Recent Session Announcements:

Internet Media Guides (IMGs)

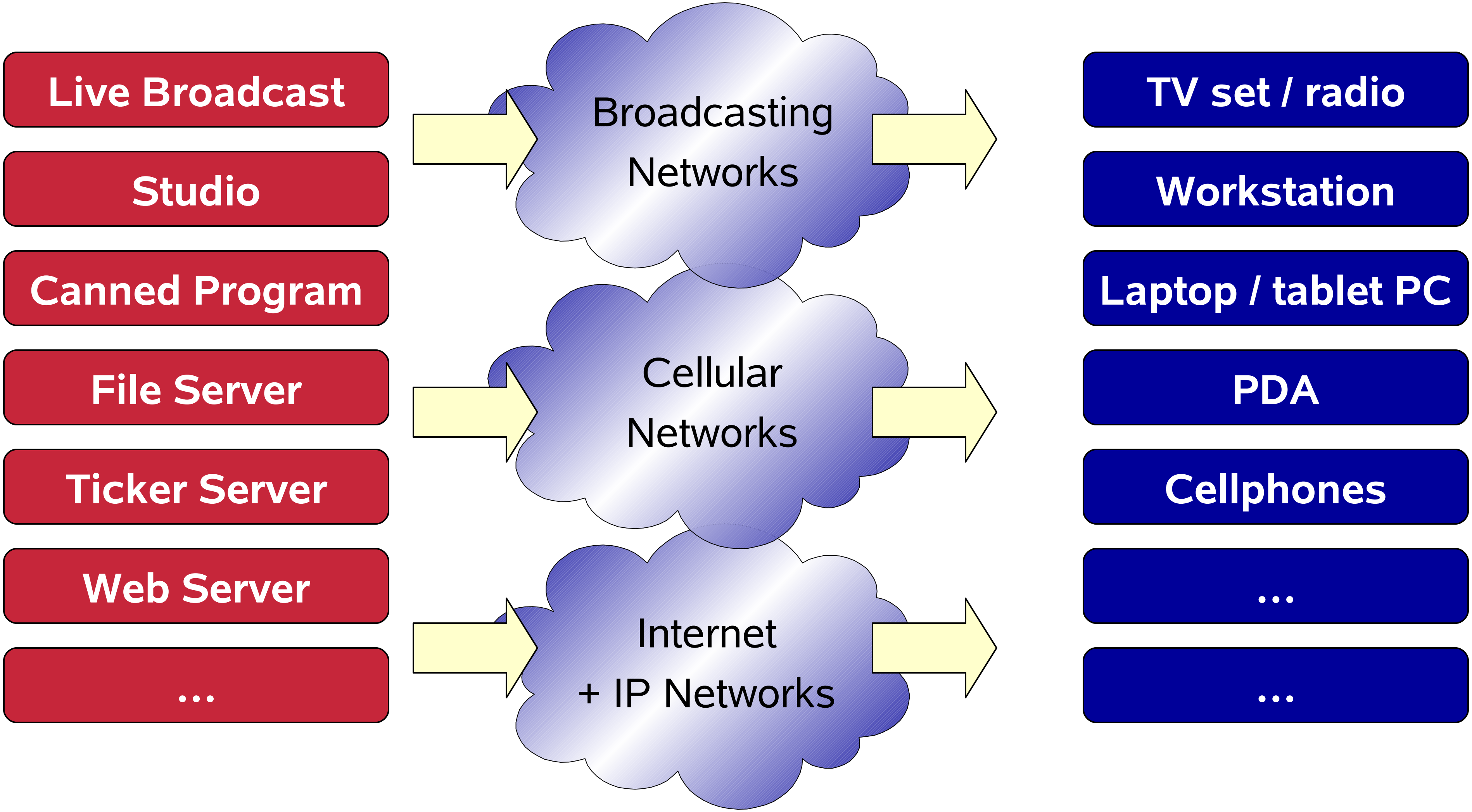
Observations

- SAP/SDP tied to IP-Multicast-based session model
- Only one distribution scheme: announcement
- Only one type of service: convey multimedia session information

- (Global) IP-Multicast has not prevailed as a distribution platform
- SAP rather experimental
- Was often used for debugging Mbone connectivity

- Summary
  - SAP/SDP too limited
  - Not appropriate as a general solution for distributing session information
  - Traditionally linked to IP-only (and Multicast-only)

Background: Ubiquitous Information Access



Live Broadcast

Studio

Canned Program

File Server

Ticker Server

Web Server

...

Broadcasting Networks

Cellular Networks

Internet + IP Networks

TV set / radio

Workstation

Laptop / tablet PC

PDA

Cellphones

...

...

“Classic” Broadcasting & Internet Multimedia

- Broadcasting has been a different world
  (including customer expectations, philosophy)
  - Encodings
    - Audio/Video largely compatible (but different quality expectations)
    - Image/text formats/HTML vs. Videotex, MHP, specific markups, tables
  - Data transmission
    - IP + UDP/TCP + RTP/… vs. MPEG multiplex (or even analog)
  - Addressing
    - IP addresses + ports vs. frequency/channel, PID, satellite position, pol., …
  - Interaction & control
    - RTSP, HTTP, SIP, … vs. MHP

- But there is a migration towards IP in various areas
  - Content providers, transmission technologies, consumer equipment

Platform/Network-Independent Content Provision

- The same content shall be available via different networks
  - Preferably without repeated authoring
- “Content” used in a broad sense
  - Original media: Audio / video broadcasts, web pages, files, news feeds, …
  - Supplementary information: background, statistics, subtitles, ads, …
- Content needs to be globally (or regionally) identifiable
- Content needs to be found
  - Descriptive metadata
  - Availability (scheduling) metadata
- Alternate access methods must be possible
  - Network + network-specific address

Internet Media Guides (IMG)

Definition of an IMG (from MMUSIC Charter)

Content:

- A **collection of multimedia session descriptions**
- Expressed using SDP, SDPng or other metadata formats
- It is used to describe a collection of multimedia sessions (e.g. television programme schedules).

Distribution:

