



V-PRISM: An Edge-based Architecture to Virtualize Multimedia Sensors in the Internet of Things

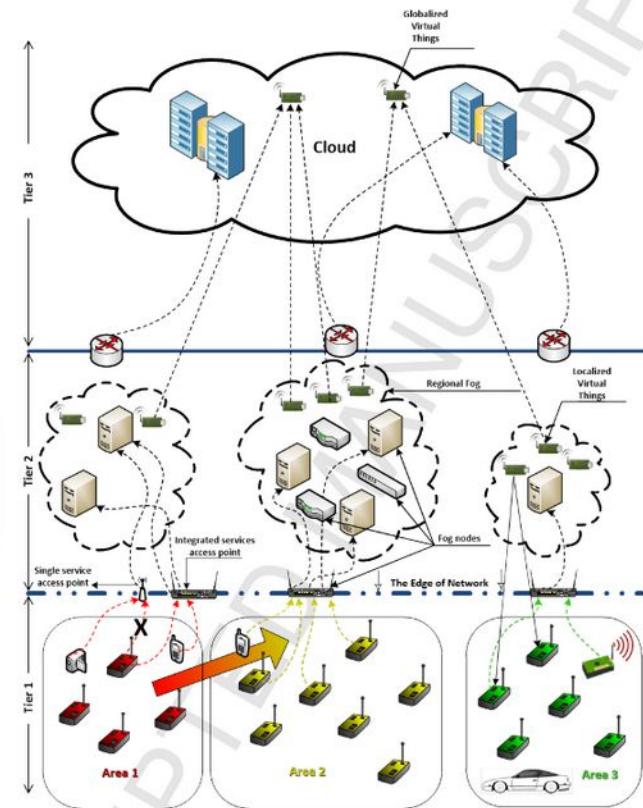
Anselmo Luiz Éden Battisti
Débora Christina Muchaluat Saade
Flávia Coimbra Delicato

Agenda

- Context & Motivation
- Goal
- V-PRISM Architecture
- Proof-of-concept & experiment
- Final Remarks

Context & Motivation

- **Internet of Things (IoT)**: things become continuous source of data;
- **Cloud of Things (CoT)**: the cloud acts as an intermediate layer between sensors and applications;
- **Virtualization**: can be used to abstract the physical world and solves problems like sensor's heterogeneity.



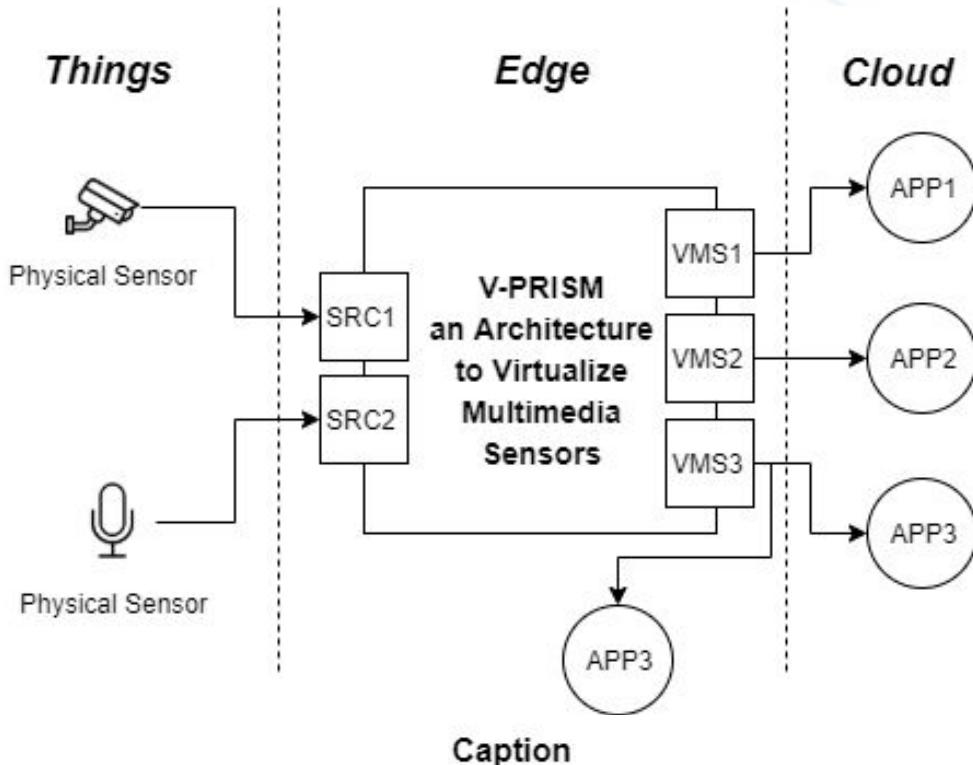
LI, W. et al. System modelling and performance evaluation of a three-tier Cloud of Things. **Future Generation Computer Systems**, v. 70, p. 104-125, 2017.

Context & Motivation

- **Internet of Multimedia Things (IoMT)**: a subset of IoT that includes only multimedia devices (camera and mic);
- **Multimedia** applications are usually **latency-sensitive**;
- Edge computing are in a prime position to process multimedia applications;
- **Edge Computing**: brings storage and computation as near as possible of the source of data.

