INFO-001 Control / Monitoring / Management Architectural Sprint
Type: Informational (INFO)
Project Leader: John Mailhot
Maturity Level: Proposed Specification
Date Published: 13 August, 2016
Location: http://www.amwa.tv/projects/INFO-001.shtml

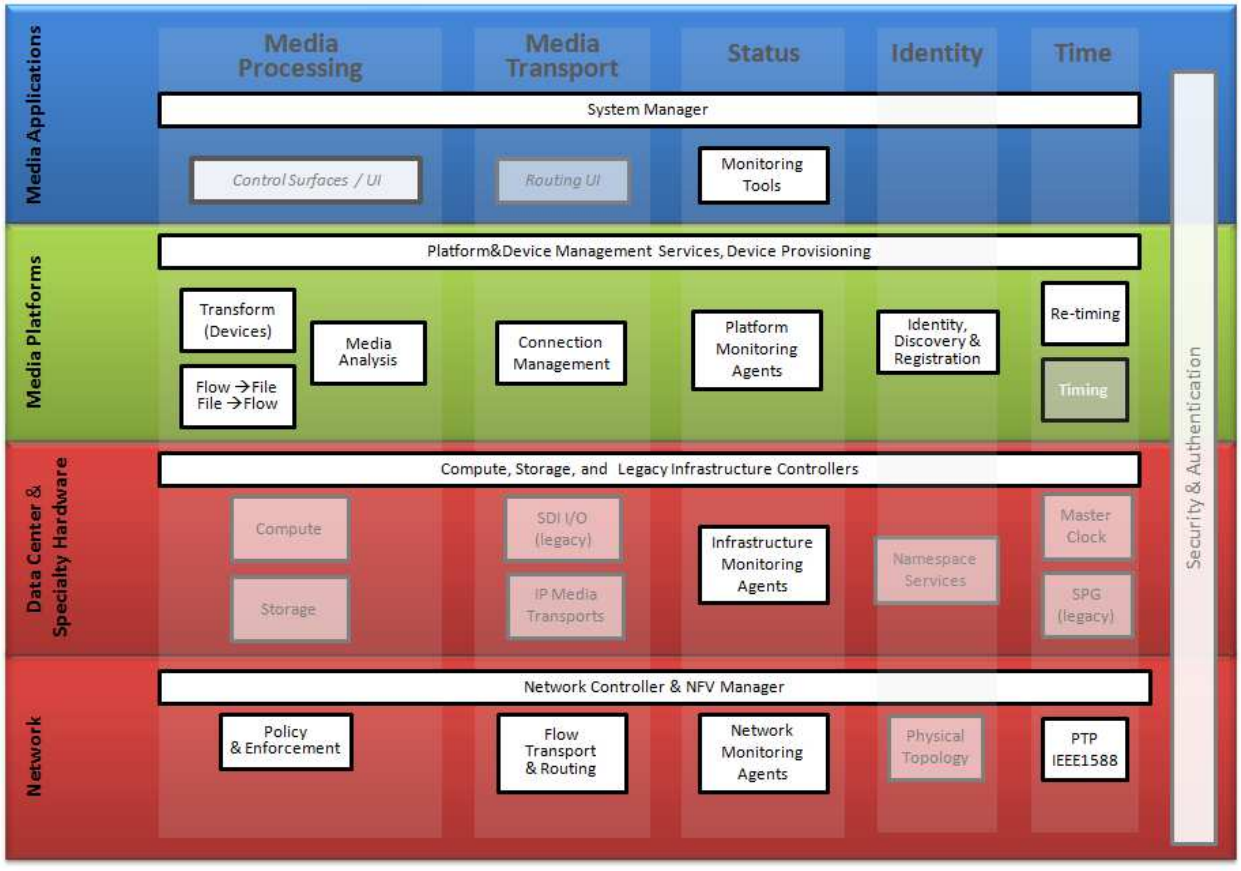
Document Status

This document is an AMWA Proposed Specification, and the project leaders request discussion and suggestions for improvements. Send comments to John Mailhot (john.mailhot@imaginecommunications.com).