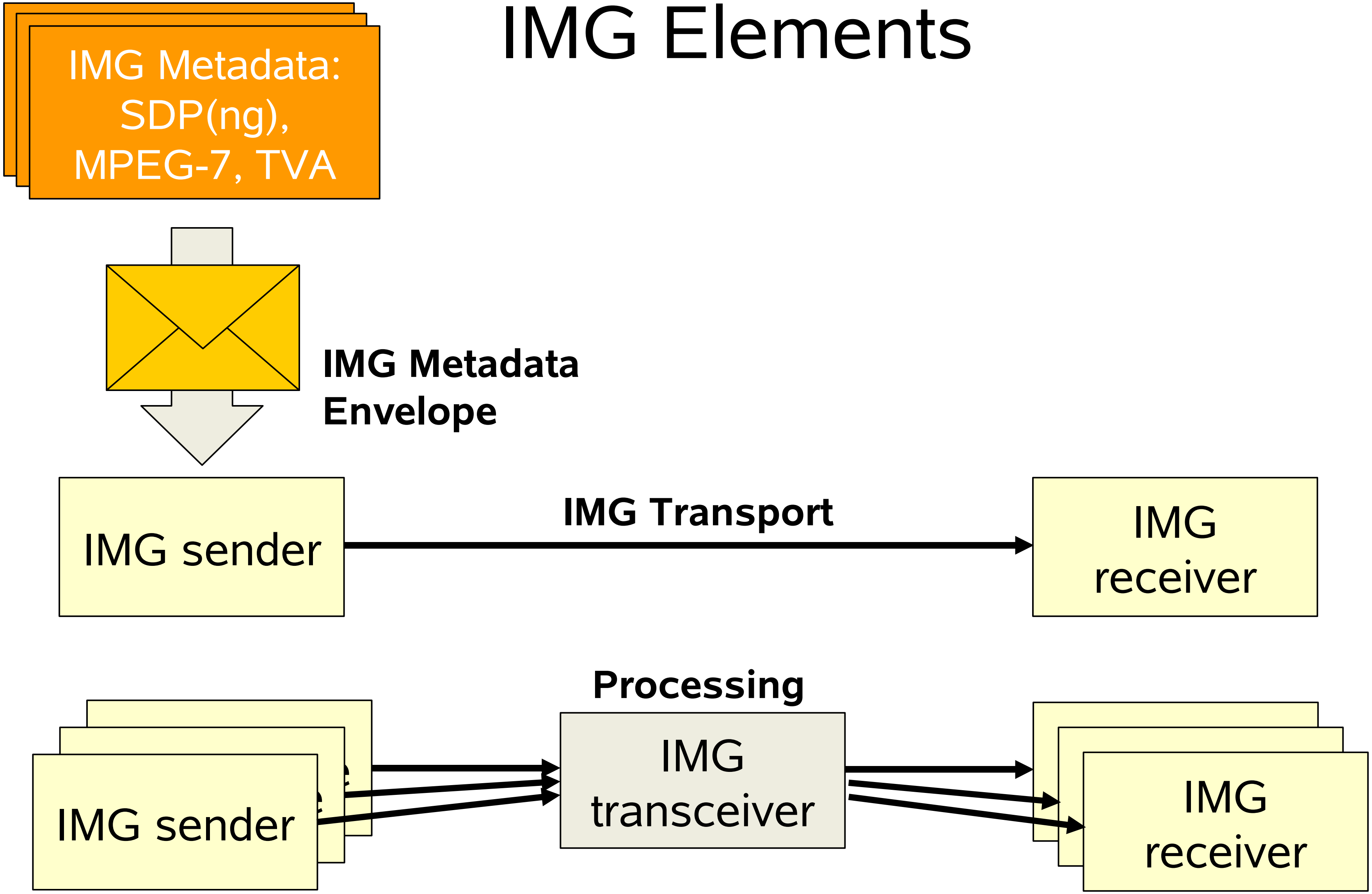
- The IMG must be **delivered to a potentially large audience (push or pull)**, who use it to join a subset of the sessions described, and who may need to be **notified of changes** to the IMG.

IMG ≈ EPG

- Generalized for arbitrary…
  - Types of media
  - Types of sessions and interactions: services!
  - Classes of devices
- Plurality of access methods
  - Physical delivery
  - (Reliable) Broadcast / multicast (push)
  - Interactive retrieval (pull)
  - Provision of full IMGs and of deltas
  - Notification about changes
- Network-independent
  - For the delivery of IMGs
  - For the (request and) transmission of actual media in sessions

The same IMGs should be usable everywhere.

