Continuous Integration
& **Buildbot**

TIZEN OPENSTACK POWERTOP YOCTO CONNMAN XEN GUPNP KVM POKY OFONO LINUX KEI
ANDROID FOR INTEL ARCHITECTURE INTEL LINUX WIRELESS
INTEL LINUX GRAPHICS SYNCEVOLUTION SIMPLE FIRMWARE INTERFACE (SFI) ENTERPRISE SECURITY IN

Raphael Kubo da Costa
Intel Open Source Technology Center

What this talk is about

Why continuous integration is cool

Why Buildbot
is cool

How to do some
cool things with Buildbot

What this talk is **not** about

Exploration of *all* of Buildbot's coolness

Why Jenkins
is not cool

Raphael Kubo da Costa

Python since 2002(ish)
In .fi since September 2012, from .br
Software Engineer, OTC
Infra & Tools for Crosswalk

Continuous Integration

Continuous Integration

the process of **repeatedly** *and* **paranoidly**
making sure my code still works as intended

[...] works as intended

[...] works as intended

Does my code build?

[...] works as intended

Does my code
pass my unit tests?

[...] works as intended

Does my code work with my other projects?

[...] works as intended

Does my code consume more <...>?

Starting Point
"code working as intended"
is important

Manual
is fine for 1 user
(usually the developer)

But what if
You target
multiple platforms

But what if
Your build/test takes time

But what if
You want to know
what a pristine build
looks like

But what if

You want to email/ping people on IRC if something fails

Buildbot

Buildbot
is Free Software
GPLv2, public mailing list,
source code on GitHub, IRC channel

Buildbot
is written in Python
Twisted under the hood

Buildbot

has many users

Buildbot – Python – WebKit –
Mozilla – Chromium – big etc

Why not
Jenkins/Travis-CI/etc?

Many CI tools are structured as ready-to-use applications. Users fill in specific details, but the fundamental design is fixed.

Such tools embody assumptions about the structure of the project and its processes.

— **buildbot.net**

Buildbot is complex, but only if you're trying to view it as an out-of-the-box CI solution. Buildbot suddenly starts to make much more sense if you view it as a **framework** *for creating your own CI solution, not a CI server in its own right.*

— Jacob Kaplan-Moss

Demo
Fingers crossed...

Main Concepts

Master

Master

A master checks for changes in the repository

A master checks whether they are relevant

A master decides what commands to run

A master decides where the commands will be run

Scheduler

Scheduler

A Scheduler decides whether a change is relevant and triggers a Builder

Builder

Builder

A Builder is a collection of Slaves.

