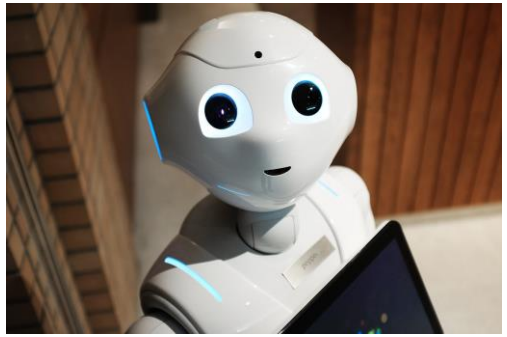
A SMART CAMERA FOR THE IOT

Giuseppe Bisicchia
Jacopo Bandoni

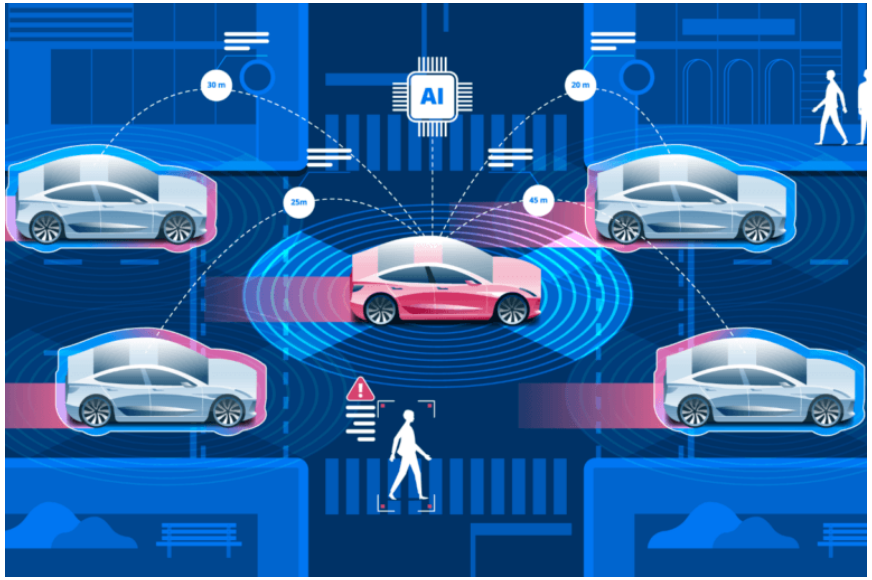
PERVASIVE IOT

Domotics

Robotics

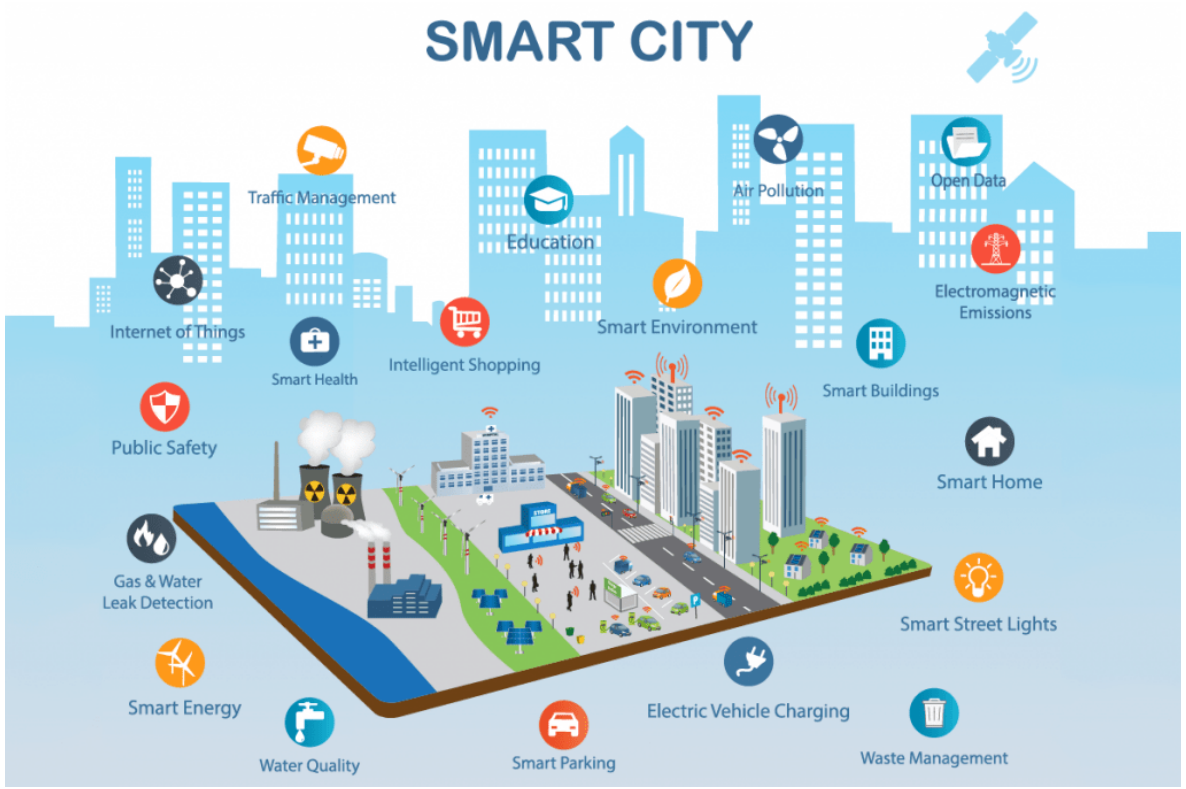


Smart Wearable

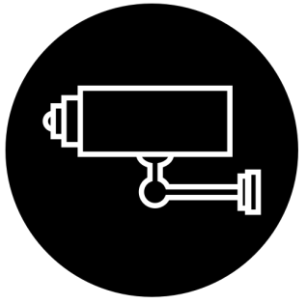


Autonomous Vehicle

SMART CITIES



SMART CAMERAS



CLOUD APPROACH

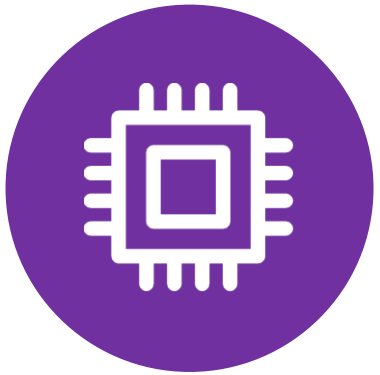
SINGLE POINT OF FAILURE

HIGH COMPUTATIONAL AND STORAGE POWER

RISK OF OVERLOADING

LOCAL APPROACH

NO INTERNET CONNECTION



LIMITED COMPUTATIONAL AND STORAGE POWER

