

APPLYING VIDEO TEST AUTOMATION TO AUTOMATE MULTIMEDIA VERIFICATION WITH EMBEDDED LINUX SW

AUTOMOTIVE LINUX SUMMIT JAPAN 2018

06/21/2018, TOKYO
NGUYEN NGUYEN
SENIOR ENGINEER
RENESAS DESIGN VIETNAM
RENESAS ELECTRONICS CORPORATION

WHO AM I?

- Name: Nguyen Bao. Nguyen (Nguyen Nguyen)
- Company: Renesas Design Vietnam
- Career: 10 years experiences in embedded software development (both software development and verification)
 - ✓ Development for Multimedia framework on Real-time OS
 - ✓ Development for Multimedia plug-in for Android Stagefright on R-Car software platform
 - ✓ Development and verification for In-vehicle software platform
 - ✓ Development for test automation solutions of In-vehicle software platform
- Email: *nguyen.nguyen.yj@renesas.com*

ABOUT RENESAS AND RENESAS DESIGN VIETNAM



Sales Companies

Renesas Electronics America
Renesas Electronics Canada
Renesas Electronics Brasil-Servicos
Renesas Electronics Europe (UK)
Renesas Electronics Europe (Germany)
Renesas Electronics (China)
Renesas Electronics (Shanghai)
Renesas Electronics Hong Kong
Renesas Electronics Taiwan
Renesas Electronics Singapore
Renesas Electronics Malaysia
Renesas Electronics India
Renesas Electronics Korea

Manufacturing and Engineering Service Companies

Renesas Semiconductor Manufacturing
Renesas Semiconductor Package & Test Solutions
Renesas Semiconductor (Beijing)
Renesas Semiconductor (Suzhou)
Renesas Semiconductor (Malaysia)
Renesas Semiconductor (Kedah)
Renesas Semiconductor Technology (Malaysia)
Renesas Semiconductor KL

Design and Application Technologies Companies

Renesas System Design
Renesas Engineering Services
Renesas Design Vietnam
Renesas Semiconductor Design (Beijing)
Renesas Semiconductor Design (Malaysia)

Business Corporation

Intersil Corporation

- Renesas Design Vietnam Co., Ltd. (RVC) was founded in October 2004, as one of the main design centers in Renesas group.
- Business line: Design of semiconductor for both hardware and software.

AGENDA

- Introduction about Video Automation Test
- Achievement with Video Automation Test Development
- The approach for Video Automation Test
- How to apply Video Automation Test with Fuego
- The future plan

LEGEND

Acronym	Meaning
VAT	Video Automation Test
Fuego	Fuego is a test framework specifically designed for embedded Linux testing .
HDMI	(High-Definition Multimedia Interface)
Ref data	Expected video playback output
VAT PC	The PC used to launch VAT
VAFT component	An application used to control VAT PC from Fuego

A photograph of a car's side profile as it drives on a road. The background is heavily blurred with horizontal streaks of green and yellow, indicating high speed. A dark blue horizontal bar is overlaid across the upper portion of the image.

INTRODUCTION ABOUT VIDEO AUTOMATION TEST

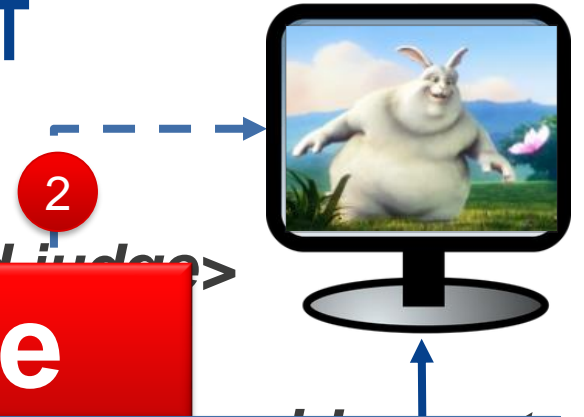
INTRODUCTION ABOUT VIDEO AUTOMATION TEST

THE MOTIVATION

❖ The Video Manual test:

- Test

The test result is un-reliable



❖ DEMAND:

→ Need to automate video testing, to make the reliable test results, NOT depends on Human Feeling

→ Replace “human role” by “**MACHINE ROLE**”

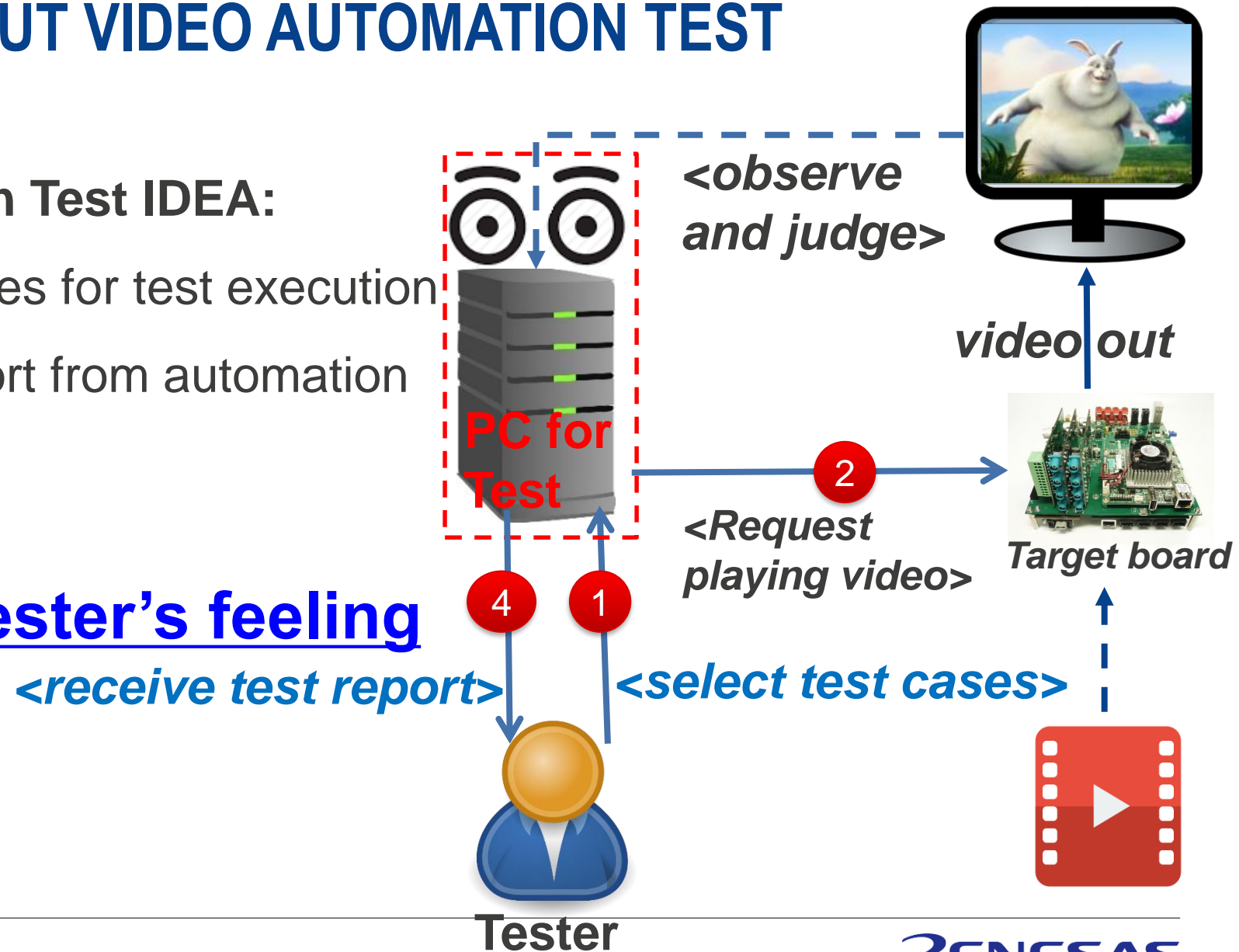
INTRODUCTION ABOUT VIDEO AUTOMATION TEST

