brightcove

MASS-SCALE, FAULT-TOLERANT LIVE STREAMING

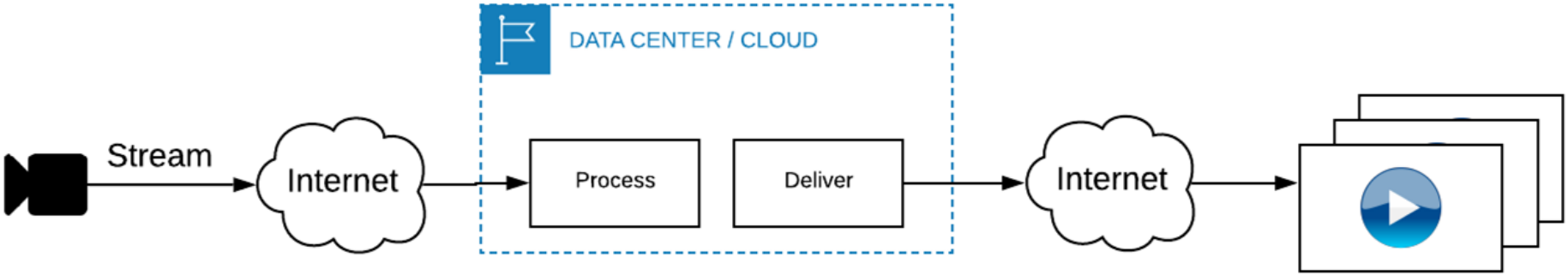
JORDI CENZANO | JCENZANO@BRIGHTCOVE.COM
YURIY REZNIK | YREZNIK@BRIGHTCOVE.COM

AGENDA

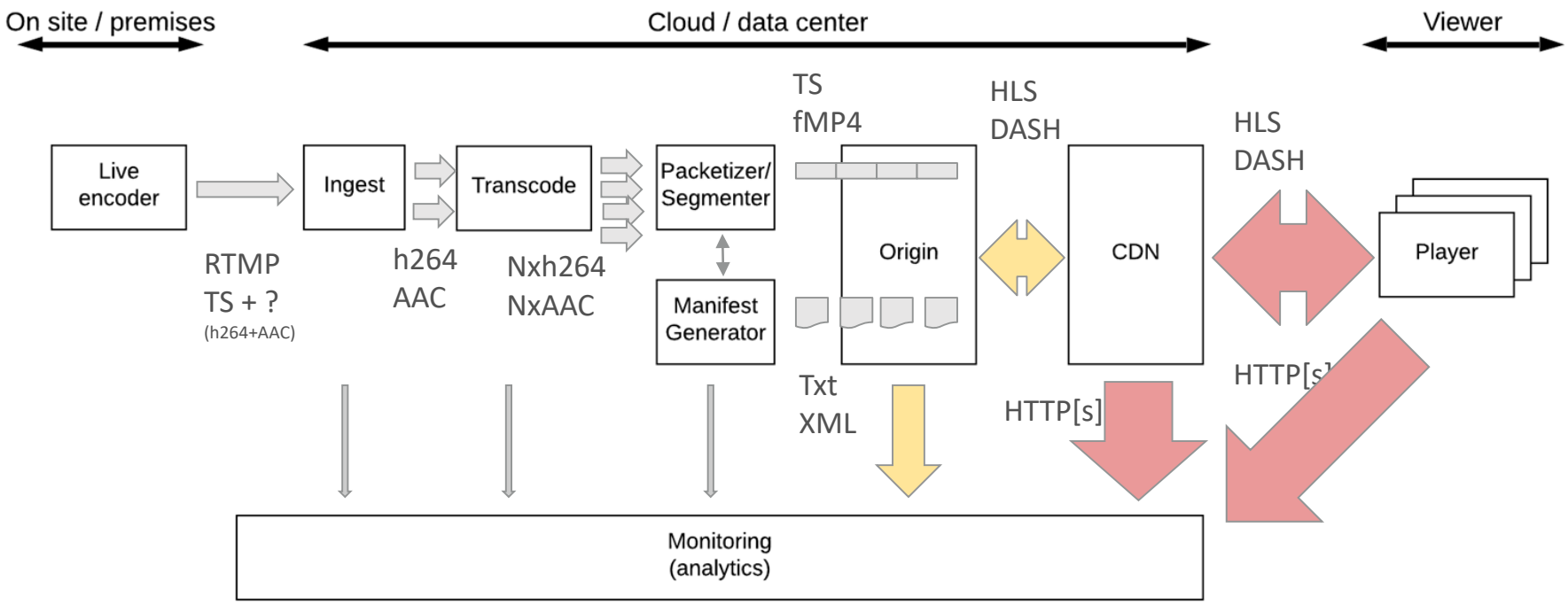
- Simple live streaming
- More complex system; key building blocks
- Some challenges
  - Ingest
  - Personalized live streams (SSAI)
  - Monitoring
  - Redundancy
- Conclusions

WHAT IS LIVE STREAMING?

