# Tizen IVI “from scratch” Customizing, building and testing

Stéphane Desneux
Senior Software Engineer
Eurogiciel
<stephane.desneux@open.eurogiciel.org>


FOSDEM ’14
Brussels
1 & 2 February 2014

# Eurogiciel


- Open source development and integration:
  - Maintainers in multiple domains on tizen.org
  - Embedded systems for real-time multimedia:
    - Widi/Miracast stack
    - Wayland/Weston
    - Webkit2 browser with HW acceleration
  - Applications: HTML5/CSS3, jquery, jqmobi, Cordova
- Location : Vannes (Brittany), France










# Agenda

- Tizen & Tizen:IVI : short introduction
- From source code to target devices
- Customize
- Build
- Flash, Run, Test !



# Tizen: a short introduction

# Definition

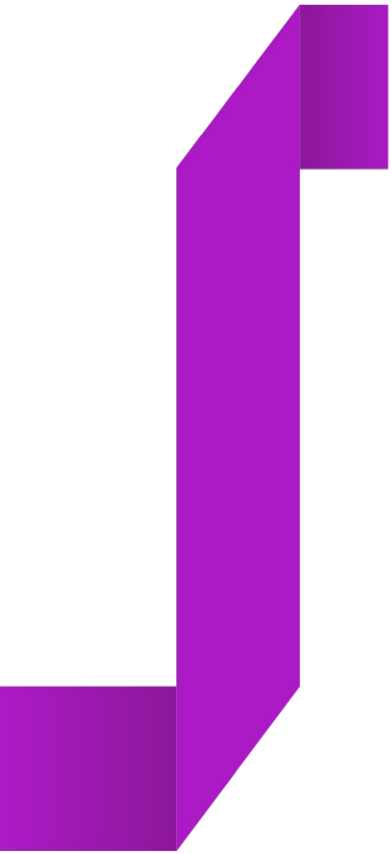
- Open source project
- Hosted at the Linux Foundation
- Innovative Web-based platform for multiple devices
- Sponsored by worldwide companies
  - Samsung & Intel are two big contributors
- Built on industry standards:
  - GNU/Linux kernel, GNU libc
  - POSIX
  - W3C
  - Many upstream Open Source projects



# Tizen Profiles

- Multiple vertical profiles
  (derived from Tizen:Generic)
  - IVI
  - Mobile
  - Future: other devices (TV, ...)
- Each profile adds its own enhancements
- Tizen packaging format: RPM



