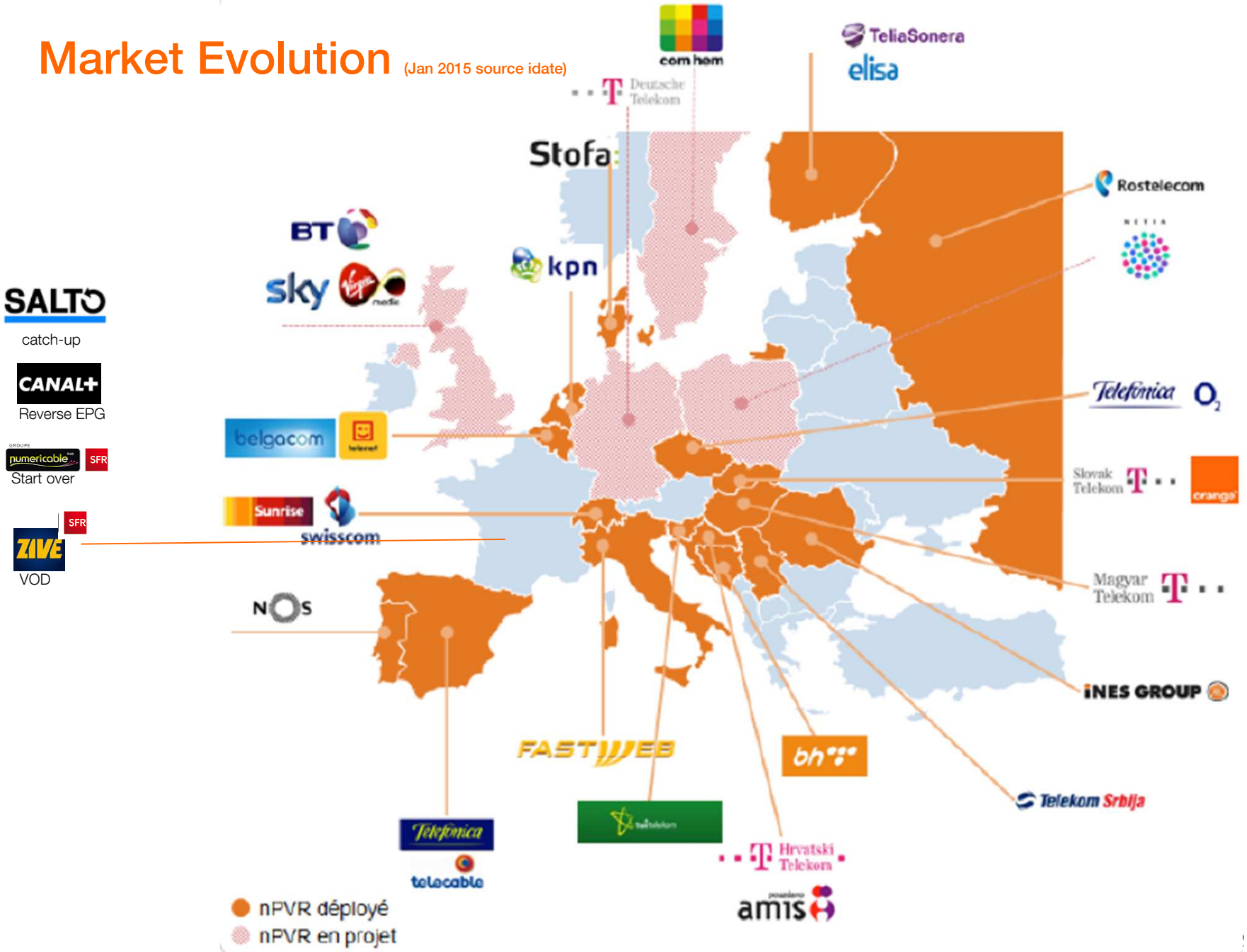
VSSP
Presentation

08th February 2017
-version 1.0

SKC & CPM on VSPP Solution

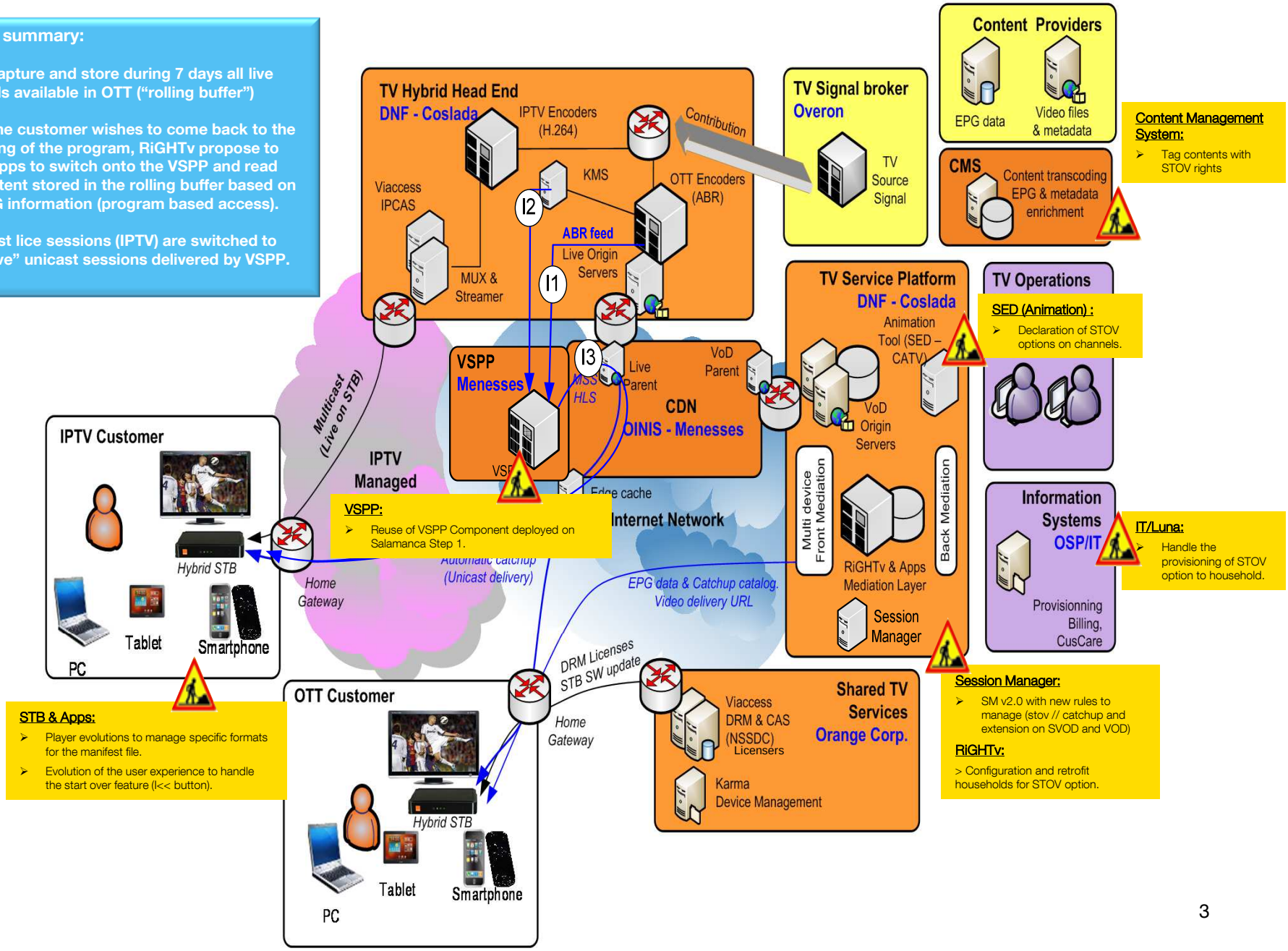
Market Evolution (Jan 2015 source idate)