SIMPLE LIVE STREAMING



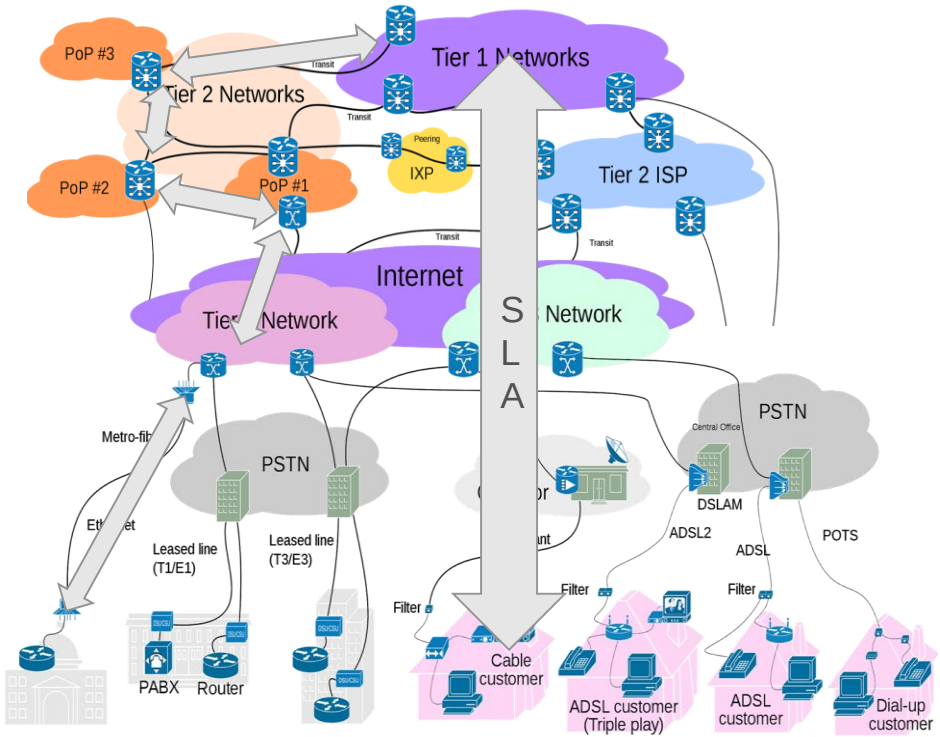
LIVE STREAMING BLOCKS



LIVE INGEST

LIVE INGEST

- Data loss
- Latency
- Jitter
- Reordering
- Corruption
- Duplication

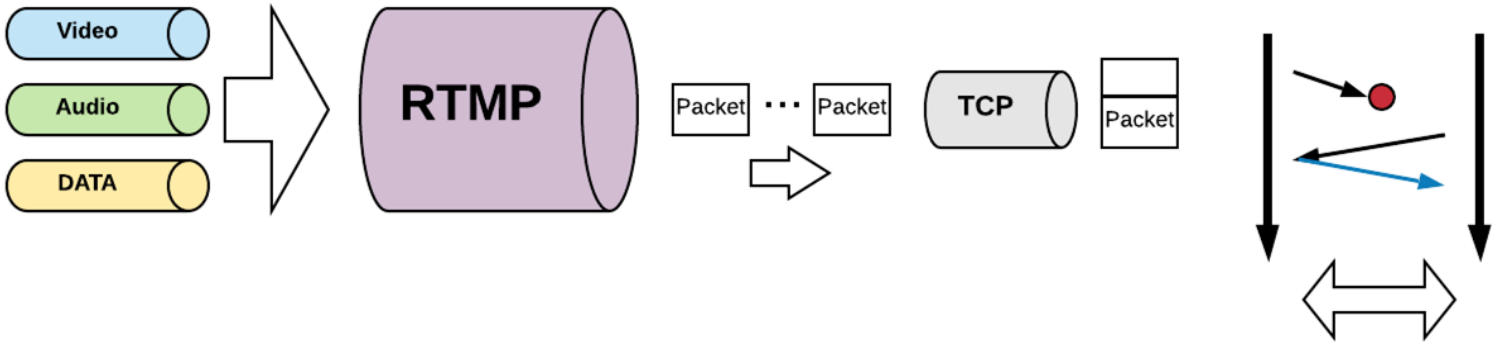


From: https://en.wikipedia.org/wiki/Internet_service_provider

LIVE INGEST OPTIONS: RTMP

- **Pros**
  - Reliable (TCP based)
  - Popular

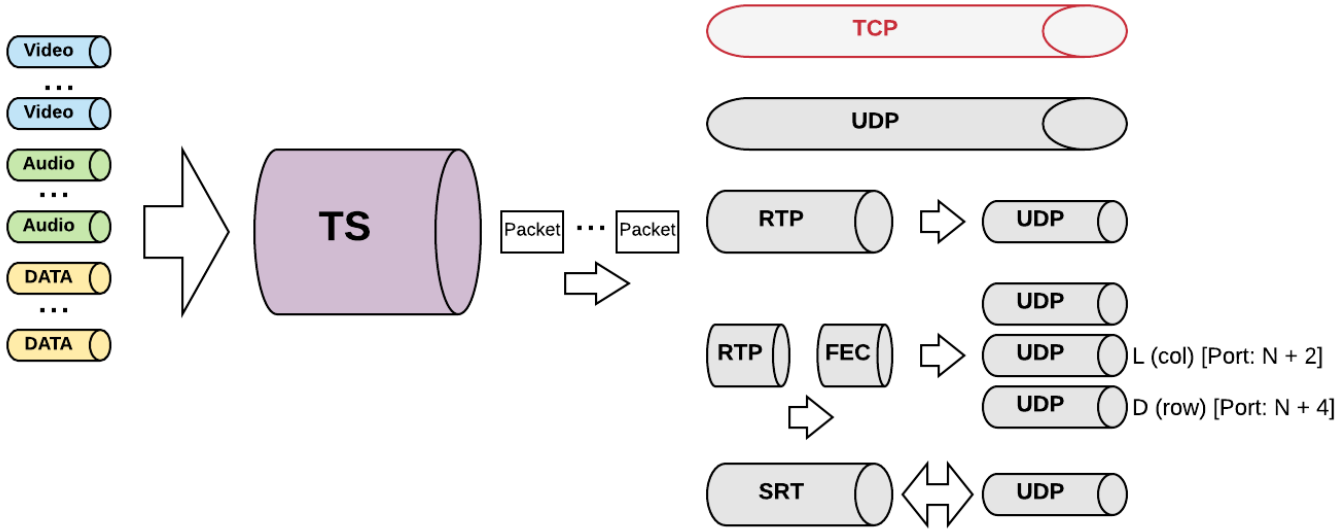
- **Cons** (for live streaming)
  - Only 1 video + 1 audio + data