POPOP THE IDEA

❖ The Video Automation Test IDEA:

- Tester **selects** test cases for test execution
- Tester **checks** the report from automation judgment
- The test result is:

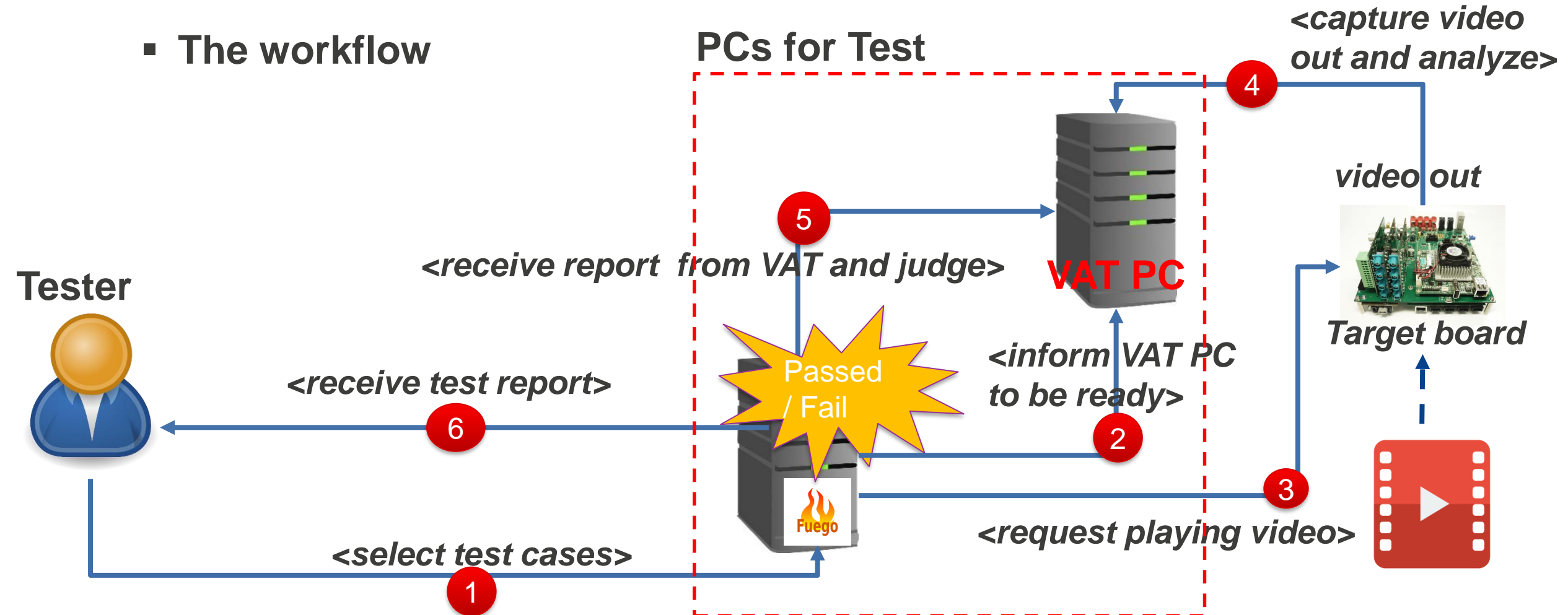
NOT depend on tester's feeling



INTRODUCTION ABOUT VIDEO AUTOMATION TEST

TRANSFORM IDEA TO SOLUTION

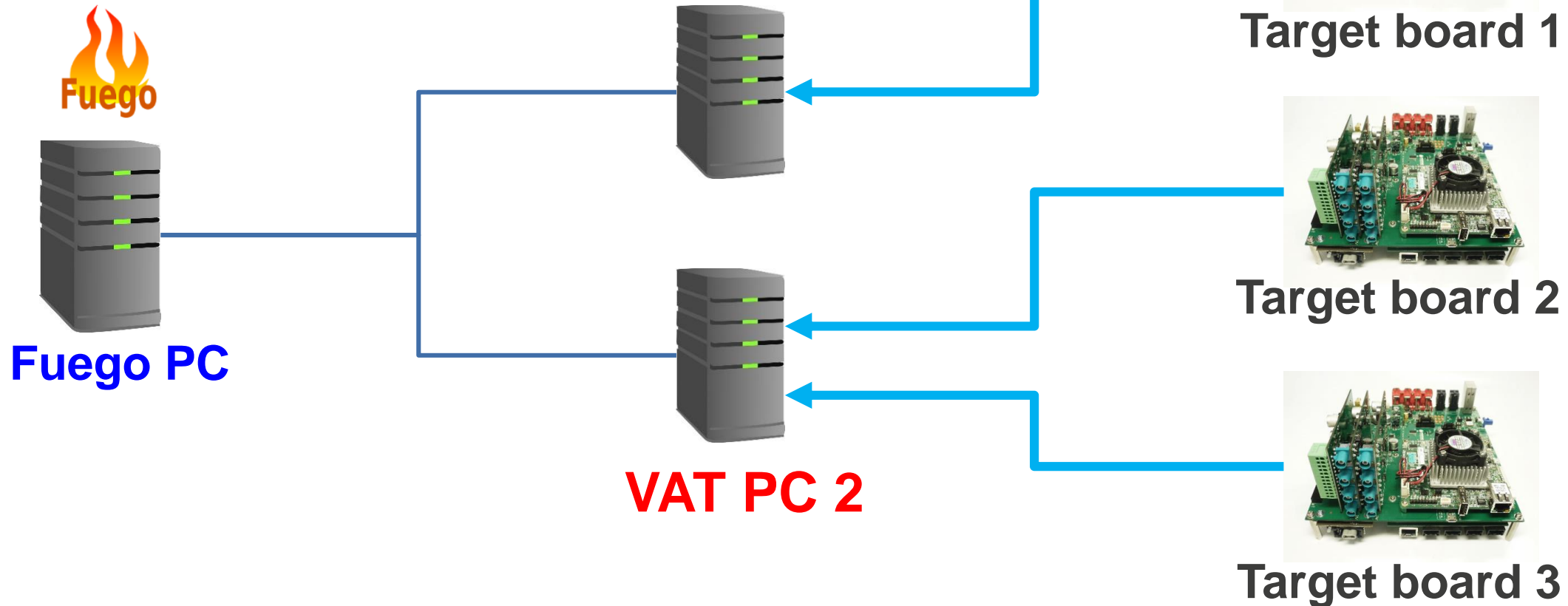
▪ The workflow



INTRODUCTION ABOUT VIDEO AUTOMATION TEST

TRANSFORM IDEA TO SOLUTION

- Devices connection



ACHIEVEMENT WITH VIDEO AUTOMATION TEST

VIDEO CODEC AND RESOLUTION

- Support Codecs:

MPEG2

MPEG-4



H.264
MPEG-4/AVC

H.265
HEVC
High Efficiency Video Coding

VP8

VP9

- Support video resolutions: 176x144, 352x288, 352x480, 352x576, 640x480, 720x480, 720x576, 1280×720, **1920x1080**, 2048×1080, ..., **3840x2160**

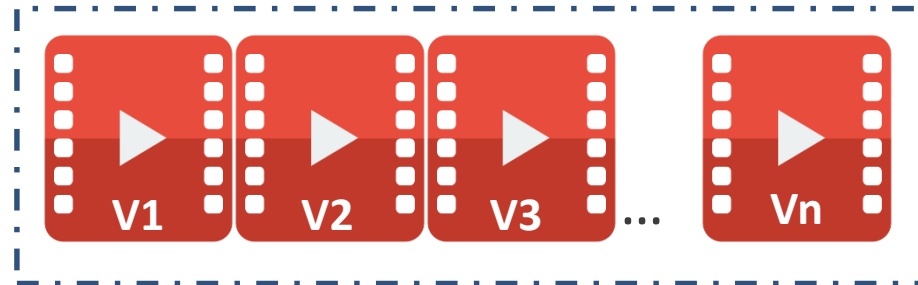
ACHIEVEMENT WITH VIDEO AUTOMATION TEST

USE-CASES

■ Supported video/movie playback use-cases:



Simple video/movie
playback



Sequence video/movie
playback



Loop input stream
and play

Repeat video/movie
playback

ACHIEVEMENT WITH VIDEO AUTOMATION TEST

VIDEO AUTOMATION TEST PERFORMANCE

■ With VAT PC specification

✓ OS: Ubuntu 14.04



✓ HW : Intel(R) Core(TM) i5-3570 CPU @ 3.40GHz, 8GB DDR3-1333 MHz RAM,
500GB HDD

✓ Video input signal: **HDMI 2.0**



✓ Performance: can adapt video playback on target board up to

FHD@60fps, UHD@30fps

ACHIEVEMENT WITH VIDEO AUTOMATION TEST

HUMAN TEST VS AUTOMATION TEST

Impact of Video Automation Test:

Compare the test run time:

❖ Manual Test:

- Video length (E.g.: **4 mins**)
- Run test + fill report : $\sim(4 + 2)$ mins

❖ Automation Test:

- Video length (4 mins)
- Auto Run test + auto report: $\sim(4 + 3)$ mins

Total: **~7 mins** (**1.16 times** comparing with manual test)

Compare the productivity:

❖ Manual Test:

- 1 person / 1 board / 1 day: **50 TCs**

❖ Automation Test:

- VAT PC / 1 board / **1 day (24-hours): ~150 TCs** (utilize overnight testing)

THE APPROACH FOR VIDEO AUTOMATION TEST



THE APPROACH FOR VIDEO AUTOMATION TEST

- Video automation test is developed to detect **issues** during video playback:
 1. Frame skip
 2. Frame delay
 3. Frame mismatch
 4. Video Playback not smooth

THE APPROACH FOR VIDEO AUTOMATION TEST

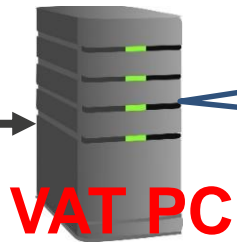
FRAME SKIP

- Definition: frame is **IN** (A) but **NOT IN** (B)

Target board



*video
out*



frames sequence in **video output** (B) :
f1 f2 f3 f4 f6 f9
Frames skip list: **f5 ,f7, f8, f10**



frames sequence in **test stream** (A) : f1 f2 f3 f4 **f5**
f6 **f7 f8** f9 **f10**

THE APPROACH FOR VIDEO AUTOMATION TEST

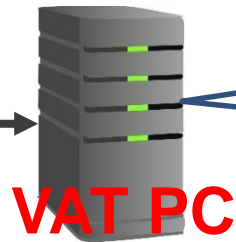
FRAME DELAY

- Definition: frame appears more than **one** time

Target board



*video
out*



frames sequence in **video output** (B) : f1
f2 **f2** f3 f4 f5 f6 f7 f8 **f8 f8** f9 f10
Frames delay list: **f2 ,f8, f8**



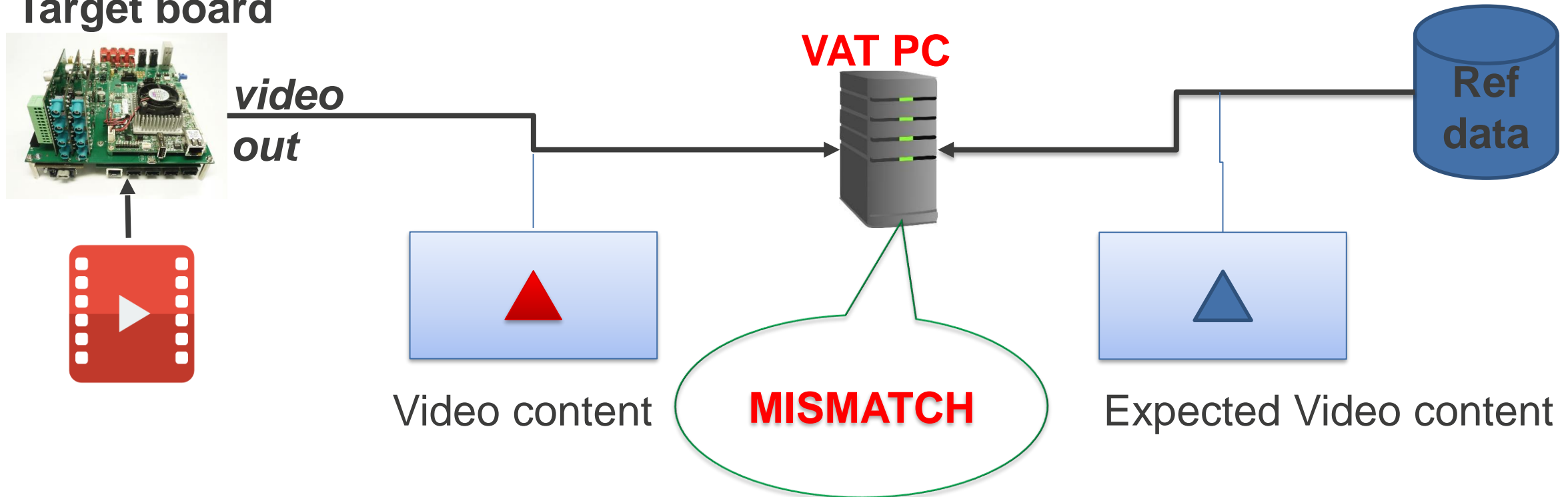
frames sequence in **test stream** (A) : f1 f2 f3 f4 **f5**
f6 **f7 f8** f9 **f10**

