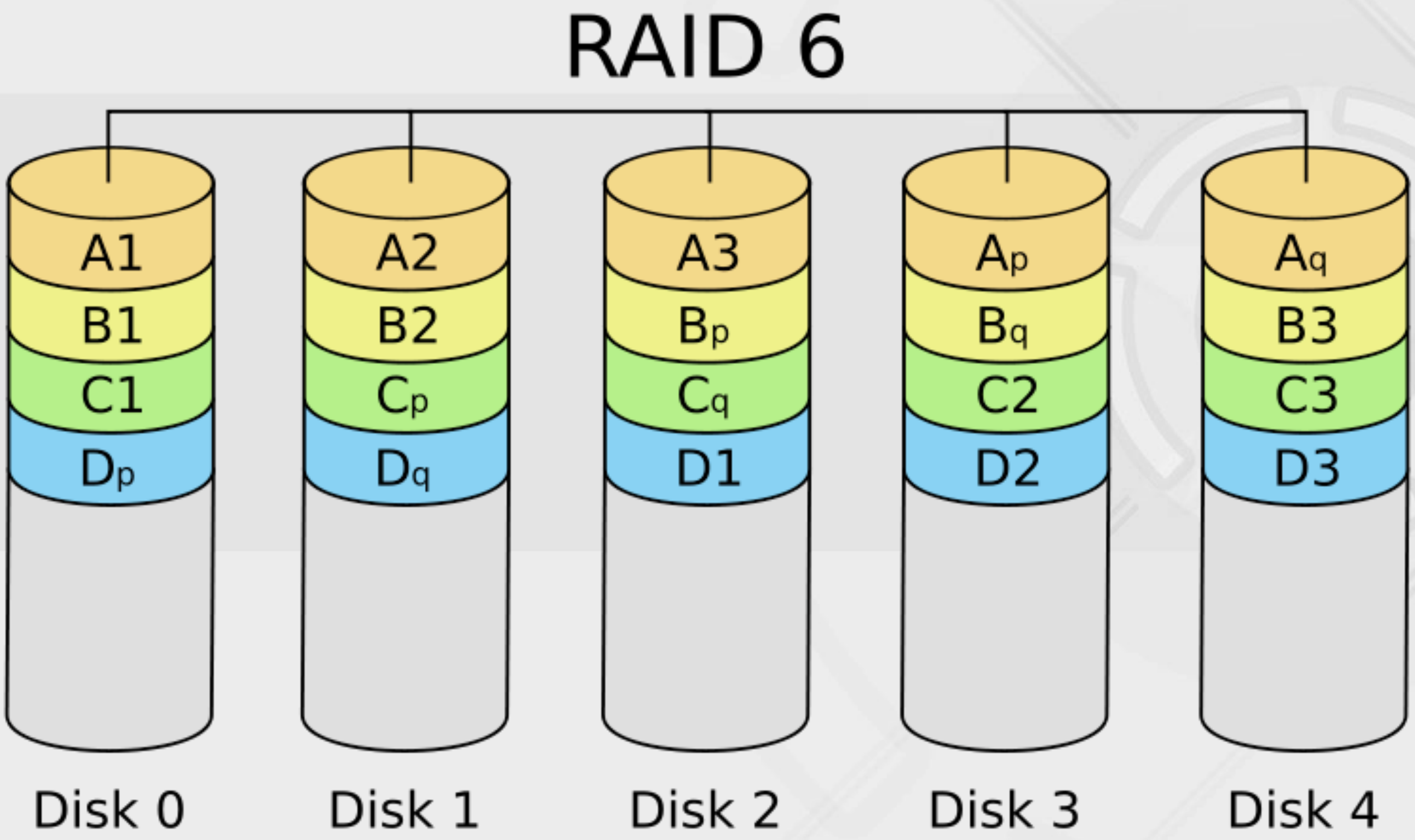


- RAID 5 is organized in a similar fashion to RAID 4 but avoids the bottleneck encountered in RAID 4.
- It does not use a dedicated parity drive
- Parity data is interspersed in the data stream and spread across multiple drives.
- Block of data falling within the specified block size requires only a single I/O access.
- Block of data are stored on a different drive, multiple concurrent block-sized accesses can be initiated.
- Good for database applications in which **most I/O occurs randomly and in small chunks.**
- Drawbacks: **high cost and low performance** for large block sizes objects such as audio and video.