IMG Elements



IMG Metadata: SDP(ng), MPEG-7, TVA

IMG Metadata Envelope

IMG sender

IMG Transport

IMG receiver

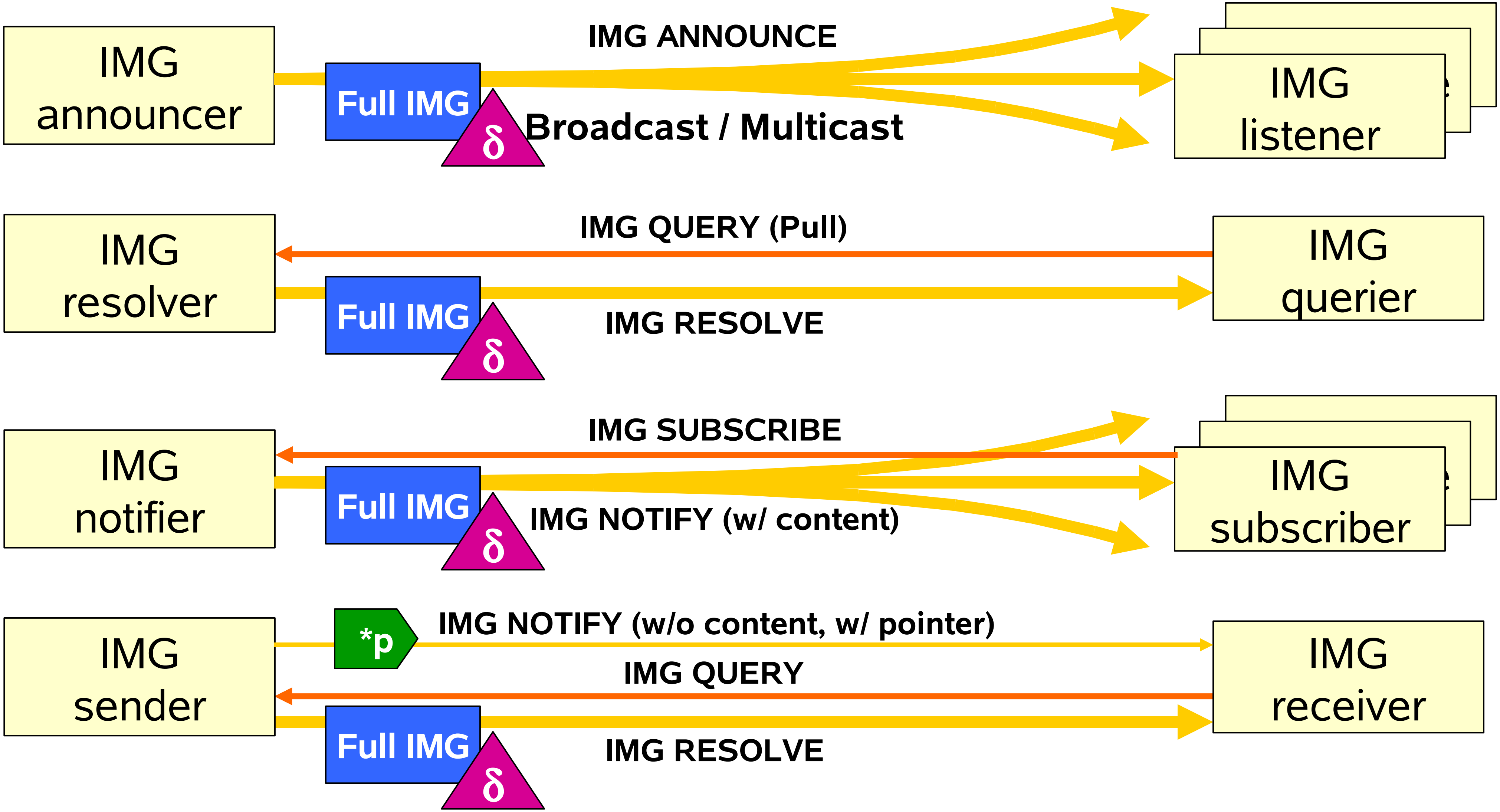
Processing

IMG sender

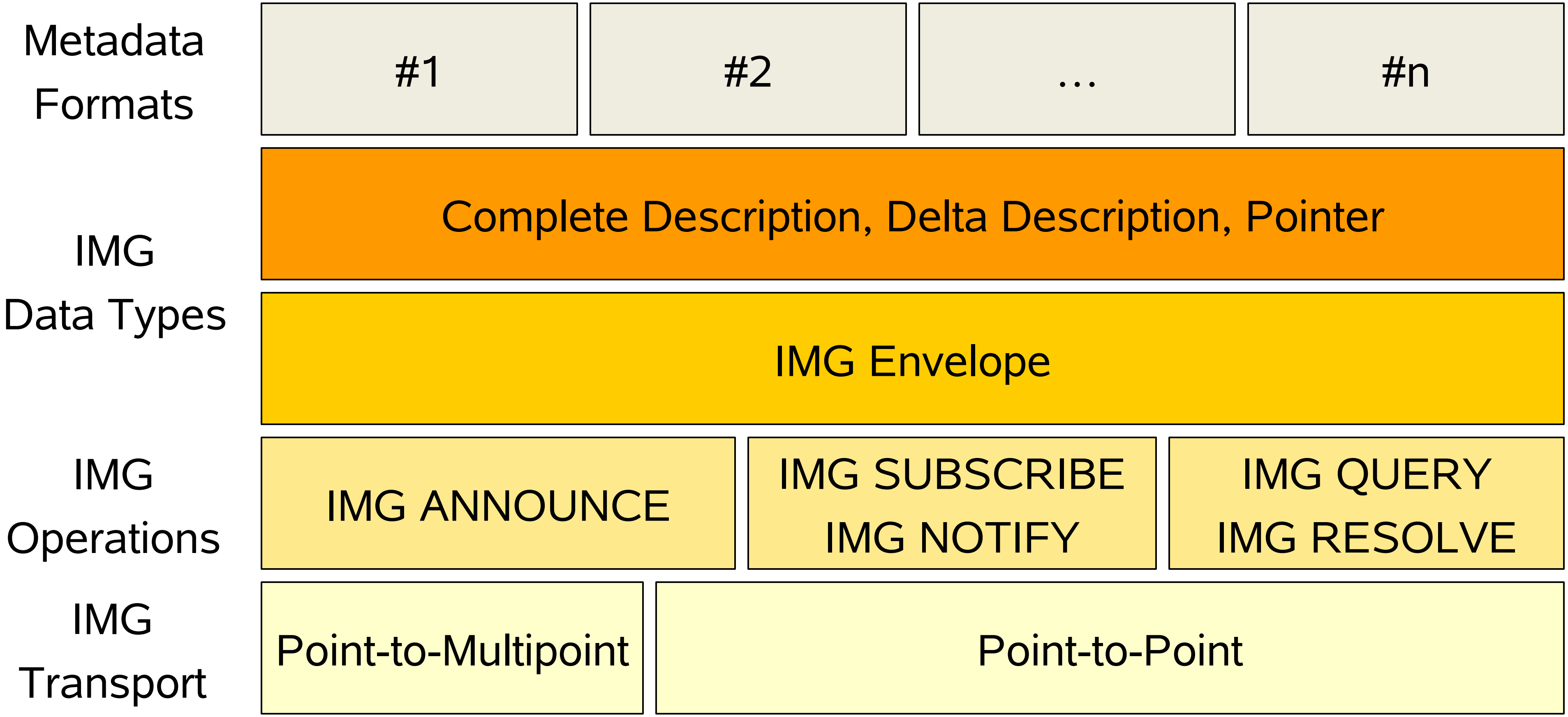
IMG transceiver

IMG receiver

IMG Delivery Models / Operations



IMG Architecture



Metadata Formats: #1 | #2 | … | #n

IMG Data Types: Complete Description, Delta Description, Pointer

IMG Envelope

IMG Operations: IMG ANNOUNCE | IMG SUBSCRIBE IMG NOTIFY | IMG QUERY IMG RESOLVE

IMG Transport: Point-to-Multipoint | Point-to-Point

IMG Envelope: Security Requirements

- Authentication + Integrity validation of contained metadata
  - Must work for complete and delta information
  - Must work across IMG transceivers
    - Aggregation, splitting, filtering of pieces of metadata
- Privacy
  - Must be able to protect (parts of) contained metadata
    - User protection + access control
  - Enable (limited) IMG transceiver functionality
- Interdependency with metadata formats
  - What to expect from metadata?
  - Granularity of embedded metadata objects
  - DRM? → metadata formats

IMG Envelope

- Container for metadata
  - Complete, delta, pointers
  - Independent of metadata
  - Likely to become some kind of wrapper mechanism
  - Metadata itself defined by other bodies

- Generic management information
  - Identification + version + validity information
  - Content-Type: to identify metadata format
  - Support for security?
    - authentication + integrity information
    - Privacy of content

MIME vs. XML

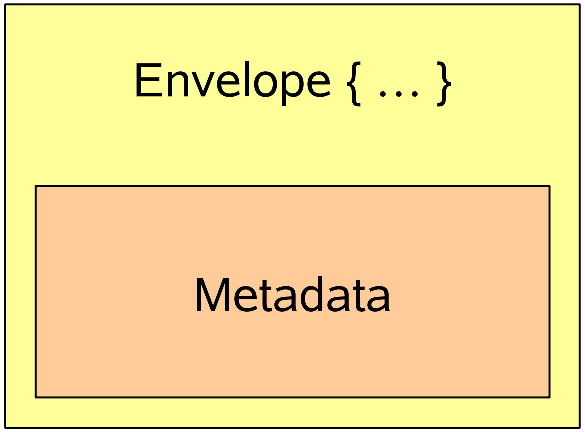
Envelope Features (1)

- Container for metadata (independent of these)
  - Complete, delta, pointers
  - Metadata itself defined by other bodies
- Version number
  - Determine the most recent (i.e., valid) copy
  - Referenced as basis for delta encoding
- Validity time
  - Period: from, to
- Metadata URI
  - Identifies the metadata element contained in the envelope
  - Helps to deal with fragments
- Content-Type
  - Defines the type of metadata contents

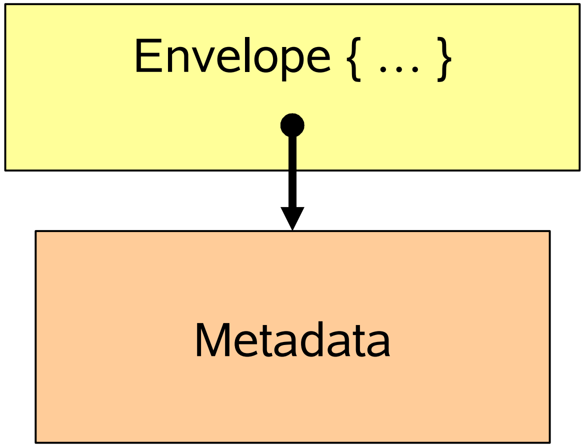
Envelope Features (2)

