

Adaptation logic for (server-side) media fragment extraction

Davy Van Deursen

W3C MFWG F2F meeting @ Barcelona

In this presentation

- Adaptation engine for extracting media fragments
 - no transcoding techniques are used
 - only high-level adaptation operations
- Presented approach is not an implementation of the Media Fragments URI scheme
 - no protocol implementation
 - but can be used to assist for example an HTTP Web server implementing Media Fragments

Format-independent content adaptation: principles

<bitstream

xml:base="myPrecious_30hz.264">

<sps>0-15</sps>

<pps>16-24</pps>

<I_picture>25-2637</I_picture>

~~<B_picture>2638-2746</B_picture>~~

~~<B_picture>2747-2903</B_picture>~~

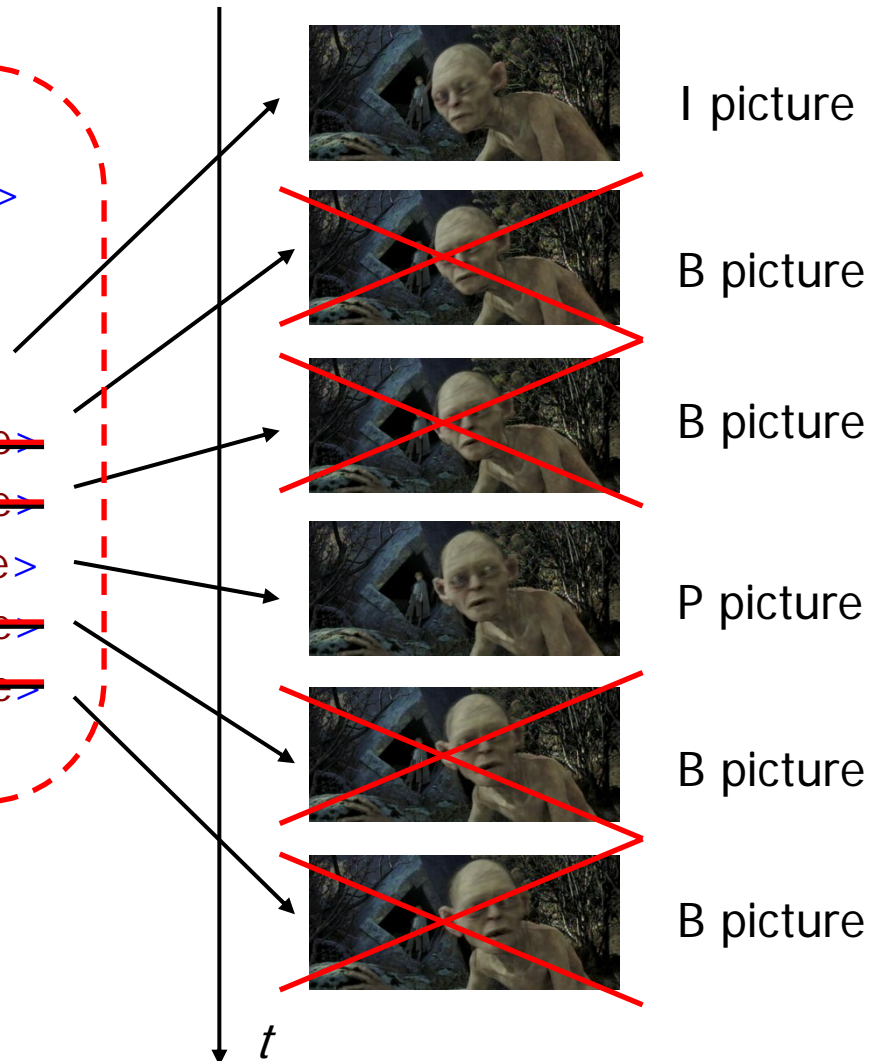
<P_picture>2903-3857</P_picture>

~~<B_picture>3857-3972</B_picture>~~

~~<B_picture>3973-4103</B_picture>~~

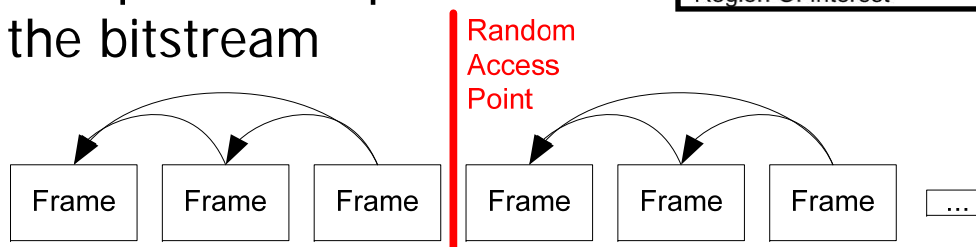
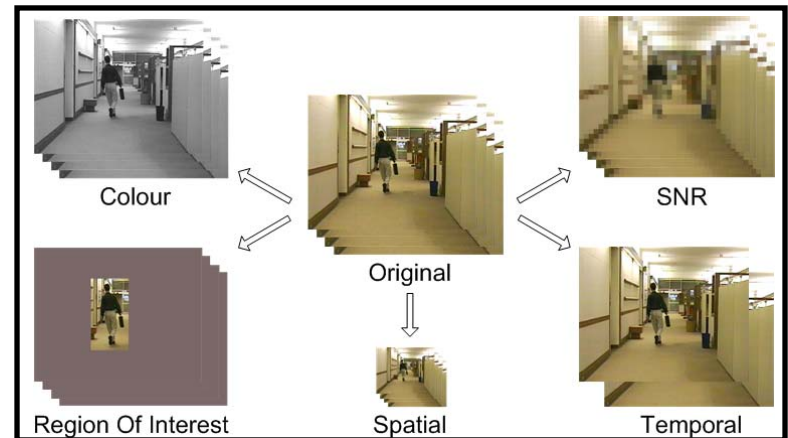
</bitstream>

Bitstream Syntax Description
(BSD)



Limitations

- only high-level adaptation operations (i.e., no transcoding)
 - removal of data blocks
 - modification of high-level syntax elements
- What is possible?
 - exploitation of scalability
 - demultiplexing
 - temporal selection
 - on condition that random access points are present in the bitstream



