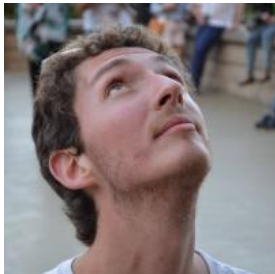




Industrialisation of applications build in embedded environment

How to build AGL (Automotive Grade Linux) applications with
Jenkins pipeline and X(cross) Development System (XDS)?



Fosdem Feb/2018
Frederic Marec Embedded Engineer
Frederic.marec@iot.bzh



Who are we ?

- **1st AGL technical contributor**
- **Located in South Brittany - France**
- **12 AGL full time developers**
 - Application Security
 - Software Development Kit
 - CI, Tests, Integration
 - Renesas Ref. Boards Support
- **Community Support**
 - Renesas Community support
 - White Papers & Conferences (Genivi, AGL, ...)
 - Documentation (kickstart, developer guides ...)

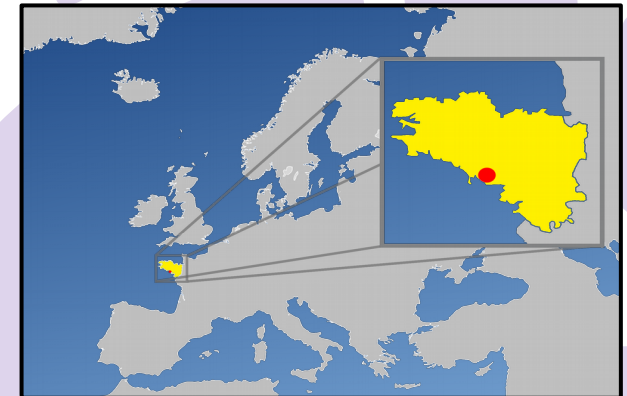
Company	Commits	Company	Commits
IoT.bzh	1684	BayLibre	11
Mentor Graphics	401	Fujitsu-Ten	11
Konsulko	203	Nexty	11
Linux Foundation	133	Panasonic	11
Renesas	106	AisinAW	10
Individual	39	Intel	10
Xevo	32	Mitsubishi	
Advanced		Electric	7
Telematics Systems	25	Microchip	5
TI	25	LG	4
ALPS	13	Samsung	4



2765 Total Commits

60 Committers 01 Jan 2017 – 01 Oct 2017

27 Companies Commits to master



AGL (Automotive Grade Linux)



AGL is a GNU/Linux system for cars

- 3rd collaborative project of the Linux Foundation
- More than 110 members
- Code first model
- Fast innovation
- Reduction of fragmentation by combining open source efforts



- Technology :
 - Based on Yocto
 - Supported boards :
 - Renesas R-Car
 - Intel MinnowBoard max
 - Raspberry Pi 3
 - QEMU
 - TI vayu



