10 Multimedia Content Management and Distribution

Literature:

Gregory C. Demetriades: Streaming Media, Wiley 2003

Tobias Künkel: Streaming Media – Technologien, Standards, Anwendungen, Addison-Wesley 2001

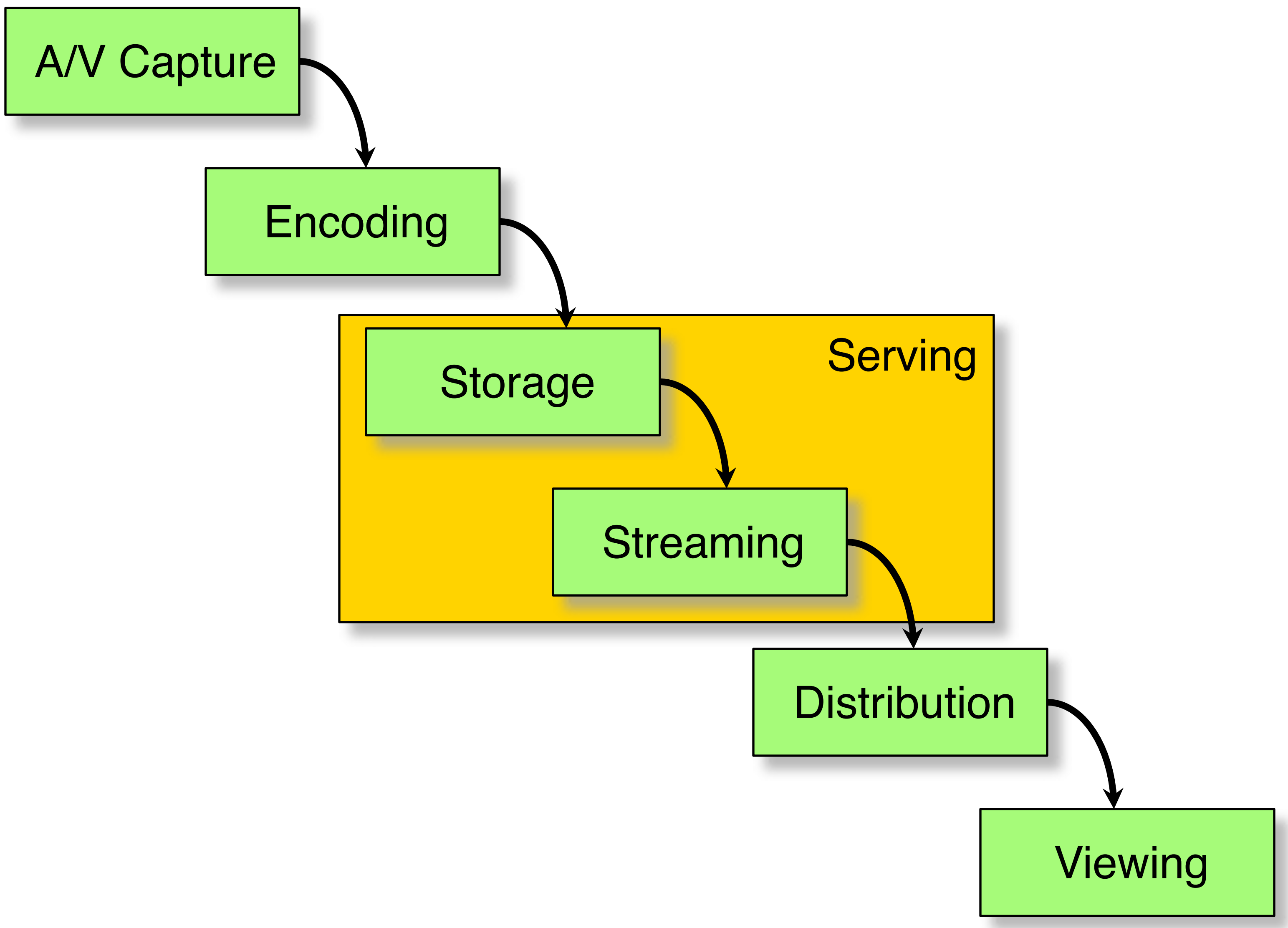
Troncy/Huet/Schenk, Multimedia Semantics - Metadata, Analysis and Interaction, Wiley 2011

Multimedia Delivery, Streaming

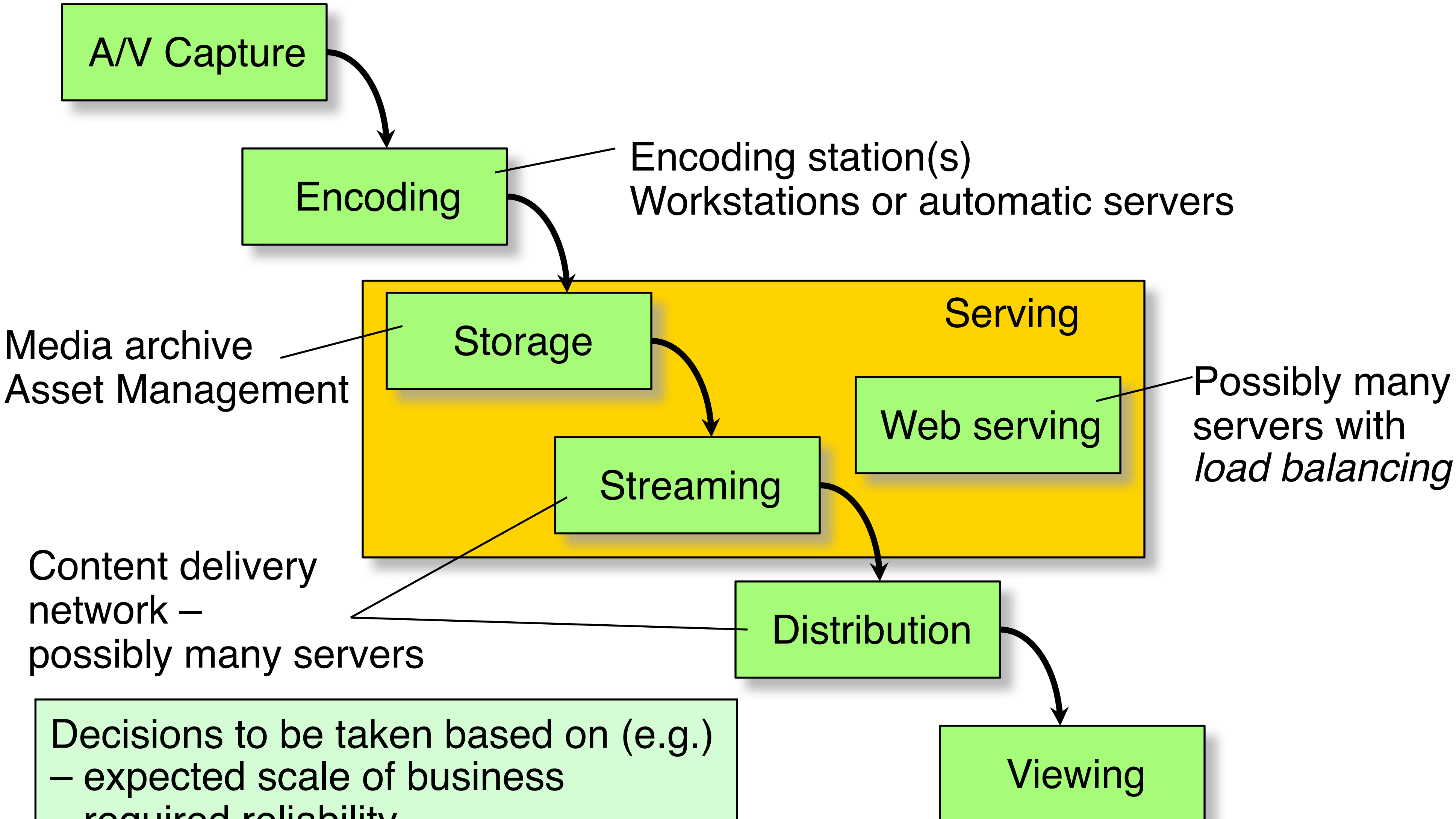
- Delivery types for audio and video content:
  - *Download and Play:*
    Content downloaded completely, then played back
  - *Streaming Media:*
    Playback starts before content is downloaded
    - *Progressive Download:*
      Playback is started while download is still in progress.
      Download rate independent of program bit rate.
    - *True Streaming:*
      Almost “real-time”: Playback rate roughly the same as data delivery rate;
      Small delay between send and receive event of data packet
- Subtypes of True Streaming:
  - *Static File Streaming:*
    Delivery of pre-recorded media files.
    Often also called *on-demand* delivery (e.g. *Video on Demand)*
  - *Live Streaming*

Based on material from www.streamingalliance.org

Streaming Delivery Chain for Audiovisual Media



Hardware in the Streaming Delivery Chain



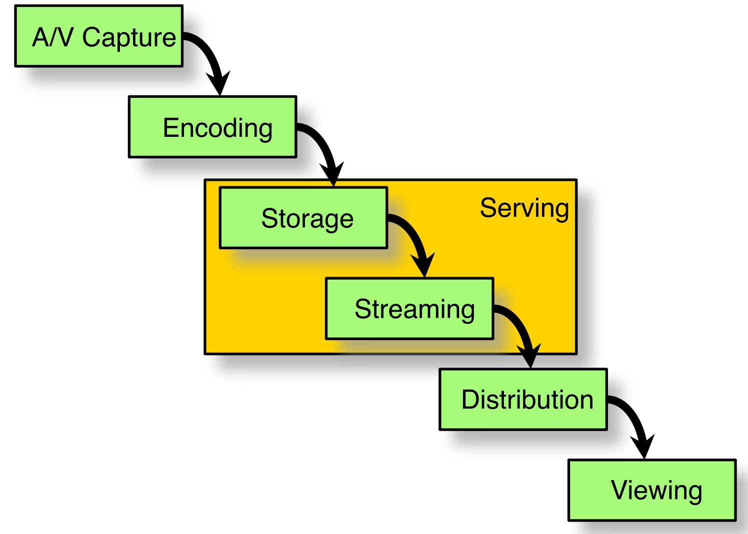
Decisions to be taken based on (e.g.)
– expected scale of business
– required reliability

Digital Asset Management

www.celum.com

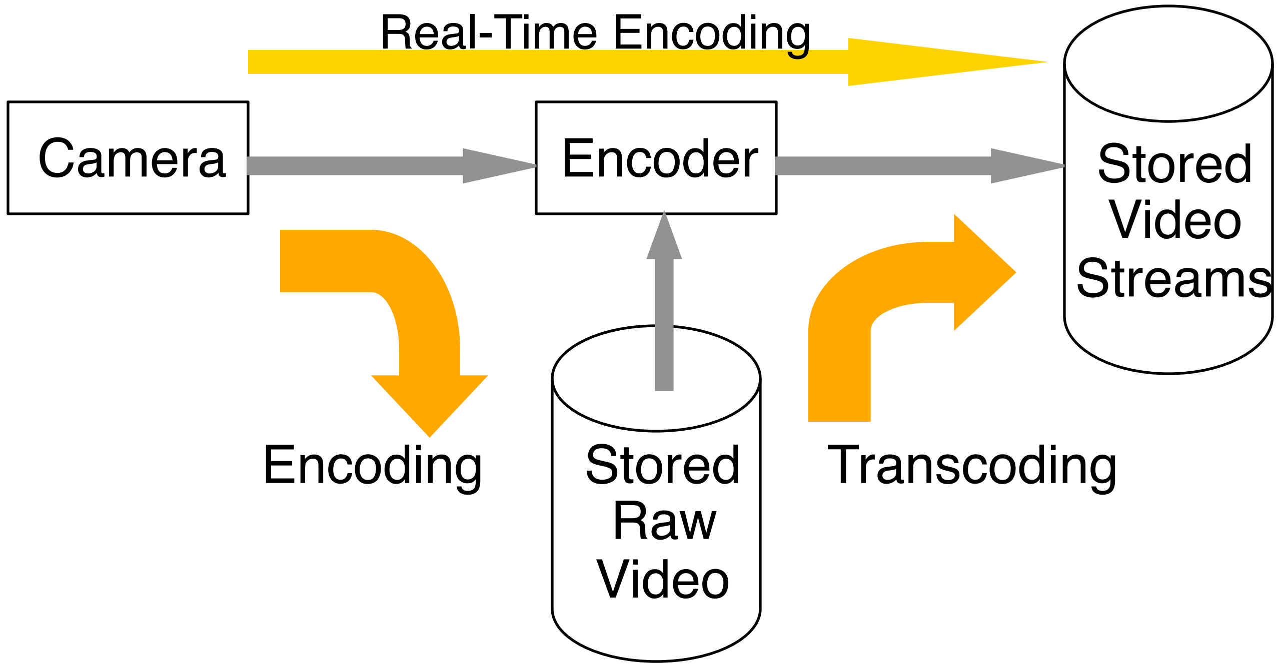
- Very similar acronyms:
  - Digital Asset Management DAM
  - Media Asset Management MAM
  - Rich Media Asset Management RMAM
  - Digital Media Management DMM
- Basic idea:
  - To make the right media material *(media assets)* available for each specific use, in the right version and the right format
- Integration technology:
  - Workflow integration
  - Integration with various media processing tools
  - Integration with content management and syndication solutions

