Multiconfig Inception

Yocto Project Summit 2021

Joshua Watt
JPEWhacker@gmail.com
IRC: JPEW

Inception

1. The establishment or starting point of an institution or activity
2. The act of instilling an idea into someone's mind by entering his or her dreams

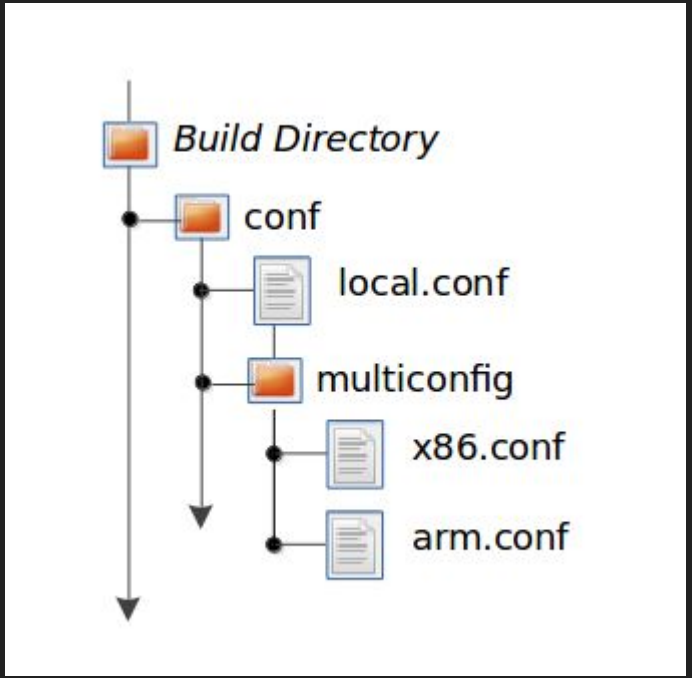
Also a 2010 Christopher Nolan Movie

What is Multiconfig?

- Multiconfig is a tool built into bitbake that allows multiple different configurations to be built simultaneously
  - It is sort of like having multiple local.conf files
- http://docs.yoctoproject.org/dev-manual/common-tasks.html#building-images-for-multiple-targets-using-multiple-configurations

Multiconfig Management

- Multiconfigs are defined by `.conf` files in a `conf/multiconfig` directory in either the build directory, or a layer
- Multiconfigs are enabled (selected) by `BBMULTICONFIG` in `local.conf`
- Bitbake will parse the recipes N + 1 times (once for each multiconfig + the base)



Multiconfig Dependencies

```
do_install[mcdepends] = "mc:cobb:arthur:arthur-image:do_image_complete"
```

Task to which the dependency applies

"from multiconfig" - The multiconfig to which the dependency to applies

"to multiconfig" - The multiconfig on which the "from multiconfig" depends

The recipe in the "from multiconfig" on which the "to multiconfig" will depend

The task in the "from multiconfig" on which the "to multiconfig" will depend

Multiconfig Dependencies

```
do_install[mcdepends] = "mc:cobb:arthur:arthur-image:do_image_complete"
```