THE APPROACH FOR VIDEO AUTOMATION TEST

FRAME MISMATCH

- Definition: frame contains video out which is **different** from Ref data

Target board

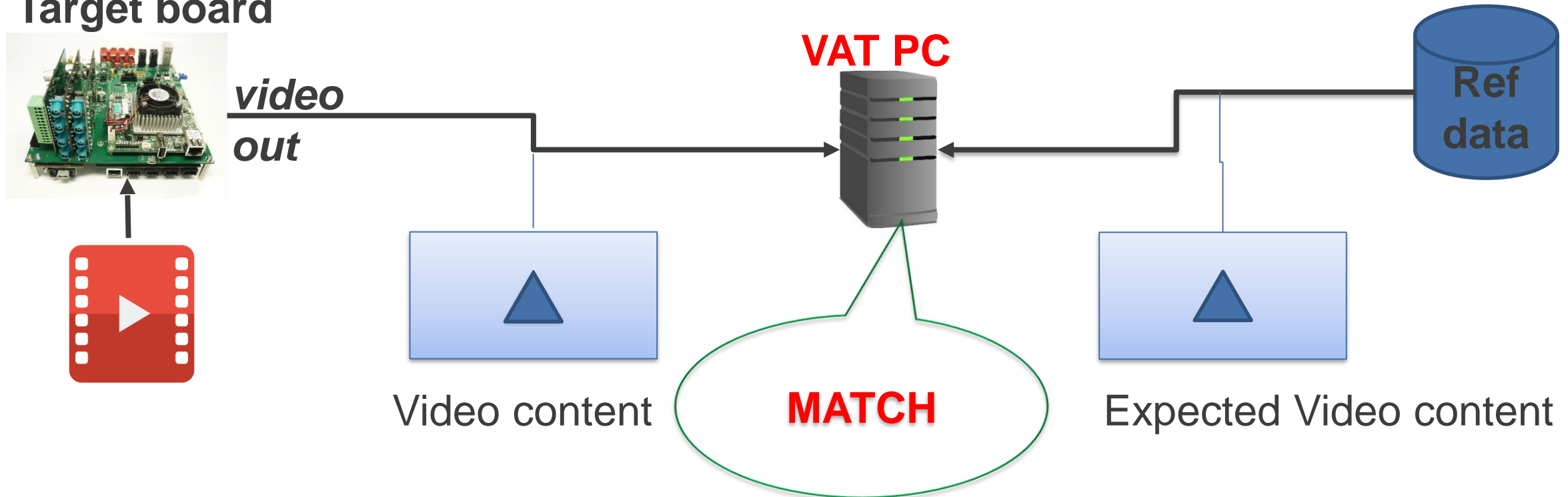


THE APPROACH FOR VIDEO AUTOMATION TEST

FRAME MISMATCH - CONT

- Definition: frame contains video out which is **different** from Ref data

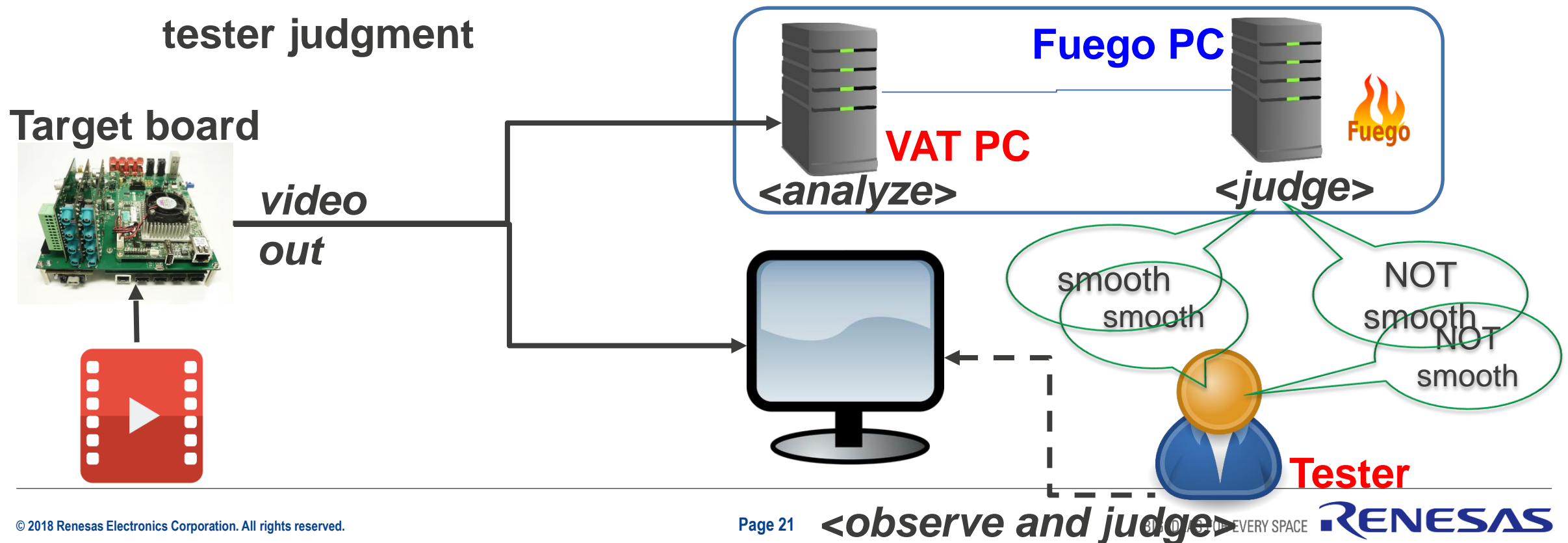
Target board



THE APPROACH FOR VIDEO AUTOMATION TEST



PLAYBACK NOT SMOOTH

- Definition: **lag** video playback
- Calibration: Video playback **judgment** feature is **calibrated** along with tester judgment



