Ground-Cloud Integration and the VSF GCCG WG

John Mailhot, Imagine Communications
Kieran Kunhya, Open Broadcast Systems

What is Ground-cloud-cloud-ground (GCCG)?

- Over the last two years we have lived our lives using the cloud:
  - Zoom, Cloud-hosted email, Social Media, Amazon, Netflix etc…
- **But television broadcast production still mainly on-premise – nearly all mid/high end production.**
- We want to make the most of cloud, scale-up/scale down. Not paying for resources that stay idle most of the time (e.g sports).

The picture can't be displayed.

The picture can't be displayed.

Cloud production (1)

- "But I've been doing live cloud production" – Yes and No
- Single Vendor Monolithic applications such as Channel-in-a-box, playout server, cloud switchers
- Multi Vendor systems but with proprietary transport (e.g NDI)

- **But we really want:**
  - **Multivendor cloud production**
    - **Uncompressed exchange between cloud instances**
    - **Agreed mechanism(s) for going to/from Ground and Cloud**

Cloud production (2)

- We need to be able to get data back to the ground and have it work into existing workflows - e.g SDI, ST 2110, satellite, cable, OTA
- All of these have well-defined linear timing models (e.g SDI, ST 2110-21, MPEG-TS VBV)
- Without a timing model you end up with variable (undefined) latency. One reason web streams are 20-30 seconds behind broadcast – They don't have a timing model!
- Cut between streams and see goal again!

I’ll just do 2110 in the cloud

- Some people claiming to have 2110 in public cloud
- But it’s not possible right now in any public cloud:
  - **No (full) PTP in the cloud – all clouds handle time their own way**
  - Cloud networks are shared and have packet loss
  - No real access to network card capabilities (e.g packet pacing) for 2110

- We don’t actually want linear processing in cloud any more
  - Allow cloud instances to process data non-linearly, sometimes faster or slower than real-time but on average real-time – known worst case
- How to handle “synthetic” sources (e.g clips, graphics) played out from cloud?

Cloud-vendor specific transport

- Since the cloud is lossy, we have to depend on cloud provider protocols with guarantees:
  - Throughput with Reliability - all my data arrives correctly
  - Latency - all my data will arrive on time

- Use efficiencies in large data transfer for “Big Data” in cloud (RDMA)
  - May not have visibility of the internals of this protocol (“black box”)
- Amazon Scalable Reliable Datagram (SRD) such an example
- Used in Amazon CDI

Amazon CDI

- How does the Amazon CDI protocol compare?
  - Handles many of the challenges discussed
  - An agreed way to exchange data between Amazon cloud instances. Defined pixel data structures, metadata (e.g HDR) etc
- Amazon guarantees throughput, reliability and bounds latency
- **A big step forward for the industry**

- Some parts need GCCG input, more detailed timing model
- GCCG could sit on top of CDI

What is the GCCG working group

- The GCCG working group is trying to solve this problem

- The last difficult problem in broadcast production (personal view):
- How can I do a complex multichannel production in the cloud, with comparable latency to on-premises and get it to the viewer?
- Numerous technical challenges

- https://vsf.tv/Ground-Cloud-Cloud-Ground.shtml

Orchestration and Control

- The boring but important part ☺
- How do cloud instances expose their capabilities (max resolution, frame rate etc.)?
- Is there an orchestration layer that connects the pieces together?
- Handling changes in signal flow (resolution/framerate, audio channels)
- Connection status (Am I ready?)

- Control of Ground to Cloud (and vice-versa) elements

What are the "operating points" we see today (for TV)

Focus on these

- Uncompressed (2022-6 or 2110-20) (already defined)
- **"PremiumCompressed" – super low latency, super high quality**
  - **JXS for (1080i @ 200m?) (UHD @ 1000-1500M)**
  - **Dual-path reliability model for super-good reliability @ *minimum latency***
- J2K-ULL (stripes) @200
- J2K (full-frame) @120
- AVCI @ 100
- **Interactive Latency ~20Mbit(HD) (ULL restricted VBV buffer) $$**
  - **ARQ/RIST or dual-path model or FEC? (Typical 4:2:2/10 profile)**
- **Rate Optimized (longer latency @ lower rate) (HD@3-10 Mbits, more delay)**
  - **ARQ/RIST or dual-path model or FEC? (Typical 4:2:0/8 profile)**
- Zoom / GoToMeeting / Webex – whatever gets a picture on the screen
  - Internet best-effort reliability work