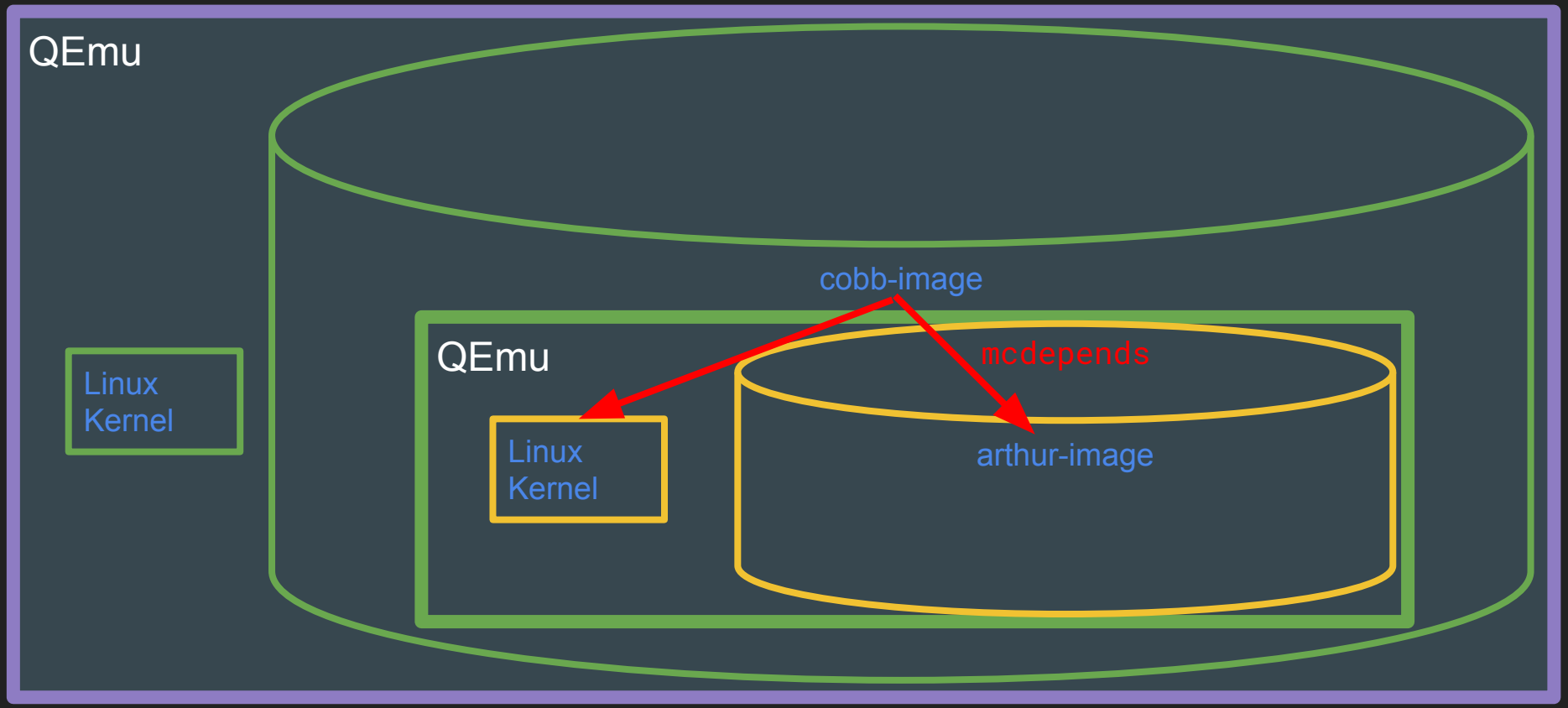
"When the current multiconfig is `cobb`, the `do_install` task depends on the task `do_image_complete` from the `arthur-image` recipe in the `arthur` multiconfig"

Multiconfig Dependencies

- The empty multiconfig ("") refers to the base configuration (e.g. no multiconfig):
  - `do_install[mcdepends] = "mc::arthur:arthur-image:do_image_complete"`

- ${BB_CURRENT_MC} refers to the current multiconfig (except the base…)
  - `do_install[mcdepends] = "mc:${BB_CURRENT_MC}:arthur:arthur-image:do_image_complete"`

QEmu within QEmu



Building

```
# Start with a new shell!

# Setup Environment
$ source ./yp-summit-may-21/poky/oe-init-build-env \
    ~/yp-summit-may-21/poky/build-config

# Clone down the demo layer
$ git clone https://github.com/JPEWdev/meta-multiconfig-demos.git \
    ../meta-multiconfig-demos

# Add layer to configuration
$ echo "BBLAYERS += \"$(pwd)/../meta-multiconfig-demos\"" >> \
    conf/bblayers.conf

# Enable multiconfig
$ echo "BBMULTICONFIG += \"cobb arthur\"" >> conf/local.conf

# Build
$ bitbake mc:cobb:cobb-image
```

