Use of OpenTimelineIO for arbitrary time-series data

Use Case

In order to quickly interpret and gain insights into the results and find the specific scene(s), object(s), time(s), ... I am interested in,

As a creative,

I want to run task-oriented analysis tools on my OpenUSD project and see the results represented on a timeline as time-series data

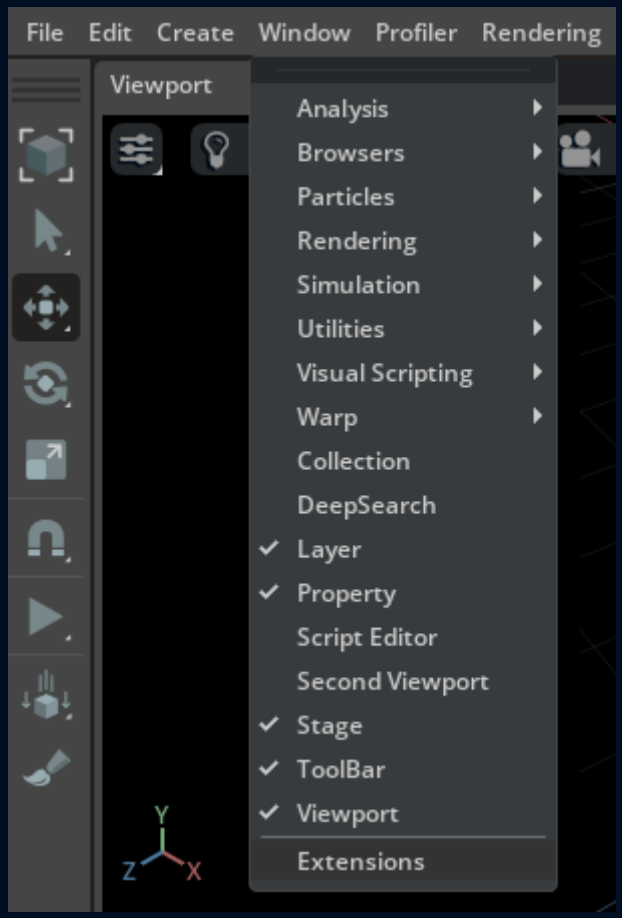
Use of OpenTimelineIO for arbitrary time-series data

Workflow

1. **Window** → **Analyse** → *{pick analysis tool}*

2. **Analysis tool produces output in OpenTimelineIO format**
   - **track labelled with tool-specific schema**

3. **Sequencer able to represent output on the timeline without needing to be schema aware**
   - **if the analysis tool is present, additional context menu options may be present**



Select the ML analysis tool(s) you want to run on the project.

This is a plugin architecture. Analyses the project, produces time-series data in OpenTimelineIO format

NLE or Omniverse App

Video: Object detection

Camera angle detection

Scene change detection

Audio: Speaker detection

Audio: Sentiment analysis

Montreal Forced Alignment

...

OpenTimelineIO

Schema is set to namespace of the tool.

e.g. x.signly.analysis.videoObjects

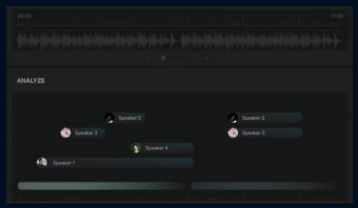
Insert into OpenUSD project as sequencer data (OpenUSD OTIO Schema).

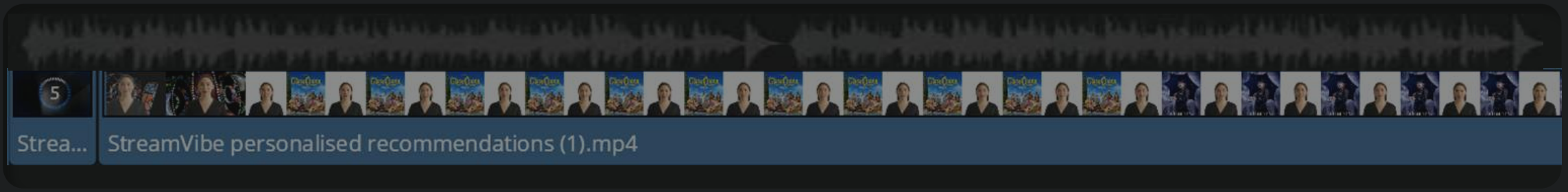
Sequencer component automatically visualises the time-series data.

Sequencer can render the time-series data without requiring the plugin to be present and without needing to know about the semantic meaning of what it is displaying.

