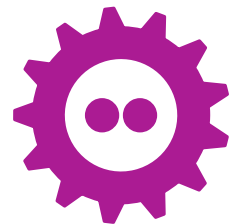


# IOT BZH

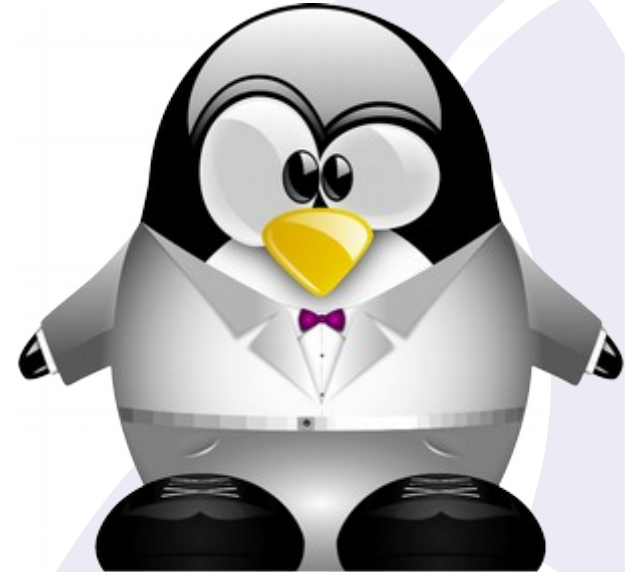


**Continuous testing in a cloud based infrastructure using  
virtualization and real hardware in the loop**  
*Armand Bénéteau – FOSDEM – February 5th 2022*



# Agenda

- Introduction and context
- Q.A. goals
- RTM introduction and overview
- RTM in the Community stack
- Real RTM challenges
- LAVA integration in real RTM
- Board integration in real RTM
- Test reporting in Q.A. system
- Roadmap
- Conclusion



# About me

- Software and QA engineer @ **IoT.bzh**
- Formerly:
  - Master degrees in IT/electronic engineering from:
    - INSA (Rennes, France)
    - Strathclyde University (Glasgow, Scotland)
  - Low Power IOT (LoRaWAN technology)
    - Wireless sensors firmware (C)
    - LoRaWAN protocol implementation (C)
    - LoRaWAN gateway, running on Linux (C)
- [armand.beneteau@iot.bzh](mailto:armand.beneteau@iot.bzh)
- <https://www.linkedin.com/in/armand-beneteau/>





# IoT.bzh at a glance

## Our location

Brittany



European CyberSecurity  
Organisation Cyber  
Valleys mapping

## Our 30-year OS background

Wind River (1990) - Intel (2009) - IoT.bzh (2015)

VxWorks®



WIND RIVER

Real Time OS leader

TIZEN

SAMSUNG

n°1 OS in TV market  
Made by Intel in Brittany



## Our team

~30 engineers



## Our product



**redpesk®** is a pre-integrated « ready-to-use » CI/CD SW factory  
generating a custom & secured OS, with Long Term Support



C.T. in cloud using virtualization and real hardware

FOSDEM - 05-02-2022

# Introduction and Context

- **Code complexity** is increasing in systems

*Boeing 787 (2004): 14 million lines of code<sup>[1]</sup>*

*Average high-end car (2012): 100 million lines of code<sup>[2]</sup>*

- **L.T.S.** is mandatory for industrial systems

*Average age of european car: 12 years old (source: ACEA)*

*Approximately the same in the U.S. (source: IHS Market)*

- **Cybersecurity** is not an option anymore

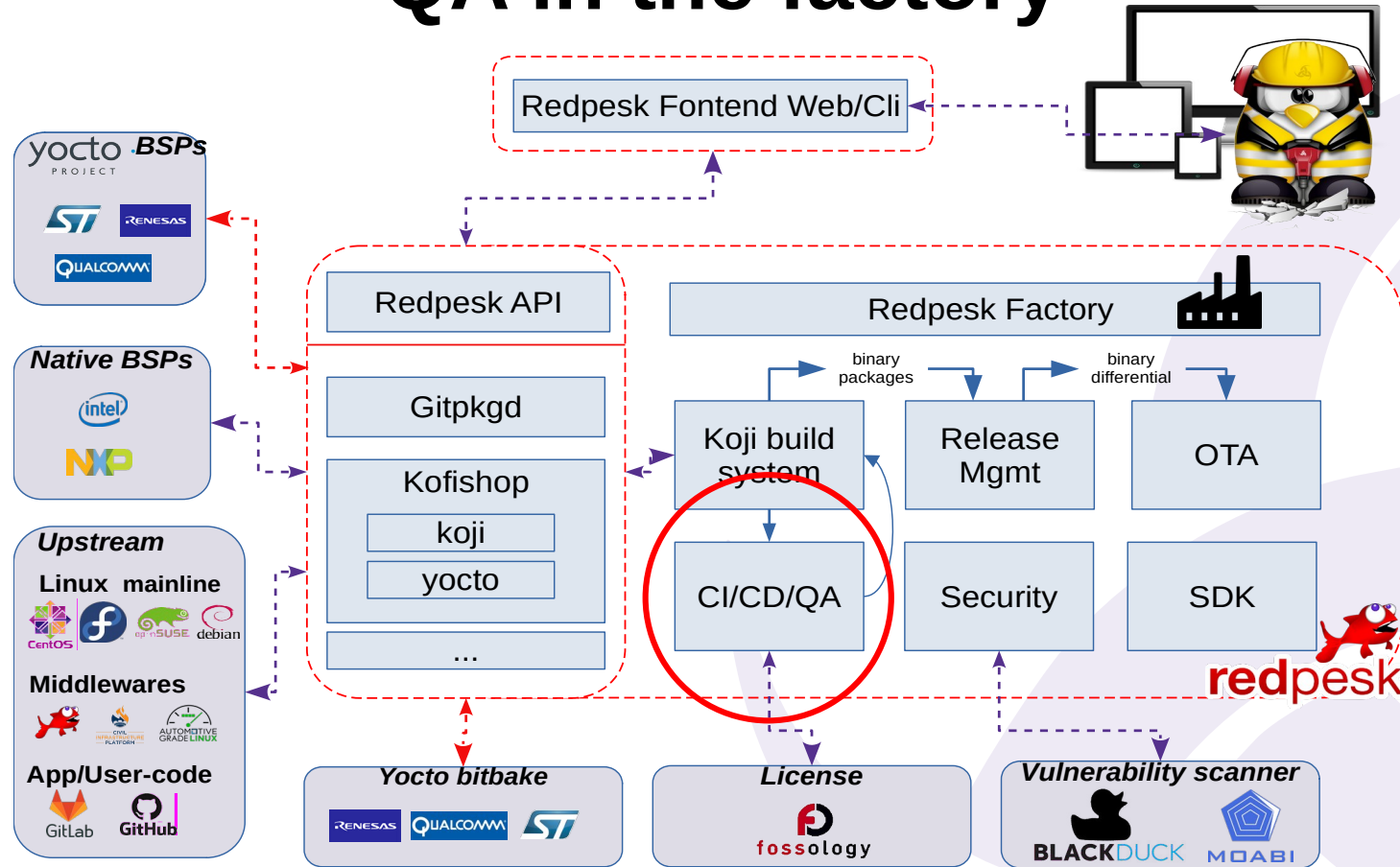
*In the first 6 months of 2021: > 1.5 billion attacks on IoT devices (source: Kaspersky)*

