

Open APIs
for Open
Minds

Kurento Real Time Media Stream Processing - Advanced

Juan Ángel Fuentes

Software Developer. Stream Oriented GE

jafuentes@naevatec.com

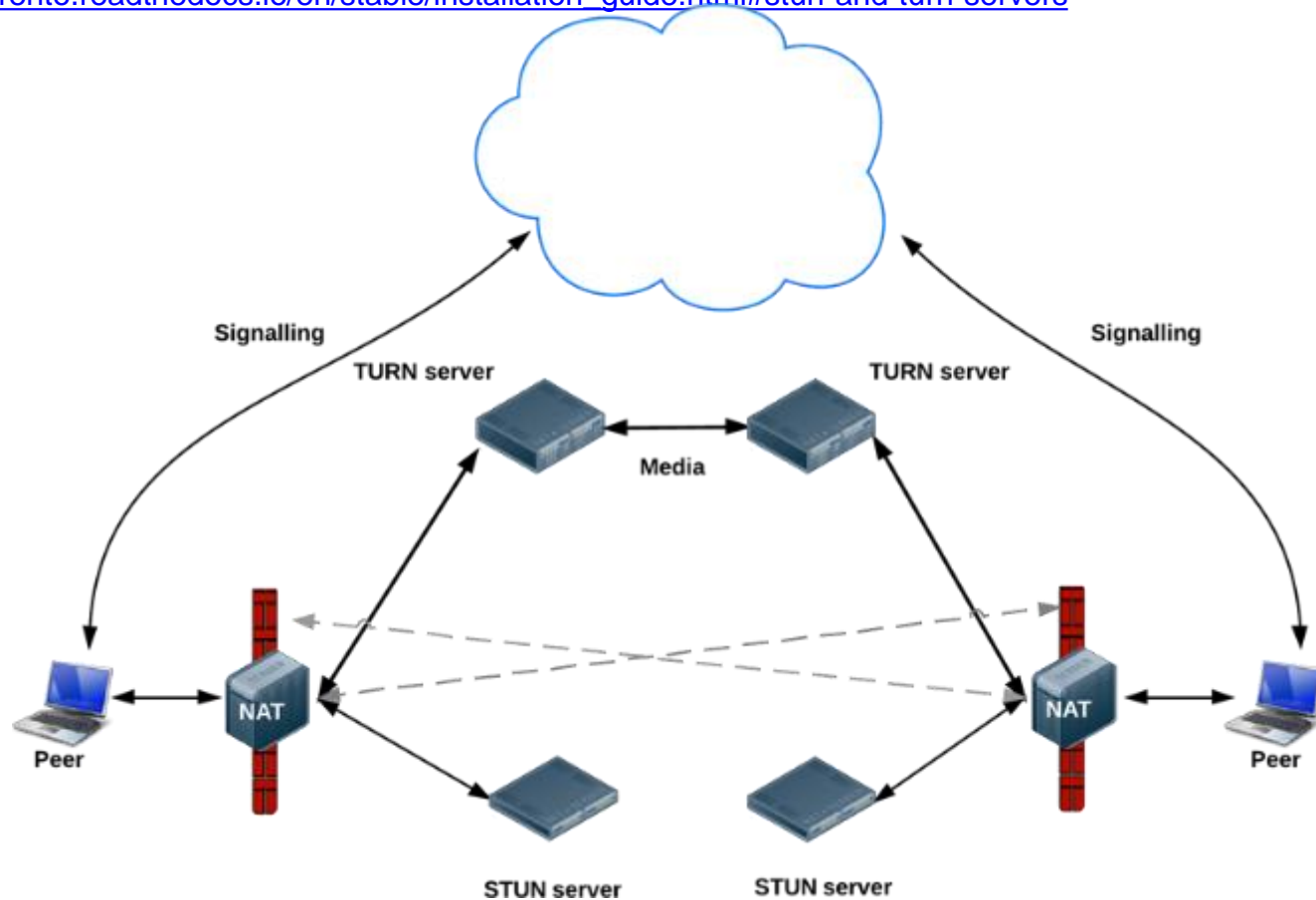
What do we need?

