OpenPOWER™

OpenPOWER firmware porting

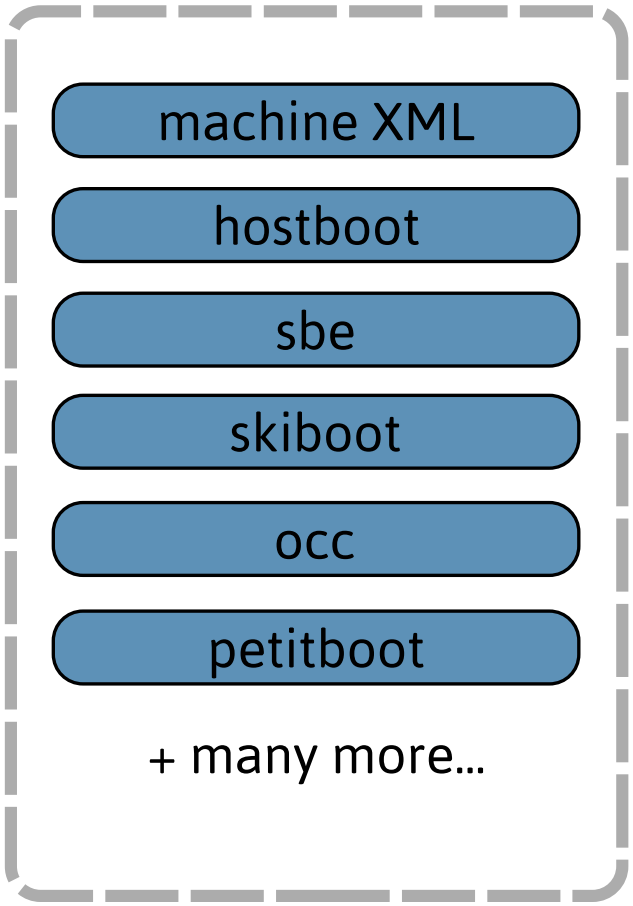
Bringing-up new hardware quickly and easily