- **Conclusion: Automatic CI/CT/QA infrastructure is a must have**

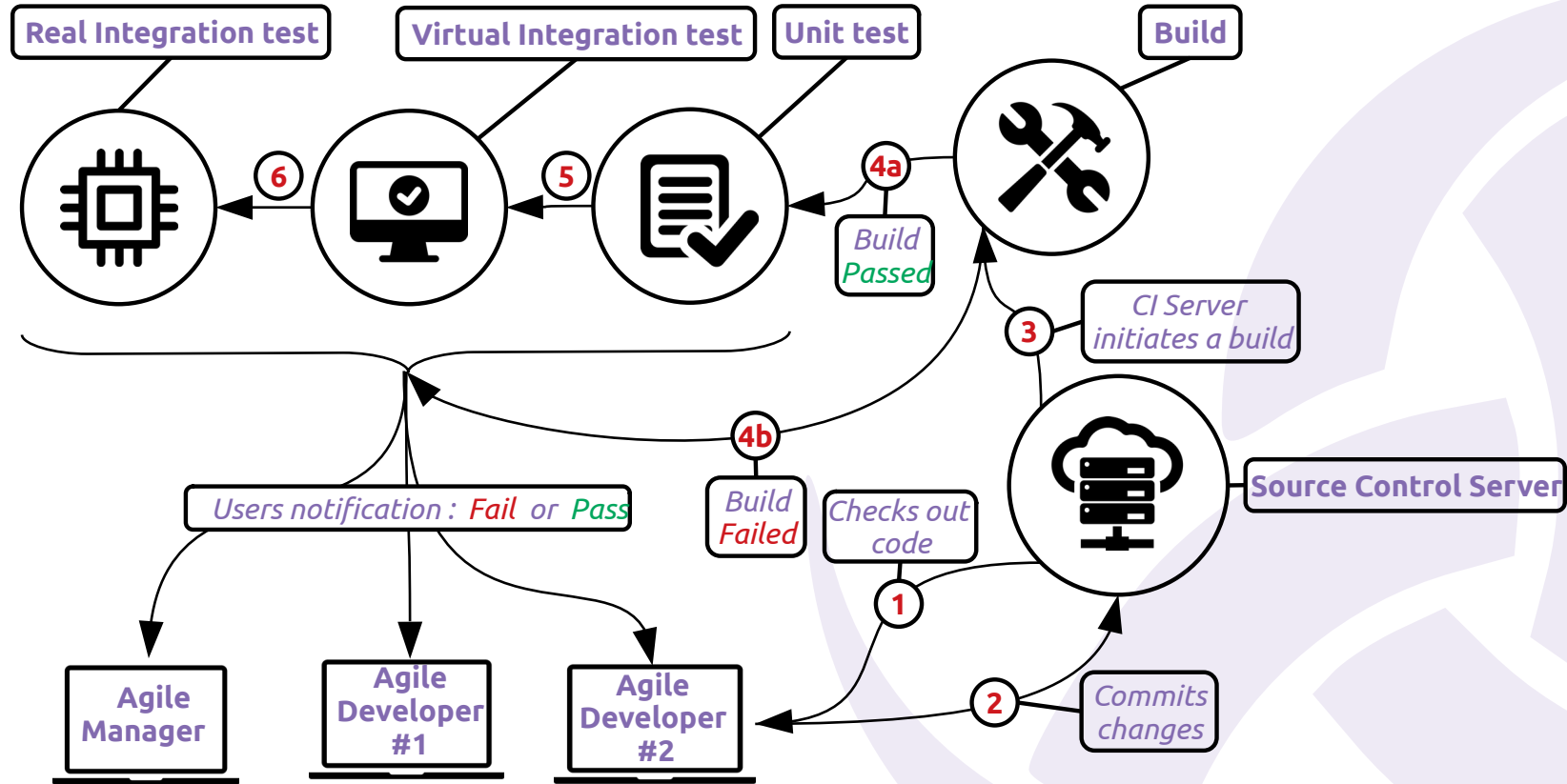
[1] <https://www.nycaviation.com/2011/09/fun-facts-revealed-at-boeings-787-technical-panel>

