SAMT’08

Semantic-driven multimedia retrieval with the MPEG Query Format

Ruben Tous and Jaime Delgado

Distributed Multimedia Applications Group (DMAG)
Universitat Politècnica de Catalunya (UPC)
Dept. Computer Architecture
Barcelona (Spain)

Presentation outline

- The MPEG Query Format (MPQF)
- MPQF evaluation model
- How to extend MPQF to allow SPARQL-like queries
- MPQF and metadata interoperability

MPEG Query Format (MPQF)

MPEG Query Format (MPQF): Introduction

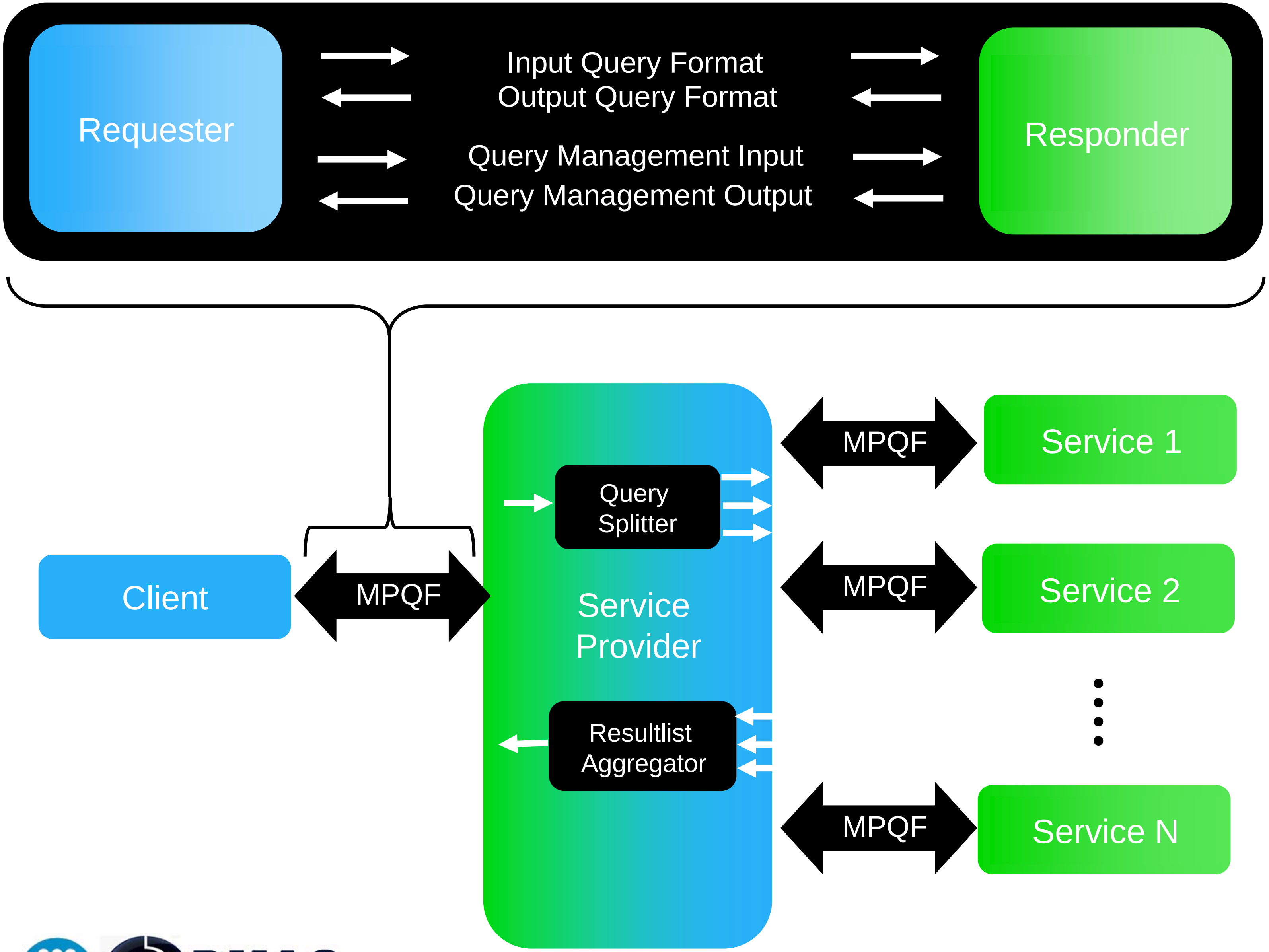
- ISO/IEC 15938-12 Query Format (MPQF)
  - Recently reached the International Standard level

- MPQF: “Provides a standardized interface for multimedia content retrieval systems (e.g.: MPEG-7 databases)”

- Allows combining Data Retrieval like conditions (XPath, XQuery) with Information Retrieval like conditions (Query-by-Example, etc.)

- Is an XML application: One XML schema describing messages:
  - Requester -> responder = Input Query Format
  - Responder -> requester = Output Query Format
  - Management messages (service properties, etc.)

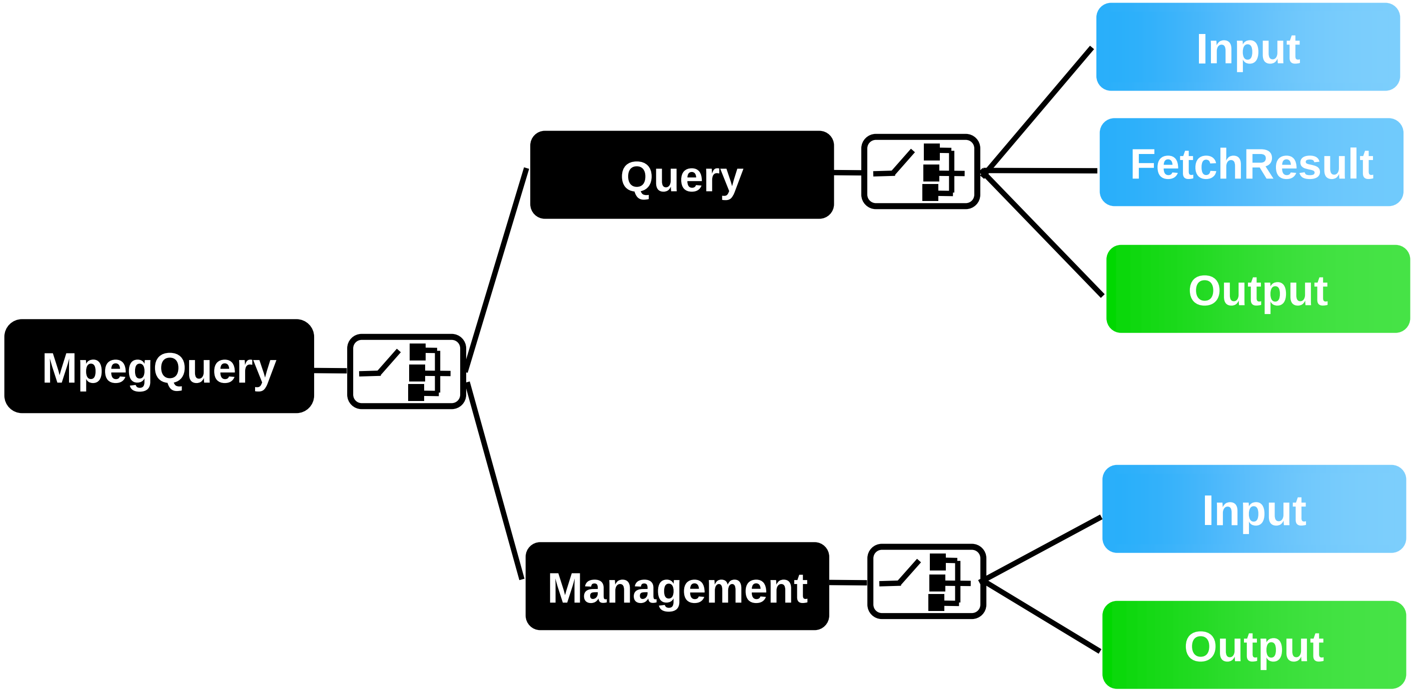
- Part 12 of ISO/IEC 15938 (MPEG-7), but **METADATA-NEUTRAL**



