



NMOS

AMWA NMOS OSSRF

Open-Source Sender/Receiver Framework

Pedro Ferreira – BISECT, Founder & CEO

pedro@bisect.pt



[NMOS](#)[SPECS](#)[JOIN](#)[CONTACT](#)[ABOUT](#)[NEWS](#)[...](#)

**Enabling vendor-independent control of all
your IP video and audio gear. IPMX, ST 2110...**

networked media

NMOS

open specifications

**A family of open-source, free-of-charge specifications
that enable interoperability on the control layer
for media devices on an IP infrastructure.**

If you use ST 2110 or IPMX, it's likely you already rely on NMOS.

🎯 NMOS is the open-source "special sauce" that enables connection, management, and control of your IP video and audio devices from different manufacturers – in a common eco-system and in an interoperable way. NMOS works whether you're using ST 2110 or IPMX, locally or in the cloud.

🎯 NMOS (networked media open specifications) lets equipment from different companies all work together. Since AMWA's NMOS is an open set of specifications, it gives you a significant advantage by preventing vendor lock-in. It means you can choose the equipment you prefer, and




OSS to Kickstart your NMOS Implementation

	Media Node Development	Testing
What		
Who		
Why		
Where		
How		

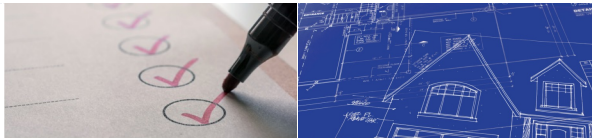
Special thanks to Gareth Sylvester-Bradley

OSS to Kickstart your NMOS Implementation

	Media Node Development	Testing
		
What		NMOS Testing Tool
Who		AMWA
Why		Conformance
Where		<u>AMWA-TV/nmos-testing</u>
How		Apache-2.0


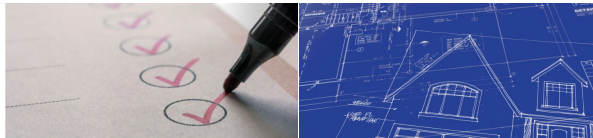
Special thanks to Gareth Sylvester-Bradley

OSS to Kickstart your NMOS Implementation

	Media Node Development	Testing	
			
What		NMOS Testing Tool	Easy NMOS (Registry, Controller, mock Node)
Who		AMWA	Joint
Why		Conformance	Interoperability
Where		<u>AMWA-TV/nmos-testing</u>	<u>rhastie/easy-nmos</u>
How		Apache-2.0	Apache-2.0



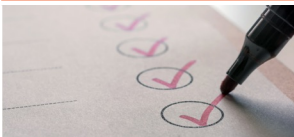

Special thanks to Gareth Sylvester-Bradley

OSS to Kickstart your NMOS Implementation

	Media Node Development	Testing
		
What	nmos-cpp	NMOS Testing Tool Easy NMOS (Registry, Controller, mock Node)
Who	Sony	AMWA Joint
Why	All the low-level building blocks	Conformance Interoperability
Where	sony/nmos-cpp	AMWA-TV/nmos-testing rhastie/easy-nmos
How	Apache-2.0	Apache-2.0 Apache-2.0






Special thanks to Gareth Sylvester-Bradley

OSS to Kickstart your NMOS Implementation

	Media Node Development		Testing	
				
What	nmos-cpp	NvNmos	NMOS Testing Tool	Easy NMOS (Registry, Controller, mock Node)
Who	Sony	NVIDIA	AMWA	Joint
Why	All the low-level building blocks	Common use cases, preassembled	Conformance	Interoperability
Where	sony/nmos-cpp	NVIDIA/nvnmos	AMWA-TV/nmos-testing	rhastie/easy-nmos
How	Apache-2.0	Apache-2.0	Apache-2.0	Apache-2.0

