

Eclipse oniro

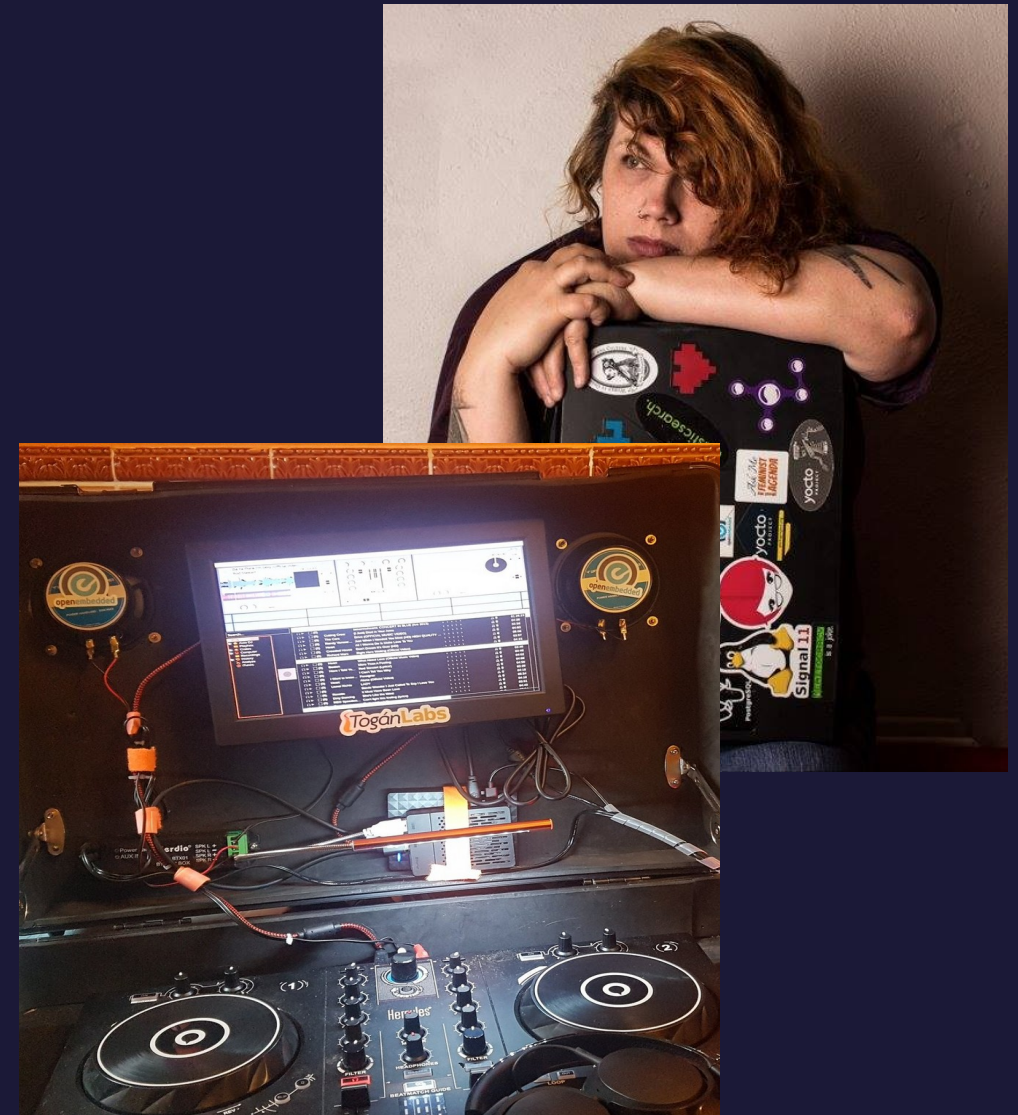
Build Systems All The Way Down
Untangling OpenEmbedded, meta-zephyr, and
zephyr integration in Eclipse Oniro

Eilís 'pidge' Ní Fhlannagáin
Principal Engineer – Huawei
FOSDEM 2022



▶ Who Am I?