[2] <https://www.wired.com/2012/12/automotive-os-war/>

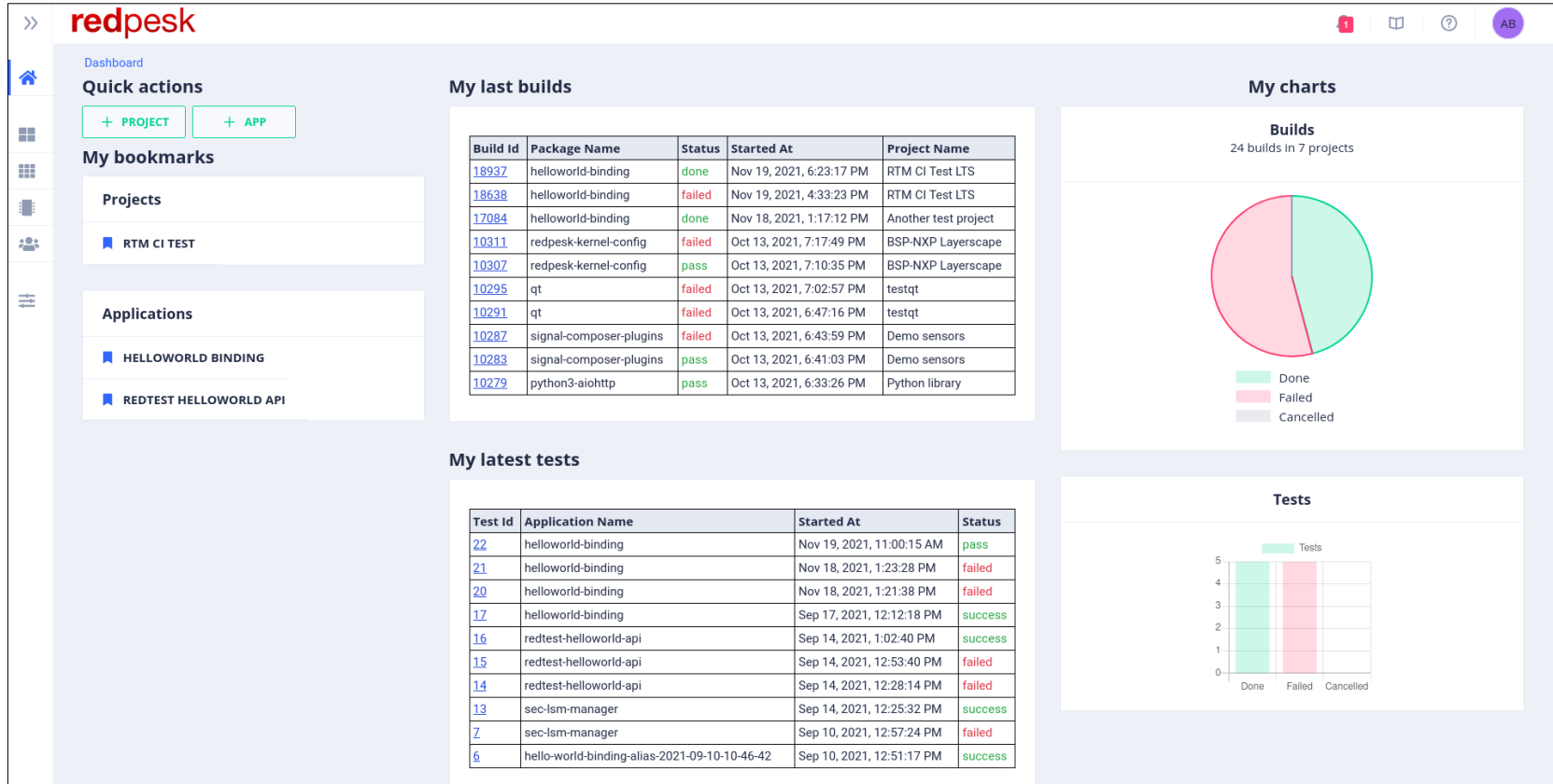
# QA in the factory



# Q.A. goals: workflow



# Q.A. goals: reporting





# RTM introduction

- **Integration tests** run in our **R.T.M.: Rackable Test Modules**

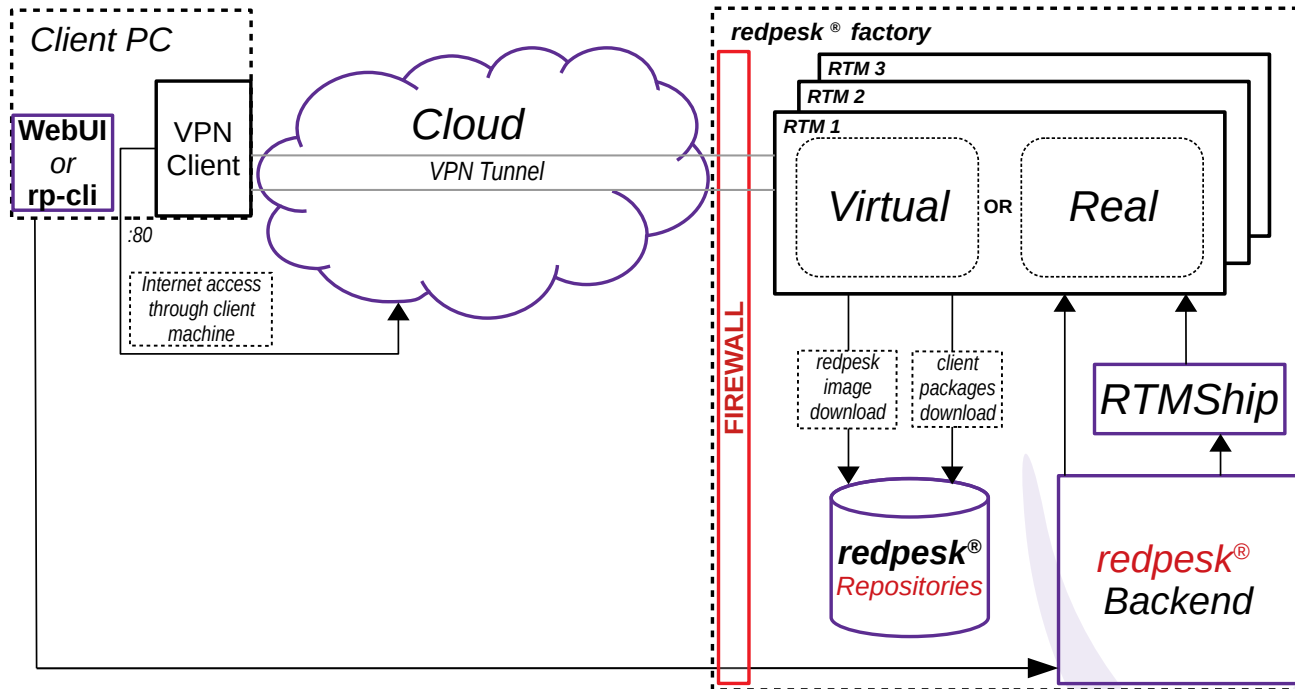
*RTM are our solution to run integration tests within the redpesk infrastructure.*