Jeremy Kerr, OpenPOWER platform architect

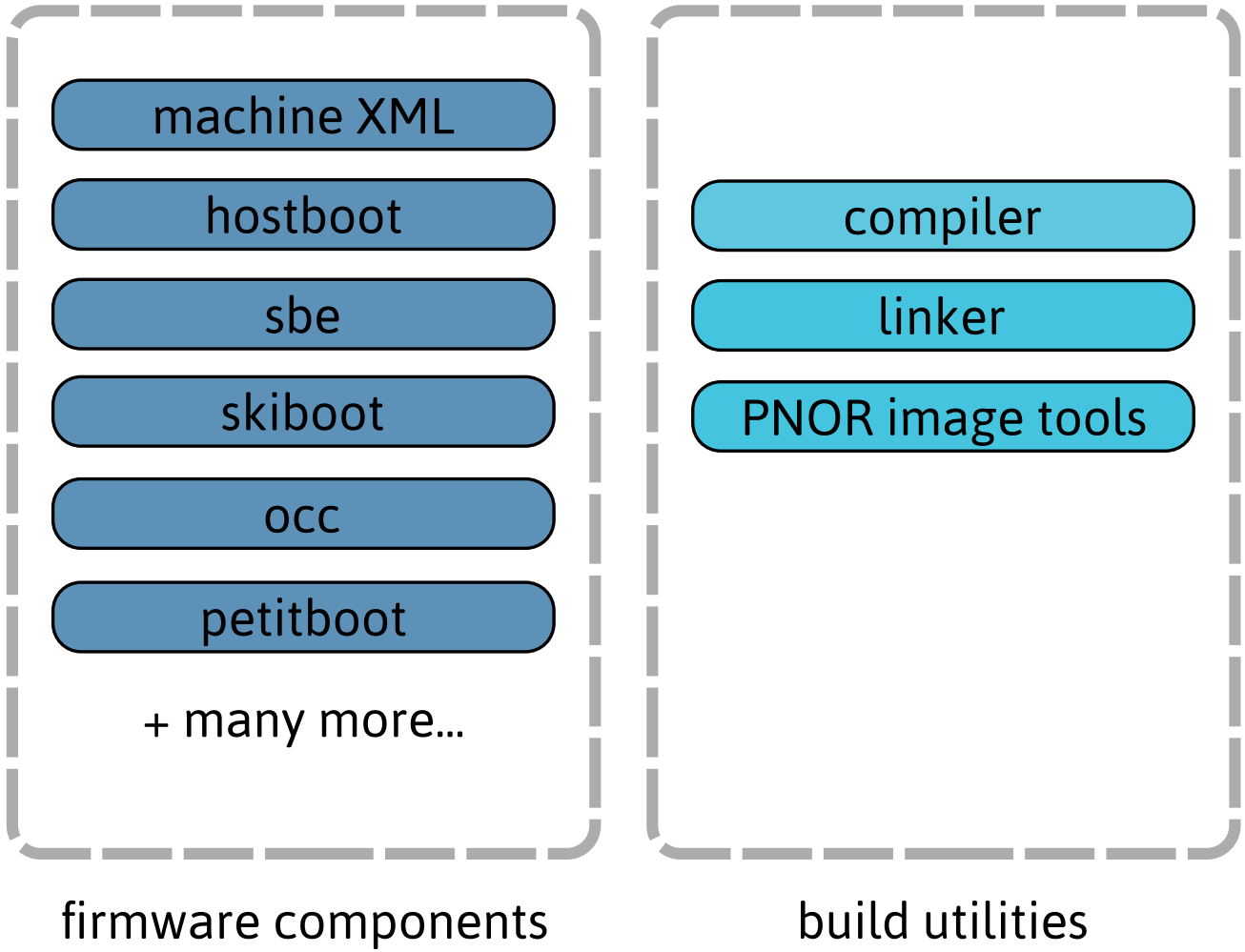
IBM Australia

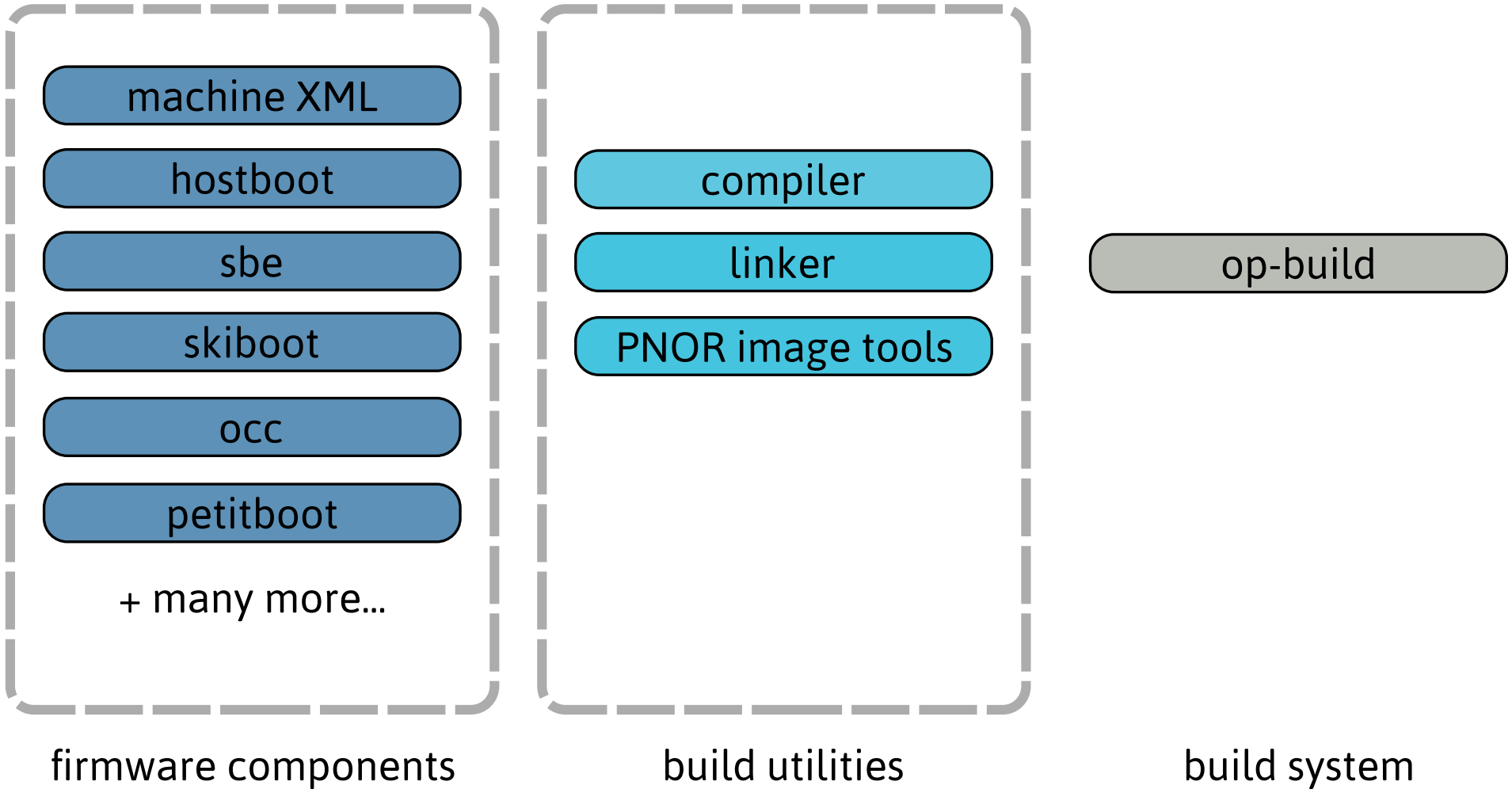
OpenPOWER™