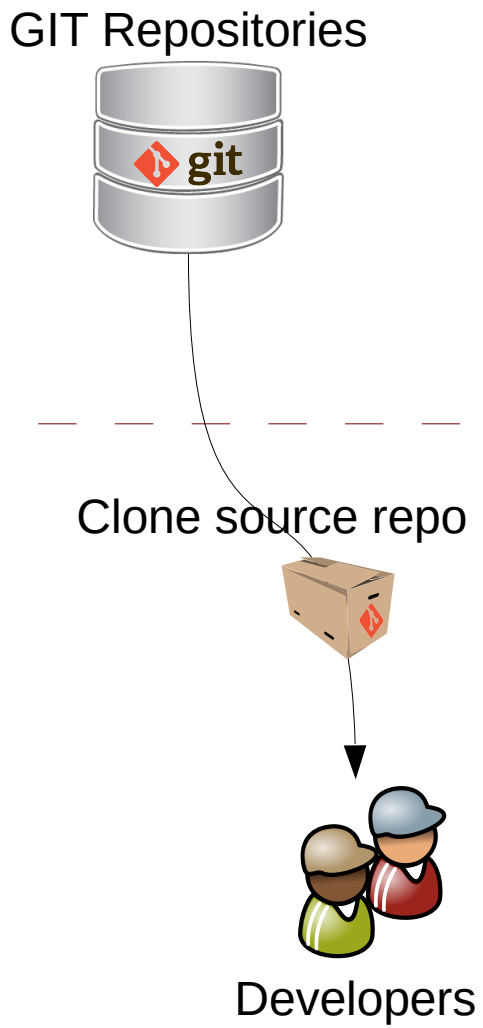


# From source code …
# … to target devices

# 1: Source code

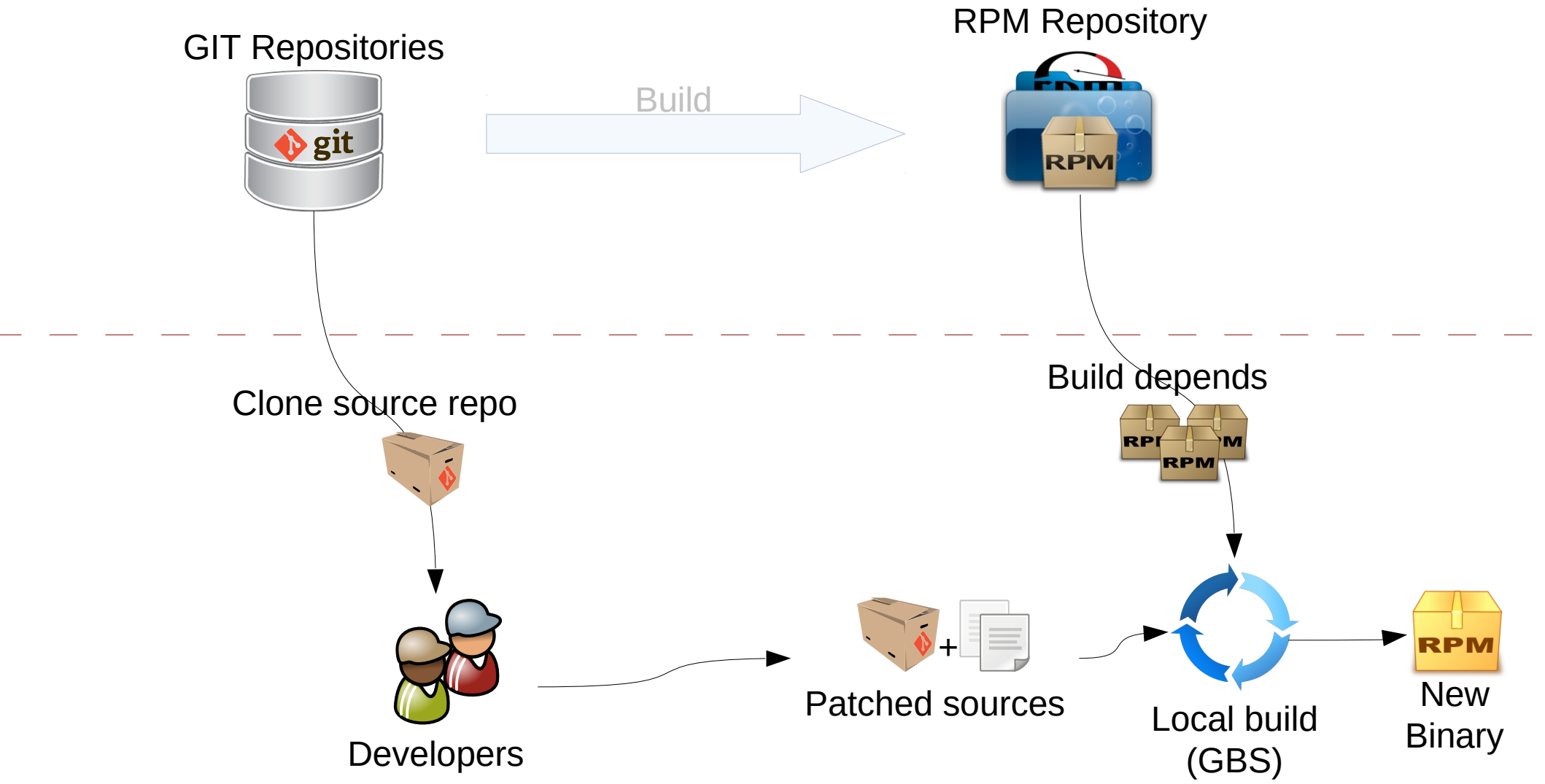
*Remote*

*Local*

