An Open Approach to Media in the Cloud

Chris Lennon

ROSS

CLOUD COMPUTING FOR LIVE MEDIA PRODUCTION

With the increase in power available in cloud computing it has started to become feasible to do live media production in cloud compute environments

NEED FOR MORE SEAMLESS MULTI-VENDOR INTEGRATION

Today's environments typically encompass a combination of on-premises and cloud environments. Solutions span a variety of platforms, and must interoperate, regardless of where they are hosted and who made the application

SECURITY EMERGING AS A PRIORITY

Interest in securing media production technology is on the rise, with more and more organizations prioritizing secure access to their on premises and cloud deployments.

The Goal

Make it easy to secure, connect and control a multi-vendor ecosystem of media processing services and microservices.

What is needed?

Standardization of communication methods between (micro)services and full products – designed for hybrid cloud and on-prem solutions

Some Design Goals

Meet the demanding requirements of live video processing. And work for others!

- Generic enough to satisfy a wide variety of needs
- Multi-Vendor simultaneously
- Powerful enough to satisfy complex and diverse needs
- Efficient enough for live usage and large-scale deployments
- Plug and play while operational
- Secure enough for hybrid cloud and ground environments

ROSS

Catena

A proposed standard to address interoperability in control for Media Services

ca·te·na

- /kəˈtēnə/
- a connected series of related things

The basis for Catena is already Proven

Evolution of Catena

- The openGear protocol (OGP) is widely used within Ross products as well as in partner organizations.
- OGP is fully Ross owned and controlled (benevolently!) and was designed and used for multi vendor solutions from its inception.
- With the rise of cloud compute and microservices we looked at what changes we would want to do to the openGear protocol for its next incarnation.

ROSS

Evolution of Catena –
open source

- As part of creating this evolution from OGP into Catena we also decided to open it up to the world as a **standard** as this would benefit the entire industry.

- We are doing this work within the OSA, and once ready, taking it to SMPTE for full standardization

- We also plan to create open-source libraries implementing the Catena specifications for easy adoption by anyone who would like to wants to

What is Catena?

Standard suite will define:

- **Registration**: Service registration and capability announcements
  - Control and Status Parameters, Commands/API
- **Status**: Methods for sending and receiving status announcements
- **Control**: Methods for changing parameter settings + receiving updates
- **Commands/API**: Method for issuing commands to services
- **Security**: Based on zero trust tenets
- **Orchestration**: Catena devices must be able to plug into an orchestration system

*** Catena is NOT an orchestration system, this is about plugging INTO an orchestration system*

ROSS

What kinds of services does Catena apply to?

- Catena can be used with a wide variety of service sizes.
  - An entire application, “traditional” broadcast applications such as video servers, production switchers, audio consoles etc..
  - Catena is designed to work with cloud environments in mind
  - Catena is also designed for on-Prem environments
  - Catena is designed to work in hybrid environments where some services are cloud based and some are on-Prem

ROSS

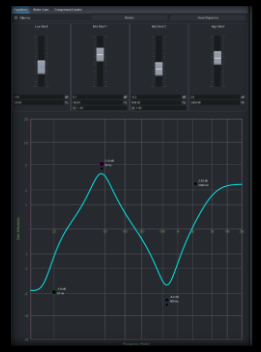
Key Ingredients

- At its core is a device
- A device can be a very large physical product (with millions of parameters), or a very small micro service in a cloud instance with only a few settings.

Large software applications (onPrem or cloud hosted)

Large Complex onPrem devices

Small cloud microservices



Small onPrem devices

Devices that don't even touch the media
(The robot, in this example)

Key Ingredients – Data Model

Catena's data model enables media processing services and devices to describe themselves to clients.