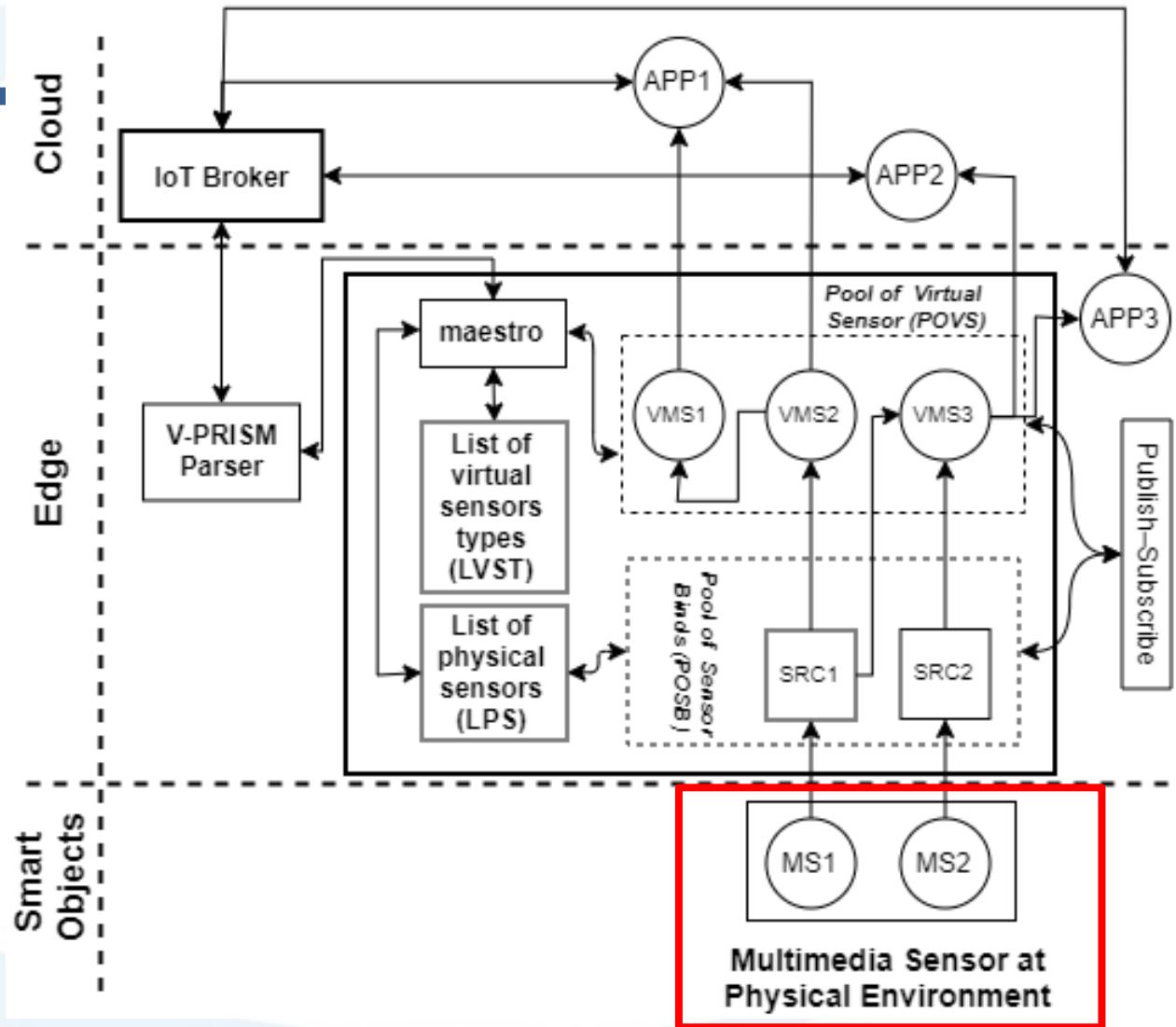
Goal

- Propose an architecture for virtualization of multimedia sensors at the network edge, called V-PRISM (Virtual Programmable IoT Multimedia Sensor);
- Components are deployed at the Edge tier and responsible for processing the multimedia streams
 - produced by physical sensors;
 - virtualized as Virtual Multimedia Sensors (VMS).