RISK OF DATA LOSS

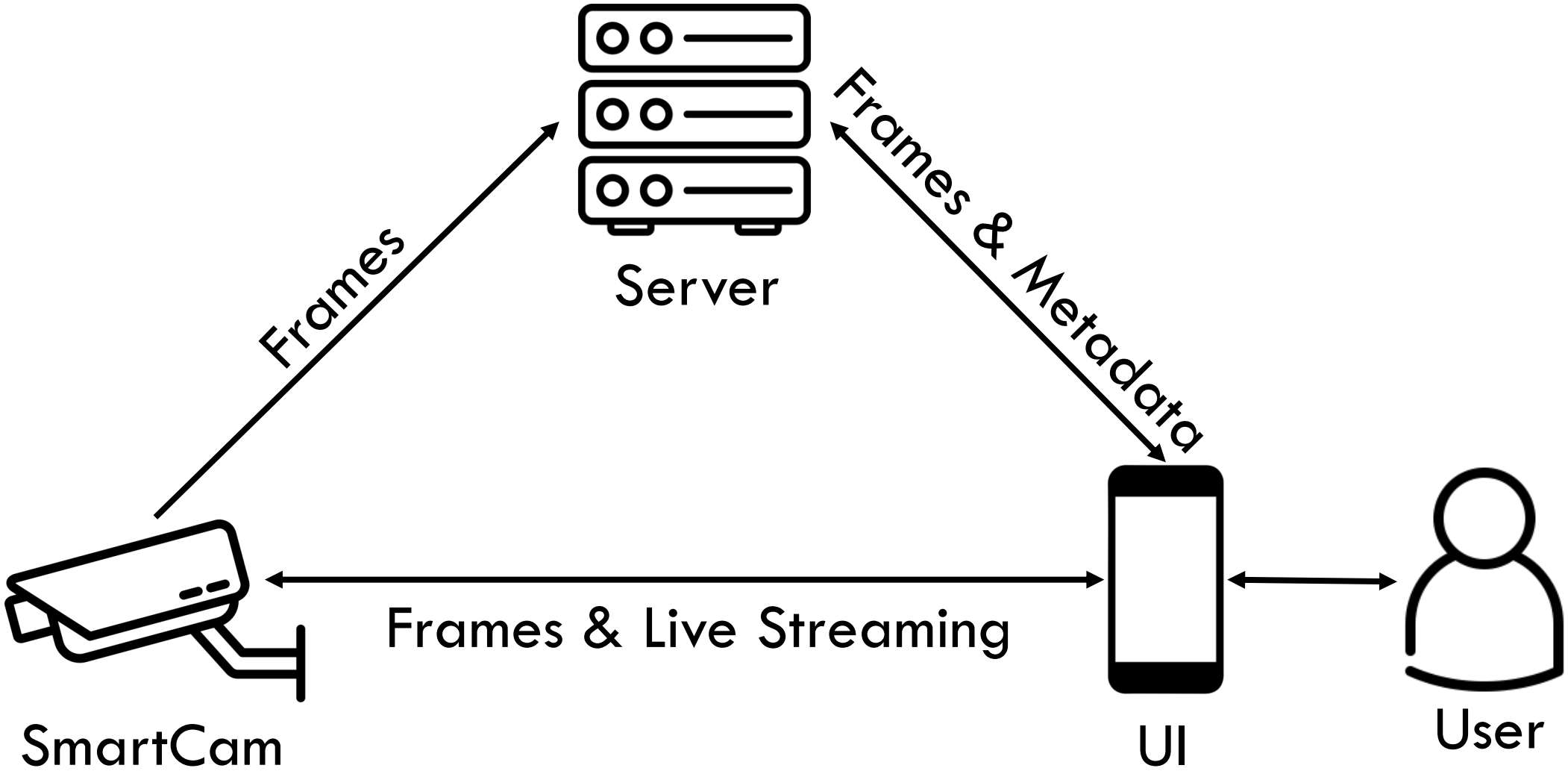
USE CASE: SMART CITY

- In a smart city there can be numerous heterogeneous IoT cameras.
- It is not possible to guarantee that cameras are always connected to the Internet.
- It is important that the management of the network is as efficient as possible given the amount of data.
- It is necessary to have a global view of the whole city.
- It may also be necessary to focus on a specific area to make detailed observations.
- In some cases the cameras can be placed in areas where nothing happens most of the time.

