Introduction to AES70

Ethan Wetzell

Platform Strategist

Bosch Communications Systems

AES70 Concepts

Why?

- Scalability
- Flexibility
- Security
- Extensibility

- Usable for large-scale, mission-critical applications, but also suitable for smaller, simpler products

Who uses it?

- Products on the market today from:
  - Beckhoff Automation
  - Bosch
  - d&b
  - Dynacord
  - Electro-Voice
  - Fitcan
  - Focusrite
  - RTS
  - And more

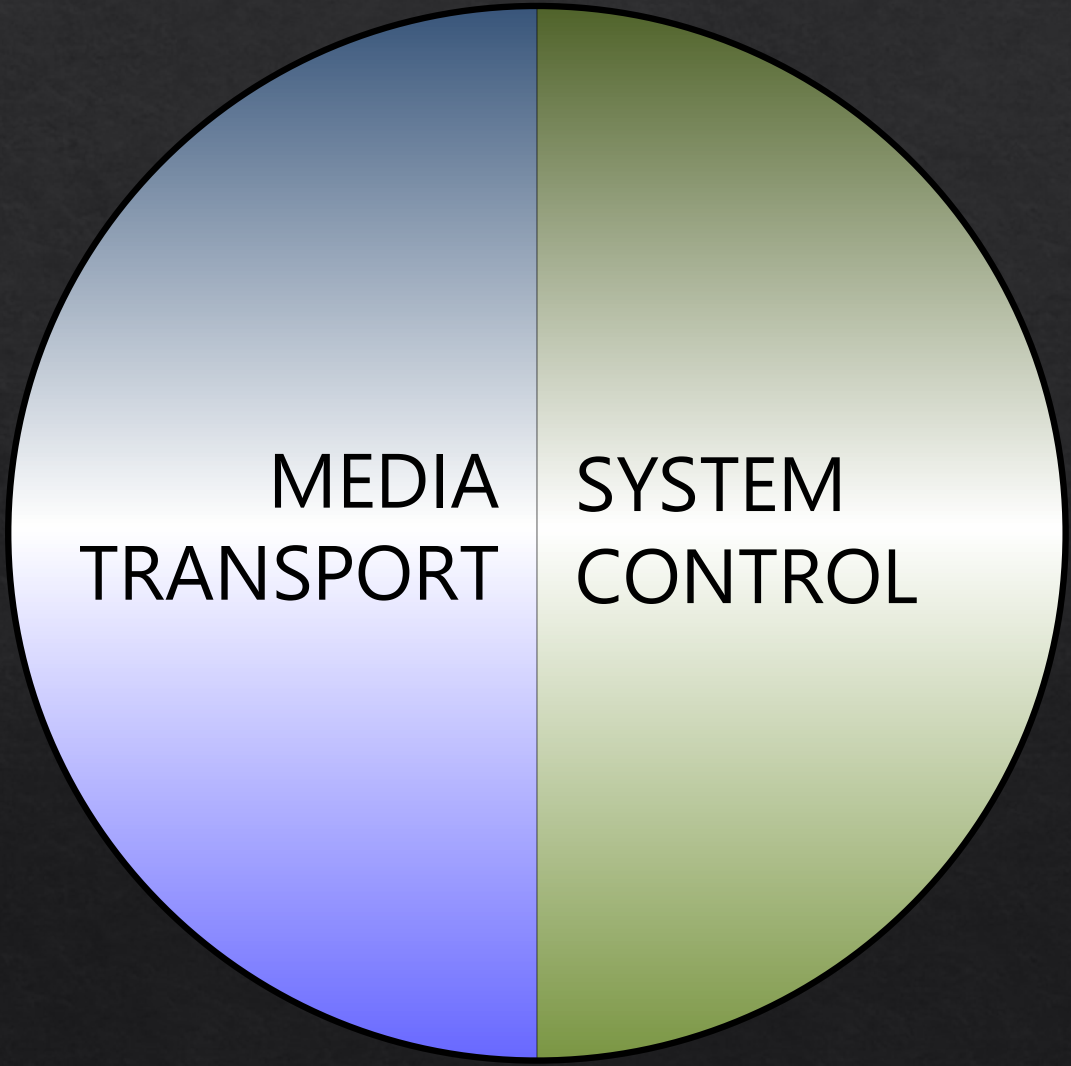
AES70 Concepts

What's in AES70

- An object-oriented framework for control interfaces that audio devices present to a data network;
- A standardized device object model for controllable devices;
- A rich and extensible repertoire of control class definitions that represent the signal processing, control logic, and network connection functions of modern audio devices; a class may be thought of as an API for a particular function or set of functions.
- An application protocol called OCP.1 ("Open Control Protocol 1") that defines command and response formats and sequences for control and monitoring of OCA-compliant devices over IP networks.