SRC = Source
VMS = Virtual Multimedia Sensor
APP = Application

V-PRISM



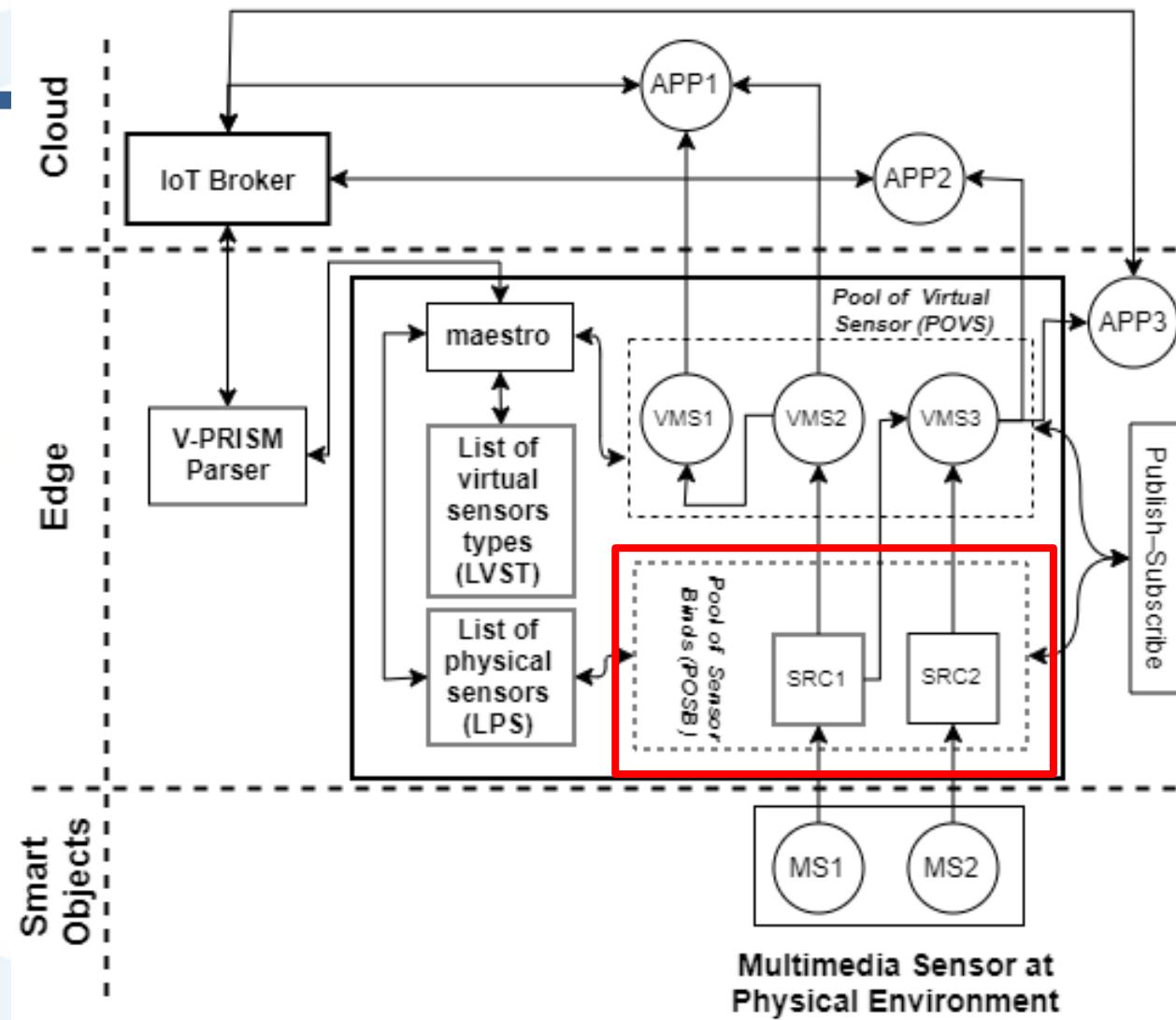
Caption

SRC = Source

VMS = Virtual Multimedia Sensor

APP = Application

V-PRISM



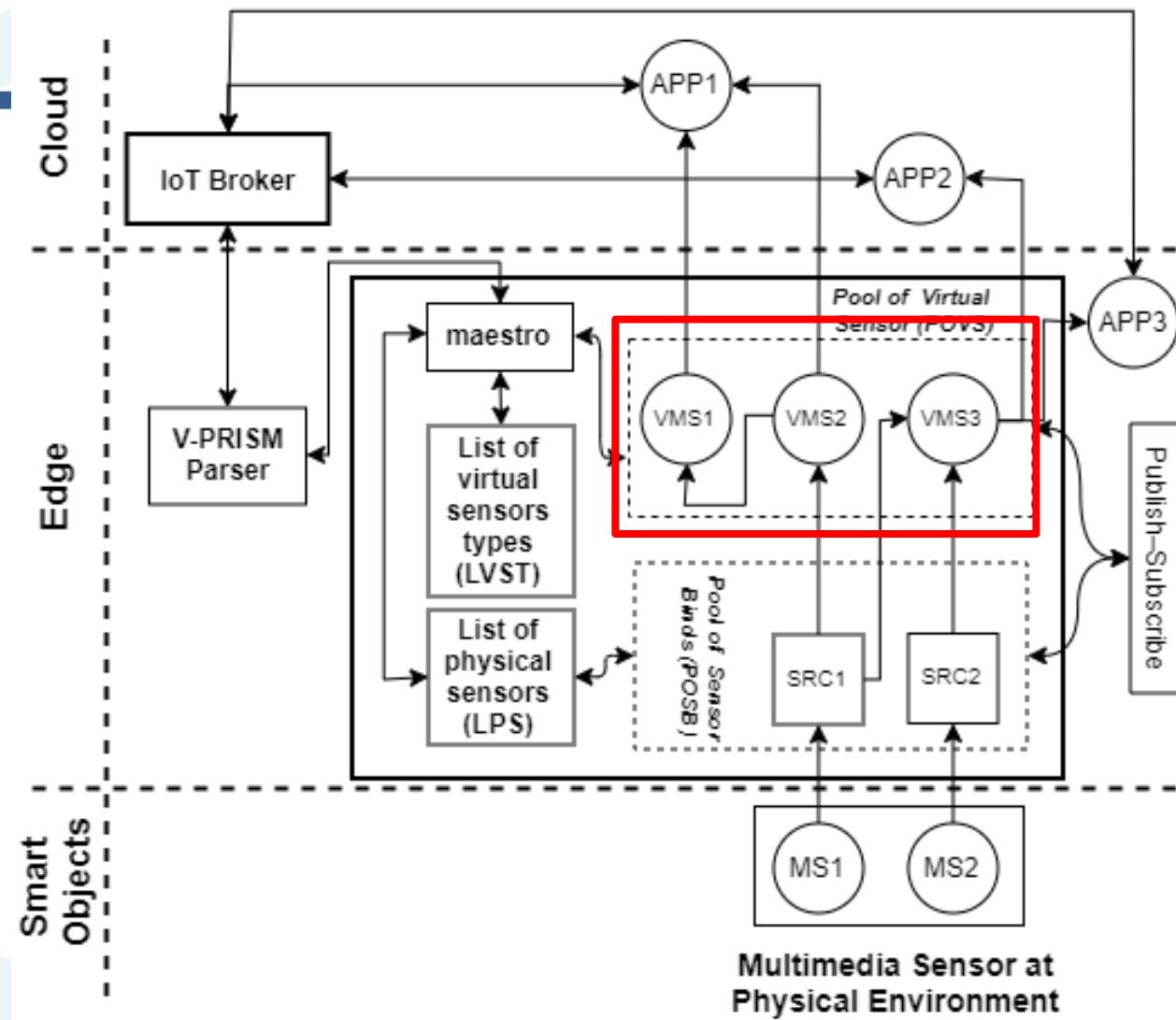
Caption

SRC = Source

VMS = Virtual Multimedia Sensor

APP = Application

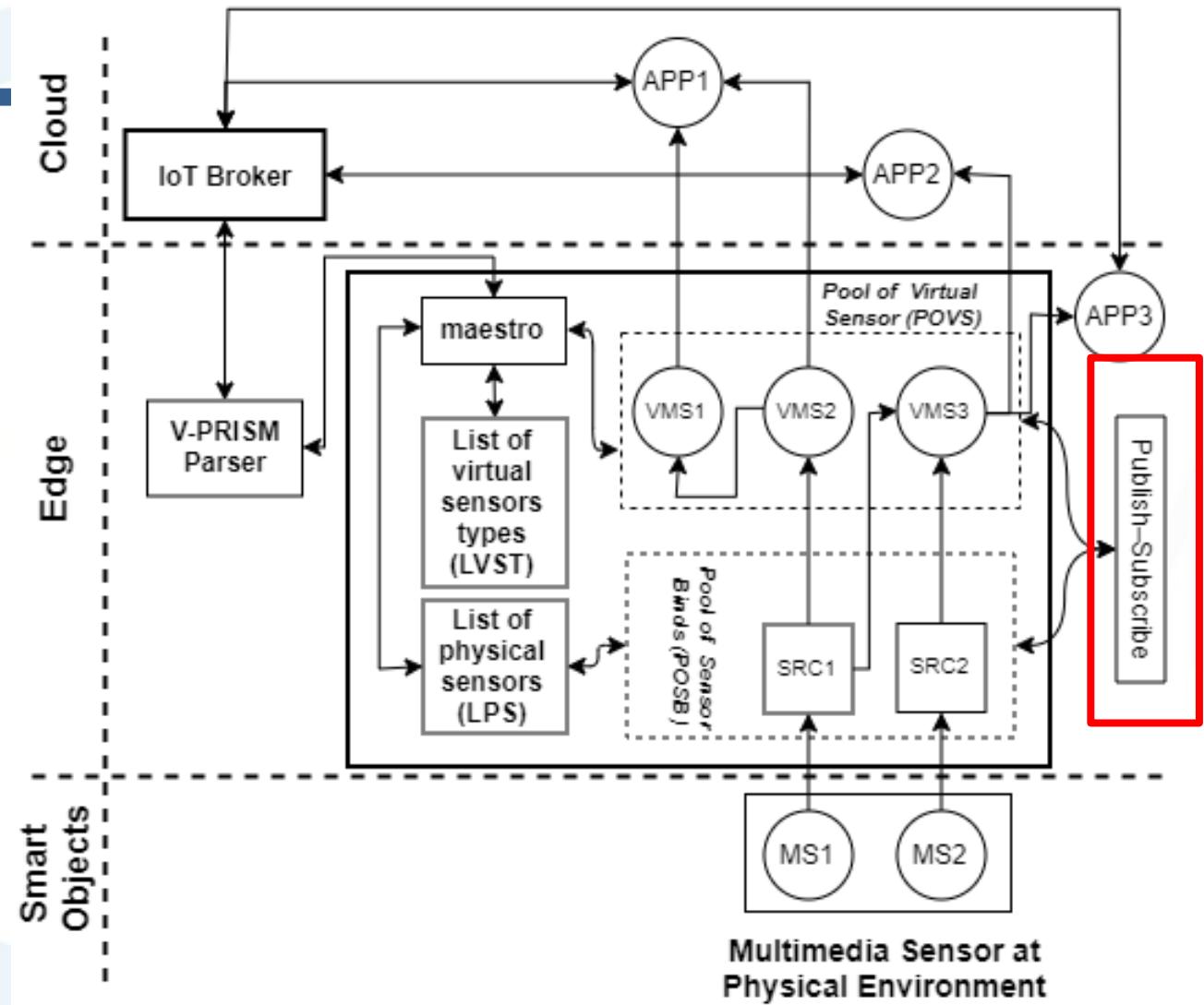
V-PRISM



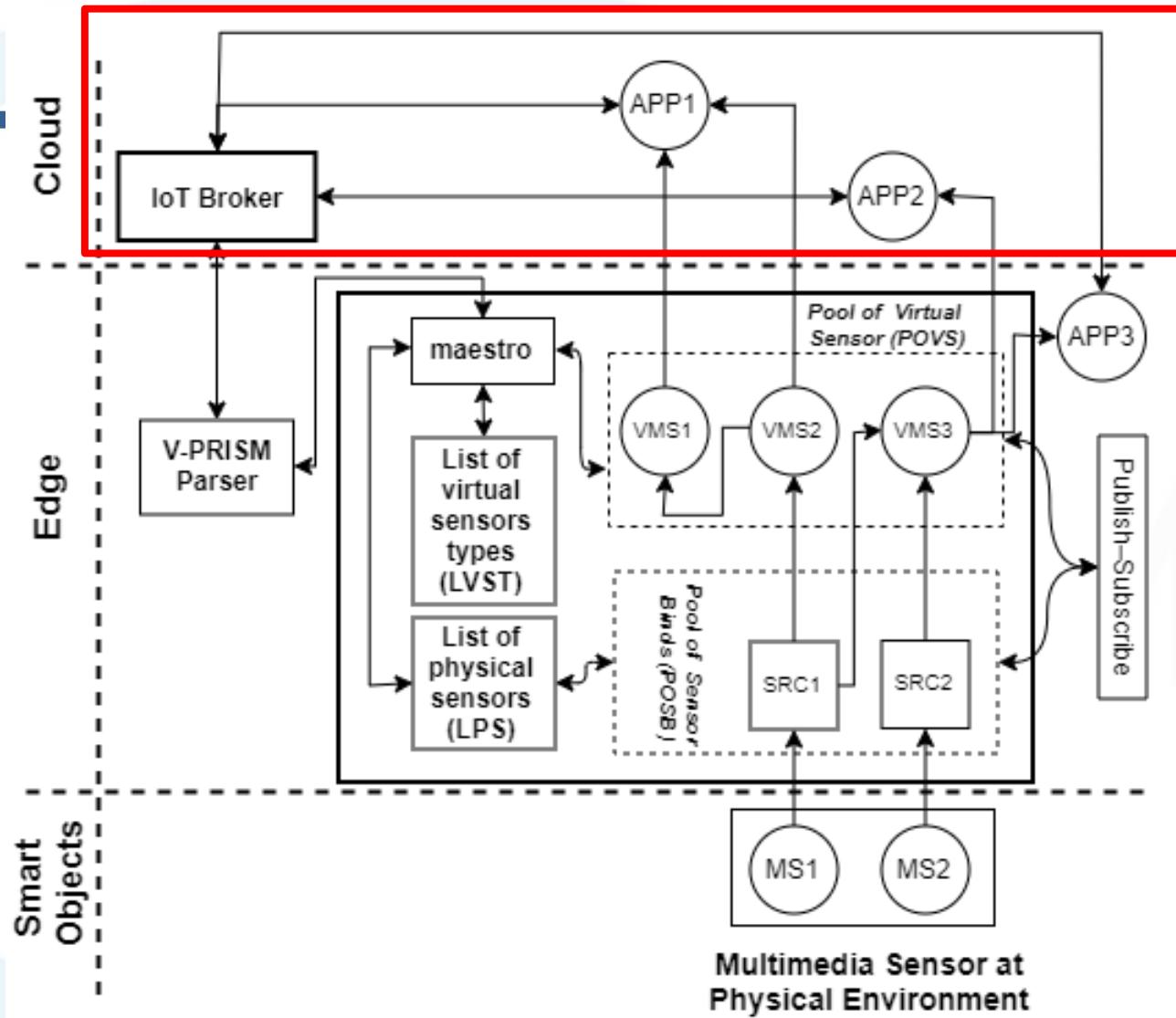
Caption

SRC = Source
VMS = Virtual Multimedia Sensor
APP = Application

V-PRISM



V-PRISM



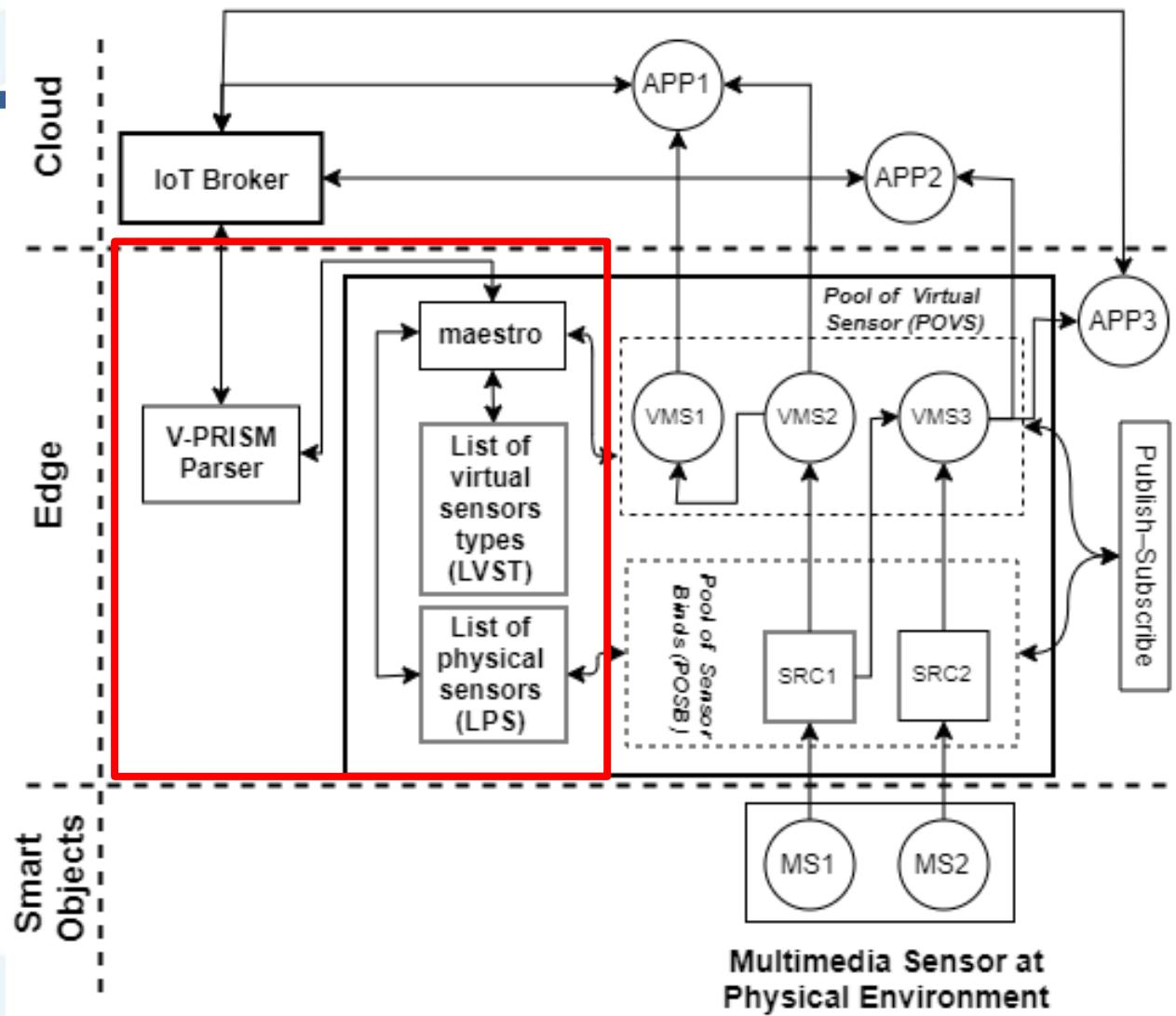
Caption

SRC = Source

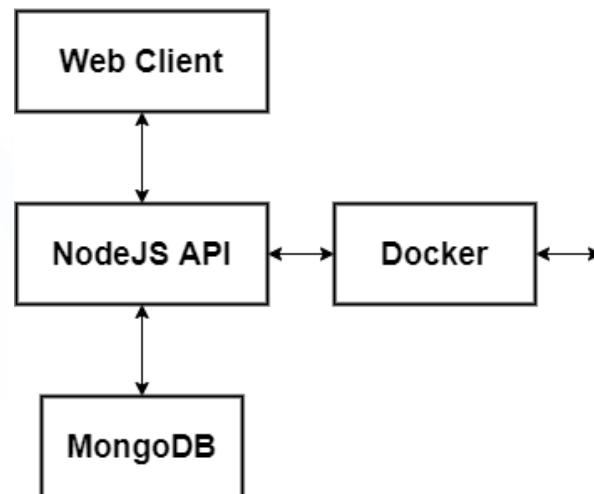
VMS = Virtual Multimedia Sensor

APP = Application

V-PRISM



Proof-of-concept



Virtual Node Types

VMS - Containers

-  Color Camera
-  Noise Detector
-  People Counter
-  Gunshot Detector

SRC - Containers

-  USB Camera
-  RTSP Camera
-  USB Mic
-  File Source

Proof-of-concept

The screenshot shows a web-based management interface for V-PRISM. The title bar indicates the application is "web-app" running at "localhost:8080/#/vmsType". The main content area is titled "V-PRISM Manager" and displays a list of "VMS Types Plugins".