THE PROBLEM

Is it possible to design a system of several cameras that is efficient from network point of view and that is able to work both alone and integrated with a centralised cloud?

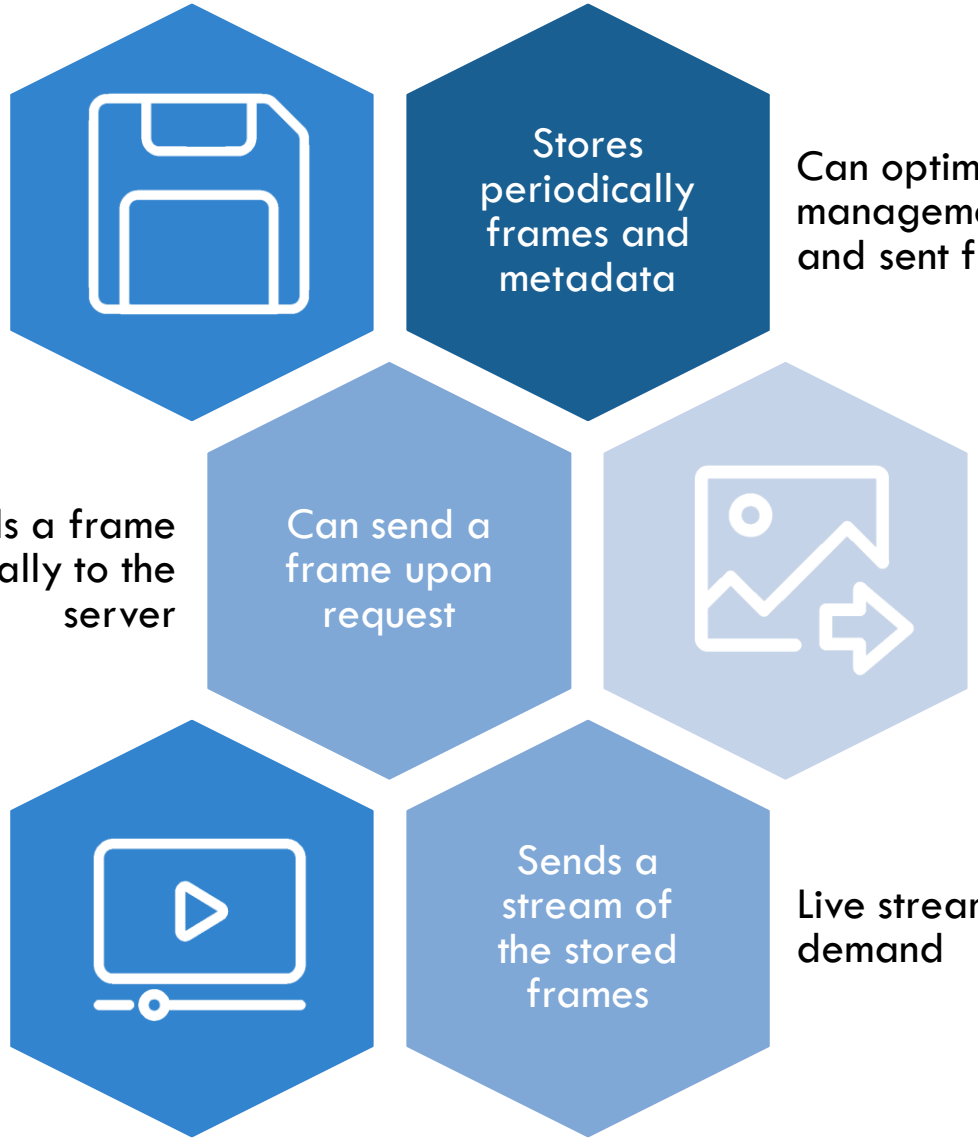
A HYBRID APPROACH



SERVER

- Stores frames and metadata
- Send a frame upon request
- Send a stream of the stored frames