*They are intended to meet user requirements regarding qualifications, certifications, continuous integration, etc.*

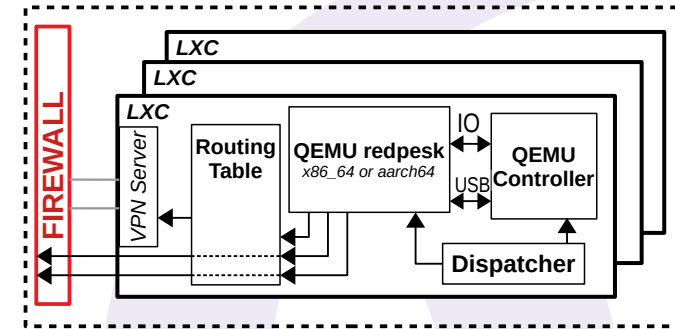
*They can be dynamically started by the infrastructure or by the developer to run integration tests.*

*They are the heart of the continuous integration and continuous testing inside redpesk factory.*

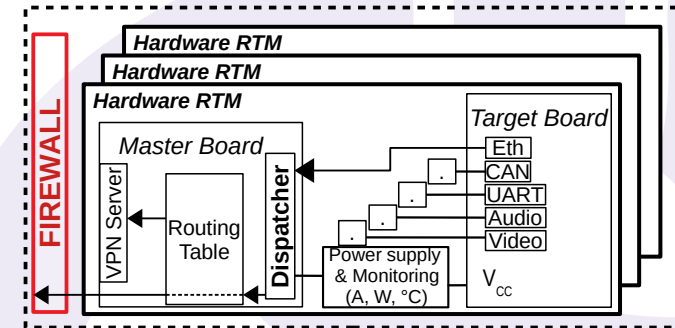
# RTM overview: infrastructure overall view