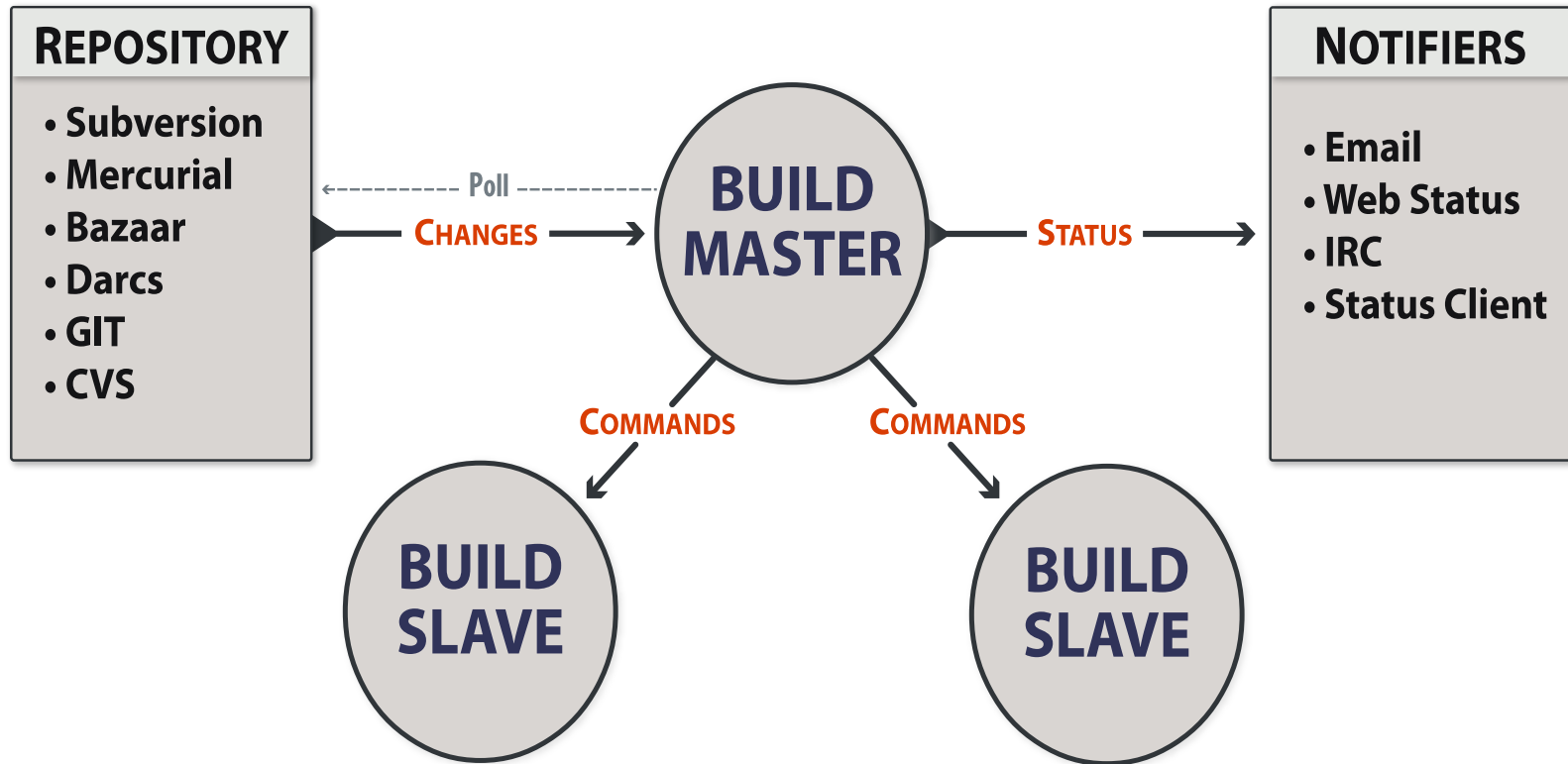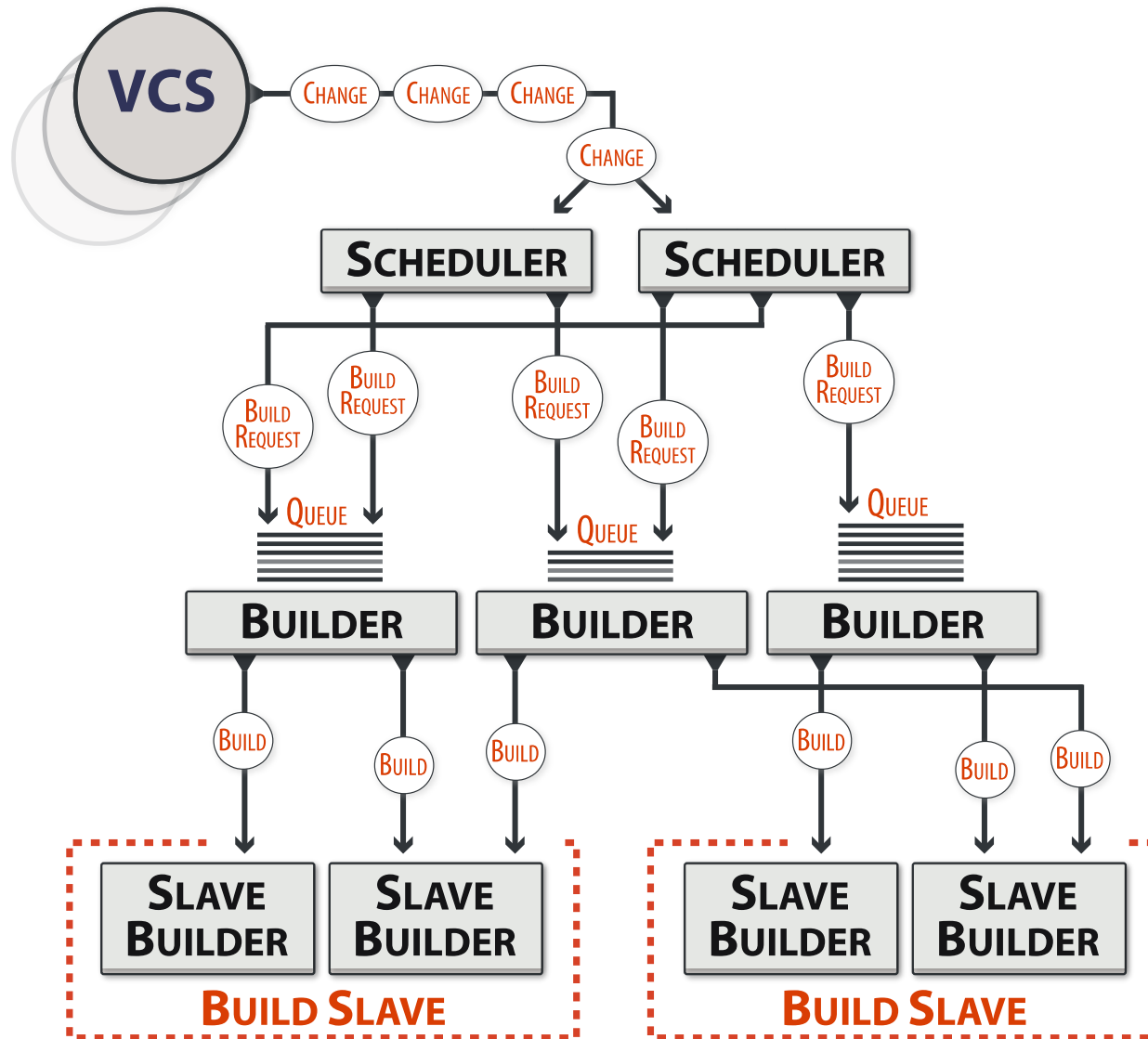A Builder decides which Slaves will run the build steps the Master wants.

Slave

Slave

A Slave actually runs what we want.

TL;DR

Let's go to
master.cfg

Buildbot is a
framework

In real life

Lwan is a
toy web server created
by a friend of mine

Crosswalk is a web runtime based on Chromium

Chromium is ~~not a good example~~ a web browser

Buildbot is a **framework**

Build documentation
Email notifications
Try slaves
UI tests
Device tests
Run periodic tasks

Build documentation
Email notifications
Try slaves
UI tests
Device tests
Run periodic tasks

Buildbot is
Free Software

Kiitos