- Support for digital signatures (on parts of the envelope)
- Support for encryption
  - Only partly specified so far
  - May use S/MIME
- Metadata contents:

Inline



External (via pointer)



Envelope Encoding: XML vs. MIME

- **Present focus: XML (also used by 3GPP MBMS)**
- **Example (with SDP as metadata)**

```
<?xml version="1.0" encoding="UTF-8"?>
       <metadataEnvelope
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="envelope.xsd"
         metadataURI="http/www.example.com/img001/session001.sdp"
         version="1"
         validFrom="2003-12-17T09:30:47-05:00"
         validUntil="2003-12-17T09:30:47-05:00"
         contentType="application/sdp">
         <metadataFragment>
           v=0
           o=jo 2890844526 2890842807 IN IP4 10.33.57.27
           s=SDP Seminar
           c=IN IP4 224.2.17.12/127
           t=2873397496 2873404696
           a=recvonly
           m=audio 49170 RTP/AVP 0
           m=video 51372 RTP/AVP 31
         </metadataFragment>
       </metadataEnvelope>
```

IMG Metadata

- Past focus on traditional contents
  - Conveying plain TV-schedules
  - Streaming in 3GPP Release 6
- Broadening the scope
  - Cover services in a more general fashion
  - Provide region/location information
  - Support personalized inquiries
  - Address issues of cost
    - Make offers automatically comparable
- Technical level: enable service discovery (and location)
- Business level: support adequate service selection

IMG URN

- IMGs need to be identified globally
  - In particular, across different networks and providers
- Motivates the use of IMG URNs
- Format

  urn:img: ProviderId : DateId : IMGResourceId [: FragmentId]
  - ProviderId: domain name
  - DateId: Point in time when the domain name was owned by the entity
  - IMGResourceId: provider-selected string
  - FragmentId: some identifier for a piece of an IMG
- Examples
  - urn:img:example.org:20051021:my-img
  - urn:img:example.org:20051021:my-img:subset
- Mapping to URIs (e.g., HTTP, SIP) to be defined

IMG Transports

- Need to provide mechanisms for IMG Operations
- ANNOUNCE
  - Reliable multicast transport protocol: FLUTE + MUPPET
- SUBSCRIBE / NOTIFY
  - Session Initiation Protocol (SIP): Extensions for Subscription/Notification
- QUERY / RESOLVE
  - HTTP
- Identify IMGs properly across protocols: IMG URN (yet tbd.)
  - Mappings to individual protocols for actual processing

IMG ANNOUNCE: Reliable Multicast

- Layered Coding Transport (LCT)
  - Single sender multicast transport
  - Defines single or multi-object delivery across an LCT session
    - Provides identifiers for objects (TOI)
    - Provides session identification (TSI)
  - LCT session comprises a group of channels
    - Each identified by the respective (multicast) transport address
- Forward Error Correction (FEC)
  - General container for various FEC schemes
  - Alows to identify payload + provides in-band signaling of FEC parameters
- Asynchronous Layered Coding (ALC)
  - Simple combination of LCT and FEC

IMG ANNOUNCE: FLUTE Basics

- File Delivery over Unidirectional Transport
- Uses ALC (= LCT + FEC)
  - Fixed parameter sets for the protocol instantiation
- Specifies semantics of objects
  - Files
  - File Delivery Table (FDT)
- FDT
  - XML-based format to carry file attributes (name, location, size, etc.)
    - Carried as Transport Object ID = 0
  - Transmitted in a carrousel style together with files

IMG ANNOUNCE: FLUTE FDT

- XML-based structured information

- Example

```
<FDT-Payload Expires="<date>" complete="true">
    <File
                    Content-Location=
                    TOI=
                    Content-Length=
                    Transfer-Length=
                    Content-Type=
                    Content-Encoding=
                    Content-MD5=
                    ... plus some FEC stuff ... >
    <File ...>
    ...
</FDT-Payload>
```

IMG ANNOUNCE: MUPPET

- Specific usage of FLUTE for carrying IMG envelopes
- Defines various lower layer parameters
- Defines usage of multiple layers