## Virtual

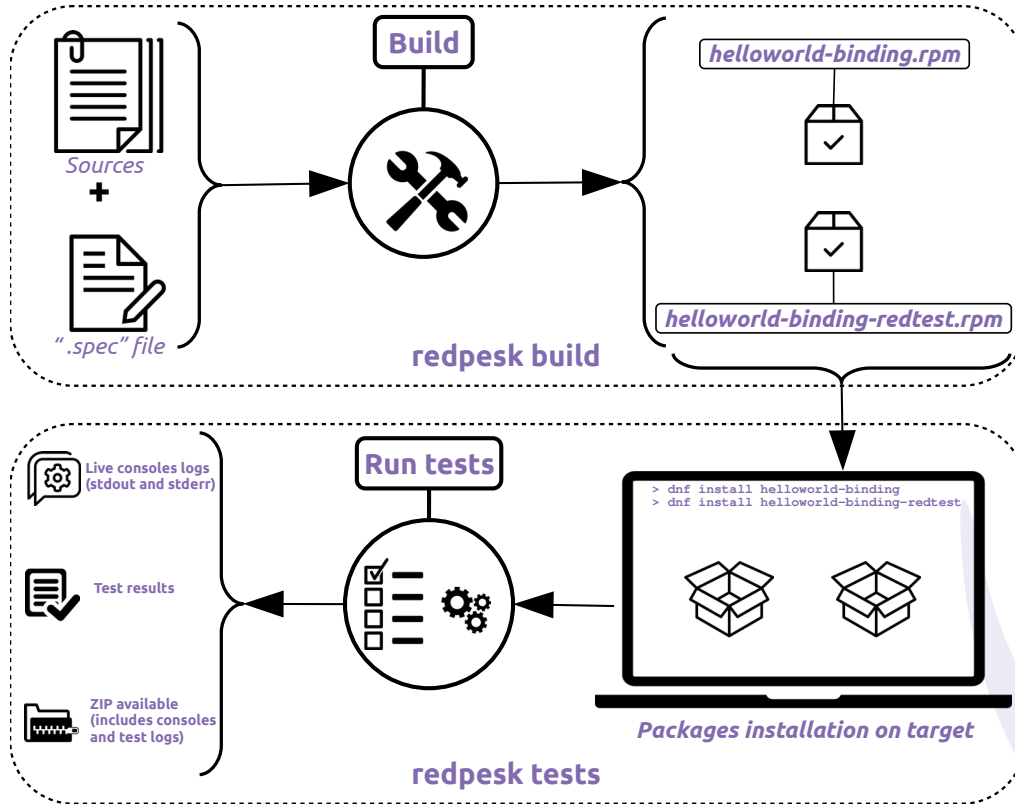


## Real



# Developer testing process in Community

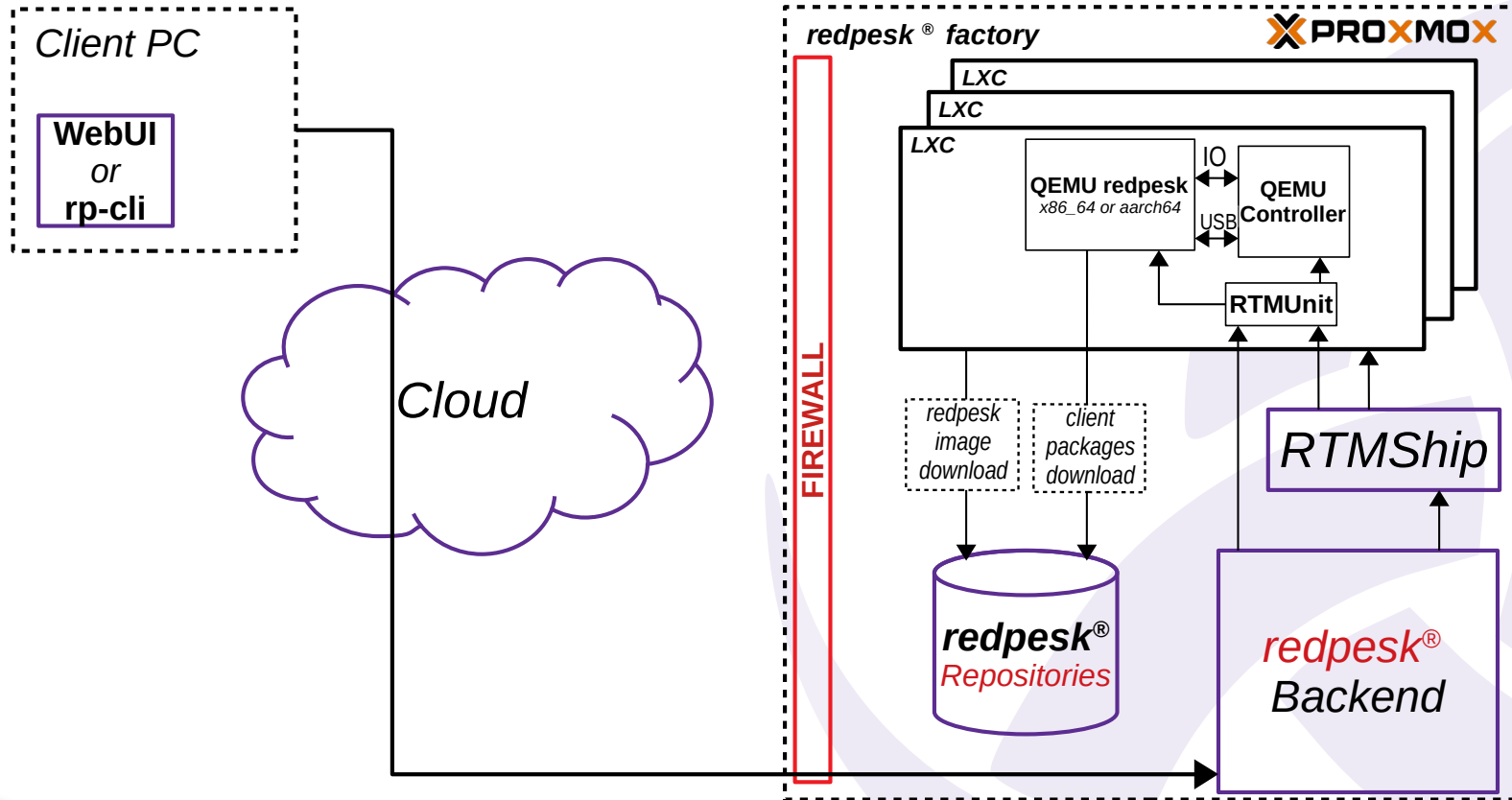
The icons used in this diagram are made by Freepik from [www.flaticon.com](http://www.flaticon.com)