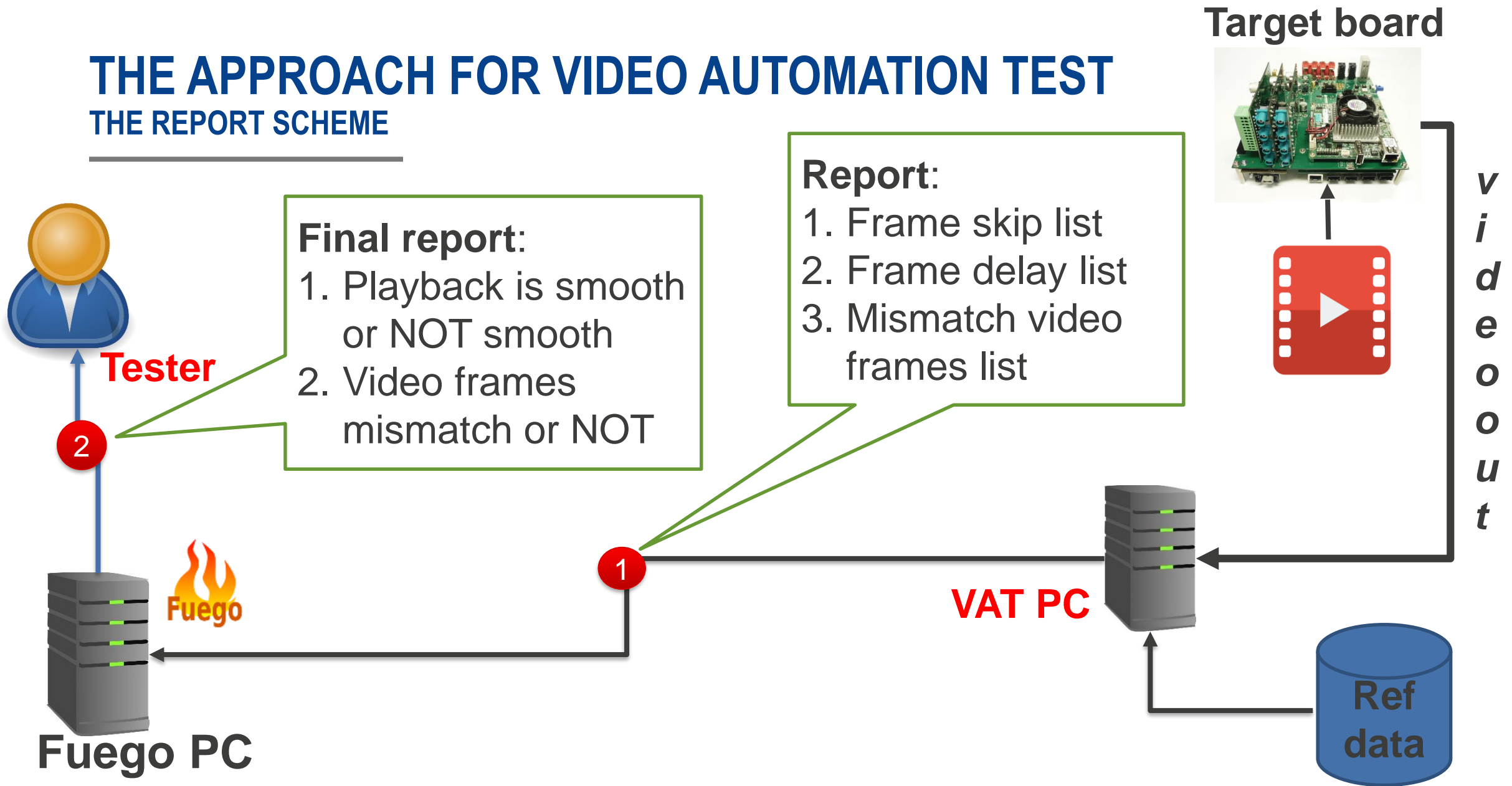
THE APPROACH FOR VIDEO AUTOMATION TEST

TOOLS SUPPORT VIDEO AUTOMATION TEST DEVELOPMENT

No	Item	Tools
1	Frame skip	  FFmpeg
2	Frame delay	
3	Frame mismatch	
4	Playback not smooth	Self development algorithm

THE APPROACH FOR VIDEO AUTOMATION TEST

THE REPORT SCHEME



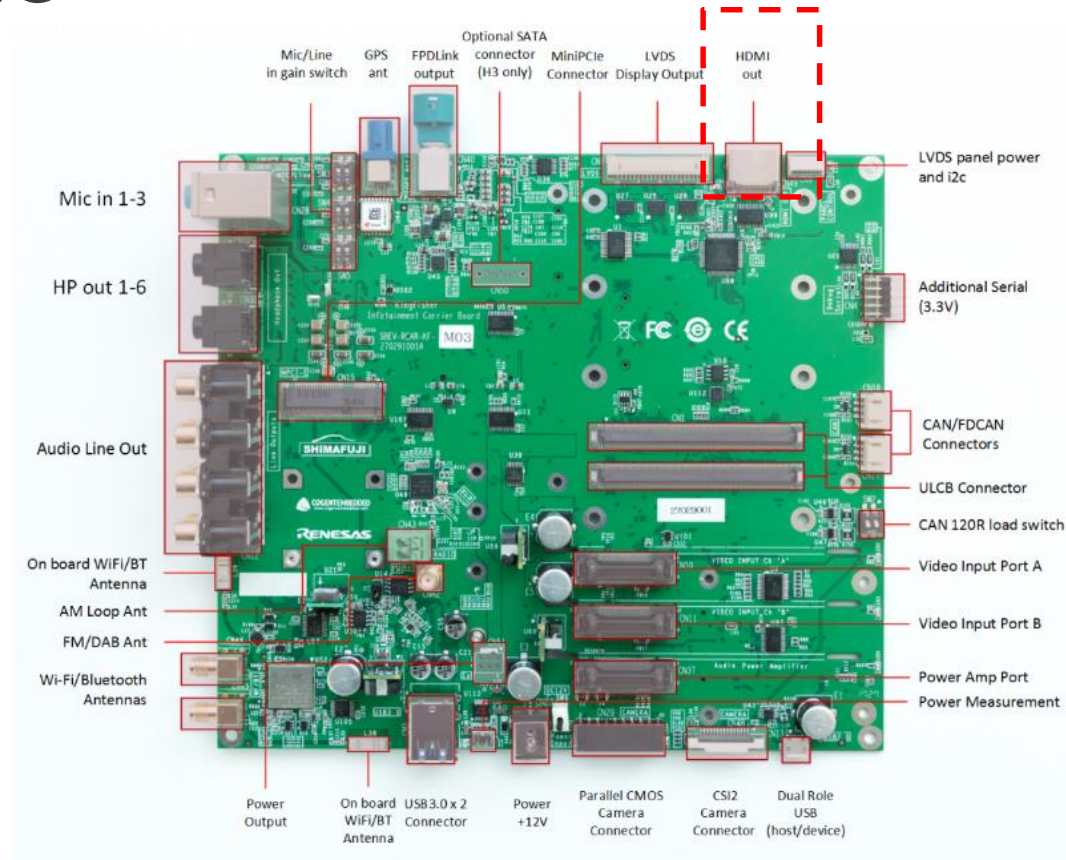
A detailed, high-contrast black and white photograph of an internal combustion engine. The image shows various mechanical components like the timing belt, pulleys, and valve train. A solid blue horizontal bar is superimposed over the upper portion of the image, containing the title text in white.

