6th UCAAT User Conference on Advanced Automated Testing

ETSI

Testing Solution for VR apps

Presented by Carlos Cárdenas (DEKRA)

DEKRA

Motivation Facts

- Mobile applications consume data differently depending on various network conditions.
- Carries need to understand how the most popular Android and iOS apps consume data from the network.
- Carriers need to understand the network conditions that drive poor/excellent User Experiences.
- Carriers need to test an app in the exact same manner that customer use apps. No simulations…just real apps consuming real data.

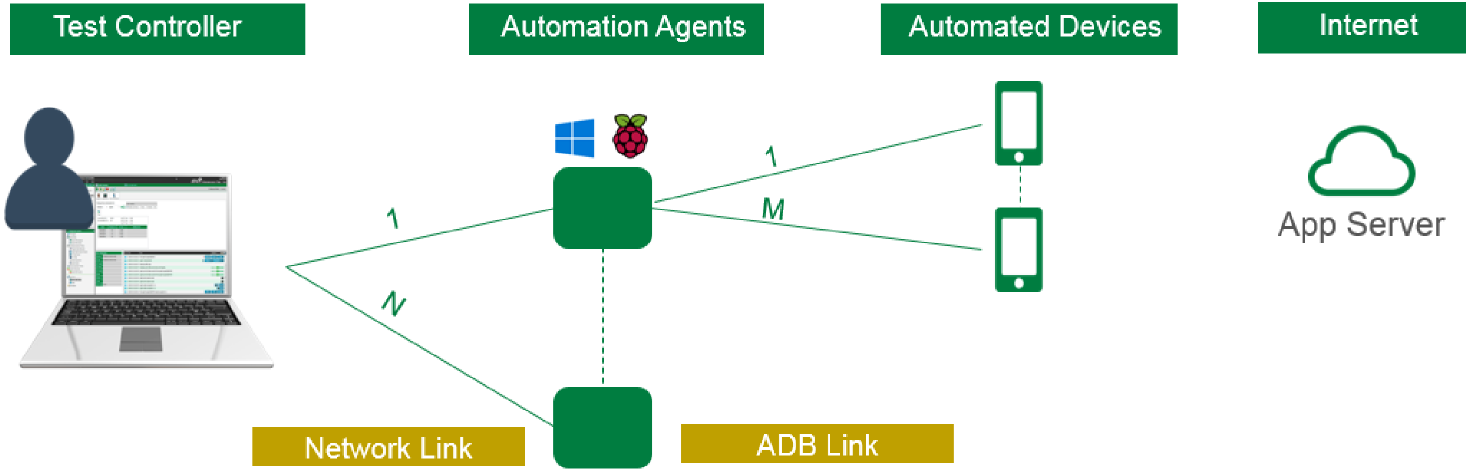
DEKRA’s current solution for non VR apps

- Non VR Apps:
  - Downlink Intensive Video Streaming (including 4k)
  - Uplink Intensive Video Streaming
  - Two-way Video Streaming
  - Social Media

DEKRA’s current solution for non VR apps

Testing Topology:

High Scalability: M x N devices can be automated simultaneously



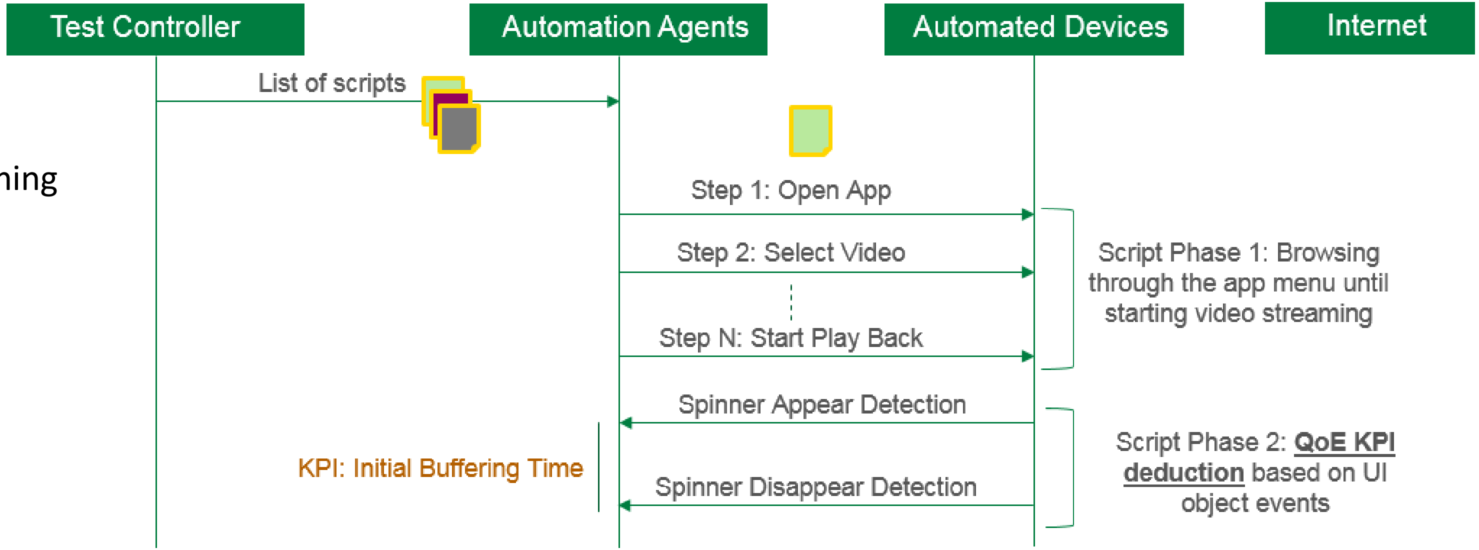
Automation Agents can be deployed in remote sites

ADB (Android Device Bridge) can use USB or Wi-Fi. in either case, Automated Devices must be in locally attached to the Automated Agent.

DEKRA's current solution for non VR apps

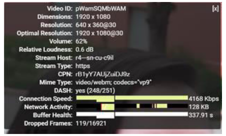
Automation Test Flow:

Example – Video Streaming App



DEKRA’s current solution for non VR apps

- Appium [*open source test automation framework for use with native, hybrid and mobile web apps*] for
  - Browsing through the App menu
  - Recognizing UI objects (e.g., spinner, progression bar)
- ADB (Android Device Bridge) for device data consumption reporting.
- OCR (Optical Character Recognition) for extracting App information:



Video Resolution
Buffer Health
….

DEKRA's current solution for non VR apps

The following KPIs have been proved:

Mobile Apps	KPIs
All (App Agnostic)	Battery, Data Usage, Throughput
Netflix	Initial Buffering, Re-bufferings
YouTube	Initial Buffering, Re-bufferings, Video Resolution
Instagram	Access Time, Initial Buffering, Re-bufferings
Periscope	Initial Buffering, Re-bufferings
Skype Video Call	Call Setup Time, Call Result, MOS
WhatsApp	Sharing Time, MOS
…	

DEKRA’s current solution for non VR apps

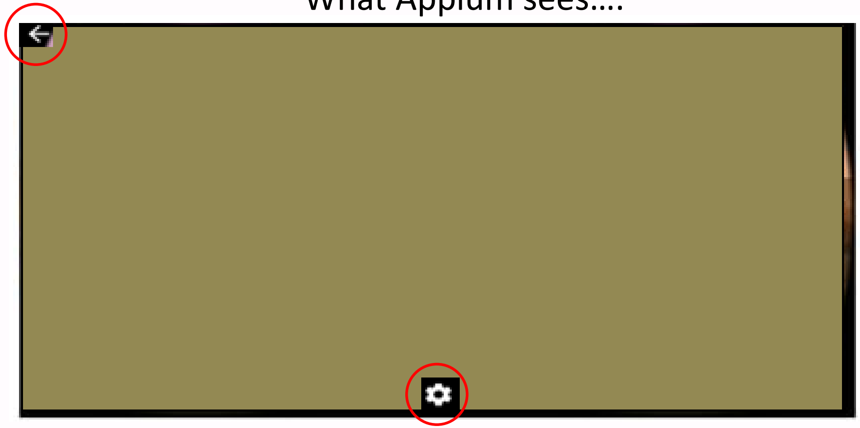
- Limitations of this approach for testing VR/Gaming apps:
  - Performing Movement
    - VR and gaming apps require physical movement of the hosting device. As the gyroscope and accelerometer cannot be mocked, a hardware platform is required.
  - Retrieving App state:
    - Unlike other apps, VR and gaming apps are programmed in an Android UI Canvas where the graphical engine works (e.g., Open GL). Appium (or similar) cannot recognize UI objects inside the App gfx canvas.