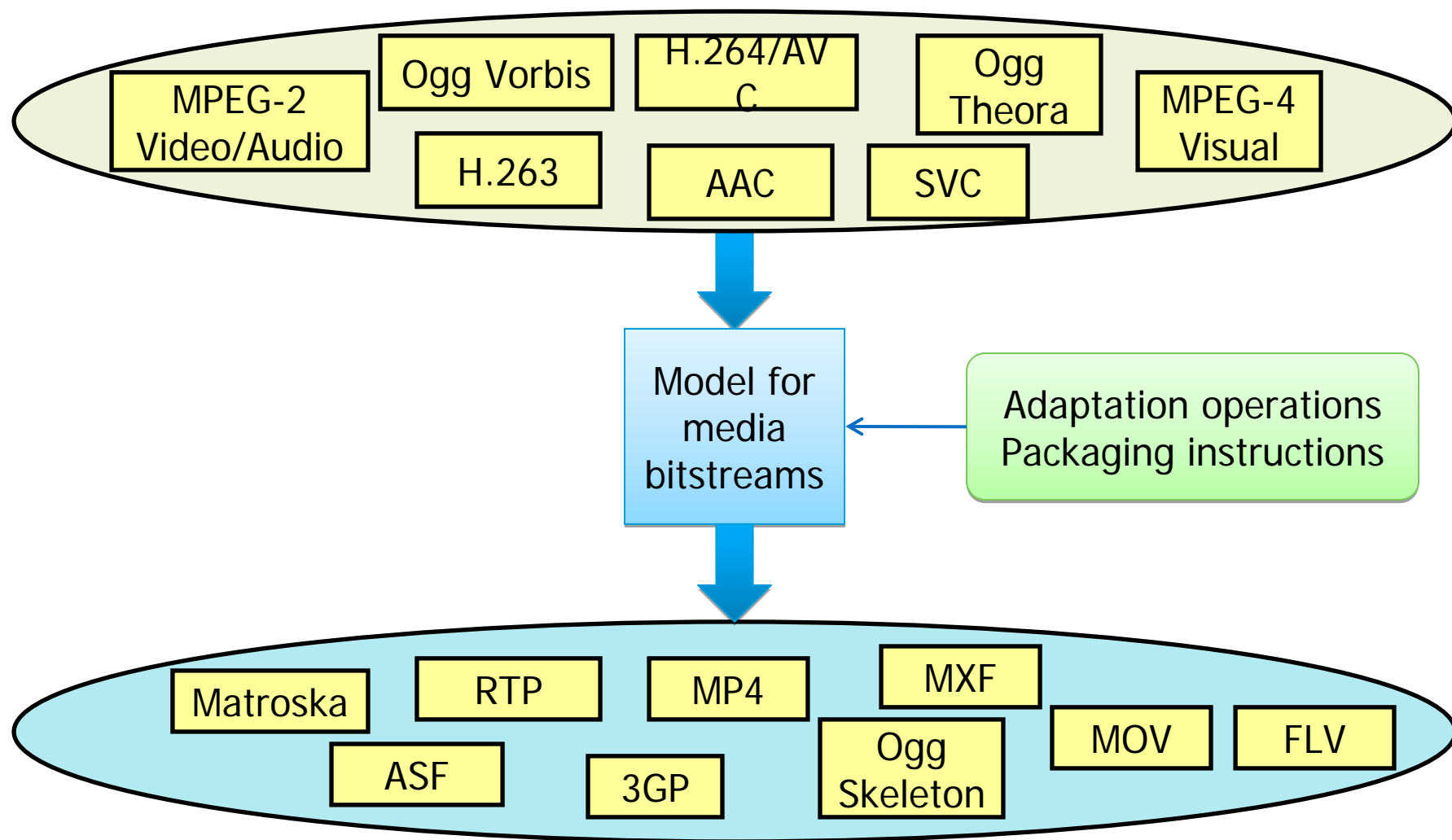
(XML-based) BSDs

- Benefits
 - enables the use of adaptation software that is independent of the underlying media format
 - easy extensible
 - link with other metadata
 - e.g., description of the media content
 - existing XML tools can be used for BSD manipulation