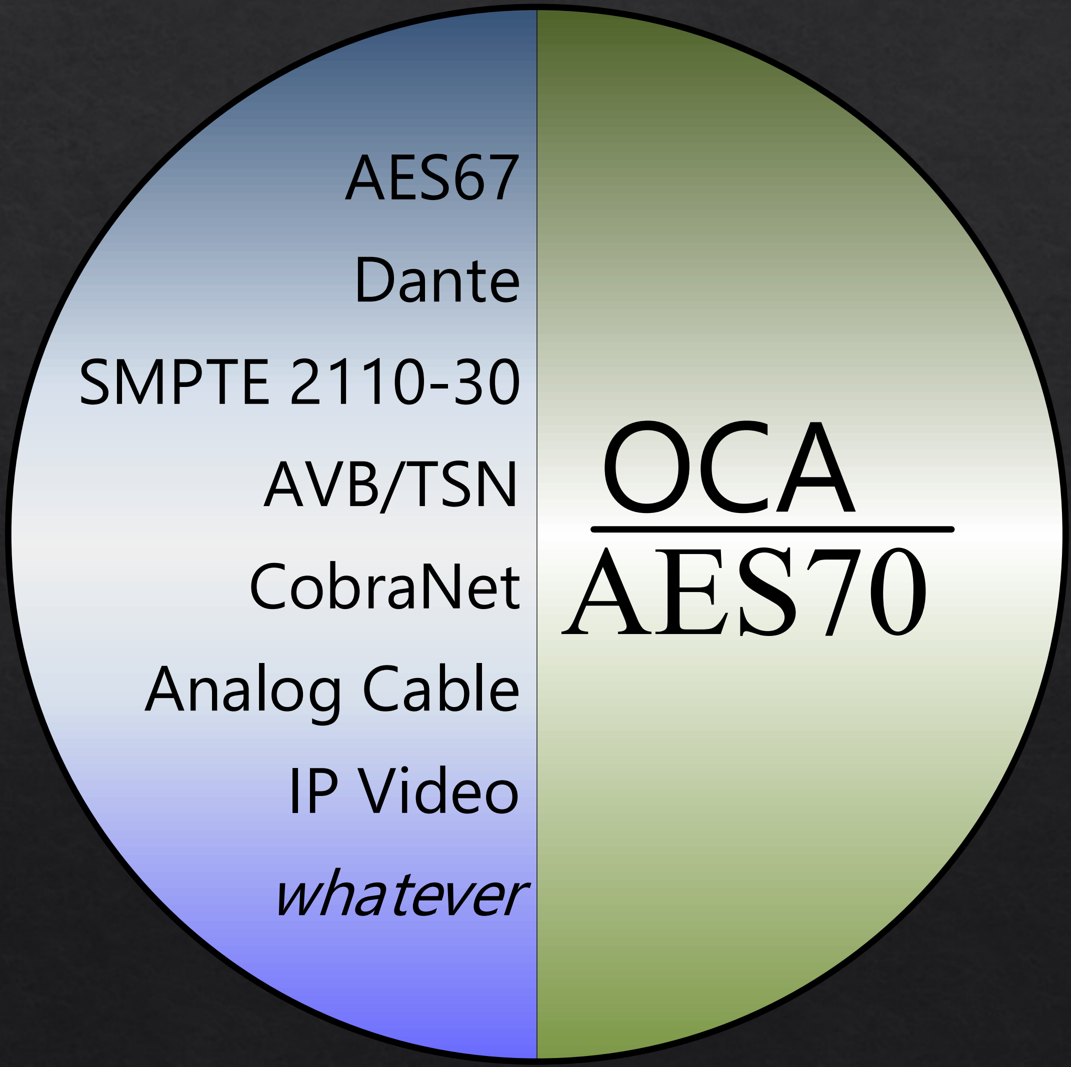
What's *not* in AES70

- Audio program transport;
- A programming model for OCA-compliant devices;
- A user interface definition or generation scheme for OCA-compliant devices;
- Standardized semantics for controllable elements (e.g. standard filter shapes);
- Standard device profiles (e.g. "standard mixer", "standard power amp").

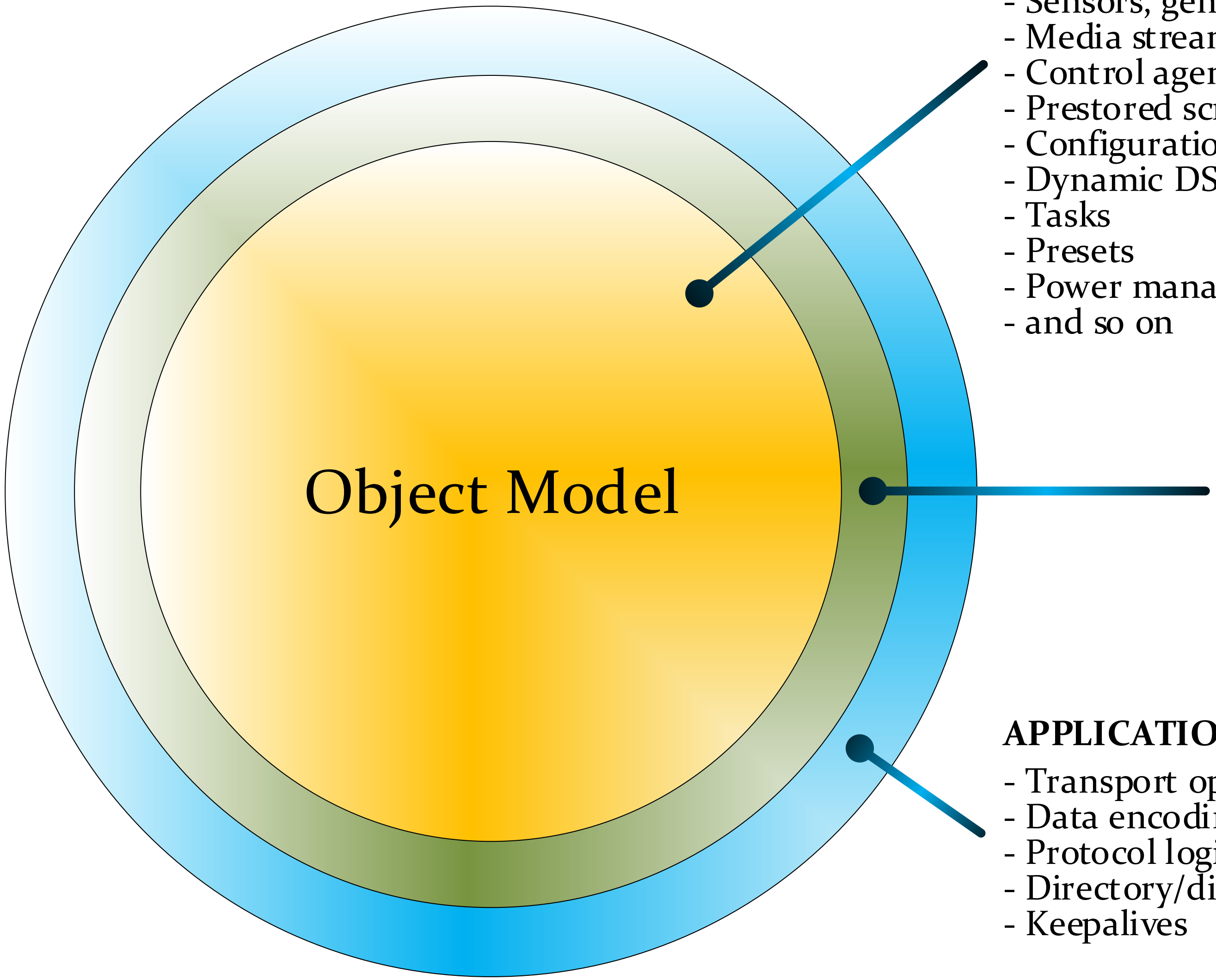


MEDIA TRANSPORT

SYSTEM CONTROL


AES67
Dante
SMPTE 2110-30
AVB/TSN
CobraNet
Analog Cable
IP Video
whatever
OCA
AES70