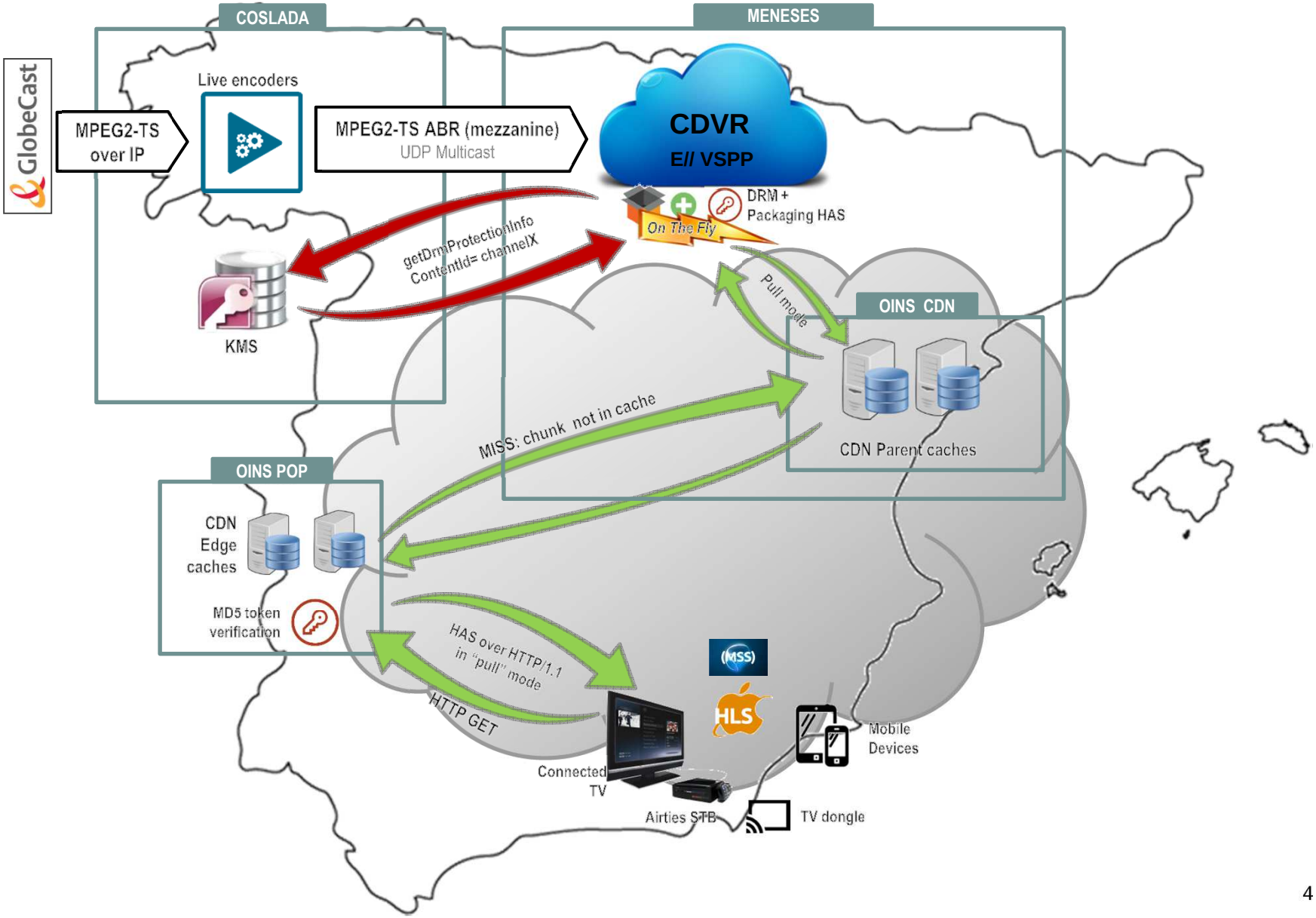
Solution overview & high level impacts

Solution summary:

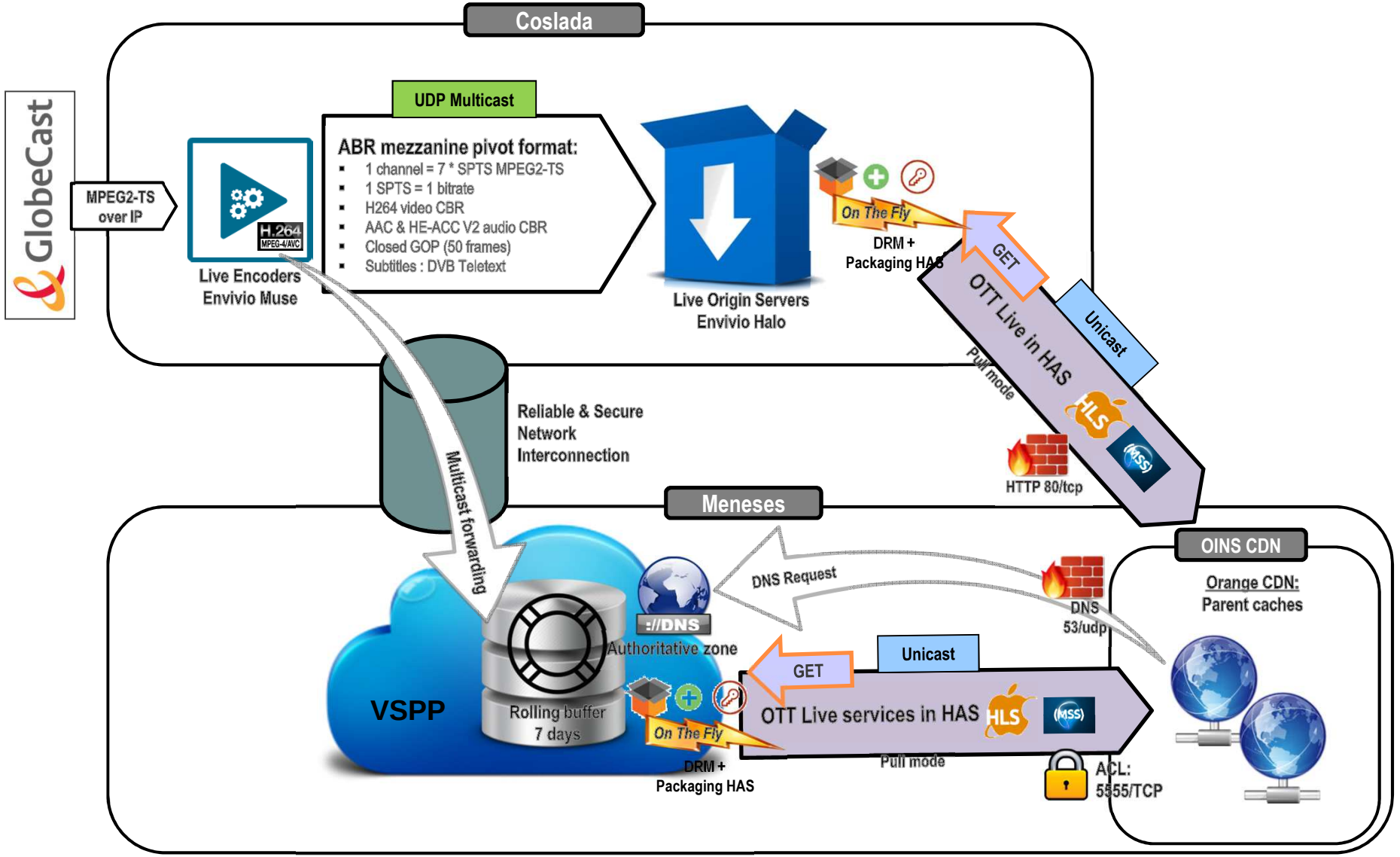
- VSPP capture and store during 7 days all live channels available in OTT (“rolling buffer”)
- When the customer wishes to come back to the beginning of the program, RiGHTv propose to STB / Apps to switch onto the VSPP and read the content stored in the rolling buffer based on the EPG information (program based access).
- Multicast lice sessions (IPTV) are switched to “near live” unicast sessions delivered by VSPP.



Global overview



Network interconnection



Orange CDN integration

- VSPP is configured in “Live Mode” with a 7 days rolling buffer
- The Orange CDN has to retrieve the video and the audio contents (MSS & HLS) from the VSPP nodes by using the “Pull Mode” (HTTP GET method)
- VSPP nodes are acting as Live origin servers for all the HAS protocols: each of them can be accessed through their streaming IP address (EGRS VLAN)
- In order to select the best VSPP node for streaming, whatever the live channel or the HAS protocol, a dedicated authoritative DNS zone managed by the VSPP managers should be used (ex of internal FQDN: strm.poda.manager.cdrv.local)
- The “static” URL format defined by VSPP is predictable and every MSS fragments or HLS chunks can be cached by the Orange CDN
- Manifest file (MSS) and playlist (HLS) may be cached but only for 2 seconds: for the same URL, their content could be different on certain use case (Start-Over, NTC) and 2 seconds late could be acceptable

Supported HTTP features

- HTTP 1.1 (connection: keep-alive) must be enabled on the CDN to increase the network performance
- If the CDN specifies the “Range: bytes=X-Y” header in the request, VSPP will always provide the full content of the fragment with “HTTP 200 OK” and “Content-Length” equals to its total size
- The “HTTP Cache-Control: max-age=XXX” response header is managed by VSPP through a specific configuration parameter
- The “If-Modified-Since” request header is supported: HTTP code 304 will be returned if the fragment is still in the VSPP node cache. But, because the fragment remains unique, it must be used only if the fragment is requested again by a client and only after the “max-age” period has expired
- The “Last-Modified” response header is always “Thu, 01 Jan 1970 00:00:00 GMT”.

Live Smooth Streaming

- Playout URL received by the CDN:

```
http://{cdvr_mss_fqdn}/{token_data}/shss/LIVE${channel_id}/{fragment_length}.ism/
manifest?start=LIVE&end=END&device={device_profile}
```

```
cdvr_mss_fqdn: CDVR FQDN for MSS(ex: cdvr-mss.tv.orange.es)
token_data: Verivue data = "token/expiration time/additional data"
channel_id: id of the live channel (ex: CNN, FOX)
fragment_length: length of the HSS fragment in seconds (Default: 2 s)
device_profile: name of the device profile for MSS (ex: MSS_LOW, MSS_HIGH)
```

- URL rewritten by CDN and requested to VSPP:

```
http://{vspp_fqdn}:{tcp_port}/shss/LIVE${channel_id}/{fragment_length}.ism/
manifest?start=LIVE&end=END&device={device profile}
```

```
vspp_fqdn: FQDN of VSPP (ex: strm.poda.manager.cdvr.internal)
tcp_port: 5555/tcp port used by VSPP for streaming
channel_id: id of the live channel (ex: CNN, FOX)
fragment_length: length of the HSS fragment in seconds (Default: 2 s)
device_profile: name of the device profile for MSS (ex: MSS_LOW, MSS_HIGH)
```

- Example:

```
http://cdvr-mss.tv.orange.es/6Vp8kZhdgEKhULIuvcCaWA/0000000000/null/null/null/shss/
LIVE$FOX/2.ism/manifest?start=LIVE&end=END&device=MSS_LOW
```

```
http://strm.poda.manager.cdvr.internal:5555/shss/LIVE$FOX/2.ism/manifest?
start=LIVE&end=END&device=MSS_LOW
```

Live HLS

- Playout URL received by the CDN:

```
http://{cdvr_hls_fqdn}/{token_data}/shls/LIVE${channel_id}/{chunk_length}.m3u8?
start=LIVE&end=END&device={device_profile}
```

```
cdvr_hls_fqdn: CDVR FQDN for MSS(ex: cdvr-hls.tv.orange.es)
token_data: Verivue data = "token/expiration time/additional data"
channel_id: id of the live channel (ex: CNN, FOX)
chunk_length: length of the HLS chunk in seconds (Default: 10 s)
device_profile: name of the device profile for HLS (ex: HLS_LOW, HLS_HIGH)
```