  - Connection oriented
  - Retries consume BW
  - Slow packet loss recovery
  - **No real time guaranteed**



LIVE INGEST OPTIONS: TS, TS+FEC, SRT

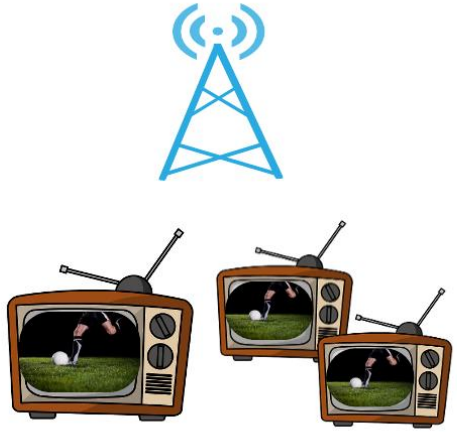
- **Pros**
  - Can transport **everything**
  - Not connection oriented
  - Designed for live
  - Very common in the broadcast world

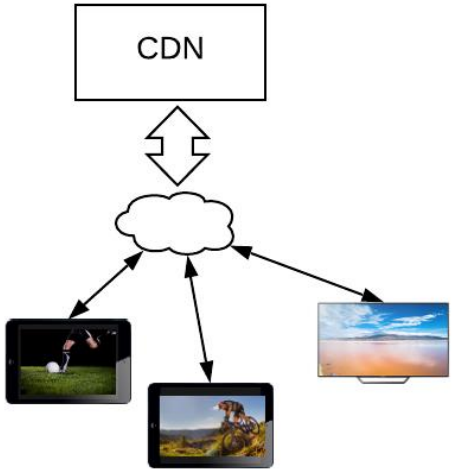
- **Cons**
  - Not reliable
  - Complexity



PERSONALIZATION

(SSAI: Server Side ad insertion)