Encoding



- Format conversions
  - E.g. analog/digital conversion
  - E.g. downscaling of picture size
- Compression
  - Adequate for player capabilities and typical transmission bandwidth
- Indexing
  - Analyzing internal structure
- Metadata creation
  - Possibly including digital rights specification

Encoding and Transcoding

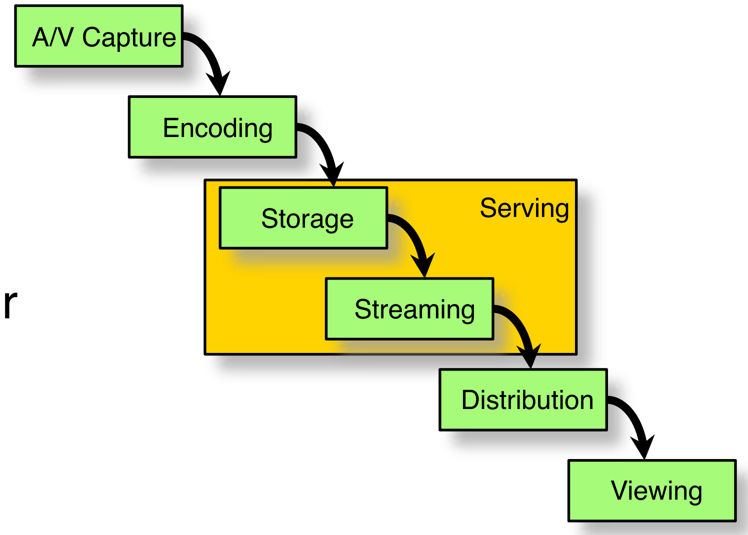


- Audio and video needs to be converted for streaming delivery
  – Compression, proprietary formats
- *Transcoding*: Conversion of media files from one format to another
- *Repurposing:* Using existing content for new purposes
  – e.g. using TV ads as streaming content

Automated Transcoding

- Example 1: Multiple Formats
  - TV Broadcaster
  - Repurposing into streaming media for Web-based Video-on-Demand
  - Live capturing, encoding (e.g. MPEG)
  - After program end: transcoding to different bit rates, delivery to streaming server
- Example 2: Flipping on Demand
  - Media archive for a cable channel to be made available through Web
  - Media kept in single, high-quality format
  - On demand (request), files are transcoded, watermarked, streamed
- Example 3: Collaboration Distribution
  - Large company working on marketing materials
  - One rough cut of a new commercial to be distributed to 100 clients with varying quality expectations and platforms
  - *Content distribution service* transcodes according to client requirements
- Example product: Telestream Vantage (www.telestream.net)

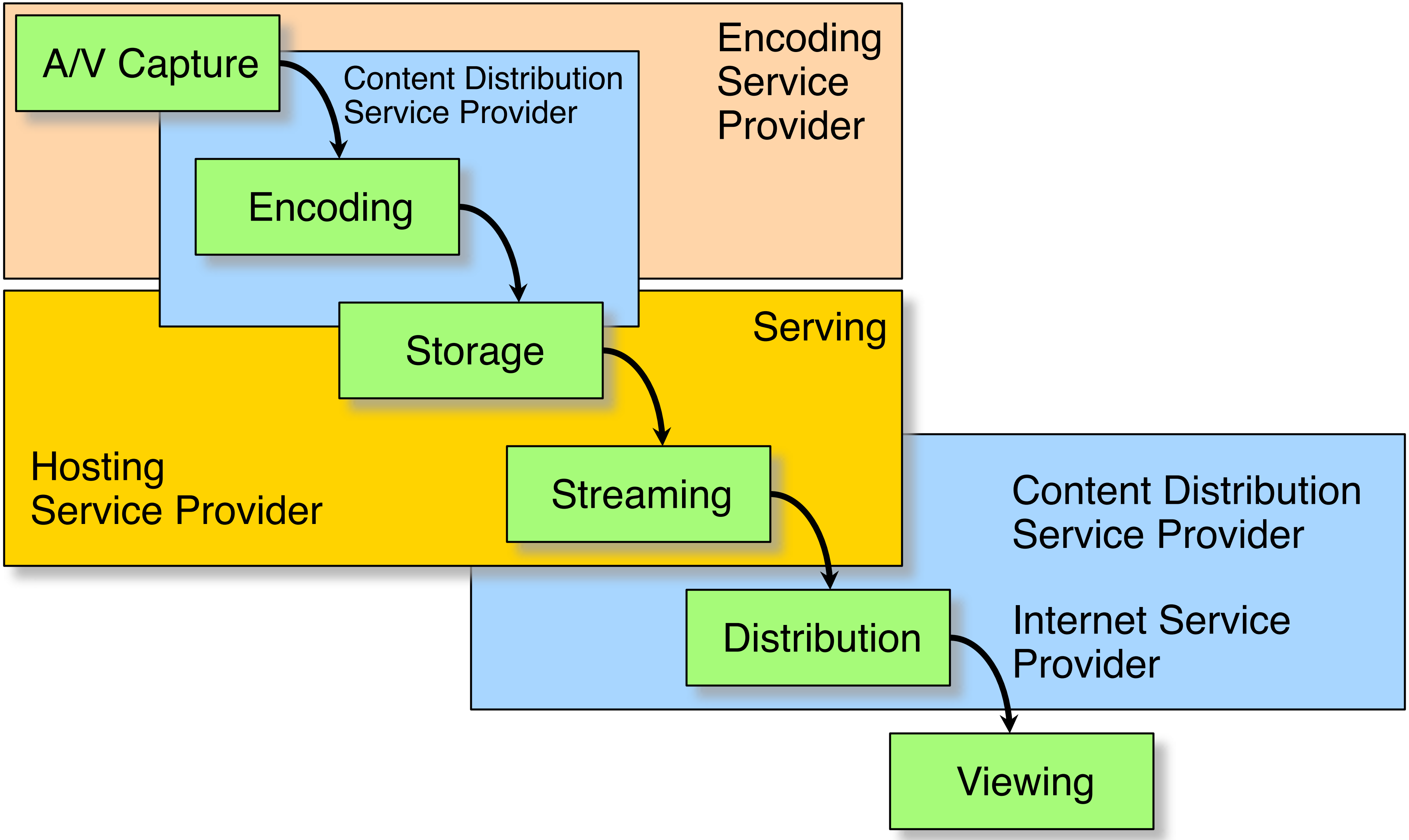
Distribution



- Key topic: Quality of Service (QoS)
  - Determining realizable bandwidth, delay, jitter

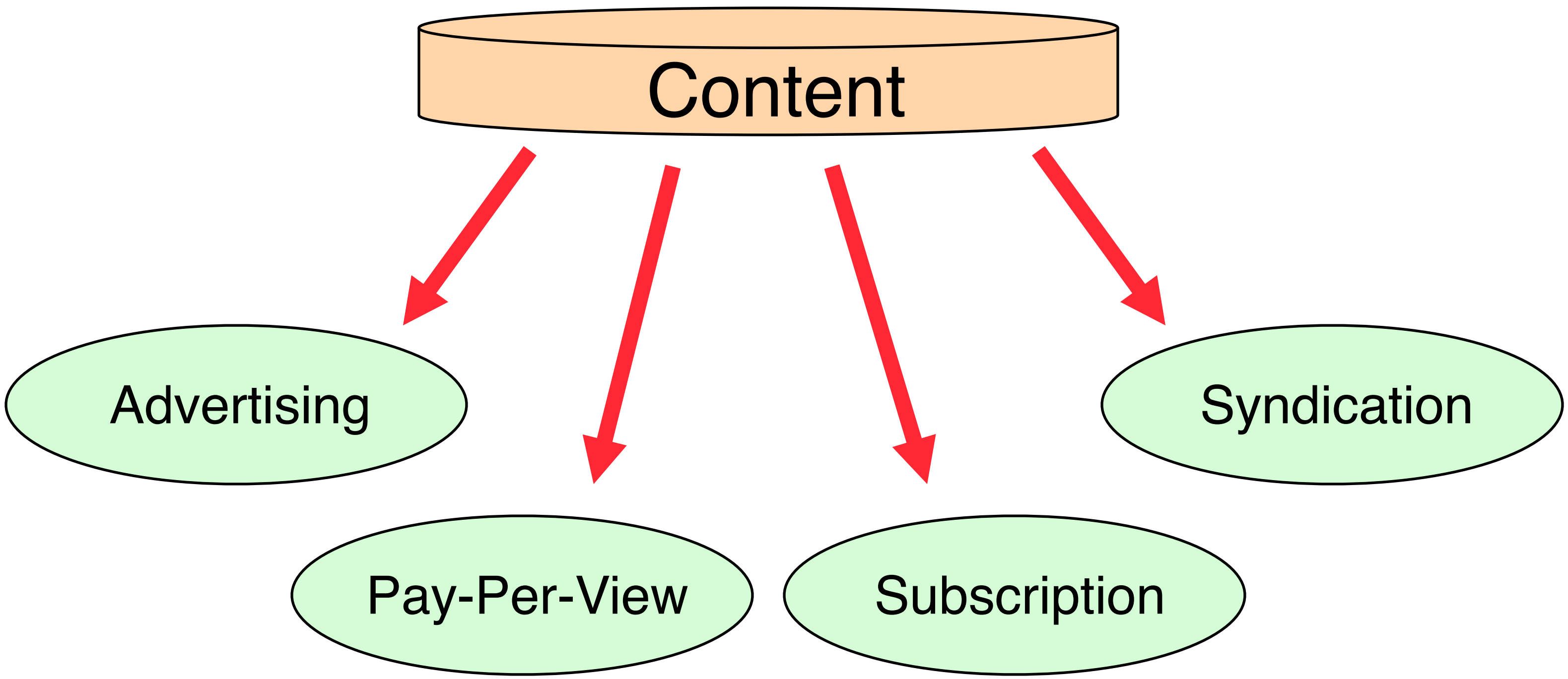
- Key concepts:
  - Overprovisioning
  - Detailed reservations (“Integrated Services”, reservation protocol RSVP)
  - Traffic classes (“Differentiated Services”)
  - Resource management layer
  - Solutions based on specific network technologies
    - » E.g. ATM (Asynchronous Transfer Mode)

Organisations in the Streaming Delivery Chain



Content Monetization

- There are several traditional models for gaining a return on investment on content
  - Network-based media enable the integration of all models



10 Multimedia Content Management and Distribution

Literature:

David Austerberry: The Technology of Video & Audio Streaming, Focal Press 2002

Stephan Rupp, Gerd Siegmund, Wolfgang Lautenschlager: SIP – Multimediale Dienste im Internet. dpunkt 2002

A. Begen, T. Akgul, M. Baugher: Watching Video over the Web, Part I: Streaming Protocols, *IEEE Internet Computing*, March/April 2011

Push and Pull Models for Streaming

- Push model (e.g. Darwin Streaming Server):
  - Session-level connection established between server and client
  - Server continues sending packets *downstream* to client
  - Server listens to commands given by client (see later for protocols)
  - True real-time data distribution
  - Adaptive bandwidth control
- Pull model (e.g. YouTube):
  - Session-level connection established between server and client
  - Server is idle as long as client does not request data
  - Client continues requesting packets from server (e.g. by HTTP)
  - Playback starts after a certain client-side buffer level is reached
  - Essentially "progressive download", enhanced in modern approaches (see later)

IP and TCP

- Internet Protocol
  - Network communication protocol (ISO layer 3)
  - Packets transferred from address to address (through routers)
  - Main problems:
    - » Variable network latency
    - » Packet order on arrival may be different than on sending
    - » Packets may be lost
- Transport Control Protocol (TCP)
  - Connection establishment (by “three-way handshake”)
    - » Connections are sequences of associated IP packets
  - Sequencing of bytes with forwarding acknowledgement number
  - Non-acknowledged bytes are re-transmitted after a defined time period
  - Flow control
- For audio/video streaming:
  - Retransmissions (and associated delays) are harmful
  - Lost packets can be tolerated to some extent

UDP

- User Datagram Protocol (UDP)
- Extremely simple transport protocol over IP
  - Connectionless (TCP: connection-oriented)
  - Unreliable (TCP: reliable)
  - No flow control (TCP: has flow control)
- Contents of a UDP datagram:
  - Ports used by application program
  - Checksum
- Basically adequate for media data transport
  - In particular for **push**-model true streaming
  - Very efficient, protocol overhead of TCP avoided
  - Flow control and handling of packet loss have to be handled by higher protocol layer