DEKRA’s current solution for non VR apps

- Limitations of this approach for testing VR/Gaming apps:

Actual App UI

What Appium sees….

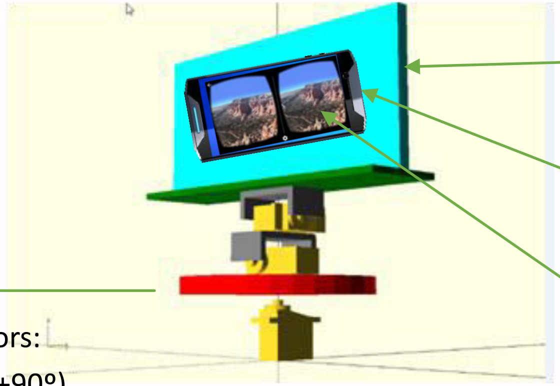


Testing solution for VR/Gaming apps

- In order to overcome those limitations we have upgrade the architecture:

Automation Agents

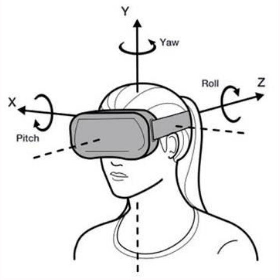
USB



Robotic Platform

Automated Device

VR App Under Test

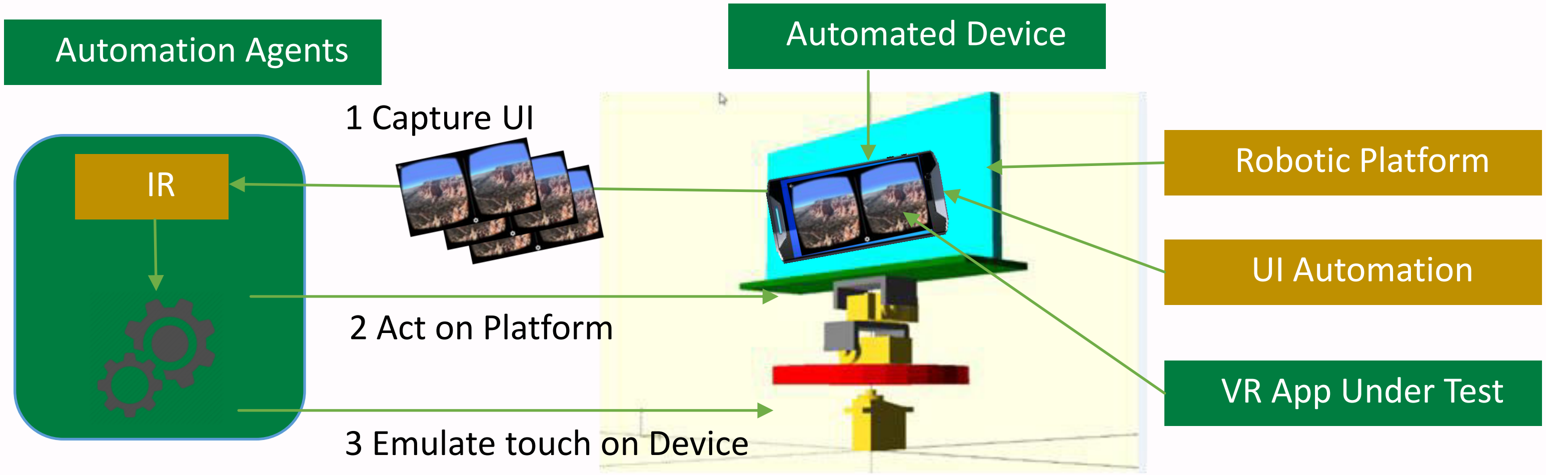


Four Servo-Motors:

1. Pitch (-90º, +90º)
2. Yaw 1 (-180º, 0º)
3. Yaw 2 (0º, +180º)
4. Roll (-90º, +90º)

Testing solution for VR/Gaming apps

Architecture



DEKRA

Test Solution Requirements

Key Performance Indicators

- "Time to load a virtual scene" (t_2-t_1), where
  - t_1 = user clicks on "start scene/experience" button
  - t_2 = the scene is totally rendered
- "Lagging" (t_4-t_3), where
  - t_3 = user sends command to the app (e.g., roll phone)
  - t_4 = device UI shows command response (e.g., airplane has rolled)
- "Frame per seconds" as smoothness indicator…
- "Data Consumption"

Test Solution Requirements

Performance Requirements

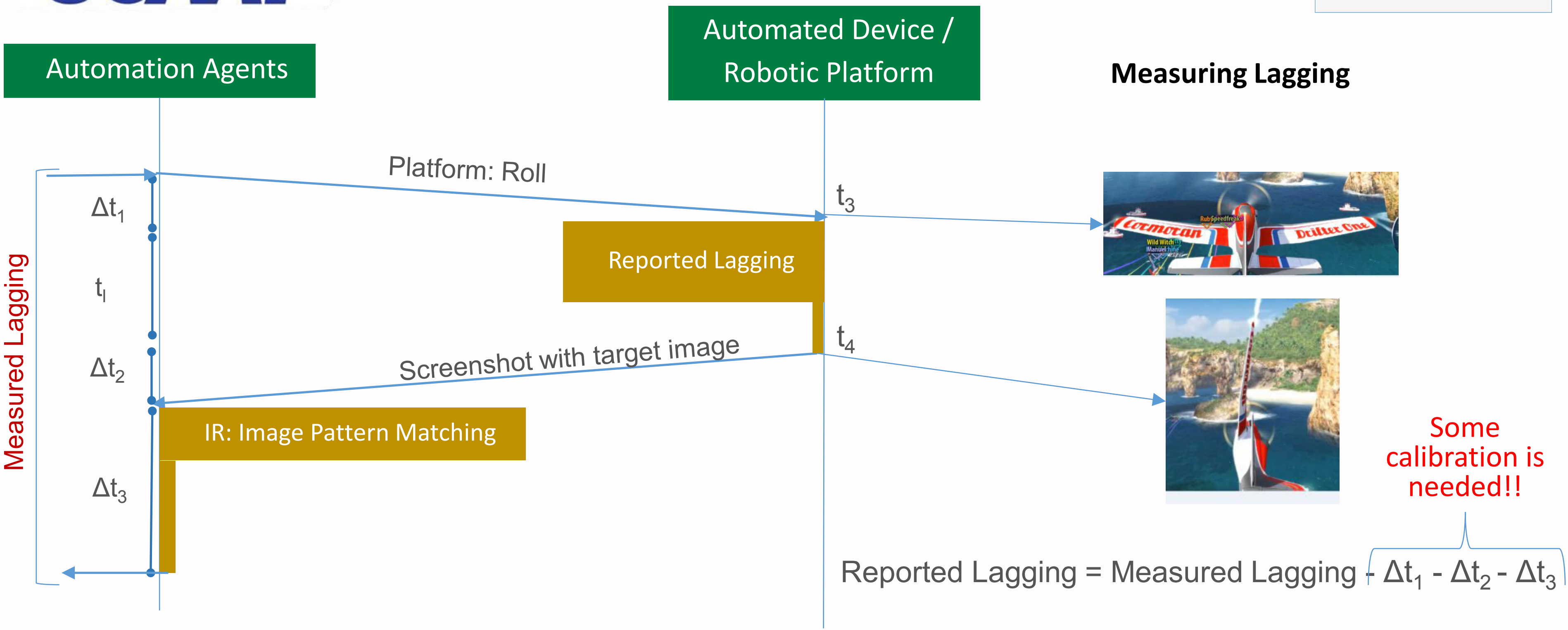
- Minimize “reaction time” t_5-t_6, where
  - t_5 = target appears on the screen
  - t_6 = tap/touch on that target

Why?