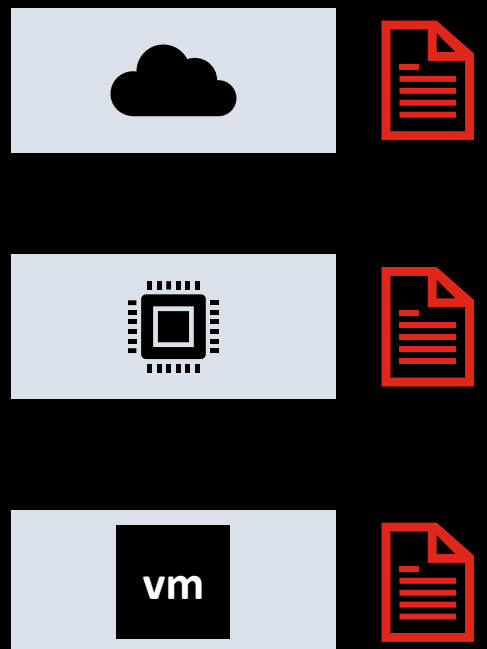
Parameters and Status values are described using the Catena protocol to include information such as:

- Data type + constraints

Commands are described as to how many parameters and which data types are used for the command.

VERY IMPORTANT: This information is provided to other devices and services during operation to enable plug and play functionality.

Systems made up of many different services do NOT need to have developers reprogram them to add functionality.