https://github.com/open-power/

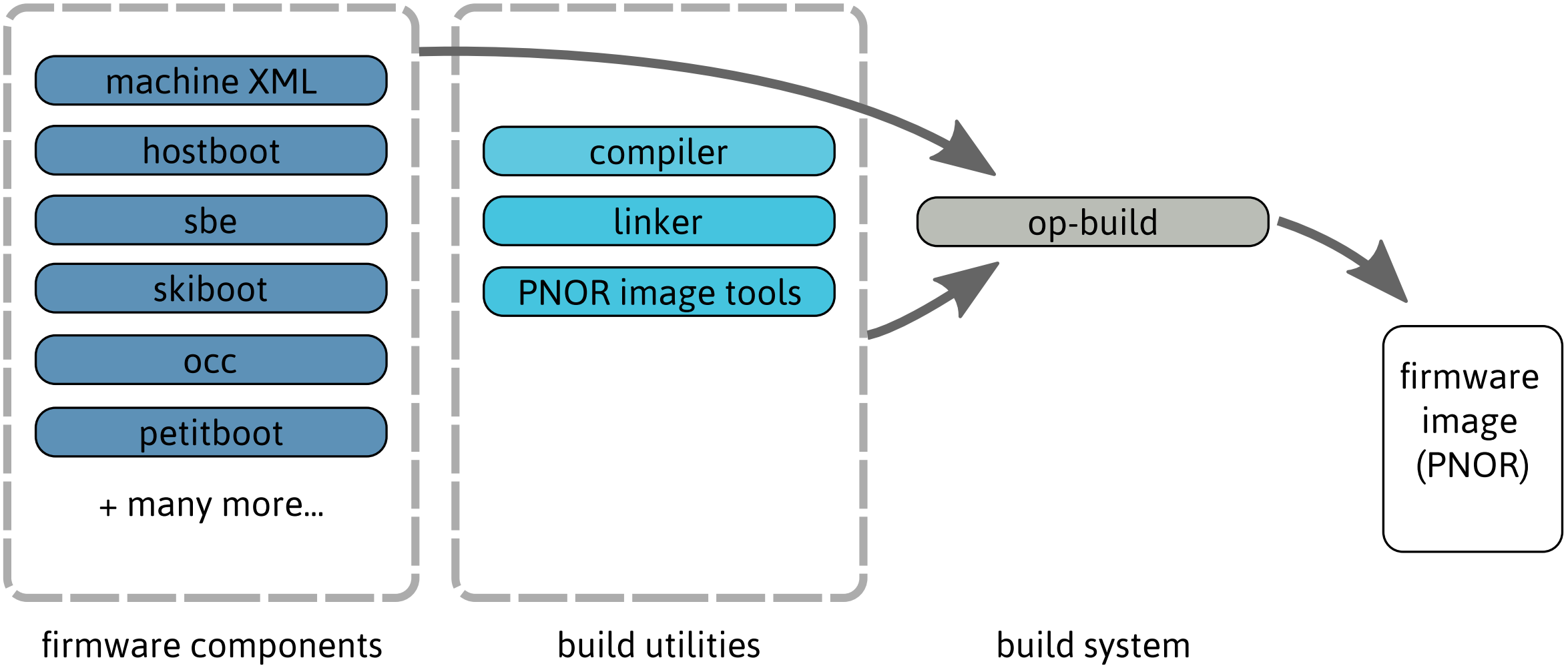


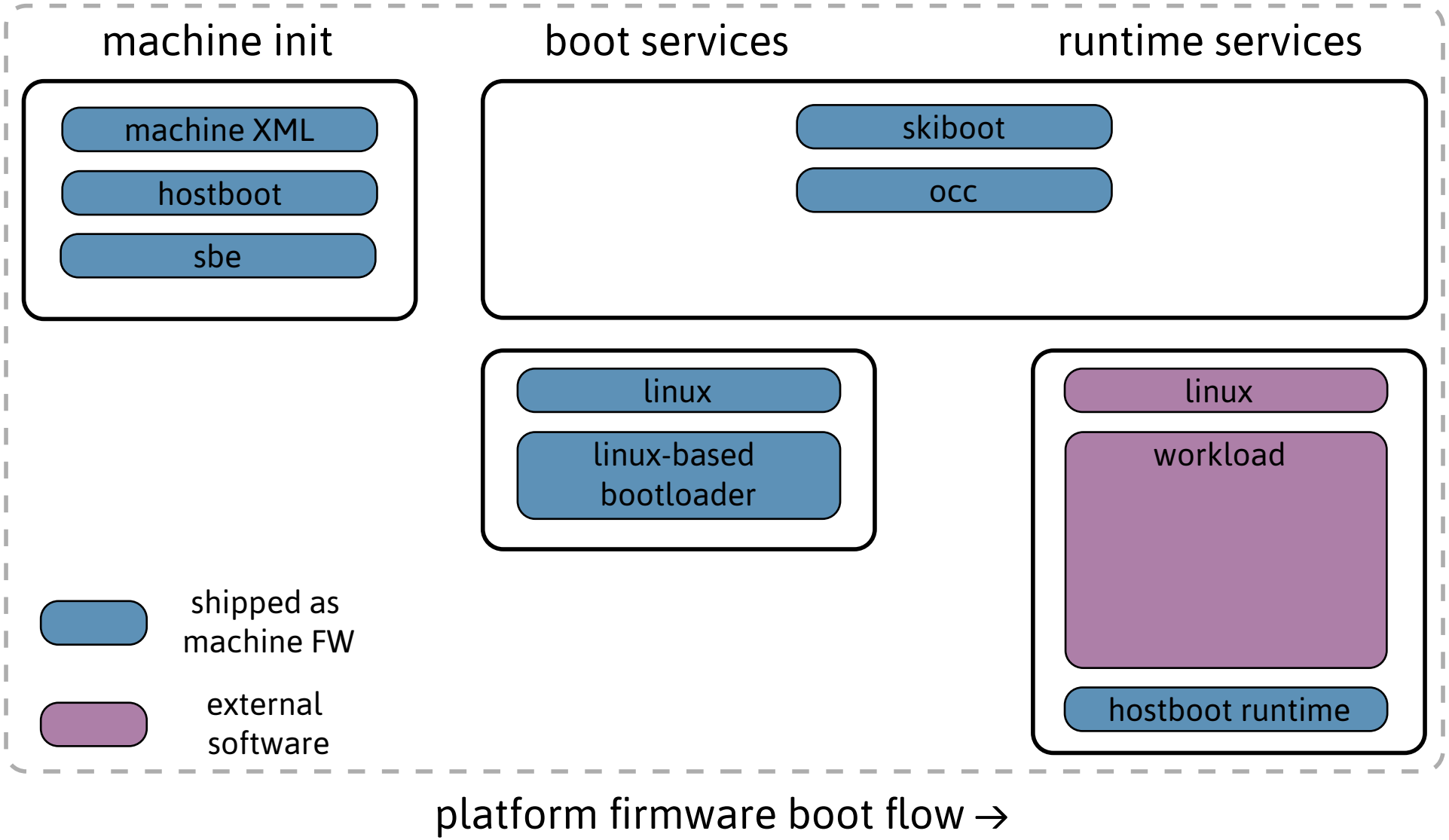
firmware components