- URL rewritten by CDN and requested to VSPP:

```
http://{vspp_fqdn}:{tcp_port}/shls/LIVE${channel_id}/{chunk_length}.m3u8?
start=LIVE&end=END&device={device profile}
```

```
vspp_fqdn: FQDN of VSPP (ex: strm.poda.manager.cdvr.internal)
tcp_port: 5555/tcp port used by VSPP for streaming (default: 5555)
channel_id: id of the live channel (ex: CNN, FOX)
chunk_length: length of the HLS chunk in seconds (Default: 10 s)
device_profile: name of the device profile for HLS (ex: HLS_LOW, HLS_HIGH)
```

- Example:

```
http://cdvr-hls.tv.orange.es/6Vp8kZhdgEKhULIuvcCaWA/0000000000/null/null/null/shls/
LIVE$FOX/10.m3u8?start=LIVE&end=END&device=HLS_LOW
```

```
http://strm.poda.manager.cdvr.internal:5555/shls/LIVE$FOX/10.m3u8?start=LIVE&
end=END&device=HLS_LOW
```

Catch-up, Reverse EPG, Start-Over & NTC in MSS

- Player URL received by the CDN:

```
http://{cdn_mss_fqdn}/{token_data}/shss/LIVE${channel_id}/{fragment_length}.ism/
manifest?start={start_time}&end={end_time}&device={device_profile}
```

```
cdvr_mss_fqdn: CDVR FQDN for MSS (ex: cdvr-mss.tv.orange.es)
token_data: Verivue data = "token/expiration time/additional data"
channel_id: id of the live channel (ex: CNN, FOX)
fragment_length: length of the HSS fragment in seconds (Default: 2 s)
start_time: iso timestamp (UTC) ➔ start of the program (yyyy-MM-ddTHH:MM:SSZ)
end_time: iso timestamp (UTC) ➔ end of the program (yyyy-MM-ddTHH:MM:SSZ)
device_profile: name of the device profile for MSS (ex: MSS_LOW, MSS_HIGH)
```

- URL rewritten by CDN and requested to VSPP:

```
http://{vspp_fqdn}:{tcp_port}/shss/LIVE${channel_id}/{fragment_length}.ism/
manifest? start={start_time}&end={end_time}&device={device profile}
```

```
vspp_fqdn: FQDN of VSPP (ex: strm.poda.manager.cdvr.internal)
tcp_port: 5555/tcp port used by VSPP for streaming
channel_id: id of the live channel (ex: CNN, FOX)
fragment_length: length of the HSS fragment in seconds (Default: 2 s)
start_time: iso timestamp (UTC) ➔ start of the program (yyyy-MM-ddTHH:MM:SSZ)
end_time: iso timestamp (UTC) ➔ end of the program (yyyy-MM-ddTHH:MM:SSZ)
device_profile: name of the device profile for MSS (ex: MSS_LOW, MSS_HIGH)
```

- Example (FOX: 1 hour program on 2016-Mar-03 at 10am GMT):

```
http://cdvr-mss.tv.orange.es/6Vp8kZhdgEKhULIuvcCaWA/0000000000/null/null/null/shss/
LIVE$FOX/2.ism/manifest?start=2016-03-04T10:00:00Z&end=2016-03-04T11:00:00Z&device=MSS_LOW
```

```
http://strm.poda.manager.cdvr.internal:5555/shss/LIVE$FOX/2.ism/manifest?
start=2016-03-04T10:00:00Z&end=2016-03-04T11:00:00Z&device=MSS_LOW
```

Catch-up, Reverse EPG, Start-Over & NTC in HLS

- Playout URL received by the CDN:

```
http://{cdvr_hls_fqdn}/{token_data}/shls/LIVE${channel_id}/{chunk_length}.m3u8?
start={start_time}&end={end_time}&device={device_profile}
```

```
cdvr_hls_fqdn: CDVR FQDN for HLS (ex: cdvr-hls.tv.orange.es)
token_data: Verivue data = "token/expiration time/additional data"
channel_id: id of the live channel (ex: CNN, FOX)
chunk_length: length of the HLS chunk in seconds (Default: 10 s)
start_time: iso timestamp (UTC) ➔ start of the program (yyyy-MM-ddTHH:MM:SSZ)
end_time: iso timestamp (UTC) ➔ end of the program (yyyy-MM-ddTHH:MM:SSZ)
device_profile: name of the device profile for HLS (ex: HLS_LOW, HLS_HIGH)
```

- URL rewritten by CDN and requested to VSPP:

```
http://{vspp_fqdn}:{tcp_port}/shls/LIVE${channel_id}/{chunk_length}.m3u8?
start={start_time}&end={end_time}&device={device profile}
```

```
vspp_fqdn: FQDN of VSPP (ex: strm.poda.manager.cdvr.internal)
tcp_port: 5555/tcp port used by VSPP for streaming
channel_id: id of the live channel (ex: CNN, FOX)
chunk_length: length of the HLS chunk in seconds (Default: 10 s)
start_time: iso timestamp (UTC) ➔ start of the program (yyyy-MM-ddTHH:MM:SSZ)
end_time: iso timestamp (UTC) ➔ end of the program (yyyy-MM-ddTHH:MM:SSZ)
device_profile: name of the device profile for HLS (ex: HLS_LOW, HLS_HIGH)
```

- Example (FOX: 1 hour program on 2016-Mar-03 at 10am GMT):

```
http://cdvr-hls.tv.orange.es/6Vp8kZhdgEKhULIuvcCaWA/0000000000/null/null/null/shls/
LIVE$FOX/10.m3u8?start=2016-03-04T10:00:00Z&end=2016-03-04T11:00:00Z&device=HLS_LOW
```

```
http://strm.poda.manager.cdvr.internal:5555/shls/LIVE$FOX/10.m3u8?start=2016-03-
04T10:00:00Z&end=2016-03-04T11:00:00Z&device=HLS_LOW
```

Fragment URL with DRM protection

- Smooth Streaming URL protected by DRM:

```
http://{cdn_mss_fqdn}/{token_data}/shss/LIVE${channel_id}/{fragment_length}.ism/{DRM_profile}/
QualityLevels({bitrate_level})/{method}({track_id}={timestamp})
```