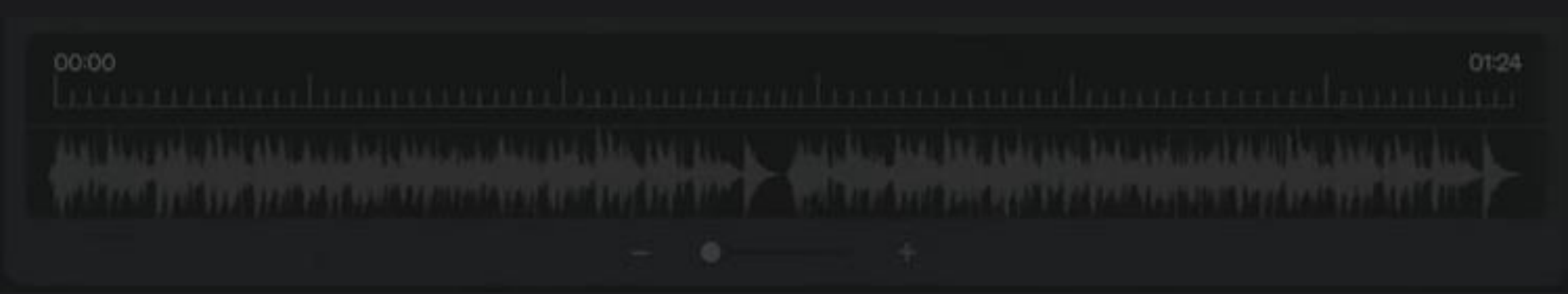
In Omniverse terminology, a plugin is an Omniverse extension.

Additional functionality (such as contextual menus) are enabled if the appropriate extension is present. And sequencer / extension manager can find and install the indicated extensions.

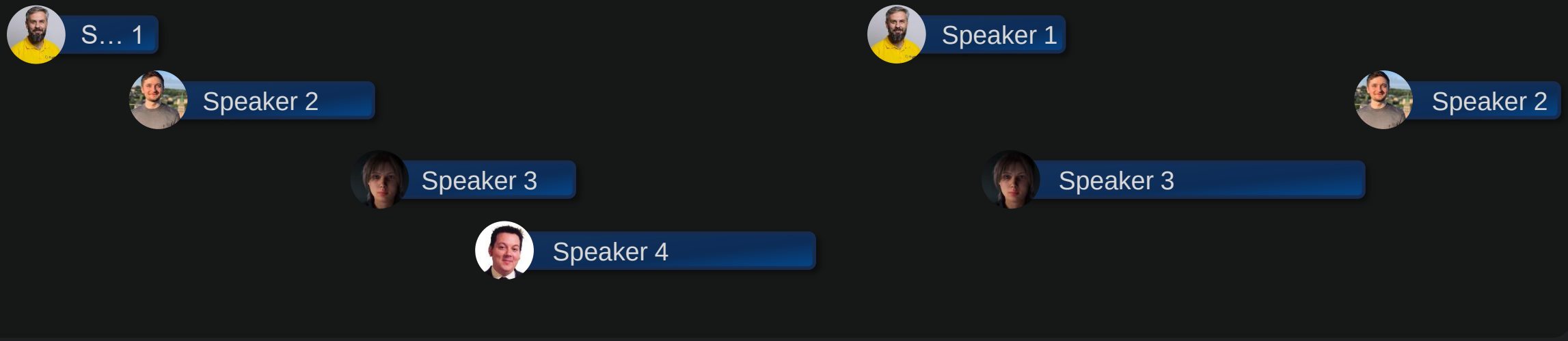




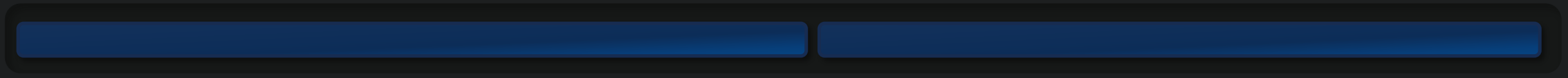
v Object Analysis



v Speaker Analysis



v Scene Analysis



Use of OpenTimelineIO for arbitrary time-series data

Command line

1. Embed the OpenTimelineIO metadata from otio.json into a OpenUSD file:

```
usdotio add otio.json usdfile.usd
```

2. Save the OpenTimelineIO metadata written by #1 to a JSON file:

```
usdotio save -o otio.json usdfile.usd
```

3. Find the Omniverse sequencer information, remove, convert to OpenTimelineIO format and add back in again as OTIO metadata

```
usdotio update -v2 usdfile.usd
```