- **Software Developer for 30 years**
 - **Initial Linux Install Slackware circa 1994ish**
- **Principal Open Source Architect at Huawei**
- **Embedded Systems the past 20 years**
 - **OpenEmbedded/Yocto Project since 2011**
- **Maker of funny demos**
 - **Yocto Blimp**
 - **Yocto Portable DJ**
 - **Embedded Hurdy Gurdy**
- **Lead the team responsible for the initial work on meta-zephyr**



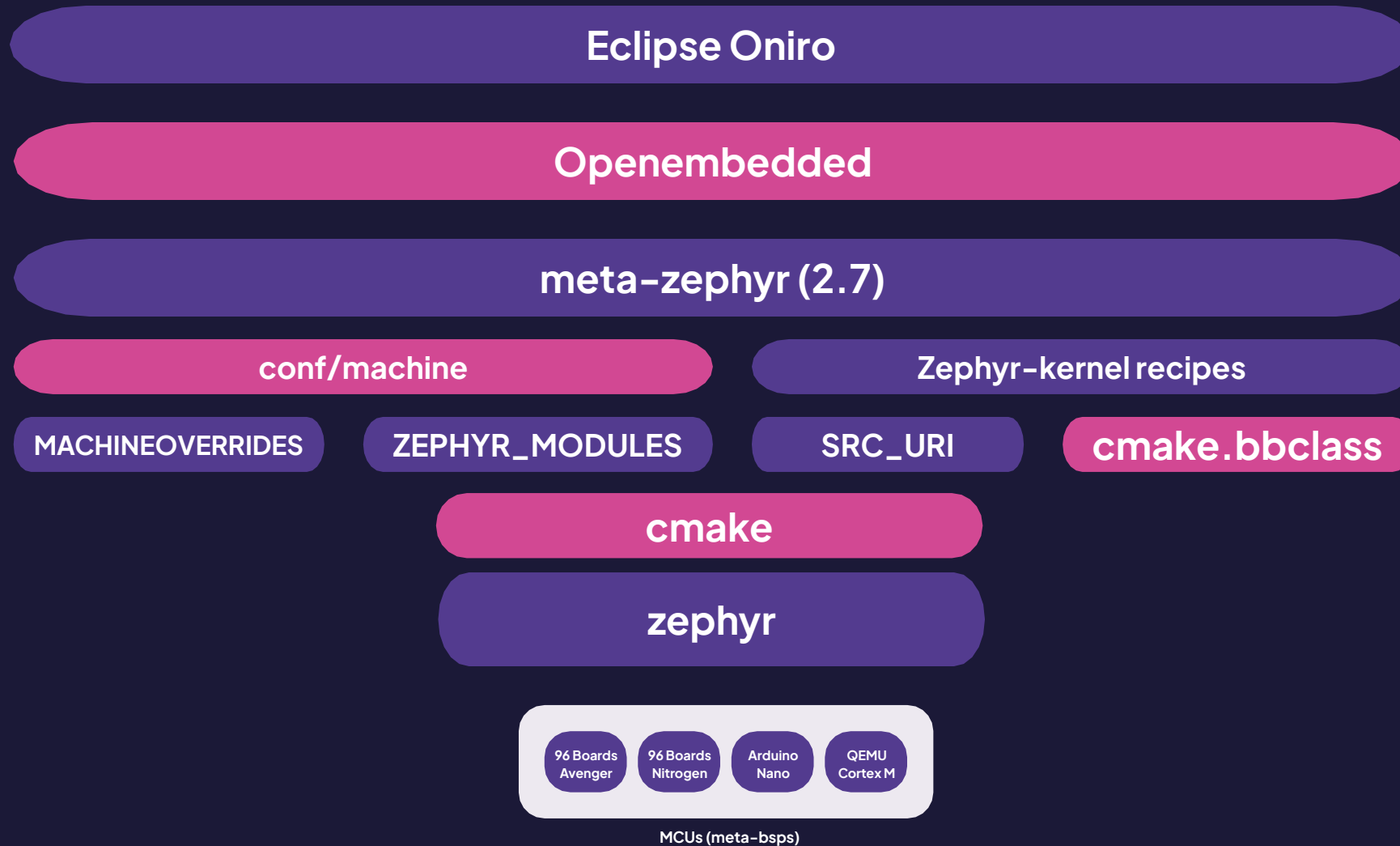
▶ Eclipse Oniro Project

- Focused on the development of a distributed open source operating system for consumer devices, regardless of the brand, model, make.
- OpenEmbedded + Yocto Project + Zephyr
- Blueprints, Documentation