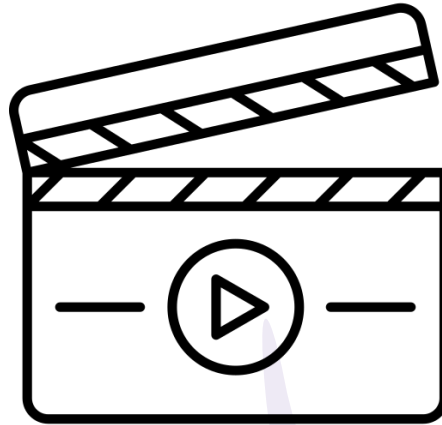
- Definition of “**redtests**”<sup>[1]</sup>:
  - Allows one to have a “standard” for tests in CI
- Really simple:
  - A “**redtest**” subpackage
  - ‘**run-redtest**’ script in this package
  - Test logs need to respect “**TAP**” format<sup>[2]</sup>

[1] [https://docs.redpesk.bzh/docs/en/master/redpesk-factory/docs/2\\_integration-tests.html](https://docs.redpesk.bzh/docs/en/master/redpesk-factory/docs/2_integration-tests.html)  
[2] <https://testanything.org/tap-specification.html>

# RTM infrastructure in Community



# Demo in redpesk “Community”





# Real RTM challenges

## ➤ **Sharing boards** between users

- *We do not have one board per user (price, availability)*
- *The real RTM system needs to manage the users access*

## ➤ **Management of board's power supply**

- In order to start and stop the boards correctly

## ➤ **Management of redpesk OS image loading**

- We need to load a full redpesk image (several partitions and size > 2G)

## ➤ **Management of board's boot**

- Grub, uboot, prompt, etc.

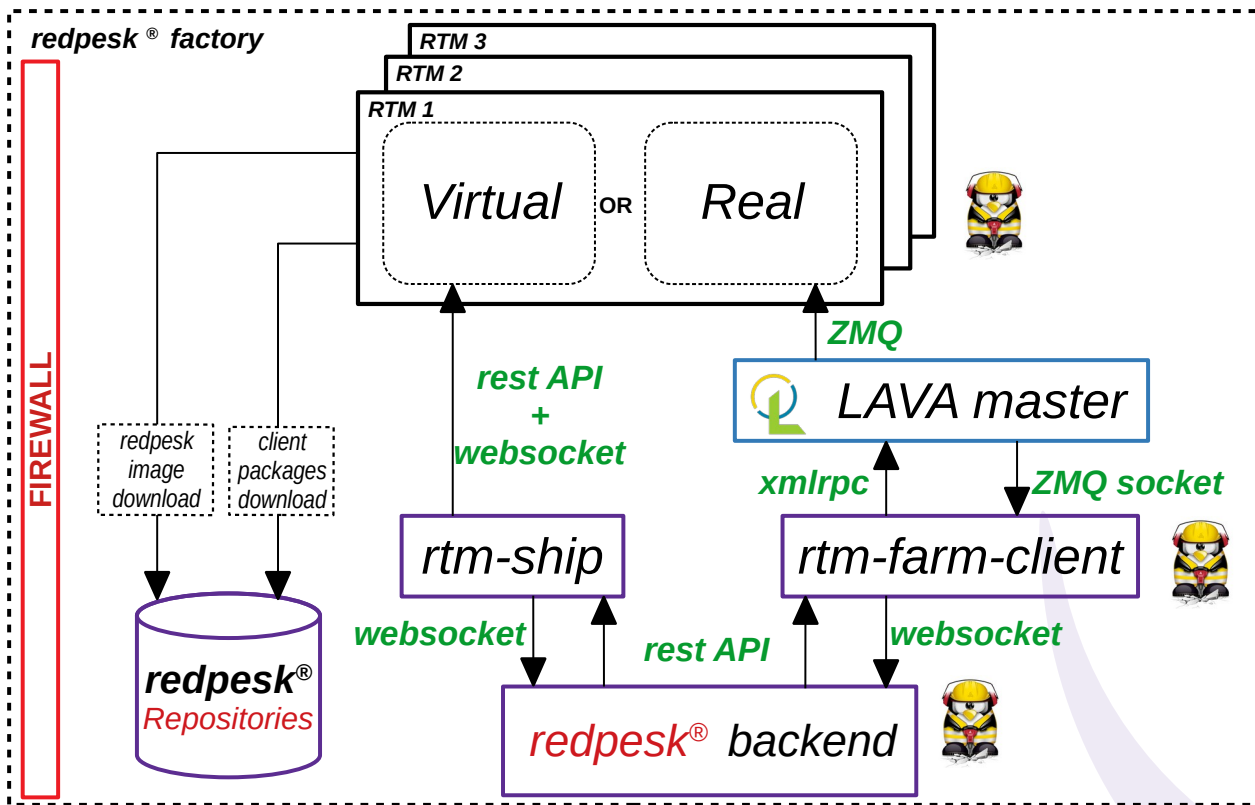
# Real RTM: LAVA, the “missing link”

- **LAVA** (*Linaro Automated Validation Architecture*)<sup>[1]</sup>:
  - **Continuous Integration system** for deploying OS onto *physical and virtual hardware* for running tests
  - *Used a lot in Kernel validation* (e.g.: in KernelCI<sup>[2]</sup>)
  - **Fully open-source**<sup>[3]</sup>
  - *Already existing board definitions*
  - **Uboot, grub and fastboot** management



[1] <https://www.lavasoftware.org/>  
[2] <https://kernelci.org/>  
[3] <https://git.lavasoftware.org/lava>