DMAG
DISTRIBUTED MULTIMEDIA APPLICATIONS GROUP

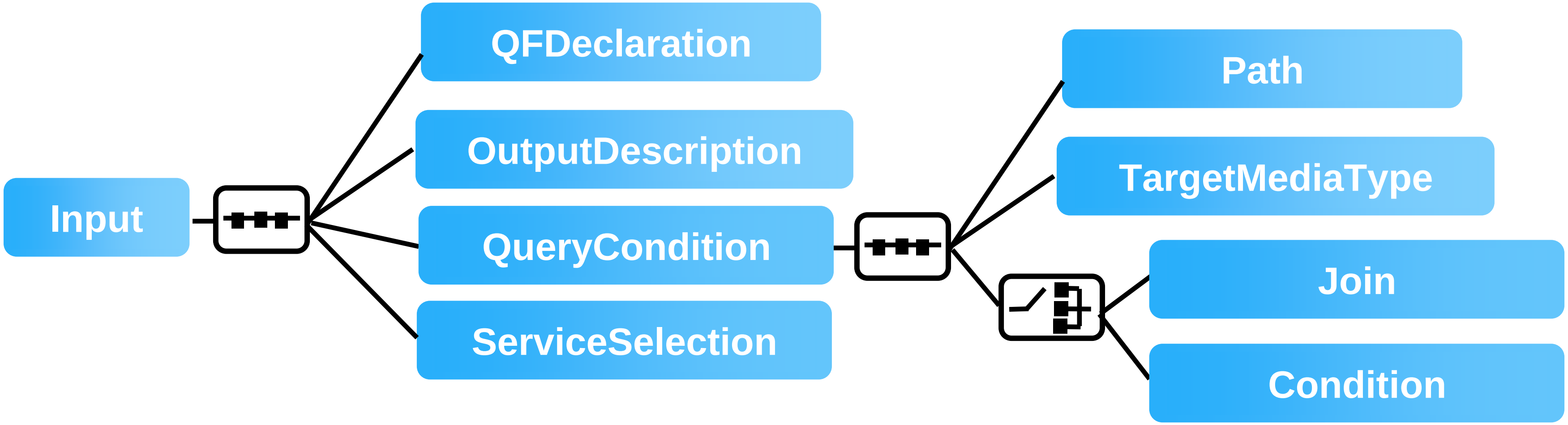
MPEG Query Format (MPQF)

- Schema overview of the uppermost elements of MPQF

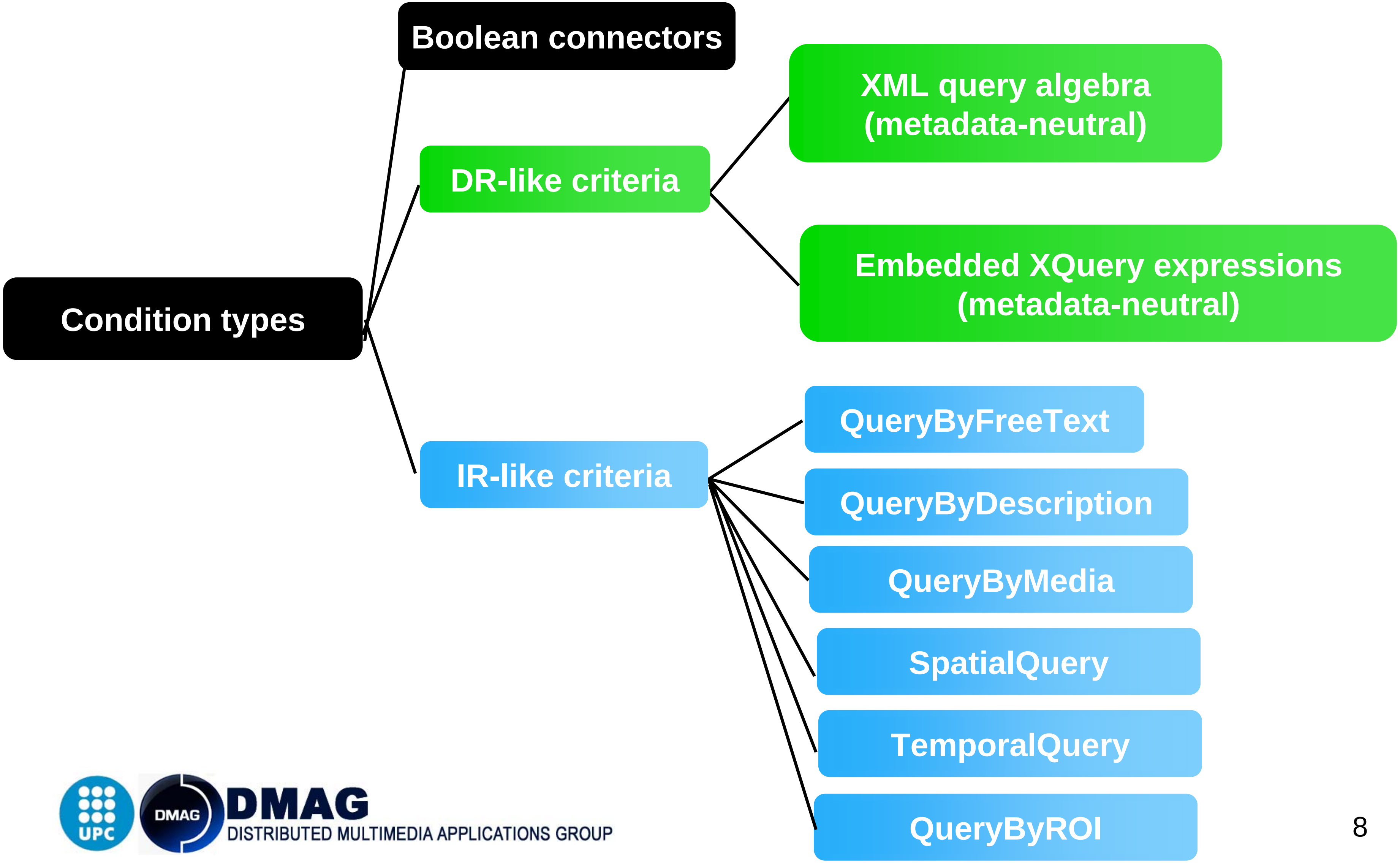


MPEG Query Format (MPQF)

- Input Query Format (IQF)

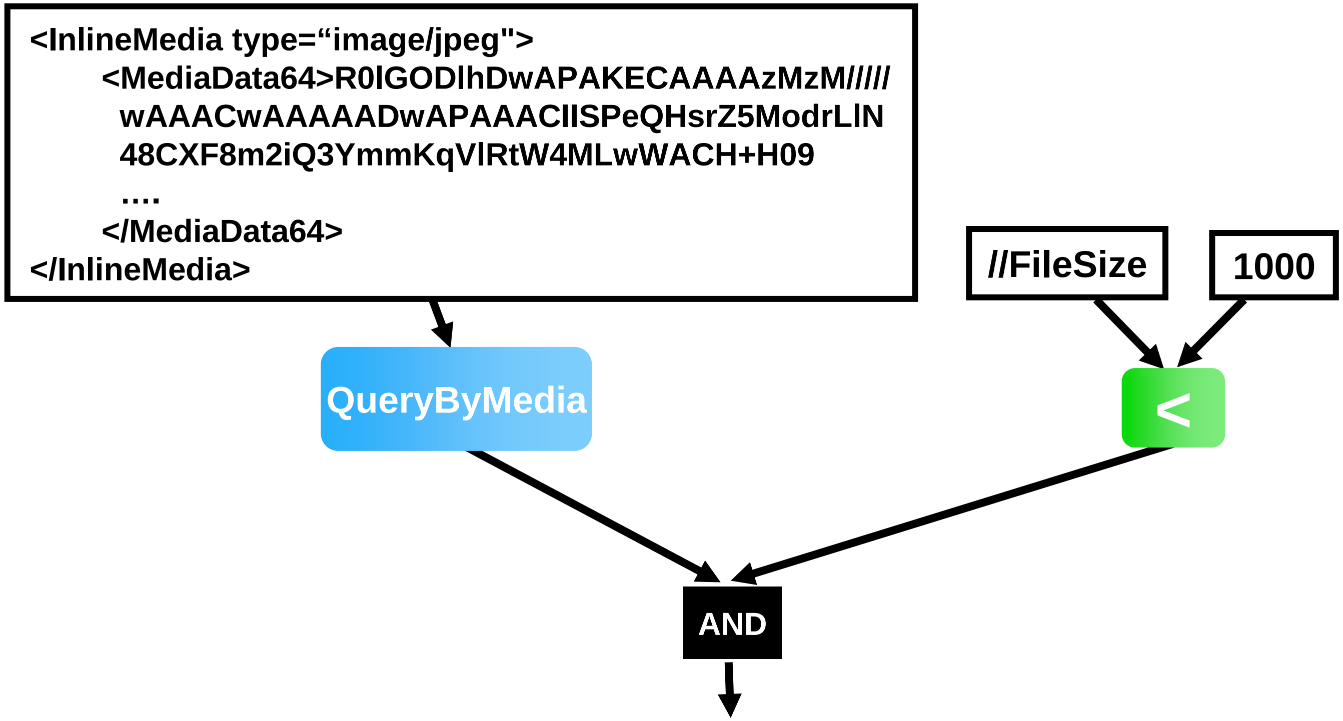


MPEG Query Format (MPQF)