AMWA Control / Monitoring / Management Architectural Sprint

At the request of the AMWA board, a team of volunteers, including end-users, network equipment providers, processing equipment providers, and related software subsystem developers was charged with the task of adding a layer of detail to the JT-NM reference architecture in the area of control, monitoring and management interfaces. This document is the result of a deliberative process over the course of several weeks, and seeks to add some additional details to the JT-NM RA.

As a first step, the team reviewed and modified the diagram below, augmenting the RA and inserting management entity placeholders which span the columns. This re-formulation of the capabilities diagram should not be taken as a prescript that a single monolithic management layer is required at each of these levels of the RA, but as an acknowledgement that there is an opportunity for management API at each of these levels of the abstraction.



In this diagram, each box represents a capability within the overall framework, and the long narrow boxes separating the layers represent a management point(s) across the layer.

Regarding Dynamic Device Instantiation:
Historically, facilities were composed of static devices of fixed functionality; it is widely viewed that new facilities will likely include a mix of tangible devices (whose function is fixed at the date of manufacture) and dynamically instantiated devices which appear, serve a purpose, and then are gone. During the time when the devices are instantiated, they will act and operate as "devices" in the paradigm and expose the typical device characteristics. At the times when these virtual devices are not instantiated, they remain latent capabilities but are not active or managed (except as required in order to instantiate them).

The Application-Layer "System Manager" capability includes managing the instantiation and dis-instantiation of the of these dynamic devices.

Management API Standardization for Composable Systems
In order to foster the integration and composition of large-scale systems, it is necessary (or at least helpful) that some of the basic management capabilities exposed by the layers adhere to some standard practices and common APIs.

For each of the Capabilities and Management elements in the diagram, the working group assessed the classes of management information which would likely be exposed as an API of the capability, and considered which of those management information classes was critical for the composition of large systems. For each of these critical management information classes, to the extent that there is a common industry approach (whether standard, or de-facto standard, or open specification) that approach is noted.

Capability/Element	**Management Information Class exposed through API**	**Is this Management Information class important to standardize?**
System Management (top layer)	Inventory of Current Resources (local names, URL/URI, etc)	Yes, enables other subsystems to find and interact with devices
	Inventory of latent (not currently instantiated) resource capabilities, and create/destroy dynamic device instances	Basic inventory and CRUD operations could be standardized in future work
	Resource allocation/association to users or activities	Topic for future work
Monitoring Tools	Aggregated Alarm Information	Yes, upstream notification
	Telemetry from devices (counters, rates, utilizations, etc)	Normalized telemetry in a standardized format/container would be helpful to composing larger systems/environments;

		however the underlying telemetry will be of many forms and meanings evolving over time.
Platform & Device Managers Device Provisioning	Basic Device Health & readiness to do work	Yes, enables high-level monitoring
	Inventory / Create / Destroy a virtual device with a named function	Yes (but the details of what the function *means* are very vendor-specific)
	Device Operational control capability	A neutral self-describing parametric and telemetry schema would be helpful in large systems. It will be useful to eventually standardize the form (but not necessarily the detailed semantics) of reported telemetry items. (pub/sub work in IETF may be relevant)
	Patterns/APIs for specifying parameters and telemetry	
	Telemetry	
	Details of the parameters	The actual semantic details of the parameters can be extremely vendor-specific and context-specific
Media Analysis Capabilities (whether embodied in a special-purpose device, or embodied within some other device)	Input/Output Flow Details	Yes, enables connection control and inventory of flows
	Summary results of analysis	Partial - helpful if there were a common reporting form for basic health of a signal (similar to ETR 101290 for TS signals).
	Detailed results of analysis	No, every analysis device has different things it can measure and analyze
Media Ingest/Playout Devices	Input/Output Flow Details	Yes, enables connection control of Flows and monitoring of Flows and Streams.
	Ingest/Playout Controls (mount/record/play/stop)	Yes, helpful for automation
	Ingest/Playout Status	Yes, helpful for monitoring
	Media File Formats	Some standard, some not