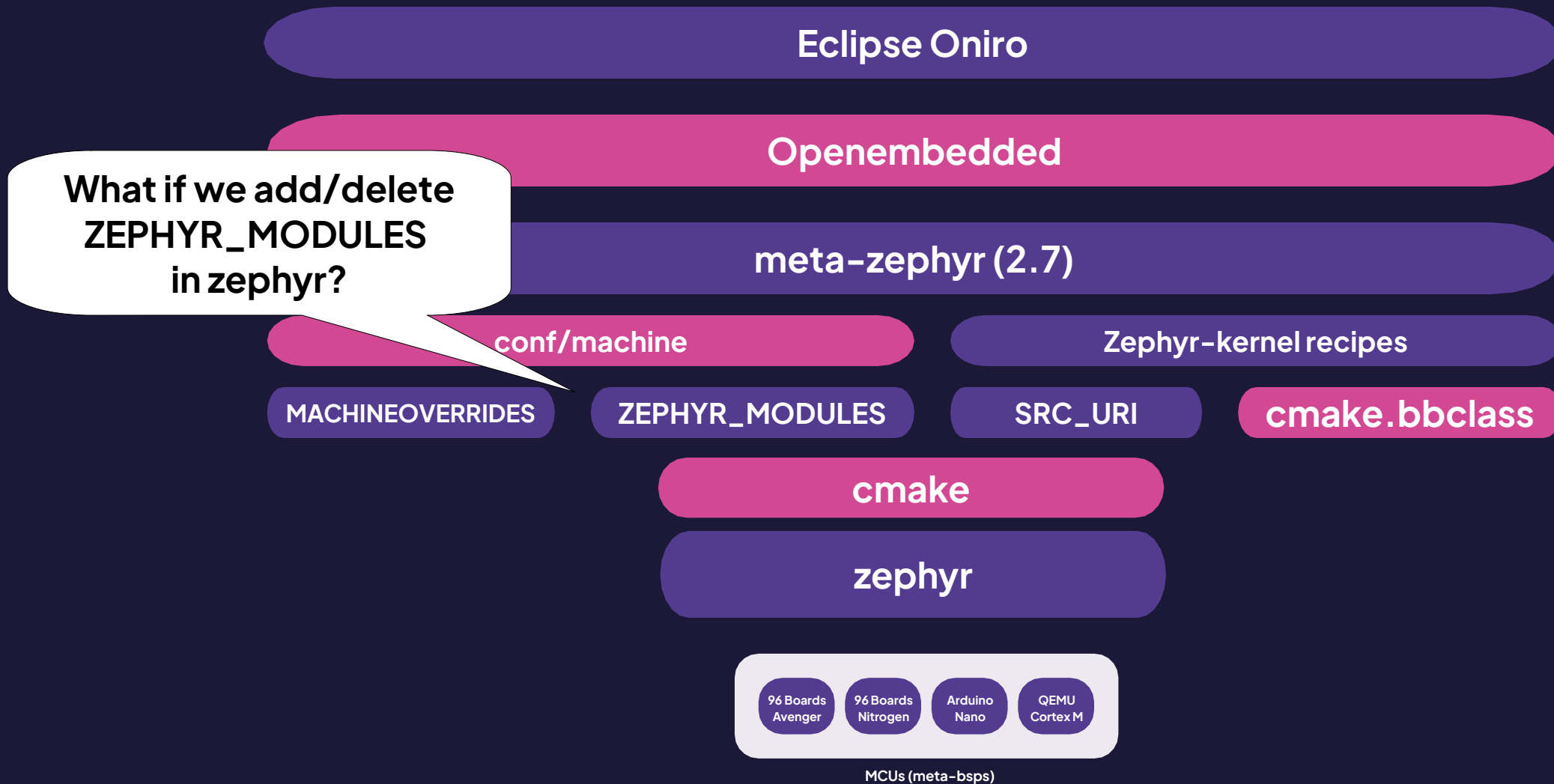


oniroTM

▶ Openembedded/zephyr stack