Speed up applications build process

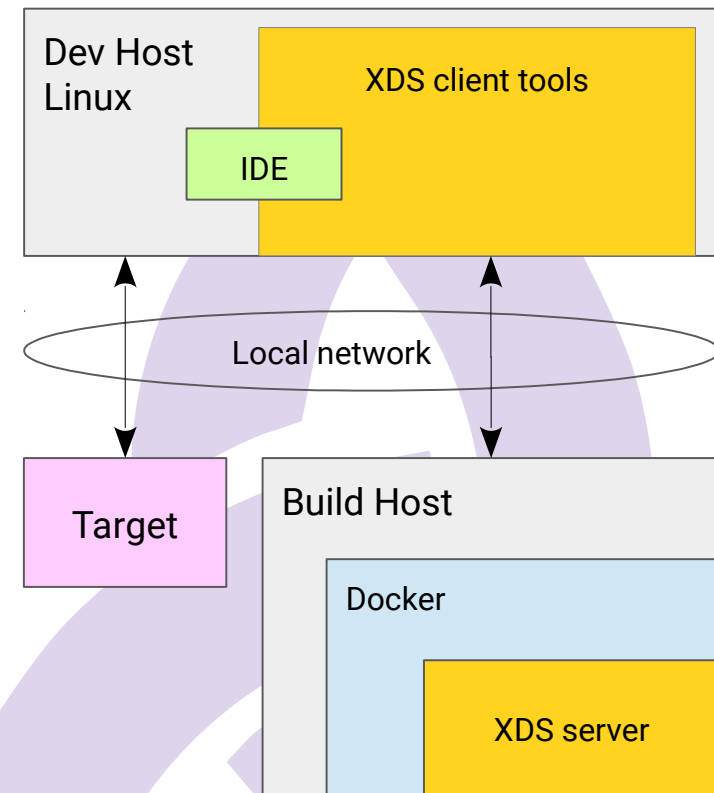
- Building applications manually is
 - Long
 - Repetitive
 - Boring
- An exponential matrix of complexity
 - Number of apps
 - Versions
 - System configuration
 - Hardware flavours
- Automating the build process is the only way to go

“AGL by IoT.bzh” build process

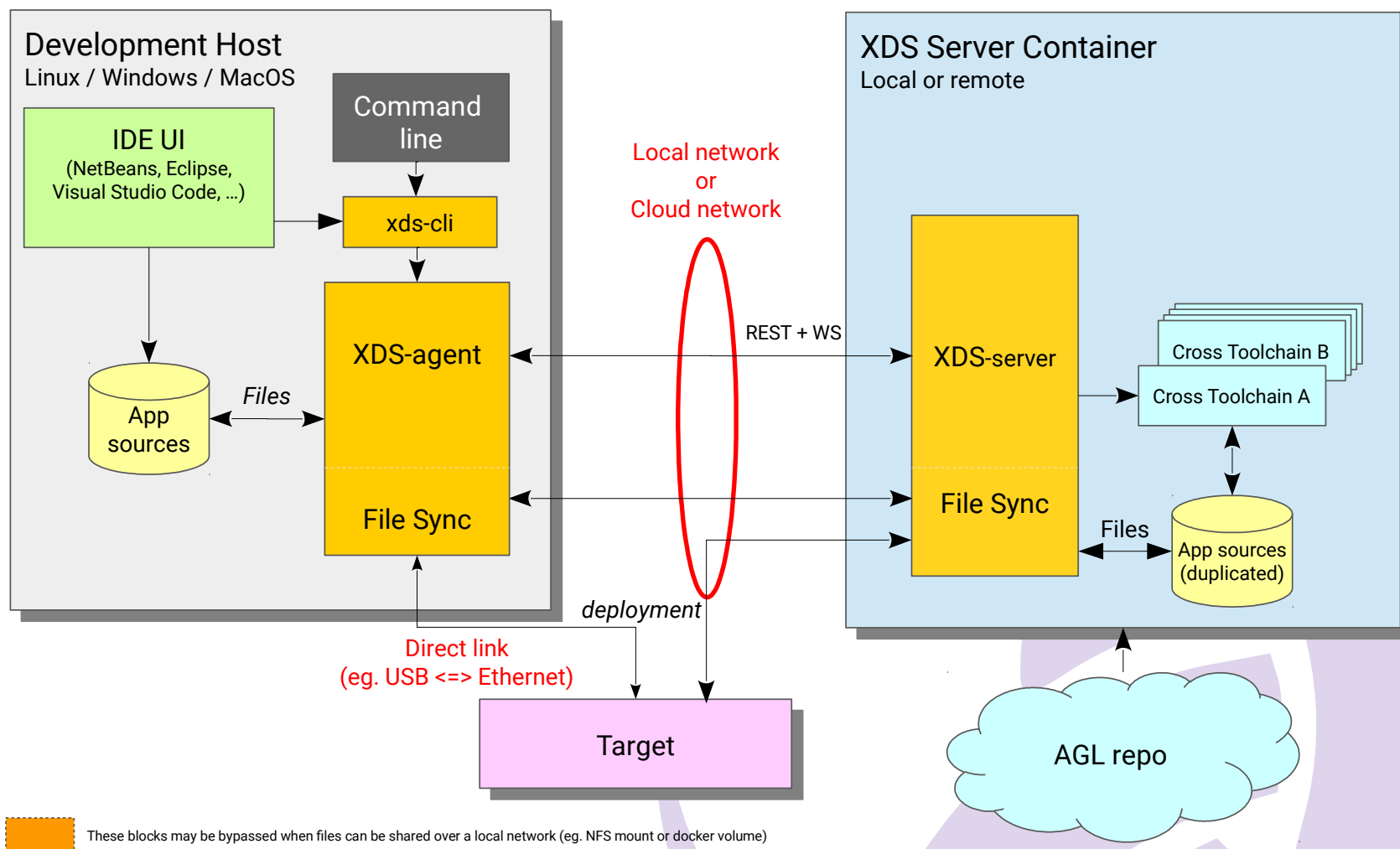
- Build platform
 - Yocto
- Cross development tool
 - XDS
- Automation CI tools
 - Jenkins
- Q A tools
 - Fuego
 - Lava

