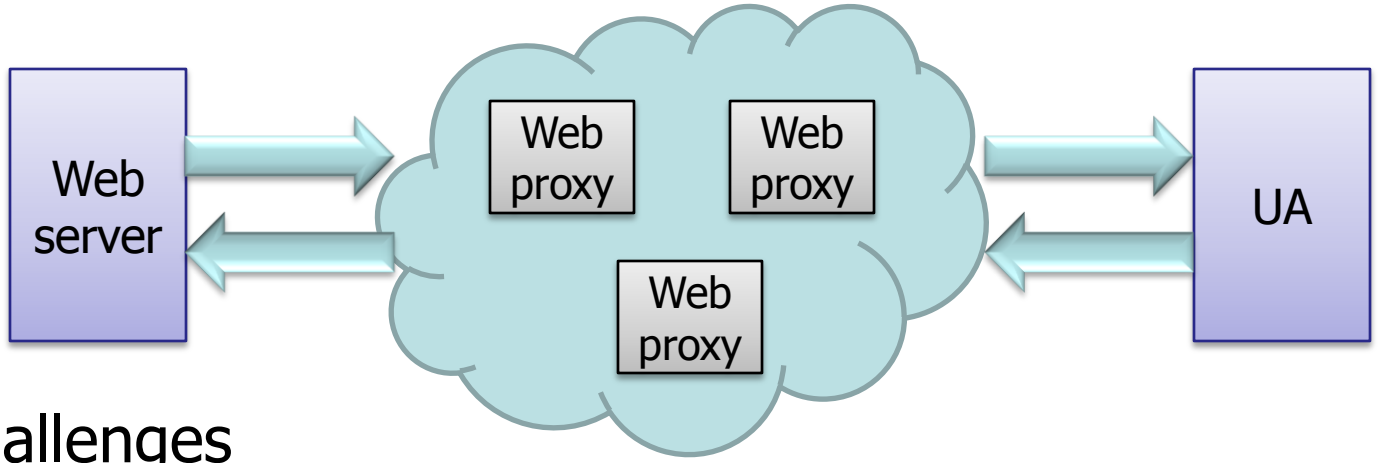
Media Delivery UC

Davy Van Deursen

Delivery of media resources



- Challenges
  - keep Web proxies independent of media fragments
  - keep, to a certain extent, the UA independent of container/coding format
  - Web server needs to be able to handle all container/coding formats (?)
    - could announce which formats are supported

Client- vs. server-side processing

- Client-side
  - e.g., use of fragments (#)
  - full media resource is retrieved, after which the fragments are interpreted by the UA
  - could avoid too many requests to the server
- Server-side
  - e.g., use of query (?) or semi-colon (;)
  - URI is interpreted by the server, only the requested media fragments are sent to the UA
  - limits bandwidth usage and processing at UA
- Combination of client- and server-side processing

Existing technologies (1/2)

- MPEG-21 Part 17: Fragment Identification of MPEG Resources
  - normative syntax for URI fragment identifiers to be used for addressing parts of *MPEG resources*
  - use of fragments (#), no delivery protocol specified
  - trivial scenario: client-side processing
  - other possibility (ISO Base Media Format only)
    - see http://lists.w3.org/Archives/Public/public-media-fragment/2008Oct/0028.html
    - UA gets first N bytes representing headers with timing and byte offset information of the media resource
    - UA makes the translation between time and bytes and requests these bytes

Existing technologies (2/2)

- Temporal URI
  - specifies a syntax for addressing time intervals within time-based Web resources
  - support for client- (#) and server-side (?) processing
  - provides an implementation with HTTP
    - 3 new HTTP headers are specified to support HTTP proxy servers and the identification of enabled HTTP servers
  - proposal for an RTSP implementation is also available
- Any other relevant technologies?

Media fragment delivery protocol (1/2)

- HTTP
  - four-way handshake is needed to make media fragment resources cacheble on the Web
  - see also: http://lists.w3.org/Archives/Public/public-media-fragment/2008Oct/0060.html
    1. UA requests the desired fragments
    2. Web server determines which byte ranges correspond to the requested fragments and returns them to the UA
    3. UA makes use of HTTP byte ranges to get the proper content
    4. Web server sends the requested content
  - Temporal URI makes use of the four-way handshake

Media fragment delivery protocol (2/2)

- RTSP
  - suppose we need to support RTSP caches, could we apply the same four-way handshake?
  - specification provides a Range header
    - similar to the HTTP byte range mechanism
    - however, byte ranges are not used within RTSP
      - time is used to specify the range (npt, smpte, clock)
  - result
    - temporal fragments could be supported by a two-way handshake (use Range header)
    - spatial fragments are not supported!