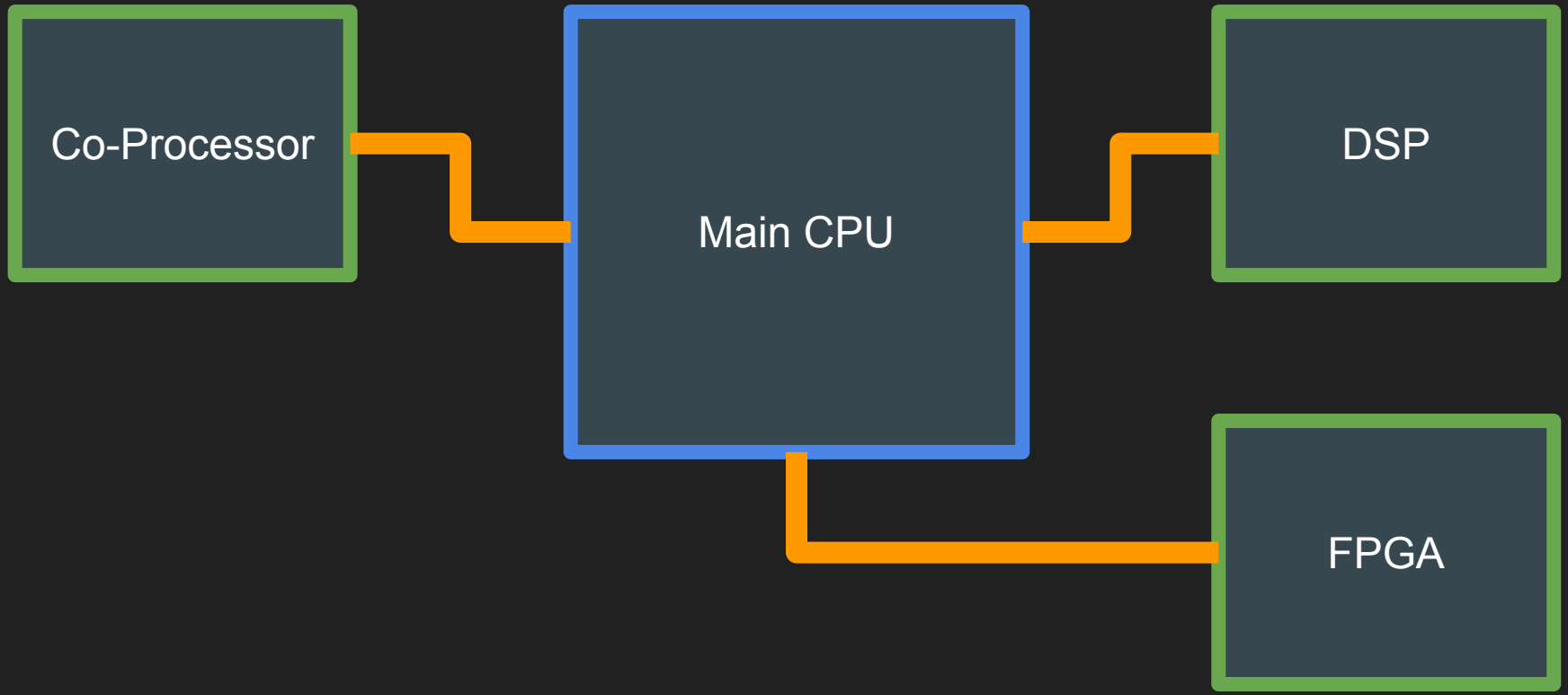
Recursive "Dreaming"

```
$ MULTICONFIG=cobb runqemu nographic serialstdio slirp kvm
$ cat /etc/dreamer.conf
cobb
Sweet dreams!

$ run-arthur
$ cat /etc/dreamer.conf
arthur
Sweet dreams!

# Exit Arthur's dream
$ halt
```