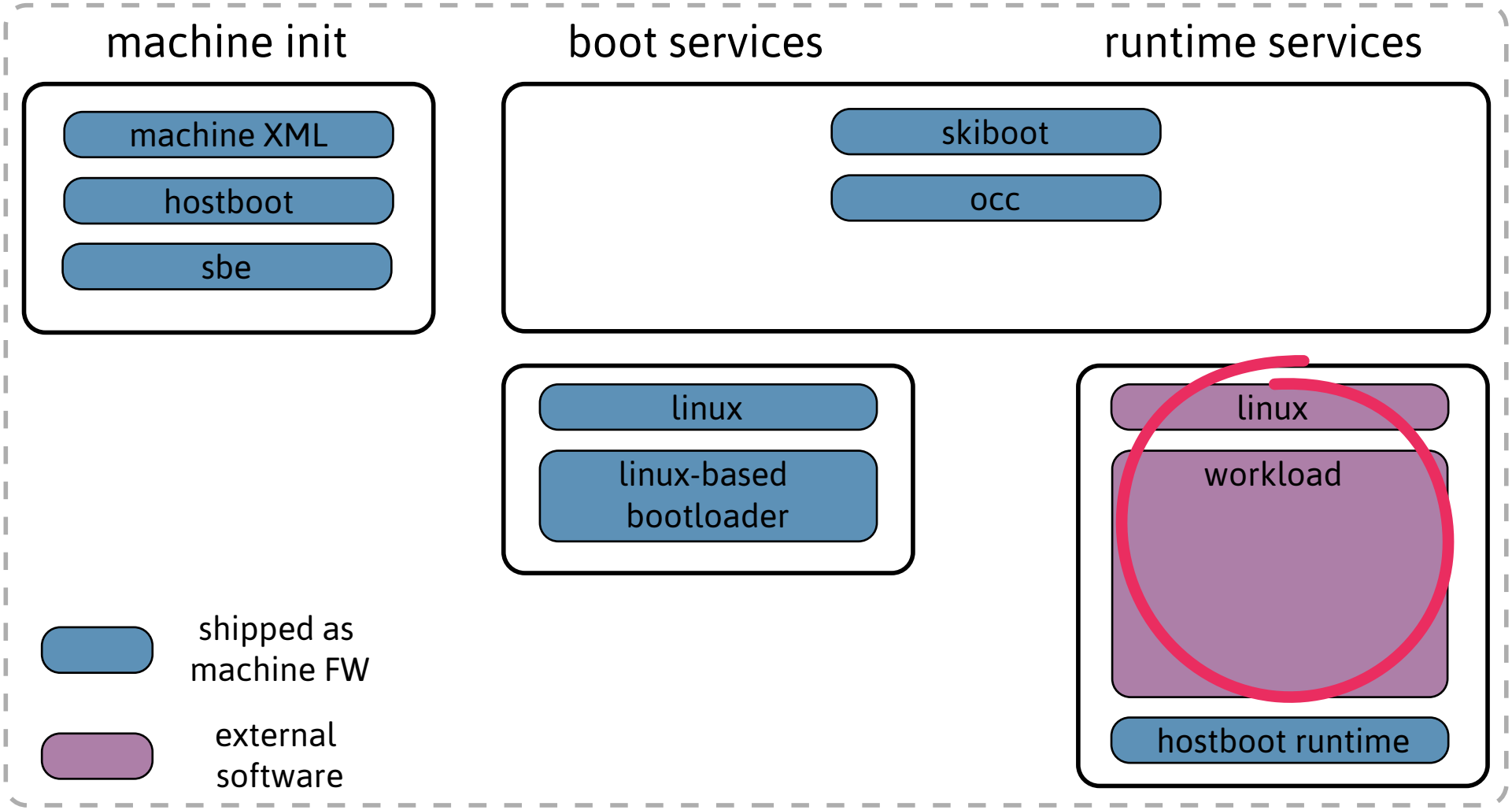

machine XML
hostboot
sbe
skiboot
occ
petitboot
+ many more...
compiler
linker
PNOR image tools
firmware components
build utilities