Real-Time Transport Protocol RTP

- Transport protocol specifically developed for streaming data
  - IETF (Internet Engineering Task Force) RFC (Request for Comments) 1889
- RTP usually carried over UDP
- Used for **push** model of media transmission mainly
- Very important:
  - **RTP does not at all change the way how IP packets are transferred in the network!**
  - To achieve “Quality of Service”, additional network technologies are required (see above)
- Early uses of RTP:
  - Apple QuickTime architecture
  - RealSystems streaming architecture
- RTP still supported in state-of-the-art systems

Current Example of Live Streaming

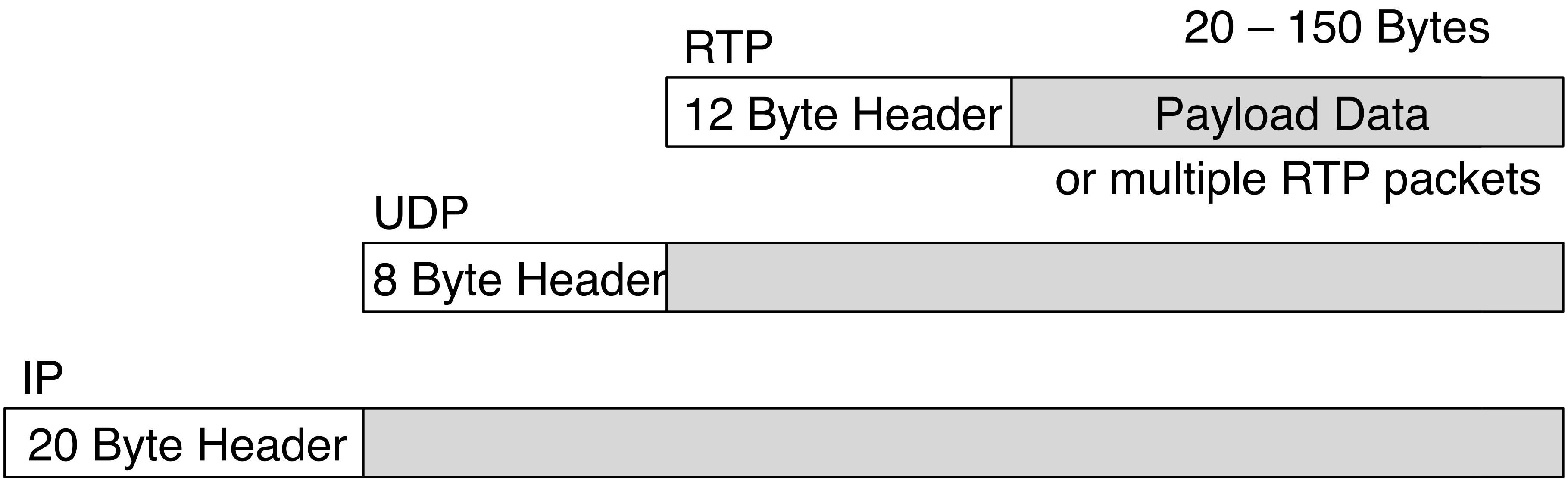
On-board and Remote Proxy Recording over RTP/RTSP

TERADEK CUBE

Live H.264 encoder

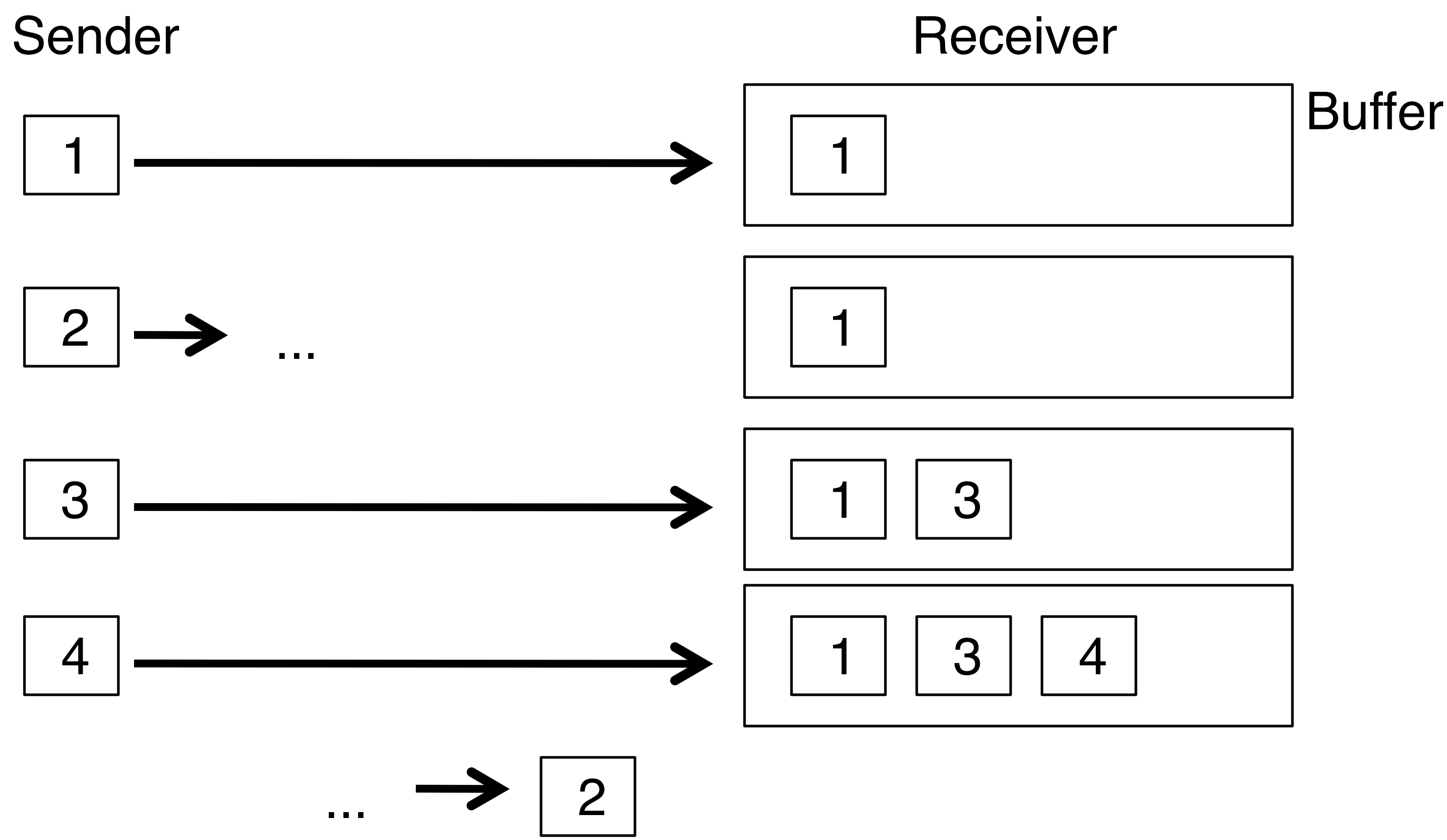
Source: teradek.com

RTP Packets and Other Protocols



- IP Header:
  - Source address, destination address, length, time to live, ...
- UDP Header:
  - Port numbers (source and target processes), length, checksum
- RTP Header:
  - Codec type, sequence number, timestamp, synchronization source

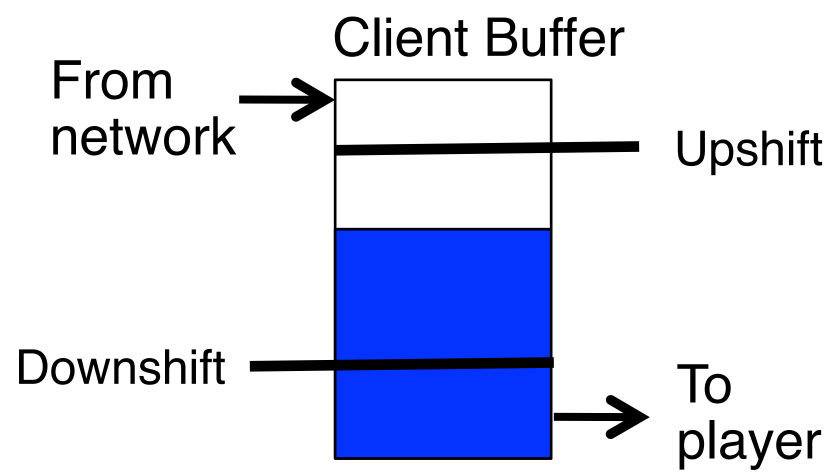
Jitter and Loss Compensation



- Options for application on receiver side:
  – Wait (*not* adequate), repeat last packet (1), interpolate (between 1 and 3)
  – Missing audio information is difficult, missing video can be compensated

Adaptive Transmission Rate Control

- Application-level mechanism
- Define “downshift” and “upshift” thresholds on buffer (possibly several)
- Downshift:
  - Buffer is close to drain (“underrun”)
  - Server should select lower-bitrate encoding (less quality)
- Upshift:
  - Buffer is close to be full (“overrun”)
  - Server can try to select higher-bitrate encoding (better quality) (if network can support that)
  - Alternatively server may switch to lower transmission rate
- Communication between client/server required!
  - No longer in mainstream use (see next section)



Session Description Protocol (SDP)

- IETF Standard (RFC 4566) 2006
- Description mechanism for multimedia communication sessions
  - announcement, invitation, parameter negotiation
- Defines:
  - media components (an their types) belonging to a session
  - alternative formats available for media components
  - session profiles
- Applicable for various purposes
  - for conferencing examples see later chapter
    - » in streaming: announcement of available options from server to client

Real-Time Control Protocol RTCP

- RTCP controls the transmission (not the setup of connection)
- RTCP periodically sends monitoring information to all participants in a streaming session
- Main functions of RTCP:
  - Feedback on QoS of transmission
    - » Information for adaptive codecs, e.g. whether problem is local or global
  - Identification of sender by “canonical name”
    - » Helpful when synchronization source changes
    - » Supports lip synchronization between audio and video
  - Number of participants in a session
    - » Adaptation of sending rate of RTCP control information to number of participants, to avoid network overload
  - Transmission of additional information, e.g. names of session participants

Real Time Streaming Protocol RTSP

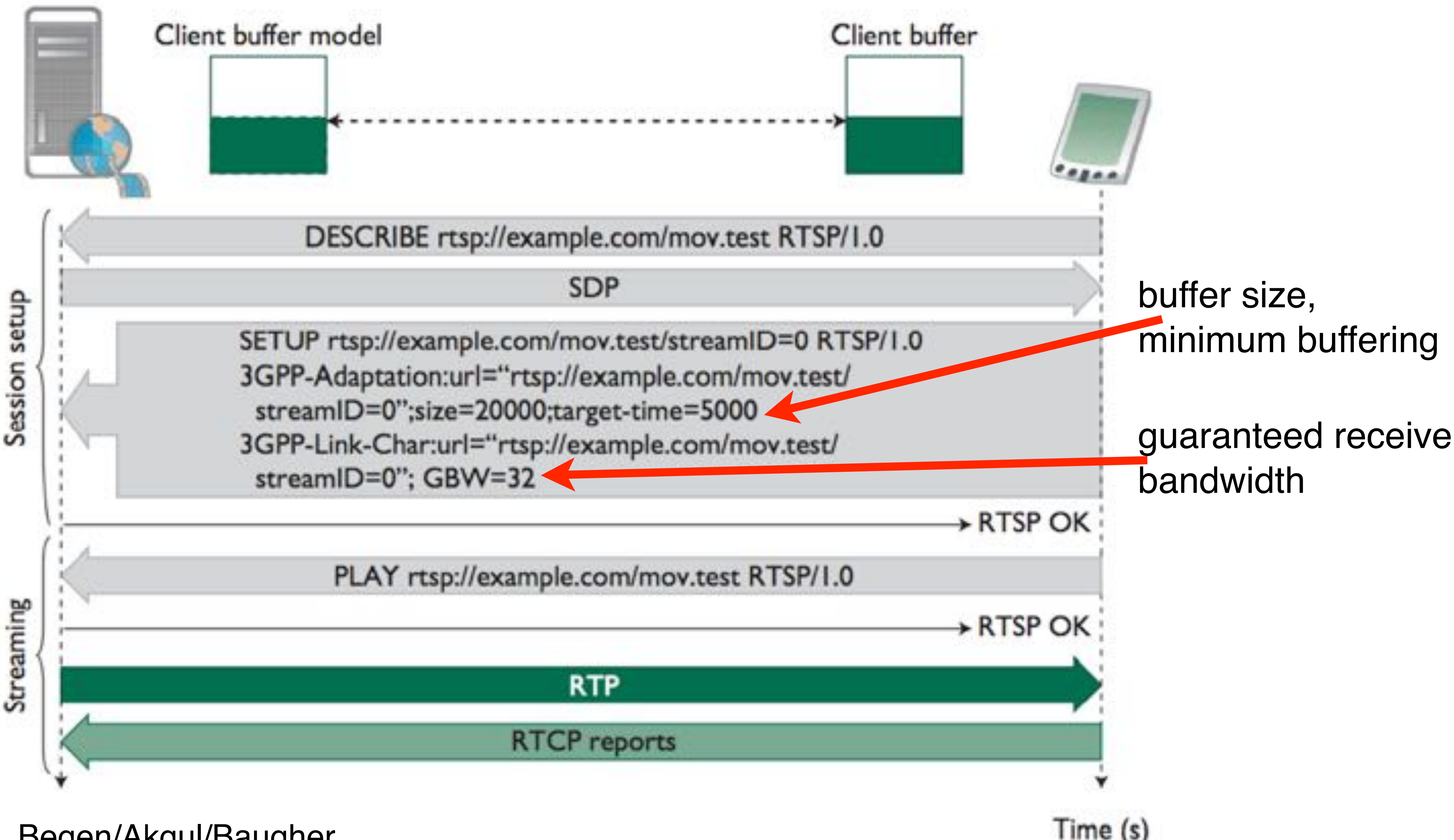
- Client-server multimedia presentation protocol, designed specifically for streamed media
  - IETF (Internet Engineering Task Force) RFC (Request for Comments) 2326 ("MMUSIC" work group), 2004
  - "The Internet VCR remote control protocol" (www.rtsp.org)
  - Independent of the use of RTP for transport
  - Syntactically similar to HTTP 1.1 (carried over TCP or UDP)
- Main operations supported by RTSP:
  - Transport & capability negotiation (DESCRIBE, SETUP)
    - » e.g. disallowing a "seek" function
  - Session control (SETUP, REDIRECT, TEARDOWN)
  - Control of media playback (PLAY, PAUSE, PING)
  - Invitation of a media server to a conference

RTMP, RTMPT, RTMPS, ...