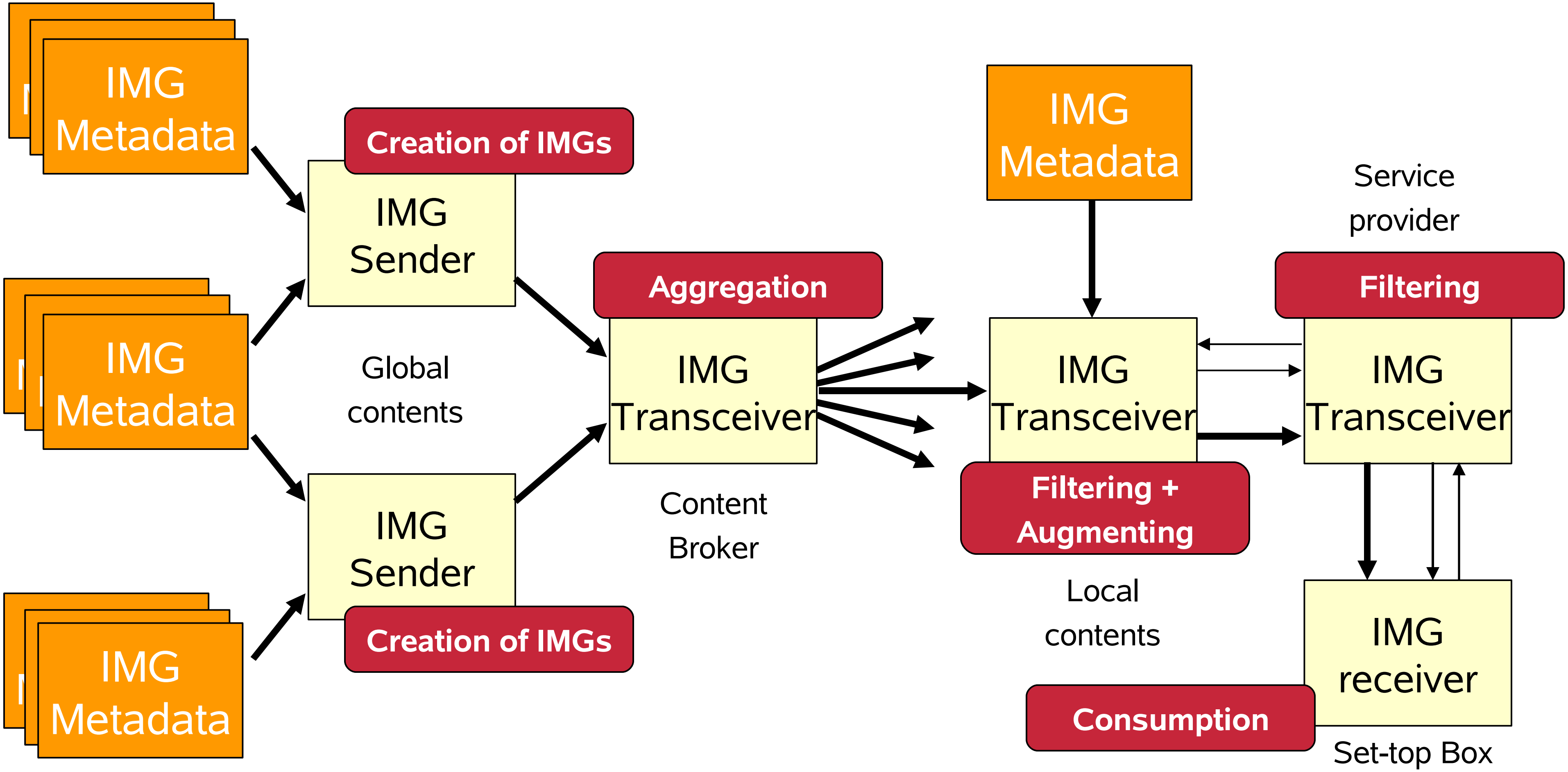
IMG QUERY / RESOLVE

- “Naturally” maps to HTTP GET + 200 OK
- HTTP URI: http://<hostname>/<resource>?param1&param2&…
  - Parameters identify IMG version
    - type: full or delta IMG, pointer
    - version requested
    - diffVersion: base for delta IMG
- Querier response format selection
  - Accept: application/img-envelope+xml
    - Provide IMG in envelope format
  - Accept: text/plain, text/html
    - Provide a human-readable description of an IMG as optional fallback
  - Allow for directly returning the plain metadata without envelope?
- 200 OK carries response in body
- HTTP headers used accordingly

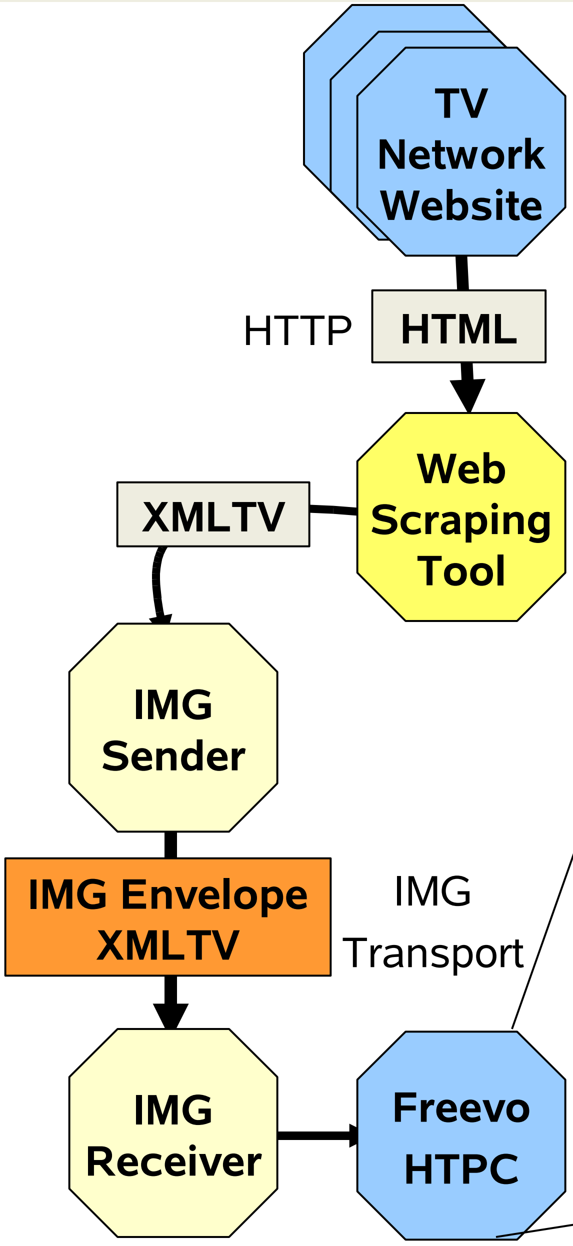
IMG SUBSCRIBE / NOTIFY

- Based upon the Session Initiation Protocol (SIP)
  - Particularly its SUBSCRIBE / NOTIFY mechanism
  - Details to be discussed
- SUBSCRIBE / NOTIFY
  - Register interest in (part of) an IMG
  - Receive an immediate response and updates upon changes
  - Soft-state based: subscription times out and needs refreshing
- IMG usage of SIP SUBSCRIBE / NOTIFY
  - Define SIP event package: img
  - Presently suggests a MIME-based IMG envelope
    - Natural choice for SIP
  - Content-Type:, Content-Location:
  - Content-ID: major.minor, Expires: valid-until