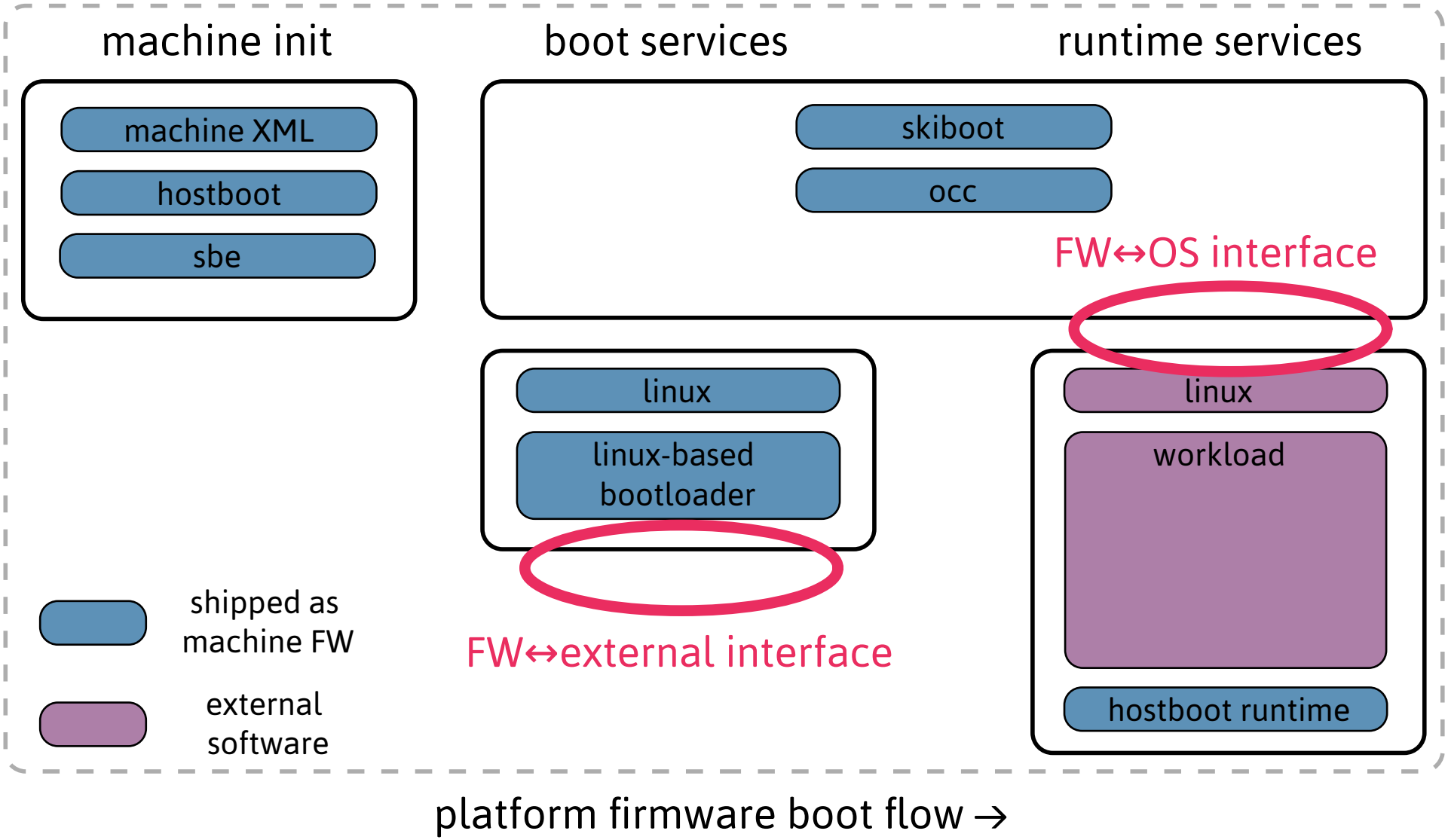

machine XML
hostboot
sbe
skiboot
occ
petitboot
+ many more...
compiler
linker
PNOR image tools
op-build
firmware components
build utilities
build system