浙江大学计算机学院
数字媒体与网络技术

RAID Level 6-7

- Fault-Tolerant and Heterogeneous System



RAID Level 6-7

- Fault-Tolerant and Heterogeneous System

- RAID 6 has become a common feature in many systems. RAID 6 is an improvement over RAID 5 model through the addition error recovery information.

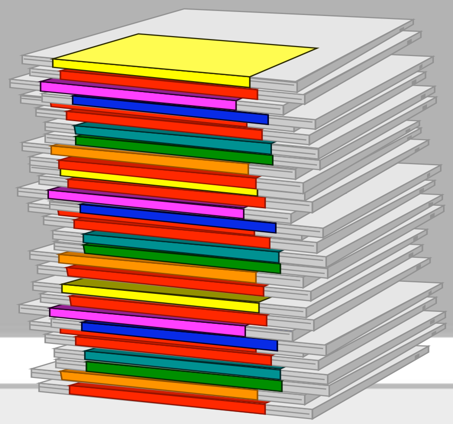
- Conceptually, the disks are considered to be in a matrix formation and the parity is generated for rows and for columns of disks in the matrix. The multi-dimensional level of parity is computed and distributed among the disks in the matrix.

- RAID 7 is the most recent development in the RAID taxonomy. Its architecture allows **each individual drive to access data as fast as possible** by incorporating a few crucial features.

- With the growth in **the speed of computers** and communications in response to **the demands for speed & reliability**, the RAID theme has begun to attract significant attention as a potential **mass storage solution** for the future.

Data Storage

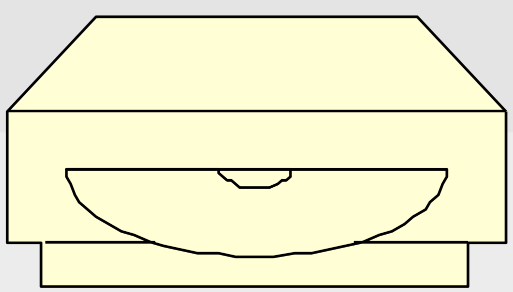
- The strategy adopted for data storage will depend on the **storage technology, storage design,** and **the nature of data itself**.
- Any storage has the following parameters:
  - Storage capacity
  - Standard operations of Read and Write
  - Unit of transfer for Read and Write
  - Physical organization of storage units
  - Read-Write heads, Cylinders per Disc, Tracks per Cylinder, and Sectors per Track
  - Read time and seek time
- Of the storage technologies that are available as computer peripherals, the optical medium is the most popular in the multimedia context.



Magnetic

- **Hard Disk**
- **Floppy Disk**
- **PCMCIA**

Advantages:	- Faster than tape - Allows direct access to data
Disadvantages:	- Performance relies on speed of mechanical heads - Neither fault nor damage resistant



Optical

- **CD-ROM, DVD**
- **Magneto-Optical Disk**

Advantages:	-More data capacity than magnetic disk -High quality storage of sound and images
Disadvantages:	-Data capacity is small for videos in CD and DVD are better -Limited Data densities