Special thanks to Gareth Sylvester-Bradley

OSS to Kickstart your NMOS Implementation

	Media Node Development			Testing	
					
What	nmos-cpp	NvNmos	Sender/Receiver Framework	NMOS Testing Tool	Easy NMOS (Registry, Controller, mock Node)
Who	Sony	NVIDIA	AMWA	AMWA	Joint
Why	All the low-level building blocks	Common use cases, preassembled	Higher-level building blocks	Conformance	Interoperability
Where	sony/nmos-cpp	NVIDIA/nvnmos	AMWA-TV/nmos-sender-receiver-framework	AMWA-TV/nmos-testing	rhastie/easy-nmos
How	Apache-2.0	Apache-2.0	Apache-2.0	Apache-2.0	Apache-2.0

Special thanks to Gareth Sylvester-Bradley

NMOS Open-Source Sender/Receiver Framework

➤ What

- A library, a set of GStreamer plugins and demo applications to simplify the implementation of NMOS media nodes

➤ Where

<http://github.com/amWA-TV/nmos-sender-receiver-framework/>

➤ When

- Now!



© Copyright VSF 2025
Confidential

<http://github.com/amWA-TV/nmos-sender-receiver-framework/>

The screenshot shows the GitHub repository page for `nmos-sender-receiver-framework` by `AMWA-TV`. The repository is public and was generated from the `AMWA-TV/info-template`. The main branch is `main`, and there are 3 branches and 0 tags. The repository has 56 commits and 4 stars. The file list includes:

File	Commit Message	Time
<code>.github/workflows</code>	osurf: Add existing developed code	3 months ago
<code>.lint</code>	Initial commit	last year
<code>.render</code>	Replace template text with placeholders	last year
<code>cpp</code>	Added README	2 weeks ago
<code>docs</code>	Replace template text with placeholders	last year
<code>images</code>	Basic plugin template done	2 months ago
<code>scripts</code>	Added README	2 weeks ago
<code>.gitignore</code>	osurf: Add existing developed code	3 months ago
<code>CHANGELOG.md</code>	Initial commit	last year
<code>CMakeLists.txt</code>	osurf: Add existing developed code	3 months ago
<code>CONTRIBUTING.md</code>	Initial commit	last year
<code>LICENSE</code>	Initial commit	last year
<code>NOTICE</code>	Initial commit	last year
<code>README.md</code>	Added README	2 weeks ago
<code>conan.lock</code>	Add Conan lockfile	3 months ago
<code>conanfile.py</code>	osurf: Add existing developed code	3 months ago
<code>spec.yml</code>	Replace template text with placeholders	last year

The right sidebar shows the repository's metadata, including the `specs.amwa.tv/nmos-sender-receiver-...` link, the `framework` tag, and the `Apache-2.0 license`. The `Releases` section shows no releases published, and the `Packages` section shows no packages published. The `Contributors` section shows 5 contributors, and the `Languages` section shows a bar chart for C++ (91.8%), CMake (4.7%), Shell (2.2%), C (0.6%), Dockerfile (0.4%), and Python (0.1%).

NMOS Open-Source Sender/Receiver Framework

➤ Phase 1

- Library with a simple API to create senders and receivers
- Simple low-res ST2110-20 and –30 sender and receiver
- Demo applications
- Launched summer 2025