SMARTCAM

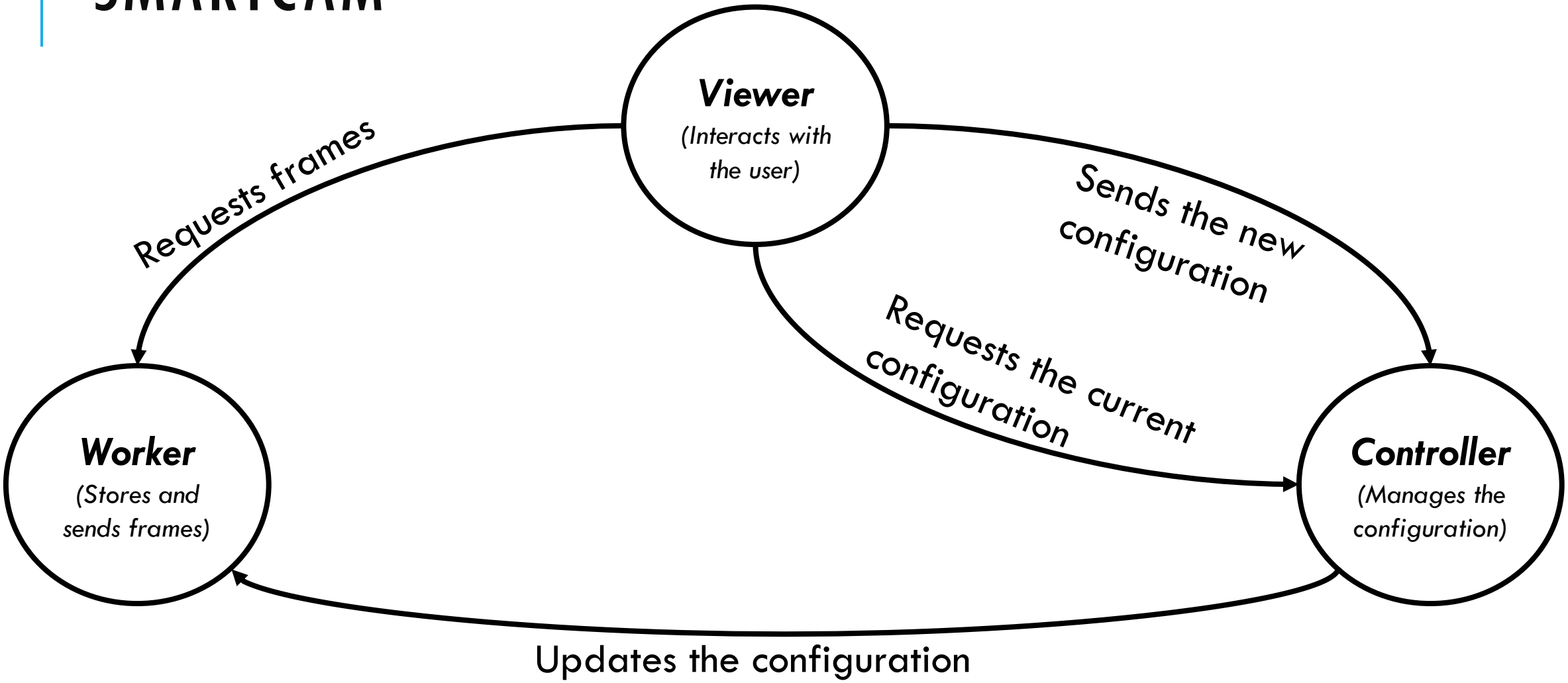


Can optimise the management of stored and sent frames

Sends a frame periodically to the server

Live streaming on demand

SMARTCAM


Viewer
(Interacts with the user)
Requests frames
Sends the new configuration
Requests the current configuration
Worker
(Stores and sends frames)
Controller
(Manages the configuration)
Updates the configuration

USER INTERFACE

- Establishes the connection with the camera and the server
- Retrieves a preview of frames from the server
- Retrieves higher quality data from the camera when needed

IMPLEMENTATION: THE SERVER

Swagger
Supported by SMARTBEAR

/openapi.json

Explore

Server 0.1 OAS3

/openapi.json

Server REST API

Servers

Frames

GET /frames Return a streams of frames

POST /frames Store a frame

Frame

GET /frames/{frame_id} Get a frame

python Flask SQLAlchemy Swagger Supported by SMARTBEAR pythonanywhere

IMPLEMENTATION: THE SMARTCAM

python

Flask

Swagger™
Supported by SMARTBEAR

OpenCV

scikit-image
image processing in python

Raspberry Pi®

SmartCam 0.1 OAS3

/openapi.json

SmartCam REST API

Servers

Controller

GET	/controller	Get the SmartCam Configuration
POST	/controller	Change the SmartCam Configuration
GET	/status	Get the SmartCam Status