AES70



OBJECT MODEL
- Actuators, generic and audio-oriented
- Sensors, generic and audio-oriented
- Media stream connection management
- Control agents
- Prestored scripts
- Configuration libraries
- Dynamic DSP reconfiguration
- Tasks
- Presets
- Power management
- and so on

CONTROL FRAMEWORK
- Commands & responses
- Events & subscriptions
- Media stream connections
- Device control model
- Security
- Robustness
- Scalability

APPLICATION PROTOCOL(S)
- Transport options
- Data encoding options
- Protocol logic
- Directory/discovery interface
- Keepalives

Objects

Categories of Control Objects

- **Managers** Standard housekeeping objects, mostly the same in every device.
- **Workers** Objects that correspond to audio processing control functions.
- **Agents** Devices that provide various control functions or modify the control command stream, but do not map directly to signal processing elements.
- **Networks** AES70's connection management feature set.

Elements of Control Objects

- **Properties** Variables that define the state of the object
- **Methods** Operators that change properties and cause actions
- **Events** Signals emitted by objects to indicate state changes
- **Object Number** (aka ONo) Unique identifier of object within the device

Classes

- Templates from which control objects are created.
- Every class is uniquely identified by a **class ID**, a structured identifier used in various ways throughout AES70.
- AES70 classes inherit elements in the standard object-oriented manner. Only simple inheritance is supported.
- The set of OCA classes (aka "class tree" aka "OCC") defines AES70 's functional repertoire.
- The class tree will evolve over time to accommodate new device types and new manufacturers.
- Class tree inheritance rules create a constrained evolution regime that maximizes upward compatibility and ensures graceful evolution through orderly class specialization.
- Class tree inheritance rules support the addition of proprietary classes to the class tree in a way that maximizes compatibility with the standard classes
- Each class definition maps algorithmically to a particular control protocol data unit (PDU) format. PDU definitions devolve automatically from class definitions by a set of formatting and marshalling rules defined in the standard.
- AES70 currently defines a control protocol ("OCP.1") for IP networks. In the future, other network types ("OCP.2", "OCP.3", etc.) may be supported. Future network types may or may not use the same control PDU formats, but they will all have the same object model.

Events

- Event: transient state of an object that can cause it to send one or more event notification messages.
- Events have class-specific types. Each class may have a repertoire of events of various types.
- Event definitions are inherited.
- The most commonly used event is OcaPropertyChanged, an event that causes an object to emit a notification whenever a value of any of its properties changes.
  - OcaPropertyChanged is an event of the root class OcaRoot, and is therefore defined for all classes in the tree.
- Notifications are sent only to subscribing objects.
- Subscriptions are registered with and managed by the Subscription Manager.

AES70 Object Model

Device Model

REQUIRED MANAGERS

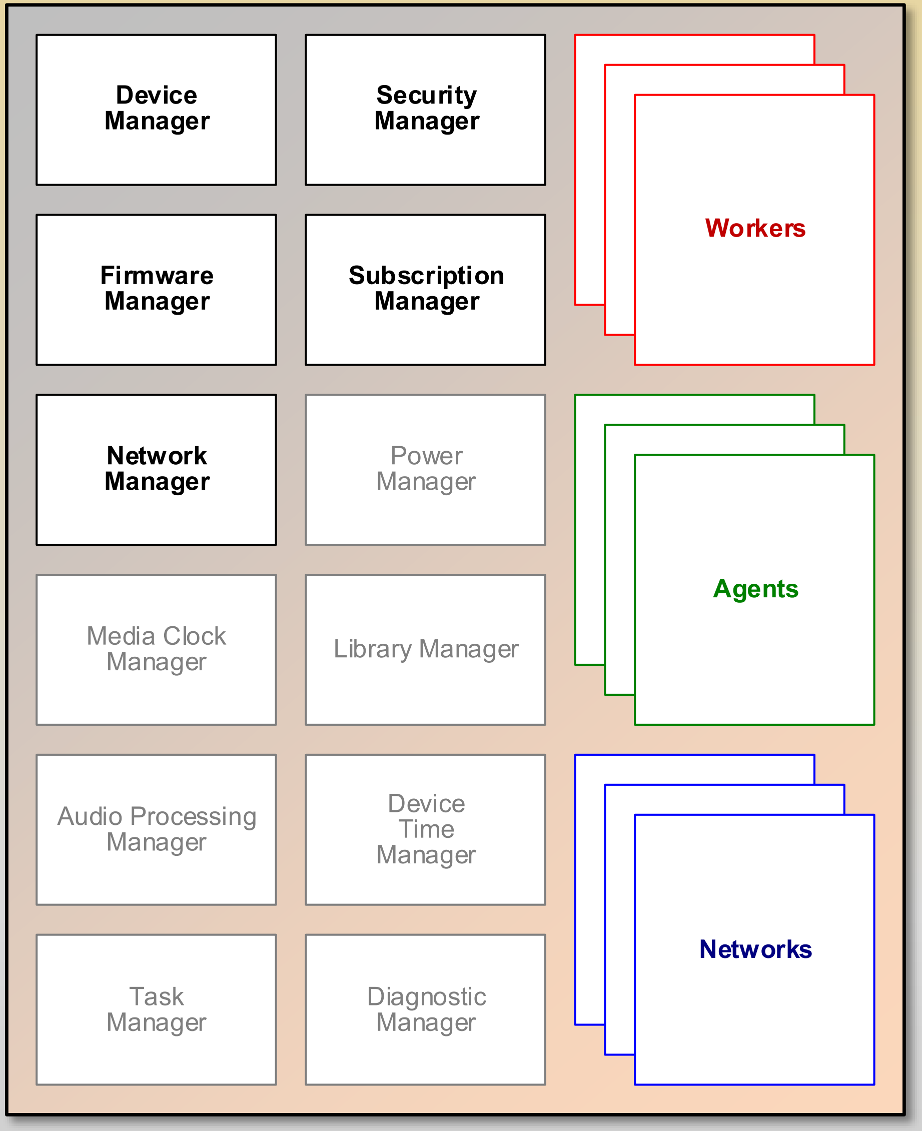
Device Manager
Manages information relevant to the whole device.