# Real RTM integration in redpesk CI



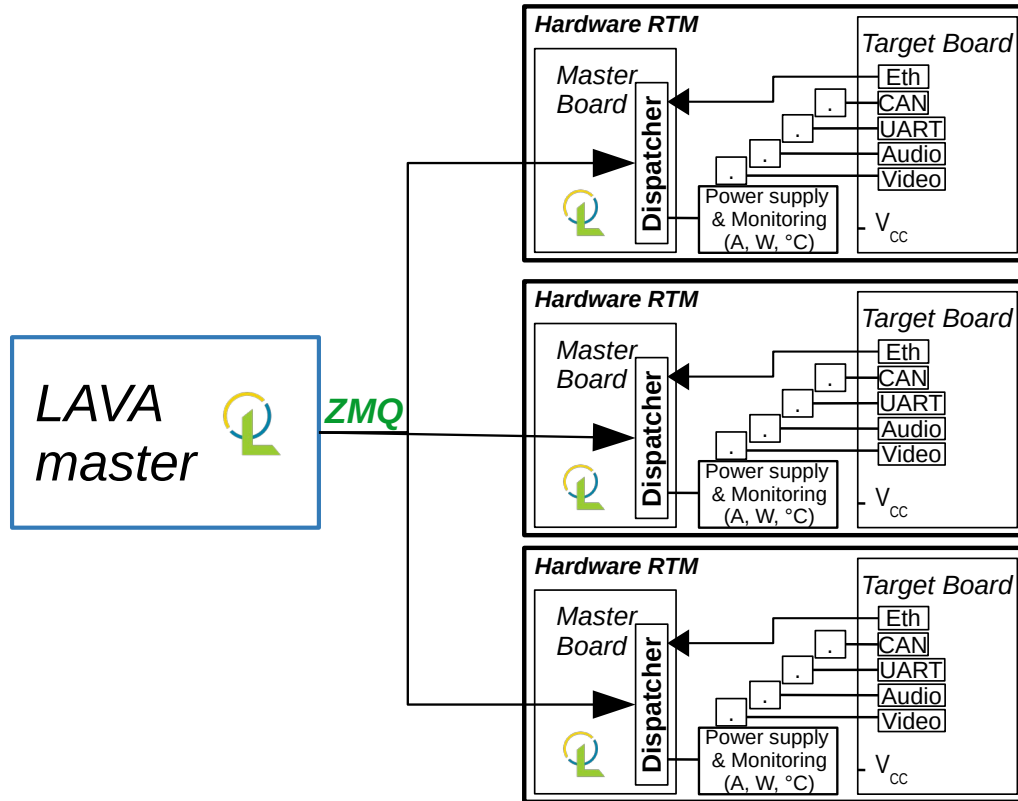
## ➤ **rtm-farm-client**

- New async  $\mu$ service
- Xmlrpc and zmq to communicate with LAVA master
- Allows homogeneous communication with backend (rest API + ws)

## ➤ Some refactoring in **backend** needed

## ➤ Integration of “**redpesk boards**” in LAVA environment

# Real RTM architecture



- **1 dispatcher per DUT:** to be able to run more complex tests
  - *HDMI grab*
  - *Audio grab*
  - *Etc...*
- Job submissions, test **results**, test **logs** fully managed by LAVA
- **LAVA master – dispatcher** communication is LAVA business not redpesk

# Powering a real RTM

## ➤ *How to control the power supply?*

- First easy solution: a remotely controlled multi-socket
  - *Energenie EG-PMS2-LAN*
  - Controlled through ethernet
  - Binary available in Linux distributions<sup>[1]</sup>: `/usr/bin/egctl`
- Second, more complete, solution: laboratory power supply
  - *Joy-it JT-DPM8605*
  - Controlled through RS485 (ModBus) protocol
  - Simulate “**low**” or “**high**” voltage situations to test boards in their limits



[1] <https://manpages.ubuntu.com/manpages/focal/man1/egctl.1.html>



# Real RTM: network boot (with PXE)

- *First option: NFS filesystem*
  - Not usable for us because **NFS does not propagate the SELinux or Smack labels...**
- *Second option: NBD (Network Block Devices) filesystem*
  - **Works well** (really fast on a good network)
  - No need to flash the redpesk image locally → **does not wear the board memory out (SD card, eMMC)**
  - But:
    - Slower or faster → **not exactly the same behaviour** as local boot

# Real RTM: boot with fastboot

## ➤ ***Fastboot** (coming from Android)*

- The DUT behaves as **USB storage** where the image can be **flashed**
- To enter in “fastboot” mode, “U-boot” needs to be stopped
- **Supported by LAVA**
- Allows us to integrate 3 boards out of 4:
  - *Solidrun Solidsense*
  - *Renesas-Gen3*
  - *Raspberry Pi 4*

```
Hit any key to stop autoboot: 2
end: 1.3.2 bootloader-interrupt (duration 00:00:02) [common]
start: 1.3.3 connect-lxc (timeout 00:39:54) [common]
No LXC device requested
end: 1.3.3 connect-lxc (duration 00:00:00) [common]
Setting prompt string to ''
Changing prompt to
uboot-enter-fastboot: Wait for prompt (timeout 00:39:55)
fastboot 0
fastboot 0
end: 1.3 uboot-enter-fastboot (duration 00:00:05) [common]
start: 1.4 download-retry (timeout 00:39:54) [common]
start: 1.4.1 http-download (timeout 00:39:54) [common]
downloading http://download.lorient.iot/redpesk-nightly/redpesk-devel_arz/smack/release/minimal/aarch64/renesas-gen3/2021-11-14/redpesk-devel-arz-
smack-minimal-aarch64-renesas-gen3-2021-11-14.raw.xz
saving as /var/lib/lava/dispatcher/tmp/1370/fastboot-deploy-nhxiqvsg/all/redpesk-devel-arz-smack-minimal-aarch64-renesas-gen3-2021-11-14.raw
total size: 723683728 (690MB)
Using unxz to decompress xz
```