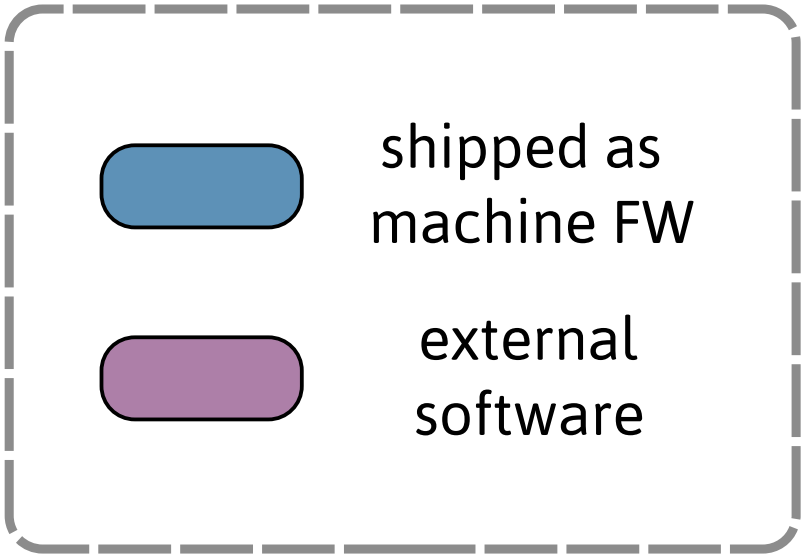
OpenPOWER™

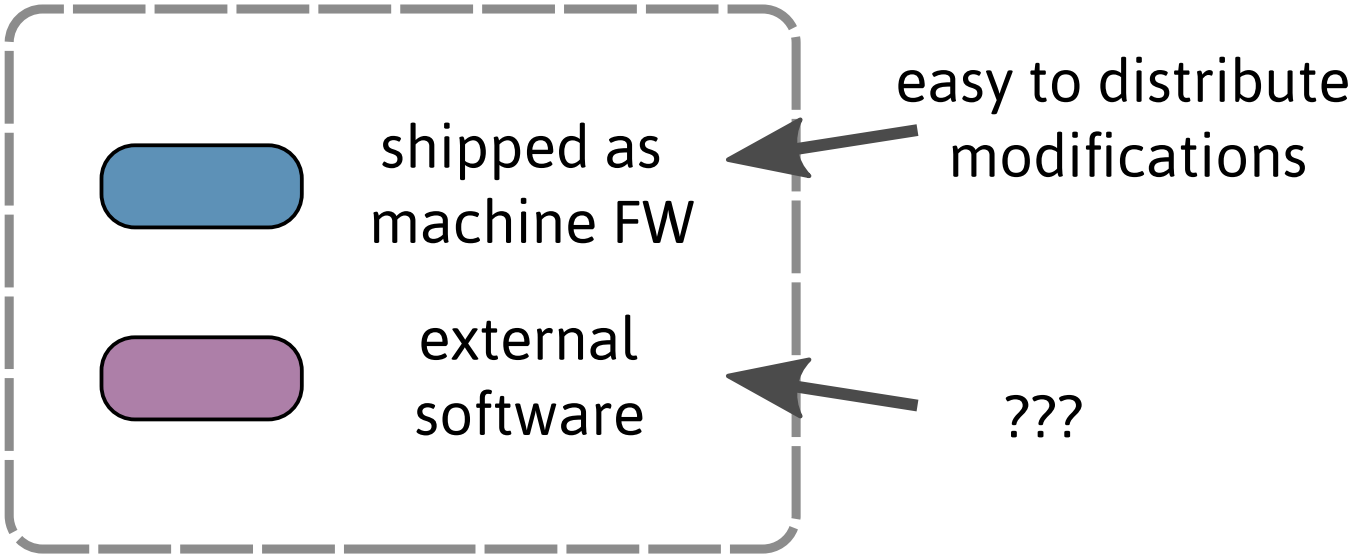

machine XML
hostboot
sbe
skiboot
occ
petitboot
+ many more...
compiler
linker
PNOR image tools
op-build
firmware image (PNOR)
firmware components
build utilities
build system