VR/Gaming: Automate the browsing through the app where some UI could be moving objects.

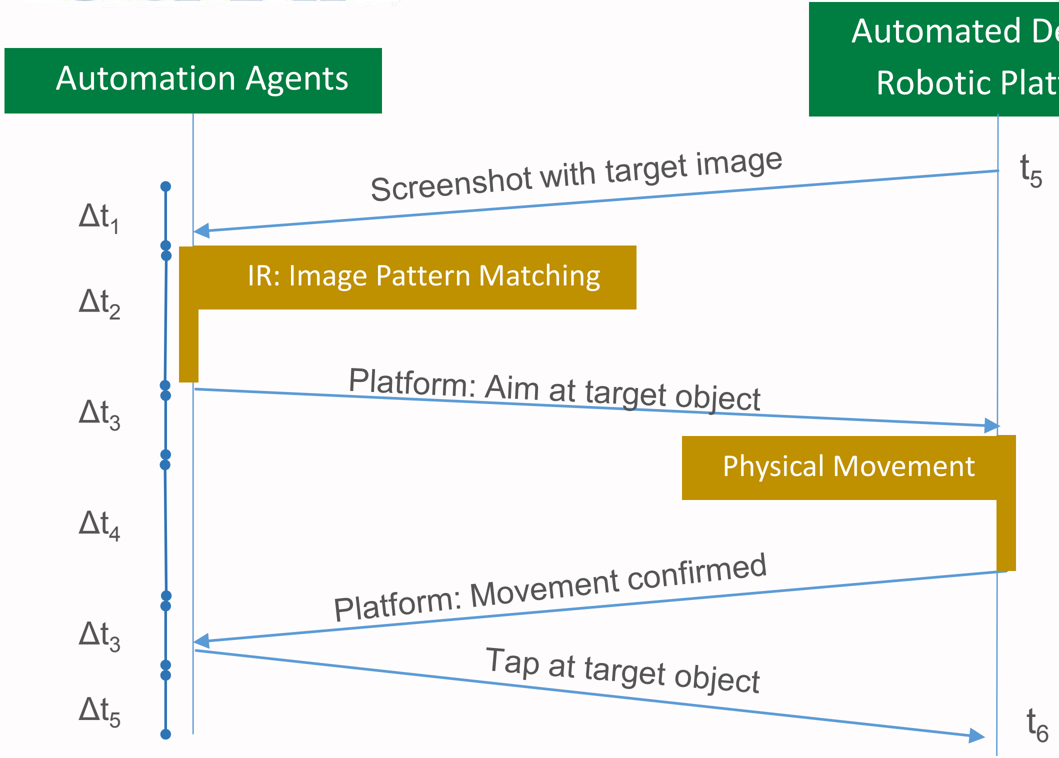
Gaming: Shoot at moving target

Measuring Lagging



6th UCAAT User Conference on Advanced Automated Testing

ETSI The Standards People



More on Reaction Time....

Goal

Average Human Reaction Time: **284 ms**

[https://www.humanbenchmark.com/tests/reactiontime/statistics]

Implemented Reaction Time:

Δt_1 ~1-10 ms → TCP socket latency

Δt_2 ~100-200 ms → Image Recognition performance

Δt_3 ~1-10 ms → Serial Comm latency

Δt_4 ~ 100 ms → Time to aim at object (Robotic action)

Δt_5 ~1-10 ms → TCP socket latency

DEKRA's solution Reaction Time = **(230, 330) ms**

DEKRA

Test Solution Requirements

Performance Requirements

- High performance screen capture
  - Requirement: Higher than 24 frames per second
- Low delay screen touch
  - Requirement: Lower than 10 ms
- IR (Image Recognition)
  - Requirement: High pattern matching accuracy and high performance