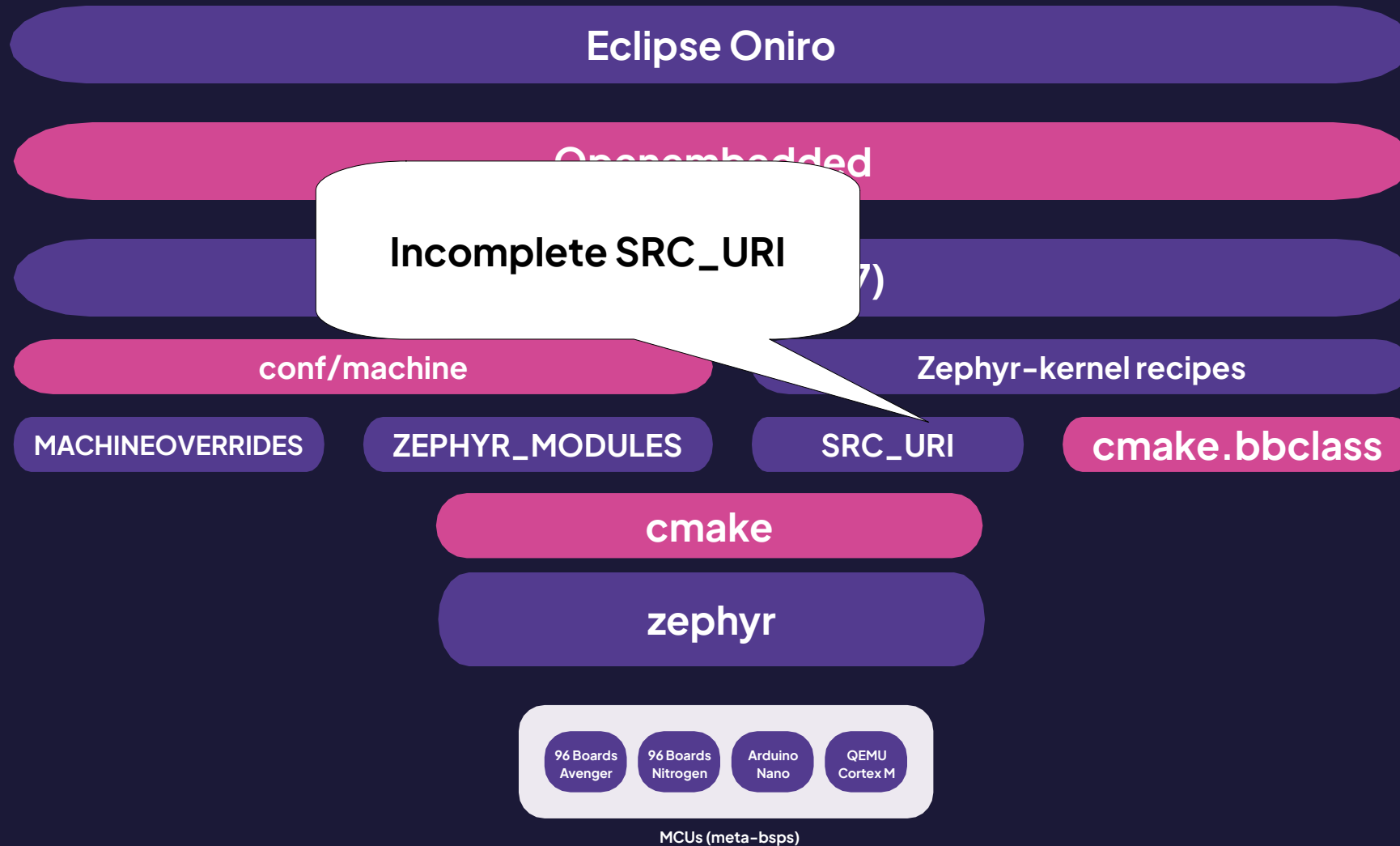


▶ Openembedded/zephyr stack