 - e.g., XSLT or STX
- Two technologies were standardized within MPEG-21 DIA
 - MPEG-B BSDL & MPEG-21 gBS Schema
- Related technologies
 - XFlavor, (g)BFlavor, Preon, ...

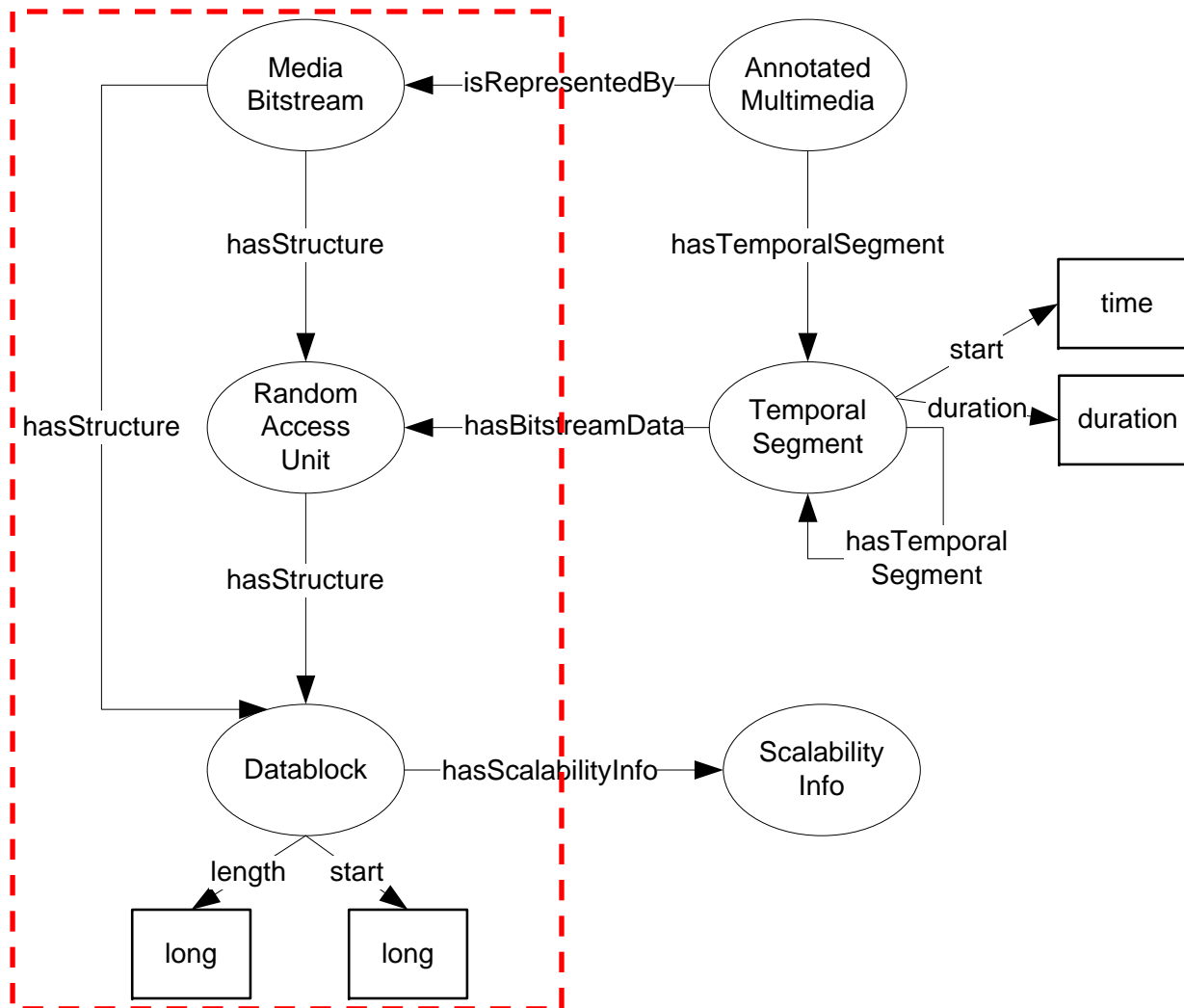
RDF-driven content adaptation & delivery