The left sidebar contains navigation links:

- Home
- VMS
 - Running
 - VMS
- Locations
- Devices
- VMS Types Plugins
- VMS Types SRC
- Github

The main content area has a header "⌘ VMS Types Plugins" with a "+ New" button. It includes a table with columns: Name, Docker Image, and Actions.

Name	Docker Image	Actions
VMS = Forward UDP to UDP changing video to greyscale	alfa/plugin/udp_video_black_white	<button>Start New VMS</button> <button>Edit</button> <button>Remove</button>
VMS = Forward UDP to UDP cropping the video	alfa/plugin/udp_video_crop	<button>Start New VMS</button> <button>Edit</button> <button>Remove</button>
VMS = UDP to UDP	alfa/plugin/udp_to_udp	<button>Start New VMS</button> <button>Edit</button> <button>Remove</button>
VMS = Noise Detector	alfa/plugin/noise_detector	<button>Start New VMS</button> <button>Edit</button> <button>Remove</button>
VMS = Video Merge	alfa/plugin/video_merge	<button>Start New VMS</button> <button>Edit</button> <button>Remove</button>

A total count of "Total: 5" is displayed at the bottom right.

Proof-of-concept

The screenshot shows a web browser window titled "web-app" displaying the "V-PRISM Manager". The URL is "localhost:8080/#/vms/new/5da332970f6e1d0024209d97/". The left sidebar has a navigation menu with the following items:

- Home
- VMS
 - Running
 - VMS
- Locations
- Devices
- VMS Types Plugins
- VMS Types SRC
- Github

The main content area is titled "New VMS" and displays "Details of VMS Type". It shows the following information:

- VMS Type:** VMS = Noise Detector
- Docker Image:** alfa/plugin/noise_detector
- Description:** This send to a MQTT server the noise captured
- Startup Parameters Example:** SENSITIVENESS TOPIC
MQTT_SERVER MQTT_PORT | Example 0.02 topic_alert
172.17.0.1 1883

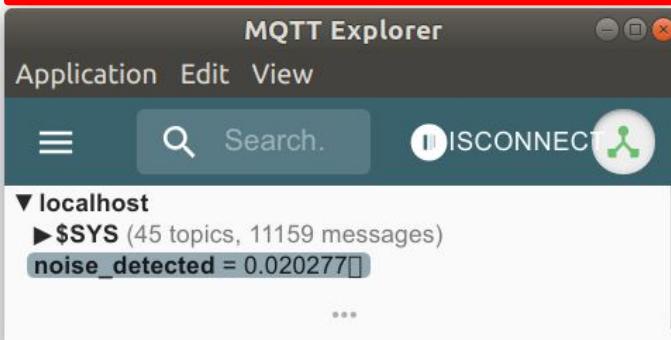
Below this, there are input fields for "Name:" and "Startup Parameters:", both containing the value "Light Noise Detector". At the bottom are two buttons: "Start" (blue) and "Back" (grey).