XDS X/cross Development System

- Enable embedded development outside Yocto
 - Faster process
 - Platform and application are loosely coupled
 - Standard development environment
 - Easy adoption by any developer (mobile, web, ...)
- Cross-platform build using AGL SDK toolchain
 - Intel
 - ARM32
 - ARM64
- Secure packaging
 - Widget signatures
- Deploy on development boards (or Qemu image)
- REST API based
 - Simple integration with CI
 - Integration with IDE



XDS Internal Architecture



http://docs.automotivelinux.org/docs/devguides/en/dev/reference/xds/part-1/0_Abstract.html

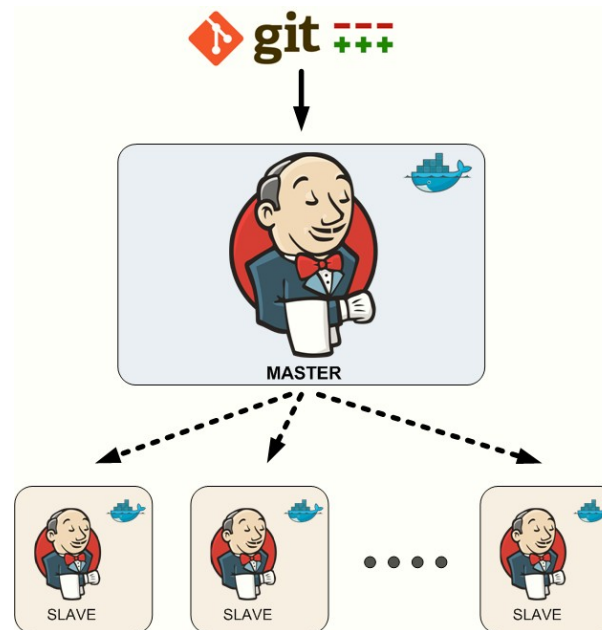
XDS Key Features

- **Multi-platform** : no dependencies on developer host (Linux / Windows / MacOS)
- **Easy to setup**
Near-zero install, no admin privileges required
- **Application sources remain local**
Compatibility with existing IT policies (e.g. corporate backup, git, ...)
- **Cross toolchain & tools embedded in a container** :
 - Local : run locally (local subsystem, virtual machine, docker container ...)
 - On-premises : run on a local build server
 - Cloud : SaaS

Jenkins CI

Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks related to :

- Building
- Testing
- Delivering
- Deploying software



Jenkins V2 improvements

- Multiple configurations per tasks (pipeline)
- Easy locating issues
- Tasks factorization
- Easy configuration
- Easy maintenance