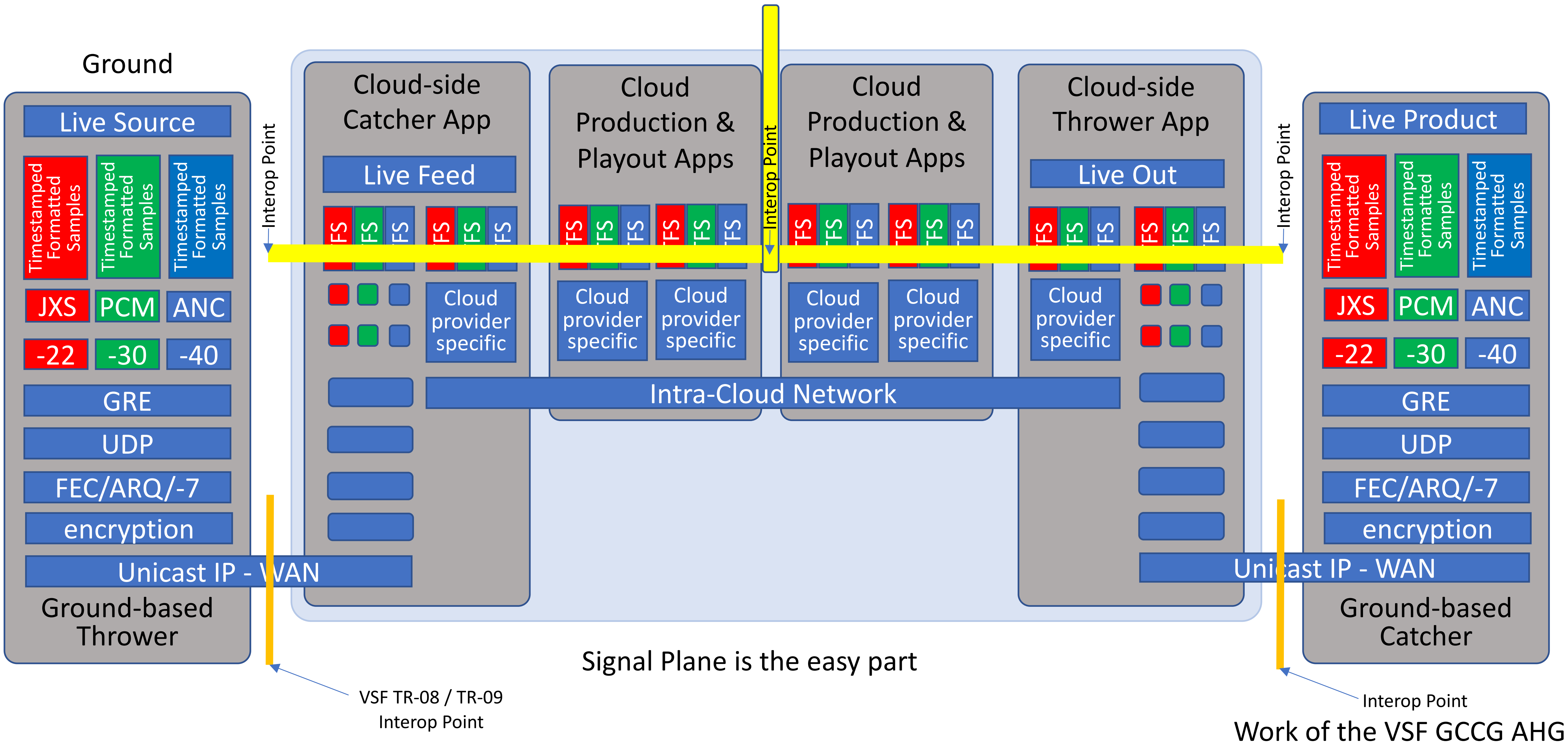
Premium Compressed (JPEG XS)
VSF TR-08: codec & LAN 2110-22 base
VSF TR-09: WAN 2110-x extension
*** reference TR-08**
*** opt: with 2022-7**
*** opt: with FEC (small intl, 1D)**
*** opt: GRE Tunnel (ref RIST)**
*** opt: encryption (ref RIST)**

Interactive Latency (small GOP or Prog Refresh)
- **(HD) H264 (constrained VBV, 4:2:2, 10bit)**
- **(UHD) H265 (constrained for latency)**
- **2022-2(TS-RTP)**
- **RIST-FEC (+/- multi-path)**

Bandwidth Optimized
- **(HD) H264 (420/8 standard vbv)**
- **(UHD) H265 ()**
- **2022-2(TS-RTP)**
- **ARQ or FEC**
- **Opt: encryption (ref RIST)**

Work of the VSF GCCG AHG

Where are the points-of-interoperability?