Proof-of-concept

Running VMS

+ Start a New VMS

Vms Type	Container ID	Startup Parameters	IP	Binded To	Status / Up Time	Actions			
Noise Detector	b9559f970816	0.02 noise_detected 172.17.0.1 1883	172.17.0.3	Local Mic hw:0 #1	Running / Up About an hour				



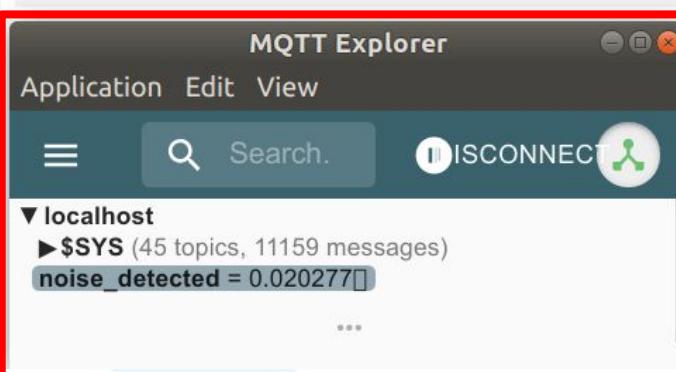
```
battisti@battisti-VirtualBox: ~
File Edit View Search Terminal Help
~$ docker ps --format 'table {{.ID}}\t{{.Image}}\t{{.Command}}\t{{.RunningFor}}'
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
b9559f970816        alfa/plugin/noise_detector   "/start.sh '0.02 no..."   About an hour ago
0b28ed21b64a        alfa/src/mic_device       "/start.sh '5d8f68c..."   About an hour ago
2c1d33a99f06        api_app              "docker-entrypoint.s..."  2 days ago
49bab60166d6        mongo                "docker-entrypoint.s..."  8 days ago
f915d928f7ae        eclipse-mosquitto      "/docker-entrypoint..."   8 days ago
~$
```

Proof-of-concept

Running VMS

+ Start a New VMS

Vms Type	Container ID	Startup Parameters	IP	Binded To	Status / Up Time	Actions			
Noise Detector	b9559f970816	0.02 noise_detected 172.17.0.1 1883	172.17.0.3	Local Mic hw:0 #1	Running / Up About an hour	 Bind SRC	 Details	 Stop	 View



```
battisti@battisti-VirtualBox: ~
File Edit View Search Terminal Help
~$ docker ps --format 'table {{.ID}}\t{{.Image}}\t{{.Command}}\t{{.RunningFor}}'
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
b9559f970816      alfa/plugin/noise_detector   "/start.sh '0.02 no..."   About an hour ago
0b28ed21b64a      alfa/src/mic_device        "/start.sh '5d8f68c..."   About an hour ago
2c1d33a99f06      api_app              "docker-entrypoint.s..."  2 days ago
49bab60166d6      mongo                "docker-entrypoint.s..."  8 days ago
f915d928f7ae      eclipse-mosquitto       "/docker-entrypoint..."   8 days ago
~$
```

Proof-of-concept

Running VMS

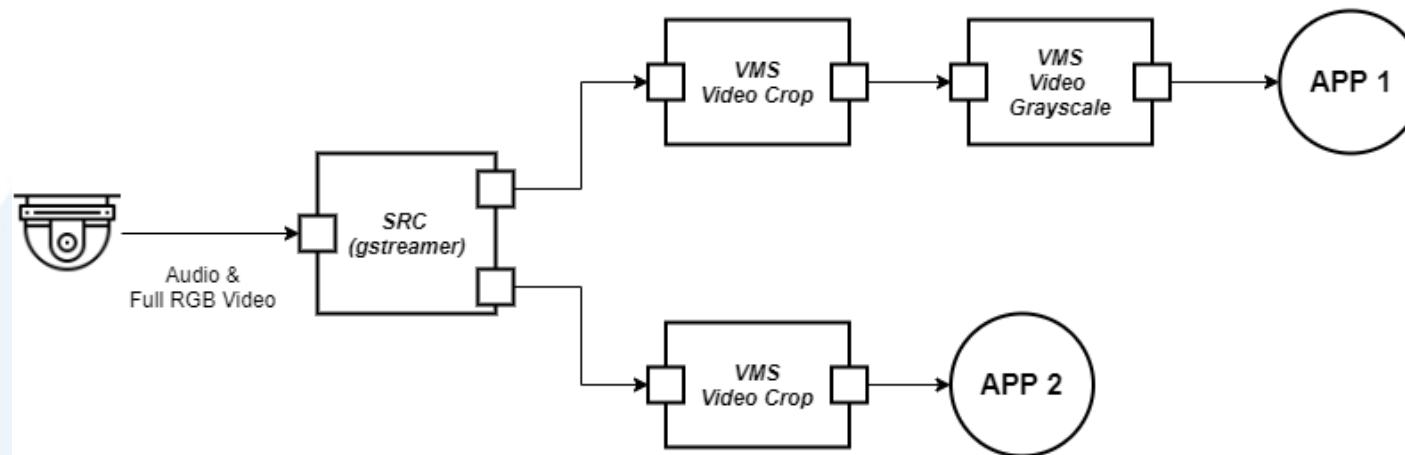
+ Start a New VMS

Vms Type	Container ID	Startup Parameters	IP	Binded To	Status / Up Time	Actions			
Noise Detector	b9559f970816	0.02 noise_detected 172.17.0.1 1883	172.17.0.3	Local Mic hw:0 #1	Running / Up About an hour				