Security Manager
Manages security keys.

Firmware Manager
Manages firmware versions and, optionally, updates.

Subscription Manager
Manages event subscriptions.

Network Manager
Manages connection(s) to network(s).



OPTIONAL MANAGERS

Power Manager
Manages power supplies and batteries.

Media Clock Manager
Manages media clocks.

Library Manager
Manages stored parameter settings.

Audio Processing Manager
Holds global signal processing parameters.

Power Manager
Manages power supplies and batteries.

Device Time Manager
Manages time reference objects.

Task Manager
Manages stored processing sequences.

Diagnostic Manager
Offers features to help installation and setup.

Class Tree Overview

Workers	*Classes that deal with audio processing*
Actuators	*Classes that control audio processing*
Sensors	*Classes that monitor the device*
Blocks and Matrices	*Classes that define device control and processing groups*
Agents	*Classes that affect the flow and timing of control*
Networks	*Connection management classes*
Managers	*Device housekeeping classes*

Workers

OcaActuator	*Base class for classes that control audio processing*
OcaMute	*Signal mute*
OcaPolarity	*Signal inversion*
OcaSwitch	*1 of n selector*
OcaGain	*Simple gain in dB*
OcaPanBalance	*Pan or balance control*
OcaDelay	*Signal delay in mSec*
OcaDelayExtended	*Signal delay in mSec, ft, m*
OcaFrequencyActuator	*Frequency*
OcaFilterClassical	*Bessel, Butterworth, etc.*
OcaFilterParametric	*Peaking or shelving parametric filter*
OcaFilterPolynomial	*Rational polynomial filter*
OcaFilterFIR	*FIR specified by coefficients*
OcaFilterArbitraryCurve	*Magnitude vs freq curve*
OcaDynamics	*Generalized compressor/expander*
OcaDynamicsDetector	*Side-chain detector*
OcaDynamicsCurve	*Dynamics input vs output level curve*
OcaSignalGenerator	*Multi-waveform signal generator*
OcaSignalInput	*Device signal input port*
OcaSignalOutput	*Device signal output port*
OcaTemperatureActuator	*Temperature parameter*
OcaIdentificationActuator	*Device identification light or other flag*

Actuators, continued	
OcaBasicActuator	*Base class for weakly typed actuators*
OcaBooleanActuator	*Weakly typed actuators ...*
OcaInt8Actuator	...
OcaInt16Actuator	...
OcaInt32Actuator	...
OcaInt64Actuator	...
OcaUint8Actuator	...
OcaUint16Actuator	...
OcaUint32Actuator	...
OcaUint64Actuator	...
OcaFloat32Actuator	...
OcaFloat64Actuator	...
OcaStringActuator	...
OcaBitStringActuator	...

Workers

OcaSensor	*Base class for classes that monitor the device*
OcaLevelSensor	*Signal level*
OcaAudioLevelSensor	*Audio level with standard meter laws*
OcaTimeIntervalSensor	*Time interval*
OcaFrequencySensor	*Frequency*
OcaTemperatureSensor	*Temperature*
OcaIdentificationSensor	*Monitors a button push or something*
OcaBasicSensor	*Base class for weakly typed sensors for general use*
OcaBooleanSensor	...
OcaInt8Sensor	...
OcaInt16Sensor	...
OcaInt32Sensor	...
OcaInt64Sensor	...
OcaUint8Sensor	...
OcaUint16Sensor	...
OcaUint32Sensor	...
OcaUint64Sensor	...
OcaFloat32Sensor	...
OcaFloat64Sensor	...
OcaStringSensor	...
OcaBitStringSensor	...

Workers

Blocks and Matrices	***Classes that allow grouping of device functions***
OcaBlock	*Container for Workers, Agents, and Networks that defines a related set of device functions*
OcaBlockFactory	*Constructor for OcaBlock objects; to be used with dynamically-reconfigurable DSP devices*
OcaMatrix	*Specialized container for 2-dimensional arrays of processing elements; superset of conventional gain matrix.*

Networks	***Connection management classes***
OcaApplicationNetwork	*Abstract base class for other network classes*
OcaControlNetwork	*Application network for transport of control traffic (e.g. an AES70 network)*
OcaMediaTransportNetwork	*Application network for transport of media content (e.g. an AES67 network)*