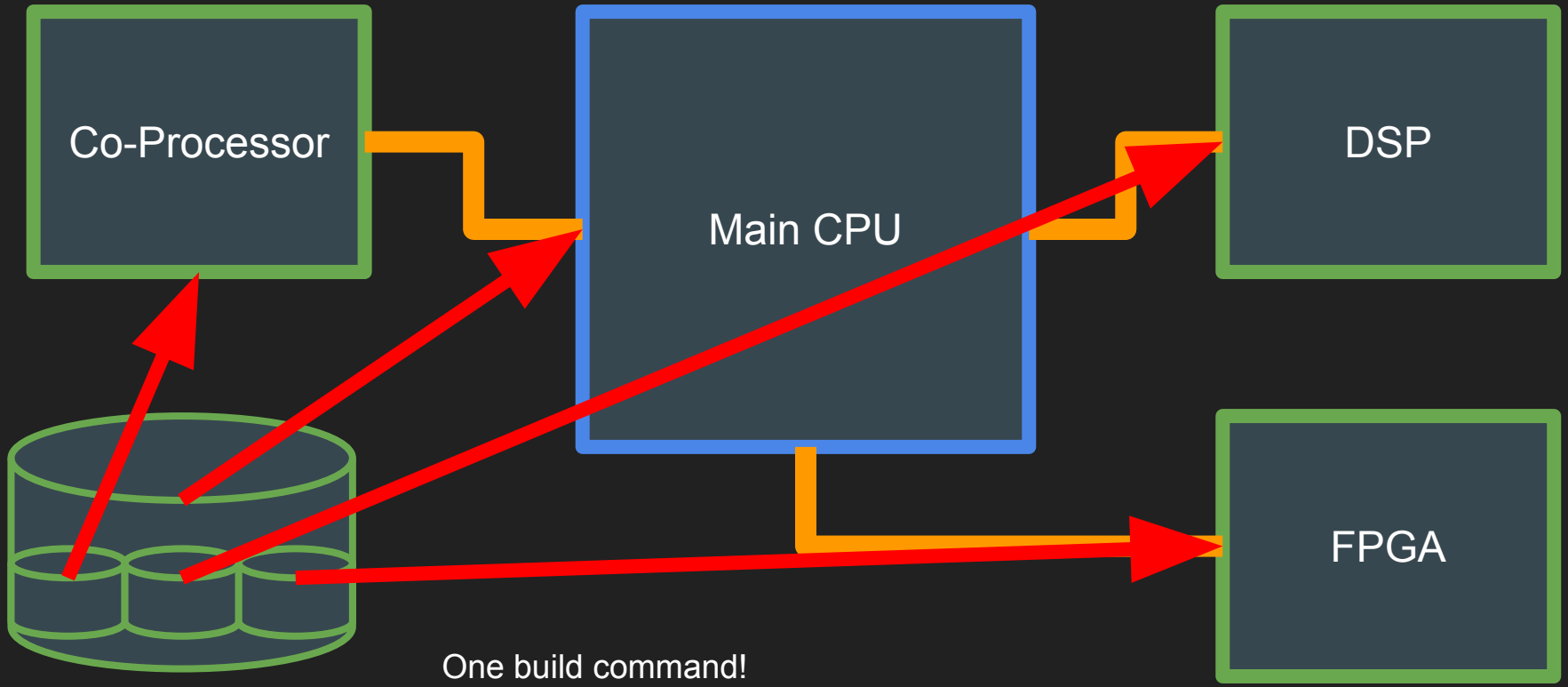
Why Do This?

- Build Optimization
  - sstate and download caches are shared, so one invocation of bitbake maybe more efficient than invoking it several different times for different configurations
- Complex Dependencies
  - Multiconfigs can depend on each other which allows complex systems to be constructed with a single bitbake invocation
- Configuration Management
  - Multiconfigs can be used as "pre-canned" local.conf equivalents, and can be shipped with a layer
  - Users can still keep their local.conf