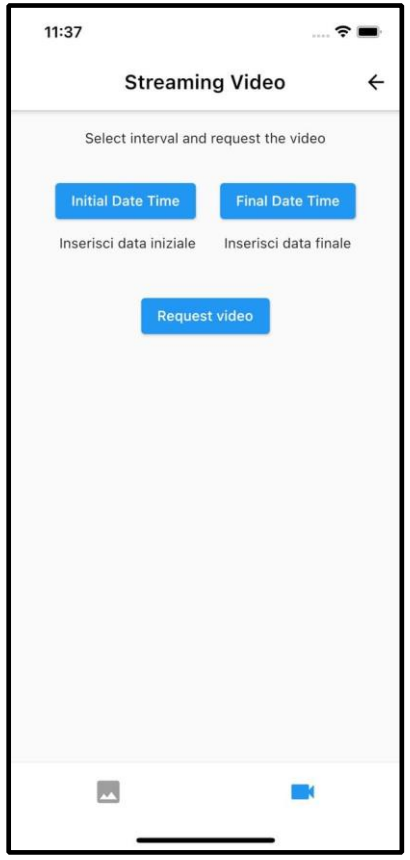
Viewer

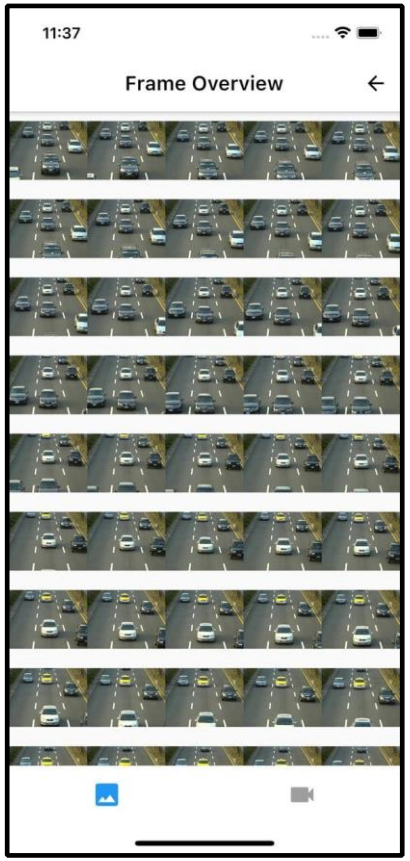
GET	/last	Get the last stored frame
GET	/live	Live streaming
GET	/photo	Get the current frame
GET	/video	Get the stream of local frame