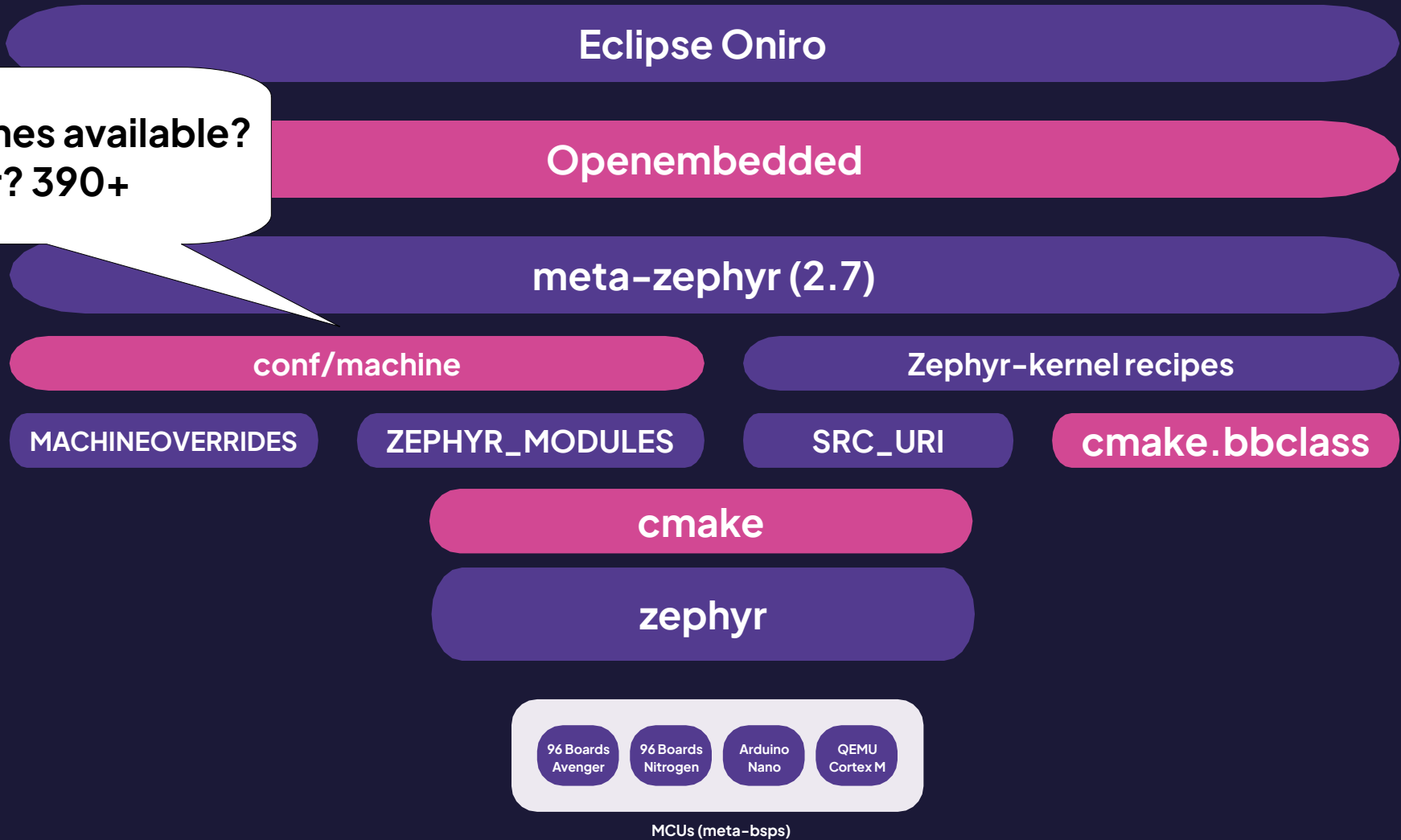
What if we add/delete ZEPHYR_MODULES in zephyr?