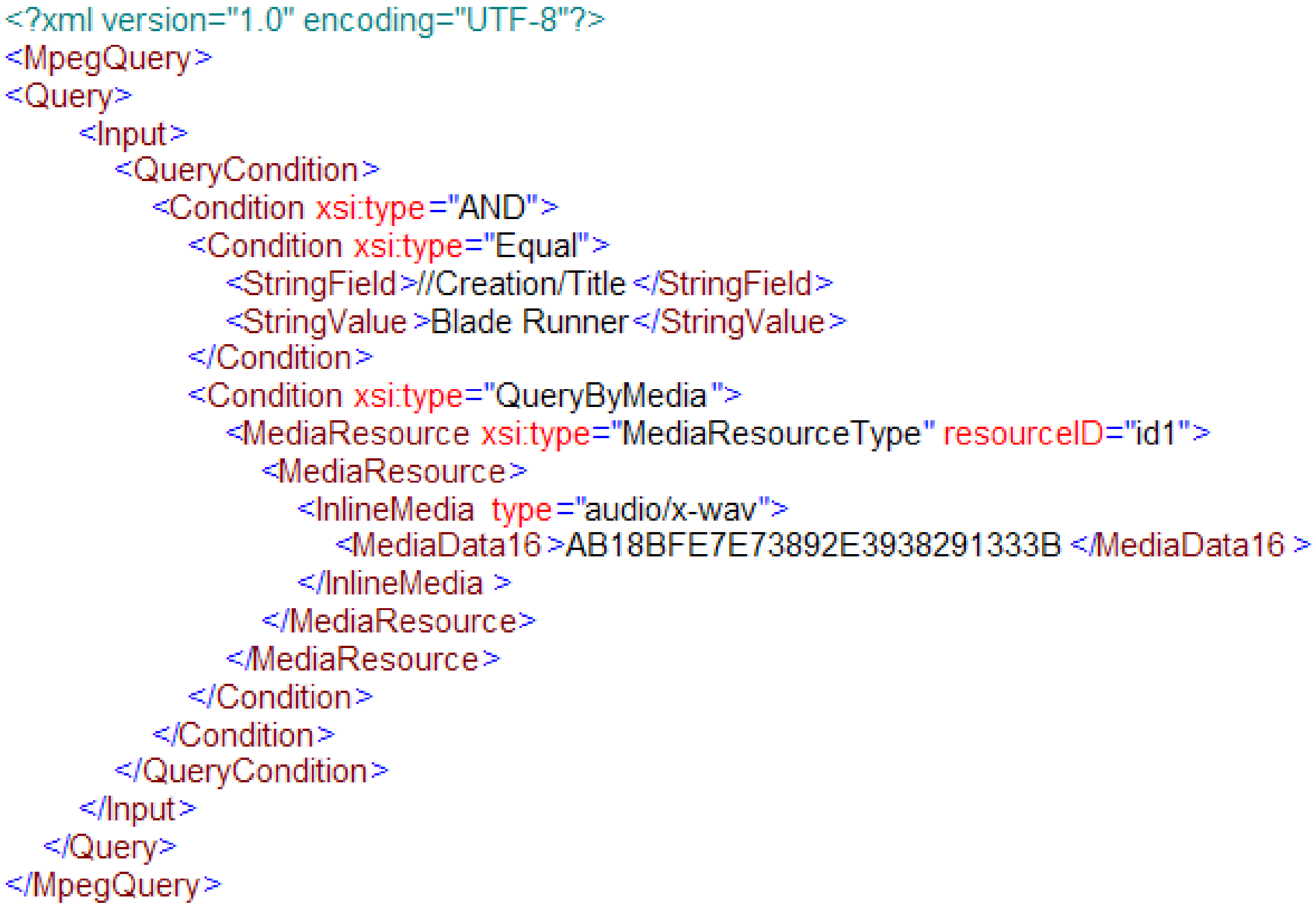
MPEG Query Format (MPQF)

- Example input query condition tree (graphical view)



```
<InlineMedia type=“image/jpeg">
        <MediaData64>R0lGODlhDwAPAKECAAAAzMzM/////
         wAAACwAAAAADwAPAAACIISPeQHsrZ5ModrLlN
         48CXF8m2iQ3YmmKqVlRtW4MLwWACH+H09
         ….
        </MediaData64>
</InlineMedia>
```

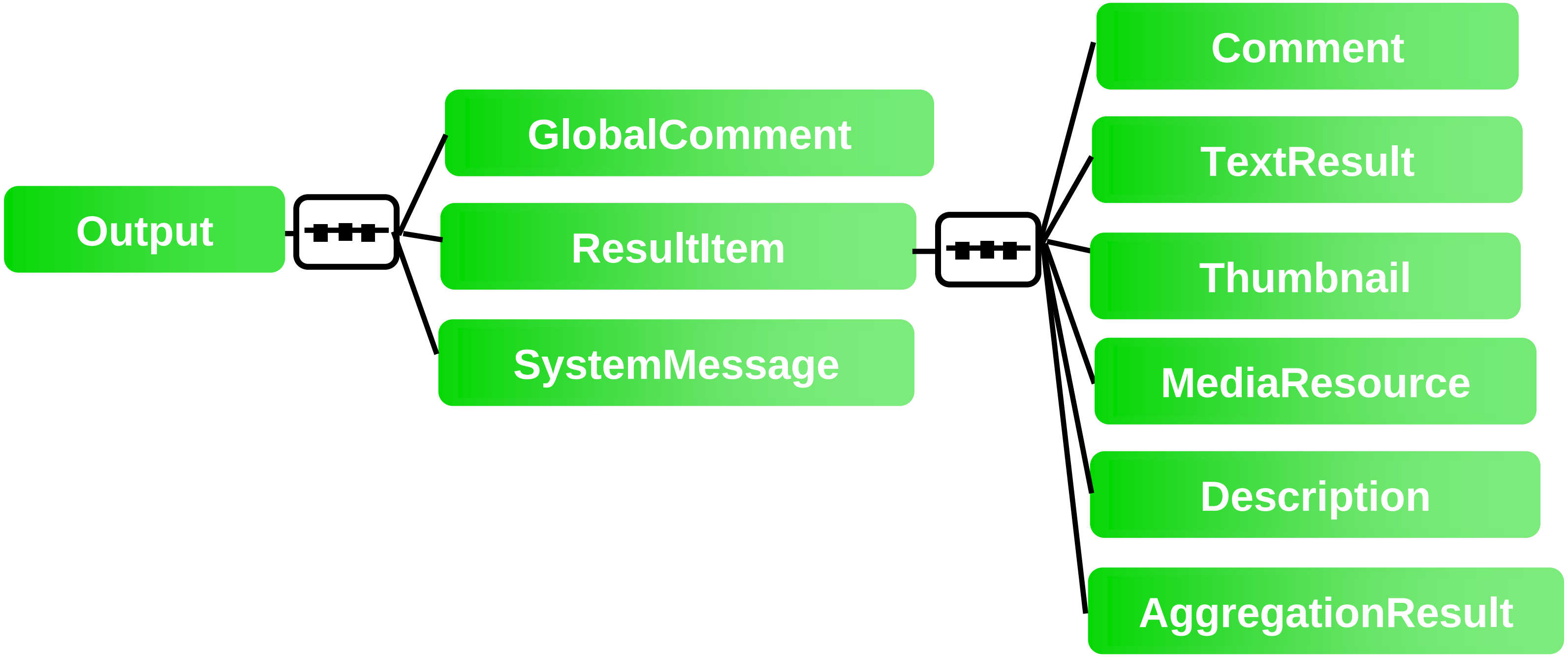
Input Query Format (Example)