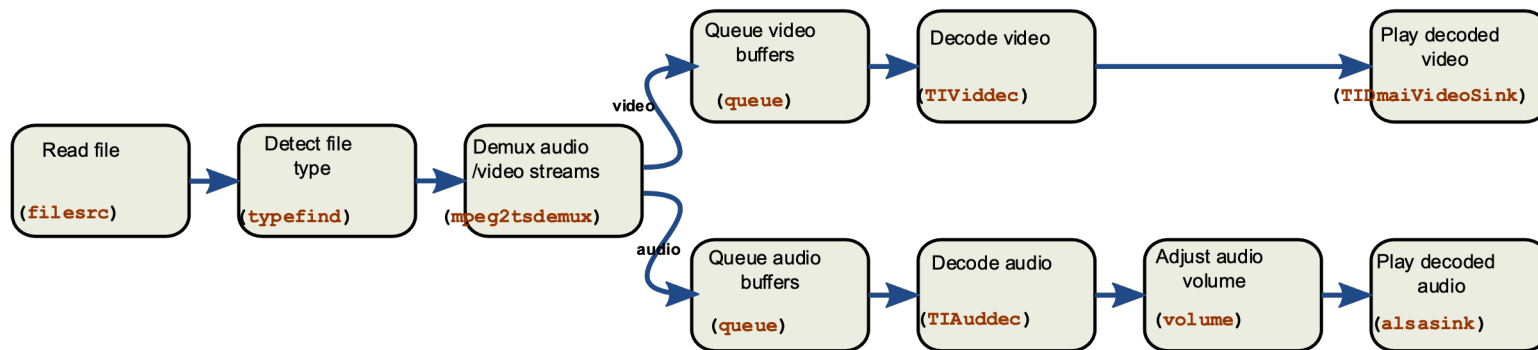
GStreamer

- <https://gstreamer.freedesktop.org/>
- Media processing pipelines
- Can be launched from the command-line
- Can be embedded in an application

Example GStreamer Pipeline

The goals of GStreamer are to separate the application (e.g. Video player, Video editor, etc.) from the streaming media complexity (e.g. hardware acceleration, remoteness)
 GStreamer – streaming media
 D-Bus – inter process communication
 Use **gst-launch** command to create the GStreamer pipeline