Regionalization & Personalization with IMGs



TV-EPG Distribution



TV Network Website

HTTP

HTML

Web Scraping Tool

XMLTV

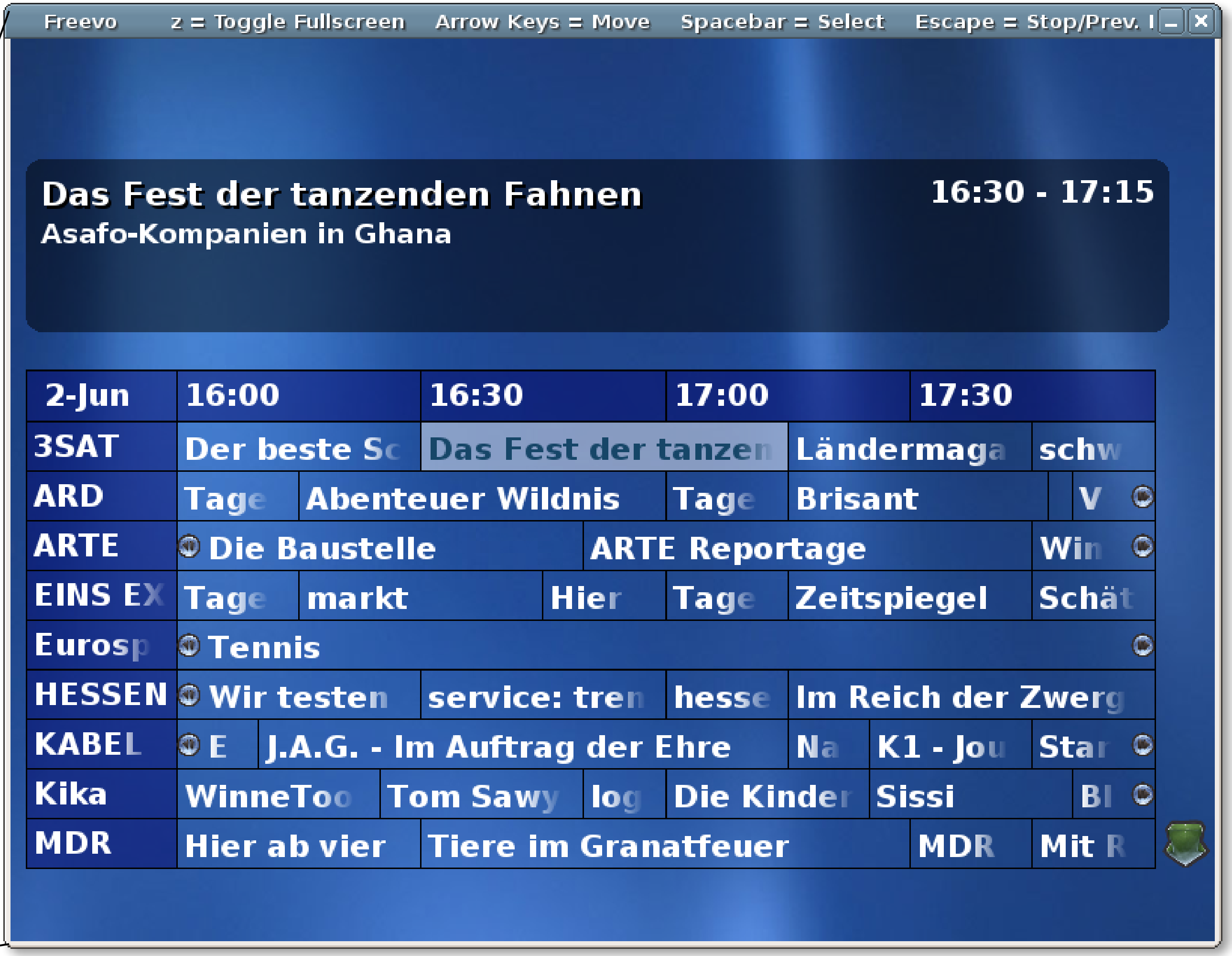
IMG Sender

IMG Envelope XMLTV

IMG Transport

IMG Receiver

Freevo HTPC



Freevo   z = Toggle Fullscreen   Arrow Keys = Move   Spacebar = Select   Escape = Stop/Prev. I

Das Fest der tanzenden Fahnen 16:30 - 17:15
Asafo-Kompanien in Ghana

2-Jun	16:00	16:30	17:00	17:30
3SAT	Der beste Sc	Das Fest der tanzen	Ländermaga	schw
ARD	Tage Abenteuer Wildnis		Tage Brisant	V
ARTE	Die Baustelle	ARTE Reportage		Win
EINS EX	Tage markt	Hier	Tage Zeitspiegel	Schät
Eurosp	Tennis			
HESSEN	Wir testen	service: tren	hesse Im Reich der Zwerg	
KABEL	E J.A.G. - Im Auftrag der Ehre		Na K1 - Jou	Star
Kika	WinneToo Tom Sawy	log	Die Kinder Sissi	Bl
MDR	Hier ab vier	Tiere im Granatfeuer		MDR Mit R

IMGs: “Final” Remarks

- Content formats: various
  - Simple tables in DVB/MPEG (backwards compatibility)
  - XML-based data sets for IMGs
- IMGs in use in 3GPP MBMS
- Stalled in the IETF some years ago (further work abandoned)
- TV industry going various other ways
  - Specific EPGs in DVB
  - TV Anytime forum
  - Web/RSS-based program pages of TV magazines and broadcasters
  - Open source platforms use yet other formats
  - XMLTV

“HTTP Streaming”

- Tunneling media and control in an HTTP connection

- Simplest case
  - Start replay before download is complete
  - No extensions needed
  - Mainly client-side operation
  - But: server needs to use appropriate media file format

- Alternative: add additional headers (MS)
  - Preserve packetization of media within a TCP connection

Old(?) MS HTTP Streaming Format

```
       0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+00   |             Type              |         Chunk Length          |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+04   |                        Sequence Number                        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+08   |      Flags (Unknown)          |     Chunk Length (again?)     |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+12   |                                                               |
      :                        Chunk Data Block                       :
      |                                                               |
      +                                                               +
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Sample Request Header (1/2)

```
GET test.asf HTTP/1.0
Accept: */*
User-Agent: NSPlayer/4.1.0.3856“
Host: media_host
Pragma: no-cache,rate=1.000000,stream-time=0,stream-offset=0:0,
                request-context=1,max-duration=0
Pragma: xClientGUID={c77e7400-738a-11d2-9add-0020af0a3278}
Connection: Close
```

Sample Request Header (2/2)

```
GET test.asf HTTP/1.0
Accept: */*
User-Agent: NSPlayer/4.1.0.3856
Host: media_host
Pragma: no-cache,rate=1.000000,stream-time=0,
        stream-offset=0:0,request-context=2,max-duration=40“
Pragma: xPlayStrm=1
Pragma: xClientGUID={c77e7400-738a-11d2-9add-0020af0a3278}
Pragma: stream-switch-count=1
Pragma: stream-switch-entry=ffff:1:0
Connection: Close
```

Sample Response Header

```
HTTP/1.1 200 OK
Content-Type: application/octet-stream
Server: Cougar 4.1.0.3920
Cache-Control: no-cache
Pragma: no-cache
Pragma: features=“broadcast"
```

Another Example: HTTP GET

```
GET /media/Videos/200710/200710A0/29102007002.mp4 HTTP/1.1
Content-length: 0
User-Agent: Java/1.5.0_10
Host: 192.168.1.100:50 004
Accept: video/mp4, text/html, image/gif, image/jpeg, *; q=.2, */*; q=. 2
Connection: keep-alive
```

Another Example: 200 OK

HTTP/1.1 200 OK
CONTENT-TYPE: video/mp4
CONTENT-LENGTH: 7667062

