

# isar

## Platform for Binary Distributions

Agenda  
Meet Cedric!  
Binary vs Source Distributions  
Our Challenges  
Testing

We have been using isar and Debian as a base for several products.  
isar is a great platform but there are some rough edges (upstream and downstream)

### Agenda:

- Who am I?
- Challenges with packages produced by isar
- Testing isar and changes to isar

# Meet Cedric!

A boring but short bio...

Agenda  
Meet Cedric!  
Binary vs Source Distributions  
Our Challenges  
Testing

- French (whatever that means these days....)
- Linux user since ~1995
- Started my career in an RTOS company in 2000
- Joined MontaVista in 2007
  - Started in the networking group
  - Later moved in the Auto group where we pioneered use of Linux in vehicles
- Joined Mentor Graphics in 2013 (Auto group acquired)
  - Continued with Auto stuff
  - Later moved to Medical & Industrial markets with Android, Debian and Yocto
- Joined Siemens in 2017 (via yet another acquisition 😊)
  - Continued with existing Mentor Graphics Linux business
  - Later focused on Industrial and Debian-based products in particular
  - And here I am!

# Binary Distributions

## vs Source Distributions

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

- Debian is a binary Linux distribution
  - ➔ same binaries if there are no changes to sources

Heavy reliance on the apt package manager

- Yocto is a source-based Linux distributions:  
project may re-configure packages via bbappends or e.g. local.conf
  - ➔ binaries are possibly project-specific  
(reproducible builds remain supported)

Main objective: produce an optimized target image

Binary package feeds may be created/used: less common though

# Our Challenges

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

## Goal:

- use Isar to offer a subset of Debian + custom packages in the form of package feeds

## Challenges:

- bitbake: God and Evil
- Bootstraps in a multiconfig world
- Non-compliant packages to Debian's policy
- Missing-or-bad debian/changelog for Isar-generated packages?
- Missing-or-bad maintainer scripts
- Non-reproducible builds

# Bitbake: God and Evil

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

bitbake excels at evaluating meta-data dynamically according to layers and variables

Debian (source) packages also support build-time options:

- DEB\_BUILD\_PROFILES
- DEB\_BUILD\_OPTIONS

But at the end of the day, binary packages are defined by their name + version + architecture and a single instance of a source package may exist in the apt repository

Recommendations:

- Carefully review assignments to DEBIAN\_\* and DPKG\_\* variables in Isar
- Your goal shall be to produce unique and deterministic debian/ meta-data
  - ➔ Leverage DEB\_BUILD\_PROFILES and DEB\_BUILD\_OPTIONS for conditional stuff in debian/control and Debian/rules
- Suffix binary packages with e.g. -\${MACHINE} if they carry machine-specific bits

# Bootstraps in a multiconfig world

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

## Goal:

- Use multiconfig builds to generate packages for all supported configurations for a Isar-based distribution

## Problem:

- Some hardware enable layers add custom sources:  
`DISTRO_APT_SOURCES:append = " conf/machine/nvidia-common-36.4.3.list"`
- Bootstraps generated for this multiconfig differ from others → broken mc

## Work-around:

- Use same sources for all mc builds → not great

## Solution?

- Use SBUILD\_FLAVOR for package and image builds

# Non-compliant Packages

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

Most packages claim compliance against Standard-Version but

- Inconsistencies in Standard-Version (should Isar mandate a minimum version?)
- Should dpkg classes call lintian in a do\_qa task of some sort?
- Should Epoch default to a more recent date?

```
E: no-ctrl-alt-del changes: bad-distribution-in-changes-file isar
E: no-ctrl-alt-del: extended-description-is-empty
E: no-ctrl-alt-del: no-copyright-file
E: no-ctrl-alt-del: package-contains-ancient-file 1970-01-01 [usr/]
E: no-ctrl-alt-del changes: unreleased-changes
W: no-ctrl-alt-del: empty-binary-package
W: no-ctrl-alt-del: maintainer-script-calls-systemctl [postinst:3]
W: no-ctrl-alt-del: maintainer-script-calls-systemctl [postrm:3]
W: no-ctrl-alt-del: maintainer-script-ignores-errors [postinst]
W: no-ctrl-alt-del: maintainer-script-ignores-errors [postrm]
```

# Bad Maintainer Scripts

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

The following is a commonly found pattern for package recipes shipping a system unit:

```
# postinst file shipped by a recipe
# no {pre,post}rm scripts

...
# do some stuff here
...

systemctl enable foo-bar.service
```

➔ If `${PN} == "foo-bar"` and `${WORKDIR}/${PN}.service` is found (listed in `SRC_URI`):

- Isar will copy the `.service` file to `${S}/debian/`
- debhelper will install its magic in maintainer scripts

➔ Uninstallation not covered (may not be a use-case for you but packagers should aim for compliance to Debian standards)

# Bad Maintainer Scripts

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

Let debhelper do its magic by using #DEBHELPER#:

```
# postinst file shipped by a recipe
# no custom steps in {pre,post}rm scripts

...
# do some stuff here
...

#DEBHELPER#
```

- ➔ debhelper will add its code to install/uninstall the service
- ➔ Your package may now be used in a binary Distribution
- ➔ We need a linter for our package recipes!