- An instance of the Kurento Media Server
 - **FIWARE-LAB** (<https://catalogue.fiware.org/enablers/stream-oriented-kurento/creating-instances>)
 - **Local installation** (http://doc-kurento.readthedocs.io/en/stable/installation_guide.html)
- Maven
- NPM
- Bower
- Your favorite IDE

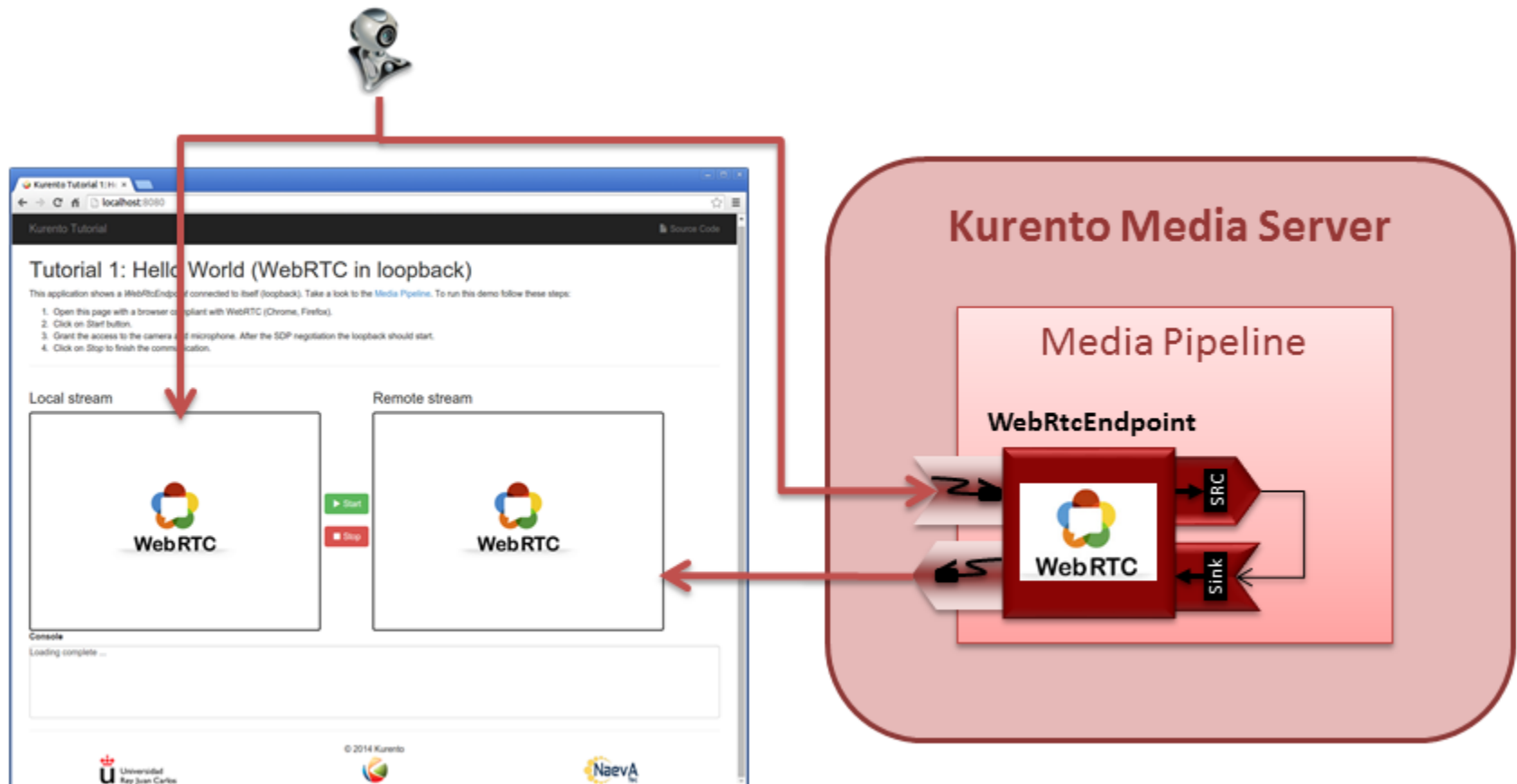
Installation Tip

- Setup a turn and a stun server in configuration to deal with NATs

http://doc-kurento.readthedocs.io/en/stable/installation_guide.html#stun-and-turn-servers