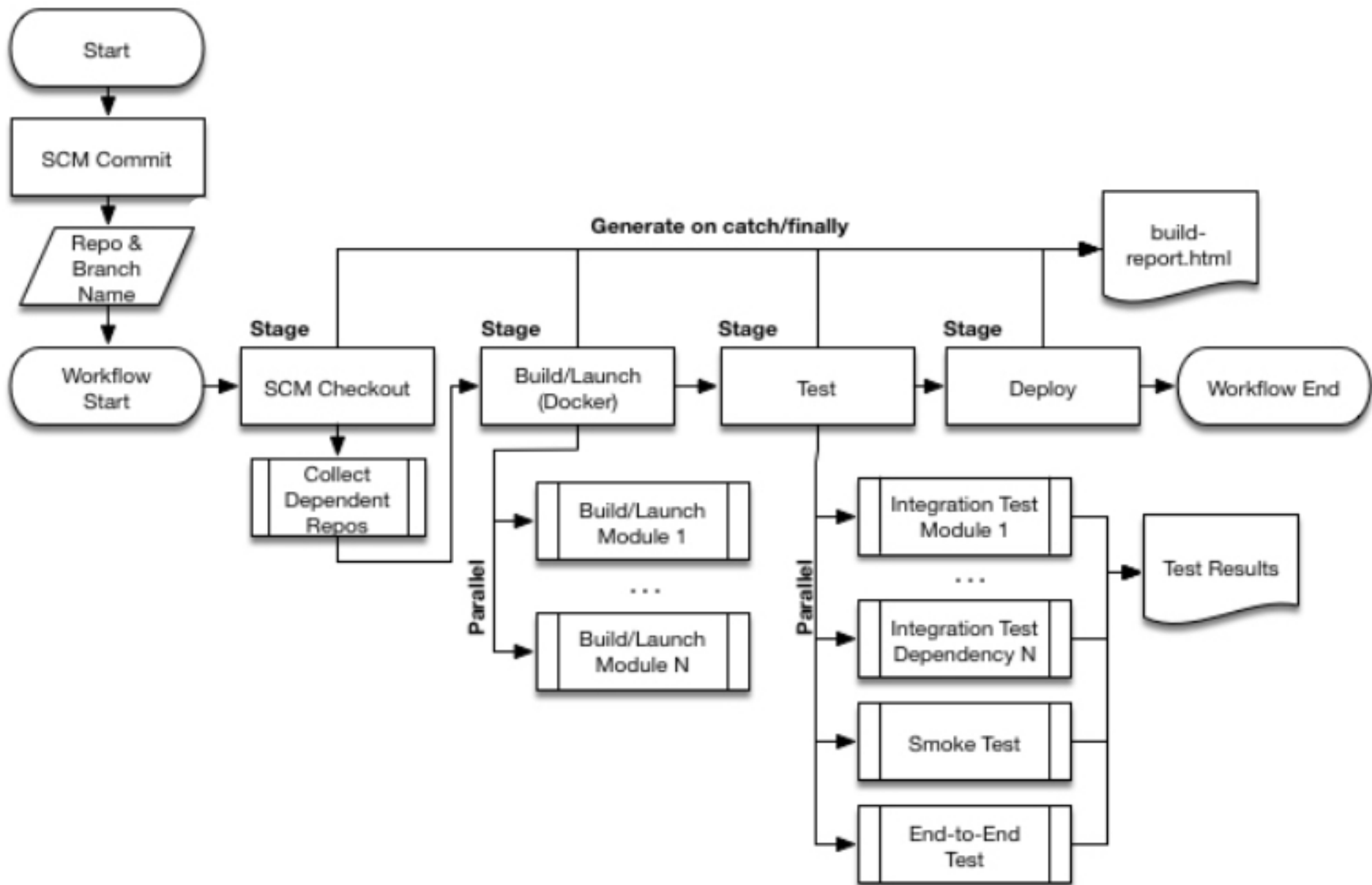


Jenkins v2 Pipeline

Pipeline adds a powerful set of automation tools onto Jenkins :