- "Real-Time Messaging Protocol"
  - Variants: Tunneling (through HTTP), secure transmission…
- Adobe-proprietary protocols
  - Mainly used for Flash-based players
  - Open format, specification freely available

3GPP Bitrate Adaptation

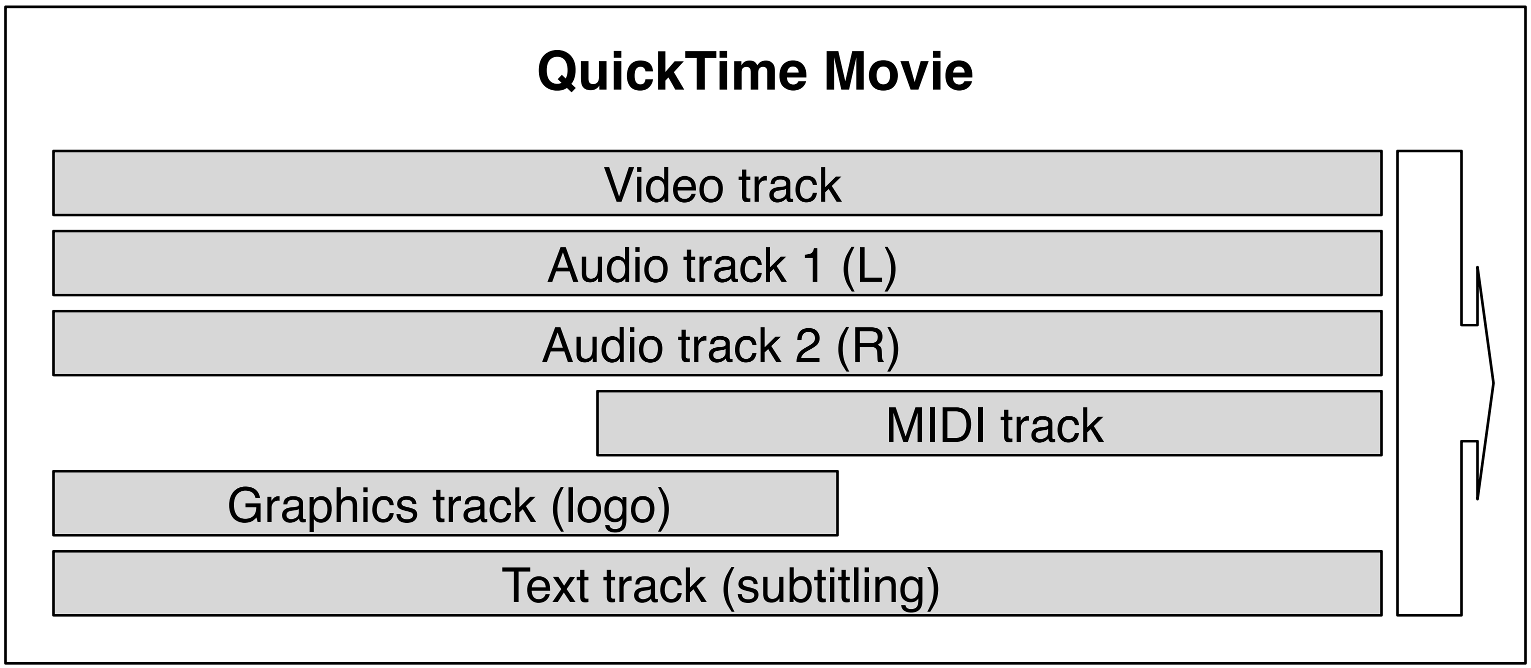
3rd Generation Partnership Project (3GPP)



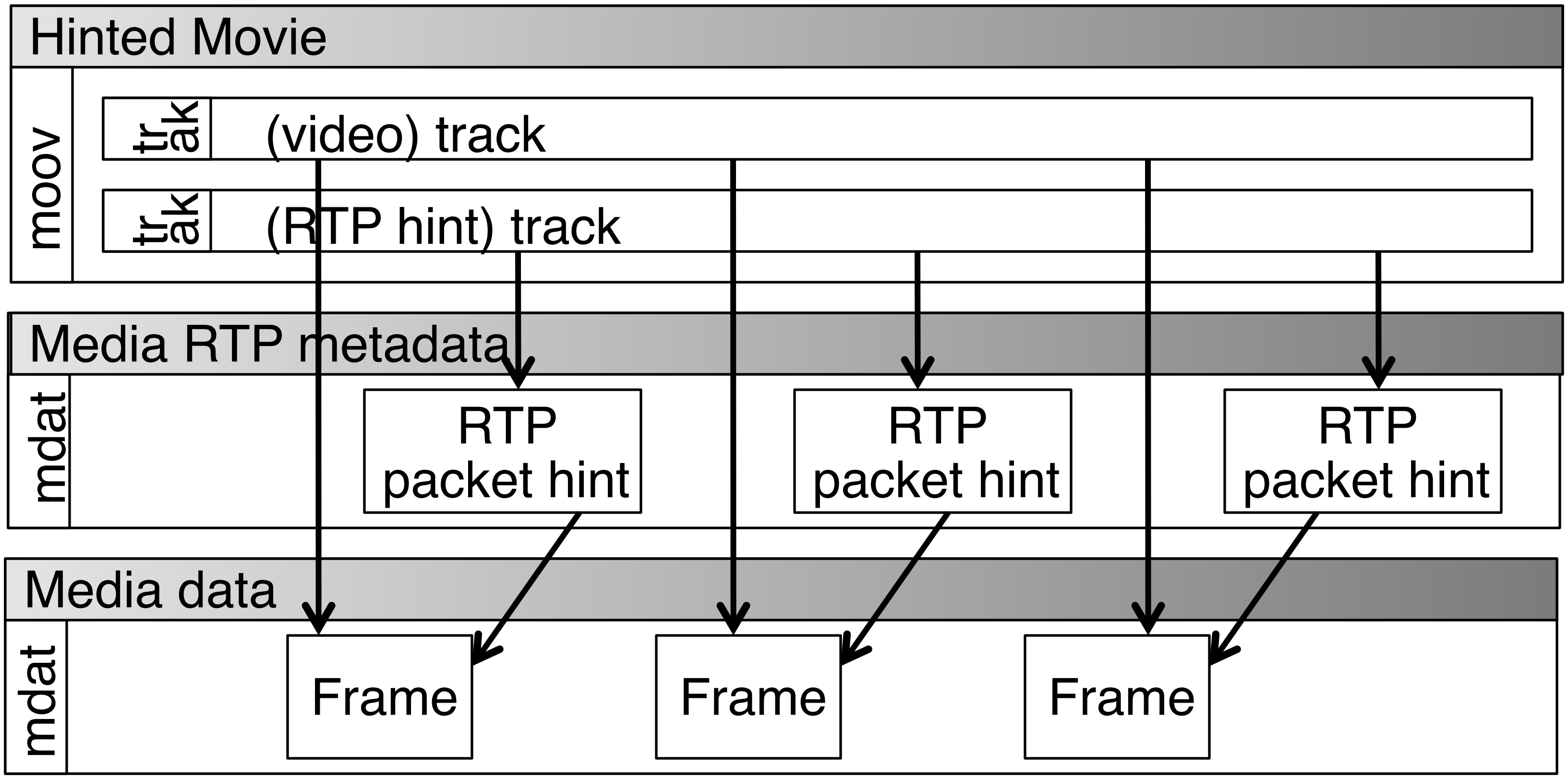
Begen/Akgul/Baugher

Example File Format: QuickTime Movie Files

- Modular and flexible architecture
  - Multimedia files organized in tracks
  - Example:



Hint Tracks in QuickTime and MPEG-4



- Hint track gives server software pointers to the RTP information to serve the relevant media chunks
- Concept from QuickTime, integrated in MPEG-4 (streaming)

10 Multimedia Content Management and Distribution

Literature:

A. Begen, T. Akgul, M. Baugher: Watching Video over the Web, Part I: Streaming Protocols, *IEEE Internet Computing*, March/April 2011

I. Sodagar: The MPEG-DASH Standard for Multimedia Streaming Over the Internet, *IEEE Multimedia*, Oct/Dec 2011

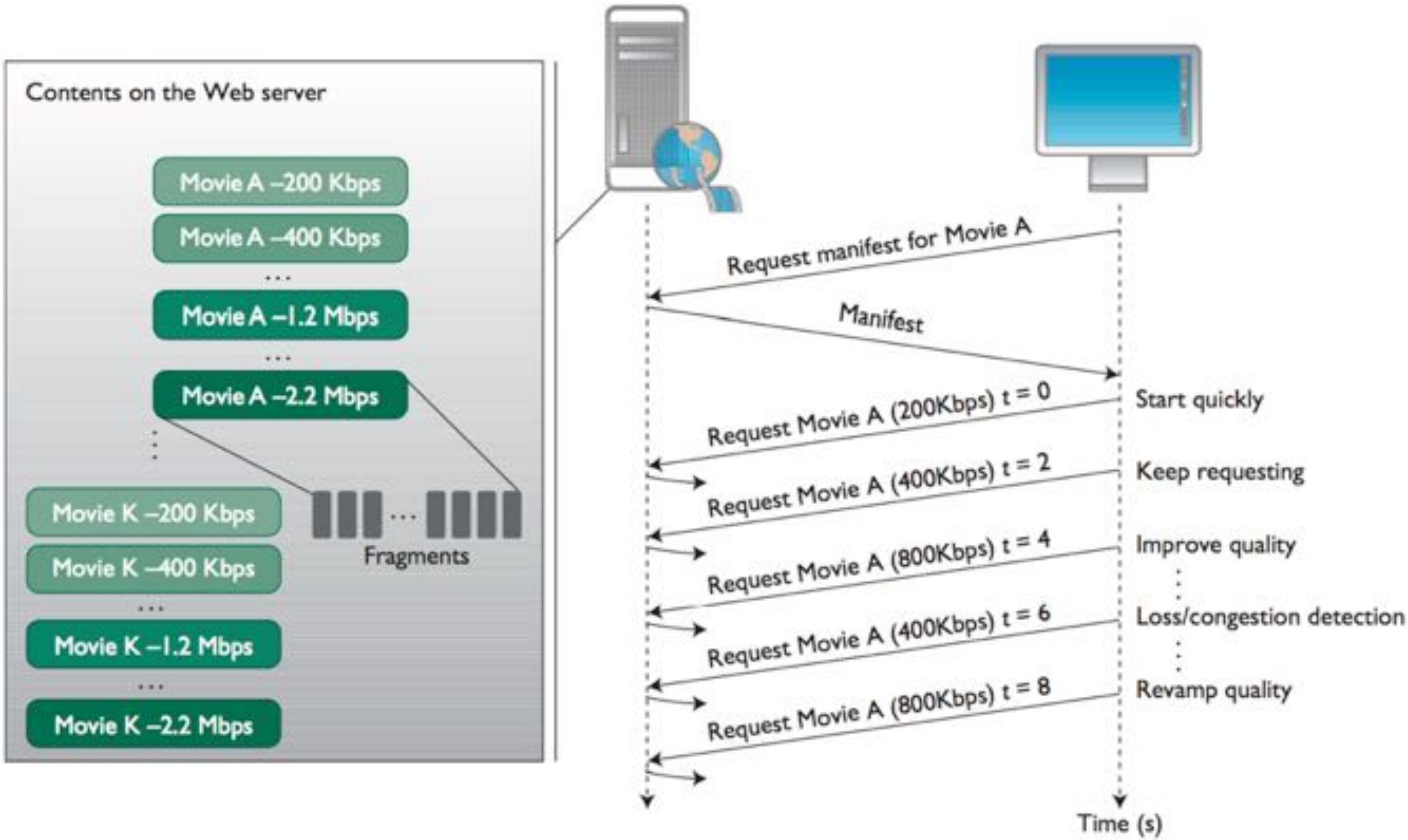
Trend Towards Pull-Model Streaming

- Internet video around 2009, alternatives:
  - Push-model, server-centric streaming with bitrate control (e.g. RealPlayer)
  - Progressive download (pull-model) without bitrate control (e.g. YouTube)
- Disadvantages of push-model streaming:
  - Firewalls, NAT devices etc. support HTTP ports but often not other protocols
  - Server responsibility for bitrate adaptation is problematic for scalability
- Disadvantages of pull-model progressive download:
  - Unpredictable quality of experience, in particular frequent stalling of video
  - Legal issue: Progressive download may produce full (temporary) copy of file
- Technology evolution:
  - Generally higher network speed, better support for larger chunks of data
  - HTTP caching infrastructure
- How to create a scalable hybrid approach?
  - Pull-model, HTTP-based like progressive download
  - Adaptive bitrate control