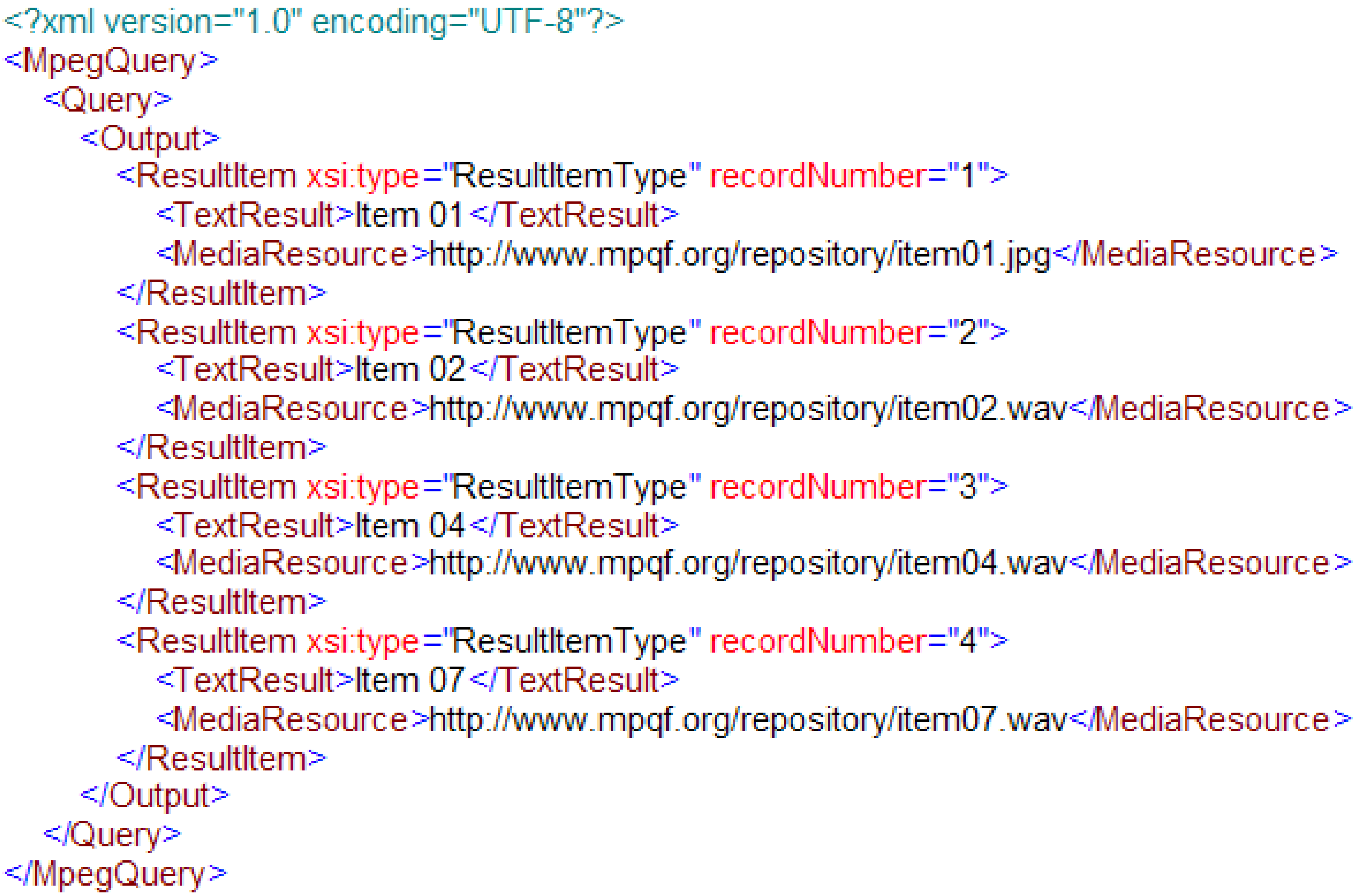
```xml
<?xml version="1.0" encoding="UTF-8"?>
<MpegQuery>
<Query>
    <Input>
      <QueryCondition>
        <Condition xsi:type="AND">
          <Condition xsi:type="Equal">
            <StringField>//Creation/Title </StringField>
            <StringValue >Blade Runner</StringValue>
          </Condition>
          <Condition xsi:type="QueryByMedia ">
            <MediaResource xsi:type="MediaResourceType" resourceID="id1">
              <MediaResource>
                <InlineMedia  type="audio/x-wav">
                  <MediaData16 >AB18BFE7E73892E3938291333B </MediaData16 >
                </InlineMedia >
              </MediaResource>
            </MediaResource>
          </Condition>
        </Condition>
      </QueryCondition>
    </Input>
  </Query>
</MpegQuery>
```

MPEG Query Format (MPQF)

- Output Query Format (OQF)



Output Query Format (Example)



```
<?xml version="1.0" encoding="UTF-8"?>
<MpegQuery>
    <Query>
        <Output>
            <ResultItem xsi:type="ResultItemType" recordNumber="1">
                <TextResult>Item 01</TextResult>
                <MediaResource>http://www.mpqf.org/repository/item01.jpg</MediaResource>
            </ResultItem>
            <ResultItem xsi:type="ResultItemType" recordNumber="2">
                <TextResult>Item 02</TextResult>
                <MediaResource>http://www.mpqf.org/repository/item02.wav</MediaResource>
            </ResultItem>
            <ResultItem xsi:type="ResultItemType" recordNumber="3">
                <TextResult>Item 04</TextResult>
                <MediaResource>http://www.mpqf.org/repository/item04.wav</MediaResource>
            </ResultItem>
            <ResultItem xsi:type="ResultItemType" recordNumber="4">
                <TextResult>Item 07</TextResult>
                <MediaResource>http://www.mpqf.org/repository/item07.wav</MediaResource>
            </ResultItem>
        </Output>
    </Query>
</MpegQuery>
```

