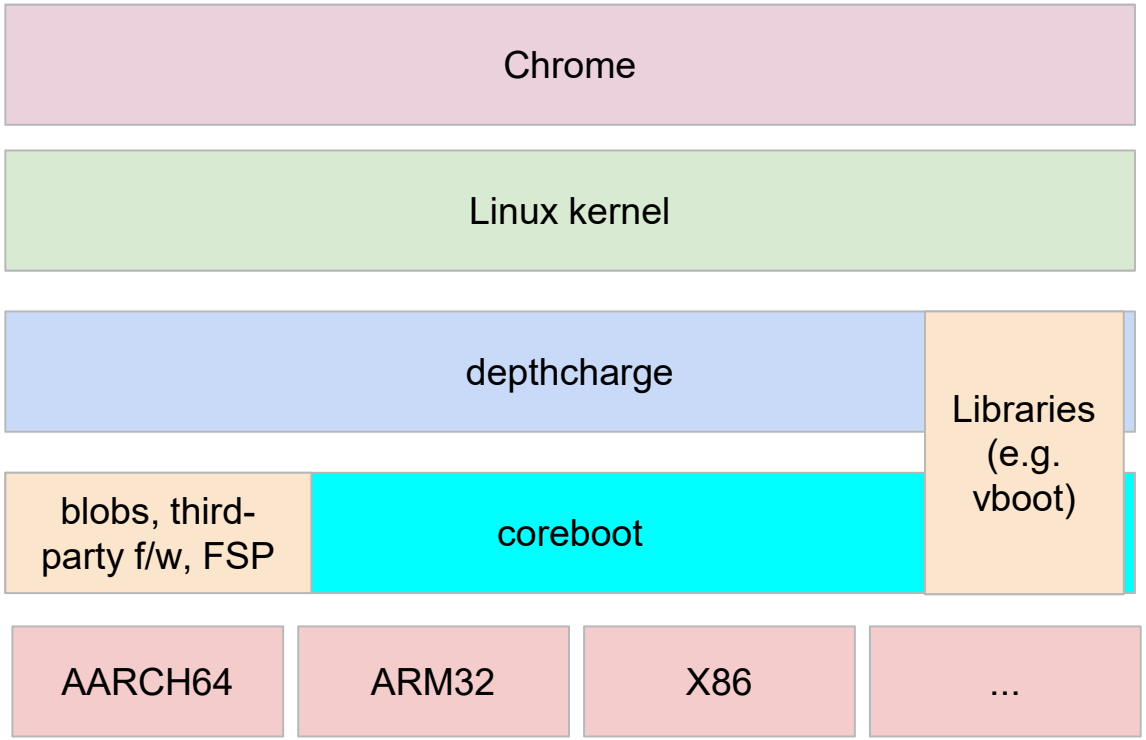
A tale of reusability in coreboot

Furquan Shaikh

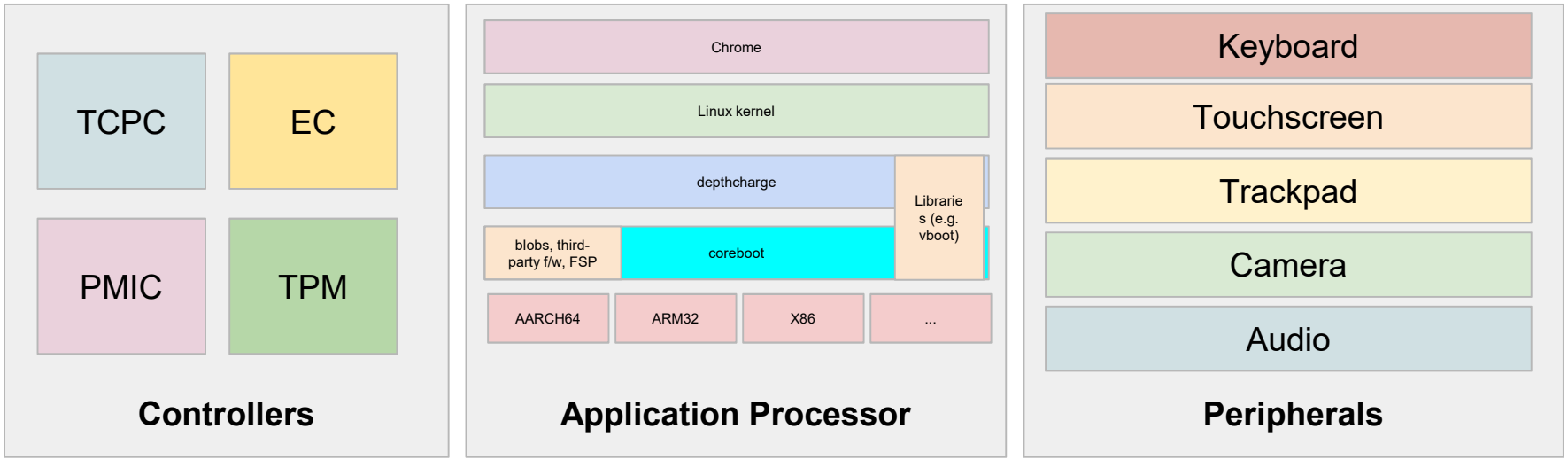