Adaptive HTTP Streaming

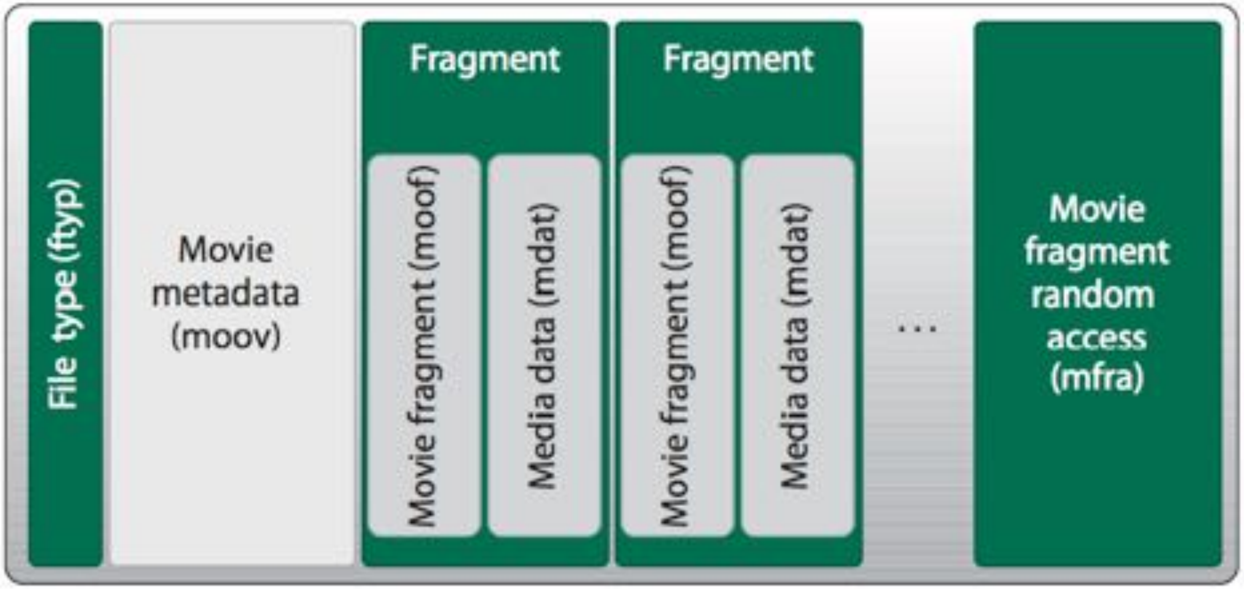
- First development step:
  Independent proprietary solutions for adaptive HTTP-based streaming:
  - Apple HTTP Live Streaming
  - Microsoft Smooth Streaming
  - Adobe HTTP Dynamic Streaming
- Second development step:
  - DASH standard for adaptive HTTP-based streaming (see below)
- Basic idea: Small video file fragments
  - Fragments exist at different bit rates
  - Index file for all available fragments
  - Client requests appropriate next fragment by GET request

Bitrate Adaptation in HTTP Adaptive Streaming



Begen/Akgul/Baugher

Example: Video Fragmentation in Microsoft Smooth Streaming



MPEG4-based format for fragmented video (PIFF)

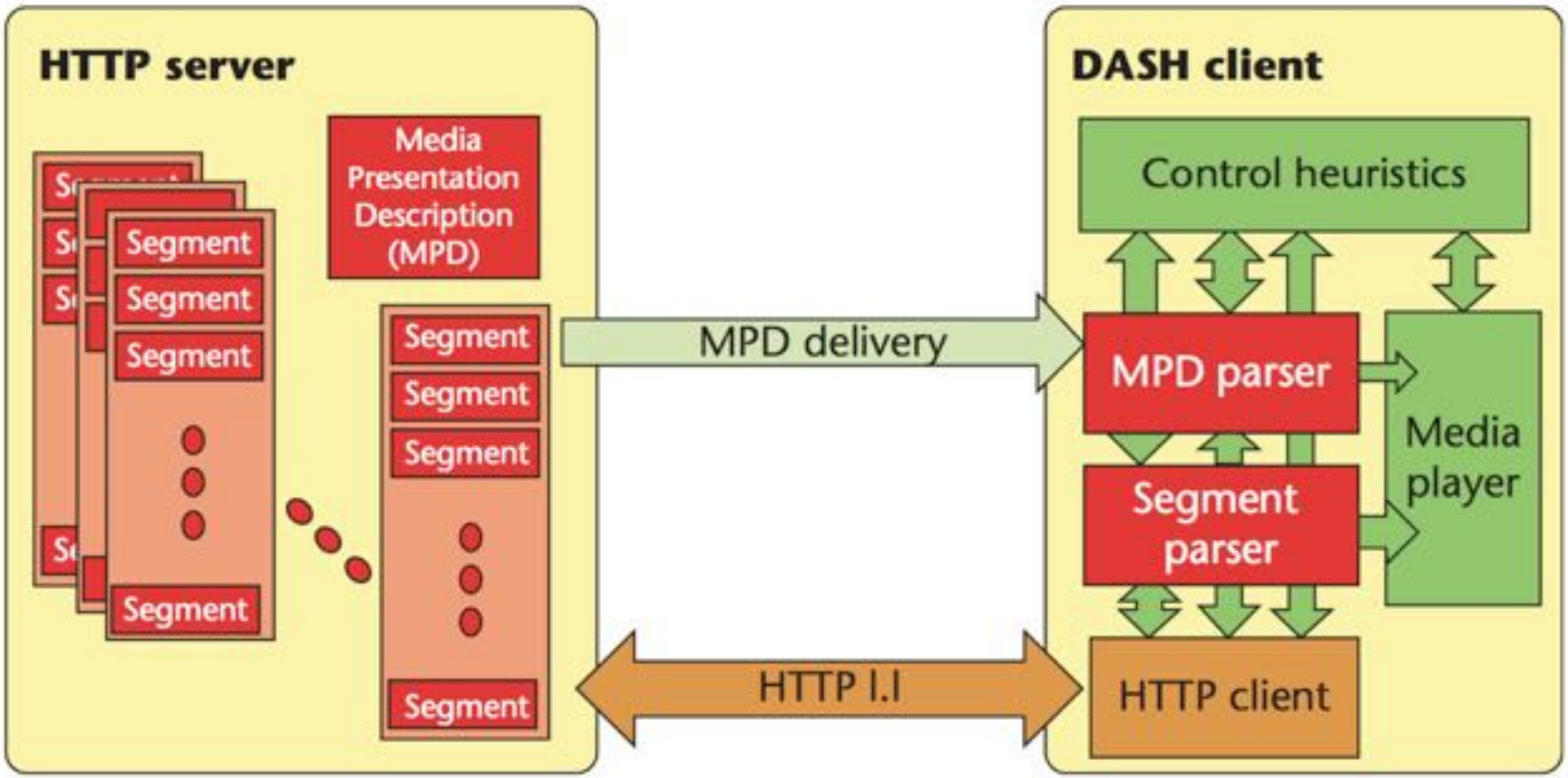
- Fragments always contain complete Groups of Pictures (GOP)
- Client (Silverlight-based Web player) sends requests which contain:
  - requested bitrate (version of the video)
  - requested time offset for fragment

```
GET/example.host/myvideo.ism/QualityLevels(64000)/
  Fragments(video=150324) HTTP/1.1
```

Begen/Akgul/Baugher, based on alexzambelli.com/blog/2009/02/10/smooth-streaming-architecture/

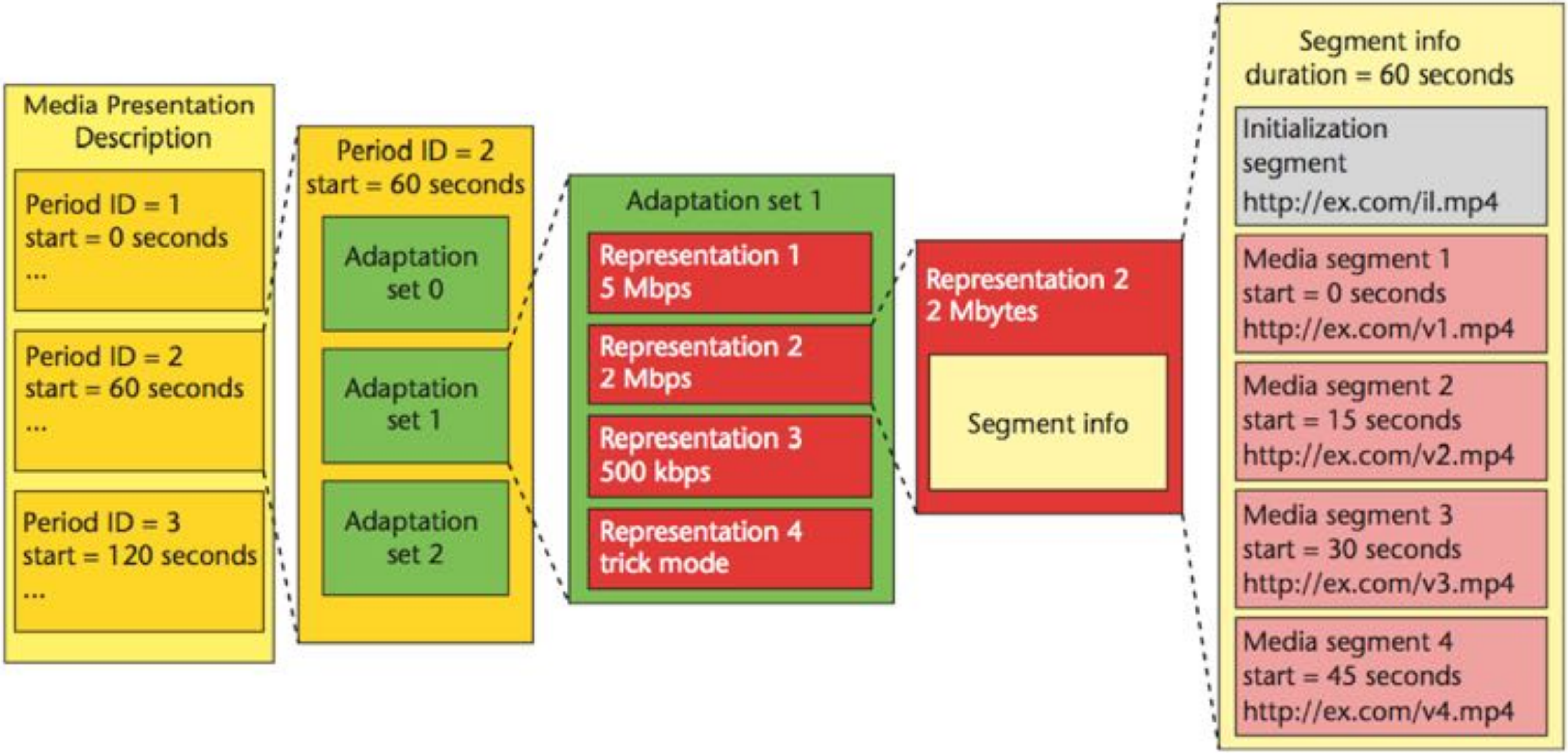
MPEG-DASH Standard

- MPEG call and selection process for HTTP streaming standard (2009–2011)
- ISO/IEC standard (ISO 23009) April 2012