Outline

1. MM content organization

2. MM database system architecture

3. MM system service model

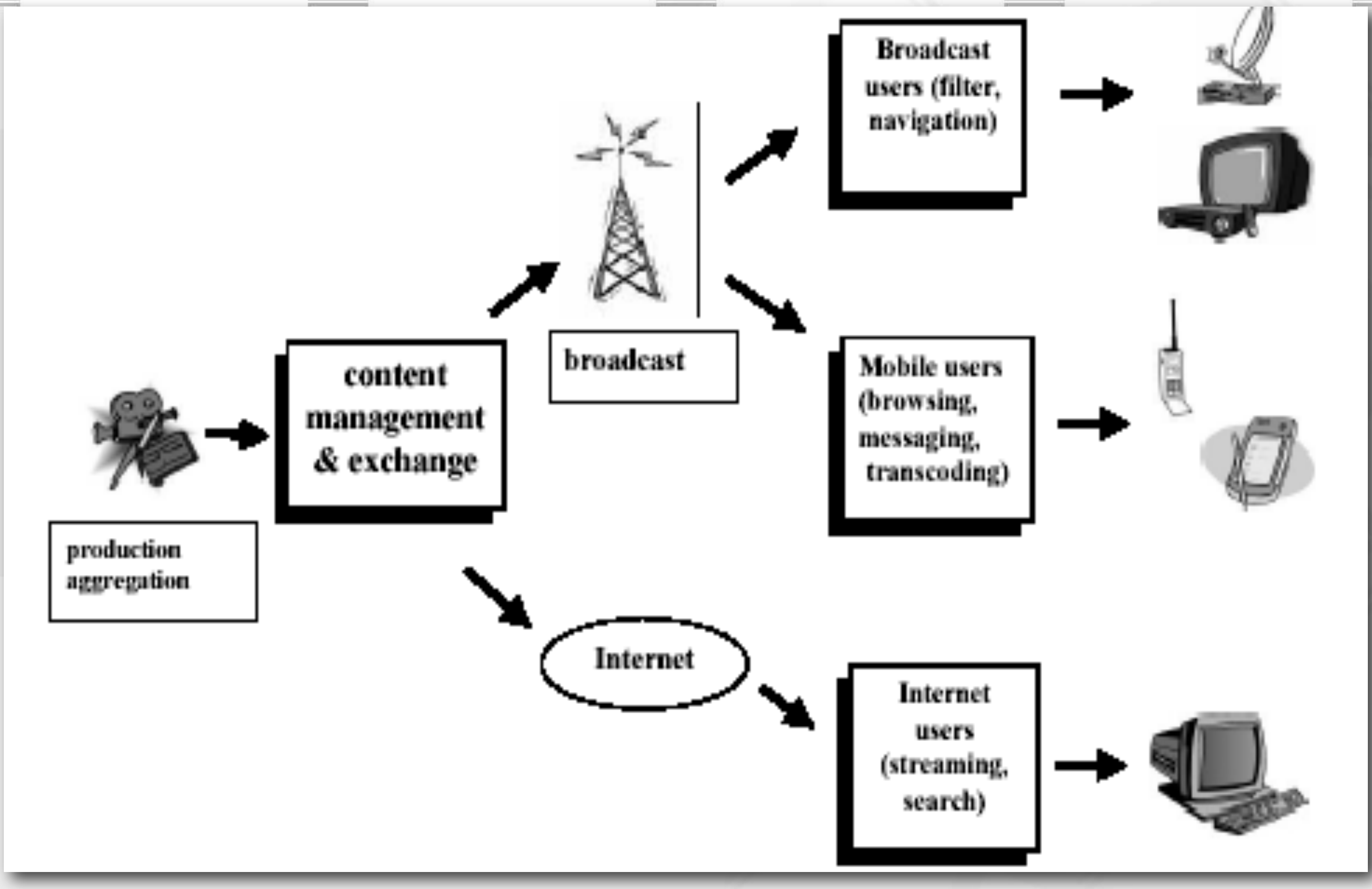
4. Multimedia Data Storage

5. Multimedia application

3.5 Multimedia System Application

Multimedia Systems Application Chain

Multimedia Systems Application Chain



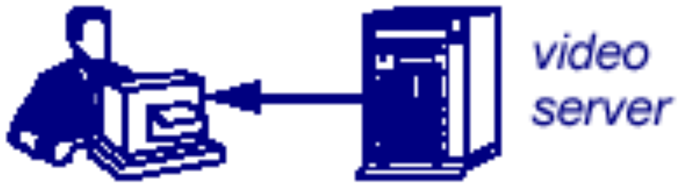
Applications of Multimedia

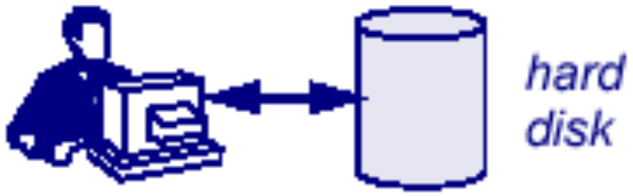
Application classes:

- Local
- Distributed

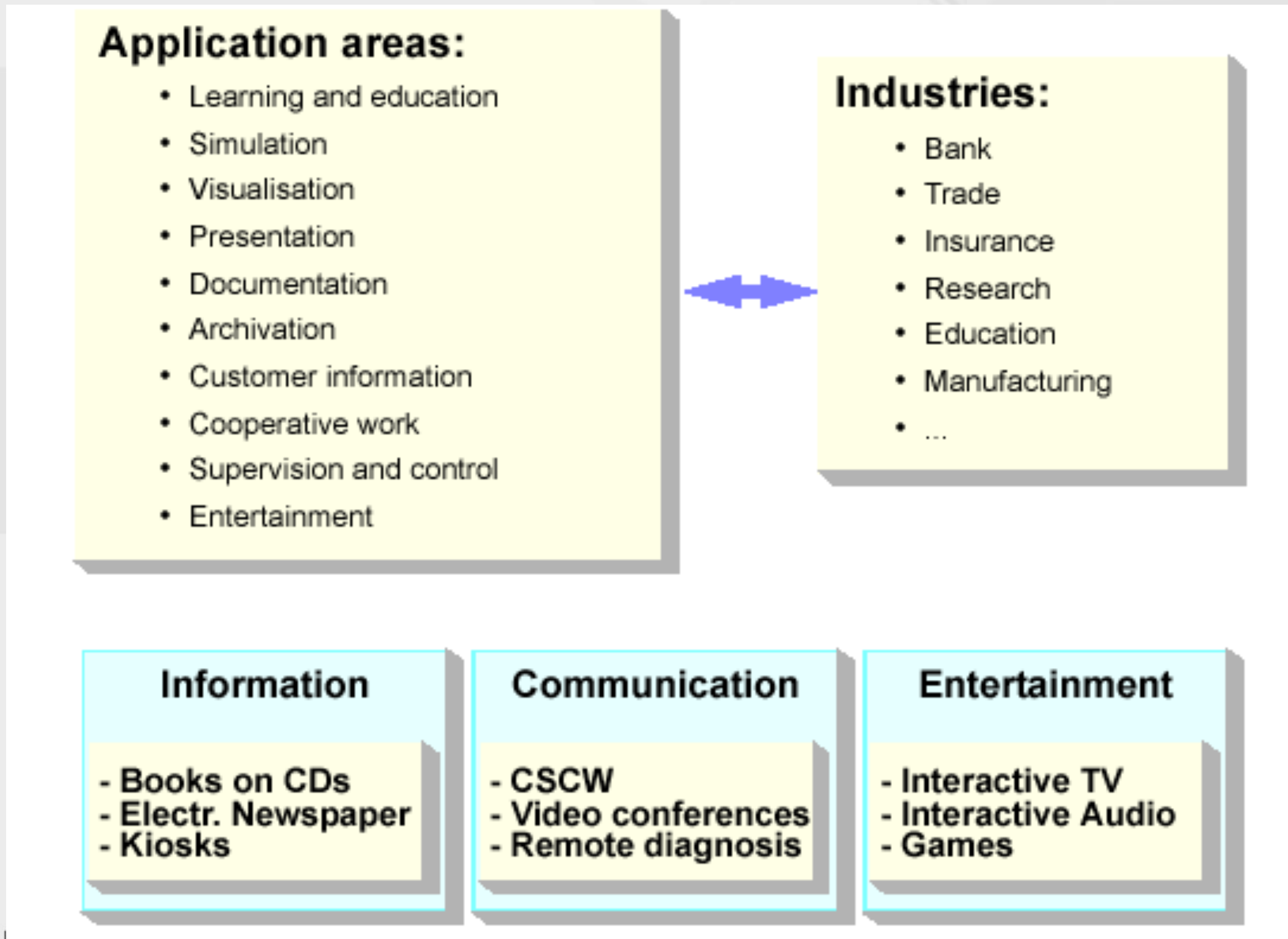
Basic multimedia services:

- Interpersonal communication
- Information retrieval
- Information recording and editing





Application Areas, Industries and Usage



Multimedia Applications

- Hypermedia courseware
- Video conferencing
- Video on demand
- Interactive TV
- Home shopping
- Game
- Digital video editing and production systems

Q&A