Table of contents

Once upon a time.....

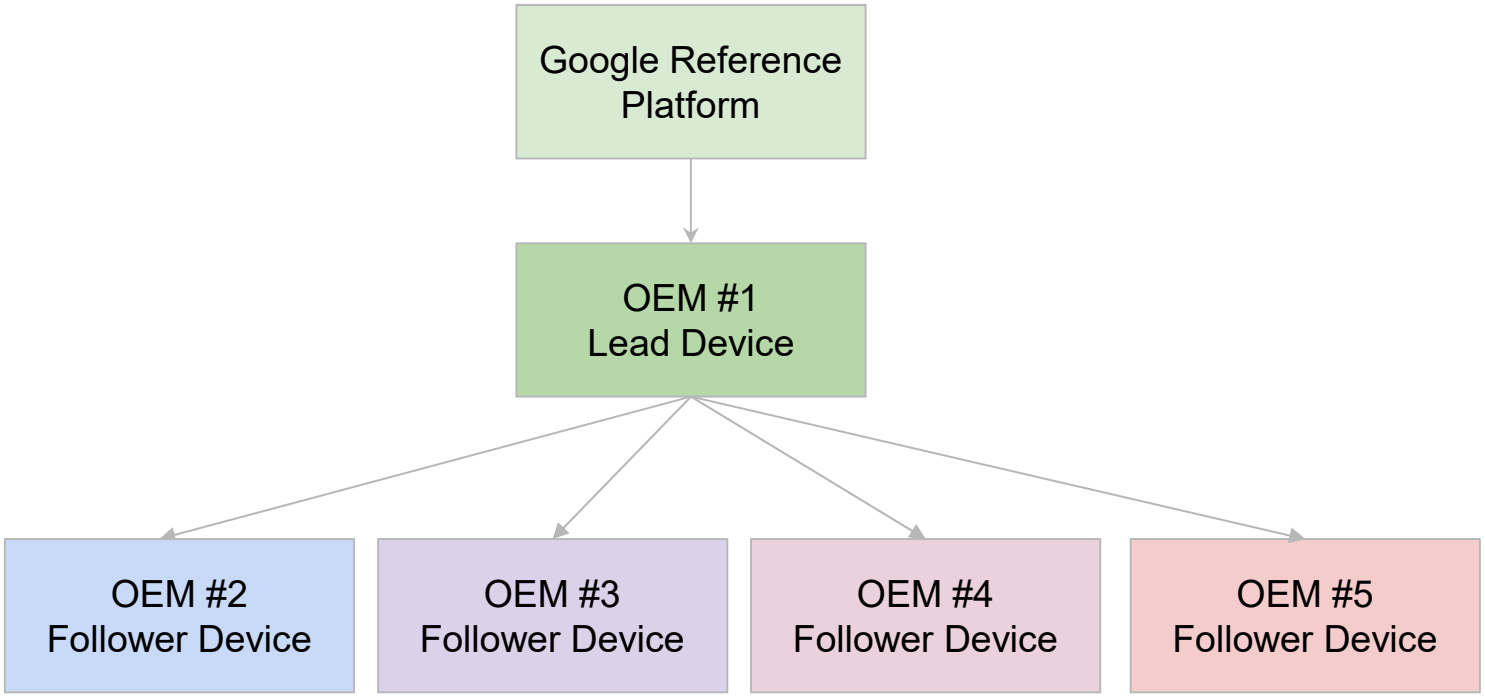
What is Chrome OS ?