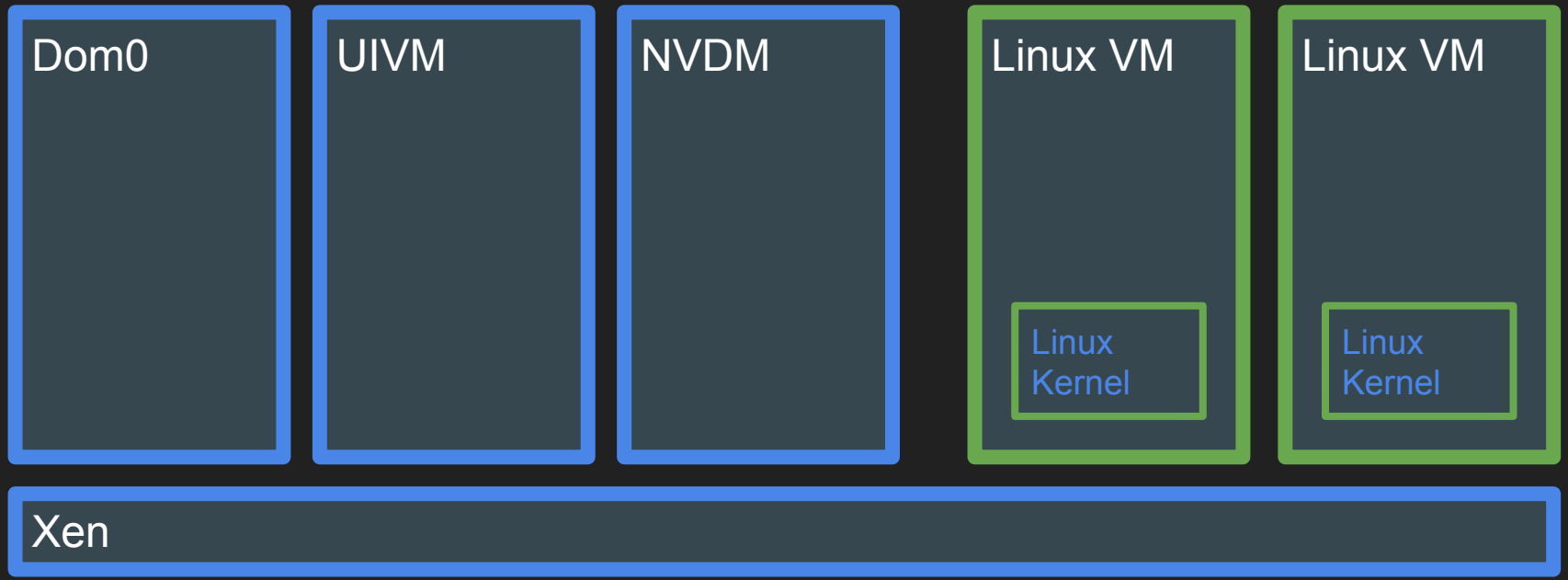
Embedded System Partitioning with Multiconfig



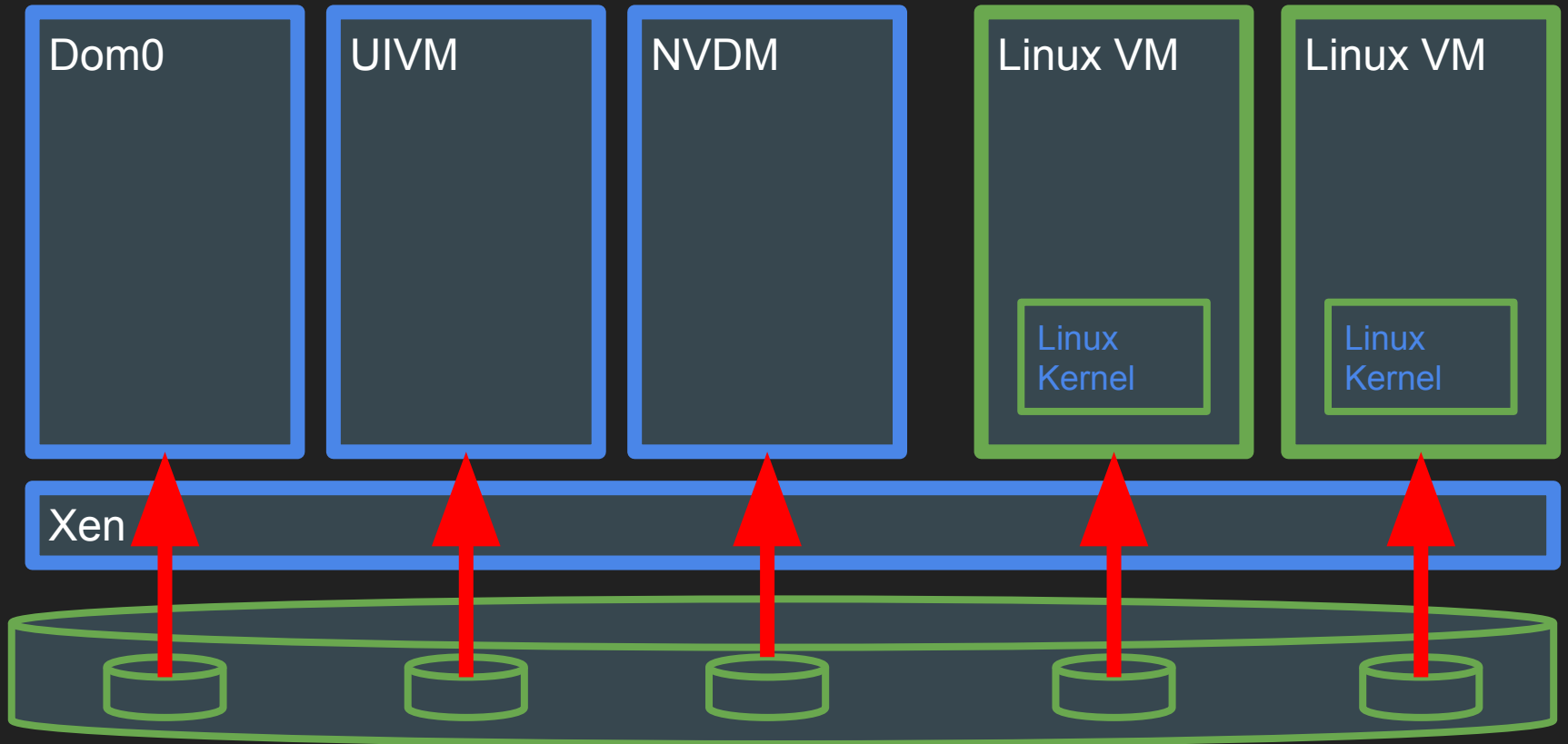
Embedded System Partitioning with Multiconfig