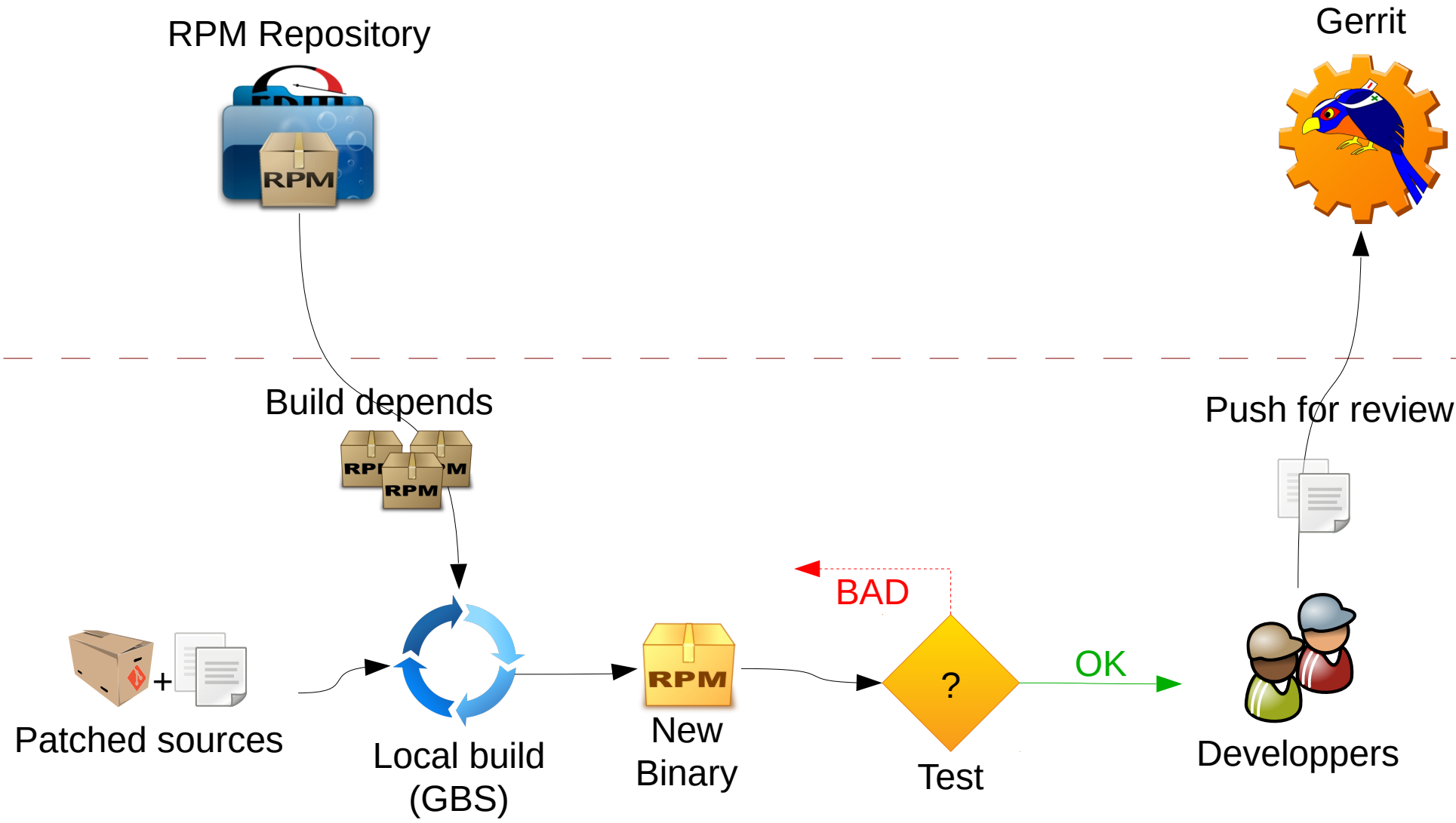




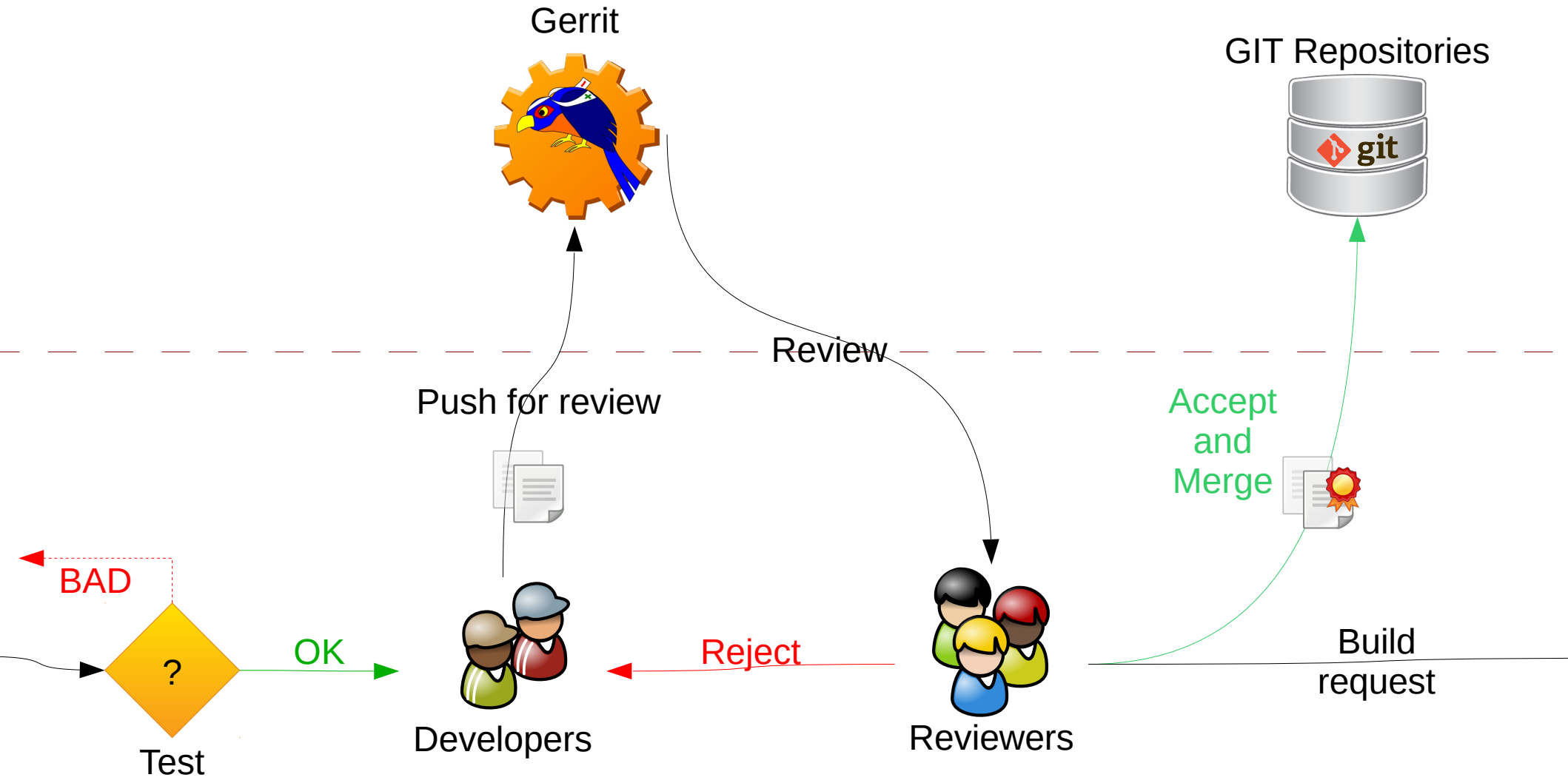

# 2: Developer local build



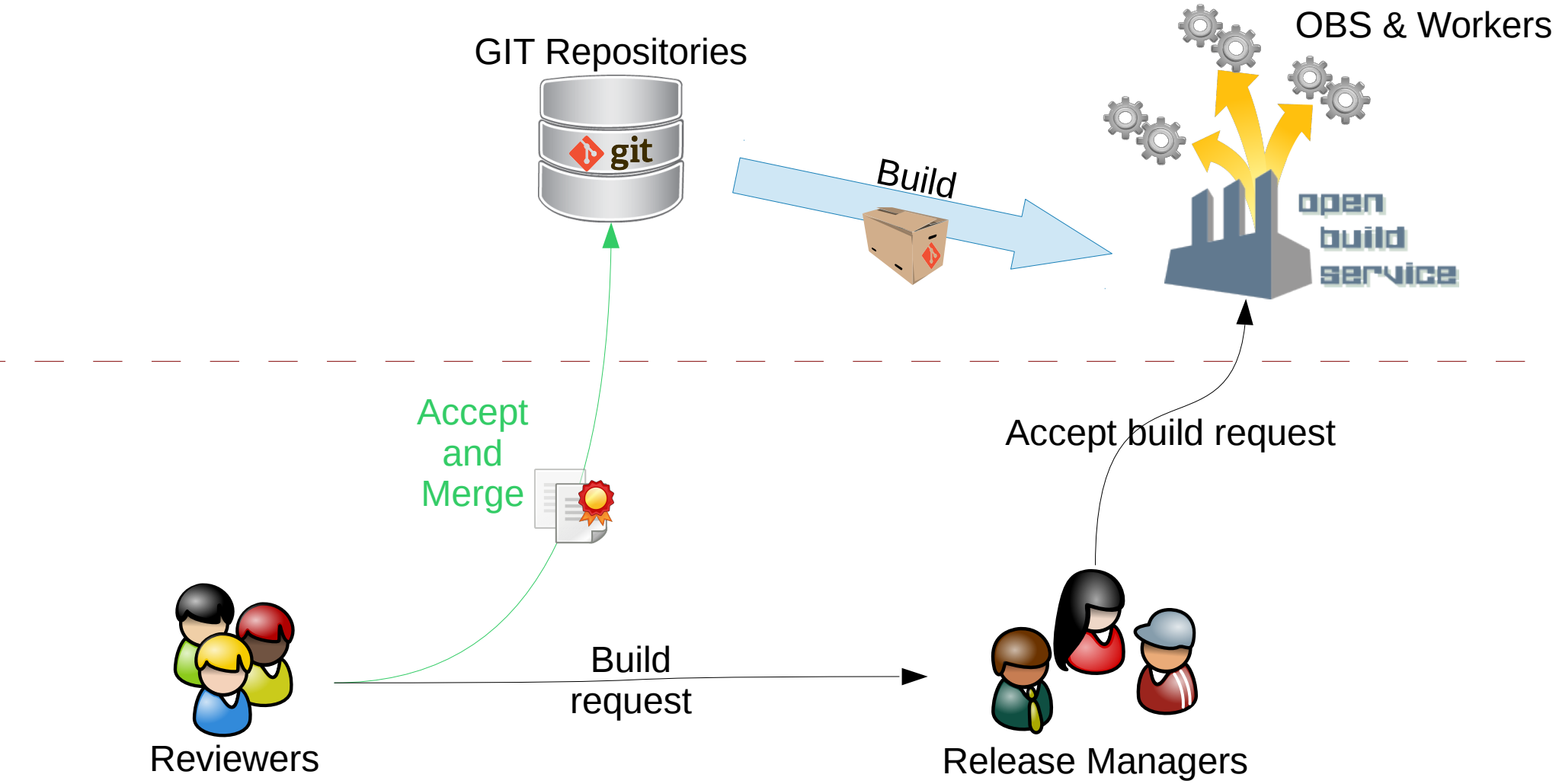
# 3: Verify & push for