Sodagar

Media Presentation Description in DASH



Sodagar

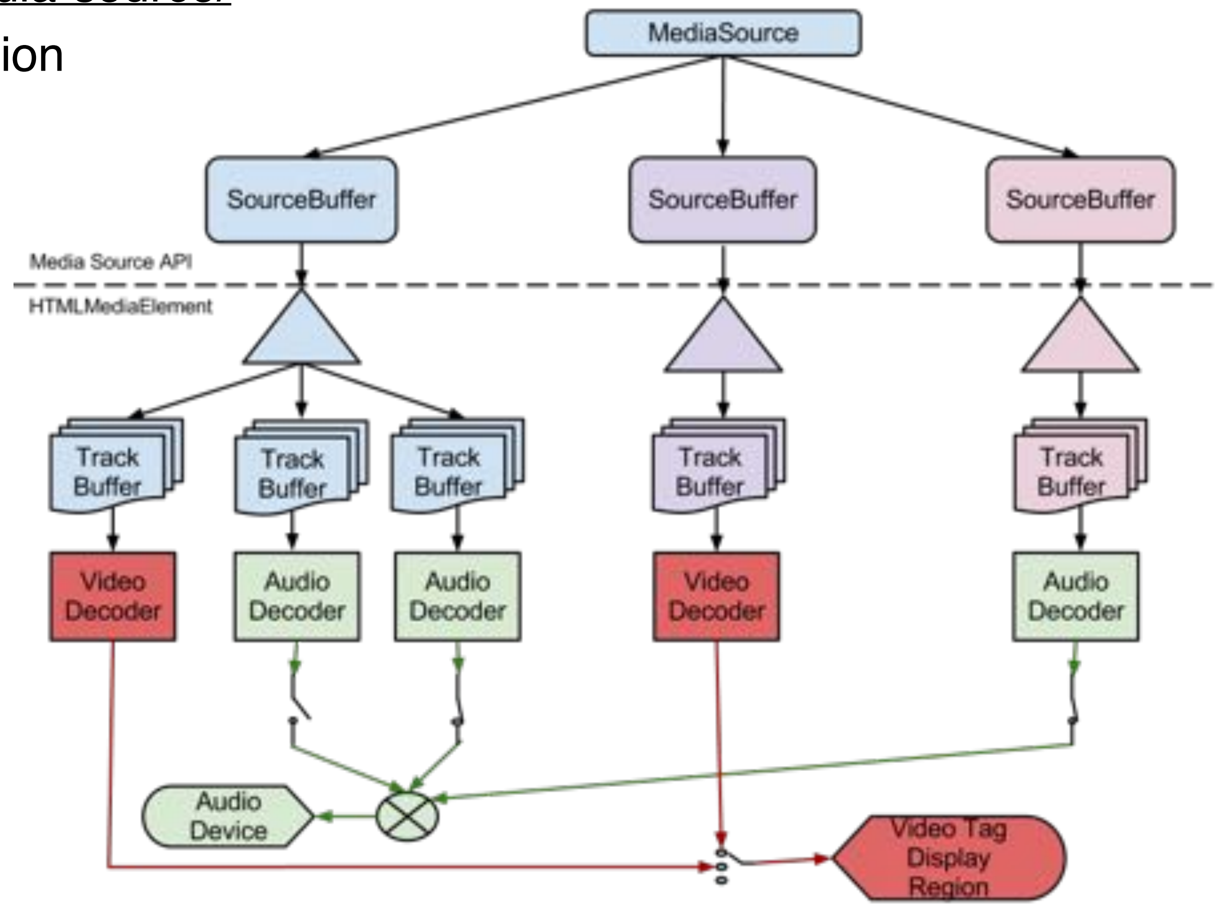
Adaptive HTTP Streaming, Client-Side

- DASH access library:
  - Parsing of Media Presentation description (MPD)
  - Management of HTTP download of segments
  - Reference implementation *libdash* (https://github.com/bitmovin/libdash/)
- DASH streaming control:
  - Heuristics of switching between representations
  - "Quality of Experience" (QoE) optimization
  - Usually part of player component
  - Trend: Self-learning clients (machine learning)
- DASH player implementations:
  - Plugins for media players (e.g. DASH plugin for VLC player)
  - DASH player components for legacy browsers, e.g. based on Adobe Flash
  - JavaScript-based DASH players
    - » JavaScript code building on browser APIs

Example: YouTube "Stats for Nerds"

HTML5 Media Source Extensions (MSE)

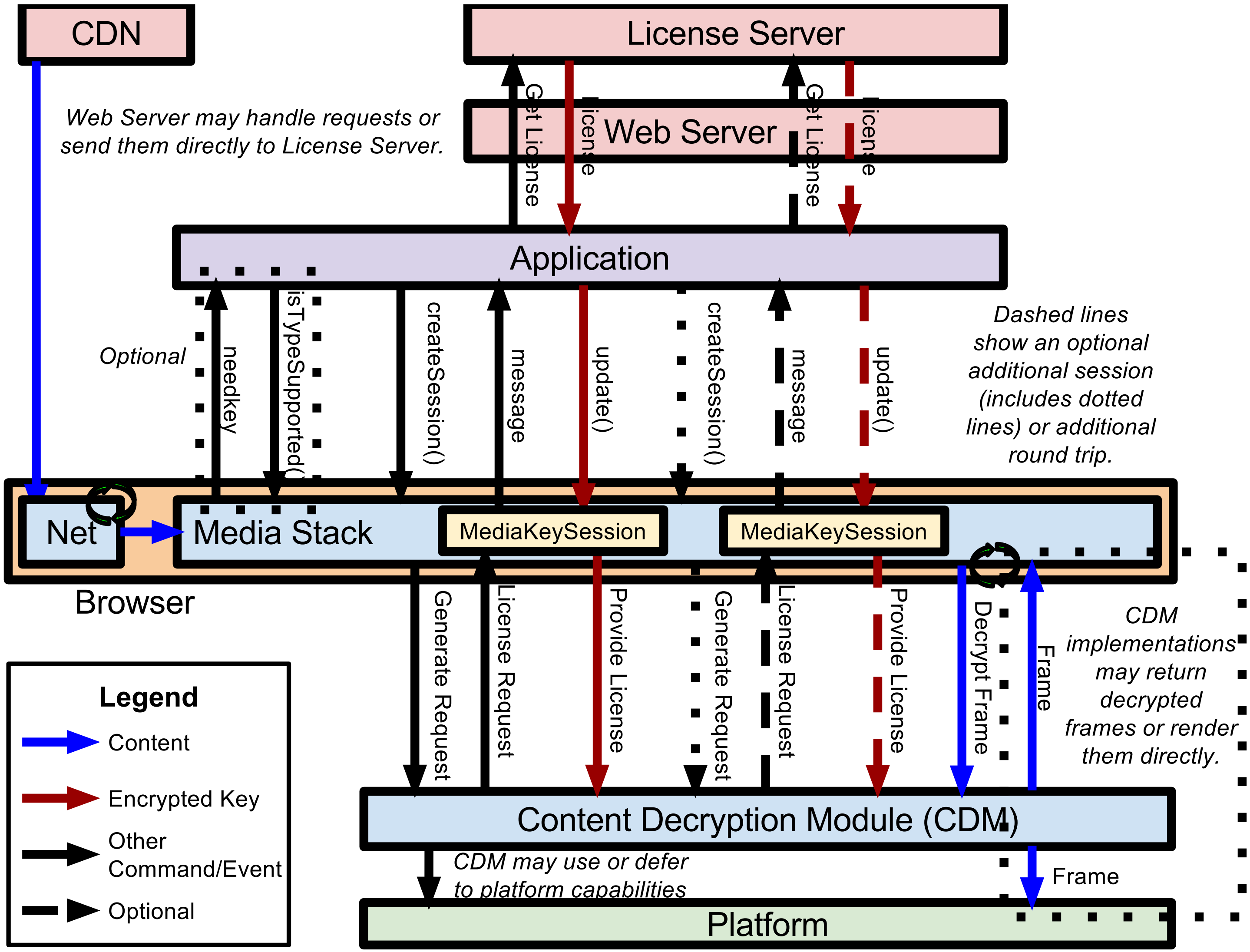
- W3C recommendation under development
  - http://www.w3.org/TR/media-source/
  - Candidate Recommendation 12 November 2015
- Goals:
  - JavaScript interface for HTML Media Elements to receive spliced and buffered media streams
  - Use cases:
    - » adaptive streaming
    - » ad-insertion
    - » time-shifting
    - » video editing



HTML5 Encrypted Media Extensions (EME)

- W3C recommendation under development
  - http://www.w3.org/TR/encrypted-media/
  - Latest Editor's Draft 16 December 2015
- Highly controversial within HTML5 community!
- Main authors: Google, Microsoft, Netflix
- (JavaScript) API for HTML5 media elements to control playback of encrypted content
  - API to discover, select and interact with DRM systems