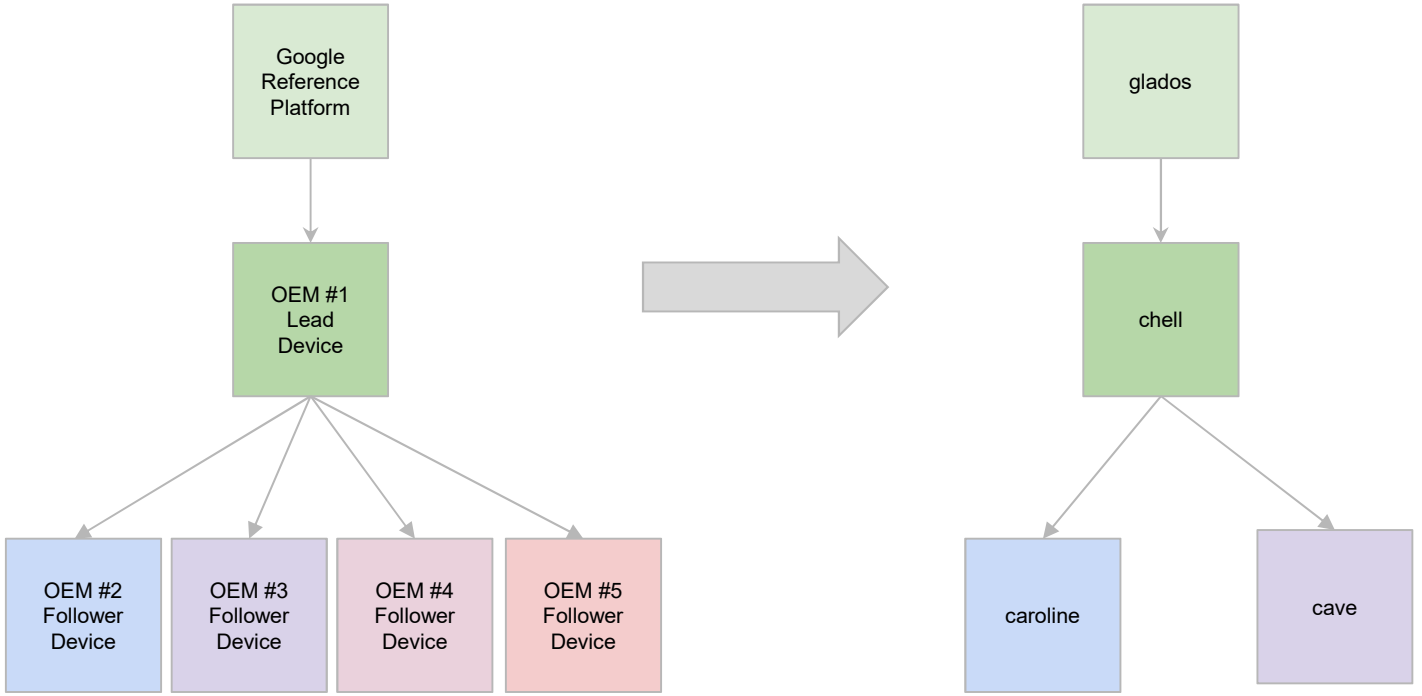
What does Chrome OS platform look like ?