- **Code**
 - Implemented in code
 - Checked into source control
 - Giving teams the ability to edit
 - Review, and iterate
- **Durable**
 - Can survive restarts of the Jenkins master
- **Pausable**
 - Can stop and wait for input or approval
- **Versatile**
 - Support complex real-world continuous delivery requirements
 - Ability to fork/join, loop, and perform work in parallel

Pipeline flow example : continuous delivery



Average stage times:
(Average full run time: ~9s)

Build	Declarative: Checkout SCM	Setup	Build	Publish	Declarative: Post Actions
#37	1s	3s	1s	104ms	66ms
#36	1s	4s	1s	124ms	60ms
#35	2s	4s	535ms	117ms	56ms
#34	2s	4s	441ms	44ms	82ms
#33	2s	4s	441ms	44ms	82ms

Failed with the following error(s)

Shell Script script returned exit code 1

See stage logs for more detail.

Logs

Jenkins Blue Ocean

Designed for Jenkins Pipeline :

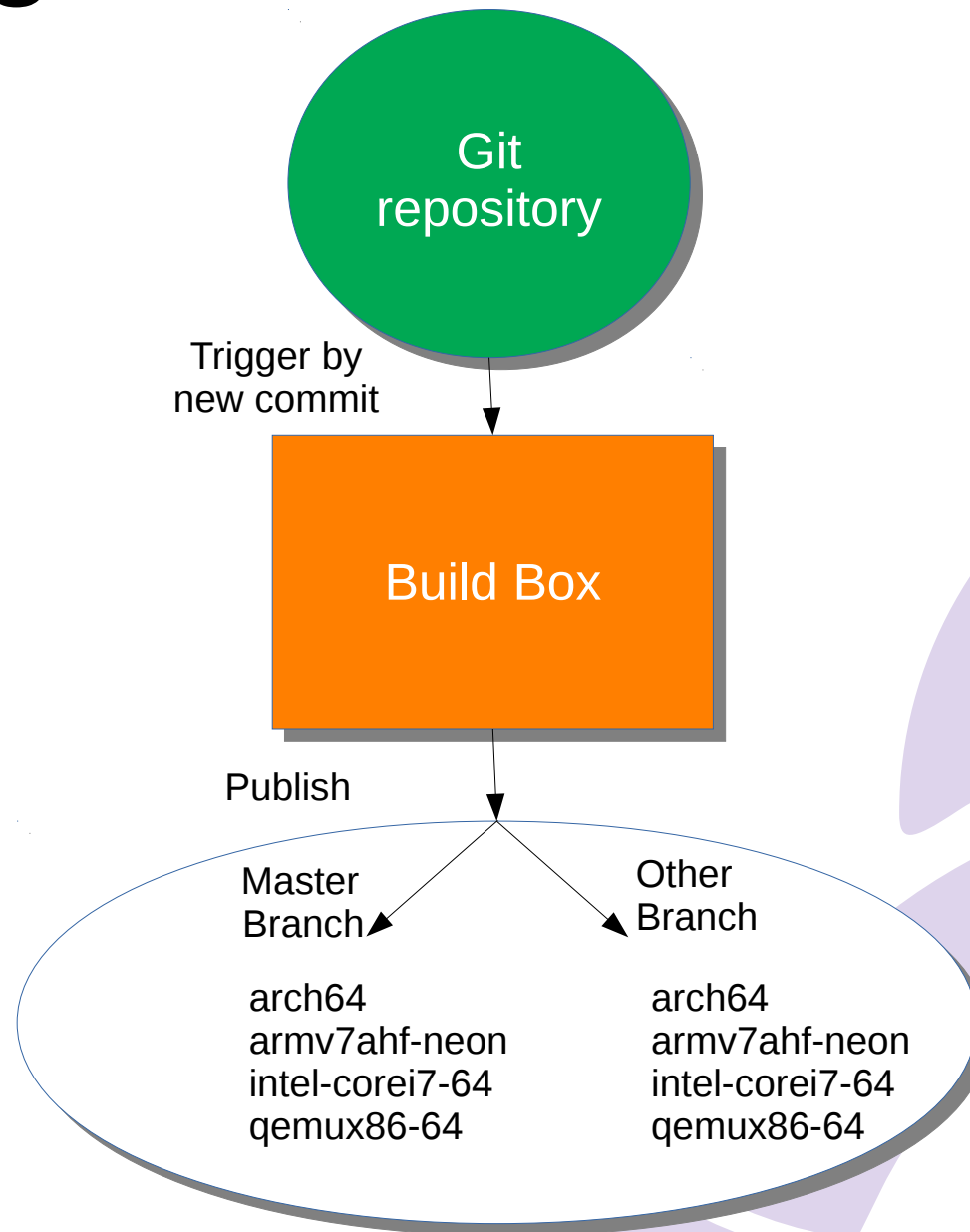
- Allowing fast and intuitive comprehension of pipeline's status
- Pinpoint precision when intervention is needed and/or issues arise
- Native integration for branch and pull requests