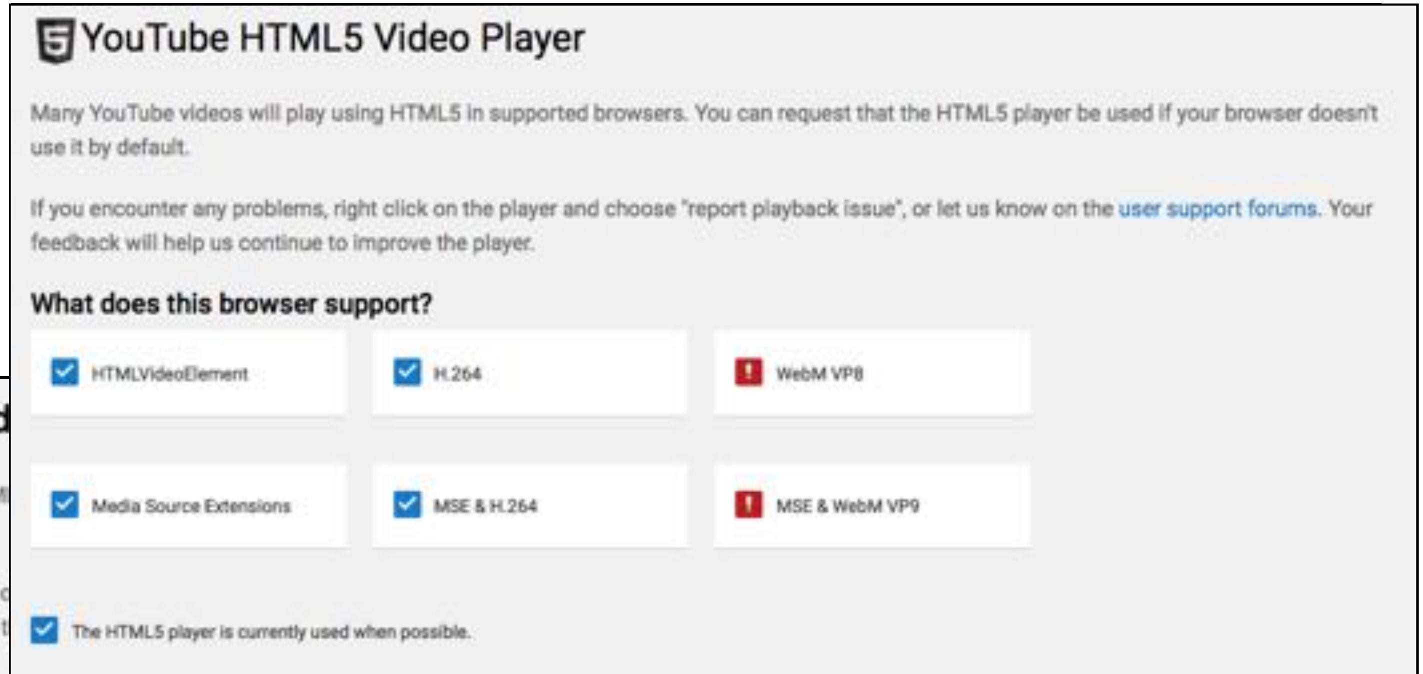
HTML5 EME Stack Overview



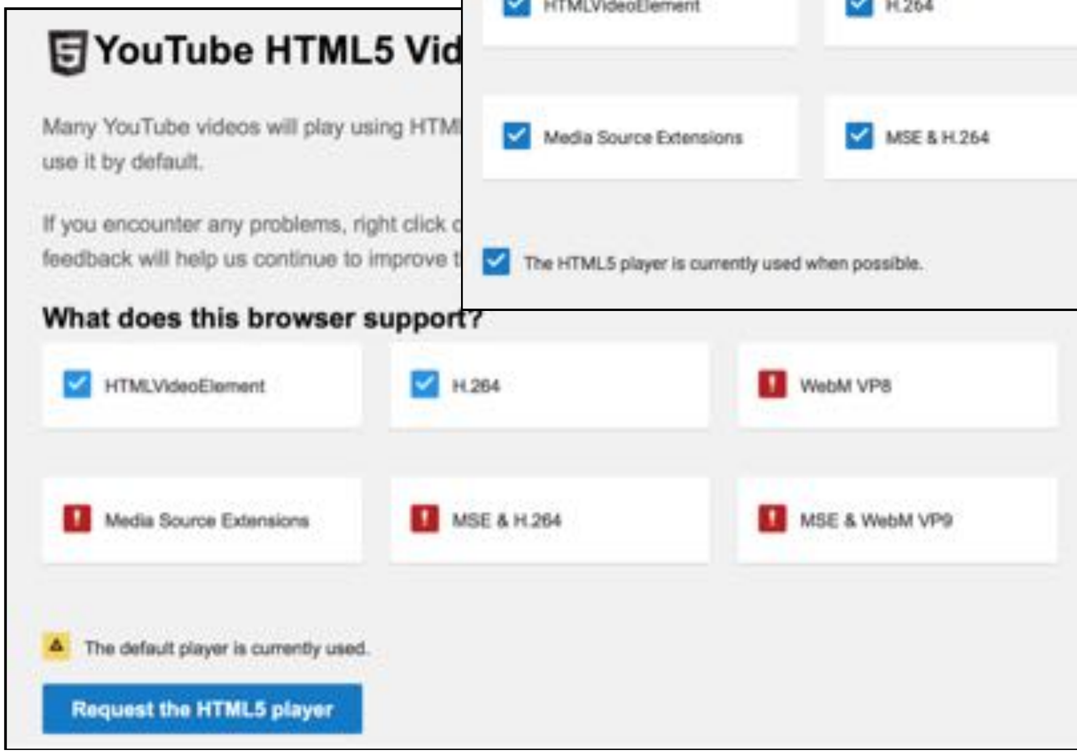
Source: W3C

Example: YouTube HTML5 Video Player

Chrome 47



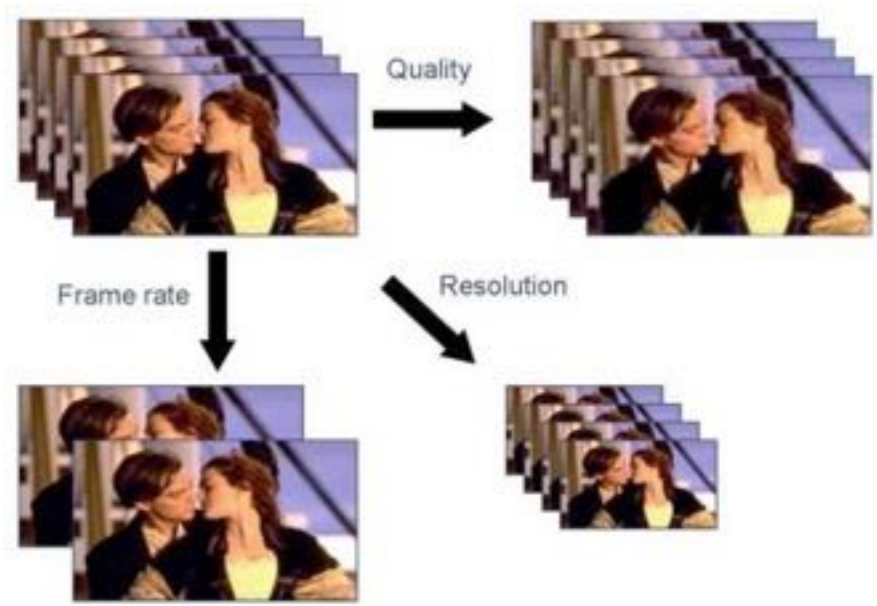
Safari 9
(OS X 10.11 El Capitan)



Safari 7 (OS X 10.9 Mavericks)

MPEG-4 Scalable Video Coding (SVC)

- SVC: Splitting of raw video data into multiple streams
  - MPEG/ITU-Standard 2007 (H.264/AVC Extension)
  - subsets of the data, self-contained
  - subsets can be combined to achieve better quality
  - dimensions: temporal resolution (frame rate), spatial resolution, SNR (loss)
- DASH is compatible with SVC
- Strategical question for player:
  - Download additional quality information for current segment, or
  - Download future segment (in lower quality)?
- See: Andelin et al., Quality selection for Dynamic Adaptive Streaming over HTTP with Scalable Video Coding, 3rd Multimedia Systems Conference, 2012



www.imec.be

MPEG-4 Multiple View Coding (MVC)

- SVC: Multiple camera images in a multiplexed video stream
  - MPEG/ITU-Standard 2008 (H.264/AVC Extension)
- Main application area: Stereoscopic (3D) video
  - Generalizable to multiple camera positions
- DASH is compatible with MVC
  - Server announces availability in MPD
  - Client adapts requests to the current needs

10 Multimedia Content Management and Distribution

Literature:

David Austerberry: The Technology of Video & Audio Streaming, Focal Press 2002

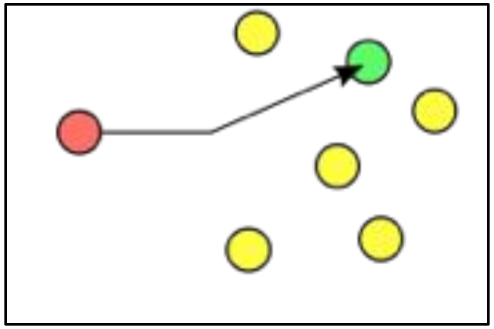
Gregory C. Demetriades: Streaming Media, Wiley 2003

Xueyan Tang et al.: Web Content Delivery, Springer 2005

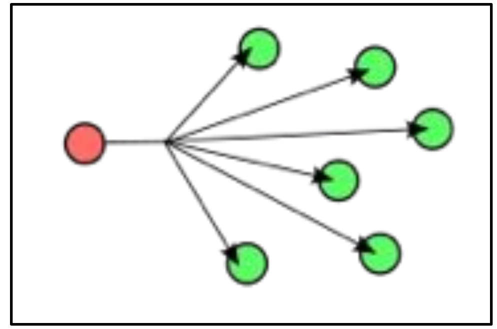
Bandwidth Economy

- Fully heterogeneous individual requests:
  - Required bandwidth = stream bandwidth x number requests
- Homogeneity of requests helps saving bandwidth:
  - Same content for many clients, but different playback times:
    - » Broadcast with caching
  - Same content at same time for many clients
    - » Multicast (splitting streams)
- Pre-planning saves bandwidth
  - (Individual) transmission of pre-booked content in non-real time ("download and play")

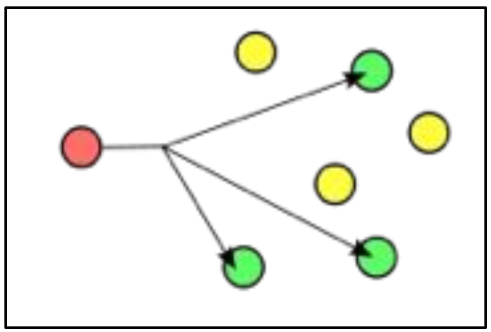
Unicast, Broadcast, Multicast, Anycast



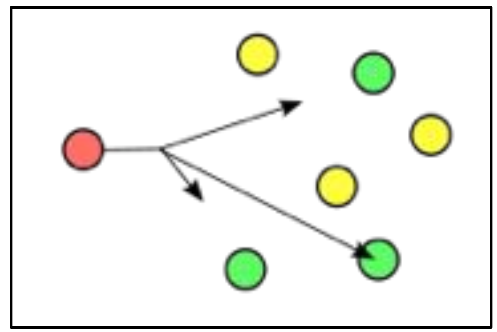
Unicast:
One specific receiver



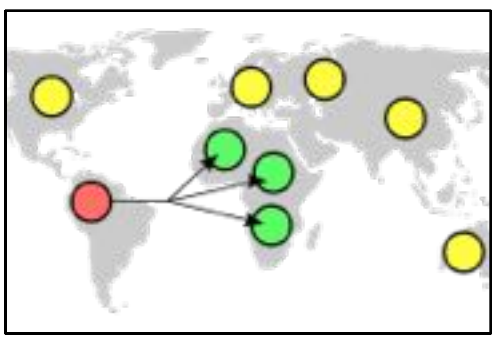
Broadcast:
Many receivers, all on the network



Multicast:
Many receivers, all of a specific group



Anycast:
One receiver, "nearest" of a specific group

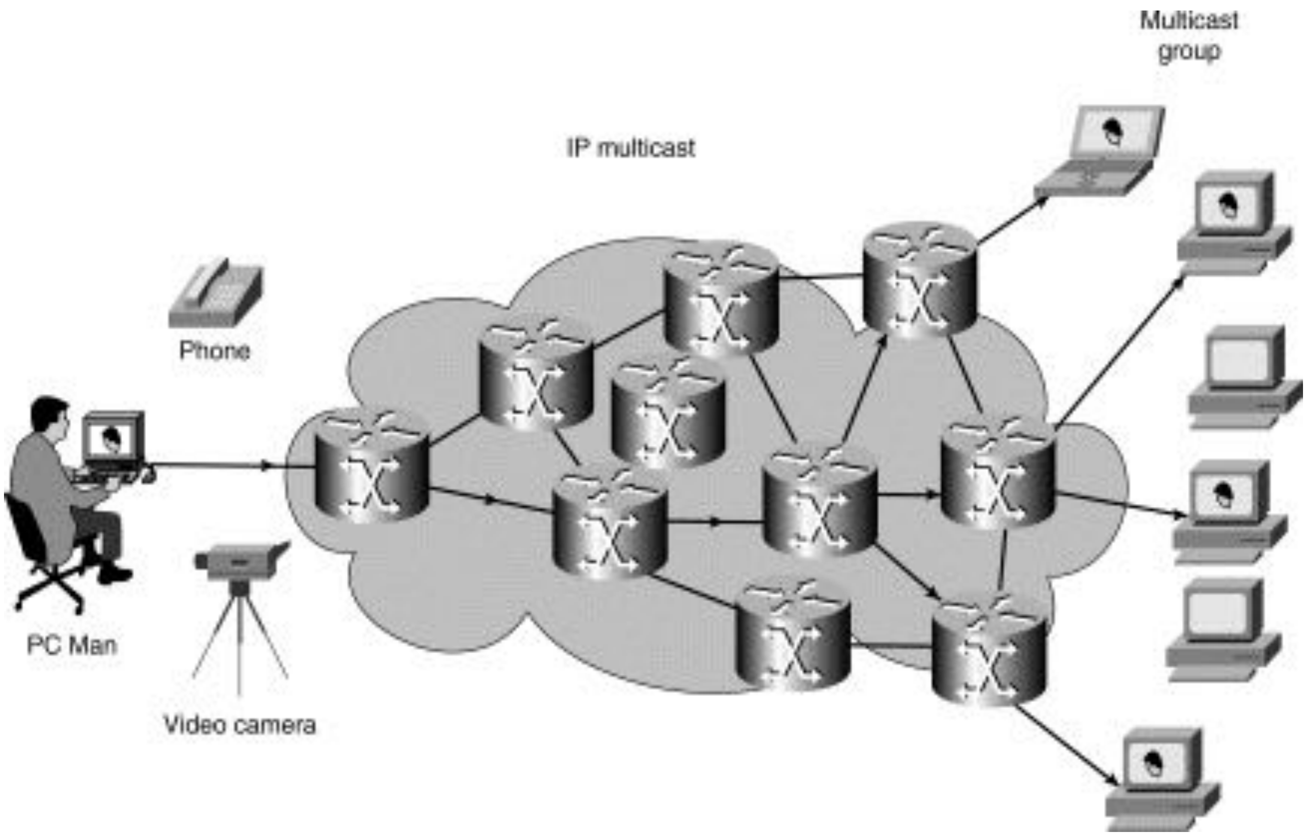


Geocast:
Many receivers, all of a geographic region

Pictures: Wikipedia

Traditional Solution: IP Multicast

- Multicast relatively easy to integrate in routers
- IP versions 4 and 6 provide special multicast addresses



- Reliable multicast: e.g. “Mbone” overlay network, 1992
- Multicast still rarely used in today’s Internet
- Problems: Charging and access control

Content Delivery Networks (CDN)

- Serve content closer to to the user
  - "edge serving"
- Main components of CDN:
  - Smart routing
  - Edge caching of content
  - Proxy serving
  - Splitting of live webcasts
- Examples:
  - Akamai
    - » Runs 160,000+ servers in 95 countries
  - Limelight Networks
    - Own physical network
  - Amazon CloudFront

