NMOS at scale

Networked Media. Working.

Tim Hall IBC 2022

About me

- UK Based independent consultant
- Started career in TV audio
- System design
- Station builds
- Specialising in workflows, control, configuration, intercom
- Several IP projects delivered:
  - ST2022-6, ST2110, AES67, Dante
  - Mixture of Control and Orchestration systems used

- Real World Implementation
- Issues
- How can we improve things?

Networked Media Open Specifications (NMOS)

- Provides an open and simple to use control-plane solution that enables interoperability and management of IP connected audio and video devices.
- Provides seamless management of video, audio and data streams for interactive and interoperable production of live content.
- Avoids vendor lock in
- Lets you find, connect and configure media devices to enable video and audio on your IP network.
- Can be used to build and manage small and large multi-vendor systems.

Real world implementation

- 30+ vendors, across multiple sites
- Two large datacentres housing back-end processing/mixing
- A mix of NMOS and native API control
- 300,000 endpoints (100k are NMOS)
- No NMOS registry
- Creation of offline data model and tools to gather, process and prepare configuration for:
    - Devices under control (where external configuration API is available)
    - Orchestrator
    - Higher level control system

Constraints on our implementation

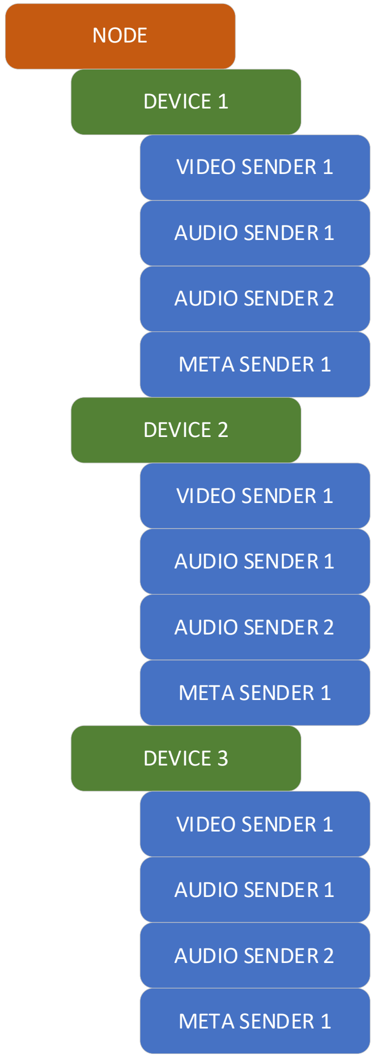
- Our orchestration layer GUI only displays a single label which is inherited from the IS-04 label
- Logical Grouping is not used by orchestration layer
- Sheer amount of data makes navigation very hard
- Although the two datacentres are isolated, they are linked to enable efficient workflows
- The higher level control system has an integer based model for dealing with a source-destination connections
- A very long project, 3+ years
- The facilities are on-air, and still growing

Issues

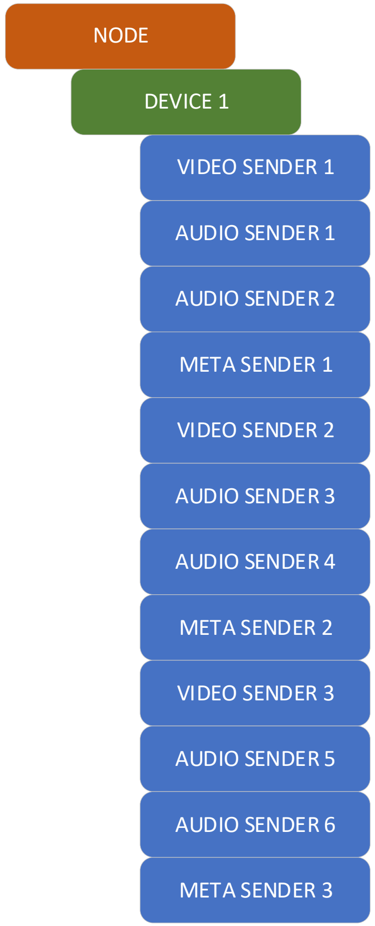
- Troubleshooting is challenging using native IS-04 labels if they are a mixture of formats/presentations
- Not all vendors implement BCP-002-01 Natural Grouping – when they do, consistency across products is not guaranteed
- Some NMOS implementations make it difficult to distinguish between instances of given device type
- Large scale implementations take time – we need as many clues to identify individual instances / groupings to ease the configuration burden
- It is a lot of work to replace an object when a GUID changes, let alone when 100 GUIDs change

Vendor Node/Device/Sender/Receiver structures

- Different approaches to the same problem e.g.
- Lack of a clear explanation of:
  - The structure and how it could change with reconfiguration
  - Are counters 1 or 0 based?