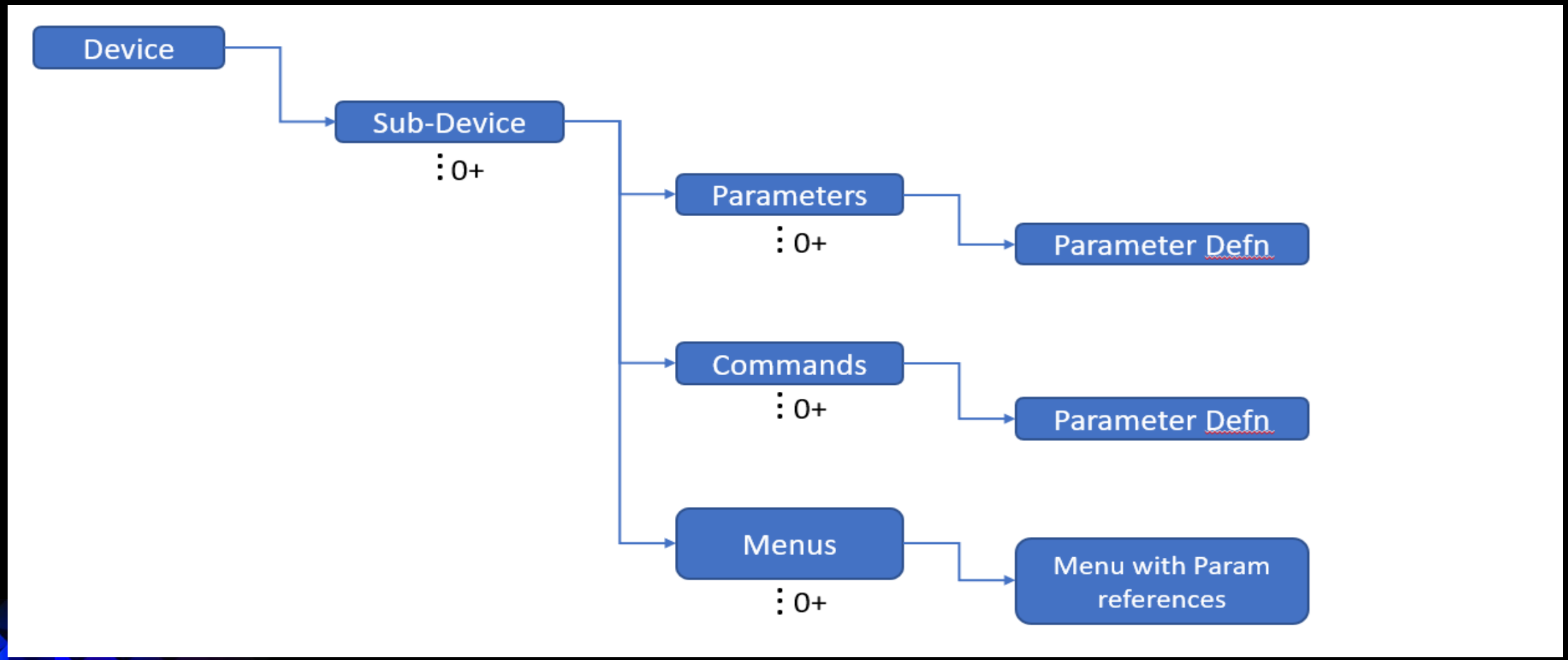


Key Ingredients – Device definition

- Devices must respond to a query for them to describe themselves.
  - This response includes a description of all of the parameters, status information and APIs/Commands that the device supports.
  - It is important to note that this happens while a device is running. i.e., it is truly plug and play during operation; The requester of the information does not need to know anything about the device ahead of time to be able to control it
  - Devices can include sub-devices in a hierarchical tree

ROSS

Key Ingredients – Device definition



Key Ingredients – Device definition

A Parameter is a value (or set of values) representing items in the device.

Parameters have:

- Type (Integer, Float, String etc.)
- Can be single value or more complex types (Arrays, Structures)
- Can have constraints (min value, max values, List of values with text etc.) (ex of a list constraint: a parameter which specifies the video format)
- UI Widget hint

ROSS

Key Ingredients – Difference between parameter and command

- A parameter has a value that can be read, some may be modifiable, some may only be status values that are read only.
- Commands are instructions to be executed, they may have 0 to many parameters as part of the instruction to be executed.
- This is to say that parameters have a state, while commands are stateless
- All parameters and commands have unique names (within the device!) and referenced as an OID (Object IDentifier) which is a text name/identifier

Example service

Let's imagine we have a service that can play one channel of video.

The service can be discovered and will respond to queries about its capabilities. A very simplified service might have:

- **Status Parameters**: The device may provide parameters on playout state, audio meter values, sync status, etc. (parameters that can be read only)
- **Control Parameters**: Inpoint and outpoint may be values that can be read and modified.
- **Commands/API**: Play (speed x), Stop, Load Clip (clip name, inpoint, outpoint)

Catena gives another Catena service, or a control/automation system the ability to get this definition and to be able to control this service (with proper security).

ROSS

Asynchronous updates and Subscriptions

Catena is designed to scale to large systems with many devices under control and many systems viewing and controlling devices.

- Without this approach, the network and device load in complex and large environments can bring the network and the device to a halt.

External controllers can subscribe to a device, and to sub elements within a device so that whenever the OIDs that have been subscribed to change, the device sends out those updates.

- This vastly reduces the network traffic vs. forcing systems to poll a device for updates, and by allowing controllers to only get updates on the OIDs that are pertinent to them

Security

Supporting the secure transport of data, commands and data at rest

Key Ingredients - Security

Catena's security specification follows zero-trust industry standard recommendations to achieve:

- traffic management (availability)
- secure communications (confidentiality, integrity)
- observability (audit, availability)
- Aiming to leverage / reuse the work done by AMWA on defining a profile for OAuth2 that can drive multi-vendor interoperability (IS-10?)