```
0040    0d 0a 0d 0a 00 00 00 1c 66 74 79 70 6d 70 34  ........ftypmp4
0050 32 00 00 00 00 6d 70 34 32 33 67 70 34 69 73 6f 2....mp423gp4iso
0060 6d 00 74 b2 13 6d 64 61 74 00 00 18 83 f2 1b fb m.t..mdat.......
0070 04 29 69 69 69 69 69 69 69 69 69 69 69 69 69 69 .)iiiiiiiiiiiiii
0080 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 iiiiiiiiiiiiiiii
0090 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 iiiiiiiiiiiiiiii
00a0 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 iiiiiiiiiiiiiiii
00b0 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 iiiiiiiiiiiiiiii
00c0 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 iiiiiiiiiiiiiiii
00d0 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 iiiiiiiiiiiiiiii
00e0 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 iiiiiiiiiiiiiiii
00f0 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 iiiiiiiiiiiiiiii
0100 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 69 iiiiiiiiiiiiiiii
```

Home Media Streaming Architectures

- No single coherent solution at this point
  - Different camps follow different approaches
  - Apple vs. other industry consortia vs. (operator) standardization bodies vs. …
- But architectural similarities
  - Devices need to support zero/autoconfiguration
    - No need for any kind of setup interaction ("plug and play")
  - Devices need to be able to find one another
  - Devices need to determine each others' capabilities
    - Self-descriptions
  - Devices need to discover resources available on/from another devices
  - Devices need to engage in communications and deliver media streams
- Example: DLNA: Digital Living Network Alliance
  - Design for small scale and closed deployments (home networks)
  - Uses Universal Plug and Play (UPnP) and UPnP AV

Example: DLNA

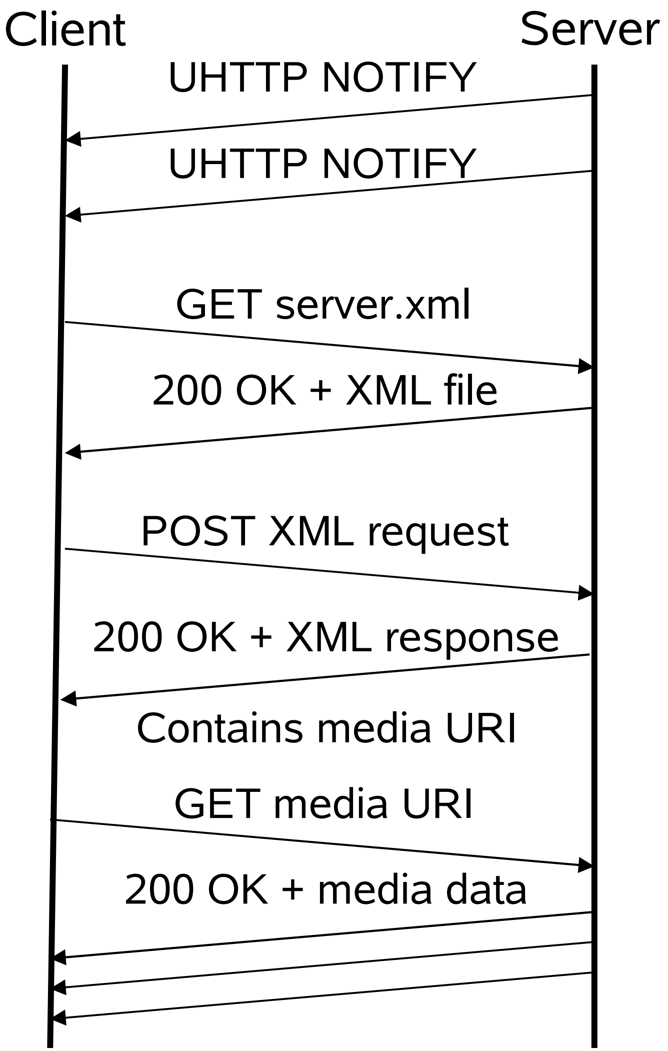
- Autoconfiguration
  - DHCP (if a DHCP server is present)
  - Zero configuration: IPv4 link local address configuration (RFC 3927)

- Device discovery: UPnP
  - Based upon something called “UHTTP”: HTTP syntax over UDP packets
  - Sent in regular intervals (no scaling as with RTCP or SAP!)

```
NOTIFY * HTTP/1.1
LOCATION: http://192.168.1.100:50004/MediaServer1/MediaServer1.xml
HOST: 239.255.255.250:1900
SERVER: Symbian/9.2 UPnP/1.0 Nokia/N95
NTS: ssdp:alive
USN: uuid:d8c66d26-1b20-10e1-9c90-001CD45CCA96
CACHE-CONTROL: max-age=1800
NT: uuid:d8c66d26-1b20-10e1-9c90-001CD45CCA96
```

Example: DLNA

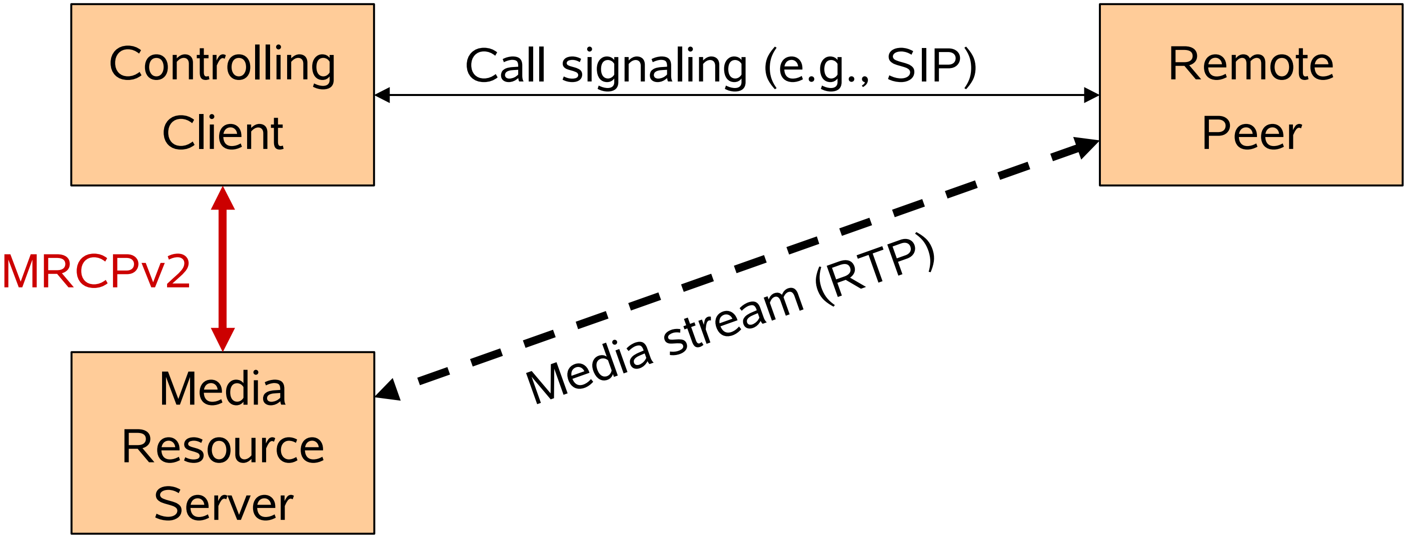
- Device capability assessment
  - HTTP-based (over TCP) query-response protocol
  - Simple Service Discovery Protocol (SSDP)
  - Retrieval of an XML-based service description
- Resource discovery
  - Based upon the device capabilities (e.g., media server profile)
  - Simple Object Access Protocol (SOAP) RPC
    - XML-encoded synchronous RPCs carried over HTTP
  - Example: get a "directory listing"
    - Using naming conventions for folders to locate contents
    - Yields URIs to access each individual media resource
- Media streaming
  - HTTP streaming: GET on the URI of the media resource



Third party media resource control

Media Resource Control Protocol (MRCPv2)

- Another protocol to control media resources
  - Based upon a proprietary version by Cisco et al. (MRCPv1, RFC 4443)
- Enable a client to task a third entity to perform on its behalf
  - Media stream generation (basic and advanced speech synthesis)
  - Media processing (recording, DTMF/speech recognition, speaker verification)



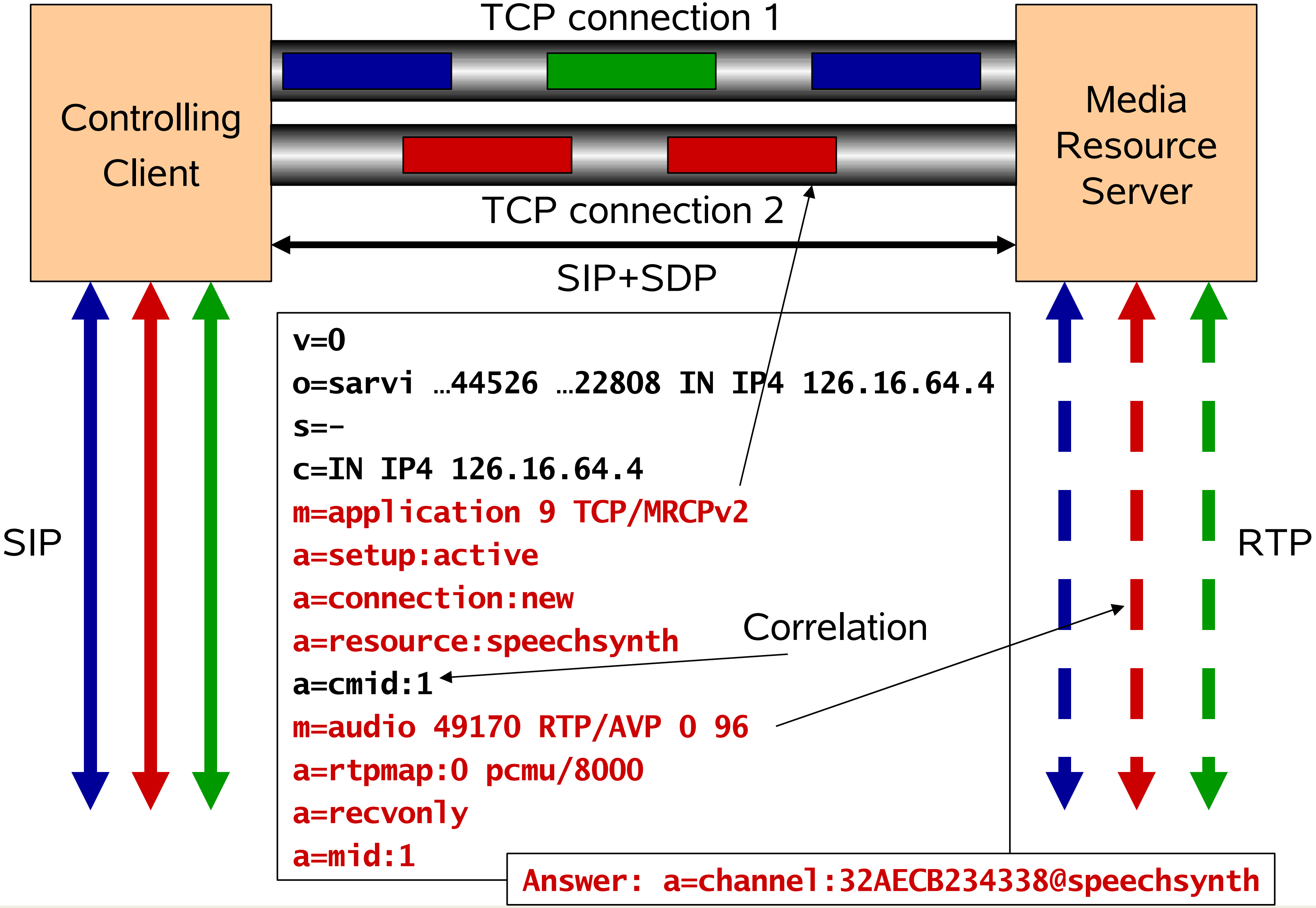
MRCPv2 Overview (1)

- MRCPv2 defines a common framework for rather different application classes
- Commonalities
  - Media stream consumption or generation by a media resource server
  - Control of the media stream generation or processing by the client
  - Report on media stream contents, characteristics, and resource server status
- Text-based protocol
  - Start line + headers + message body
  - Borrows heavily from HTTP and RTSP
  - Yet, subtle differences (later)
  - Message bodies identified by entity headers (using MIME types, etc.)
- Symmetric operation
  - Both peers can initiate actions: Methods (client->server), Events (server->client)
  - Headers + contents to parameterize operations or deliver results

MRCPv2 Overview (2)

- Uses TCP as underlying transport (+ optional TLS)
  - Reliability required; limited real-time interaction requirements only (true?)
    - Or do we assume sufficiently well interconnected clients and media resources
  - One of more TCP connections multiplexed
    - Concept of logical channels
- Uses RTP for media streams
  - Explicit correlation to TCP control channels in SDP using new grouping
- Relies on SDP offer/answer (using SIP) for session setup
  - Connection-oriented media (TCP, TLS) as well as RTP sessions

MRCP Overview (2)