- NODE
  - DEVICE 1
    - VIDEO SENDER 1
    - AUDIO SENDER 1
    - AUDIO SENDER 2
    - META SENDER 1
  - DEVICE 2
    - VIDEO SENDER 1
    - AUDIO SENDER 1
    - AUDIO SENDER 2
    - META SENDER 1
  - DEVICE 3
    - VIDEO SENDER 1
    - AUDIO SENDER 1
    - AUDIO SENDER 2
    - META SENDER 1



- NODE
  - DEVICE 1
    - VIDEO SENDER 1
    - AUDIO SENDER 1
    - AUDIO SENDER 2
    - META SENDER 1
    - VIDEO SENDER 2
    - AUDIO SENDER 3
    - AUDIO SENDER 4
    - META SENDER 2
    - VIDEO SENDER 3
    - AUDIO SENDER 5
    - AUDIO SENDER 6
    - META SENDER 3

GUIDs

- Non-unique GUIDs
  - re-use of sender/receiver GUIDs on different Nodes
- Non-persistent GUIDs
  - Not persisting GUIDs through reboots/firmware upgrades
  - Not persisting GUIDs through re-configurations

Labelling schemas

- Fully vendor defined

```
IP Audio Sender 33 of AES67 UIC Node 35 - Bay 5
2110-20 Video Receiver 1 on 05rast001_C1000046
```

- Partially vendor defined

```
11/MXR/001_16x8-IO-IP-1_i1-17_o1-9:ipMultiChRx0_RCVR_PAIR_AES67_4
11/MXR/001_16x8-IO-IP-1_i1-17_o1-9:ipMultiChRx0_RCVR_PAIR_ANC291_10
11/MXR/001_16x8-IO-IP-1_i1-17_o1-9:ipMultiChRx0_RCVR_PAIR_UNIVERSAL_0
```

- Customer defined

```
12345/CCU/001_s01_v01
12345/CCU/001_s01_a01
12345/CCU/001_s01_a02
12345/CCU/001_s01_m01
```

Different approaches to achieving this:

- Typing into vendors config tool
- Creating patterns
- Uploading a lookup table

Our vocabulary

```
12345/CCU/001_s01_v01

12345/CCU/001_s01_a01

12345/CCU/001_s01_a02

12345/CCU/001_s01_m01
```

- Each object in the design is assigned a hierarchical ‘UID’ which contains data to tell us
  - **WHERE** it is (5-digit numeric code)
  - **WHAT** it is (short form of device function: MIXER/CCU/AMU/IRD/PROC)
  - **WHICH ONE** it is (zero-padded counter)

  e.g. 12345/CCU/001 – the first CCU in area 12345 – provides a context for each object

- Usually (but not always) the device is named using this UID – when then becomes the NMOS IS-04 Node label
  - The device may not allow renaming/may not accept “/” chars
  - The physical or logical device may have several NMOS Nodes and may therefore need further qualification

Senders and Receivers

- We mandate the format of the text in the IS-04 sender and receiver labels, essentially embedding the grouping data in a label (so that we can see them in orchestrator GUI)
- For a sender/receiver:
  - s01 - “spigot” 1 - logical grouping
  - r14 - “spigot” 14 - logical grouping
- For the essences:
  - v01.. Video
  - a01.. Audio
  - m01.. Metadata
  - x01.. JPEGXS

So Vendors .... Please

- Provide implementors a clear definition of what your Node/Device/Sender/Receiver schema is – and how it could change at run-time
- Implement BCP-002-01 Natural Grouping
  - of 74 nodes at recent JT-NM tested event, 65% passed
- Only change GUIDs when really necessary
- Give implementors the opportunity and tools to implement a labelling schema that fits their project/facility

Within AMWA..

- We are discussing how we might encourage and/or enforce greater consistency within implementations
- As a first step, we have worked up some user stories, e.g.

- *As an end user, I need a much clearer way of differentiating between objects in the IS-04 tree, in order to avoid the duplication and ambiguity that result when left to the vendor to decide.*

Any Questions?

User Stories

- As an end user, I need a much clearer way of differentiating between objects in the IS-04 tree, in order to avoid the duplication and ambiguity that result when left to the vendor to decide.
- As an end user, I would like to be able to define my own labelling schema for nodes, devices, senders and receivers such that when objects are discovered or appear in the registry, in order that the labelling schema has relevancy to the project or facility build.
- As an end user, the labelling schema should have the capability to use inheritance, such that placeholder variables can be used to substitute %NODE%, %DEVICE% labels in sender/receiver labelling in order to improve the usability of the schema and to avoid errors.

User Stories

- As an end user, a sender or receiver label must be fully qualified and human readable, without the need to reference any further lookups in order to allow the user to avoid unnecessary distractions coming from using external information.
- As an end user, this scheme must work at node level and not purely in the registry or higher level control system in order to allow nodes to be named independently of upper level system components.
- As an end user, there must be no restriction on characters that can be used in the labelling schema in order to make full use of fullest flexibility in naming (including international)

User Stories

- As an end user, I would like BCP-002-01 to be mandatory and for the data therein to be more consistent across vendors in order that a complete system with multiple vendors has consistency for naming.