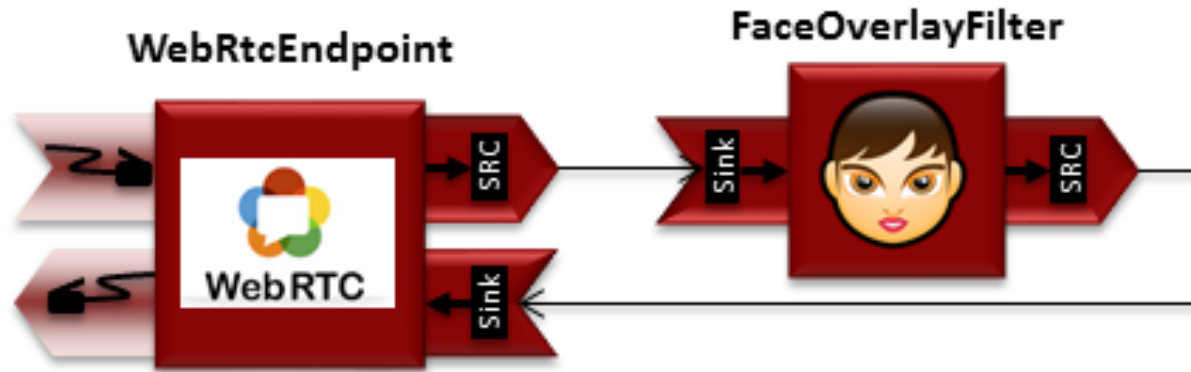


Tutorial 1 - Hello world



<http://doc-kurento.readthedocs.io/en/stable/tutorials.html#hello-world>

Tutorial 2 - WebRTC magic mirror



<http://doc-kurento.readthedocs.io/en/stable/tutorials.html#webrtc-magic-mirror>

Tutorial 3 - WebRTC one-to-one video call with recording and filtering

```
public CallMediaPipeline(KurentoClient kurento, String from, String to) {
    // Media pipeline
    MediaPipeline pipeline = kurento.createMediaPipeline();

    // Media Elements (WebRtcEndpoint, RecorderEndpoint, FaceOverlayFilter)
    webRtcCaller = new WebRtcEndpoint.Builder(pipeline).build();
    webRtcCallee = new WebRtcEndpoint.Builder(pipeline).build();

    recorderCaller = new RecorderEndpoint.Builder(pipeline, RECORDING_PATH
        + from + RECORDING_EXT).build();
    recorderCallee = new RecorderEndpoint.Builder(pipeline, RECORDING_PATH
        + to + RECORDING_EXT).build();

    FaceOverlayFilter faceOverlayFilterCaller = new FaceOverlayFilter.Builder(
        pipeline).build();
    faceOverlayFilterCaller.setOverlaidImage(
        "http://files.kurento.org/imgs/mario-wings.png", -0.35F, -1.2F,
        1.6F, 1.6F);

    FaceOverlayFilter faceOverlayFilterCallee = new FaceOverlayFilter.Builder(
        pipeline).build();
    faceOverlayFilterCallee.setOverlaidImage(
        "http://files.kurento.org/imgs/Hat.png", -0.2F, -1.35F, 1.5F,
        1.5F);

    // Connections
    webRtcCaller.connect(faceOverlayFilterCaller);
    faceOverlayFilterCaller.connect(webRtcCallee);
    faceOverlayFilterCaller.connect(recorderCaller);

    webRtcCallee.connect(faceOverlayFilterCallee);
    faceOverlayFilterCallee.connect(webRtcCaller);
    faceOverlayFilterCallee.connect(recorderCallee);
}
```


Expanding Kurento



Media Server structure

■ Kurento is a pluggable framework.