Design Parameters

Options	Trade off	
Higher screenshot resolution	Higher Δt_1 and Δt_2	Less measurements, more accurate KPI, slow for gaming apps
	Less false negative IR detections	
	Less true IR positive detections	
Lower screenshot resolution	Lower Δt_1 and Δt_2	More measurements, less accurate KPI, suitable for gaming apps
	More false positive IR detections	
	More true positive IR detections	

The IR matching score is another important trade-off parameter

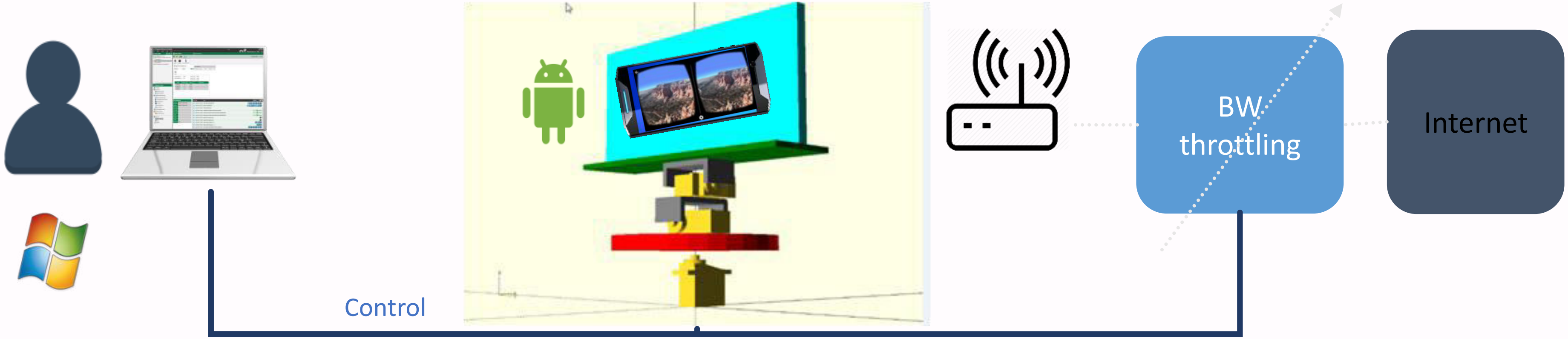
KPIs Implemented

KPI	Definition
Network Resources Usage	Data Usage, Throughput
Device Resources Usage	Battery, CPU, GPU
Time to load the virtual world	Time elapsed from selecting a scenario (world, experience, etc.) to loading the 3D visual context
Immersion Cut-off	Probability that successfully started immersion is ended by a cause other than the intentional termination by the user
Lagging	Time elapsed from acting on the device to the reaction of the UI

Showcase: Testing Google Cardboard App

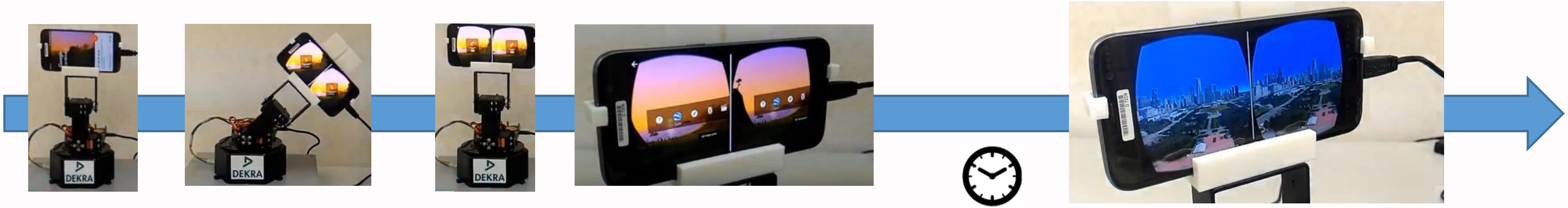
- VR experience, e.g., for Google Earth
- Replacing the mouse by the head movement