Connection Management	***Connection Group*** Requests & Status of existing requests	Yes, many entities ask to make and consume these and find status of what is connected to what.
	Inventory of Source Groups and Destination Groups.	Yes, many entities require an enumeration of these.
	Details and Methods of defining the Source Groups and Destination Groups.	Yes, an API to add/modify/delete these grouping constructs will be generally useful
	Connection Requests & Status of existing requests	Yes, many entities ask to make Connections from Sources to Destinations, or consume the status of what Source is presently connected to a Destination.
	Inventory of Sources and Destinations	Yes, many entities require an enumeration of the Sources or Destinations
Platform Monitoring Agents	Health of Platforms and availability to do work	Yes, useful to northbound monitoring systems
Device Identity & Registration	Device Inventory	Yes, in order to locate and inventory devices and capabilities (NMOS)
Signal Timing and Re-Timing	Input/Output Flow Details	Yes, enables connection management of Flows and monitoring of Flows and Streams
Compute, Storage, and Legacy Infrastructure Controllers	Inventory, Allocation, De-Allocation of resources	Yes, enables resource utilization by layers above (OpenStack API, VMWare, etc)
Infrastructure Monitoring Agents	Health of Infrastructure Elements, availability to do work	Yes
Network Controller	Create/Retrieve/Update/Delete provisioned Network Streams	Yes
	Aggregation of network health and status	Yes
	Aggregation and management of network topology (Physical and Logical)	Yes
Network Monitoring	Health of Network Elements,	Yes

Agents	Availability to do work	
	Health of Network Interfaces, packet counts, errors, etc	Yes
	Configuration of Network Elements	YANG/NETCONF
	Network Stream-Sample information	Yes (NetFlow, SFlow)
	Connectivity & Topology Information	Yes (LLDP information)
Network Flow Transport & Routing Capability	Network Stream-Sample information	Yes (SFLow/Netflow)
	Network Stream "join" and "leave" control protocol from edge devices	Yes (IGMP, …)
	Network Stream management control protocol (northbound)	No (existing switch northbound APIs)
	Configuration of network element (general config)	No (every switch is different) Configuration can be done locally or abstracted through controller
	Network Policy & Enforcement	No (every switch has different capabilities) can be done locally or abstracted through controller

In the context of developing the diagram and table above, the committee also found it helpful to clarify some of the terms of the RA and define some additional terms:.

Source -- defined in the RA as "*an abstract concept that represents the primary origin of a Flow or set of Flows*" A ***Source*** is an abstract origin of essence or data; the ***Source*** is represented in one or more ***Flows***, and these ***Flows*** may be transported over the network as ***Streams***. Each of the ***Flows*** might be in a different codec or format, but they are merely different representations of the same ***Source***. Each ***Flow*** from the same ***Source*** is considered "Editorially Equivalent".

Flow -- defined in the RA as *"a sequence of Grains from a Source; a concrete representation of content emanating from the Source"*. A ***Source*** may provide different qualities (representations) via separate ***Flows***, such as uncompressed, mezzanine and/or proxy. Each ***Flow*** contains exactly one representation of the ***Source***.

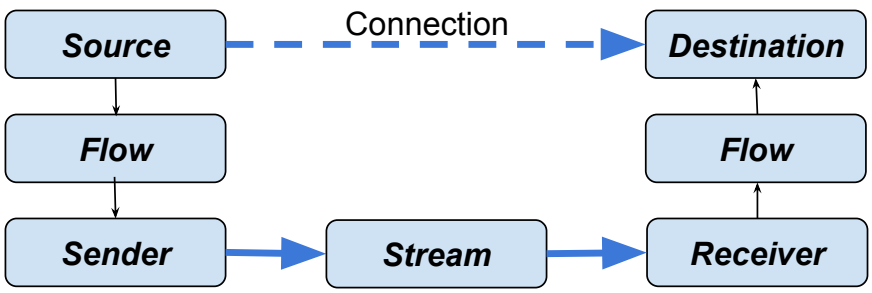
Stream (not currently defined in RA) -- a realisation of a ***Flow*** using a transport protocol, for example a TR-03 video, audio or ancillary data stream (using RTP) or an DASH stream (using HTTP).