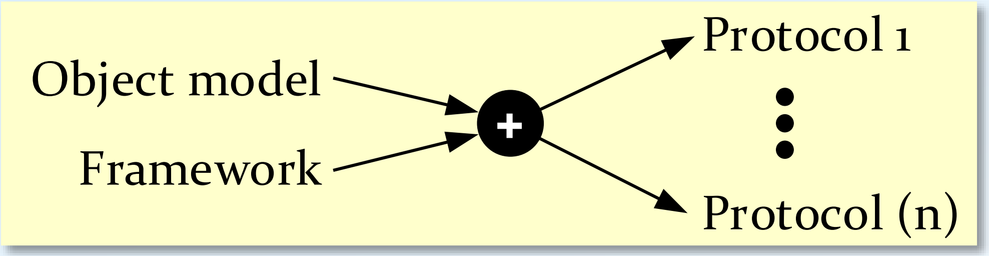
Protocols

Protocols

- OCA is protocol-agile.
- Many protocols, one object model, one framework

Current AES70

- One RPC protocol: OCP.1

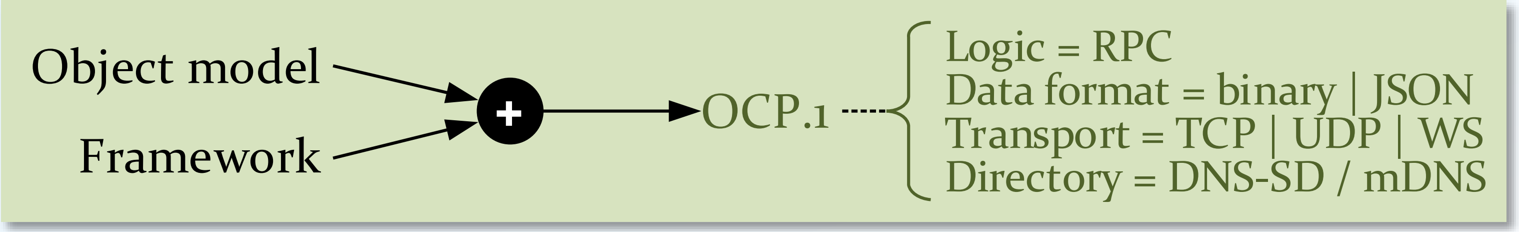


Object model

Framework

Protocol 1

Protocol (n)



Object model

Framework

OCP.1

Logic = RPC
Data format = binary | JSON
Transport = TCP | UDP | WS
Directory = DNS-SD / mDNS

Designing An OCA Device

Designing An OCA Device

Example: Eight-channel mic preamp

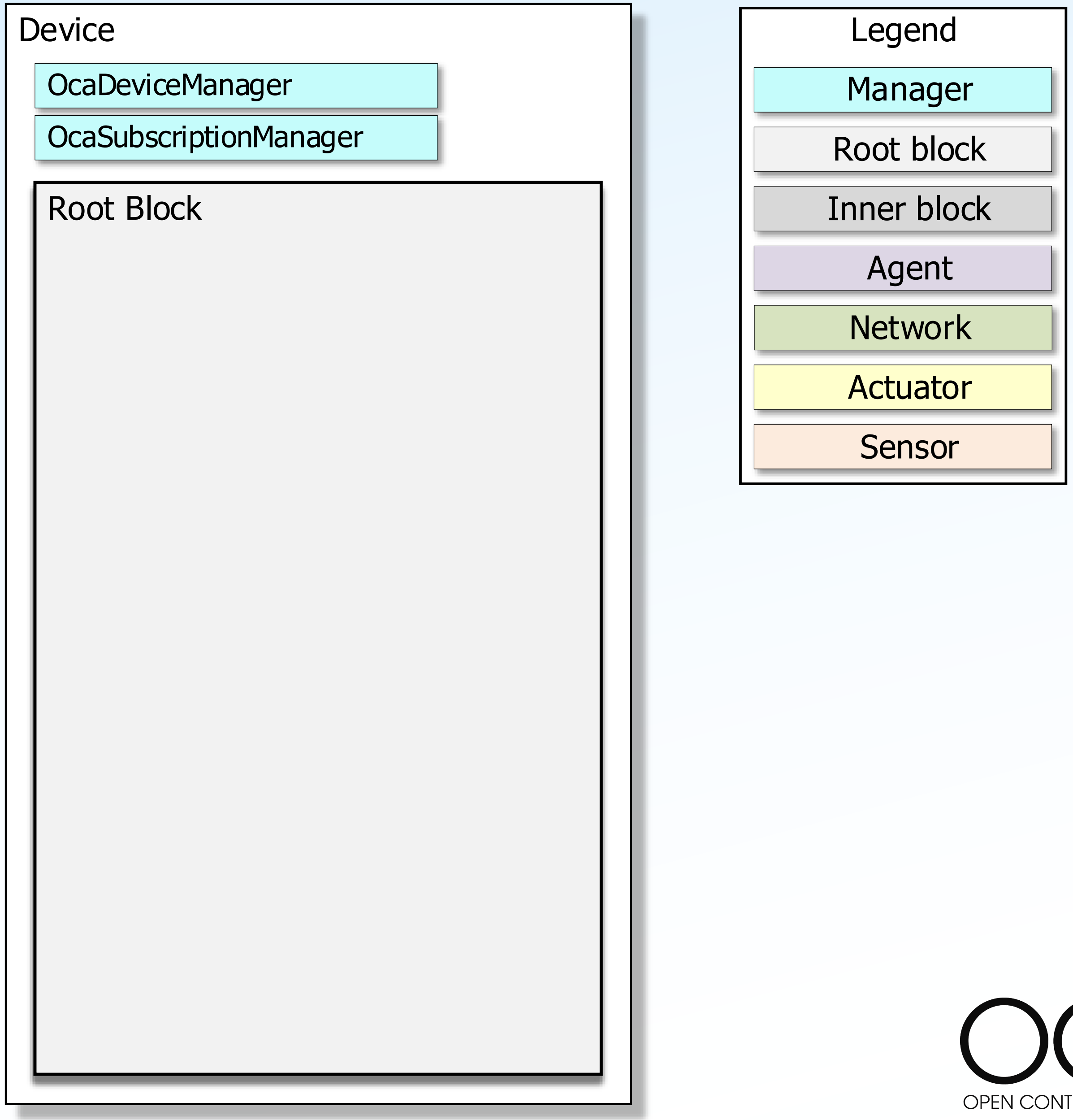
- 8 analogue inputs, switchable line/mic
- Each input with phantom power, high-pass filter, and polarity switch
- AES67 output

Step by step design follows.