MPQF evaluation model

Multimedia Database: Simplistic view

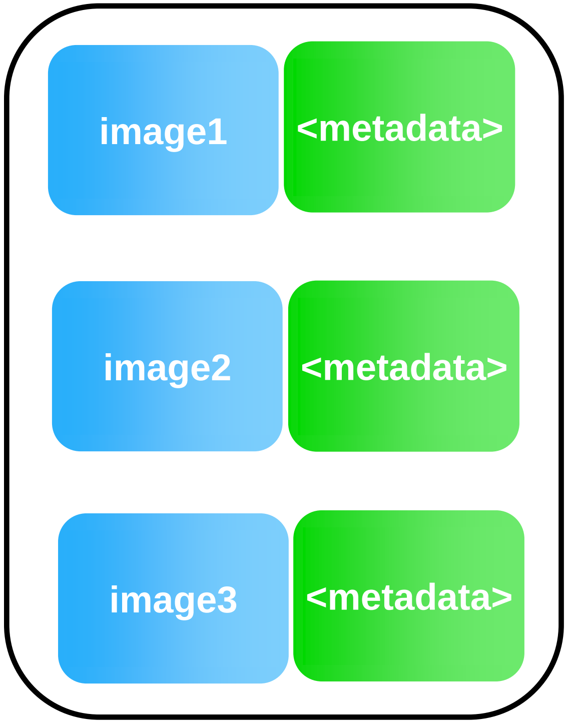
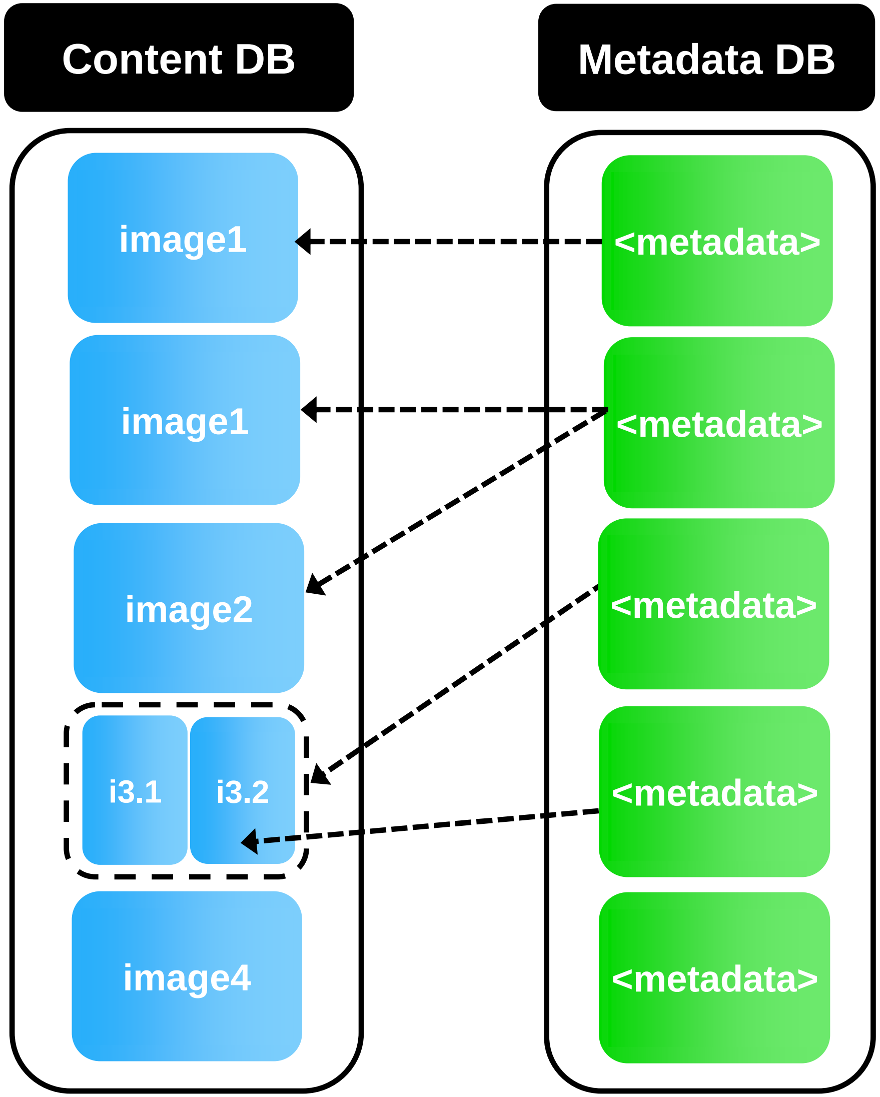


image1 <metadata>

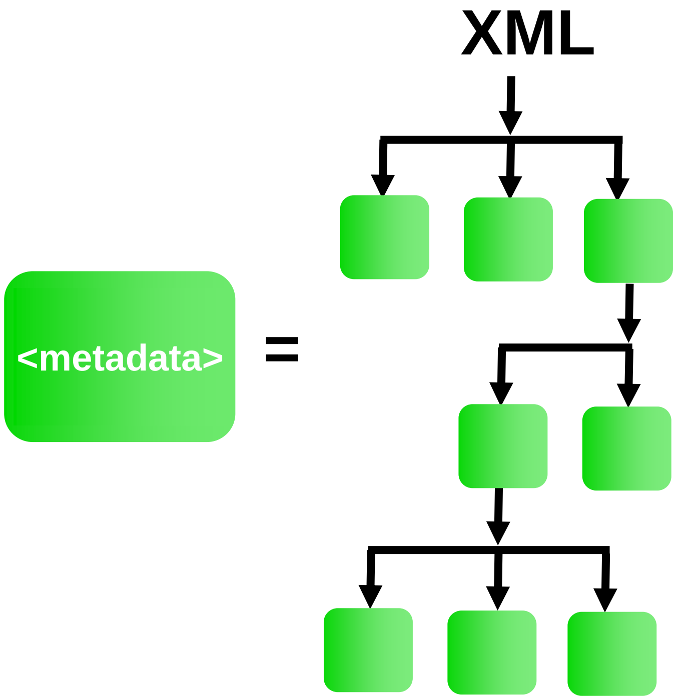
image2 <metadata>

image3 <metadata>

Multimedia Database: Complex view (the one in the Query Format)



The Query Format assumes XML metadata



MPQF in depth

- The **EvaluationPath** element

```
<MpegQuery>
  <Query>
    <Input>
      <QueryCondition>
        <EvaluationPath>//VideoSegment</EvaluationPath>
        <Condition xsi:type="QueryByFreeText">
          <FreeText>Lausanne</FreeText>
        </Condition>
      </QueryCondition>
    </Input>
  </Query>
</MpegQuery>
```

Multimedia Database: Complex view (the one in the Query Format)