Sender -- defined in the RA as "*makes a Flow available on the network".* In this document, we clarify that a ***Sender*** presents exactly one ***Flow***, packaged as a ***Stream***, into the network. A ***Source*** may be represented through multiple ***Flows***, mapped by ***Senders*** into ***Streams***, each offering a different representation of the same original ***Source***. A ***Sender*** is a component of a ***Device***, which may contain multiple ***Receivers*** and ***Senders***.

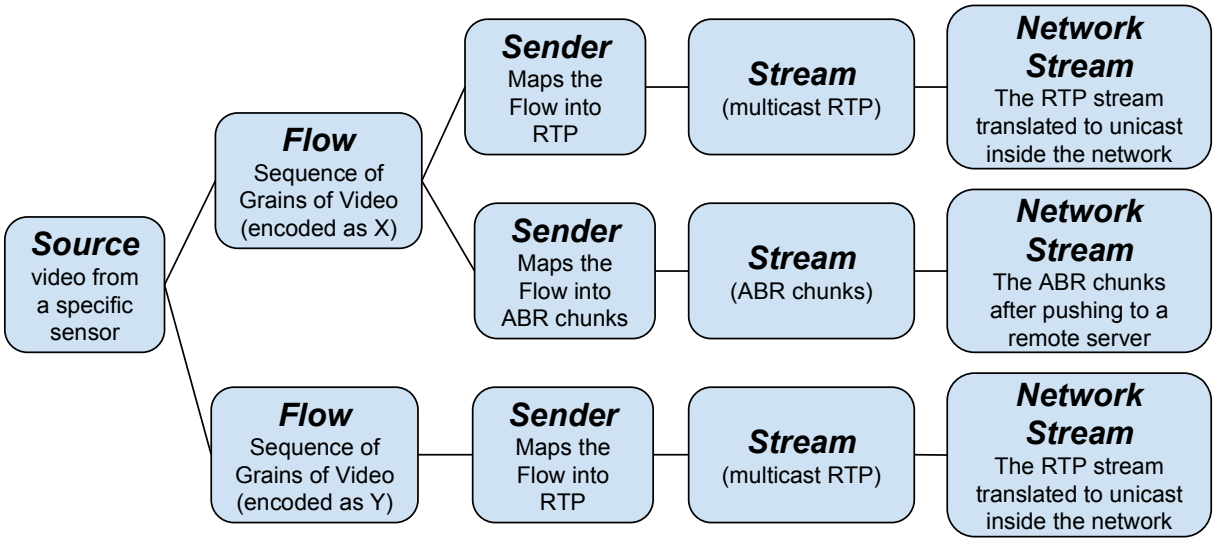
Receiver -- defined in the RA as "*consumes a Flow from a Sender".* A ***Receiver*** receives exactly one ***Stream***, which contains a ***Flow***. A ***Receiver*** is a component of a ***Device*** which may contain many ***Receivers*** and ***Senders***. A ***Receiver*** may be technically capable of reception of multiple formats, but a ***Receiver*** receives only one format at a time. A ***Receiver*** is associated with a ***Destination***.

Destination (not currently defined in RA) -- A logical consumer of essence or data; the abstract complement of ***Source*** which is defined above. Whereas a ***Source*** can be expressed at many "qualities" through multiple ***Flows*** expressed as ***Streams*** from multiple ***Senders***, at any given time a ***Destination*** is fulfilled through one ***Flow*** via one ***Stream*** through one ***Receiver***. When a new connection is requested from a ***Source*** to a ***Destination***, a specific ***Flow*** and one of its associated ***Streams*** is selected based on the exposed capabilities of the ***Receivers*** associated with the ***Destination***.

Connection (not currently defined in the RA) -- A ***Connection*** confers the essence from the ***Source*** to the ***Destination***. This is done in practice by selecting the most appropriate ***Flow*** and related ***Stream*** to convey to the Receiver.