HOW TO APPLY VIDEO AUTOMATION TEST

HOW TO APPLY VIDEO AUTOMATION TEST TARGET BOARD

- DO NOTHING

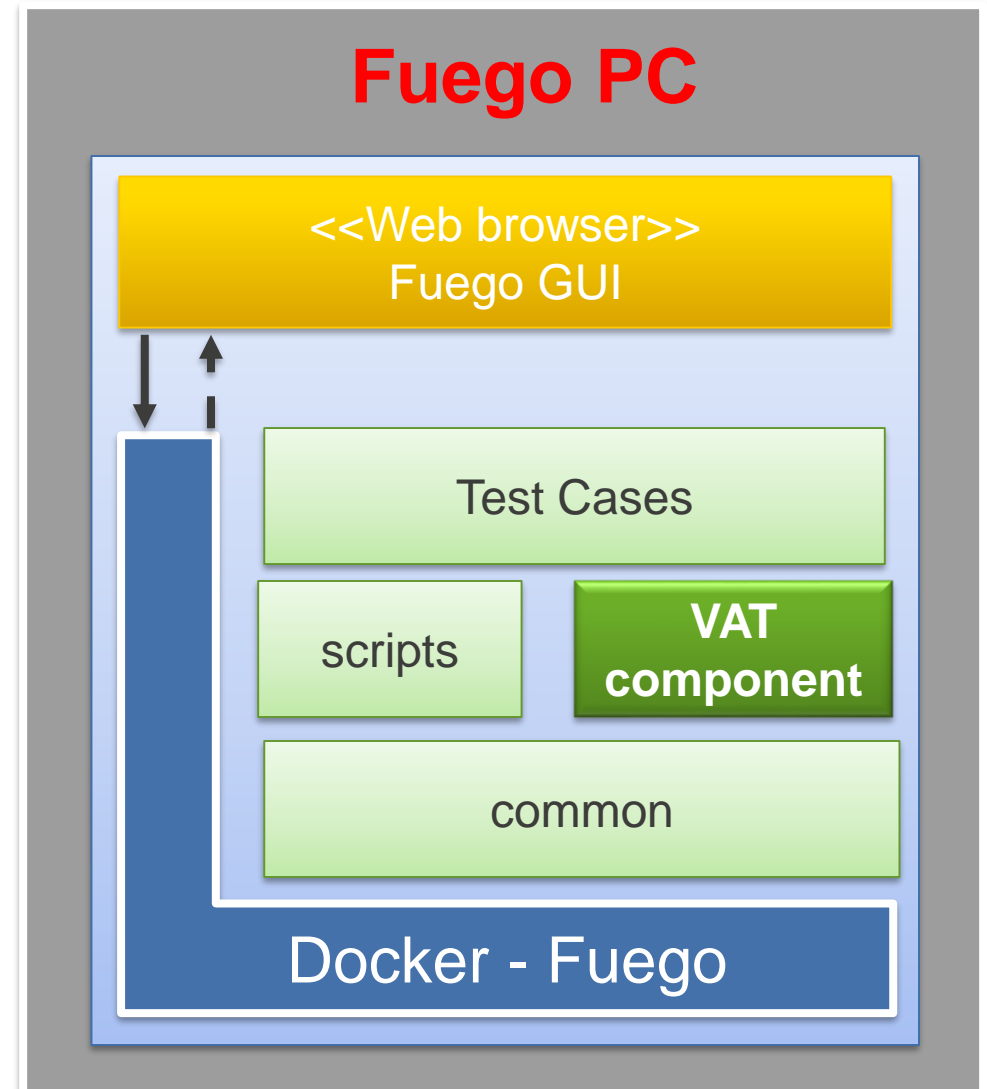
Video Out



HOW TO APPLY VIDEO AUTOMATION TEST

FUEGO PC

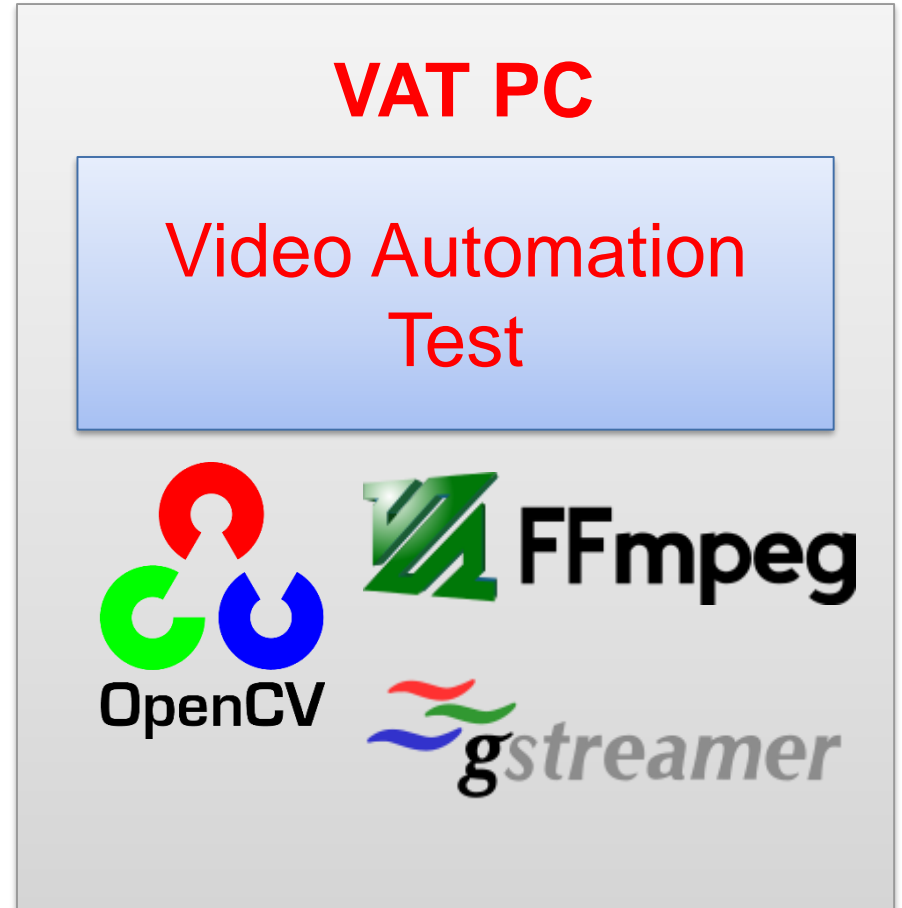
- Install Fuego on Ubuntu PC
- Install the VAT component to communicate Video Automation Test from Fuego side



HOW TO APPLY VIDEO AUTOMATION TEST

VAT PC

- Install the Video Automation Test on VAT PC
- Install Open Source Software: OpenCV, Ffmpeg, Gstreamer



THE FUTURE PLAN

A blurred photograph of a road winding through a forest, with a blue banner in the top left corner containing the text 'THE FUTURE PLAN'.

CURRENT VIDEO AUTOMATION TEST STATUS

STRONG POINTS

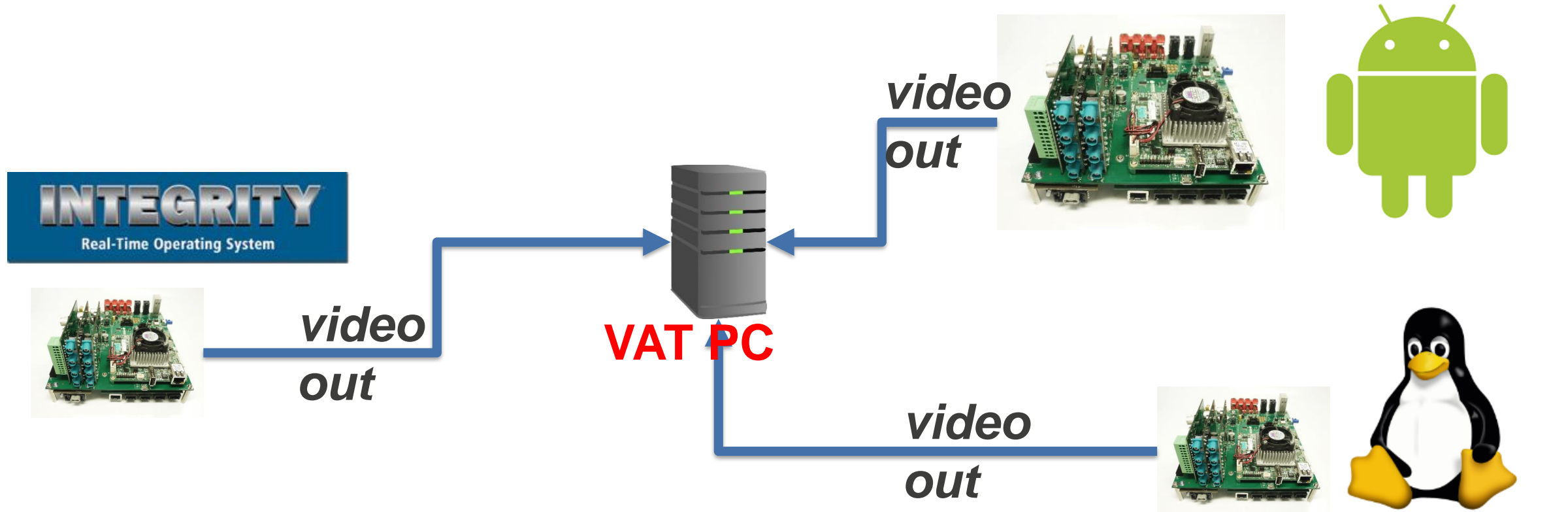
1. Thank to the Fuego - Automated test framework for the LTSI project.
2. Thank to the Open Source Software (E.g. ffmpeg, ffprope, OpenCV, ...)