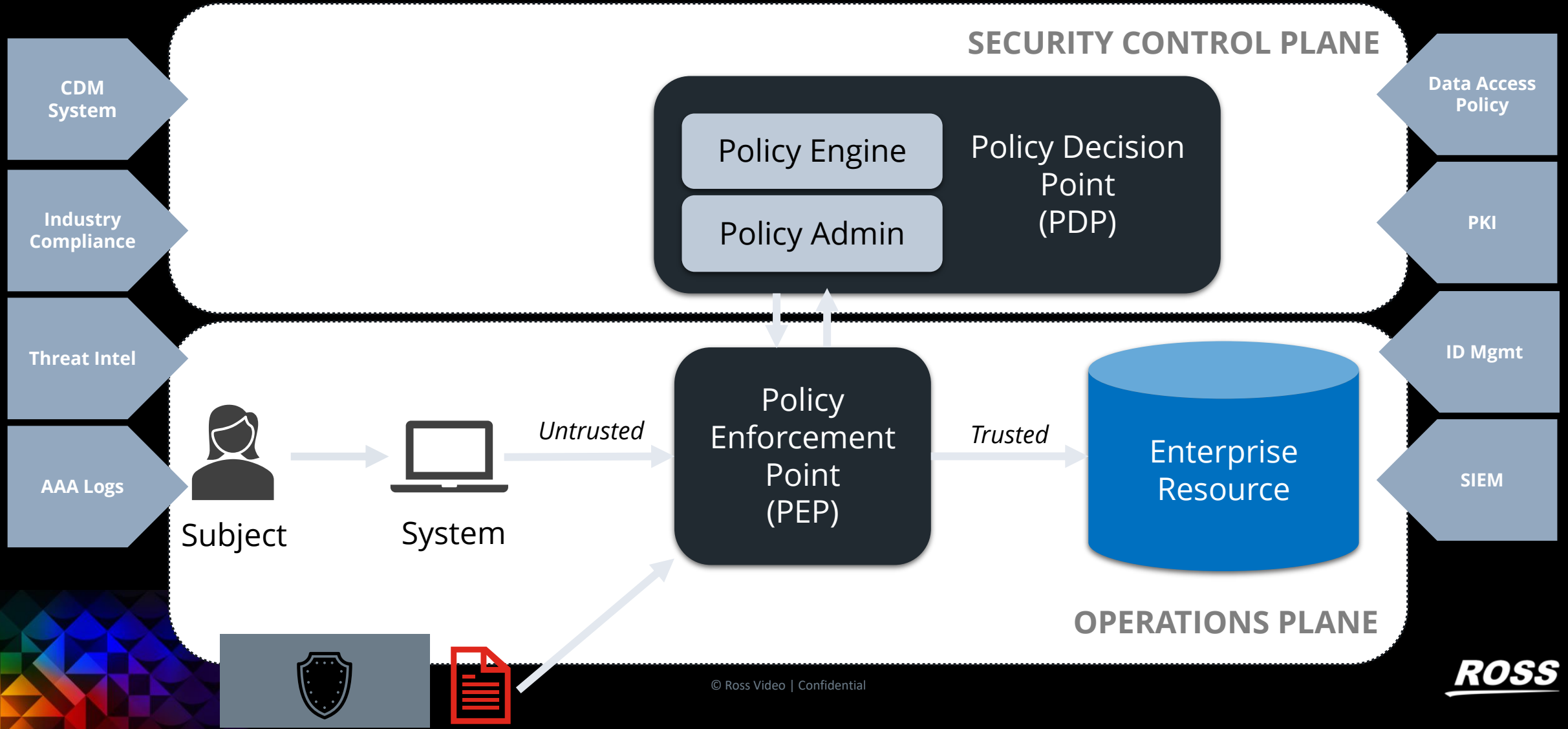
Value to Industry

Security

- Intent is to define an architecture that will be a “good citizen” within a zero-trust architecture as defined by NIST SP 800-207
- Adoption of ZT is a major trend within IT in general, we see value in aligning with this trend.
- Aiming to leverage / reuse the work done by AMWA on defining a profile for OAuth2 that can drive multi-vendor interoperability (IS-10)

Scope of Standard

Security



Overviews of Proposed Content

Security

- Definition of Policy Enforcement Point interactions with Data Model
  - strip items to which access is not granted
  - mark “read only” items to which write access is not granted
  - reject requests to change parameters or issue commands to items outside of permitted scopes
- Performant enforcement (checking every access with the Authz Server is likely bad)
- Probably a number of EGs or RPs around:
  - Image / Repo / Orchestrator / Container / Host threat environment for µservice based media processing services
  - Minimum requirements for choice of Authorization Servers

ROSS

Authentication

- We are looking at using IS-10, or its underlying technology OAuth2 as this is a well accepted security technology
- OAuth2 requires every client / device pair's access scopes to be mapped against user Roles
- Each device will generally have unique and potentially complex access scopes
- Clients x Devices x Access scopes quickly becomes a large (10k +) number.
  - administration complex and thus error prone
  - subtle and hard-to-spot chinks in the armor all but inevitable