```
v=0
o=sarvi …44526 …22808 IN IP4 126.16.64.4
s=-
c=IN IP4 126.16.64.4
m=application 9 TCP/MRCPv2
a=setup:active
a=connection:new
a=resource:speechsynth
a=cmid:1
m=audio 49170 RTP/AVP 0 96
a=rtpmap:0 pcmu/8000
a=recvonly
a=mid:1
```

```
Answer: a=channel:32AECB234338@speechsynth
```

MRCP Packages

- Different command sets defined for different packets
  - Building upon a small common subset of protocol elements
  - Otherwise largely independent of one another
  - Methods and events, response codes
  - Header fields
  - Content types (references to externally defined content formats)
- One package type per application
  - Speech Recognition
  - DTMF Recognition
  - Basic synthesis
  - Speech synthesis
  - Speaker verification
  - Recording
- Highly specialized for the specific application domain
  - You wonder why all this stuff goes into a single spec

Simple Example: Recording (1)

- Methods
  - RECORD — start recording
  - STOP — stop recording
  - START-INPUT-TIMERS — configuration
- Events
  - START-OF-INPUT — media stream recording has begun
  - RECORD-COMPLETE — recording done
- Some useful headers
  - Sensitivity-Level — for silence suppression
  - Media-Type — what to record
  - Record-URI — where to store recording
  - Trim-Length — limit length of recording
  - Capture-on-Speech — wait for speech
  - Various timeouts for input sensing, end of recording, …
- Message bodies
  - Captured recording (unless stored at a URI)

Simple Example: Recording (2)

```
C->S:   MRCP/2.0 386 RECORD 543257
        Channel-Identifier:32AECB23433802@recorder
        Record-URI:<file://mediaserver/recordings/myfile.wav>
        Capture-On-Speech:true
        Final-Silence:300
        Max-Time:6000

 S->C:  MRCP/2.0 48 456234 200 IN-PROGRESS
        Channel-Identifier:32AECB23433802@recorder

 S->C:  MRCP/2/0 49 START-OF-INPUT 456234 IN-PROGRESS
        Channel-Identifier:32AECB23433802@recorder

 S->C:  MRCP/2.0 54 RECORD-COMPLETE 456234 COMPLETE
        Channel-Identifier:32AECB23433802@recorder
        Completion-Cause:000 success-silence
        Record-URI:<file://mediaserver/recordings/myfile.wav>;
              size=242552;duration=25645
```

More Media Control

- Media Gateway Control Protocol (MEGACOP)
  - Configuring (PSTN) media gateways for IP telephony
  - Controlling media resource functions in 3GPP
- Media Server Control Markup Language and Protocol
  - Controlling conference servers
  - Controlling Interactive Voice Response (IVR) systems
- MEDIACTL WG in the IETF (newly created last week)
- Lots of non-IETF work (e.g., W3C)
- Gains importance in the context of service creation for interpersonal communications (using SIP)