Showcase: Testing Google Cardboard App



Automatic test cycles: **<u>40</u> repetitions / BW configuration**

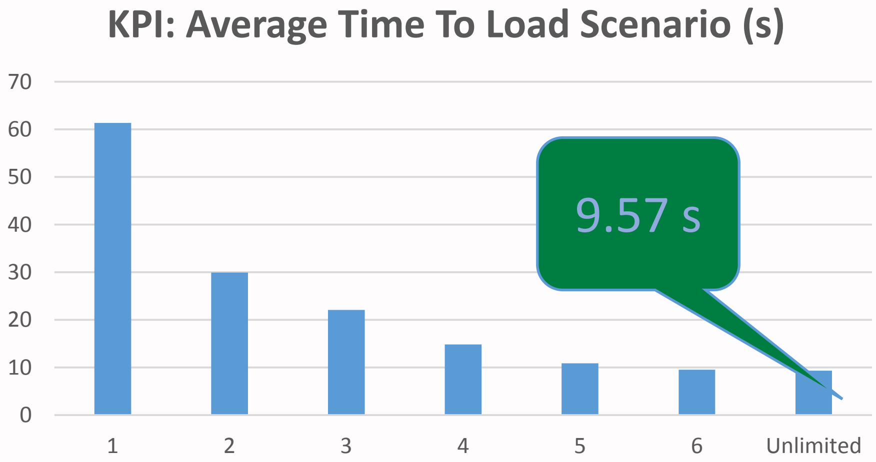
Showcase: Testing Google Cardboard App



Open App ---- Navigate through the app until click “start experience” ----------------------> Measurement

Automatic test cycles: <u>**40**</u> **repetitions / BW configuration**

Showcase: Testing Google Cardboard App



X-Axis: Imposed BW (Mbit/s)

KPI: Time to load scene

9.57 s (best scenario)

KPI: Network Data Usage

8 MB (all scenarios)

Showcase: Testing a Cloud Gaming app

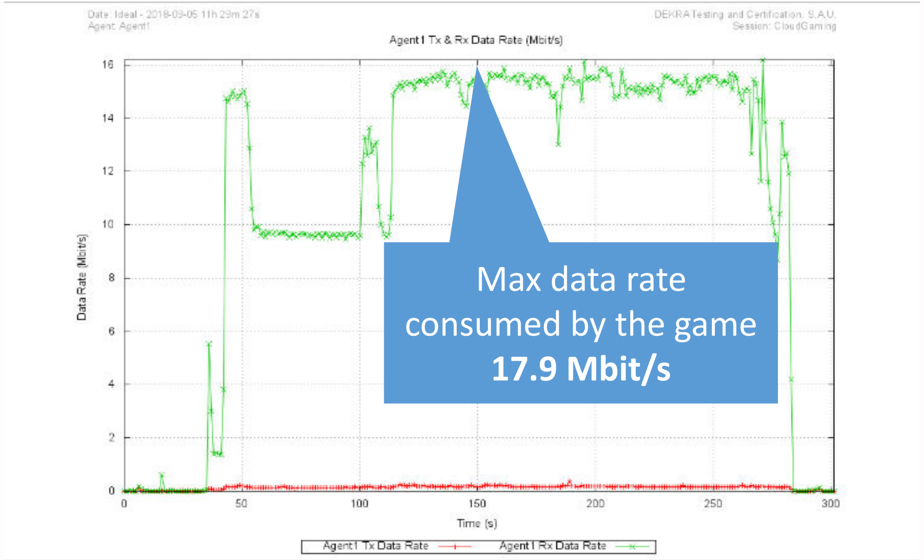
- Test script:
  - Open App, Select game
  - Start game
  - Leave the car until it crashes with the first roadblock in its way (this happens after 280 seconds approximately)
  - Close game

We have selected this use case for repeatability across different network conditions