review



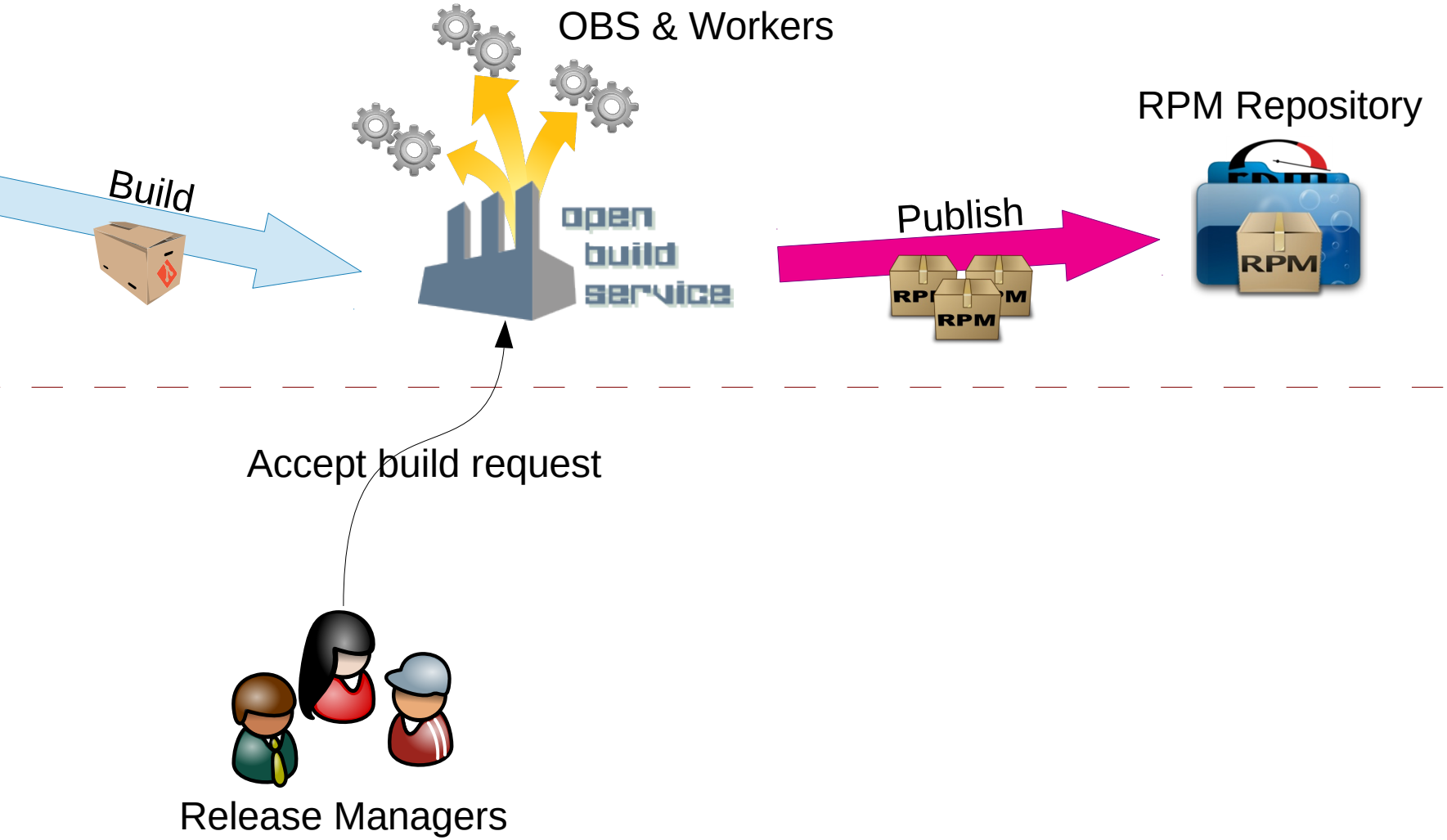
# 4: Review & Merge



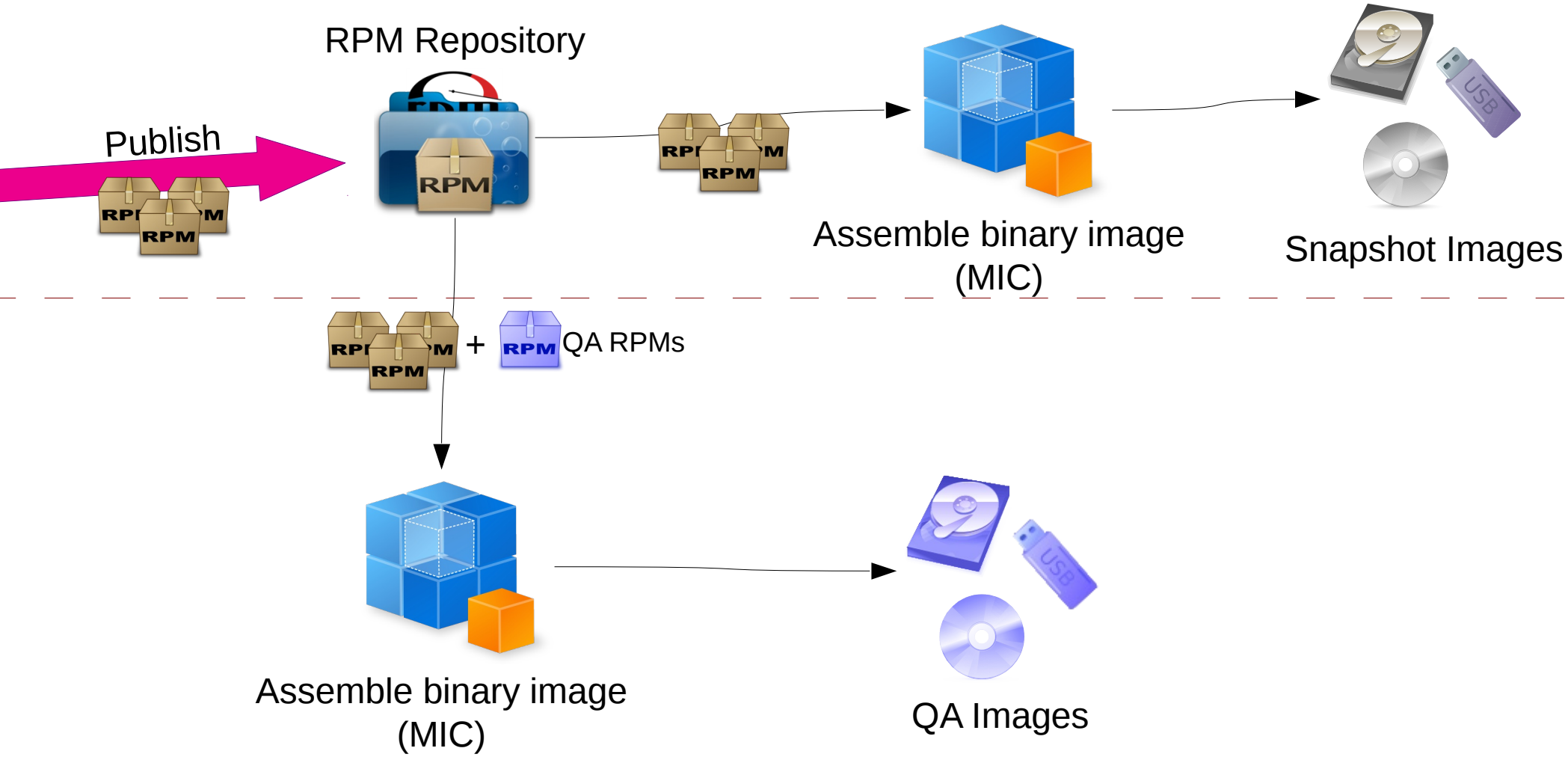
# 5: Centralized build



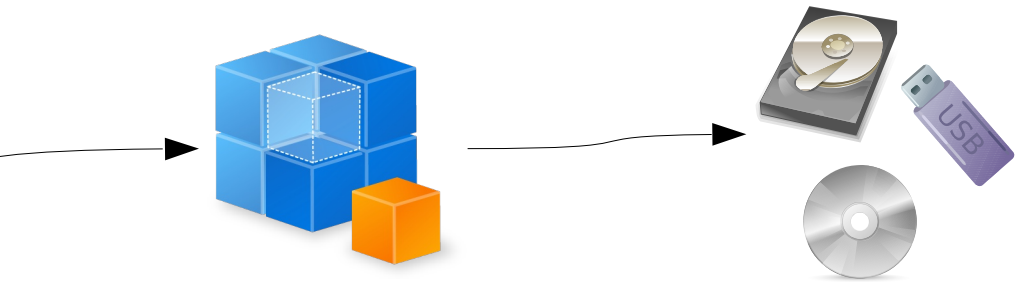
# 6: Publish repositories



# 7: Assemble binary images



# 8: installation on a target device

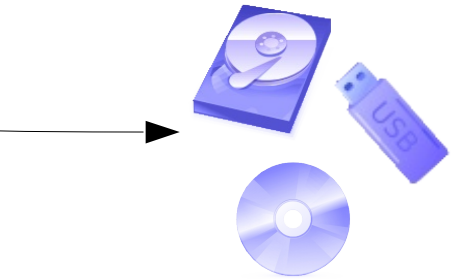


Assemble binary image (MIC)