# Non-Reproducible Builds

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

```
chombourger@DISW4XS7Q34 ~/D/m/b/base-apt (ch/perf/apt-upload) [1]> diffoscope tmp/work/lx1-bookworm-amd64/node-red/4.0.9-r0/node-red_4.0.9.tar.gz tmp/work/lx1-bookworm-arm64/node-red/4.0.9-r0/node-red_4.0.9.tar.gz
--- tmp/work/lx1-bookworm-amd64/node-red/4.0.9-r0/node-red_4.0.9.tar.gz
+++ tmp/work/lx1-bookworm-arm64/node-red/4.0.9-r0/node-red_4.0.9.tar.gz
node-red_4.0.9.tar
- file list
- @@ -1,7 +1,7 @@
- drwxr-xr-x 0 0 0 0 2025-05-08 02:36:34.000000 node-red-4.0.9/
- drwxr-xr-x 0 0 0 0 2025-05-08 02:36:34.000000 node-red-4.0.9/debian/
- -rw-r--r-- 0 0 0 143 2025-05-08 02:36:34.000000 node-red-4.0.9/debian/changelog
- -rw-r--r-- 0 0 0 370 2025-05-08 02:36:34.000000 node-red-4.0.9/debian/control
- -rw-r--r-- 0 0 0 8 2025-05-08 02:36:34.000000 node-red-4.0.9/debian/node-red.install
- -rw-r--r-- 0 0 0 202 2025-05-08 02:36:34.000000 node-red-4.0.9/debian/node-red.service
- -rw-r-xr-x 0 0 0 806 2025-05-08 02:36:34.000000 node-red-4.0.9/debian/rules
+ -rw-r-xr-x 0 0 0 810 2025-05-08 02:36:34.000000 node-red-4.0.9/debian/rules
node-red-4.0.9/debian/rules
@@ -13,12 +13,12 @@
export npm_config_cache=/home/builder/node-red/npm_cache

override_dh_clean:
rm -rf //node_modules
rm -rf

override_dh_auto_build:
- cd / && npm install --only=production --no-package-lock --verbose --arch=x64 --target_arch=x64 --no-audit --prefer-offline --prefix /home/builder/node-red/image/usr -g --build-from-source --no-save /downloads/node-red-4.0.9-bundled.tgz
+ cd / && npm install --only=production --no-package-lock --verbose --arch=arm64 --target_arch=arm64 --no-audit --prefer-offline --prefix /home/builder/node-red/image/usr -g --build-from-source --no-save /downloads/node-red-4.0.9-bundled.tgz
if [ -n "" ]; then rm -f //node_modules/.package-lock.json; fi

# disable slow stripping - not enough value for our ad-hoc npm packaging
override_dh_strip_nondeterminism:
chombourger@DISW4XS7Q34 ~/D/m/b/base-apt (ch/perf/apt-upload) [1]> |
```

- ➔ debian/rules timestamps
- ➔ hard-coded arch in debian/rules

# Non-Reproducible Builds

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

```
# Quick Fix for meta-iot2050/classes/npm.bbclass

...
# Simplified version
do_prepare_build:append() {
    cat <<EOF >>${S}/debian/rules

override_dh_auto_build:
    npm install --arch=\$(DEB_HOST_ARCH) /downloads/bundle.tgz
EOF
    touch --reference ${S}/debian/control ${S}/debian/rules
}
...
```

- ➔ Works but still not good
- ➔ Hard-coded isar path: /downloads

# Non-Reproducible Builds

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

```
# Differences for debian/pool/trixie/u/u-boot-iot2050/u-boot-iot2050_2025.04.dsc ---
--- apt/debian/pool/trixie/u/u-boot-iot2050/u-boot-iot2050_2025.04.dsc
+++ /tmp/tmp5ejsj5_e/u-boot-iot2050_2025.04.dsc
|— Files
|  @@ -1,2 +1,2 @@
|
|  - 216f7f4bb08a3f58bde057400695f848 44858640 u-boot-iot2050_2025.04.tar.gz
|  + 53cedf5091d91c7caf203b87f75e4a21 44858675 u-boot-iot2050_2025.04.tar.gz
|— u-boot-iot2050_2025.04.tar.gz
|  |— u-boot-iot2050_2025.04.tar
|  |  |— u-boot-2025.04/localversion
|  |  |  @@ -1 +1 @@
|  |  |  --2025.S08-4.2_SPRINT-1-gc3e0bf8
|  |  |  +-2025.S08-4.2_SPRINT-1-g79ee880...
```