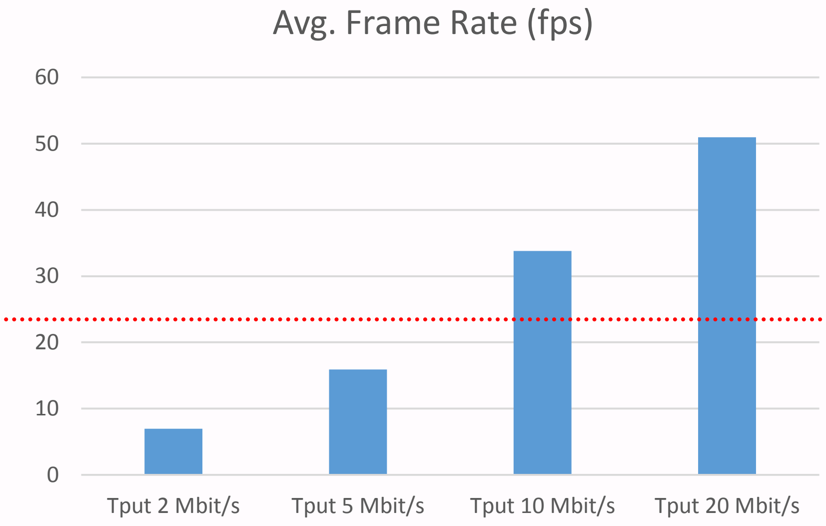
Showcase: Testing a Cloud Gaming app

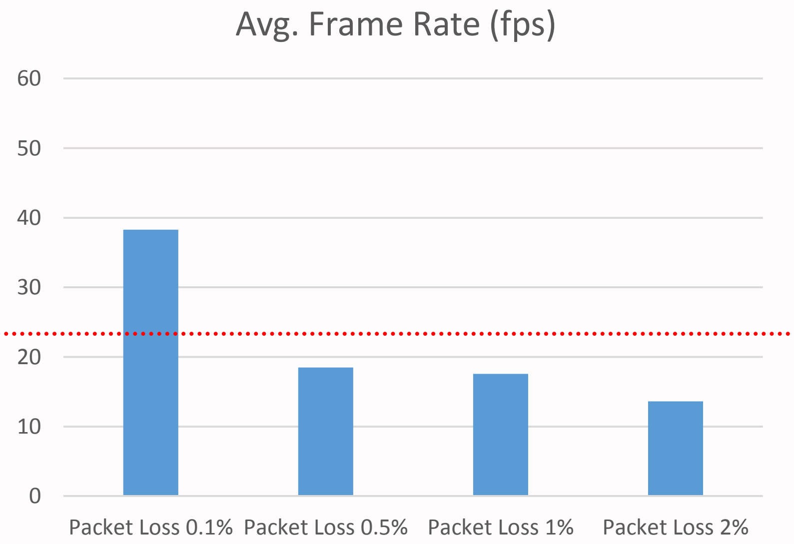
Baseline Network Condition	
RTT*	17 ms
DL Speed	90 Mbit/s
UL Speed	75 Mbit/s

Baseline Frame Rate (fps)	
Min	38
Avg	51.1
Max	55



Showcase: Testing a Cloud Gaming app





Key-takeaways

- The “time to load scenario” KPI is severely impacted by the quality of the network access, mainly the available bandwidth (Mbit/s).
  - Online Virtual Reality apps consumes huge amount of network data, which has impact on network planning and deployments.
  - Online Virtual Reality apps requires high device GPU performance, so need flag-ship device for a good User Experience.
- 4G mobile networks are not suitable for 5G cloud gaming use case because…
  - The frame rate at the user device is unacceptable with a link capacity below 10 Mbit/s, or with a link loss (at IP level) above 0.1 %
  - The frame rate and the lagging at the user device is severely affected by fluctuations in the round trip time of the network (a.k.a. jitter).

Lesson-learnt

- Objective performance measurements provide insights about 5G VR and Gaming use cases.
- Thanks to the fast closed-loop response time of the solution on Android, the solution can be also used to measure online games apps.
- The image recognition library matching score parameter has impact on the accuracy of the “time to load scenario” measurement.
- The testing solution needs another upgrade to automate a gamepad. Online games may use external gamepad (instead of gyroscope/accelerometer) for which the implemented robotic platform is not suitable.

This testing solution has been developed inside the scope of TRIANGLE project

TRIANGLE Project

5G Applications and Devices Benchmarking

Co-funded by the Horizon 2020 Framework Program of the European Union

www.triangle-project.eu

info@triangle-project.eu