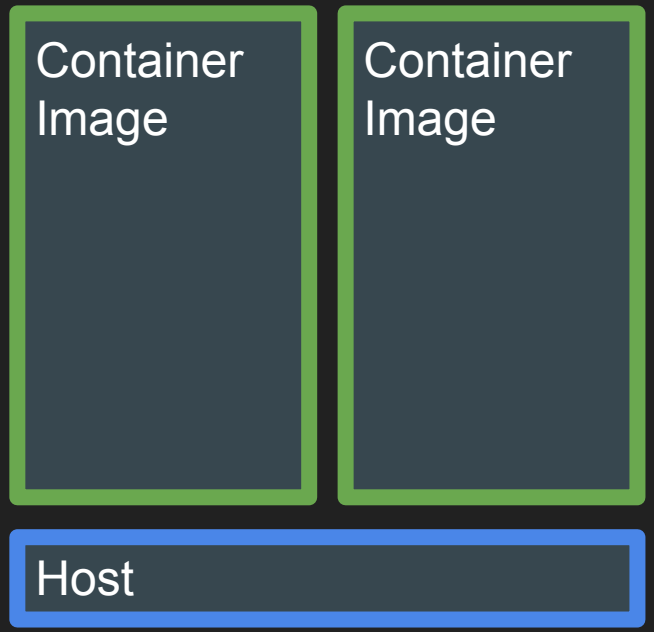
Hypervisor with Multiconfig



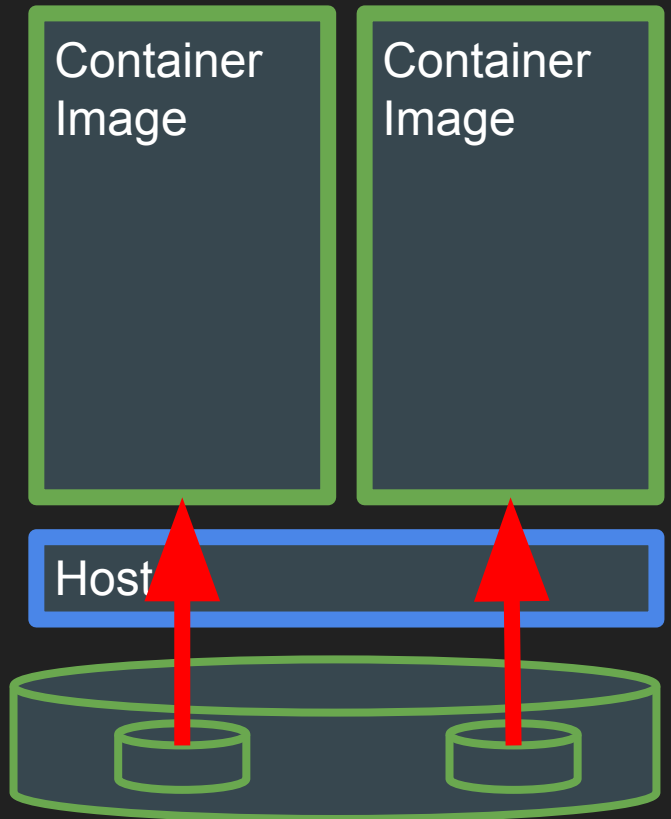
Hypervisor with Multiconfig



Containers with Multiconfig



Containers with Multiconfig



Configuration Management with Multiconfig

cobb.conf:

```
DISTRO="poky"
MACHINE="qemux86"
```

saito.conf:

```
DISTRO="pokyalt"
MACHINE="qemuarm"
```

arthur.conf

```
DISTRO="poky"
MACHINE="qemuarm"
INIT_MANAGER="systemd"
```

- Multiconfigs apply after local.conf, so these configurations are effectively "fixed"
- https://github.com/garmin/whisk (shameless plug)

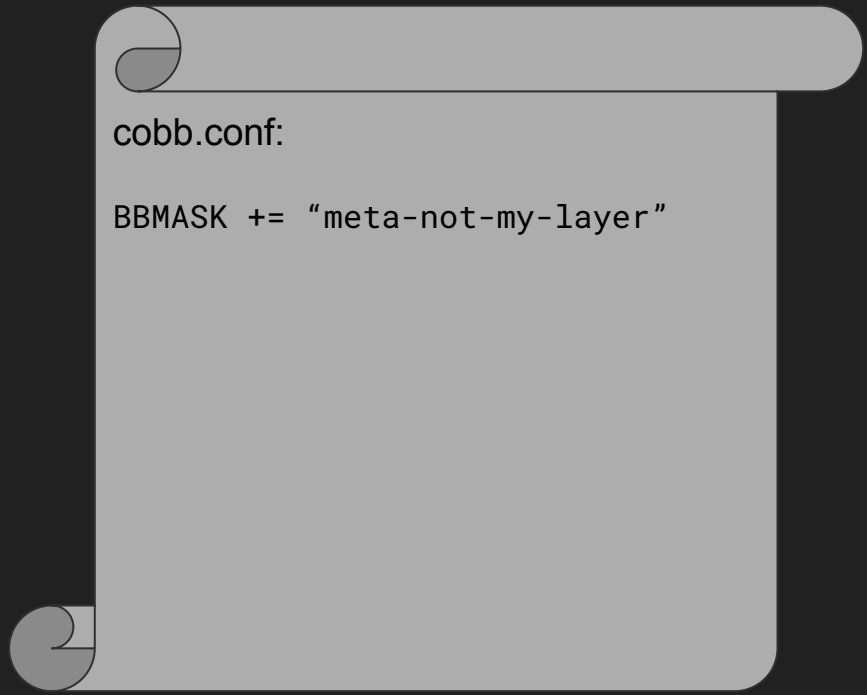
Configuration Management with Multiconfig (continued)

https://docs.yoctoproject.org/ref-manual/variables.html#term-BBMASK

BBMASK is respected per-multiconfig.

Different multiconfigs can have visibility to different sets of layers and/or recipes

Allows you to build multiconfigs with layers that may have otherwise conflicted at the same time.



BBMASK per Multiconfig

```
$ echo 'BBMASK += "dreamer.bbappend"' >> \
    ../meta-multiconfig-demos/conf/multiconfig/cobb.conf
$ bitbake mc:cobb:cobb-image
$ MULTICONFIG=cobb runqemu nographic serialstdio slirp kvm
$ cat /etc/dreamer.conf
cobb

$ run-arthur
$ cat /etc/dreamer.conf
arthur
Sweet dreams!
```

Sharing TMPDIR between multiconfigs

- Assign each multiconfig its own `${TMPDIR}`
- It is possible to have multiple share the same `${TMPDIR}`, but misconfigured recipes will cause problems
  - `${PACKAGE_ARCH}` must be set correctly
- Note that all multiconfigs will share a `${TMPDIR}` by default

Multiconfig Tips

- Dumping bitbake environment with multiconfig:
  - `bitbake -e` *`recipe`* `-> bitbake -e mc:`*`name`*`:`*`recipe`*
  - `bitbake -e  -> bitbake -e mc:`*`name`*
- Use a known `${DEPLOY_DIR}` to share dependencies

Questions?