Eight-channel mic preamp

Start

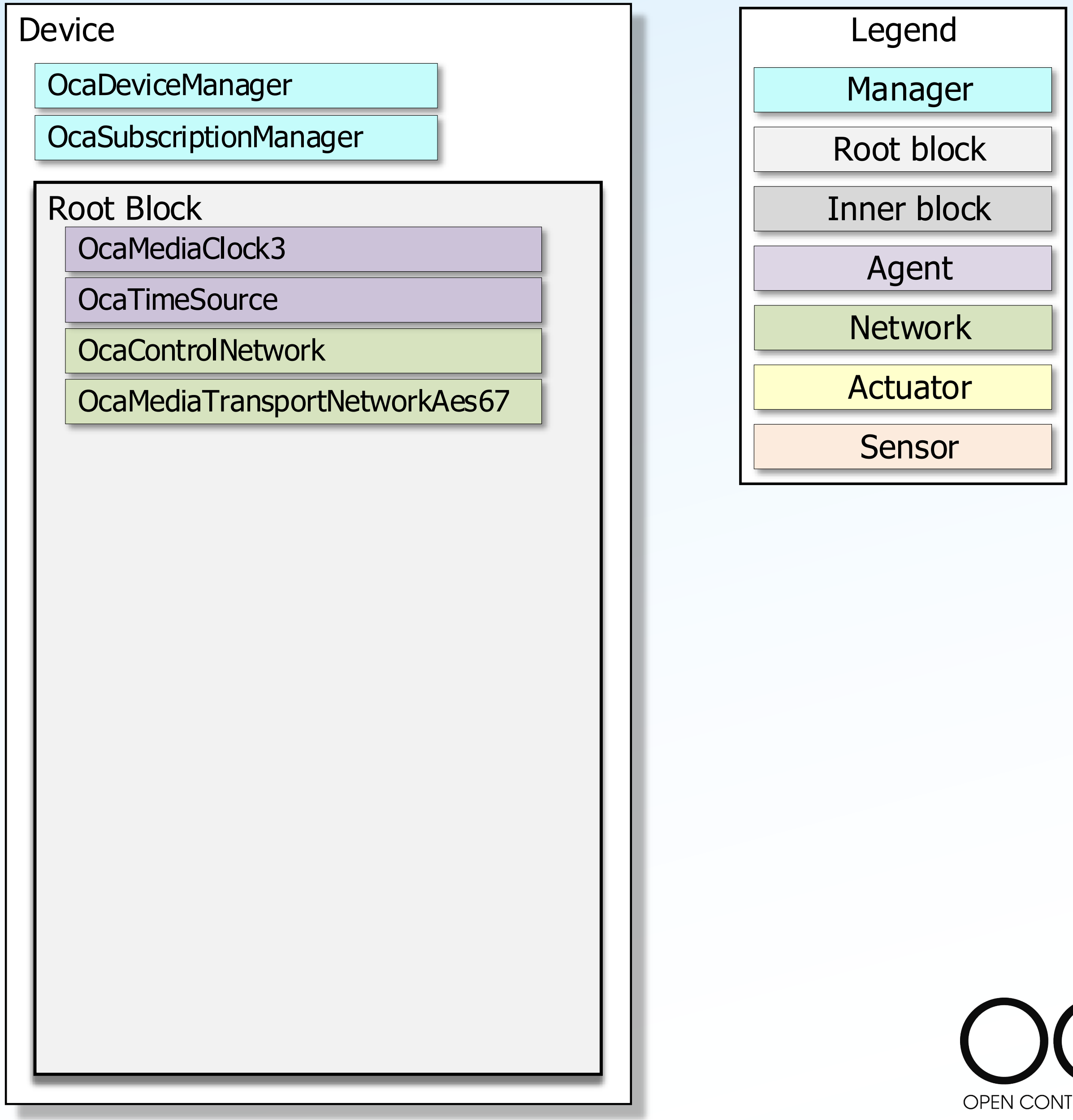
- Basic null device
- Compliant with AES70 minimum device specification