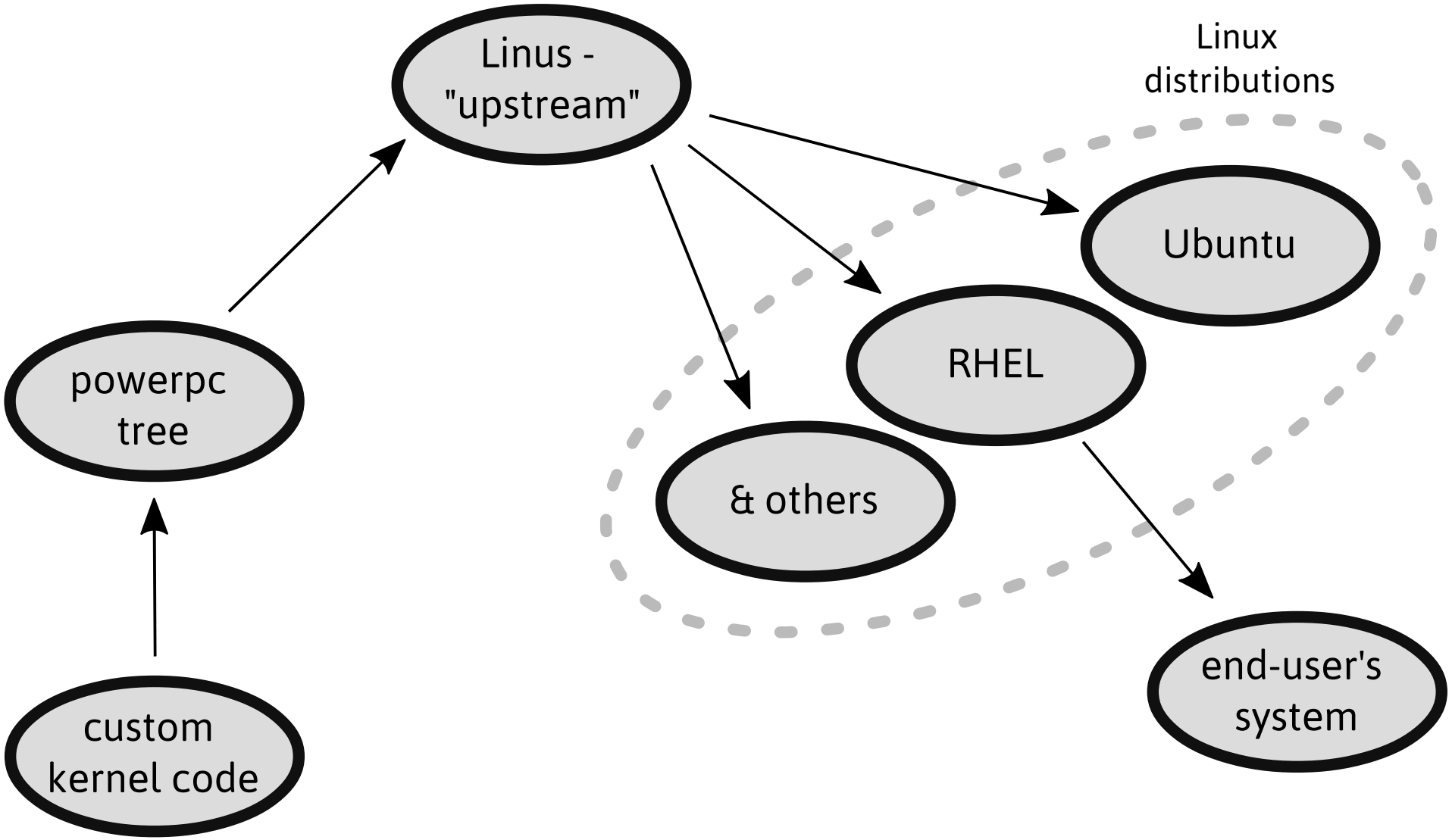

machine init
boot services
runtime services
machine XML
hostboot
sbe
skiboot
occ
linux
linux-based bootloader
linux
workload
hostboot runtime
shipped as machine FW
external software
platform firmware boot flow →


machine init
boot services
runtime services
machine XML
hostboot
sbe
skiboot
occ
linux
linux-based bootloader
linux
workload
hostboot runtime
shipped as machine FW
external software
platform firmware boot flow →


machine init
boot services
runtime services
machine XML
hostboot
sbe
skiboot
occ
FW↔OS interface
linux
linux-based bootloader
linux
workload
hostboot runtime
shipped as machine FW
external software
FW↔external interface
platform firmware boot flow →


shipped as
machine FW
external
software


shipped as
machine FW
external
software
easy to distribute
modifications
???


Linus -
"upstream"
Linux
distributions
Ubuntu
RHEL
& others
powerpc
tree
custom
kernel code
end-user's
system

OpenPOWER™

Modifying those “external software” parts

- Ask every user to adopt your custom OS?
- Ask every user to apply your custom patches?
- Are you going to support these for the life of the hardware?

Regardless, talk to the OpenPOWER community.

OpenPOWER™

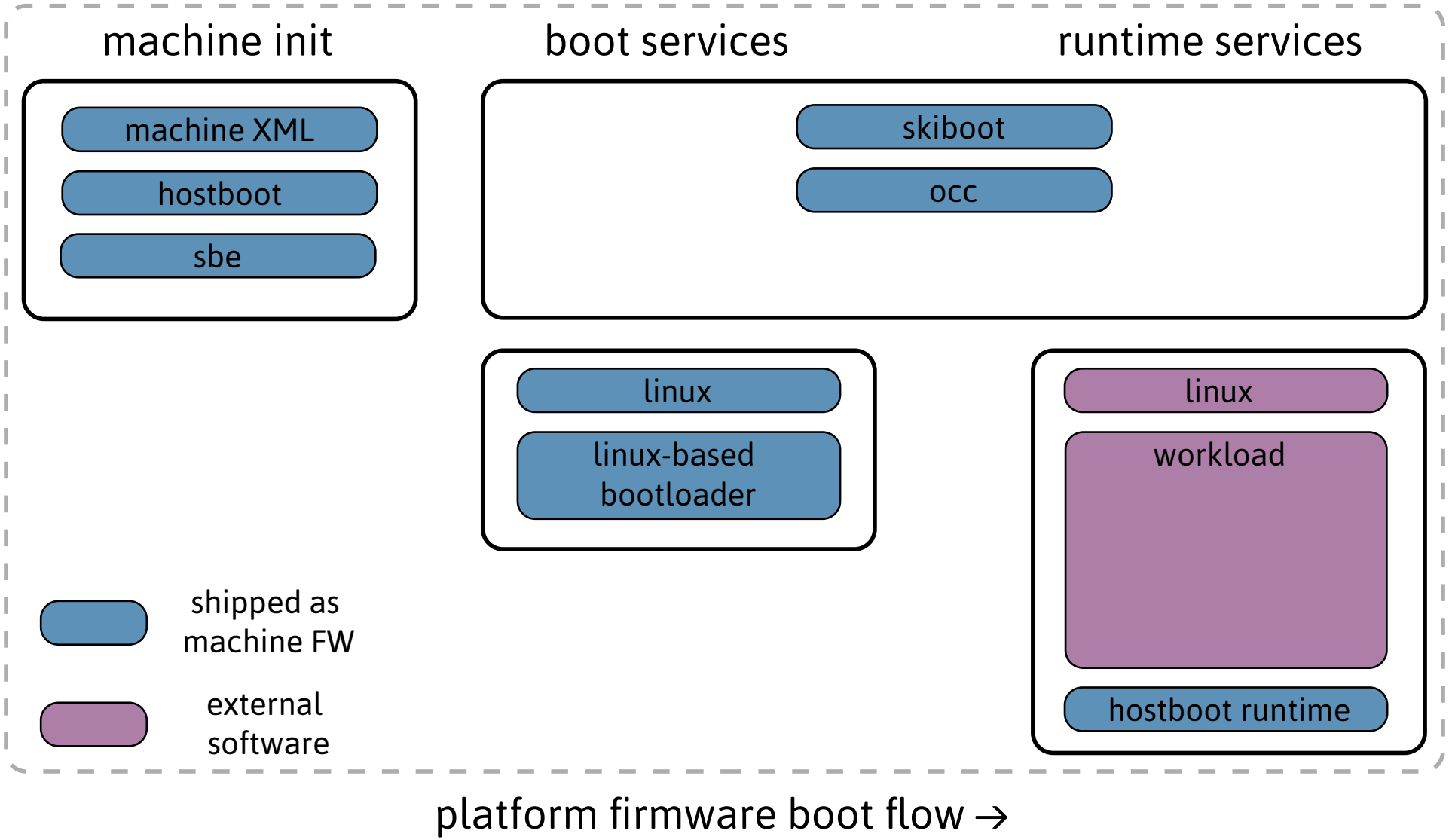
Work **with** current standards & implementations