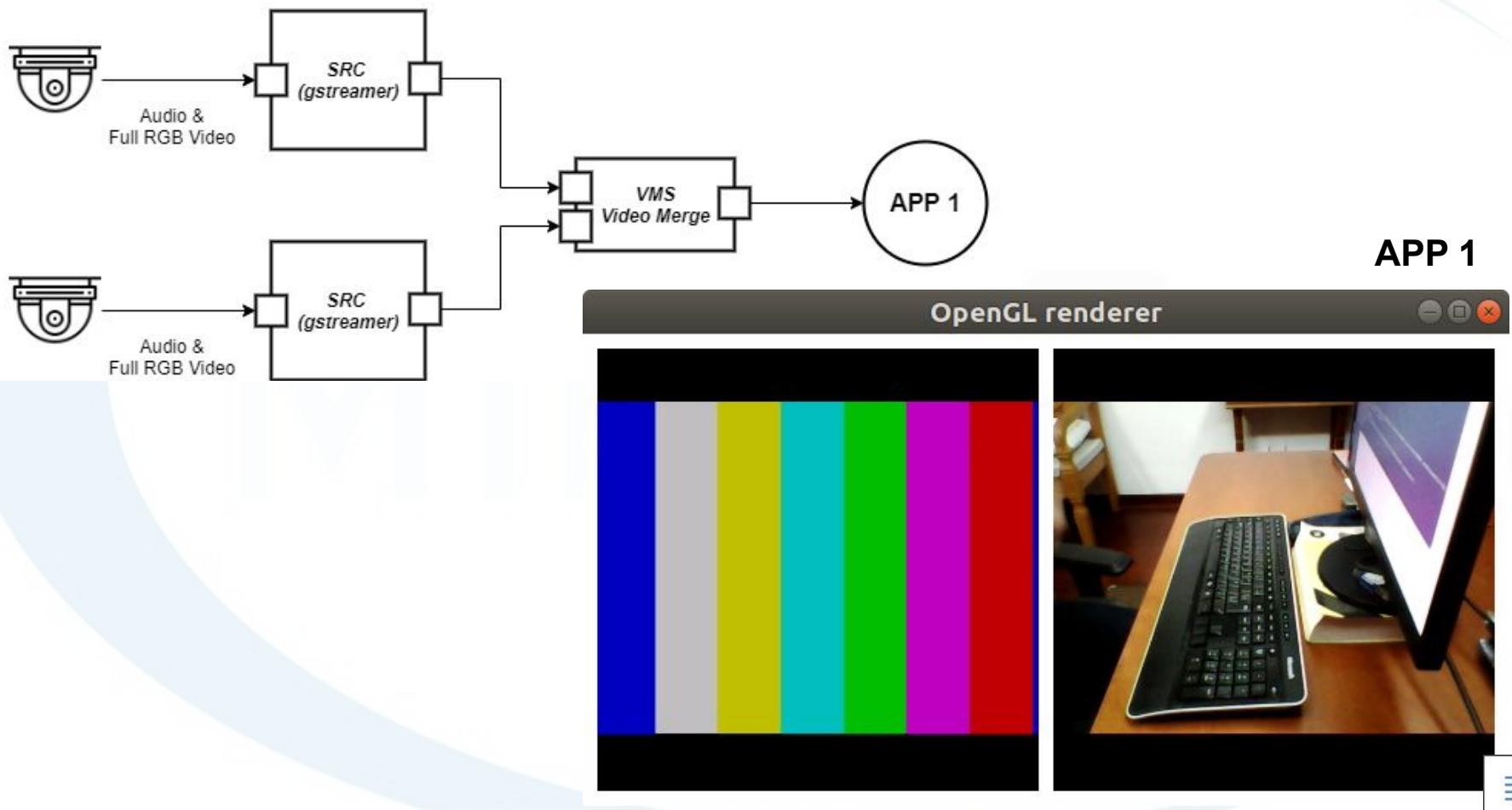
The screenshot shows a desktop interface with two main windows. On the left is the 'MQTT Explorer' application, which has a dark-themed interface with tabs for Application, Edit, View, and a search bar. It displays a list of topics under 'localhost' and a message 'noise_detected = 0.020277'. On the right is a terminal window titled 'battisti@battisti-VirtualBox: ~'. The terminal is running a command to list Docker containers and shows the following output:

```
~$ docker ps --format 'table {{.ID}}\t{{.Image}}\t{{.Command}}\t{{.RunningFor}}'
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
b9559f970816        alfa/plugin/noise_detector   "/start.sh '0.02 no..."   About an hour ago
0b28ed21b64a        alfa/src/mic_device      "/start.sh '5d8f68c..."   About an hour ago
2c1d33a99f06        api_app              "docker-entrypoint.s..."  2 days ago
49bab60166d6        mongo                "docker-entrypoint.s..."  8 days ago
f915d928f7ae        eclipse-mosquitto    "/docker-entrypoint..."   8 days ago
~$
```