The screenshot displays the Jenkins Blue Ocean interface for a pipeline named 'IOABIEE 2'. The top navigation bar includes tabs for 'Pipeline', 'Modifications', 'Tests', and 'Artefacts', along with a 'Déconnexion' button. The main header shows the pipeline status as 'Branche: master' and 'Commit: 14c56a5', with a duration of '11s' and a message 'Aucune modification'. Below this, a progress bar indicates the pipeline stages: 'Start', 'Setup', 'Build', 'Publish', and 'End'. The 'Setup', 'Build', and 'Publish' stages are marked with green checkmarks, indicating successful completion. The 'Publish' stage is expanded, showing a list of steps: 'Print Message', 'Archiver des artefacts', and 'Recursively delete the current directory from the workspace', each with a green checkmark and a duration of '<1s'.

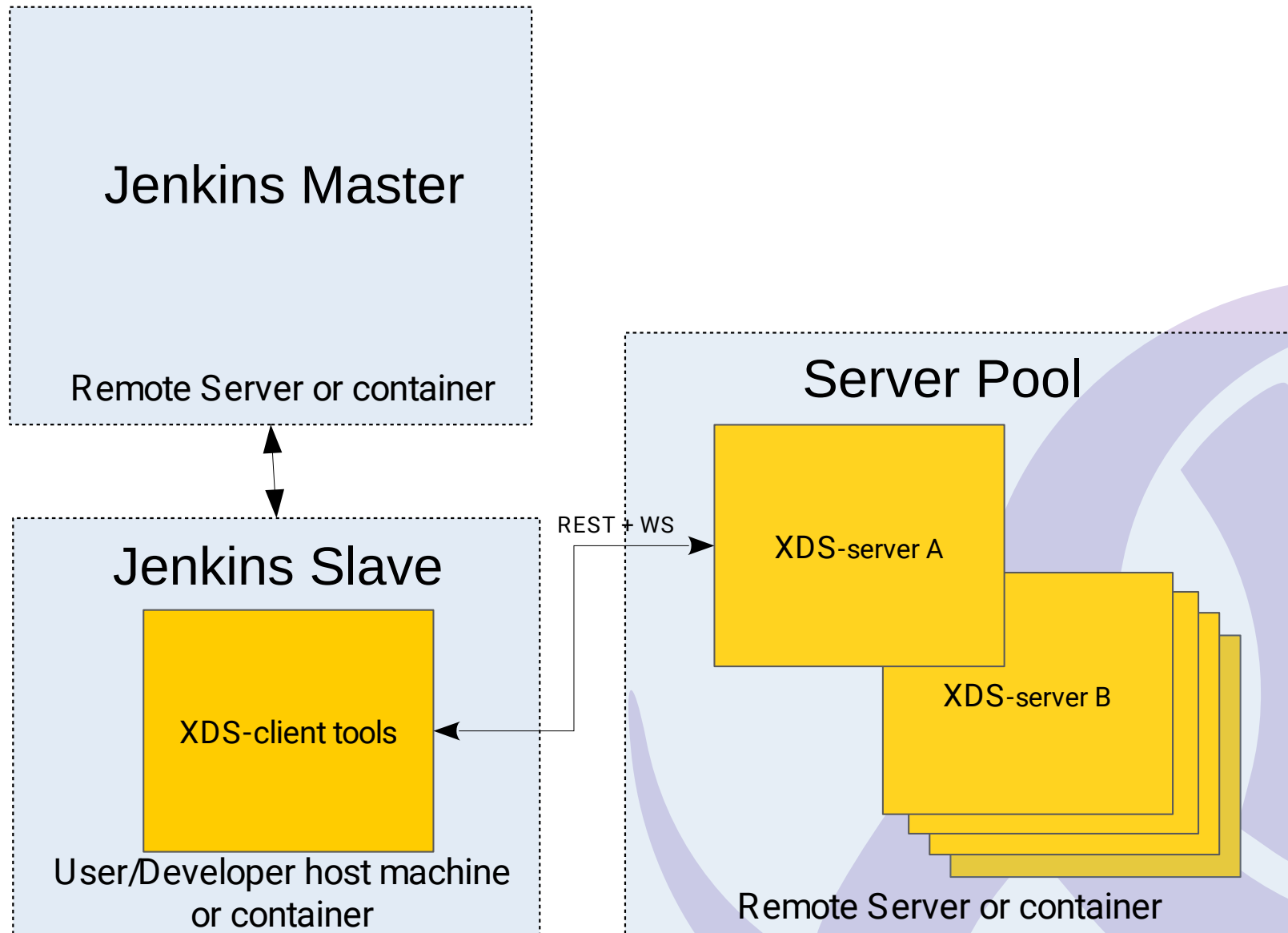
Integration needs

- Build for multi-configuration systems
- Avoid YOCTO complexity
- Trigger a job from a git repository
- Automate building
- Publish results

Integration architecture



Inside the box



XDS + Jenkins Outcome

- Jenkins can automatically trigger code build and publish results
- Jenkins pipeline can build in parallel