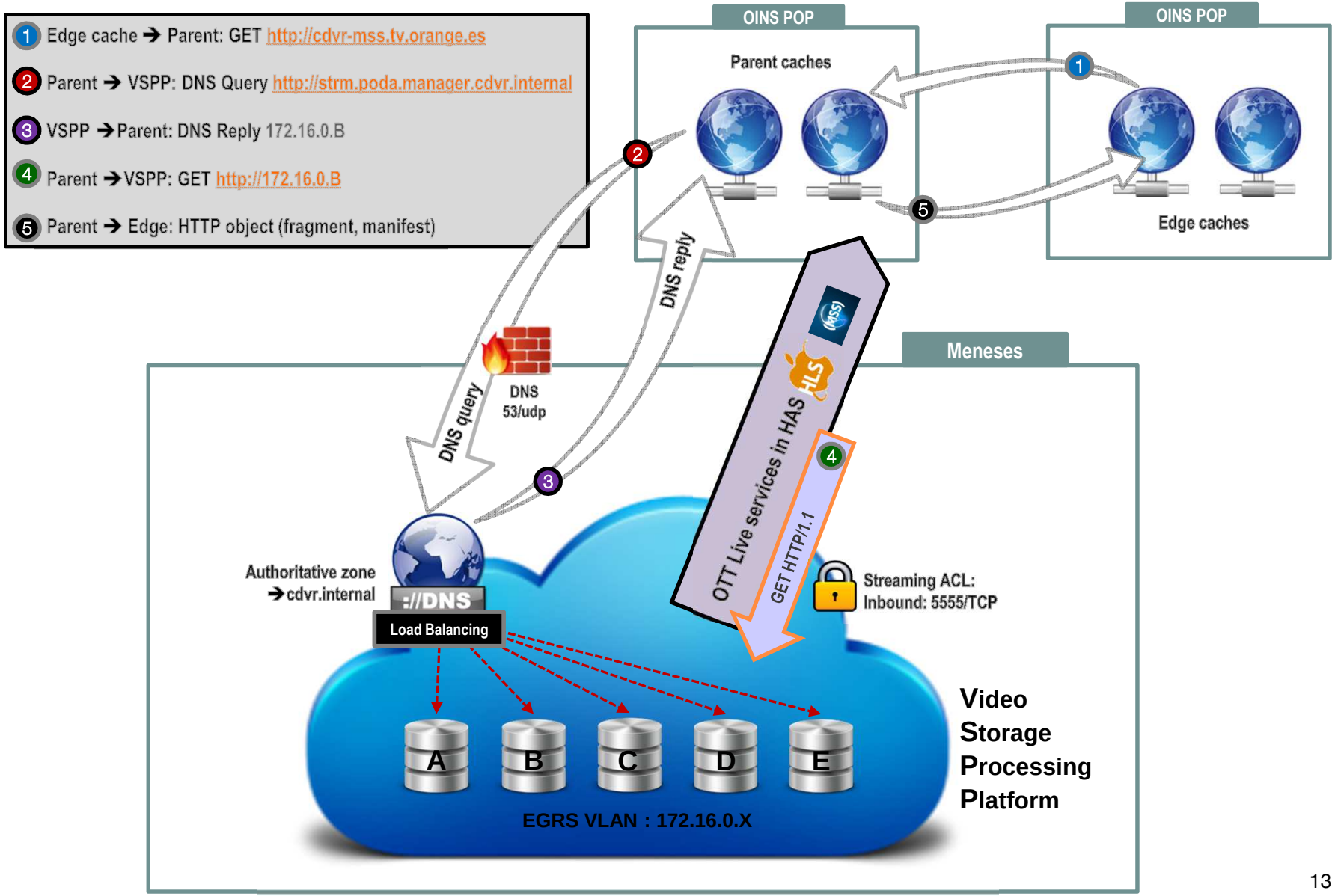
```
cdvr_mss_fqdn: CDVR FQDN for MSS (ex: cdvr-mss.tv.orange.es)
token_data: Verivue data = "token/expiration time/additional data"
channel_id: id of the live channel (ex: CNN, FOX)
fragment_length: length of the HSS fragment in seconds (Default: 2 seconds)
DRM_profile: name of the DRM profile defined in the VSPP configuration (ex: ORCA_MSS_DRM)
bitrate_level: bitrate level in b/s (ex: 680Kb/s ➔ 713031)
method: [fragments|fragmentInfo|KeyFrames]
track_id: name of the audio/video track (ex: video)
timestamp: timestamp in ms (UTC epoch time) (ex: Thu, 28 Apr 2016 13:54:12 GMT ➔ 1461851652000)
```

- URL of an HLS chunk protected by DRM:

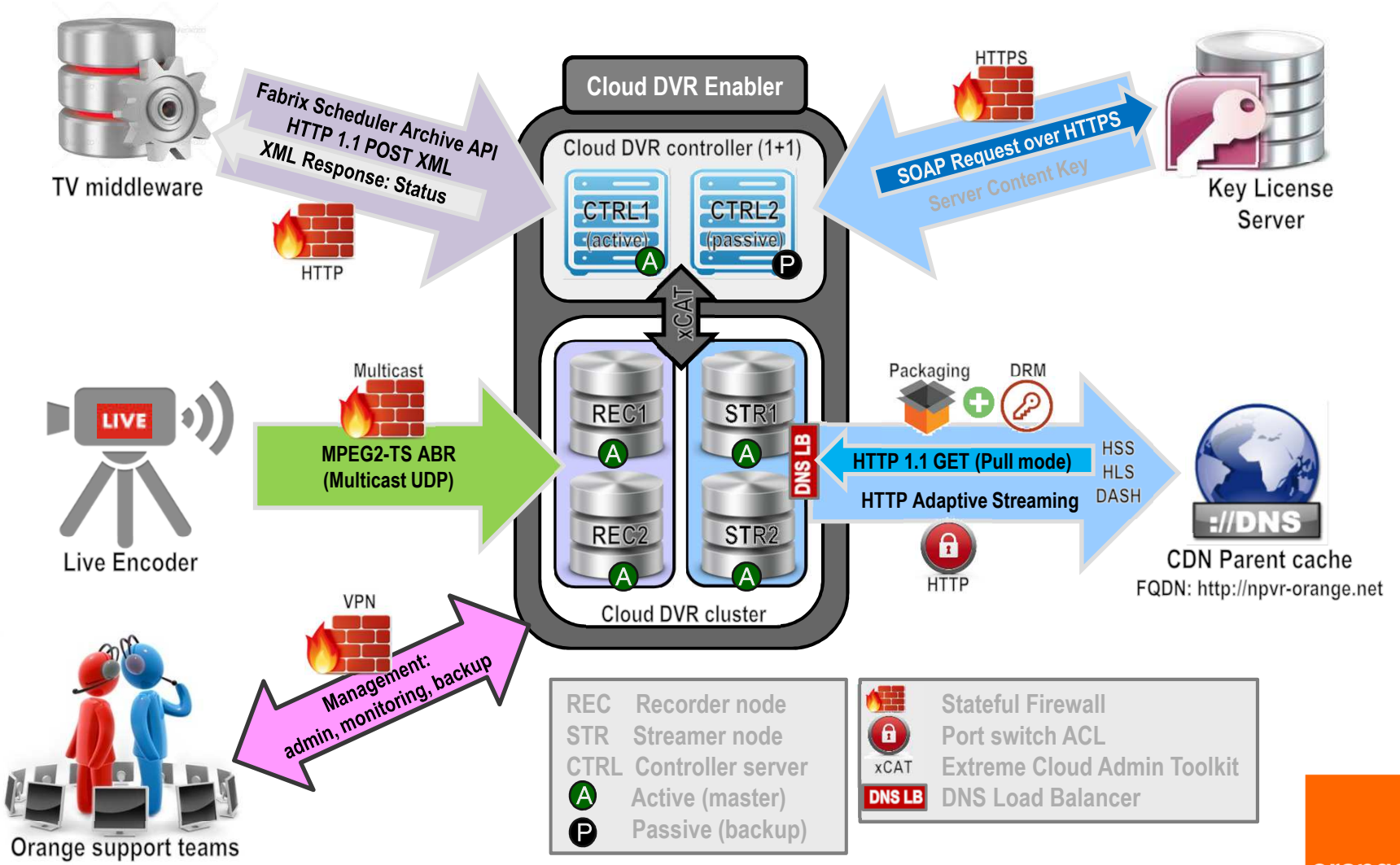
```
http://{cdvr_hls_fqdn}/{token_data}/shls/LIVE${channel_id}/{chunk_length}.m3u8
/{DRM_profile}/Level({bitrate_level})/Segment({timestamp})
```

```
cdvr_hls_fqdn: CDVR FQDN for HLS (ex: cdvr-hls.tv.orange.es)
token_data: Verivue data = "token/expiration time/additional data"
channel_id: id of the live channel (ex: CNN, FOX)
chunk_length: length of the HLS chunk in seconds (Default: 10 s)
DRM_profile: name of the DRM profile defined in the VSPP configuration (ex: ORCA_HLS_DRM)
bitrate_level: bitrate level in b/s (ex: 2.1 Mbps ➔ 2202009)
timestamp: timestamp in ms (UTC epoch time) (ex: Thu, 28 Apr 2016 13:54:12 GMT ➔ 1461851652000
```