Proof-of-concept

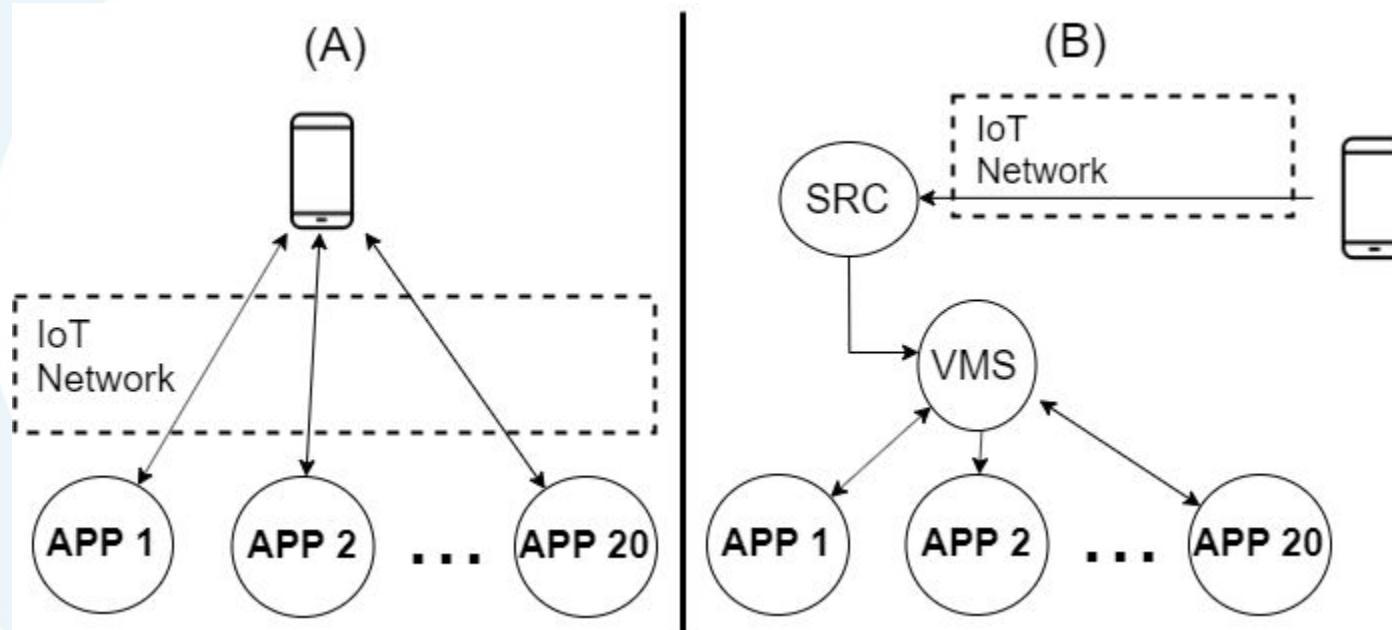


Proof-of-concept



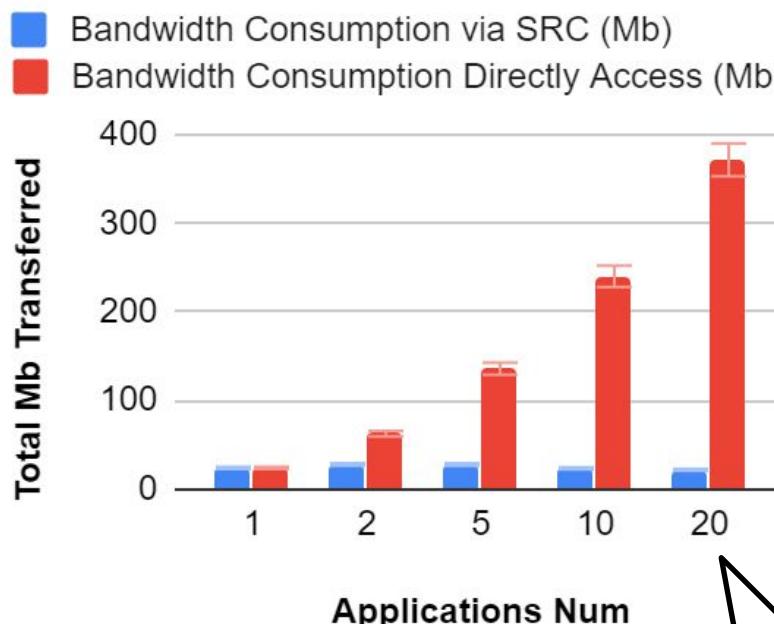
Experiment to assess battery, CPU and bandwidth.

- 20 applications to visualize image produced by a device.

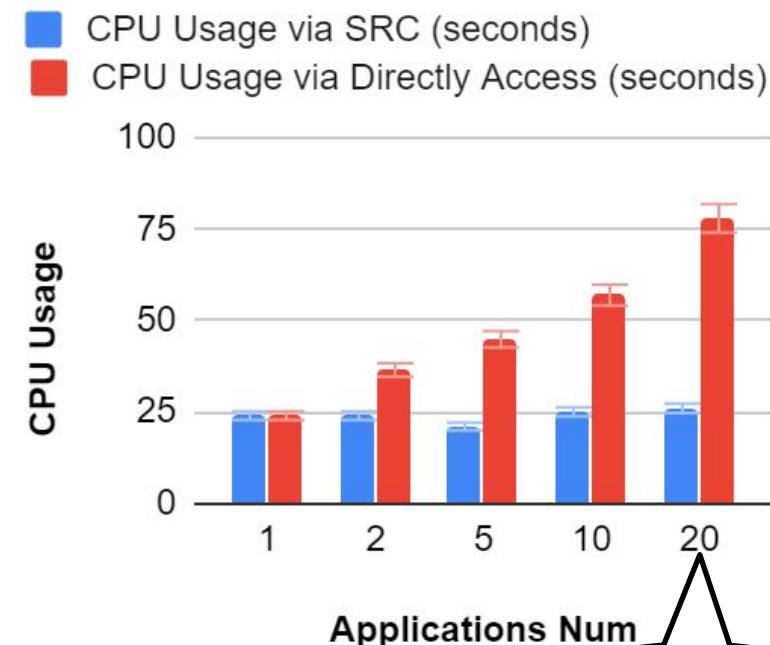


Experiment to assess battery, CPU and bandwidth.

- Each experiment was executed for 5 minutes;
- Data was collected using the *BatteryStats*.



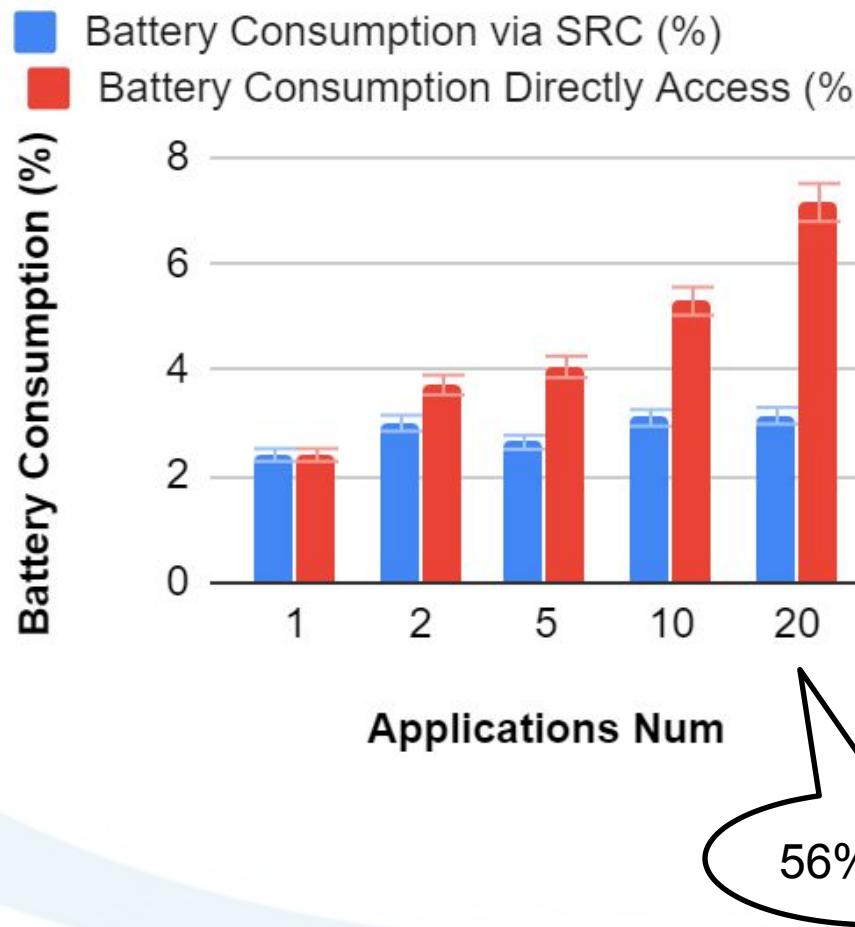
94% gain



66% gain



Experiment to assess battery, CPU and bandwidth.



Final Remarks

- We **propose** and evaluate **an architecture** to create and manage **virtual** multimedia **sensors**;
- V-PRISM adoption reduces:
 - CPU usage;
 - Battery consumption;
 - IoT network bandwidth.
- We show that a single device can be used as source of multimedia data to several VMS
 - Serving multiple applications;
 - Increasing the ROI for infrastructure providers.



Future Work

- We are developing VMS Types to perform more complex tasks like facial recognition and speech detection in the Edge;
- We are making test using small computers like Raspberry Pi to run V-PRISM VMS and SRC.

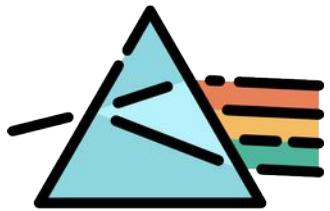


github.com/midiacom/alfa



youtube.com/watch?v=GeKvoEdtYDw





V-PRISM

Anselmo Luiz Éden Battisti
anselmo@midiacom.uff.br