- XDS tools can cross-build source code using AGL SDK toolchain avoiding YOCTO complexity

IoT.bzh is hiring

<http://iot.bzh/en/jobs>

- 6 open positions
 - Kernel / Yocto system engineer (M/F)
 - Web designer, Angular, NodeJS (M/F)
 - Audio engineer (M/F)
 - Vehicle 2 Cloud engineer (M/F)
 - Junior DevOps engineer (M/F)
 - Technical writer (M/F)
- 2 internships (engineer level)
 - Dynamic, curious, autonomous,
 - Contributor or good knowledge of open source
 - Good english level





Demo syndrome

Solution released online

<https://github.com/DDTLK/Industrialisation-of-applications-build-in-embedded-environment>

<https://github.com/DDTLK/IOABIEE>

- Next steps

- Load-balancing management of XDS-server
- Dynamic sizing of XDS-server pool
- Add test & deploy stages (LAVA + Fuego integration)

- References

- <https://www.automotivelinux.org/about/members>
- <https://www.linuxfoundation.org/projects/>
- <https://wiki.jenkins.io/#all-updates>
- <https://jenkins.io/>
- <http://docs.automotivelinux.org/>
- <https://www.automotivelinux.org/>
- <https://gerrit.automotivelinux.org/gerrit/#/admin/projects/>
- <https://www.lorientbretagnesudtourisme.fr>



<http://iot.bzh/en/jobs>



IoT.bzh is hiring

j01n u5



Thank you everyone!