▶ Openembedded/zephyr stack



▶ Openembedded/zephyr stack

Only 8 Machines available?
Zephyr? 390+



▶ Openembedded/zephyr stack

Layer structure is arguably
Not YP compatible

Eclipse Oniro

Openembedded

meta-zephyr (2.7)

conf/machine

Zephyr-kernel recipes

MACHINEOVERRIDES

ZEPHYR_MODULES

SRC_URI

cmake.bbclass

cmake

zephyr

96 Boards
Avenger

96 Boards
Nitrogen

Arduino
Nano

QEMU
Cortex M

MCUs (meta-bsps)

▶ meta-zephyr issues

- **Treats zephyr projects as cmake projects**
 - **Which they are....**
 - **But. West....**
- **No knowledge of ZEPHYR_MODULES unless told**
 - **MACHINEOVERRIDES is used for this**
 - **Really shouldn't be in a machine definition**
 - **Zephyr should really take care of this**
 - **ZEPHYR_EXTRA_MODULES**
- **8 supported machines**
 - **Out of 300+ Zephyr Machines**
 - **How to support as many as possible?**
- **Layer structure is technically not Yocto Project Compatible**

► Premises

- Machine BSP layers should stand alone
 - Yocto Project Compatibility is clear about this

“Where multiple types of functionality are present, the layer should be internally divided into sublayers to separate these components. That’s because some users may only need one of them and separability is a key best practice.”

- Zephyr should control as much about the compilation as possible
 - ZEPHYR_EXTRA_MODULES at the OE layer is fine
 - Needing to replicate metadata should be avoided
- More Machines supported
 - Means full Zephyr checkout
 - How to get as many of those machines as possible out of zephyr

▶ Getting Rid of MACHINEOVERRIDES

- **Could read Kconfig, etc.**
 - **Lot of work to build/maintain**
- **Could autogenerate MACHINEOVERRIDES from Kconfigs**
 - **Went down this path at first**
 - **Machine definitions shouldn't really contain this**
 - **Lot of work to build/maintain**
 - **Doesn't solve the problem**
- **zephyr/module.yml?**
 - **zephyr_module.py?**
 - **But it requires west?**
 - **Points to the solution**

▶ Getting Rid of MACHINEOVERRIDES

- **west is interesting**
 - Documented way to get/build zephyr
 - Swiss pocket-knife (wraps git, cmake, manifests, etc)
 - Plugin framework to extend
 - **west list** gives us what we want
- **Ensure our checkout complies with west workspace**
- **west-native DEPENDS**
- **do_get_zmods**
 - **west list|awk 'NR>1 {print \$2}'** to get ZEPHYR_MODULES
 - **nostamp**
 - **Allows us to let OE discover what modules and pass in -DZEPHYR_MODULES**
 - **Pass in extra modules with -DZEPHYR_EXTRA_MODULES**

▶ Abusing Cmake exports to generate machine configs

- This stumped me for a bit
 - Not a cmake expert
- `cmake/makefile_exports/CmakeLists.txt`
 - Looks interesting?
- `cmake/oe_exports/CmakeLists.txt`
 - Not perfect
 - Missing tunes in some machines (esp32/riscv)
- Results:
 - 395 zephyr machines
 - 323 generated somewhat correctly
 - 291 will produce a `zephyr-helloworld.elf`

▶ Autogenerated machine issues

- Does solving the problem causes others problems?
 - Vendors have their own BSP layers!
- This fixes my problem, can/does it fix everyones?
 - Work with stakeholders
- Not all machines make sense
 - Posix* doesn't really make a lot of sense

▶ Openembedded/zephyr stack post-changes

Eclipse Oniro

Openembedded

meta-zephyr-bsp (2.7)

conf/machine

generate-zephyr-machines.bb

Optional 283 Additional Machines

Reduced metadata replication

meta-zephyr-core (2.7)

zephyr.bbclass

Zephyr-kernel recipes

west

Full SRC_URI

ZEPHYR_MODULES

cmake

zephyr

96 Boards
Avenger

96 Boards
Nitrogen

Arduino
Nano

QEMU
Cortex M

MCUs (meta-bsps)

► Questions?

- Eclipse Oniro Project
 - <https://oniroproject.org/>
 - #oniroproject channel at libera.chat
- Oniro Project Documentation
 - <https://docs.oniroproject.org/en/latest/>
- meta-zephyr
 - <https://git.yoctoproject.org/meta-zephyr>