Eight-channel mic preamp

Add clocking and networking

- Clock object
- Control and AES67 media network objects



Device

OcaDeviceManager

OcaSubscriptionManager

Root Block

OcaMediaClock3

OcaTimeSource

OcaControlNetwork

OcaMediaTransportNetworkAes67

Legend

Manager

Root block

Inner block

Agent

Network

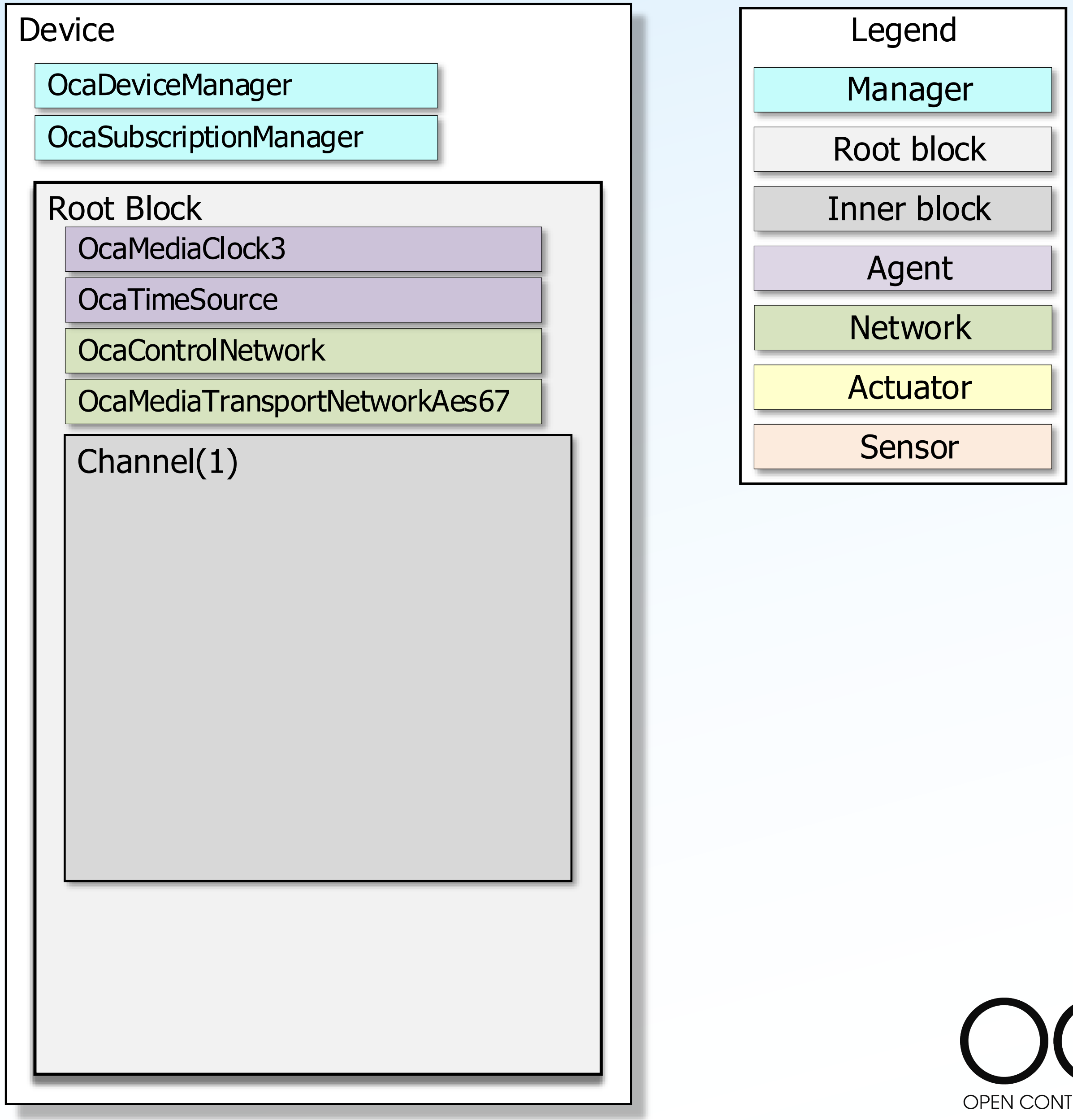
Actuator

Sensor

Eight-channel mic preamp

Add an audio channel block

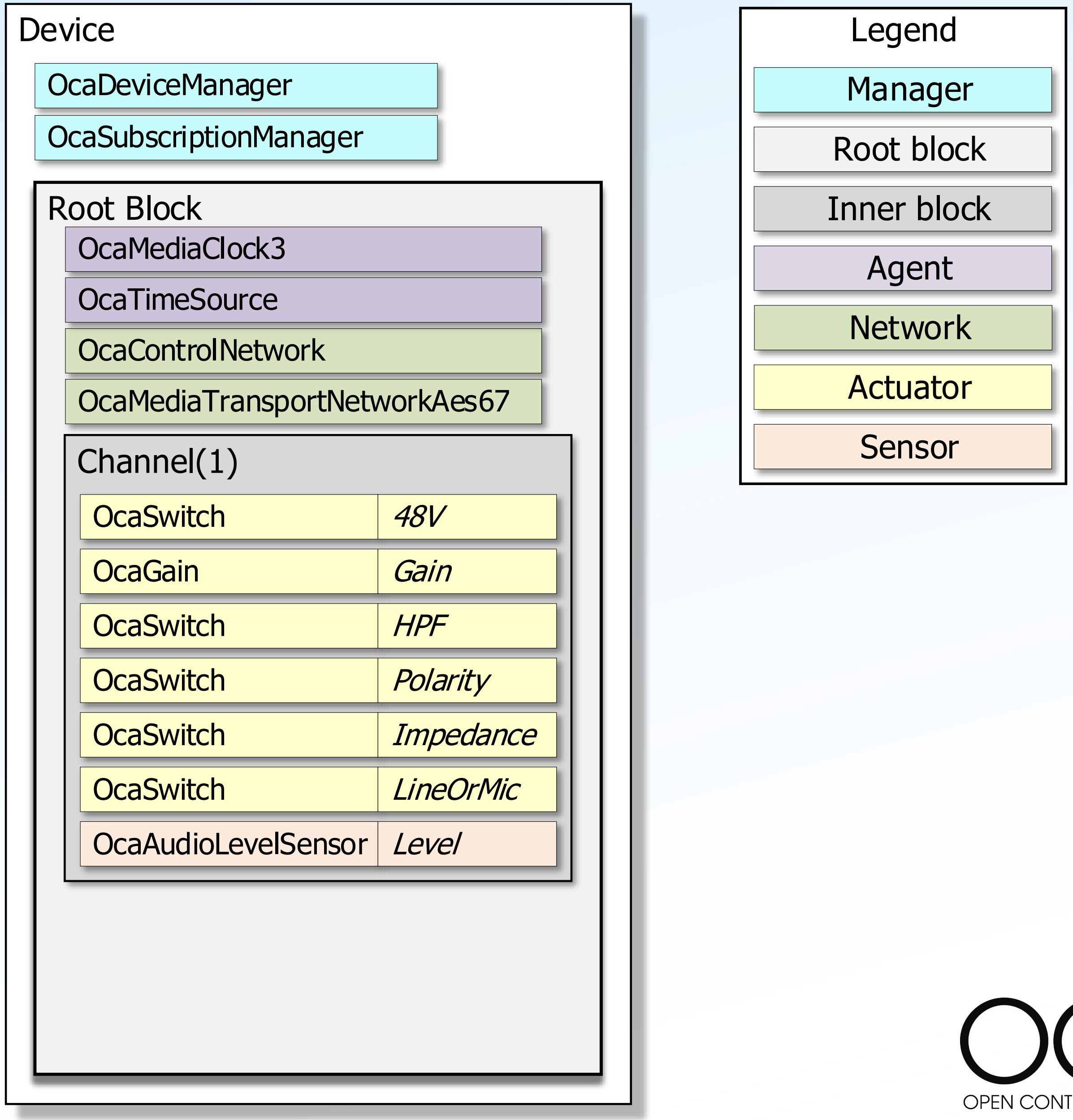
- Inner OcaBlock object



Eight-channel mic preamp

Populate the audio channel block

- Switch objects
- Gain control object
- Level monitor object



Device

- OcaDeviceManager
- OcaSubscriptionManager

Root Block

- OcaMediaClock3
- OcaTimeSource
- OcaControlNetwork
- OcaMediaTransportNetworkAes67

Channel(1)

OcaSwitch	*48V*
OcaGain	*Gain*
OcaSwitch	*HPF*
OcaSwitch	*Polarity*
OcaSwitch	*Impedance*
OcaSwitch	*LineOrMic*
OcaAudioLevelSensor	*Level*