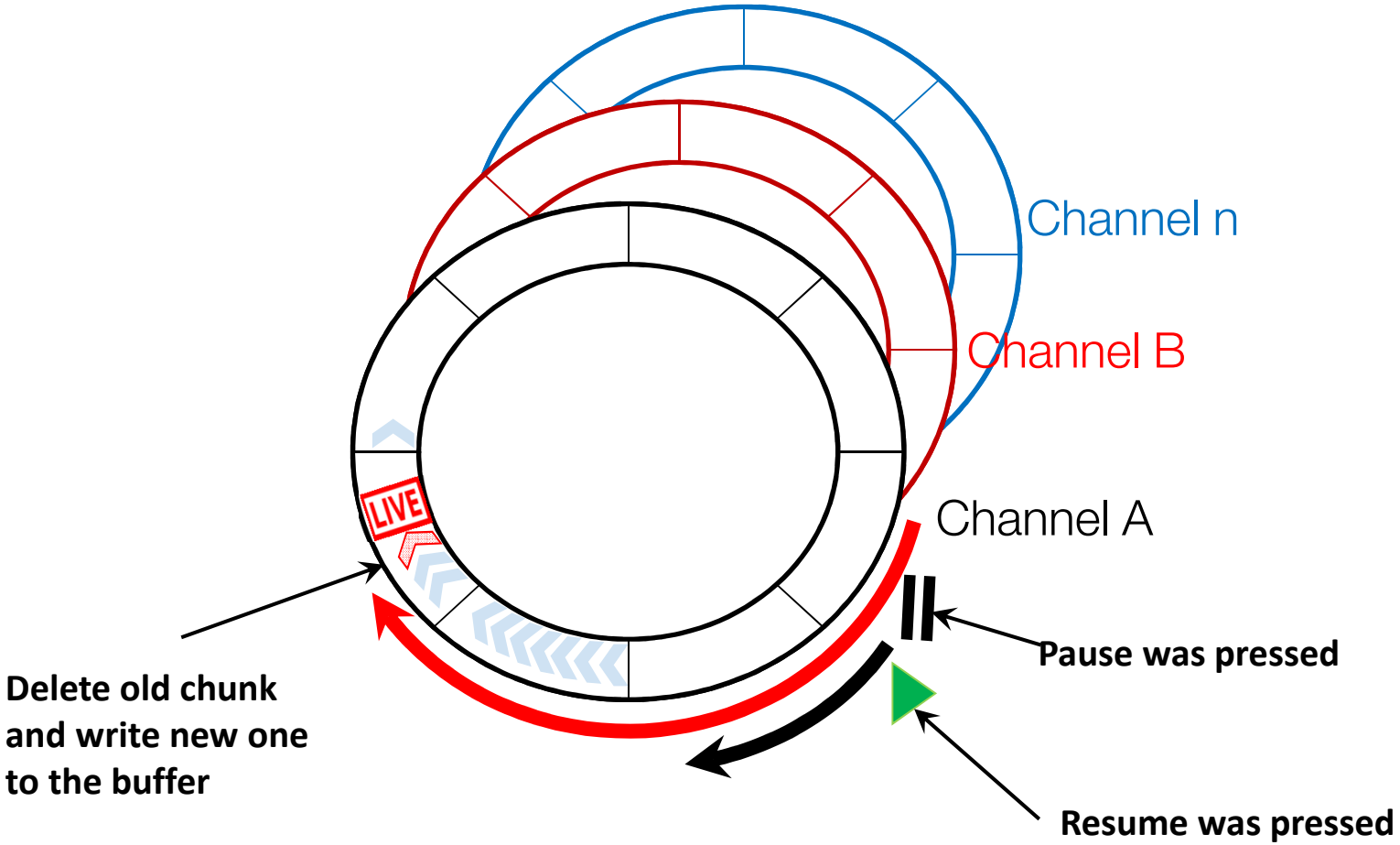
DNS authoritative zone managed by VSPP



Corporate nPVR enabler



Rolling Buffer model – Pause Live TV use case



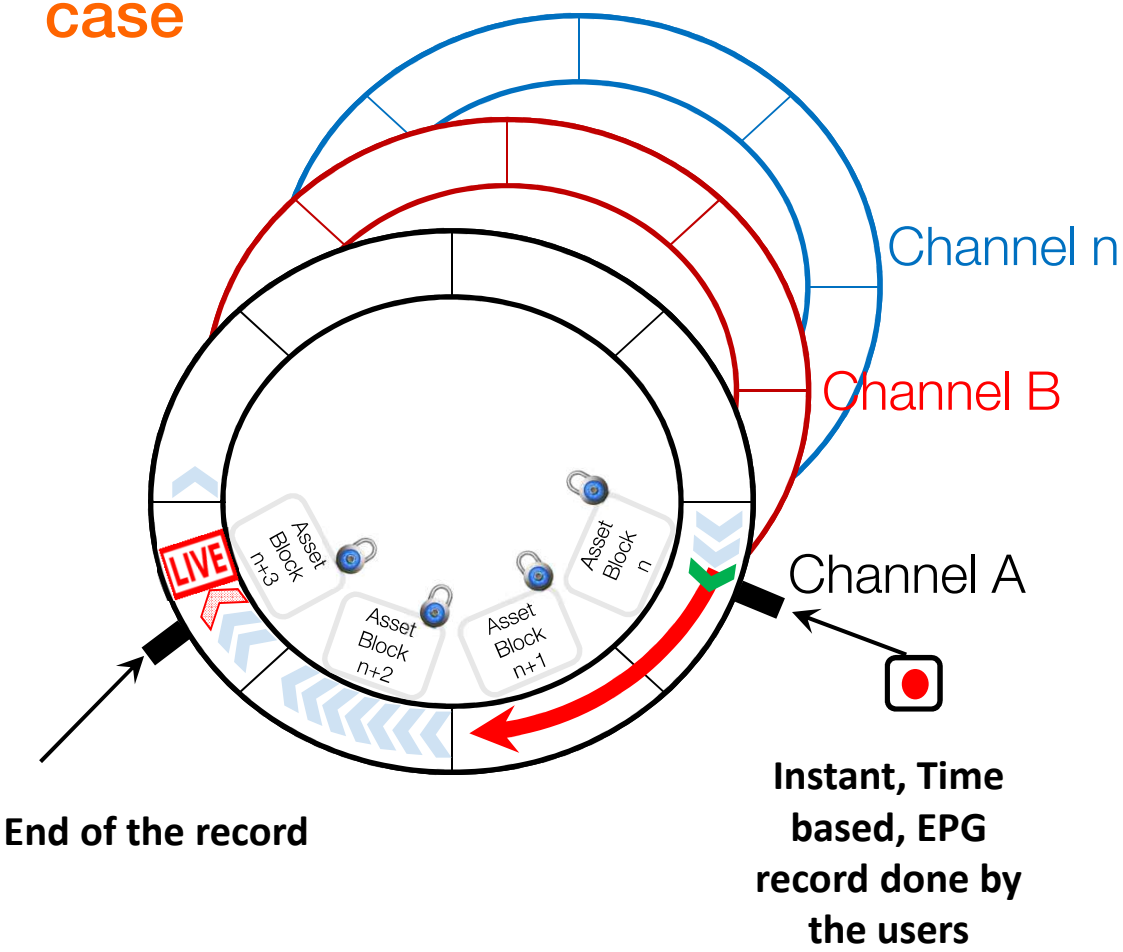
Live availability

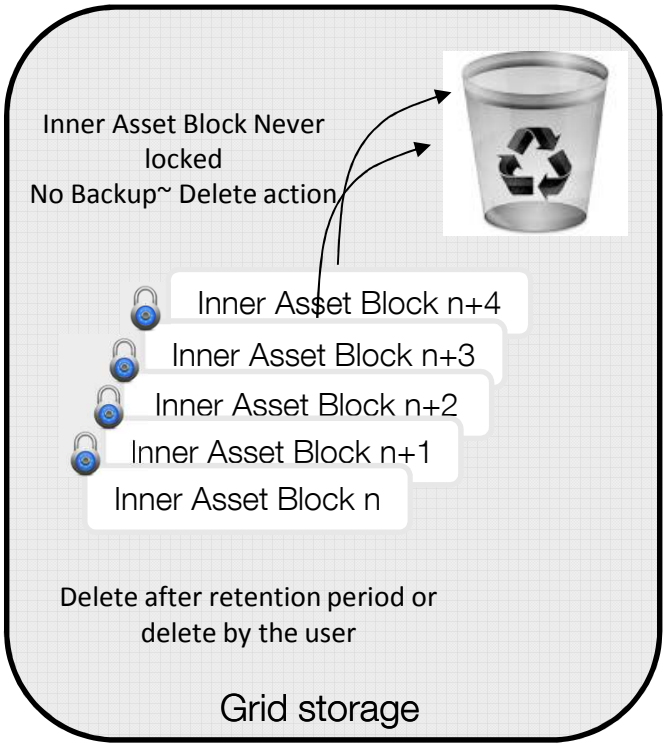
-In CDVR level , the live program is available after 2 GOPs (GOP: Group of successive pictures. Ex: 25 Pictures per second ➔ GOP of 2s = 50 pictures)

- The rolling buffer erases automatically the oldest chunk after the configured retention period.