Network Stream (not currently defined in RA) -- The specific embodiment of the ***Stream*** as it transits each link within the Network. The ***Network Stream*** may be subject to header translations, encapsulations, and other modifications as it transits the Network. Systems which are analyzing telemetry from the Network may need to be aware of the specifics of these modifications within the network in order to relate the ***Network Stream*** to the original ***Stream***.



Source Group (not currently defined in the RA) -- Frequently in television production and distribution, a collection of essence ***Sources*** are assigned a production-meaningful name -- for instance the video essence from a camera, and the audio essence from a reporter's microphone, might be grouped together logically and called "REM 2" when used operationally in the production. This ***Source Group*** may include ***Sources*** across multiple ***Devices*** - the grouping relationship is purely a control-systems concept established for the benefit of the production operation. Multiple representations (***Flows***) of the essence ***Sources*** within the ***Source Group*** may be available. Each ***Source*** may be part of many different ***Source Groups***.

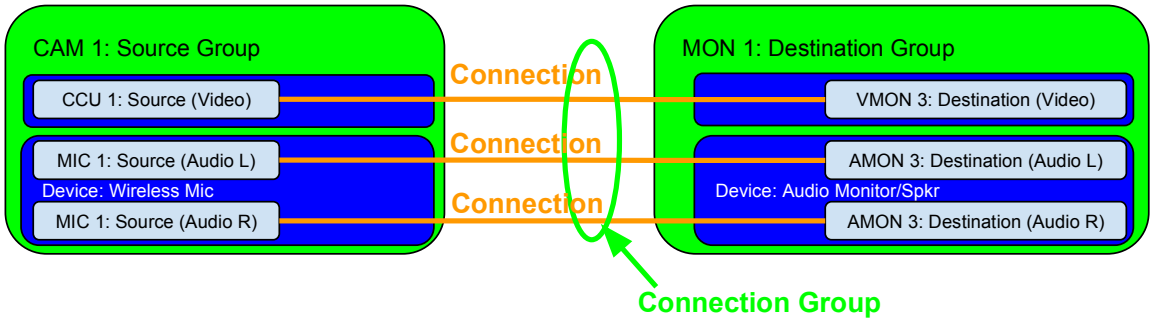
Destination Group (not currently defined in the RA) -- analogous to the ***Source Group***, a ***Destination Group*** is a production-meaningful grouping of ***Destinations***, which may (or may not) be parts of different ***Devices***.

Connection Group -- A ***Connection Group*** confers the essence ***Sources*** of the ***Source Group*** to the ***Destinations*** of the ***Destination Group***. More specifically, it is a set of ***Connections*** which convey ***Flows*** from ***Senders*** representing the ***Sources*** in the ***Source Group*** to the ***Receivers*** associated with the ***Destinations*** of the ***Destination Group***.

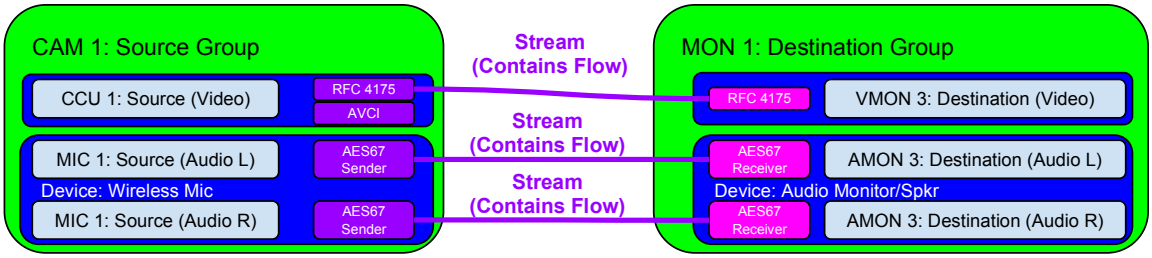
Connection Management (not defined in the RA) is the capability which accepts and implements requests from higher-level controlling entities, and implements Connection Groups and Connections. The ***Connection Management*** capability is responsible to coordinate the lower layers including the ***Network Controller***.