- ➔ Different commit hash on each build
- ➔ bitbake should use --committer-date-is-author-date

# Conclusion for Part I

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

- None of the issues are major
- This talk was to create some awareness for things we should do and check for
- We can fix all of these issues!
- How to test our changes? Or to avoid regressions?
- Let's talk about Testing!

# isar Test Suite

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

We have a test suite! It is however not easy to use ☹️

```
chombourger@DISW4XS7Q34 ~/U/d/isar (next)> ./scripts/ci_shell
2025-10-31 14:46:37 - INFO      - kas 4.8.2 started
2025-10-31 14:46:37 - INFO      - Using /repo as root for repository isar
2025-10-31 14:46:37 - INFO      - To start the default build, run: bitbake -c build core-image-minimal
🔍 Checking dependencies...
🔧 Installing missing dependencies...
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78, <> line 293.)
debconf: falling back to frontend: Readline
Extracting templates from packages: 100%

💡 Use "tlist citest.py" to list existing test cases
💡 Use "trun citest.py:TestClass.test_method" to run a specific test
💡 Use "trun -p sstate=1 citest.py" to enable sstate

✅ Test environment ready for use!

builder@9d38db016e13:/work/testsuite$ tlist citest.py
avocado-instrumented citest.py:EnvTest.test_nop
avocado-instrumented citest.py:EnvTest.test_bitbake
avocado-instrumented citest.py:EnvTest.test_build
avocado-instrumented citest.py:DevTest.test_dev
avocado-instrumented citest.py:DevTest.test_dev_apps
avocado-instrumented citest.py:DevTest.test_dev_rebuild
avocado-instrumented citest.py:DevTest.test_dev_run_amd64_bookworm
avocado-instrumented citest.py:DevTest.test_dev_run_arm64_bookworm
avocado-instrumented citest.py:DevTest.test_dev_run_arm_bookworm
avocado-instrumented citest.py:ReproTest.test_repro_signed
avocado-instrumented citest.py:ReproTest.test_repro_unsigned
avocado-instrumented citest.py:CcacheTest.test_ccache_rebuild
avocado-instrumented citest.py:InstallerTest.test_installer_build
avocado-instrumented citest.py:InstallerTest.test_installer_run
avocado-instrumented citest.py:InstallerTest.test_installer_root_partition
avocado-instrumented citest.py:CrossTest.test_cross
avocado-instrumented citest.py:CrossTest.test_cross_debsrc
avocado-instrumented citest.py:CrossTest.test_cross_trixie
avocado-instrumented citest.py:CrossTest.test_cross_kselftest
avocado-instrumented citest.py:CrossTest.test_cross_rpi
```

[PATCH] test: provide ci\_shell as a convenience wrapper

# Isar Test Suite

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

ci\_shell is a simple wrapper that:

- Uses kas-container to create a clean execution environment with avocado pre-installed
- Provides some aliases to avocado commands (should we move away from avocado!);
  - tlist citest.py → list available tests
  - trun citest.py:TestClass.test\_method → run a specific test
  - trun -p sstate=1 citest.py → enable sstate (Recommended!)