Legend

- Manager
- Root block
- Inner block
- Agent
- Network
- Actuator
- Sensor

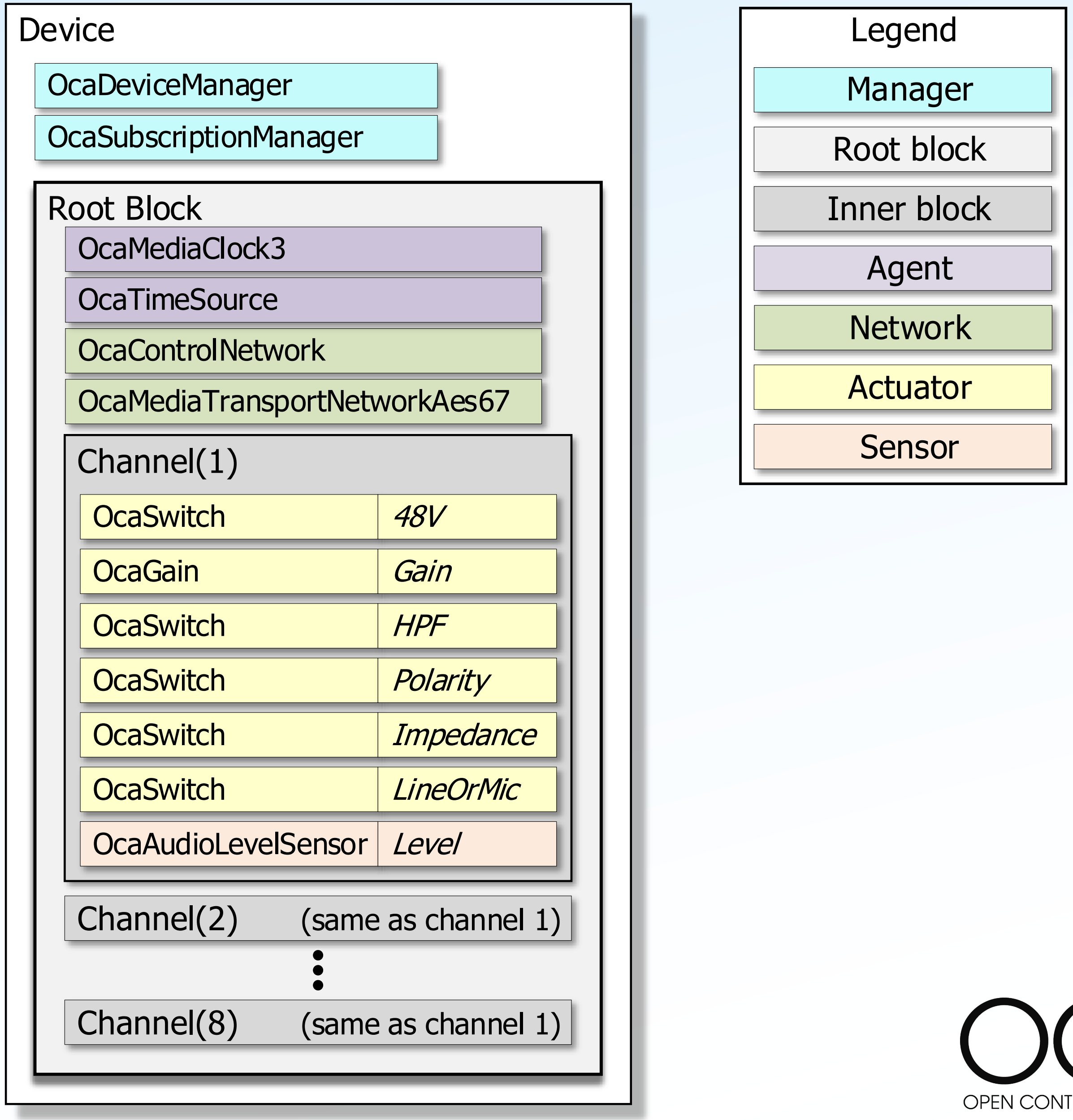
Eight-channel mic preamp

Replicate the audio channel block

- Clone seven more channels.
- Done!

What's the resulting device API?

- Each object publishes its own API.
- The device's complete control API is the union of all its objects' APIs.
- Each class's definition automatically implies a specific API definition - no further specification work is required.



Device

OcaDeviceManager

OcaSubscriptionManager

Root Block

OcaMediaClock3

OcaTimeSource

OcaControlNetwork

OcaMediaTransportNetworkAes67

Channel(1)

OcaSwitch	*48V*
OcaGain	*Gain*
OcaSwitch	*HPF*
OcaSwitch	*Polarity*
OcaSwitch	*Impedance*
OcaSwitch	*LineOrMic*
OcaAudioLevelSensor	*Level*

Channel(2) (same as channel 1)

⋮

Channel(8) (same as channel 1)

Legend

- Manager
- Root block
- Inner block
- Agent
- Network
- Actuator
- Sensor

AES70 Resources

AES70 Resources

Sites

- https://ocaalliance.github.io/ aka the "AES70 Tech Site"

  Free public technical resources for AES70 developers.

- http://ocaalliance.com/

  The usual sort of public website.

Implementations and tools available through the Tech Site

- OCA Microdemo
  A free AES70 implementation, including hardware designs for a small demo PCB. Unencumbered, fully usable for commercial purposes.
- Focusrite Rednet OCA Virtual Device
  Windows executable that simulates an AES70 device.
- oca.js JavaScript library
  Javascript library that supports AES70. For building web-based AES70 device controllers.
- OCA Wireshark plugin

  This plugin allows analyzing AES70 network traffic using Wireshark*, the popular free network protocol analyzer.*

AES70 Resources

Information on the Techsite	**AES143 Presentations** *Slides from two presentations, "How to Make an AES70 Device", and "How to Make an AES70 Controller", originally given at AES 143 in 2017 October. Audio recordings of these presentations is available through the AES website.*
Commercial AES70 Implementation	**Bosch AES70 Reference Implementations** *Fully engineered, commercial-grade AES70 development kit. Not free. Licensable from Bosch Communications or DeusO.*
OCA Alliance Resources	Additional resources are available to OCA Alliance member companies. Basic membership costs $1500 per year.

Thank You