Schemas

Controller ›

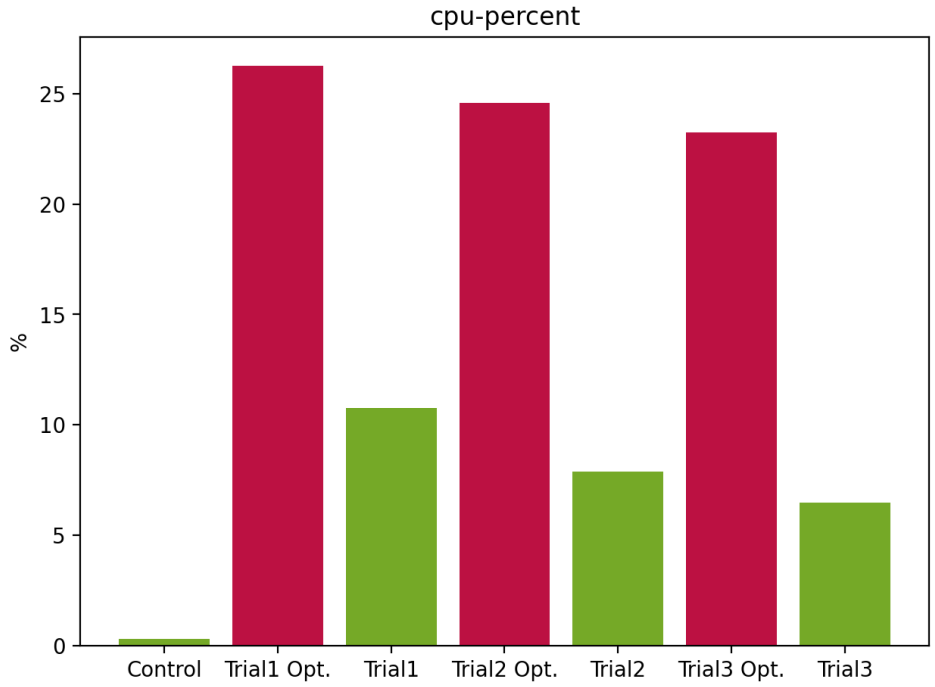
IMPLEMENTATION: THE USER INTERFACE

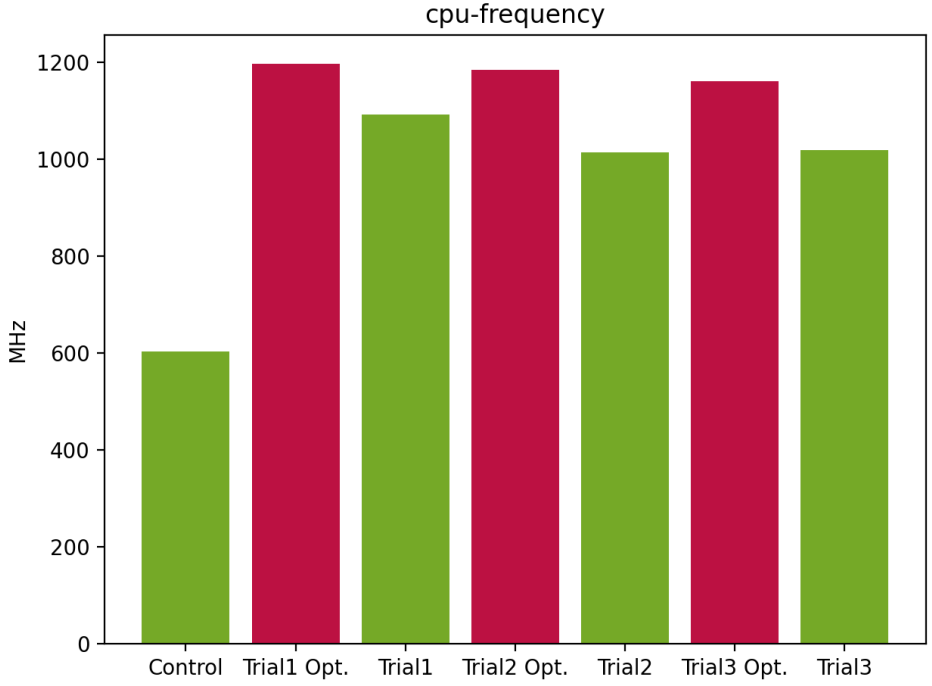


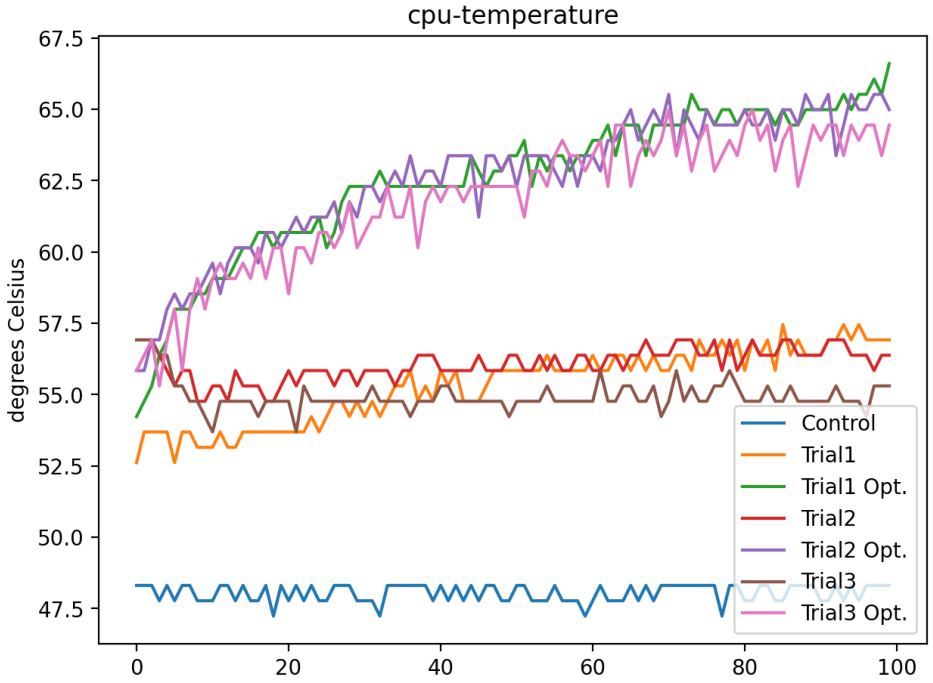


Flutter

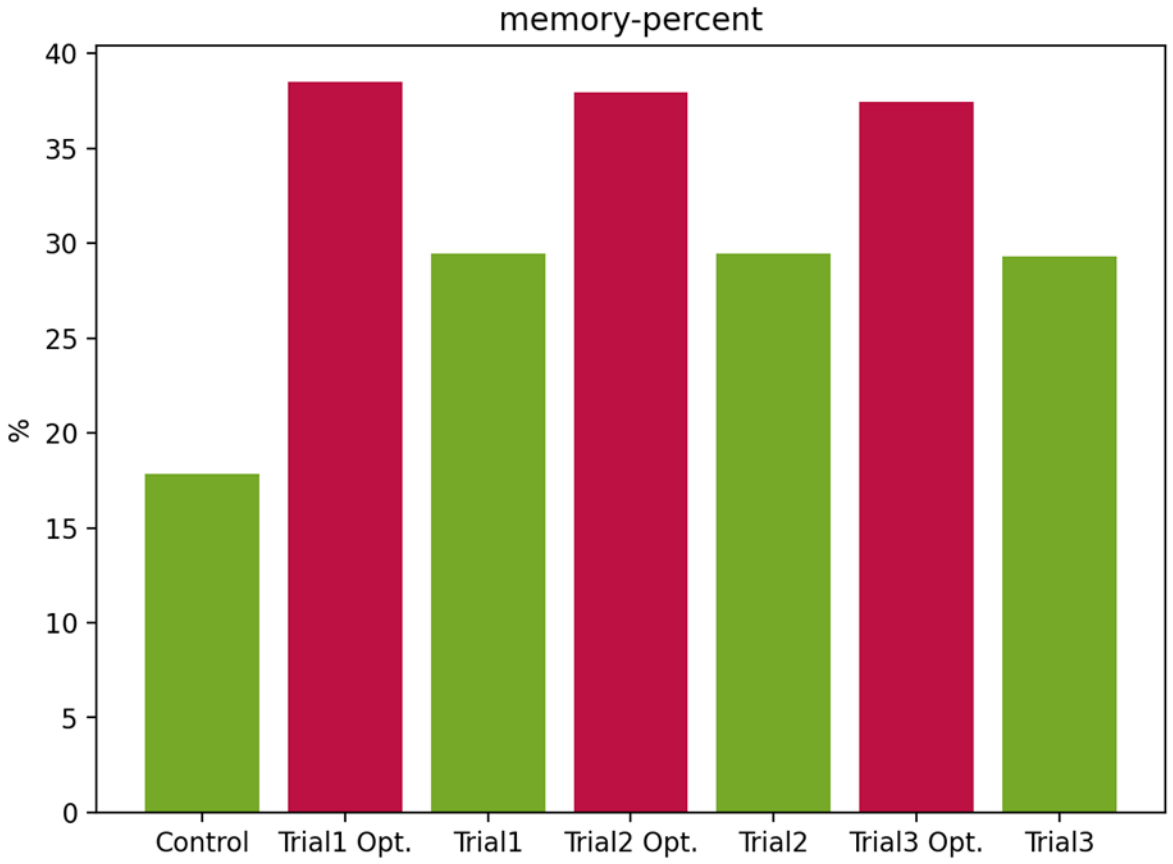
EXPERIMENTS: CPU

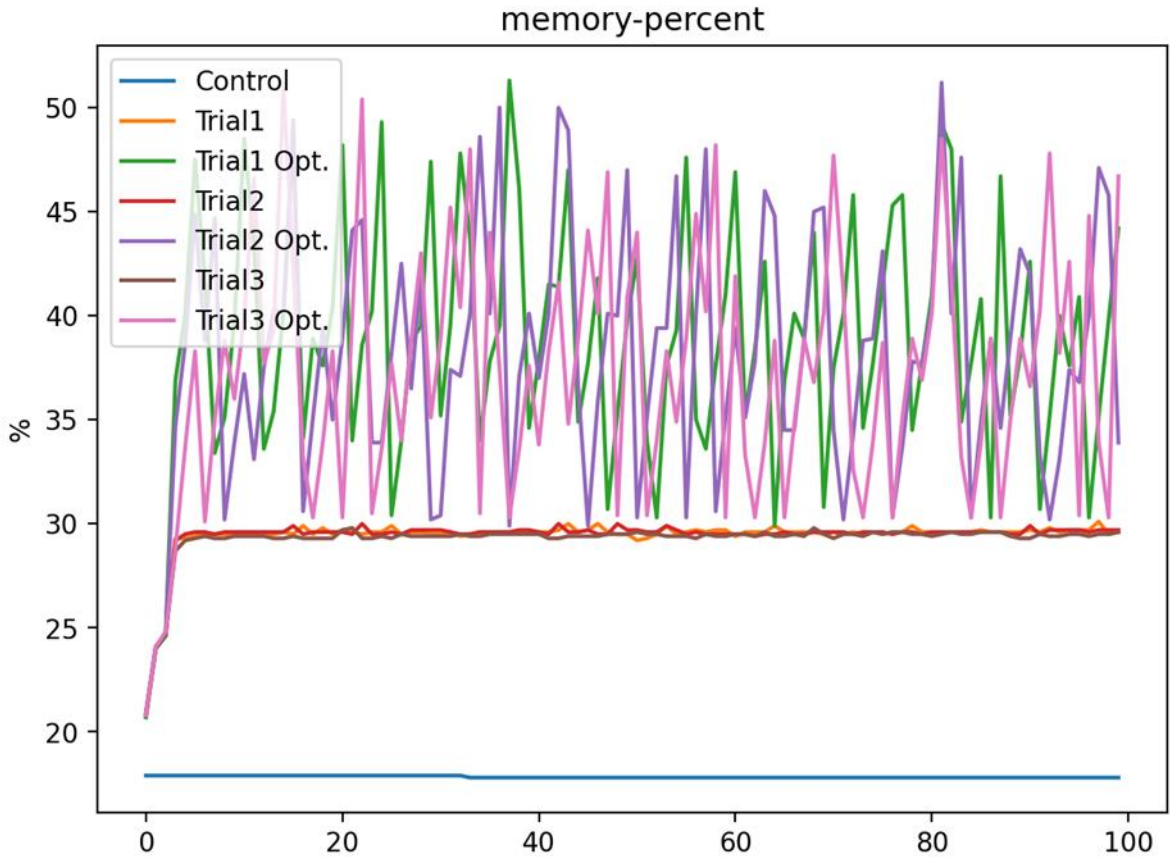