ROSS

Proposal

Grand Unified Scopes Theory: Optimized for Media Production

"GUSTO"

ROSS

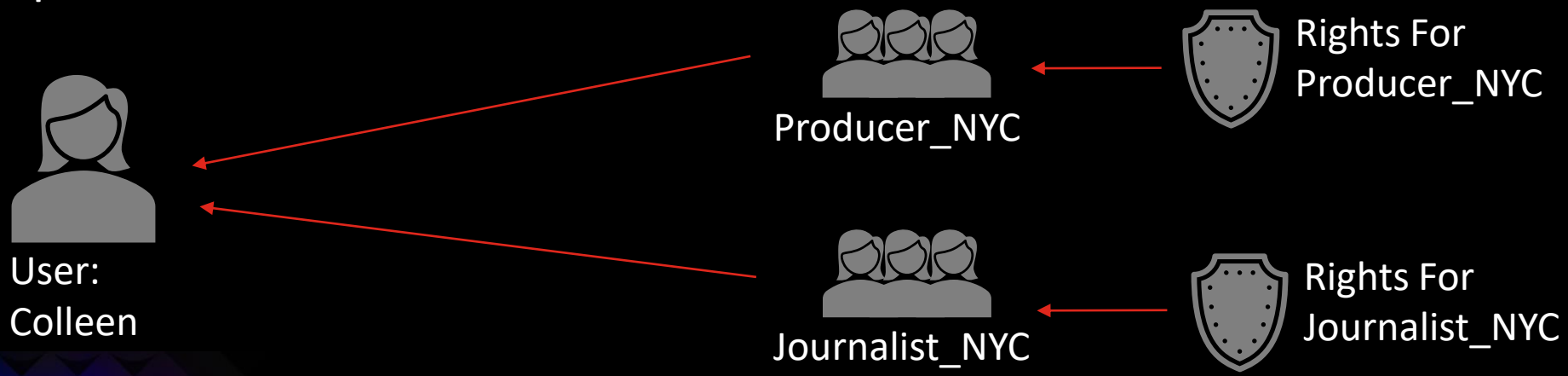
Reminder on RBAC: (Role Based Access Control)

A (smallish) set of Roles are created, example: Journalist, Producer, Technical Director. SecOps etc.

These roles have access rights assigned to them. I.e. A journalist can access the newsroom system operationally, but not a production switcher etc.

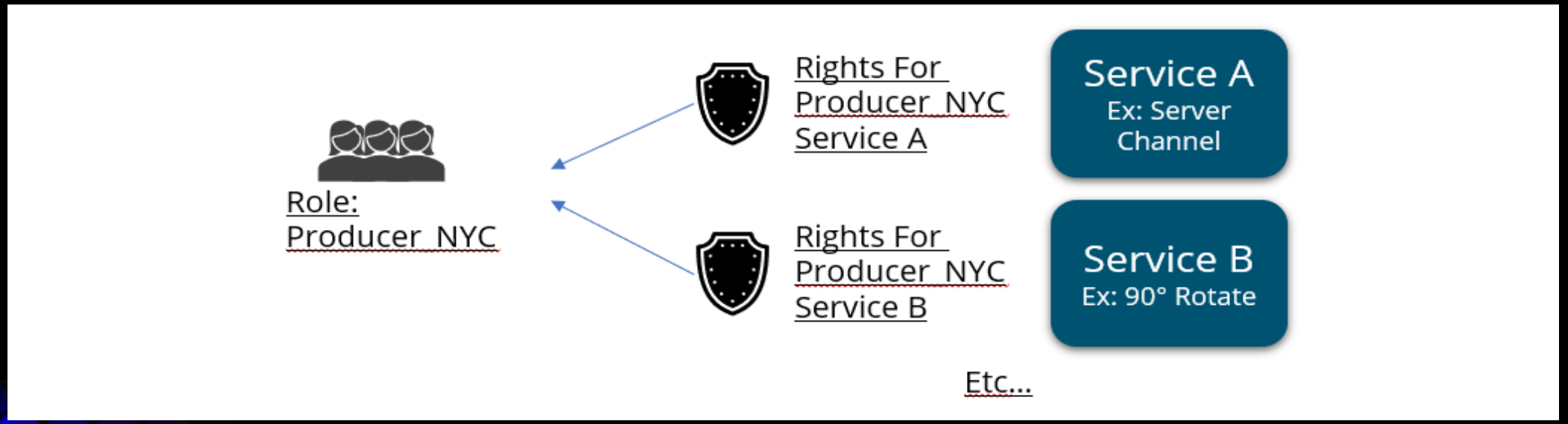
Users are then assigned to one or more roles