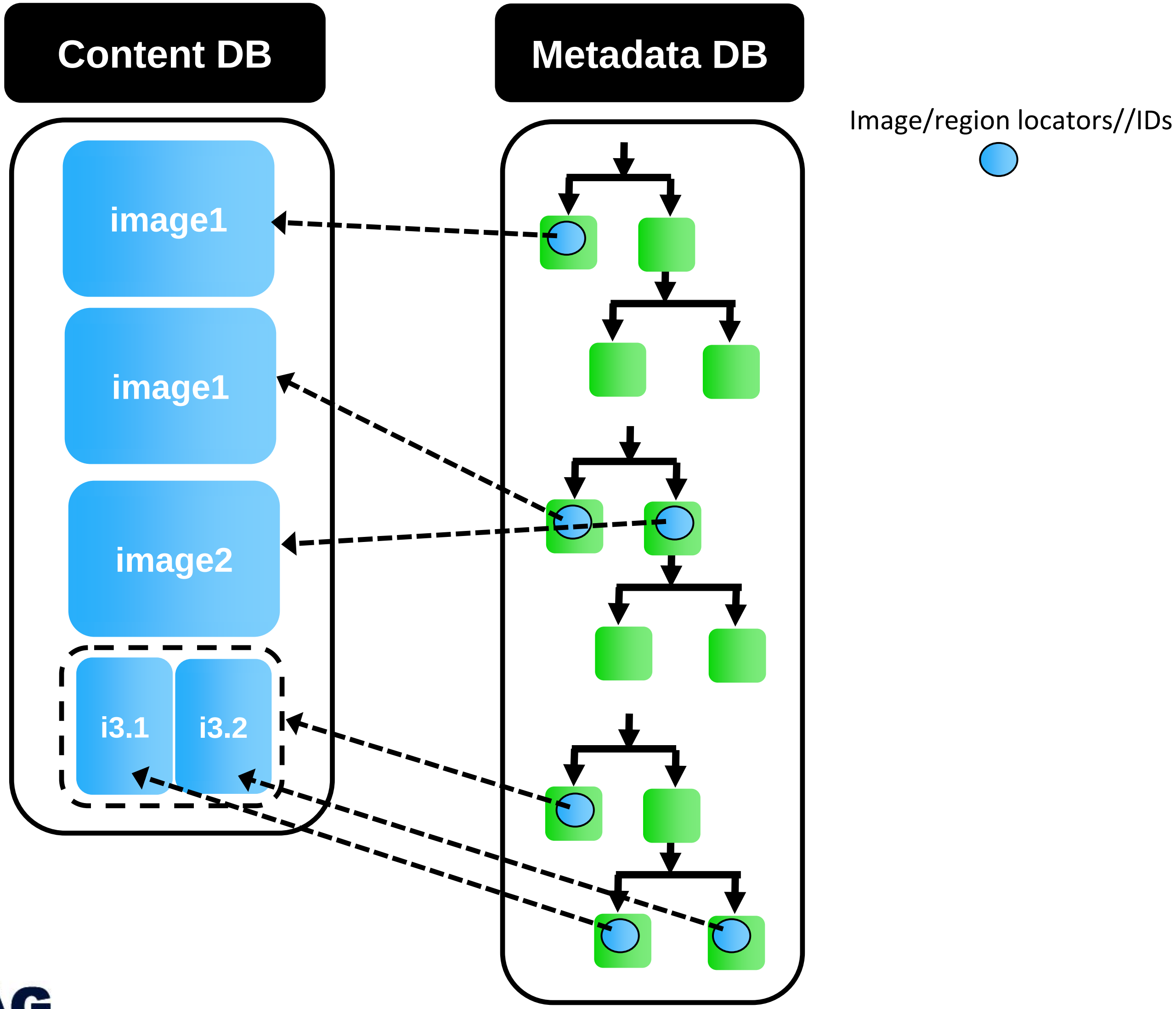
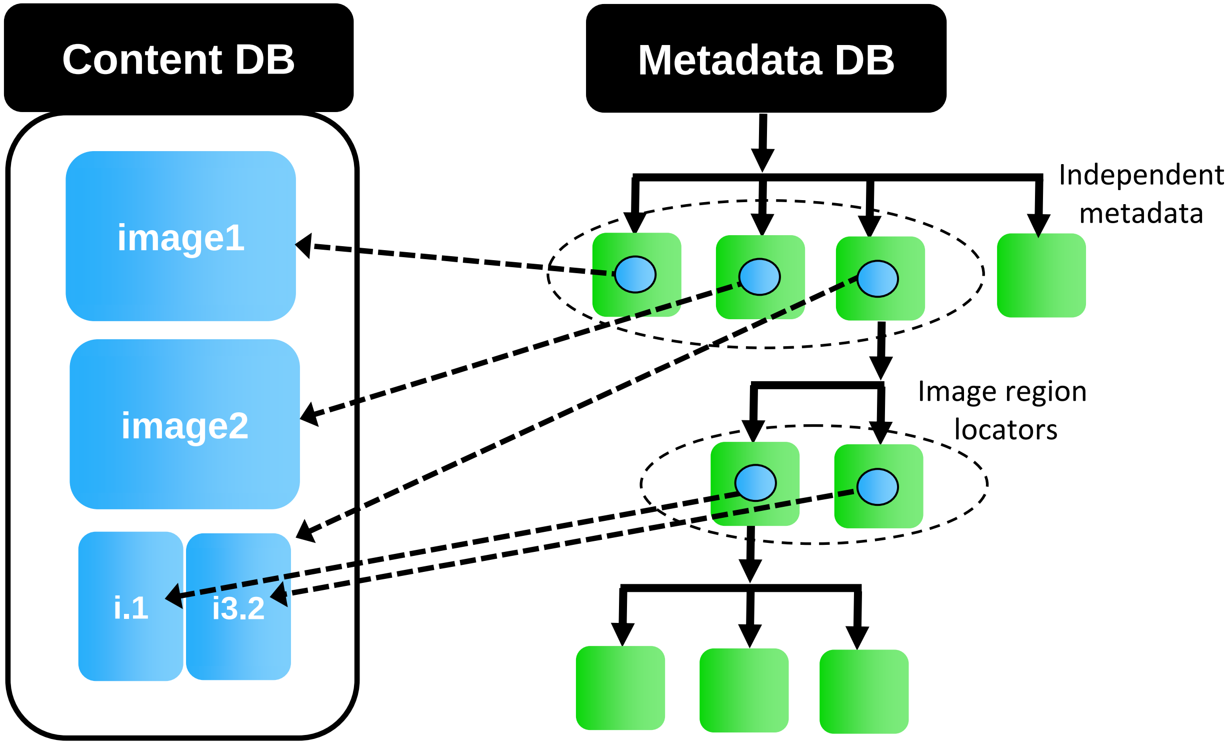
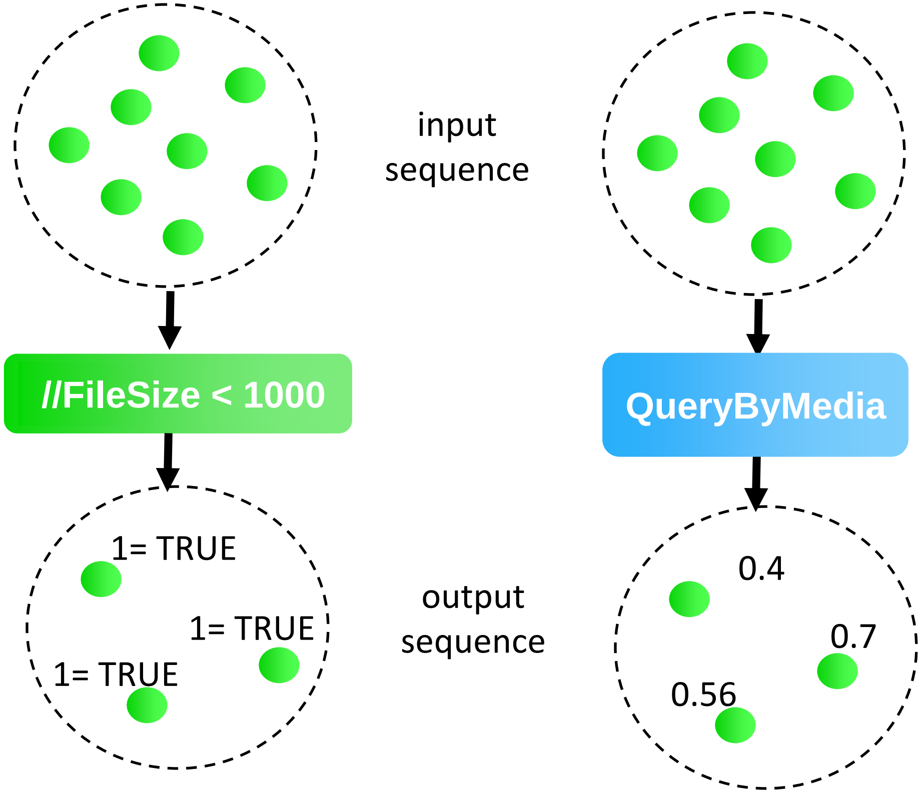


Image Database: Practical view (ONE BIG XML TREE)



MPQF in depth

- Two evaluation styles in MPQF: Boolean and fuzzy-logic

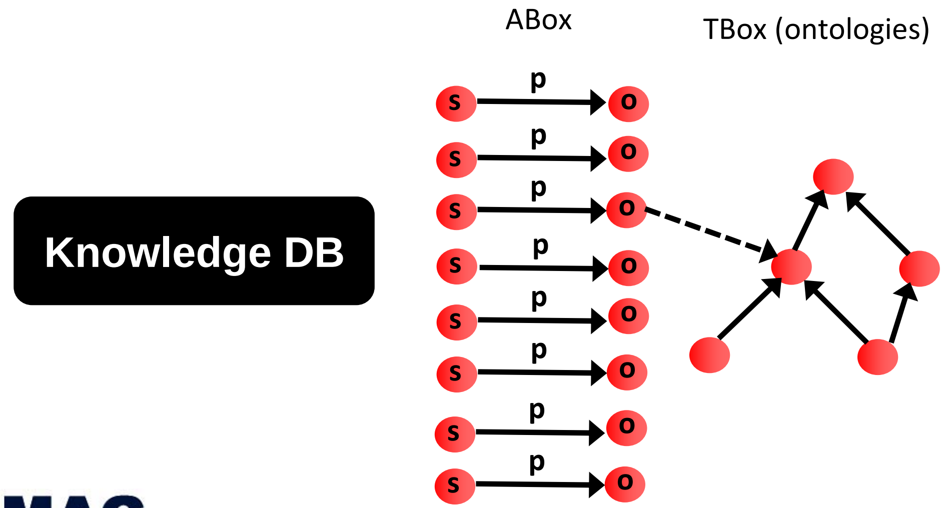


DMAG DISTRIBUTED MULTIMEDIA APPLICATIONS GROUP

How to extend MPQF to allow SPARQL-like queries

How to extend MPQF to allow SPARQL-like queries

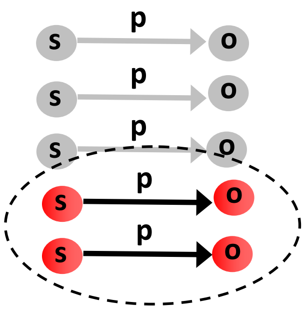
- Knowledge base (KB) = Tbox (terminological component) +Abox (assertion component)
- Some relevant initiatives are choosing the RDF language for modeling metadata metadata
- Typical Semantic Web based multimedia KB (e.g. PhotoRDF):
  - Tbox = RDFS/OWL ontologies
  - Abox = RDF statements about the media objects