- Developed during my PhD
- Inspired on principles of BSD-driven content adaptation
- Features
 - format-independent adaptation and packaging of media bitstreams
 - abstracted adaptation operations
 - integration with the Semantic Web
 - track & temporal media fragment selection
- Based on a model for media bitstreams
 - implemented in OWL

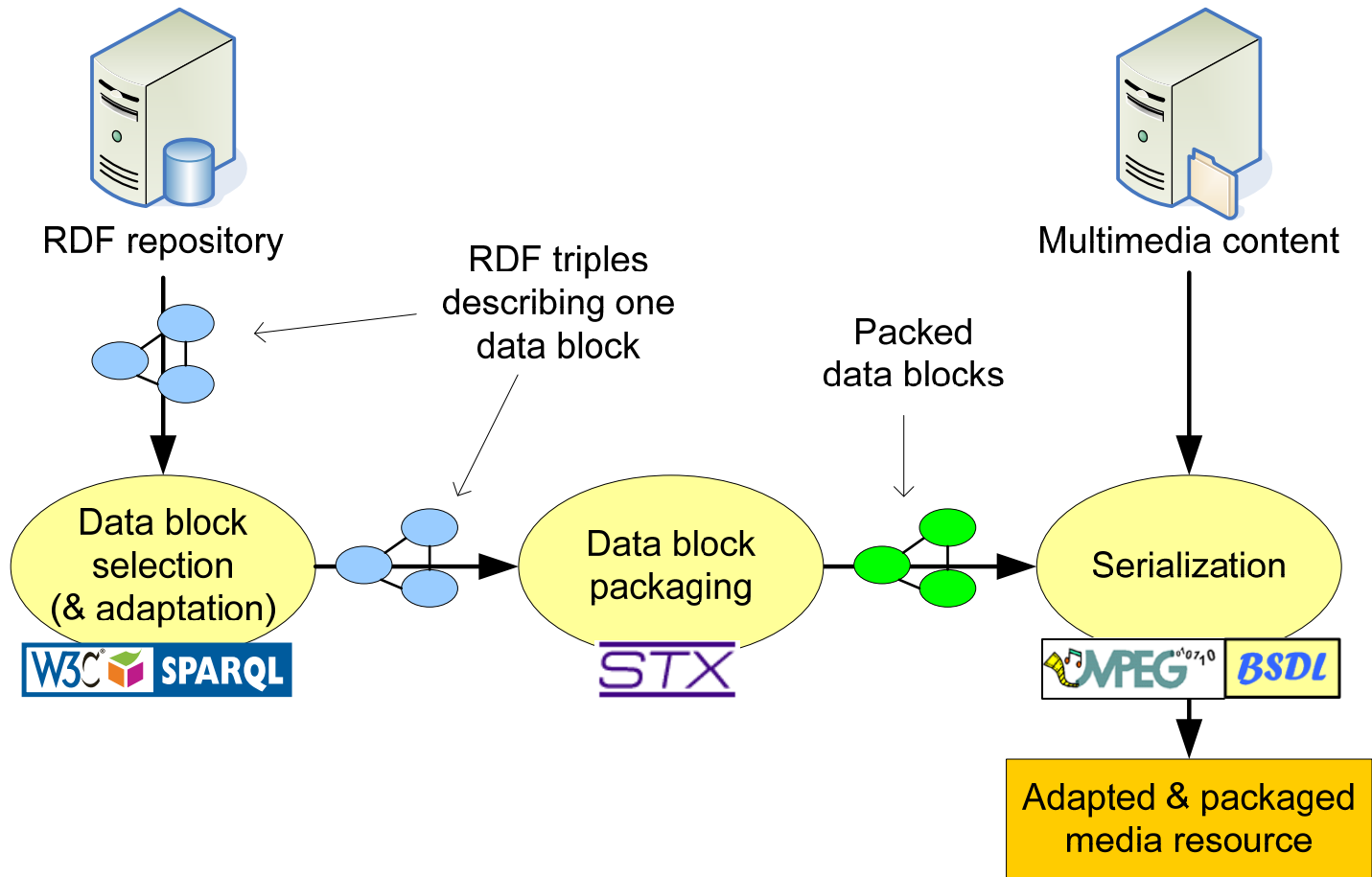
Model for media bitstreams (1/2)