CURRENT VIDEO AUTOMATION TEST STATUS

STRONG POINTS - CONT

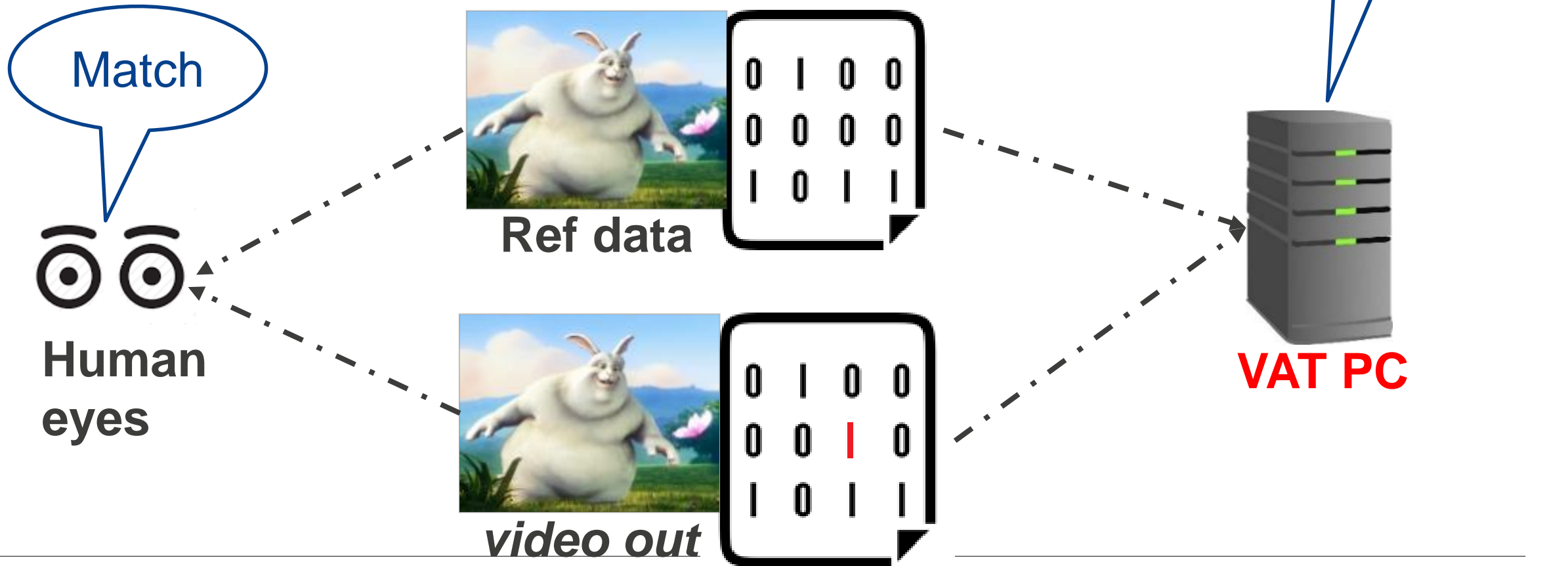
3. Independent on OS of target board



CURRENT VIDEO AUTOMATION TEST STATUS

NEED IMPROVEMENT POINTS

1. So strictly compare since **NO threshold** in comparison



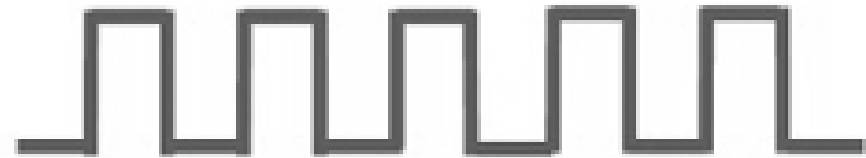
CURRENT VIDEO AUTOMATION TEST STATUS

NEED IMPROVEMENT POINTS

2. Video signal



Analog Signal



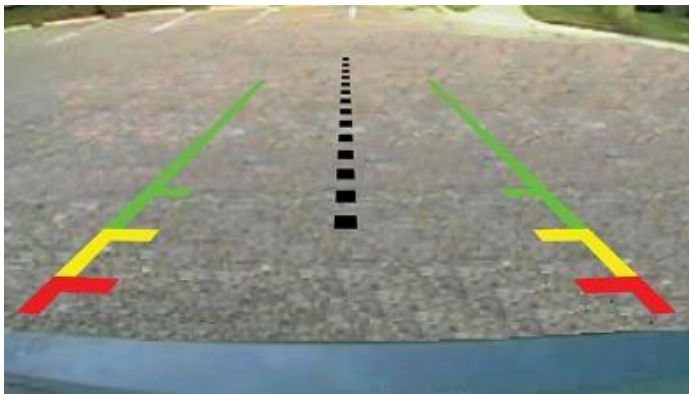
Digital Signal



CURRENT VIDEO AUTOMATION TEST STATUS

NEED IMPROVEMENT POINTS

3. Automation test video/movie playback only.



➔
**video
in**



➔
**video
out**



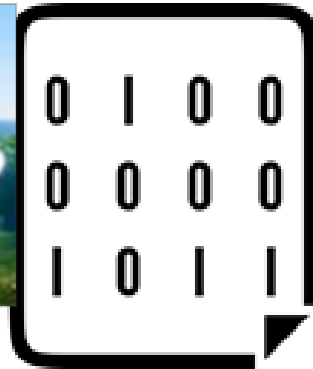
THE FUTURE PLAN

ACTION ITEMS

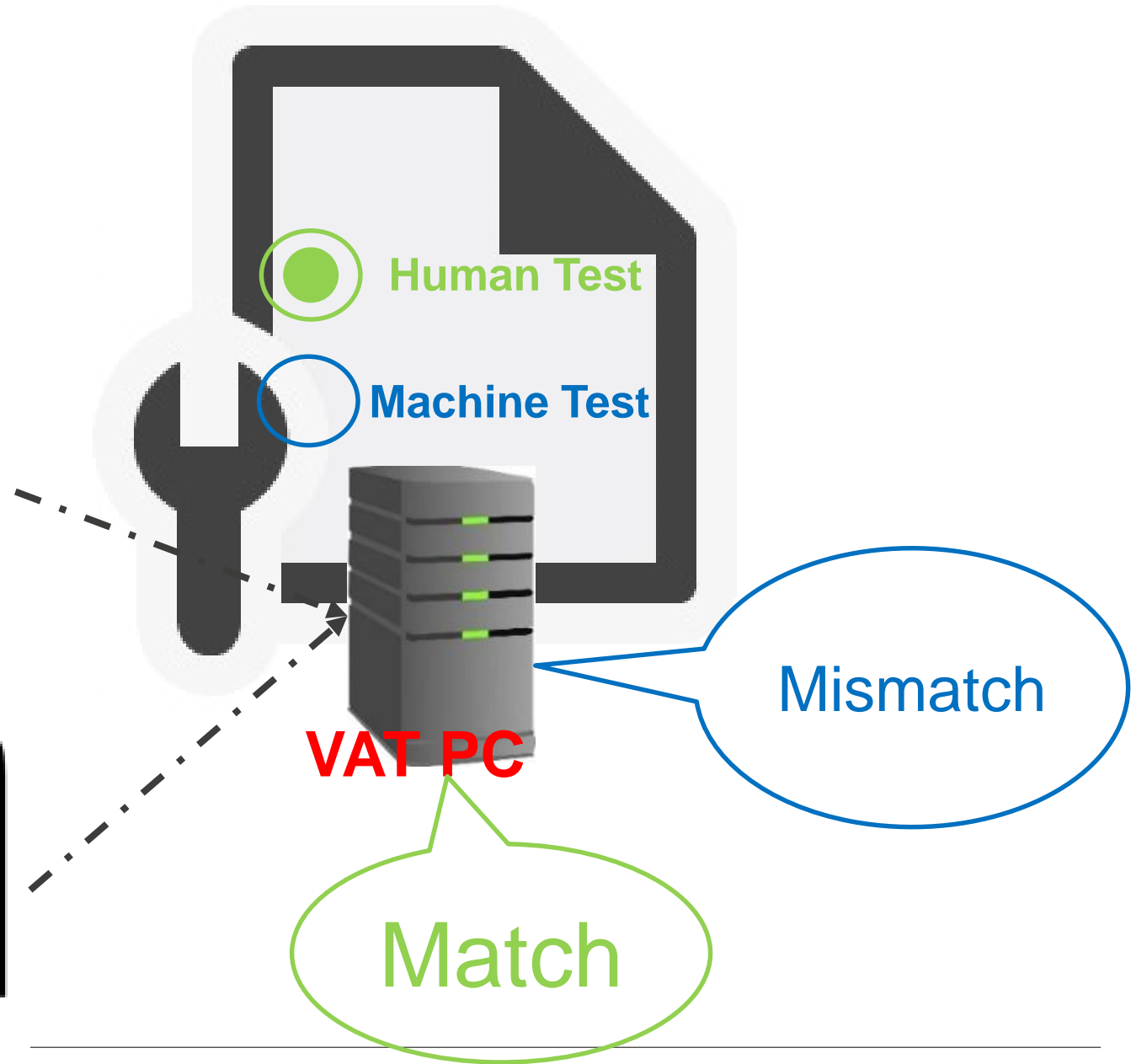
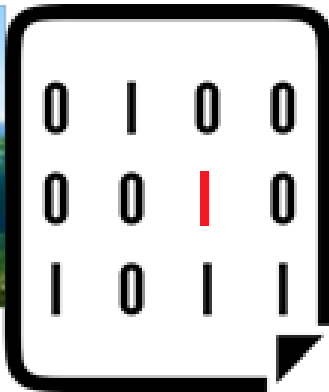
1. Support threshold



Ref data



video out



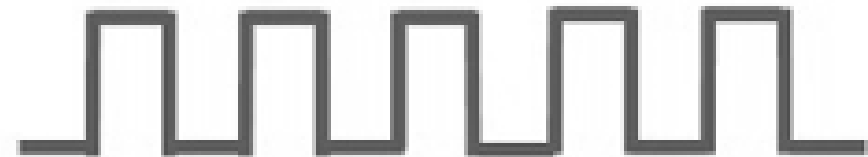
THE FUTURE PLAN

ACTION ITEMS

2. Support more Video signal types: Digital, **Analog**



Analog Signal



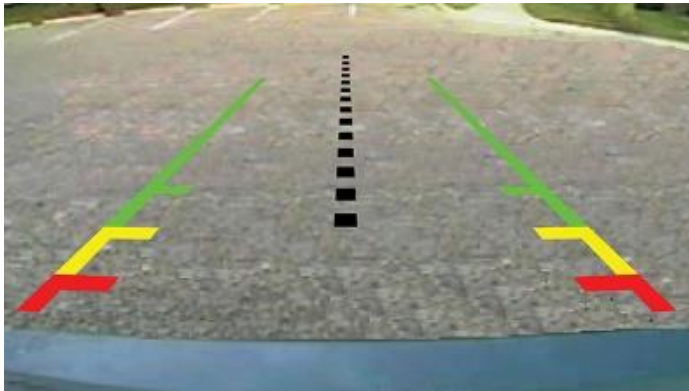
Digital Signal



THE FUTURE PLAN

ACTION ITEMS

3. Support Automation test video/movie playback and Record Video



➡
**video
in**



➡
**video
out**



THE FUTURE PLAN

ACTION ITEMS

- Summary improvement plan