How to extend MPQF to allow SPARQL-like queries

- SPARQL. W3C Recommendation on 15th January 2008
- An SPARQL query consists of a basic graph pattern, expressed as a list of triple patterns

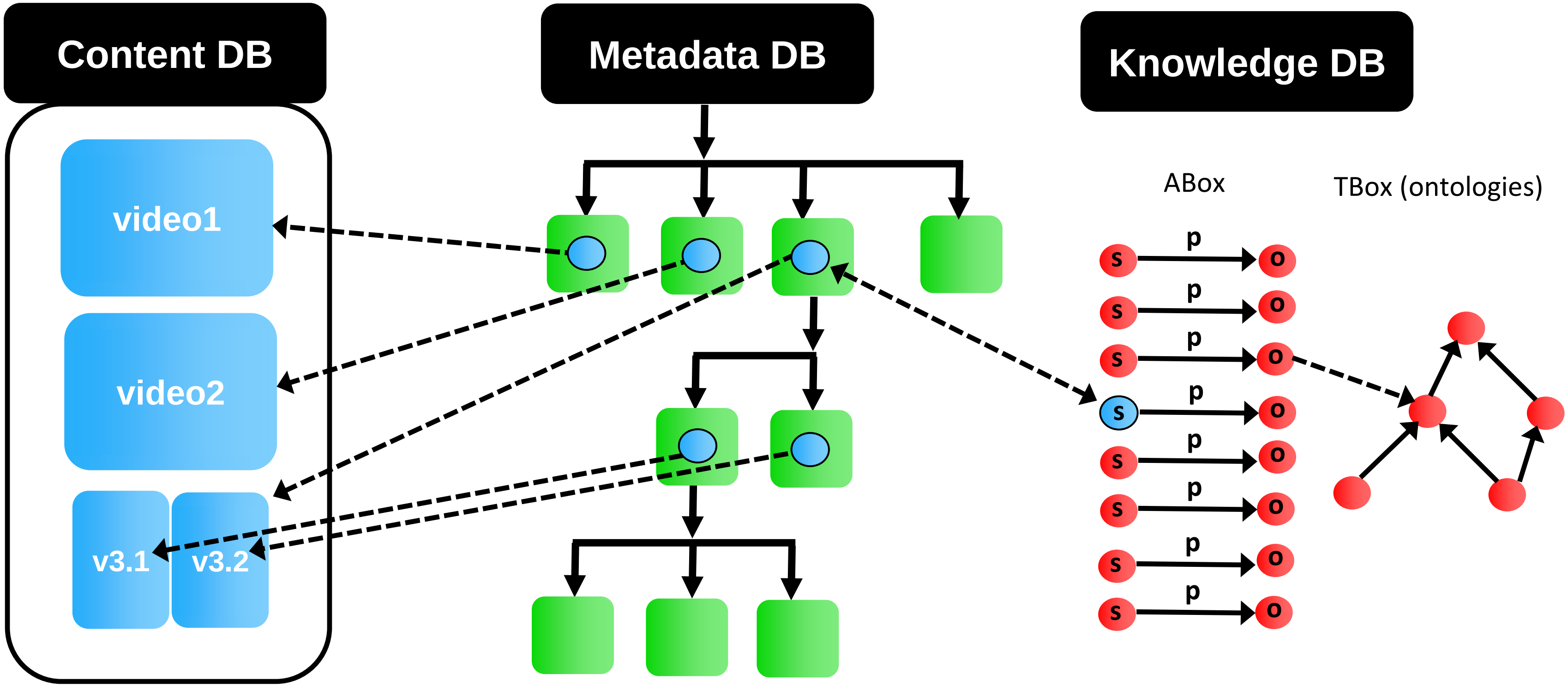
```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
SELECT ?title ?price
WHERE { ?x ns:price ?price .
FILTER (?price < 30.5)
?x dc:title ?title . }
```



DMAG
DISTRIBUTED MULTIMEDIA APPLICATIONS GROUP

How to extend MPQF to allow SPARQL-like queries

- Proposal 1/3: Extend the MPQF data model



How to extend MPQF to allow SPARQL-like queries

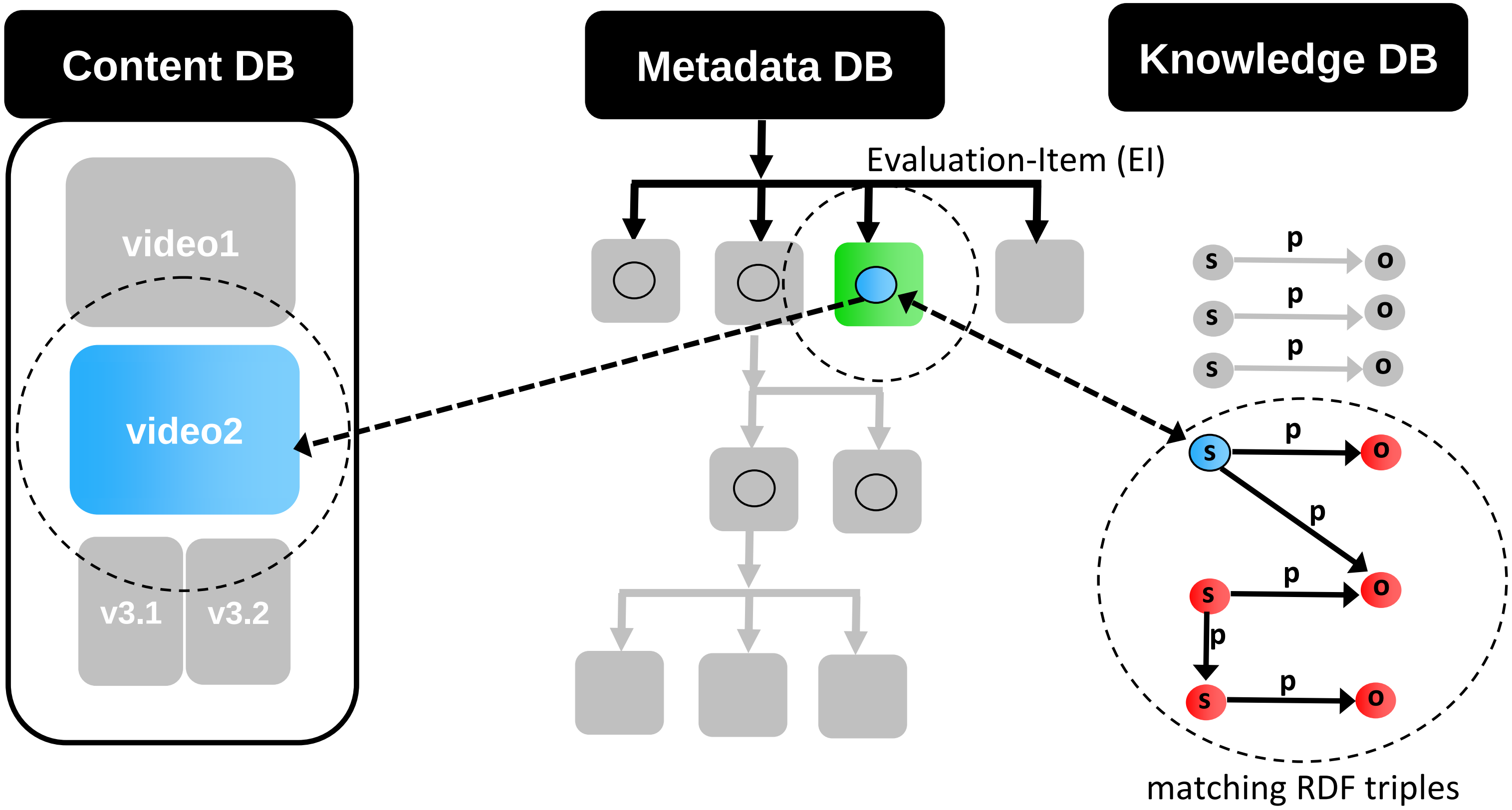
- Proposal 2/3: Include a new query type: **QueryBySPARQL**

```
<MpegQuery mpqfID="someID">
 <Query>
  <Input>
   <QueryCondition>
    <Condition xsi:type="QueryBySPARQL">
     <SPARQL>
      <![CDATA[
       PREFIX dc: <http:///>
       ASK { ?resource dc:title "Barcelona . }
      ]]>
     </SPARQL>
    </Condition>
 </QueryCondition>
 </Input>
 </Query>
</MpegQuery>
```