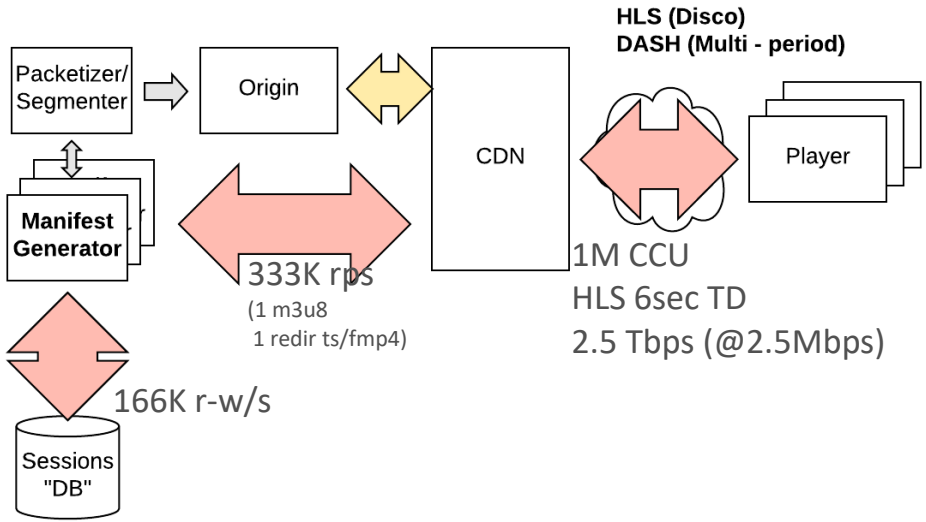


Live streaming

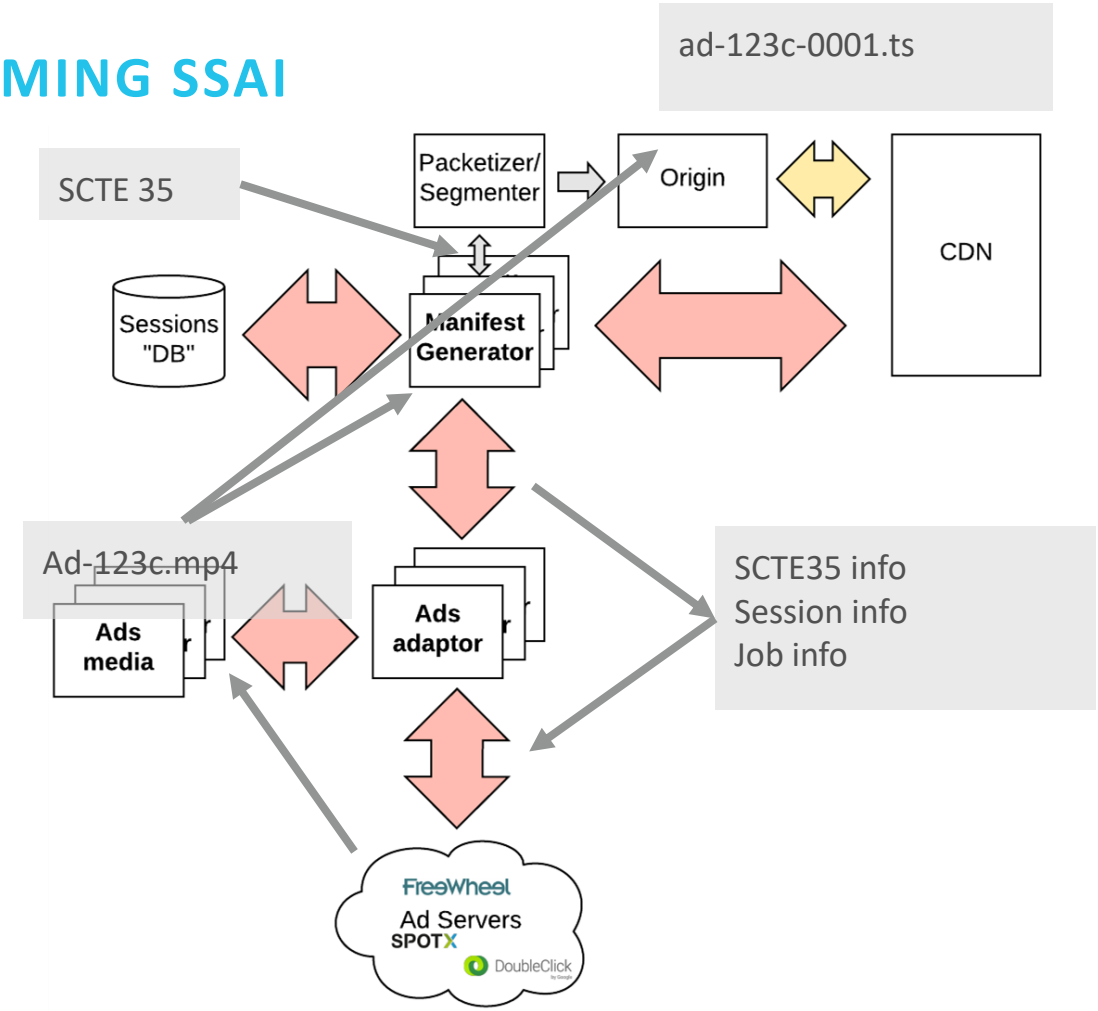
Live SSAI

LIVE STREAM PERSONALIZATION

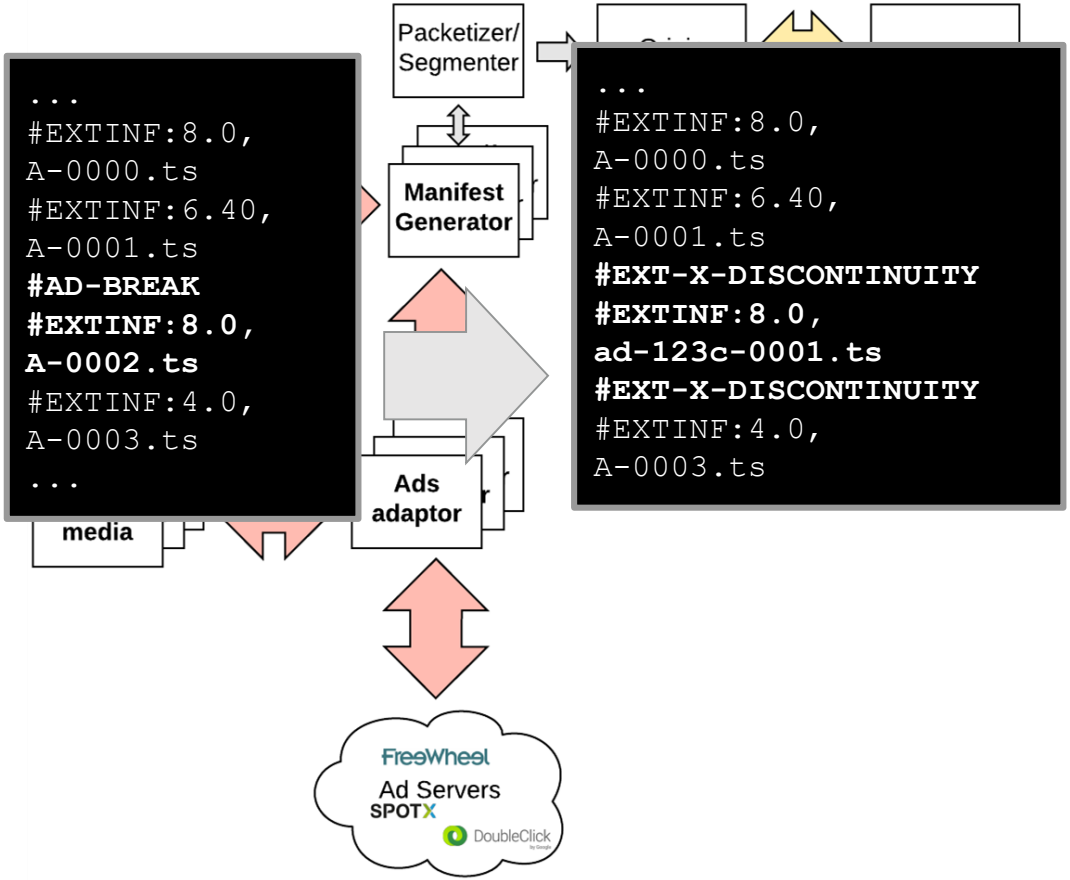
- **Considerations**
  - One manifest per viewer = 1:1
    - Same for content
    - Different for adbreaks
  - Track every viewer in a “DB”
  - **Possible compromise solution:**
    - **Cohorts** (grouping of users based on ad context)
  - **Ultimate solution:**
    - **Edge?**