Snapshot Images

Flash / Install




QA Images

Flash / Install

Target device

# 9: QA report & bugs

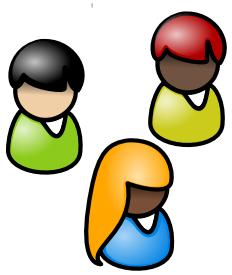

Release Images

Bug Tracker

Flash / Install

Bug Reports

Target device



QA Engineers

QA Report



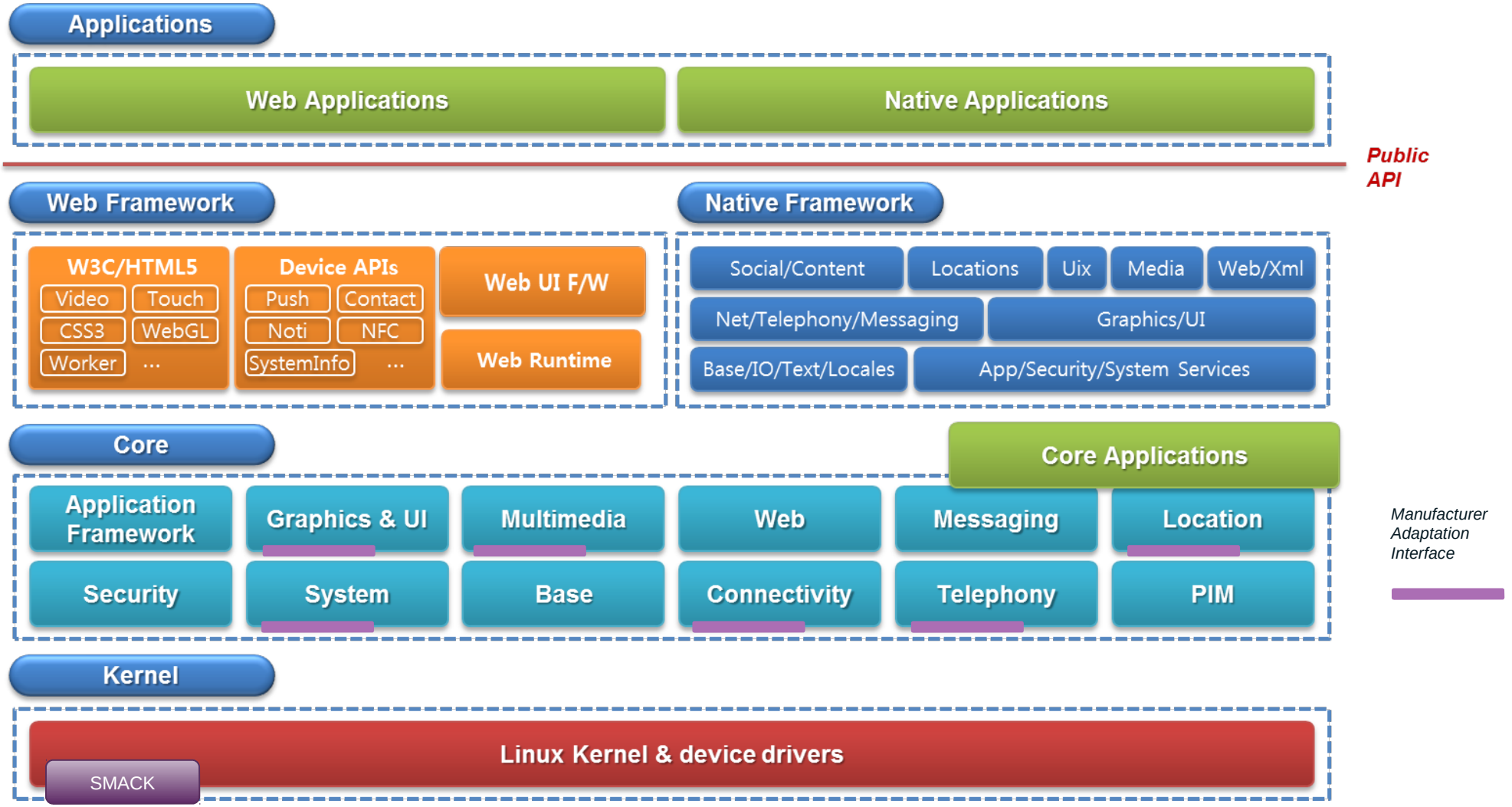
# Customize

# Why customizing ?

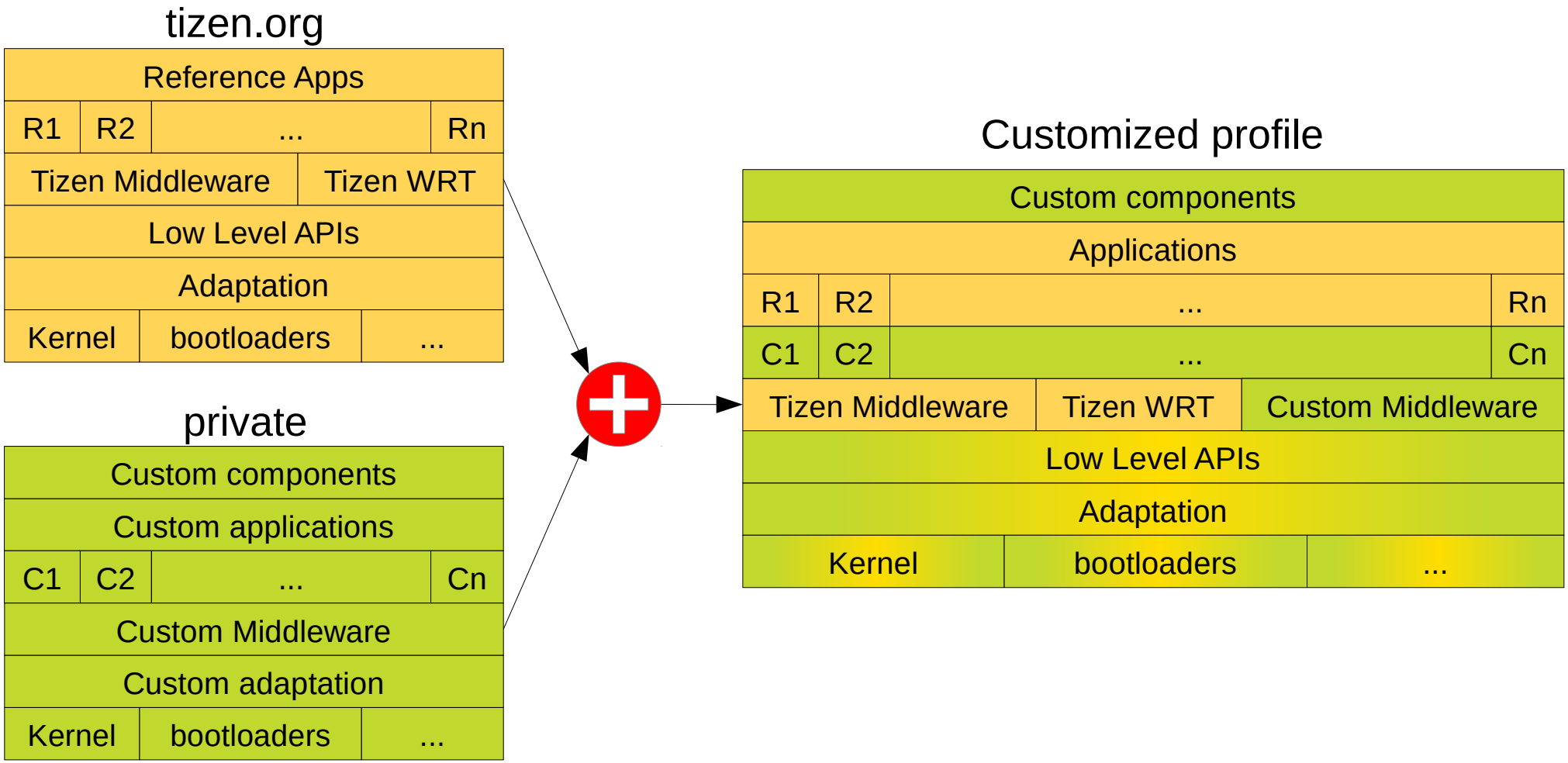
- Many reasons !
- On the software side, the vendor has to **keep control from end to end**:
  - Private sources
  - Custom hardware
  - Custom middleware
  - Extra APIs
  - Custom applications
  - Custom release & upgrade procedures