# Isar Test Suite

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

| Patch                                                                                                  | Series                                                                 | S/W/F | 📅 Date     | Submitter          |
|--------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|-------|------------|--------------------|
| isar-installer: skip inactive/invalid RAID devices in target detection                                 | isar-installer: skip inactive/invalid RAID devices in target detection | - - - | 2025-10-31 | Kasturi Shekar     |
| [v4,2/2] classes/image-postproc: Enable systemd units based on systemd presets                         | Enable systemd units based on systemd presets                          | - - - | 2025-10-30 | alexander.heinisch |
| [v4,1/2] Minor cleanup: Refactor ROOTFS_FEATURES assignment for image to make upcoming changes more... | Enable systemd units based on systemd presets                          | - - - | 2025-10-30 | alexander.heinisch |
| [v5,12/12] user_manual: Add dracut for initramfs generation                                            | Add support for dracut                                                 | - - - | 2025-10-30 | Quirin Gylstorff   |
| [v5,11/12] Use lighttpd as a example how to add a dracut module                                        | Add support for dracut                                                 | - - - | 2025-10-30 | Quirin Gylstorff   |
| [v5,10/12] Add dracut module helper                                                                    | Add support for dracut                                                 | - - - | 2025-10-30 | Quirin Gylstorff   |
| [v5,09/12] Add example dracut initramfs                                                                | Add support for dracut                                                 | - - - | 2025-10-30 | Quirin Gylstorff   |
| [v5,08/12] rootfs: add flag to use dracut if it is not part of the package list                        | Add support for dracut                                                 | - - - | 2025-10-30 | Quirin Gylstorff   |
| [v5,07/12] Add class to generate custom dracut initramfs                                               | Add support for dracut                                                 | - - - | 2025-10-30 | Quirin Gylstorff   |
| [v5,06/12] initramfs: allow to set the generator command                                               | Add support for dracut                                                 | - - - | 2025-10-30 | Quirin Gylstorff   |
| [v5,05/12] rootfs: Add dracut to initramfs generator                                                   | Add support for dracut                                                 | - - - | 2025-10-30 | Quirin Gylstorff   |
| [v5,04/12] rootfs: Copy the newly created initrd.img to the work directory                             | Add support for dracut                                                 | - - - | 2025-10-30 | Quirin Gylstorff   |
| [v5,03/12] rootfs: Add isar-work directory to rootfs mounts                                            | Add support for dracut                                                 | - - - | 2025-10-30 | Quirin Gylstorff   |
| [v5,02/12] rootfs: Allow to overwrite the initramfs generation cmds                                    | Add support for dracut                                                 | - - - | 2025-10-30 | Quirin Gylstorff   |
| [v5,01/12] Add dracut to custom kernel builds                                                          | Add support for dracut                                                 | - - - | 2025-10-30 | Quirin Gylstorff   |

Awesome patch series (I have been advocating for Dracut for ages!)

BUT no ci\_tests added ☹️

(sorry Quirin didn't mean to pick on you, I happened to be testing your v5 while working on these slides)

Virtual Lab: Can we help Quirin?

# Isar Test Suite

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

```
class InitRdTests(CIBaseTest):
    """
    Tests for initrd generation
    :avocado: tags=initrd,full
    """

    DRACUT_CONF = [
        "IMAGE_PREINSTALL:append = `dracut'",
        "IMAGE_INSTALL:remove = `enable-fsck'"
    ]

    def test_dracut_in_image(self):
        """Test switch to dracut in an image recipe."""

        targets = ['mc:qemuamd64-bookworm:isar-image-ci']
        self.init()
        self.perform_build_test(targets, lines=InitRdTests.DRACUT_CONF)
```

You may start with a simple build test. Use `lines=` to add “stuff” to `local.conf`

Verify with: `trun -p sstate=1 citest.py:InitRdTests.test_dracut_in_image`

Consider adding cross tests (similar `test_()` method, different target(s))

# Isar Test Suite

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

Additional hints:

- Use `self.perform_build_test(targets, should_fail=True)` if you expect bitbake to fail
- Use `self.exec_cmd()` to run arbitrary commands. Note that:
  - 1<sup>st</sup> positional arg is the (last) command argument, 2<sup>nd</sup> is the command  
e.g. `self.exec_cmd("mc:qemuamd64-bookworm:isar-image-ci", "bitbake -g")`
  - Returns `status, stdout, stderr`

# Isar Test Suite

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

```
class InitRdTests(CIBaseTest):
    ...

    def test_dracut_in_image(self):
        """Test switch to dracut in an image recipe."""

        machine = 'qemuamd64'
        distro = 'bookworm'
        image = 'isar-image-ci'
        targets = [f'mc:{machine}-{distro}:{image}']
        self.init()
        self.perform_build_test(targets, lines=InitRdTests.DRACUT_CONF)

        self.vm_start(
            machine.removeprefix('qemu'), distro, image=image,
            cmd="systemctl is-active dracut-shutdown"
        )
```

This will boot the generated image using QEMU and use ssh to run the specified command.

# Isar Test Suite

Meet Cedric!  
Agenda  
Binary vs Source Distributions  
Our Challenges  
Testing

- Restructure!
  - `citest.py` should not be the recipient for all tests
  - We instead want a `.py` for each theme (`sstate`, `cross`, `kernel`, `initrd`, etc.)
- Test Dependencies
  - Use avocado dependencies for proper ordering and skipping
  - Switch to another framework?
- Smaller test image?
  - `isar-image-ci` building way more than most tests require (e.g. use Debian's kernel instead of building our own for amd64?)
- Add “fast” tests leveraging `bitbake -g`  
Potential applicability:
  - `DPKG_ARCH=all` build dependencies
- Change prototype of `exec_cmd()` to be more conventional