```
gst-rtsp-server - v4l2src ! video/x-raw,width=1280,height=720 ! omxh264enc ! video/x-h264,profile=baseline ! h264parse config-interval=1 ! rtph264pay name=pay0 pt=96
```



```
gst-launch filesrc location="video.ts" ! typefind ! mpeg2tsdemux name=demux \
  demux. ! 'video/x-h264' ! queue ! TIViddec ! TIDmaVideoSink \
  demux. ! 'audio/mpeg' ! queue ! TIAuddec ! volume volume=5 ! alsasink
```

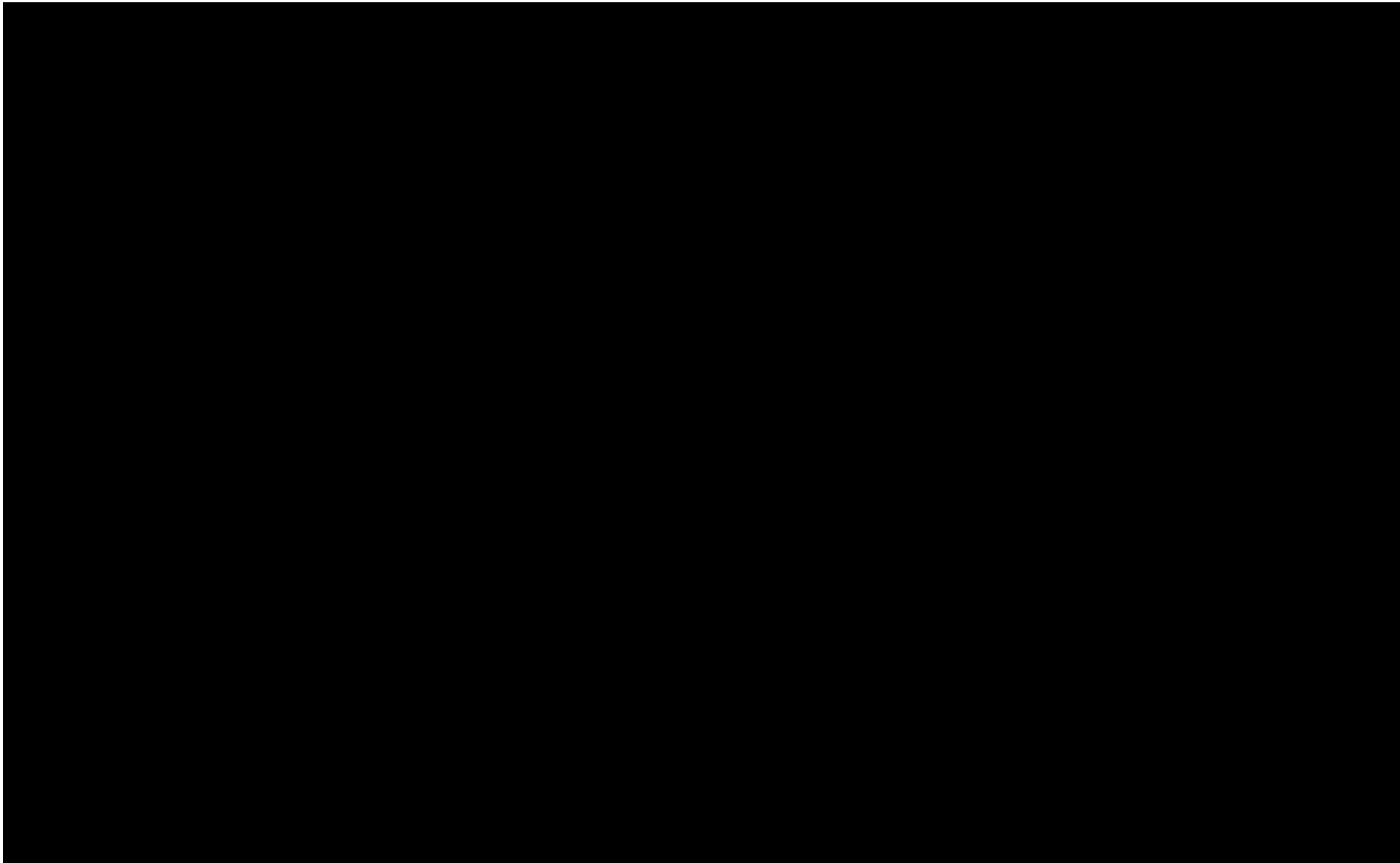
NMOS Open-Source Sender/Receiver Framework