Example:



GUSTO

- Within a device, parameters and commands are all given ‘risk level’ from a pre-defined small set of classifications, these are part of the parameter and command definition
- These classifications can now have access rights mapped to them from the roles defined
- Roles can be assigned different access rights to different physical versions of the same device.
  - (i.e. TD_NYC may not be given rights to equipment in the LA office)



Transport

Supporting communication between devices

Key Ingredients - Transport

Catena’s transport specification supports plug and play communication between devices.

Catena’s transport specification supports plug and play communication between devices.

Includes the following capabilities:

- Data serializer
- Data de-serializer
- gRPC transport (Google Remote Procedure Call)

Overview of Proposed Content
TRANSPORT

- Proto3 definitions
  - serializer
  - de-serializer
  - Proto-files for several languages, including at least Java, C++, Go, Python
- gRPC definition
  - Bi-directional mode of operation (persistent and/or on-demand)
    - Server push (i.e., audio levels values)
    - Client push (i.e., setting parameters, loudness level)
    - Defining/limiting scope based on user rights
  - Monitoring and management of Client/Server connection
  - Multiplexing over a single socket connection
  - Port definition
  - Certificate management for TLS connectivity
  - Negotiation for non-encrypted connectivity for ultra-light clients.

ORCHESTRATION

Supporting management and configuration of technologies

Scope of Standard Orchestration

For our purposes we are defining Orchestration as:

- Everything that needs to be done until a device is ready to used in normal operation (within reason to what could be considered Orchestration…)
- Monitoring the health status of an application. Note this is verifying if the application is running, not whether signals are non-sync etc..
  - And the ability to repair (if possible) these errors

There are existing IT systems that perform these tasks already, it likely makes sense to not create something that is already solved

Open Question: For complete compatibility, should we be choosing/ recommending an existing technology stack as part of Catena for these?

ROSS

UI / MENU HINTS

Catena has 2 features that provide a lot of utility for secondary systems to provide additional functionality.

1: Each OID (param and command) includes a ‘UI Hint’ as part of its definition. (dropdown, vertical slider, row of buttons, etc..)

2: There is an optional ‘Menu’ section in the device descriptor that allows a device to specify a simple set of menus, and which OIDs are attached to each menu.

With this information, *simple* user interfaces can be created by secondary systems without having to have the full featured UI provided by the device. These UIs as seen in the Orchestration slide can even be used while the device is offline.

ROSS

UI / MENU HINTS

Its important to note that these menu hints does not replace a full device UI in most cases, it will lack the polish that most device creators would want in most situations.

**Note, for those familiar with OGP, there is a UI markup language OGLML (open gear markup language) that we are currently not planning to make part of the Catena specification. OGLML is 100% compatible with Catena still for those who know it and would like to continue to use it.

ROSS

Discovery

Control systems and other devices that want to use Catena to control / view the status of a device need to know it exists, and at what IP address/Hostname.

Systems can of course be configured manually with this information.

For discovery of Catena devices automatically we are looking at whether we can use the discovery mechanism in IS-04, or the same technology at least that IS-04 uses

Final thoughts

We believe that Catena has huge value to the industry, it has the promise of creating a true multi vendor solution for media devices a reality. It is not restricted to cloud usage, or on prem but has been designed to work for both as well as hybrid environments

Catena is based upon methodologies that are field tested and proven already.

Please help make this vision a reality with the OSA, we'd love your support for the project, and, even more, join the review of the draft documents, then the open-source projects when they start

ROSS

Timeline