(wherever possible)

OpenPOWER™

Discuss with **upstream**

(as soon as possible)


machine init
boot services
runtime services
machine XML
hostboot
sbe
skiboot
occ
linux
linux-based bootloader
linux
workload
hostboot runtime
shipped as machine FW
external software
platform firmware boot flow →

OpenPOWER™

```
# build using a clean op-build tree


~$ git clone https://github.com/open-power/op-build
~$ cd op-build
op-build$ . op-build-env
op-build$ op-build <machine>_defconfig
op-build$ op-build
```

```
# modify skiboot for your new platform


~$ git clone https://github.com/open-power/skiboot
~$ cd skiboot
skiboot$ vi platforms/astbmc/my-platform.c
skiboot$ git add platforms/astbmc/my-platform.c
skiboot$ git commit
```

```
# build locally, using system cross-compiler

skiboot$ make CROSS_COMPILE=powerpc64le-linux-gnu-
```

```
# build locally, using op-build cross-compiler

skiboot$ make \
 CROSS_COMPILE=~/op-build/output/host/bin/powerpc64le-buildroot-linux-gnu-
```

OpenPOWER™

```
# perform an op-build with a custom skiboot tree,
# by overriding some of op-build's SKIBOOT_ vars
# to reference your own code:

op-build$ op-build SKIBOOT_SITE=~/skiboot \
                   SKIBOOT_SITE_METHOD=git \
                   SKIBOOT_VERSION=<version>
```

Send upstream!

```
# create 0001-<name>.patch from the last 1 commit:
skiboot$ git format-patch -1

# make any edits to email
skiboot$ vi 0001-<name>.patch

# send to skiboot mailing list
skiboot$ git send-email \
              --to skiboot@lists.ozlabs.org
              0001-<name>.patch
```

OpenPOWER™

```
# perform an op-build with a custom skiboot tree,
# by overriding some of op-build's SKIBOOT_ vars
# to reference your own code:

op-build$ op-build SKIBOOT_SITE=~/skiboot \
                   SKIBOOT_SITE_METHOD=git \
                   SKIBOOT_VERSION=<version>
```

OpenPOWER™

Components of a platform port

- Identifier (“compatible string”)
  - `OPAL_MODEL`
- Hostboot/XML machine definition
- OPAL platform definition

OpenPOWER™

Compatible string

- Used as the canonical machine identifier
- <vendor>,<name> format: "`ibm,witherspoon`"
- **Must be unique**
  - It's our very-last-resort for post-release workarounds
  - Will be one of the first things we check when debugging

```
static bool witherspoon_probe(void)
{
	if (!dt_node_is_compatible(dt_root, "ibm,witherspoon"))
		return false;

	/* early platform initialisation here */

	return true;
}
```

Test!

OpenPOWER™

Resources

- Per-project github pages
  - http://github.com/open-power/...
- Per-project mailing lists
  - OPAL/skiboot: skiboot@lists.ozlabs.org
  - Petitboot: petitboot@lists.ozlabs.org
  - Linux: linuxppc-dev@lists.ozlabs.org

OpenPOWER™

Not sure where to start?

- Overall firmware mailing list:
  - openpower-firmware@lists.ozlabs.org
- Stack Overflow community
  - http://stackoverflow.com/
  - Use the “openpower” tag for questions
- Ask me!
  - jk@linux.ibm.com

OpenPOWER™

OpenPOWER Foundation

- System Software Workgroup: http://openpowerfoundation.org/

This work represents the view of the author and does not necessarily represent the view of IBM.