# Real RTM: boot on USB

- *If fastboot not supported: **USB gadget***
  - USB gadget enabled in dispatcher (needs USB OTG)
  - Dispatcher behaves as **USB storage** → **DUT boot on USB** in this case
  - **No need to flash** the boards eMMC → **does not wear the board memory out (SD card, eMMC)**
  - In the future, it will allow us to **simulate USB devices**
    - *Mouse*
    - *Keyboards*
    - *Etc.*
  - Allows us to integrate the last board:
    - *Intel Up-Board*

# Test reporting

- *Once integration tests have passed or failed, **reporting needs to be done on the results***
  - *Boot logs (if revelant)*
  - *stdout/stderr outputs during test runs*
  - *“.tap” file containing the test results*
- *If the tests are **successful** (virtual and real), the package can go to the **next step***
  - *Vulnerability scanner*
  - *Licence analysis*
  - *Etc.*

# Test reporting in Community

## C.I. Test

### General Info

Test ID	26
Application Name	helloworld-binding
Status	success
Created at	Nov 23, 2021, 11:14:27 AM
Ended at	Nov 23, 2021, 11:15:35 AM
Board model	
Error message	
Total tests run	10
Succeeded tests	9
Failed tests	0
Skipped tests	1
Log file	<a href="#">tests_logs.zip</a>
Tap file	<a href="#">helloworld-binding_all_tests.tap</a>

### Test result file: helloworld-binding\_all\_tests.tap

```
1 # Filename: helloworld.tap
2 1..5
3 # Started on Tue Nov 23 11:15:29 2021
4 # Starting class: testPingSuccess
5 ~~~~ Begin testPingSuccess ~~~~
6 ~~~~ End testPingSuccess ~~~~
7 ok 1 testPingSuccess.testFunction
8 # Starting class: testPingSuccessAndResponse
9 ok 2 testPingSuccessAndResponse.testFunction
10 # Starting class: testPingSuccessCallback
11 ok 3 testPingSuccessCallback.testFunction
12 # Starting class: testPingError
13 ok 4 testPingError.testFunction
14 # Starting class: testPingErrorAndResponse
15 ok 5 testPingErrorAndResponse.testFunction
16 # Ran 5 tests in 0.001 seconds, 5 successes, 0 failures
17 # Filename: mapi_tests.tap
18 1..5
19 # Started on Tue Nov 23 11:15:29 2021
20 # Starting class: TestListSuccess
21 ok 1 TestListSuccess.testFunction
22 # Starting class: TestSubscribeSuccess
23 ok 2 TestSubscribeSuccess.testFunction
24 # Starting class: TestUnsubscribeSuccess
25 ok 3 TestUnsubscribeSuccess.testFunction
26 # Starting class: TestWrongVerbError
27 ok 4 TestWrongVerbError.testFunction
```



# Roadmap

- **Advanced test implementation in LAVA – redpesk CI**
  - ✓ *Tests that need external processes (HDMI or audio grab, etc.)*
  - ✓ *These processes need to be run on the “dispatcher”*
- **Remote access to boards in “development mode”**
  - ✓ *Directly through VPN*
  - ✓ *In LAVA, it corresponds to the “hacking session”<sup>[1]</sup>*
- **Adding test libraries shared between projects**
- **Integration of external module in order to go further in QA**
  - ✓ *Code scanner for cybersecurity*
  - ✓ *Flowchart generator (certification)*
  - ✓ *Etc.*

# Conclusion

- **Continuous testing is a must have** because of:
  - ✓ Increasing code complexity
  - ✓ L.T.S. need for industrial systems
  - ✓ Cybersecurity concern
- **Both virtual and real boards must be in the C.I. loop**
  - ✓ For a lot of tests a virtual target is enough
- **lot.bzh answers**
  - ✓ **Virtual RTMs** (Qemu in LXC) → Available in Community
  - ✓ **Real RTMs** → Work in progress!
- **Continuous testing is a part of the QA system**
  - ✓ Can be completed with other external modules

# Q&A

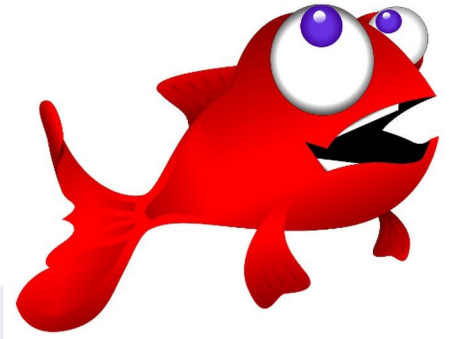


This picture is an original picture taken by Jack Mamelet in 2006. It is under the GNU Free Documentation License and the Creative Commons Attribution.

*Lorient Harbour, South Brittany, France*

# Links

- **redpesk®**
  - Website: <https://redpesk.bzh/>
  - Documentation: <https://docs.redpesk.bzh/>
  - Sources: <https://github.com/redpesk/readme>
- **IoT.bzh**
  - Website: <https://iot.bzh/>
  - Publications: <https://iot.bzh/en/publications>
  - Videos: <https://vimeo.com/search?q=redpesk>
- **Community Support**
  - Matrix.org: [+redpesk:matrix.org](https://matrix.org/join/+redpesk:matrix.org)



**redpesk®**