-No storage into the grid

Rolling Buffer model and Inner Asset Block – Record TV use case



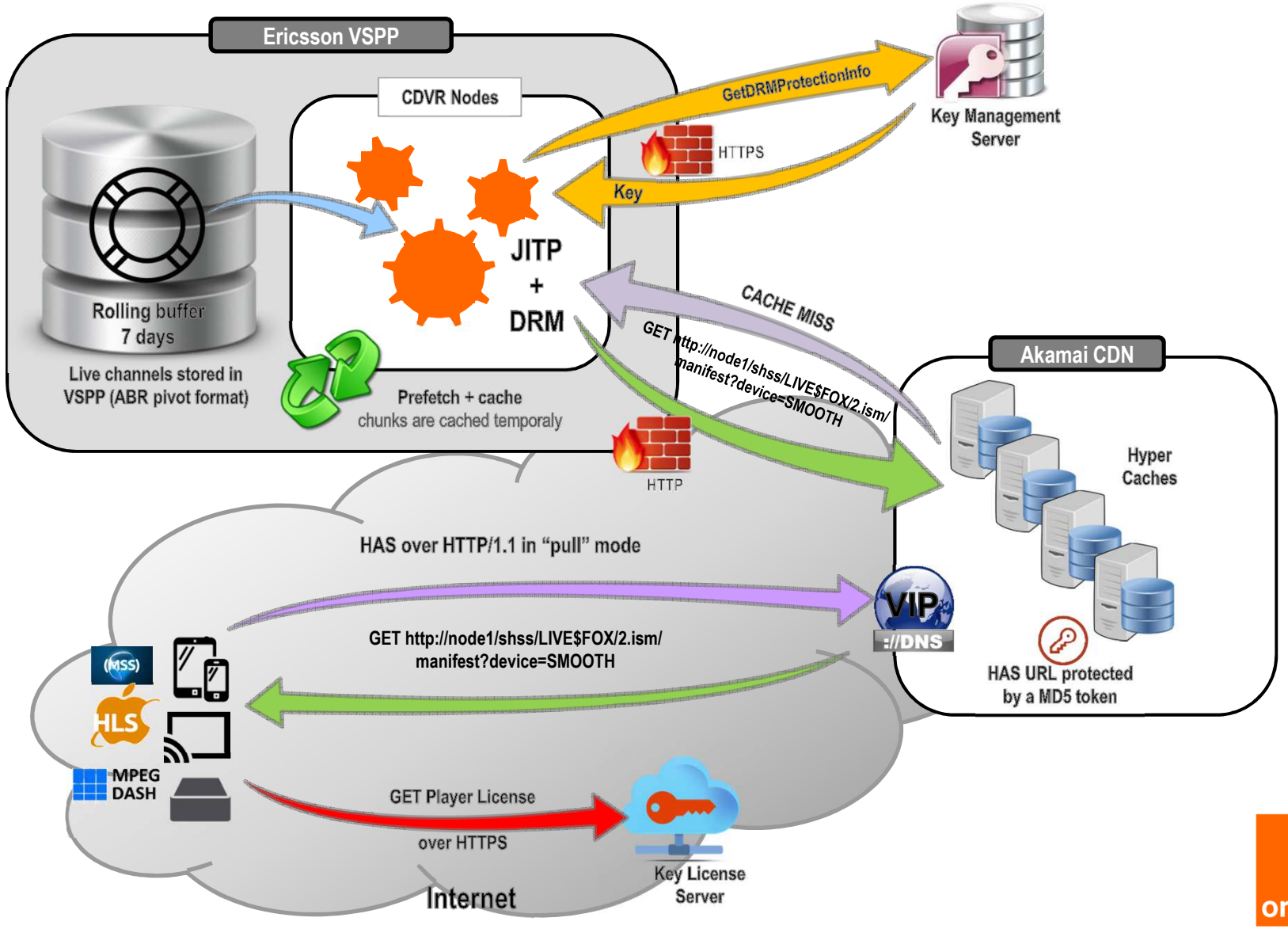


The Asset Block

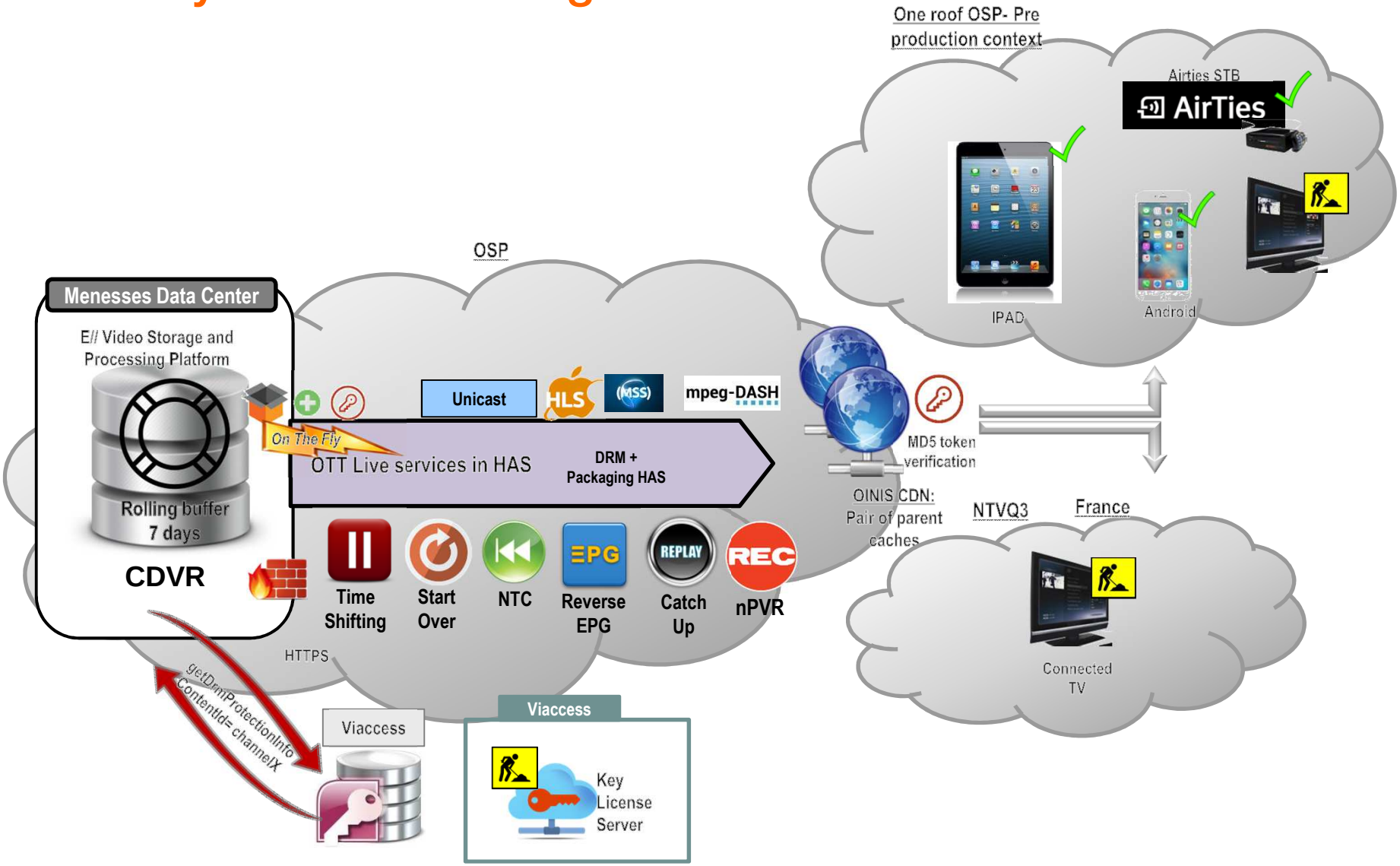
-The asset corresponding to X Chunks for several minutes. The asset duration is fixed 30 minutes. (best trade off for accessing Database)

- The asset is locked and stored into the CDVR over x days (example of retention period)

JITP: Just In Time Packaging + DRM



2 Ecosystems for testing



Garcias

Merci

Thank you

19