Signal Plane is the easy part

Work of the VSF GCCG AHG

VSF and other Related Work In Progress

- **Document the "Premium Compressed" use case for G-C and C-G: TR-08 & TR-09**
  - **TR-07 & 08 published**. JPEG-XS over TS and over 2110-22 with interop points and capability sets
  - **TR-09 is publishing soon**. Defines the data encapsulation plane, plus NMOS-like Control plane

- Is there something more to document about the TS/IP/H.26x case? Or is it good enough?
  - Can the TR-09 (ARQ, FEC, -7) control plane be applied to this class of streams ? (yes, probably)

- Time-Flow / Time-Transport / Time-Tagging model
  - How do we treat time in the concatenated virtualized systems
  - How do we integrate "real-time" on the ground with "floating time" in the cloud?
  - This requires a vocabulary and some modeling/specification effort
  - Define how to catch up / manage drift / manage change / re-integration to timeline

- The Media Containers / Object Format structures for hand-off from application to application
  - The AWS CDI structures are defined on github
  - Need to document how this handoff interacts with latency and latency accumulation

Work of the VSF GCCG AHG

The “time variability / time floating” work

How do we avoid re-creating our frame-sync pipeline in the cloud, and let software work well?

- Allow variability in the handoffs, but with an ability to predict the outcome
- Some processes must reconcile the variable inputs into a consistent output
  - Must bound the input buffering (latency) yet accommodate the variability

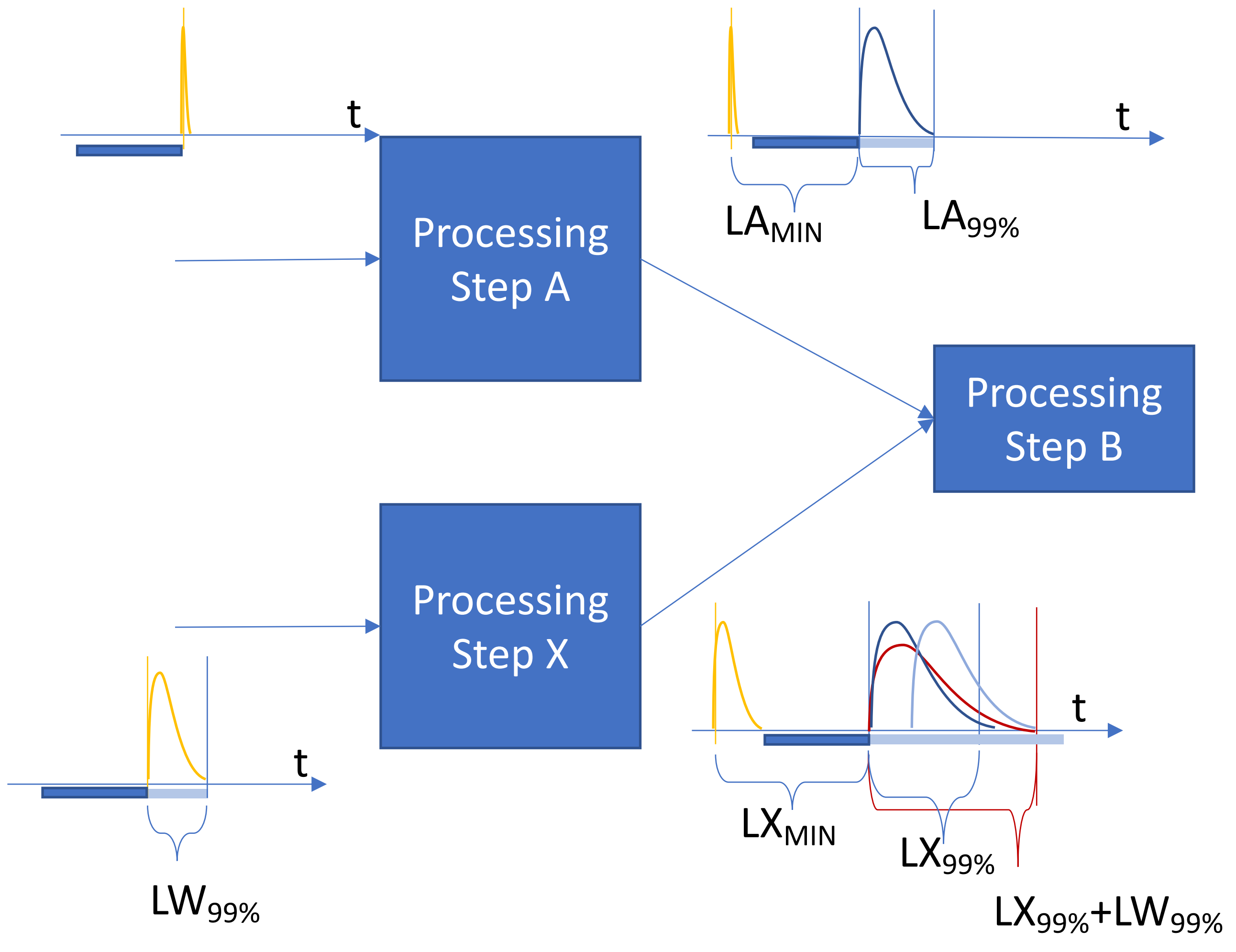
- What are the sources of delivery (arrival) variability?
  - Network-induced (probably small)
  - Processing-induced (upstream process steps with variable latency)