Network Element/Interface/Point (not currently defined in the RA) -- these terms refer to the network infrastructure itself.

Network Controller (not defined in the RA) is the management entity which represents the Network as a whole.

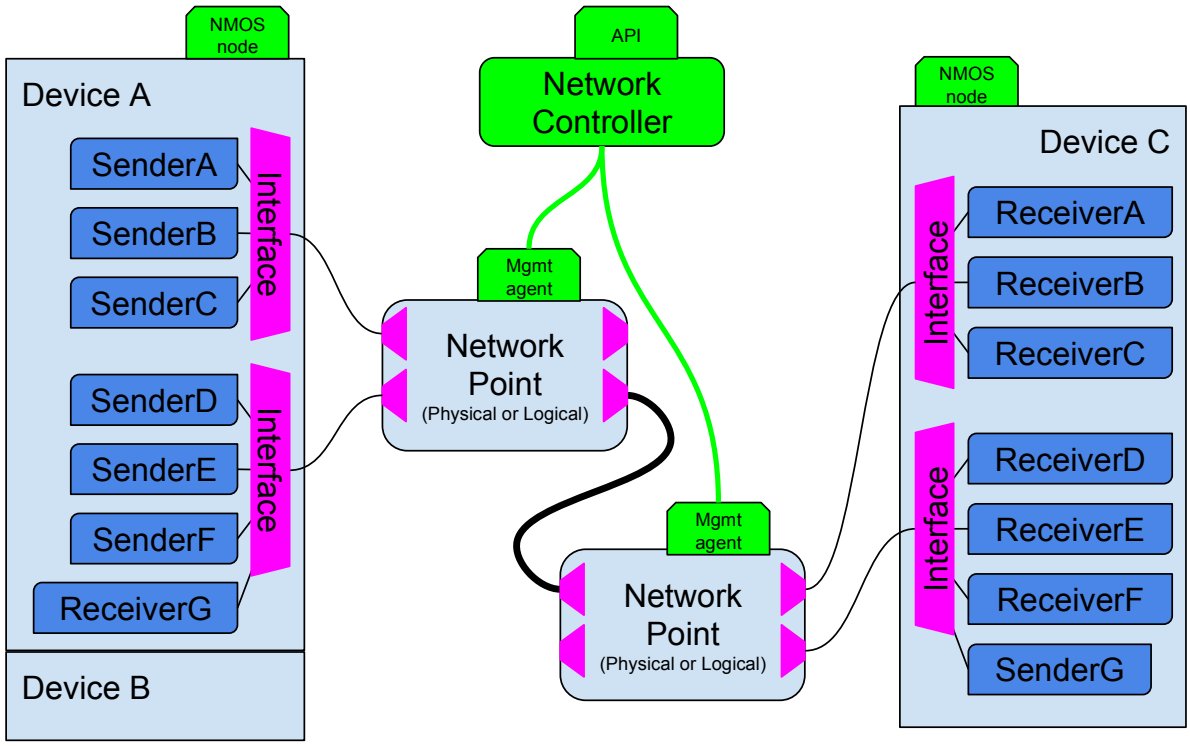


Connections are between Sources and Destinations



Flows (packaged as Streams) are between Senders and Receivers

In discussion, the committee found the following diagram useful in exploring the interaction of the Network with the Nodes, Devices, Senders, and Receivers defined under the JT-NM. Note that any of the Devices or Network Points can be physical or virtual manifestations.


NMOS node
Device A
SenderA
SenderB
SenderC
Interface
SenderD
SenderE
SenderF
ReceiverG
Interface
Device B
API
Network Controller
Mgmt agent
Network Point
(Physical or Logical)
Mgmt agent
Network Point
(Physical or Logical)
NMOS node
Device C
Interface
ReceiverA
ReceiverB
ReceiverC
Interface
ReceiverD
ReceiverE
ReceiverF
SenderG