- **Phase 2**

- GStreamer plugins for sending and receiving audio and video
- Just released

Demo#1

- **Use command line to launch 3 GStreamer pipelines:**
 - 2 video senders
 - 1 video receiver
- **Control using Matrox ConductIP**
 - Using NMOS IS-04 and IS-05



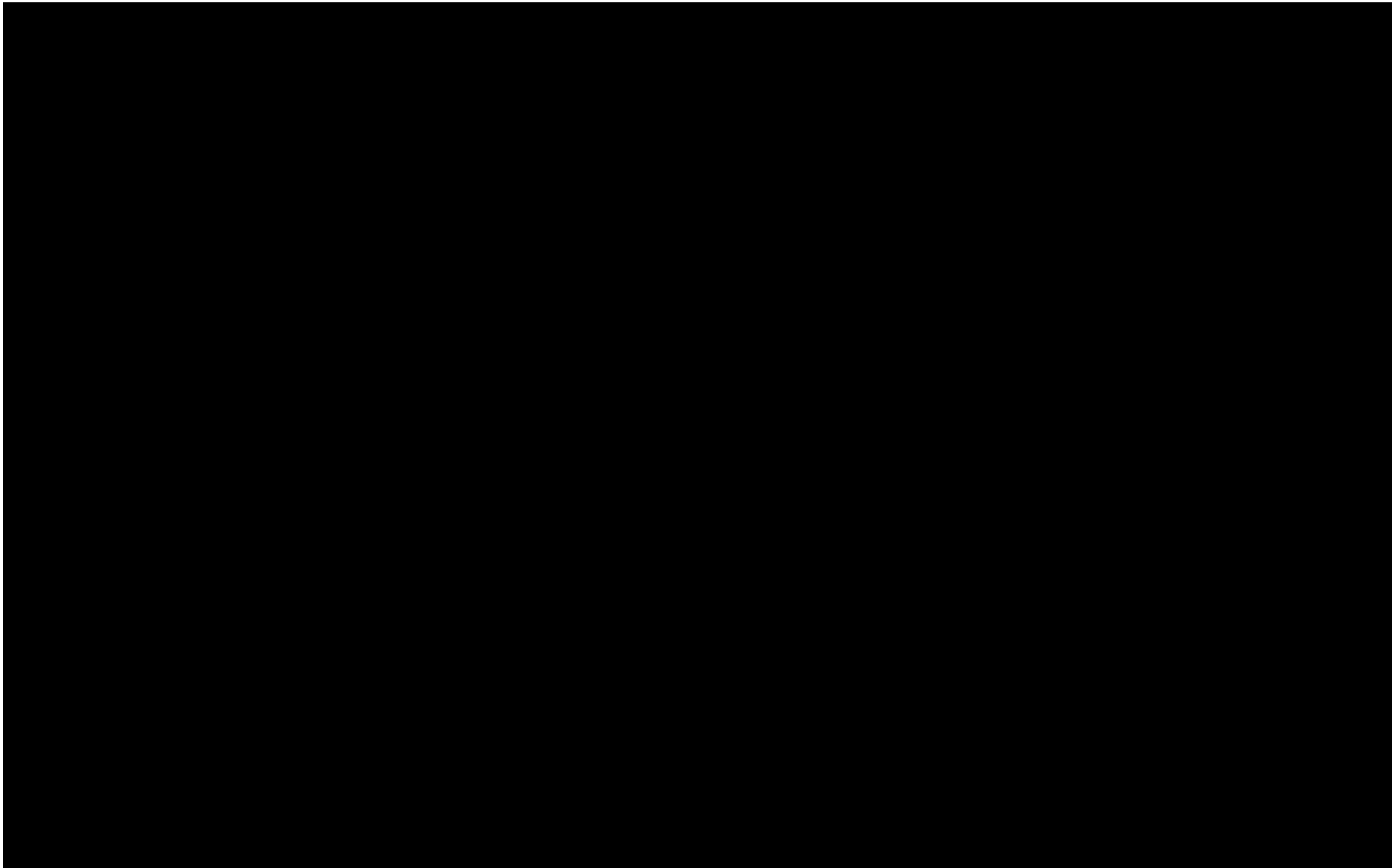
Demo#2

- **Use command line to launch 4 GStreamer pipelines:**

- 2 video senders
- 2 video receivers

- **Control using nmos-js / Easy NMOS**

- Using NMOS IS-04 and IS-05



Next Steps

- **Need feedback from the community**
- **Contribute!**

<http://github.com/amWA-TV/nmos-sender-receiver-framework/>

The screenshot shows the GitHub repository page for `nmos-sender-receiver-framework` by `AMWA-TV`. The repository is public and was generated from an `AMWA-TV/info-template`. The main branch is `main`, and there are 3 branches and 0 tags. The repository has 56 commits and 4 stars. The file list includes:

File	Commit Message	Time
<code>.github/workflows</code>	osurf: Add existing developed code	3 months ago
<code>.lint</code>	Initial commit	last year
<code>.render</code>	Replace template text with placeholders	last year
<code>cpp</code>	Added README	2 weeks ago
<code>docs</code>	Replace template text with placeholders	last year
<code>images</code>	Basic plugin template done	2 months ago
<code>scripts</code>	Added README	2 weeks ago
<code>.gitignore</code>	osurf: Add existing developed code	3 months ago
<code>CHANGELOG.md</code>	Initial commit	last year
<code>CMakeLists.txt</code>	osurf: Add existing developed code	3 months ago
<code>CONTRIBUTING.md</code>	Initial commit	last year
<code>LICENSE</code>	Initial commit	last year
<code>NOTICE</code>	Initial commit	last year
<code>README.md</code>	Added README	2 weeks ago
<code>conan.lock</code>	Add Conan lockfile	3 months ago
<code>conanfile.py</code>	osurf: Add existing developed code	3 months ago
<code>spec.yml</code>	Replace template text with placeholders	last year

The right sidebar shows the repository's metadata, including the `specs.amwa.tv/nmos-sender-receiver...` link, the `framework` tag, and the `Apache-2.0 license`. The `Releases` section shows no releases published, and the `Packages` section shows no packages published. The `Contributors` section shows 5 contributors, and the `Languages` section shows a bar chart for C++ (91.8%), CMake (4.7%), Shell (2.2%), C (0.6%), Dockerfile (0.4%), and Python (0.1%).

Thank you!



NMOS

AMWA NMOS OSSRF

Open-Source Sender/Receiver Framework