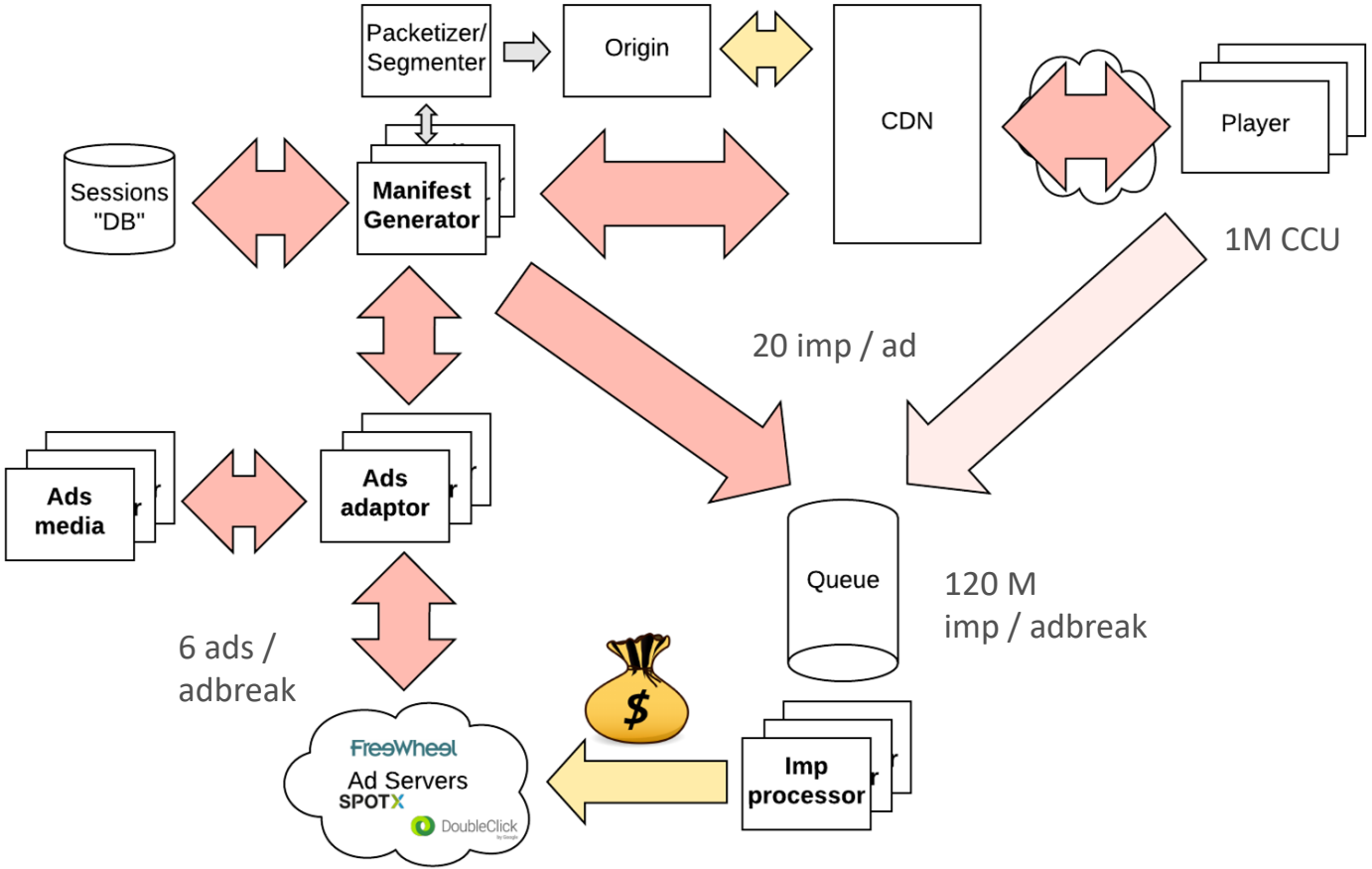
LIVE STREAMING SSAI



LIVE STREAMING SSAI

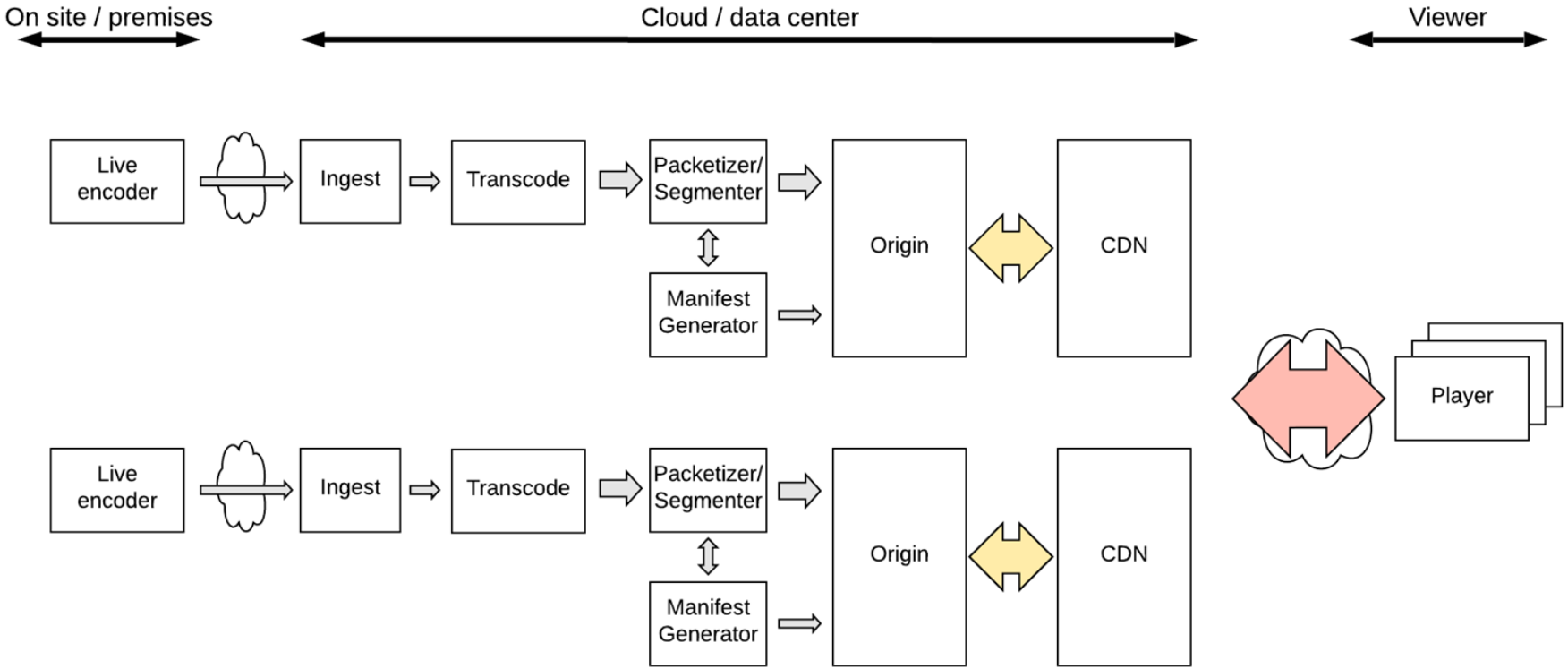


LIVE STREAMING SSAI IMPRESSIONS

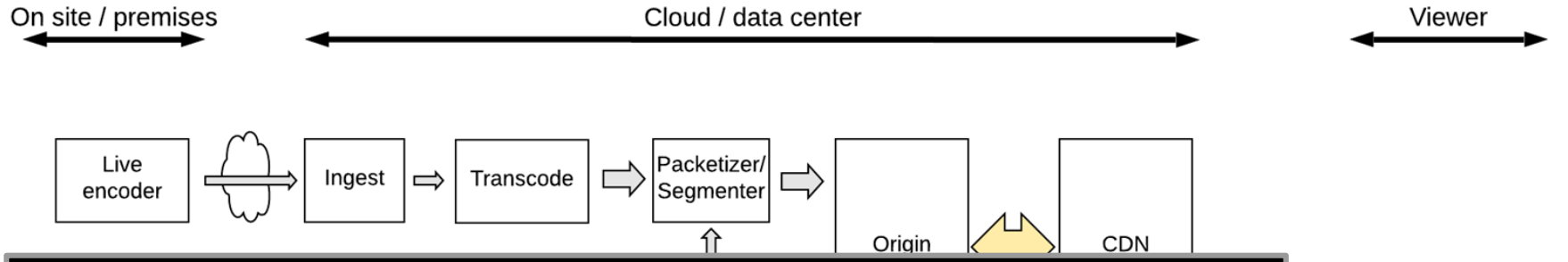


REDUNDANCY

REDUNDANCY BLOCKS



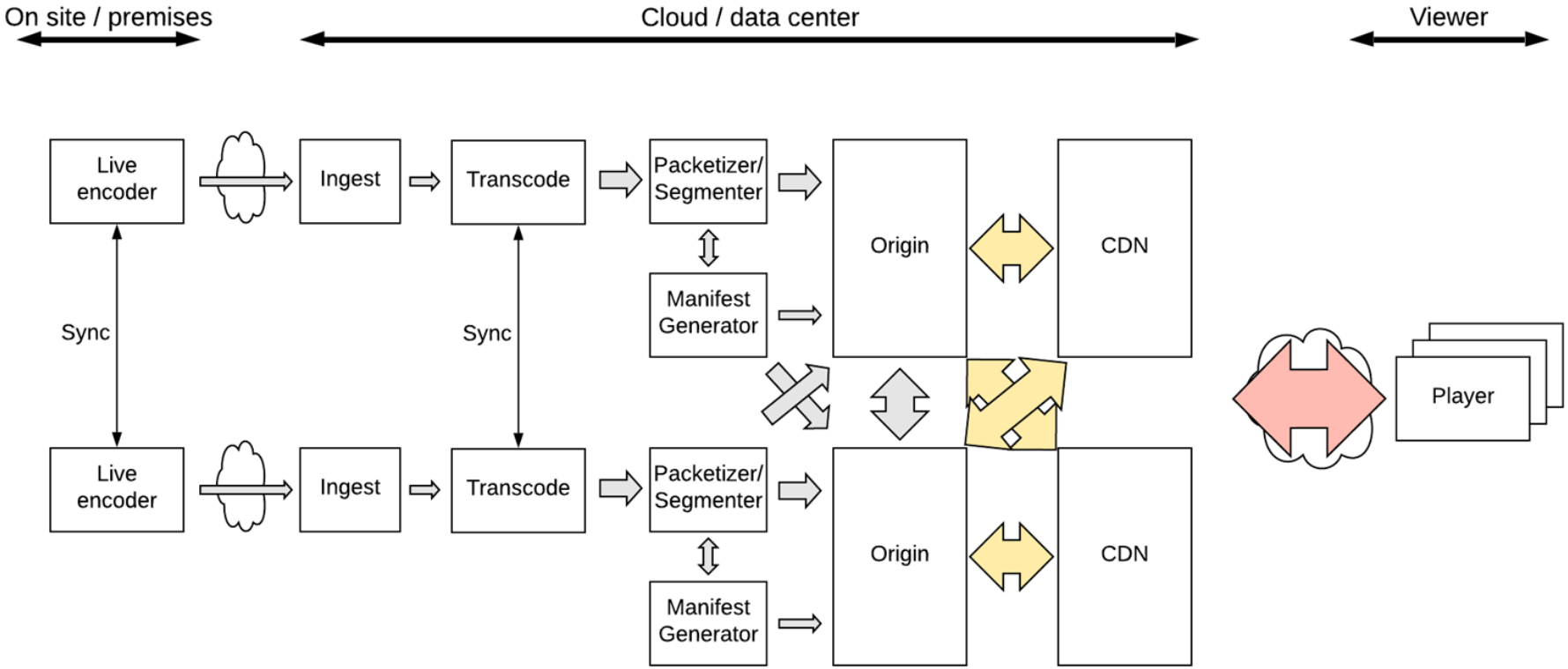
REDUNDANCY BLOCKS



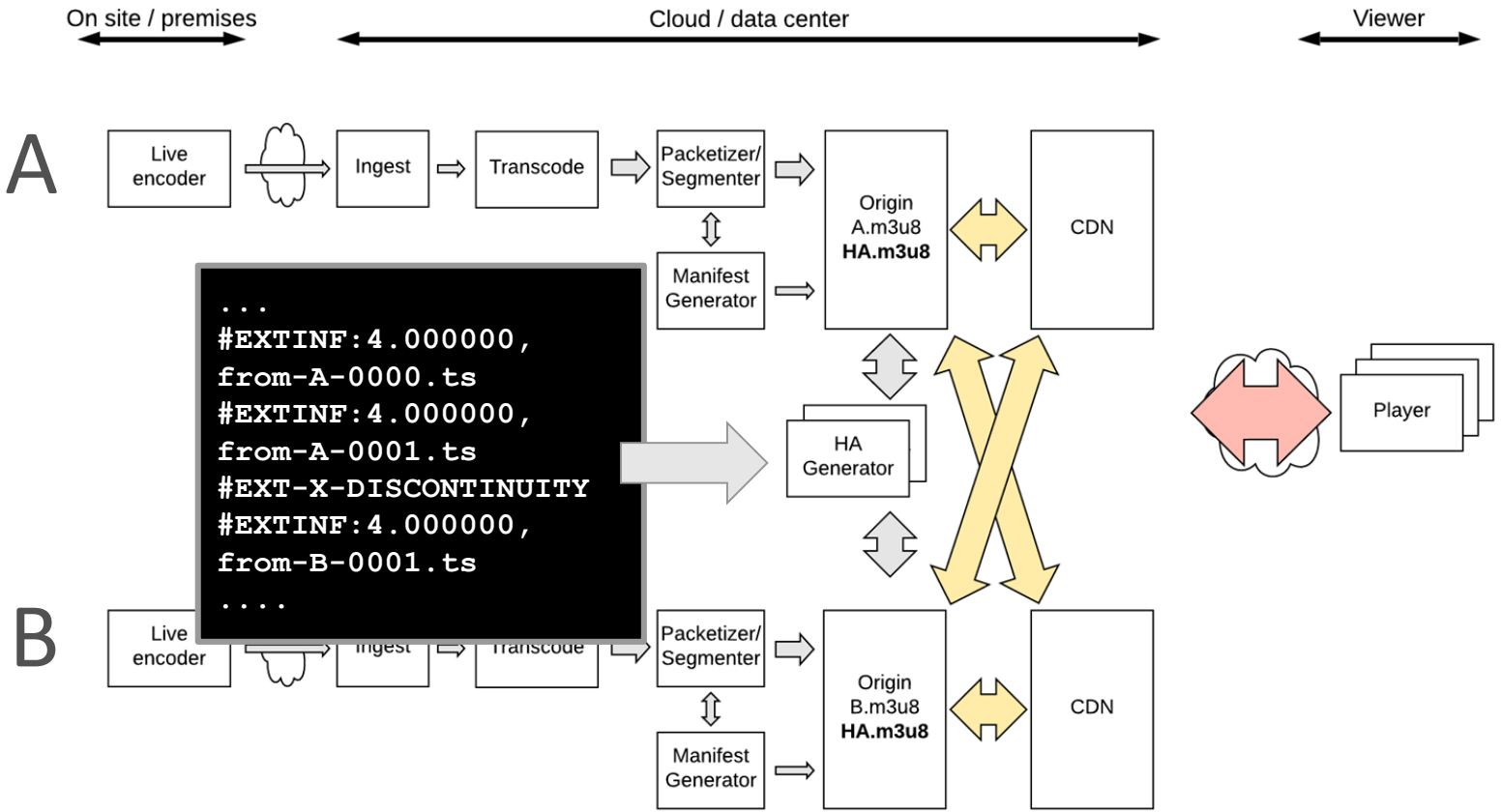

```
#EXTM3U
#Example of HLS REDUNDANT STREAMS
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=200000, RESOLUTION=720x480
http://ALPHA.mycompany.com/lo/prog_index.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=200000, RESOLUTION=720x480
http://BETA.mycompany.com/lo/prog_index.m3u8

#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=500000, RESOLUTION=1920x1080
http://ALPHA.mycompany.com/md/prog_index.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=500000, RESOLUTION=1920x1080
http://BETA.mycompany.com/md/prog_index.m3u8
```

REDUNDANCY BLOCKS: FULL SYNC APPROACH



REDUNDANCY BLOCKS: NON-SYNC APPROACH



CONCLUSIONS

- Design of robust large-scale live systems is not easy:
  - everything fails, and the objective is to survive
- We described several techniques that we implemented to ensure:
  - scale (particularly for SSAI)
  - redundancy / robustness of the system
- But equally important are operational aspects:
  - Have plan B, plan C, ..., plan Z
  - Engage 3rd parties: cloud service providers, CDNs, AdServers, etc.
  - Run load tests (not easy)
    - there are limits everywhere
    - test individually, and
    - end to end
- Large-scale live streaming is basically a challenge, and an excellent problem for engineers to keep working on!

brightcove

THANK YOU!

YURIY REZNIK | YREZNIK@BRIGHTCOVE.COM