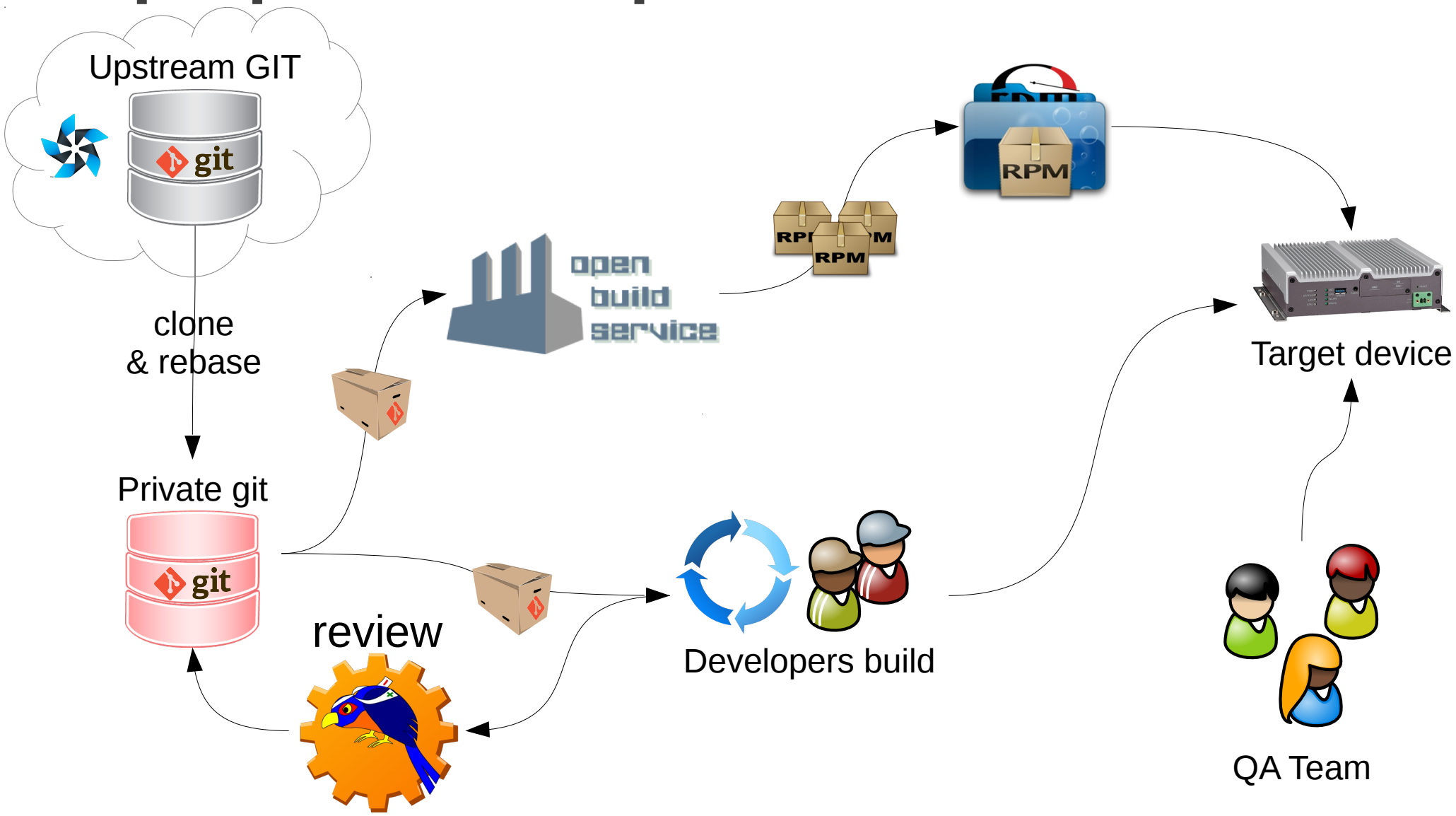
# Tizen Architecture



# How to customize ?



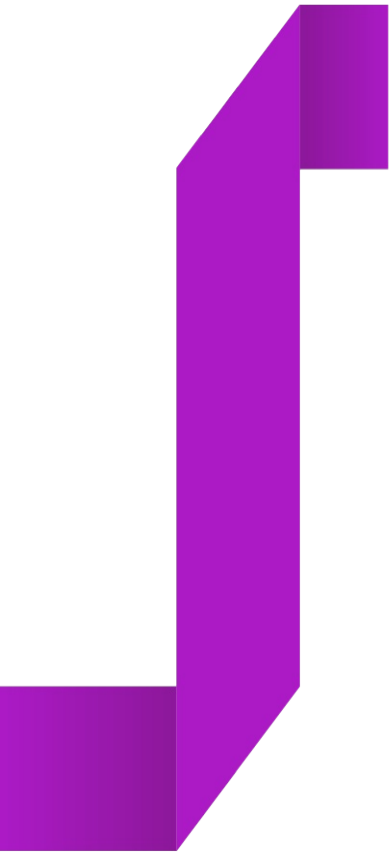
# A proposal for a private infrastructure



# Stability in movement

- Use **tizen.org as upstream** for your private git repos
- Add customizations:
  - new private repositories
  - private local branches on upstream projects
- Get the benefits from continuous integration on tizen.org and **keep synced** (`git rebase`).
- Run a *private* build infrastructure
- Generate *private* binary images for devices
- **Fast** development cycles: build / run / test / fix





# Build system

# Open Build Service (OBS)

- Open and complete platform for building a whole Linux distribution (used and maintained by openSuse)
- Provides the infrastructure to build software packages for various hardware architectures
- Fast builds: builds are distributed on multiple workers.
- Smart builds: coutinuous evaluation of the packages dependencies inside the whole project. After a change on a given package, only the needed dependent packages are rebuilt.