- Vocabulary (about each process step in the datacenter)
  - TX Variability – bound on how early/late the signal may egress, variance above baseline
  - TX Planning Delay – the baseline in the above statement – the smallest value of delay
  - Actual delay is planning delay plus some amount of variability

Work of the VSF GCCG AHG

What is the "spec" of latency and variability?

- Push model (not backpressured)
- Based on (uniform?) content "chunks"
  - might be frames, fields, stripes
  - might be blocks of audio samples
- L_{MIN} = the soonest/shortest amount of time from input to output
  - Input time = buffer 100% arrived to me
  - Output time = buffer left me 100%
  - Includes the egress transit time
- $L_{99\%}$ = the amount of variability *beyond* the L_{MIN} for 99th percentile case
- What about continuity? Do processing steps need to maintain cadence if input not there?
  - Maybe not – only if it "has to" for its own processing purposes. Otherwise best to just be late or missing and let downstream do the best it can with what it gets when it gets it.



Variability on the input accumulates into the output!!!

Work of the VSF GCCG AHG

Summary of Consensus Views and path forward

For ***Ground-to-Cloud*** and ***Cloud-to-Ground***, target TR08/TR09 for high-quality

For ***inter-instance*** (intra-cloud) coordinated handoff (a “virtual facility”)

- How to identify sender and receivers (use NMOS extended, similar to TR-09)
- How to initiate connections (IS-05 extended) (WIP in AMWA BCP-06-x)
  - What is the content description lingo? (we can recommend user requirements to AMWA)
  - What are the transport params for CDI? for other clouds? (WIP getting started in AMWA)
  - What is the timing description specification? (document our WIP)
- What is the data format of the buffer-style handoff between instances in cloud?
  - Canonical format = 2110 pgroup concatenated schema
  - Other buffer formats can exist and may be more perfect, i.e. https://vsf-tv.github.io/cef/
  - Canonical audio-only format including timing relationship and transport suggestion (TBD)
  - Canonical metadata format including timing relationship and transport suggestion (TBD)

- For inter-envelope (arms-length) handoffs within/between cloud(s)
  - Essentially this is like the ground-to-cloud case

Where do we go from here?

- The VSF GCCG group has some WIP to document,
  and some user requirements to document/forward to AMWA
- AMWA is extending NMOS to other stream types
- VSF already published TR-08
- VSF is publishing TR-09 very soon

- Be sure to attend other talks this week about the sub-topics in here

Thank You ….. Or ….. Any Questions?

John Mailhot, Imagine Communications
Kieran Kunhya, Open Broadcast Systems