Model for media bitstreams (2/2)

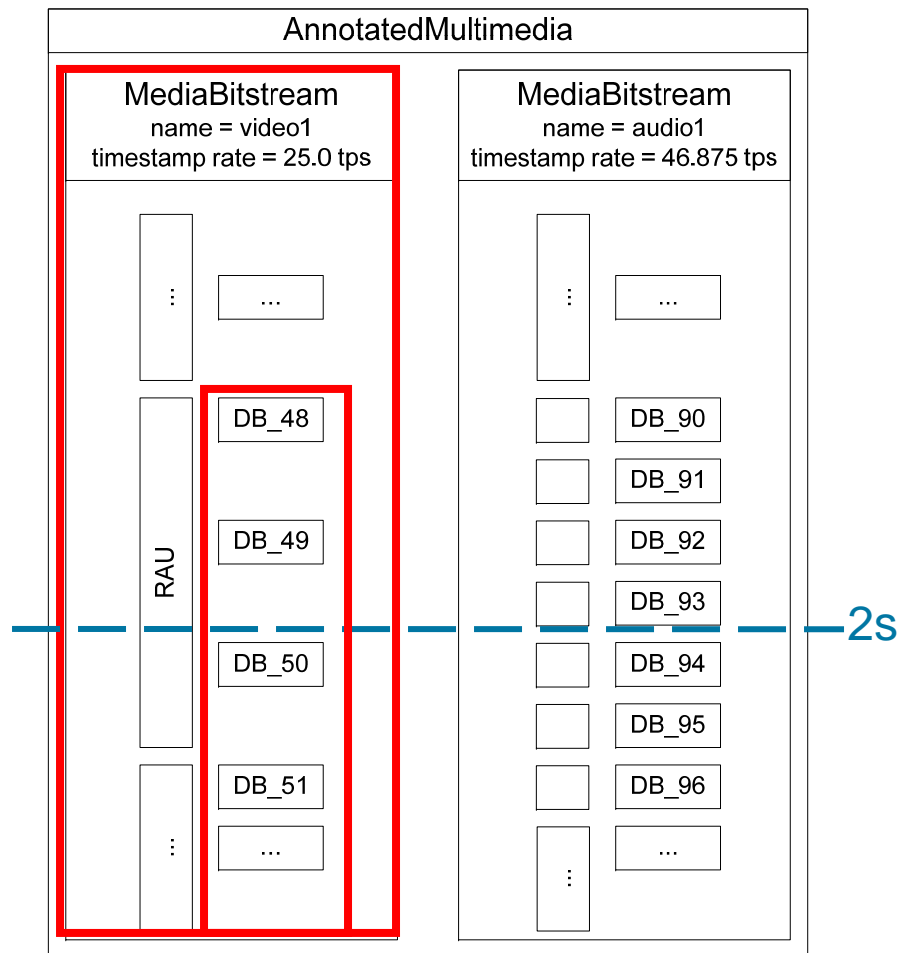


Adaptation & delivery workflow



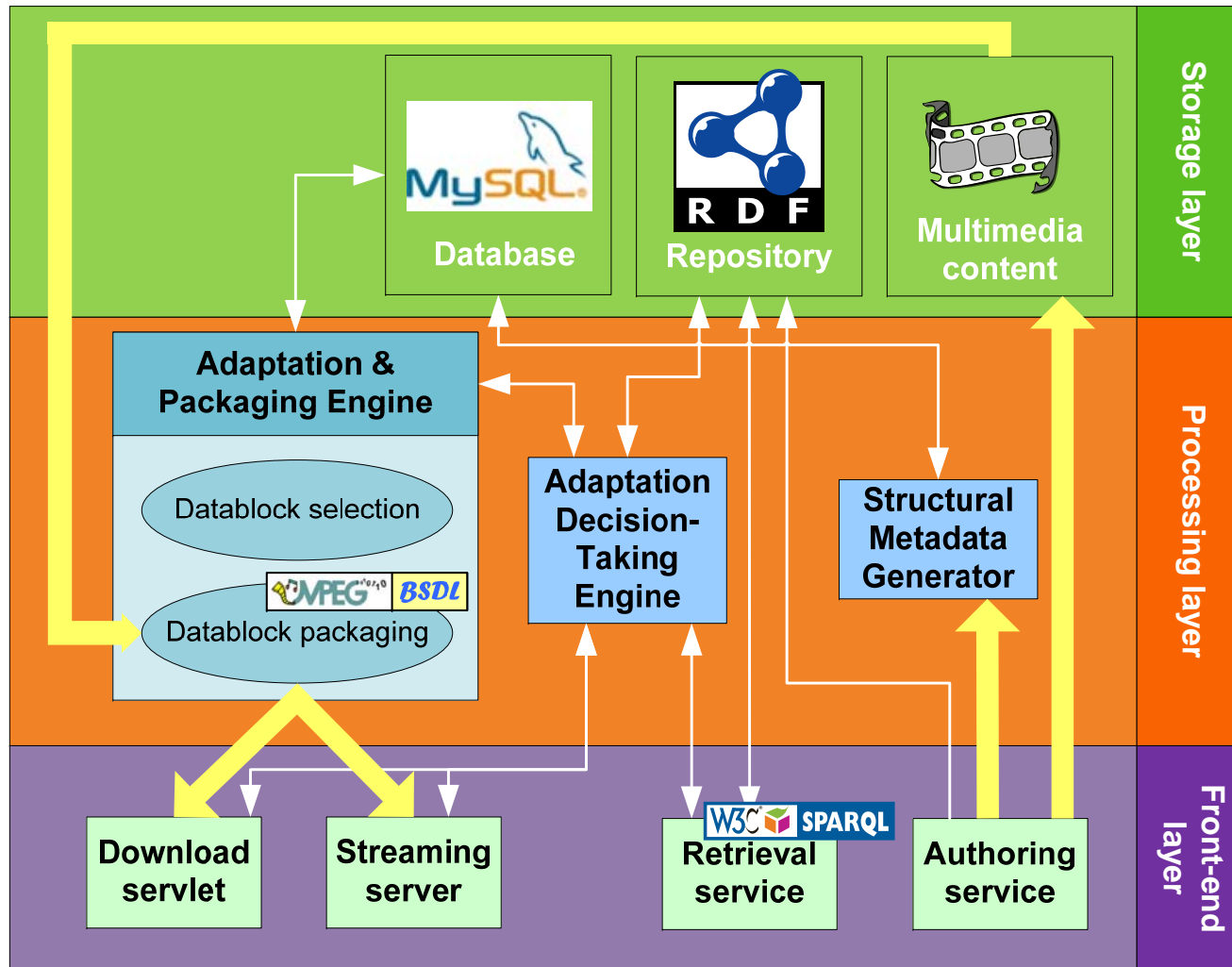
RDF-driven Media Fragment Extraction

`http://foo.com/media.mp4#track='video1'&t=2,10`



- Track selection
 - selection of *MediaBitstreams*
- Temporal selection
 - selection of *DataBlocks*
 - based on their timestamps

NinSuna: implementing RDF-driven adaptation



Remarks

- Media resources need to be ‘ingested’, before we can adapt and deliver them
 - generation of structural metadata
- `media.mp4#track='video1'&t=2,10`
 - name of AnnotatedMultimedia is ‘media’
 - delivery format is MP4
 - track & temporal selections need to be executed