# Tizen build service

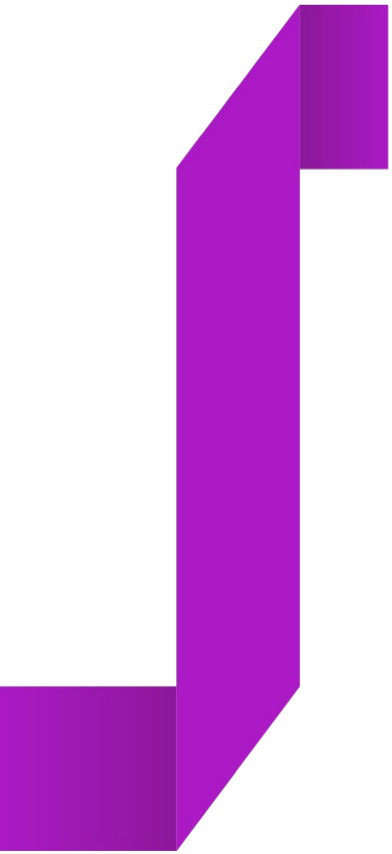
- It's a public, online OBS (read only): build.tizen.org
- Multiple HW architectures depending on the project:
  Intel archs (x86_64, ia 32), ARM (armv7l, aarch64 soon)
- Main projects:
  - Tizen:Generic X11
  - Tizen:Generic Wayland
  - Tizen:IVI
  - Tizen:Mobile
- 1 git repo on review.tizen.org == 1 source package in OBS
- Binary repos are available on download.tizen.org



# Private setup: our experience

- Used for more than 1 year to ease development of major Tizen features when a lot of packages are involved : porting Web Runtime to x86_64, Tizen:Generic setup, Multiuser mode (Tizen 3), ...
- 15 people, **15 desktop computers** bought in 2012
  (core i7 IvyBridge, 16GB RAM)
- Server side: we used the official OBS Appliance
- **No dedicated workers**. We use the developers' computers as workers.
  - 15 x 8 cores = 120 cores availables for build most of the time
  - Reduced contention between desktop activity and build activity by tuning workers config: cgroups, memory, ...
- Benchmark : a Tizen profile is usually rebuilt from scratch in **4 hours**





# Flash, Run, Test !

# Binary Images

- Binary images are automatically created when the OBS finishes a build cycle and publishes a new RPM repository.
- Special QA images are built with extra packages
  - Allow ssh and automatic login (no manual password)
  - Extra QA tools to run test suites automatically and upload reports



# Run & Test

- Developers and QA teams can pick the images in different formats:
  - RAW images (to be dumped on a HDD)
  - LiveUSB images (for USB sticks)
- The images can be shared easily worldwide to other teams.
- The target devices can be installed manually
- Useful for development and manual QA tests



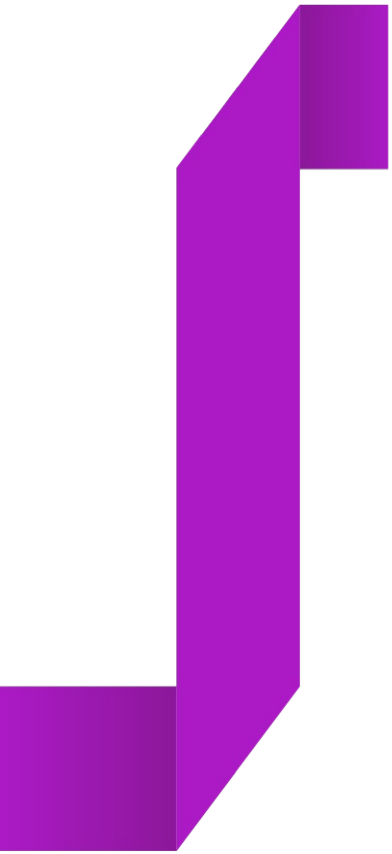
# QA automation: our experience

- Dedicated target devices for automated tests
- Boot on LAN on a custom Linux distro that runs a flashing tool: the device is flashed with the fresh Tizen snapshot and rebooted.
- At the end of the boot procedure, the QA tests start automatically
- A final test report is sent to a QAReport server
- Useful to check: sanity, performances, power consumption  tests … on every snapshot !



# Q & A

*Gulf of Morbihan, south of Brittany, France*



# Pointers

# Links – Tizen

- Main site: tizen.org
- Tizen Association: www.tizenassociation.org
- Applications Development: developer.tizen.org
- Platform Development: source.tizen.org
- Snapshots: download.tizen.org
- Documentation: developer.tizen.org/documentation
- Wiki: wiki.tizen.org
- Bugs: bugs.tizen.org



# Links – Tizen IVI

- General info: wiki.tizen.org/wiki/IVI
- GENIVI: genivi.org
- Bugs: bugs.tizen.org/jira/browse/TIVI
- Mailing list: lists.tizen.org/listinfo/ivi
- Releases and repositories: download.tizen.org/snapshots/tizen/ivi



# Links – Tizen build tools

- Development tools : download.tizen.org/tools
- GBS: source.tizen.org/documentation/reference/git-build-system
- MIC : source.tizen.org/documentation/reference/mic-image-creator
- OBS : openbuildservice.org
- GERRIT: code.google.com/p/gerrit
- GIT: git-scm.com



# Upstream projects

- Linux Kernel, SMACK, systemd, dbus
- OpenSSL, Sqlite
- X, Wayland, EFL, Enlightenment, Cairo
- Connman, BlueZ, oFono, wpa_supplicant
- Gstreamer, PulseAudio
- Webkit
- Eclipse (SDK)
- Qemu, U-boot (emulator)
- GCC, llvm, cmake, git (build)
- … and more ...