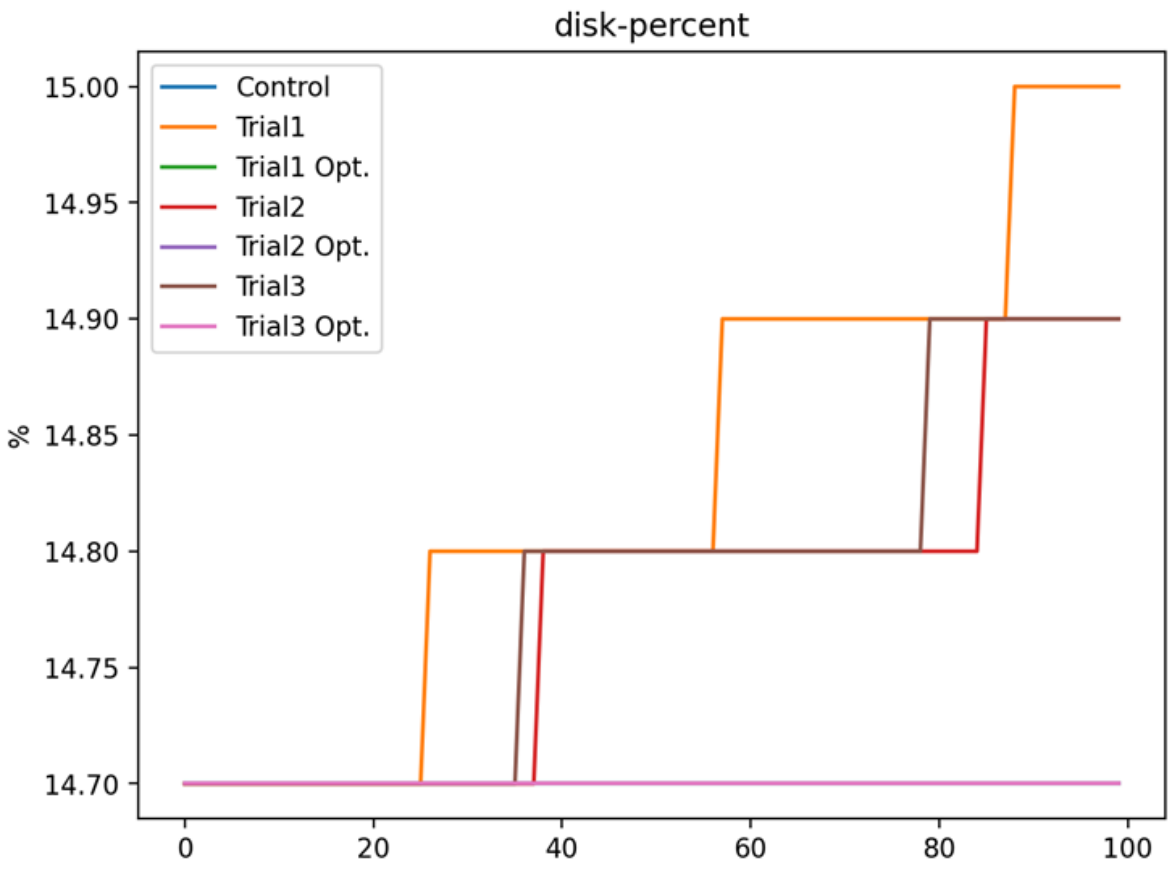


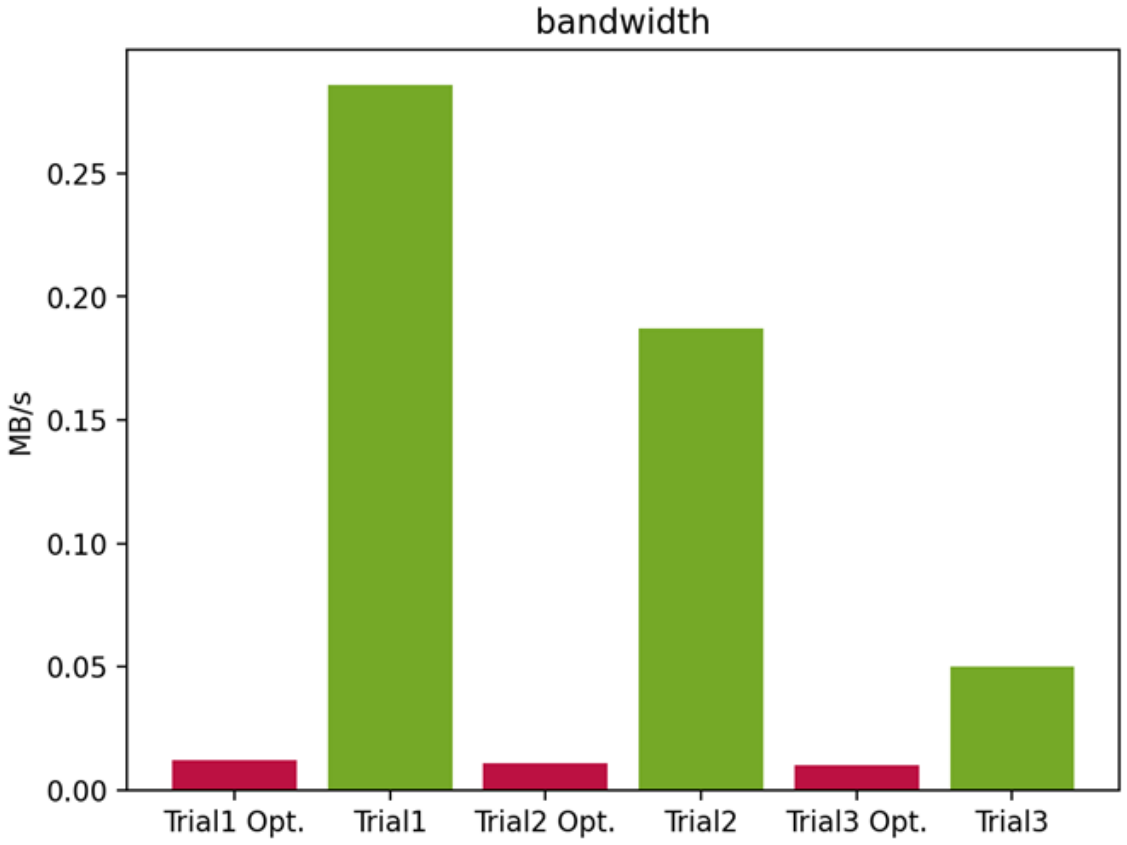
EXPERIMENTS: MEMORY



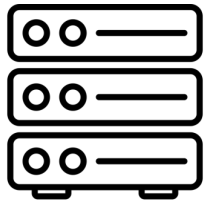


EXPERIMENTS: DISK & BANDWIDTH



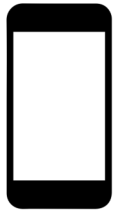


CONCLUSIONS



Hybrid approach that seeks to mitigate the cons of totally cloud or local solutions

SmartCam configurable on demand with possible options for tradeoff between the use of the CPU and quantity of stored and sent frames



User Interface that allows to interact both with the cloud and the cameras

THANKS FOR YOUR ATTENTION!