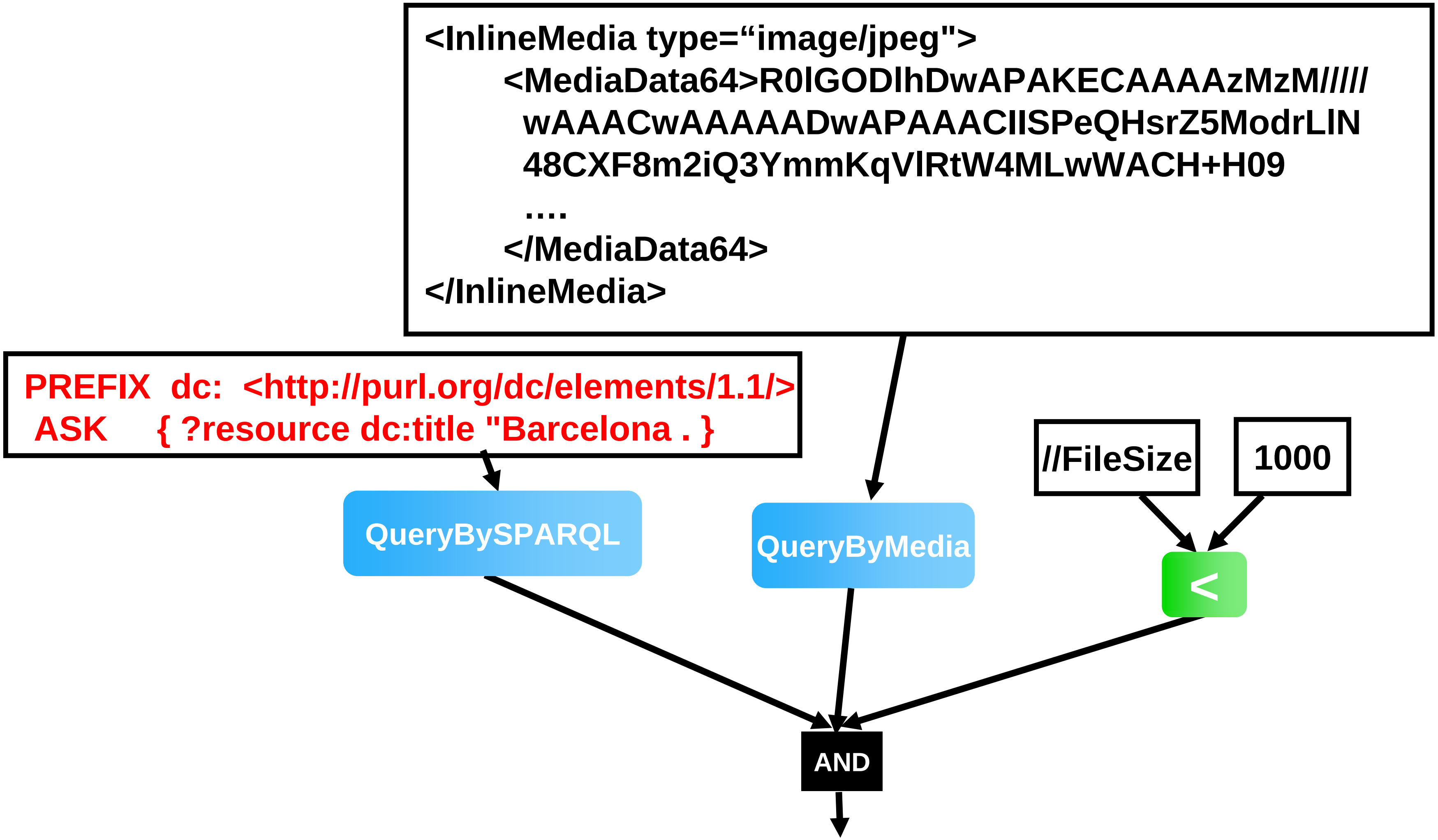
DMAG
DISTRIBUTED MULTIMEDIA APPLICATIONS GROUP

How to extend MPQF to allow SPARQL-like queries

- Visual representation of the execution of the new *QueryBySPARQL* query type



DMAG DISTRIBUTED MULTIMEDIA APPLICATIONS GROUP



```
<InlineMedia type="“image/jpeg">
        <MediaData64>R0lGODlhDwAPAKECAAAAzMzM/////
          wAAACwAAAAADwAPAAACIISPeQHsrZ5ModrLlN
          48CXF8m2iQ3YmmKqVlRtW4MLwWACH+H09
          ....
        </MediaData64>
</InlineMedia>
```

```
PREFIX  dc:  <http://purl.org/dc/elements/1.1/>
 ASK     { ?resource dc:title "Barcelona . }
```

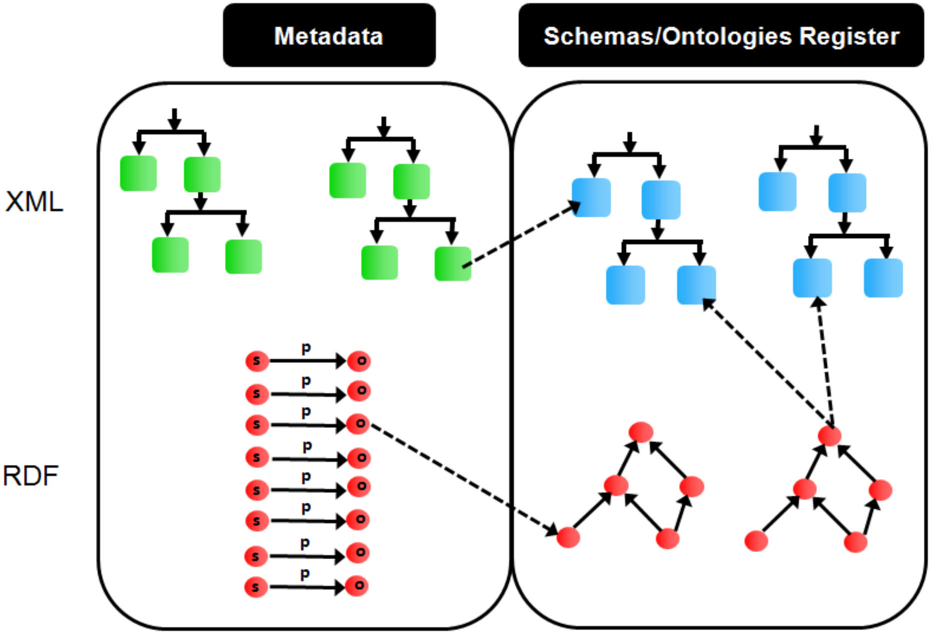
But, what about metadata interoperability with MPQF?

MPQF and metadata interoperability

- Currently MPQF is **metadata-neutral**
  - Queries refer to paths related to different metadata formats
- Several metadata formats exist: Dublin Core, EXIF, ID3, MPEG-7, etc.
- Semantic mappings can help circumventing the problem
- The mappings are (supposed to be) easiest to specify with ontologies
- (with our proposal) we can use RDF and SPARQL in MPQF but…
- Can I use “ontologies” with MPQF?
  - (currently) no, at least not in a “standard” way

MPQF and metadata interoperability

- Proposal 3/3: Extend MPQF:
  - Allow registering RDFS/OWL ontologies
  - Be aware of the registered ontologies when querying XML
  - Be aware of the registered ontologies when querying RDF
  - Be aware of the registered ontologies when querying free text?



MPQF and metadata interoperability

- Two choices to achieve interoperability with N formats:
  - Specify NxN mappings
  - Specify one reference metadata format and N mappings (better!)
- Which kind of reference metadata format?
  - An XML Schema (e.g. XMP)
  - But mappings are simpler if the reference metadata model is also an ontology
  - This issue is now being discussed at JPEG (JPSearch initiative)
  - OPEN ISSUE

Conclusions

- The MPEG Query Format is finished!
  - But now is time for amendments and corrigenda
- The implementation race has begun
- MPQF lacks the ability of handling RDF and SPARQL
  - Proposal 1/3: Extend the logical model
  - Proposal 2/3: New query type: *QueryBySPARQL*
  - Proposal 3/3: Allow registering ontologies

- A relevant piece is still missing: **A pivot metadata format**

Acknowledgments

Thank you