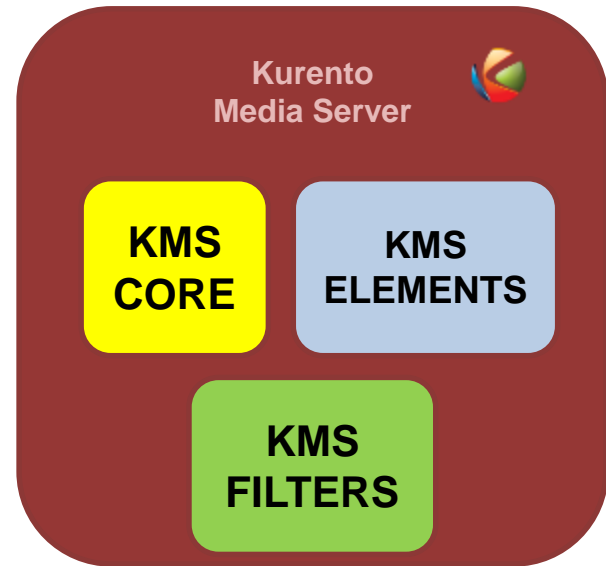
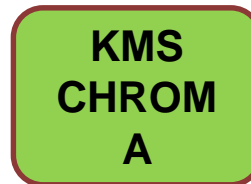
■ Kurento-media-server uses:

- Kms-core
- Kms-elements
- Kms-filters



■ Other available modules:

- Kms-crowddetector
- Kms-pointerdetector
- Kms-chroma
- Kms-platedetector



How to develop a module?

- **Two flavours:**
 - **Gstreamer developer**
 - **OpenCV developer**
- **Define the module using IDL**
- **The tool kurento-module-creator will do the magic:**
 - **Generate code for server-side**
 - **Generate code for clients**

How to develop a module?

- 1. Create module structure using kurento module scaffold**
- 2. Modify kmd file to add module properties**
- 3. Generate code**
- 4. Add filter functionality**
- 5. Compile**
- 6. Install**
- 7. Generate client-side code**

How to develop a module?

- Install Kurento Dev

```
sudo apt-get install kurento-media-server-6.0-dev
```

- OpenCV

```
kurento-module-scaffold.sh <module_name> <output_directory> opencv_filter
```

- GStreamer

```
kurento-module-scaffold.sh <module_name> <output_directory>
```

http://doc-kurento.readthedocs.io/en/stable/mastering/develop_kurento_modules.html

Example 1: For OpenCV developers

<https://github.com/Kurento/kms-opencv-plugin-sample>

Example 2: For GStreamer developers

<https://github.com/Kurento/kms-plugin-sample>

Kubernetes and the FIWARE-LAB

- FIWARE-LAB
 - Working instance of FI-WARE enabling free experimentation with technology
 - <http://lab.fiware.org>
- Creating a Kubernetes instance from an image
 - Use latest version of Kubernetes images.
- Creating a Kubernetes instance using recipes
 - Use Ubuntu 14.04 LTS clear image
 - Use latest version of Kubernetes recipes

Kubernetes and the FIWARE-LAB

Images

| Name ▾ | Status ▾ |
|---------------------------|----------|
| BoINC | active |
| CentOS-6.2-chef | active |
| CentOS-6.3-sdc | active |
| CentOS-6.3-x86_64 | active |
| LPCI-internal | active |
| Ubuntu12.04-server-x86_64 | active |
| cdva-image-r2.3 | active |
| cep-image-r2.3 | active |
| datahandling-ppl | active |
| dbanonymizer-dba | active |
| kubernetes-image-4.0.0 | active |
| kubernetes-image-r3.3 | active |

Launch Instances

1. Details 2. Access & Security 3. Post-Creation 4. Summary

Instance Name *

Flavor

Instance Count *

Description
Specify the details for launching an instance. The chart below shows the resources used by this project in relation to the project's quotas.

Flavor Details

| | |
|----------------|----------|
| Name | m1.small |
| VCPUs | 1 |
| Root Disk | 10 GB |
| Ephemeral Disk | 20 GB |
| Total Disk | 30 GB |
| RAM | 2048 MB |

Project Quotas

| | |
|---------------------------|--------------------|
| Instance Count (1) | 2 Available |
| VCPUs (1) | 5 Available |
| Disk (10 GB) | 990 GB Available |
| Memory (2048 MB) | 22952 MB Available |

* Mandatory fields.

Cancel **Next**

To learn more...

| Thank you!

<http://fiware.org>

Follow @FIWARE on Twitter