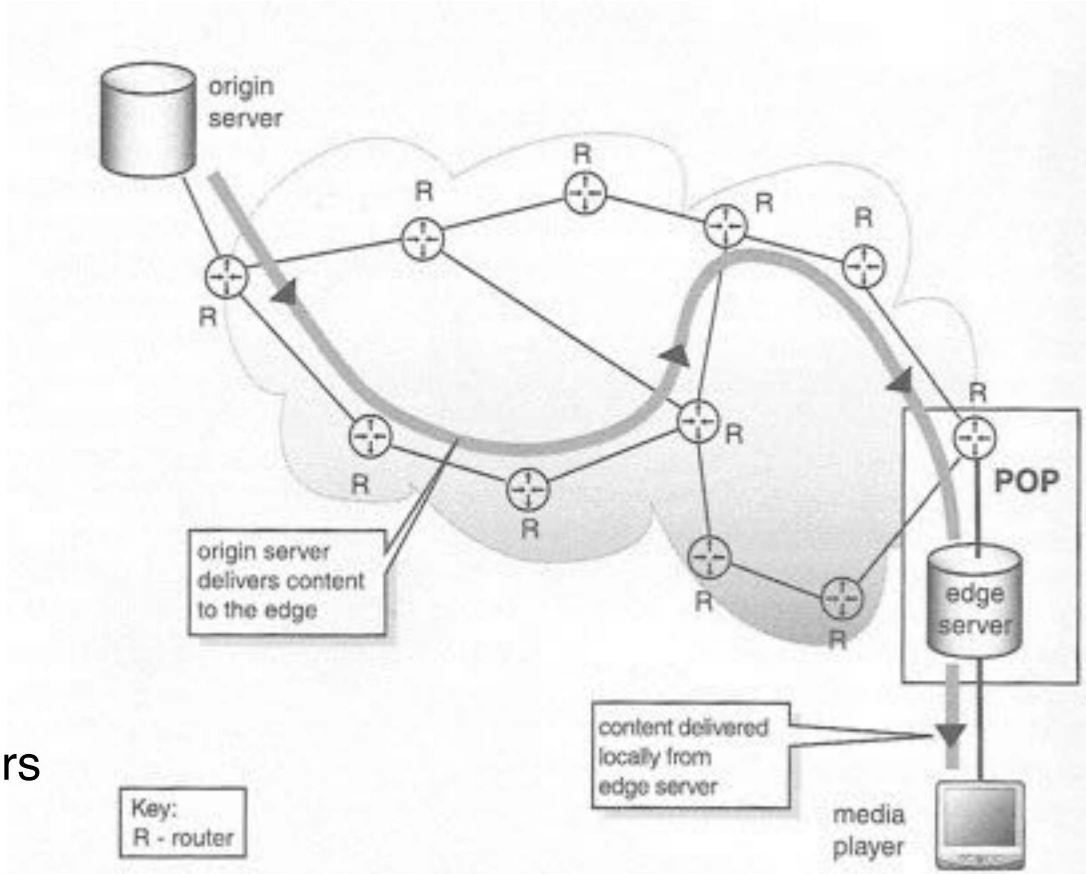


Figure 15.3 Serving from the edge with a CDN.

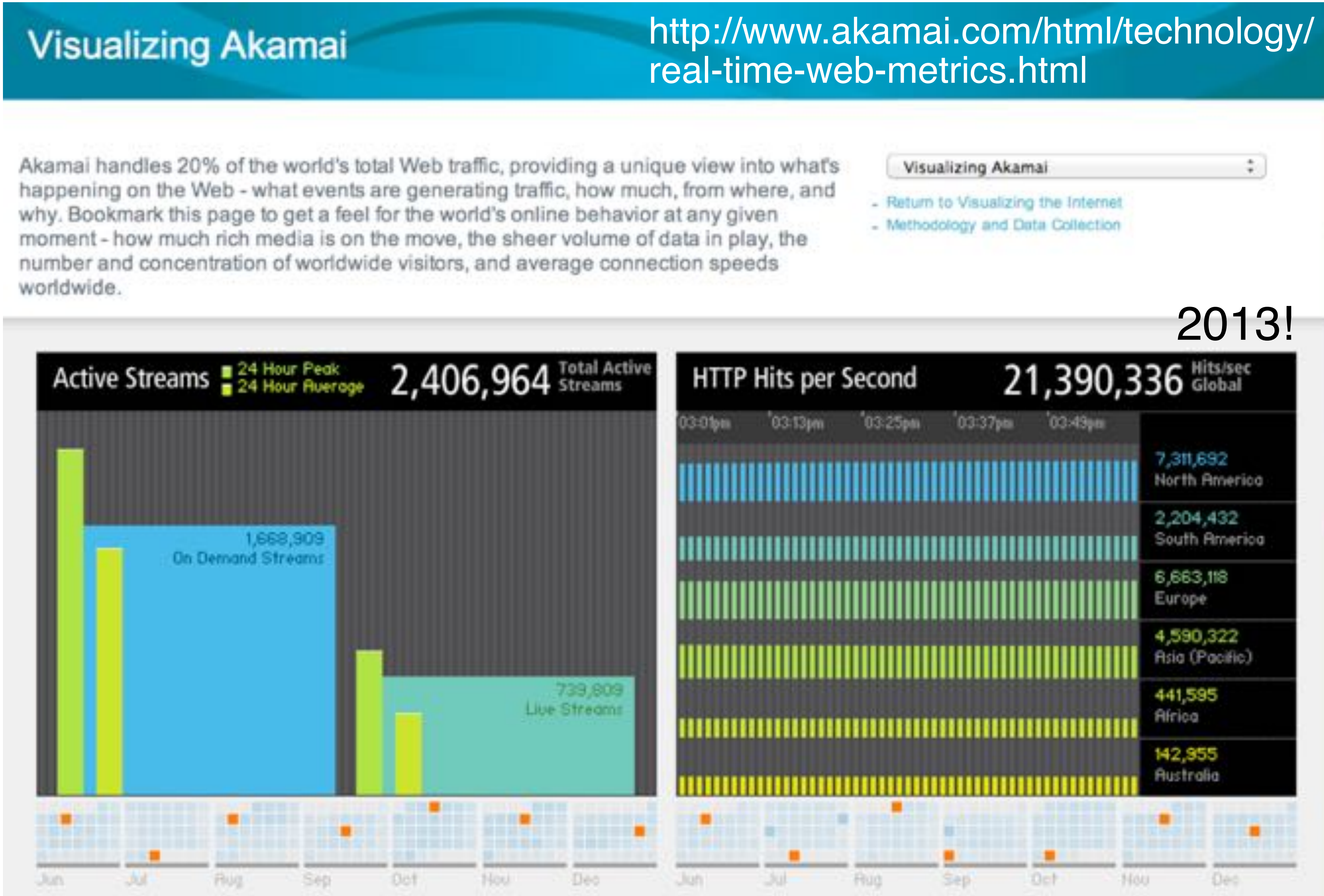
Figure from Austerberry 2002

See also www.cdnplanet.com/cdns/

Key Problems in CDNs

- Replica placement:
  - Where to place copies of web sites or other content
  - Problem is in general NP-hard (Karlsson, Karamolis, 2004)
  - Replica placement algorithms (RPA) achieve a suboptimal solution within reasonable time frame
  - Global information is difficult or costly to get - RPA uses local information mostly
  - CDN providers typically try to observe global network performance to some extent
- Request routing:
  - Mechanism and policy of redirecting client requests to a suitable server containing the requested content
  - Redirection algorithm: Decides what node to direct a client request to
  - Redirection mechanism: Way of redirecting the request (client, network)

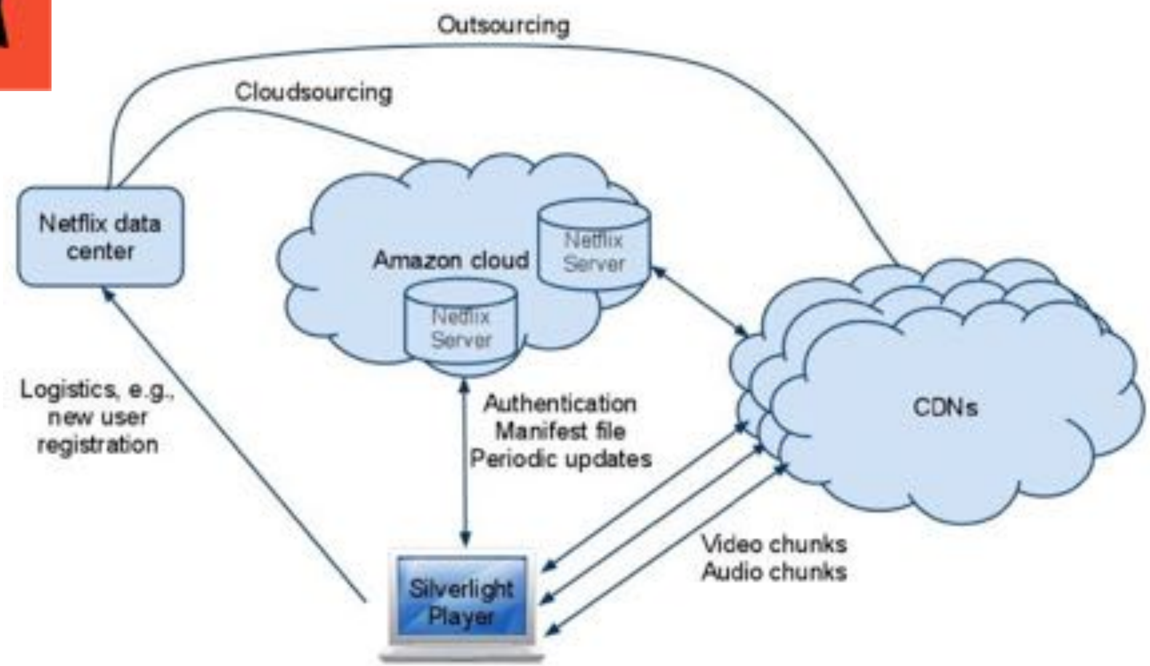
Example: Visualizing Akamai



http://www.akamai.com/html/technology/real-time-web-metrics.html

2013!

Case Study: NETFLIX



- Netflix
  - North America market leader for online movies
  - 23 mio subscribers
  - HD video, up to 3.6 Mbps
  - Single largest source of Internet traffic in US (29.7% peak downstream)
- Network architecture
  - netflix.com hosted on own data centers (only registration, payment)
  - Main servers (e.g. movies.netflix.com) hosted by Amazon cloud services
  - Three CDNs used in parallel: Akamai, Limelight, Level-3
- Player based on Silverlight using DASH, transiting to HTML5
- Dynamic CDN selection strategy influences performance, see:
  - Adhiukari et al.: Unreeling Netflix: Understanding and Improving Multi-CDN Movie Delivery, IEEE INFOCOM 2012

Case Study: YouTube
Sources

- T. Hossfeld, R. Schatz, E. Biersack, and L. Plissonneau. Internet Video Delivery in YouTube: From Traffic Measurements to Quality of Experience. In Data Traffic Monitoring and Analysis, LNCS 7754, pages 266--303. Springer Verlag, Berlin Heidelberg, Germany, 2013.

- http://www.e-biersack.eu/BPublished/Youtubelncs.pdf
- http://peering.google.com/about/delivery_ecosystem.html

Case Study: YouTube, Delivery Ecosystem

- Google's network infrastructure has four distinct elements:
  - Data centers
  - Backbone
  - Edge Points of Presence (POPs), 70+ POPs in 33 countries
  - Google's edge caching infrastructure (Google Global Cache GGC)

Google Edge POPs

Google Data Centers

Case Study: YouTube, Caching

- Original content hosted within Google Data Centers
  – Replicated to multiple data centers for redundancy and efficiency
- Multi-tiered caching platform
  – Google servers within infrastructure of network operators and ISPs

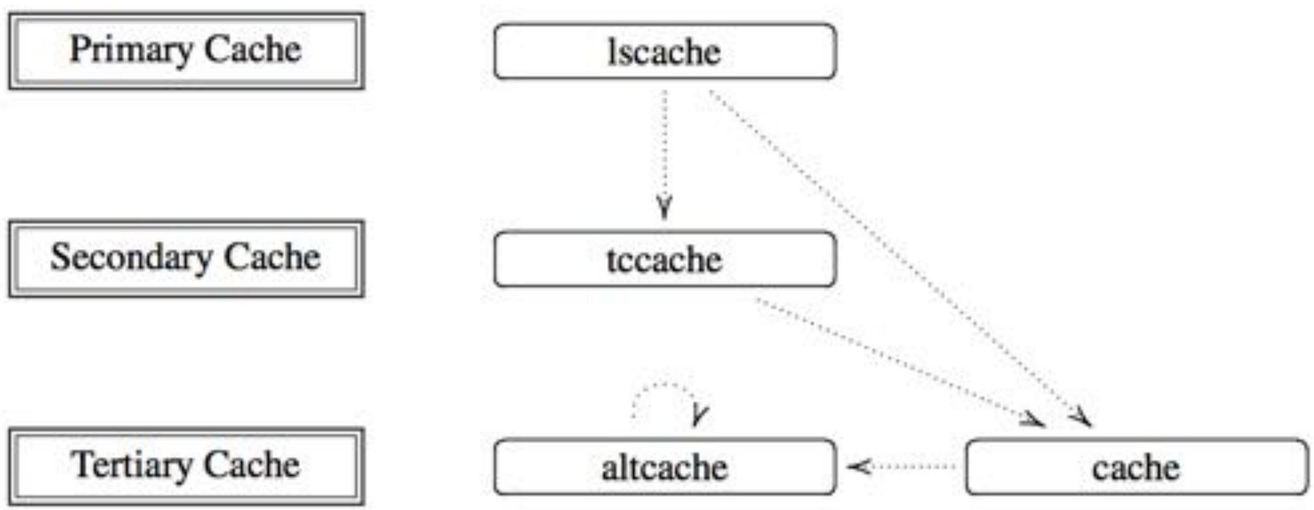


Fig. 2. Organization of the YouTube Caches; dashed lines indicate possible redirections

Google Data Centers