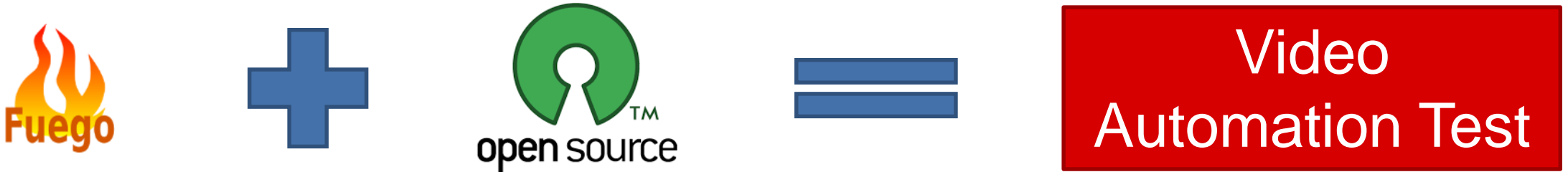
No	Item	Current Support	Future Support
1	Comparison method	Threshold: NO	Threshold : YES
2	Video signal type	Digital : YES Analog : NO	Digital : YES Analog : YES
3	Use-cases	Video playback: YES Record Video: NO	Video playback: YES Record Video: YES

CONCLUSION



CONCLUSION

- By combination between the Fuego and the Open Source Software, Renesas could make a Video Automation Test for Linux platform.



- Thanks to the Video automation test, could **reduce the testing workload** and get the **reliable results**
- Eagerly, processing the **future plan**

THE END
THANK YOU VERY MUCH!



Q&A

[Renesas.com](https://www.renesas.com)

APENDIX-FUEGO

❖ What is Fuego?

- Fuego is **a test framework** specifically designed for **embedded Linux testing**. It supports automated testing of embedded targets from a host system, as it's primary method of test execution.
- Fuego consists of a host/target script engine, with a Jenkins front-end, and over 50 pre-packaged tests, installed in a Docker container.
- Tim Bird gave a talk introducing Fuego, at *Embedded Linux Conference in April 2016*, and *LinuxCon Japan 2016*

Fuego = (Jenkins + abstraction scripts + pre-packed tests) inside a container