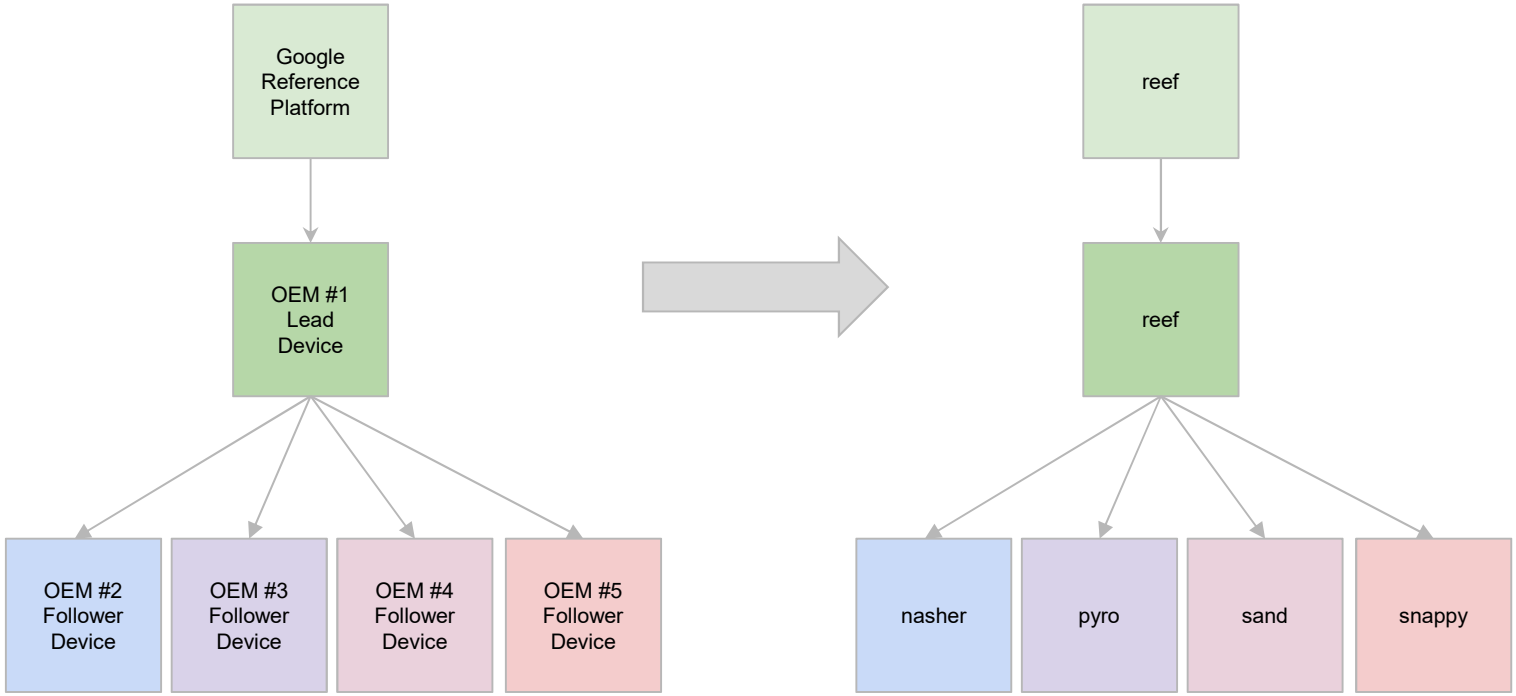
Traditional Chrome OS Platform Model



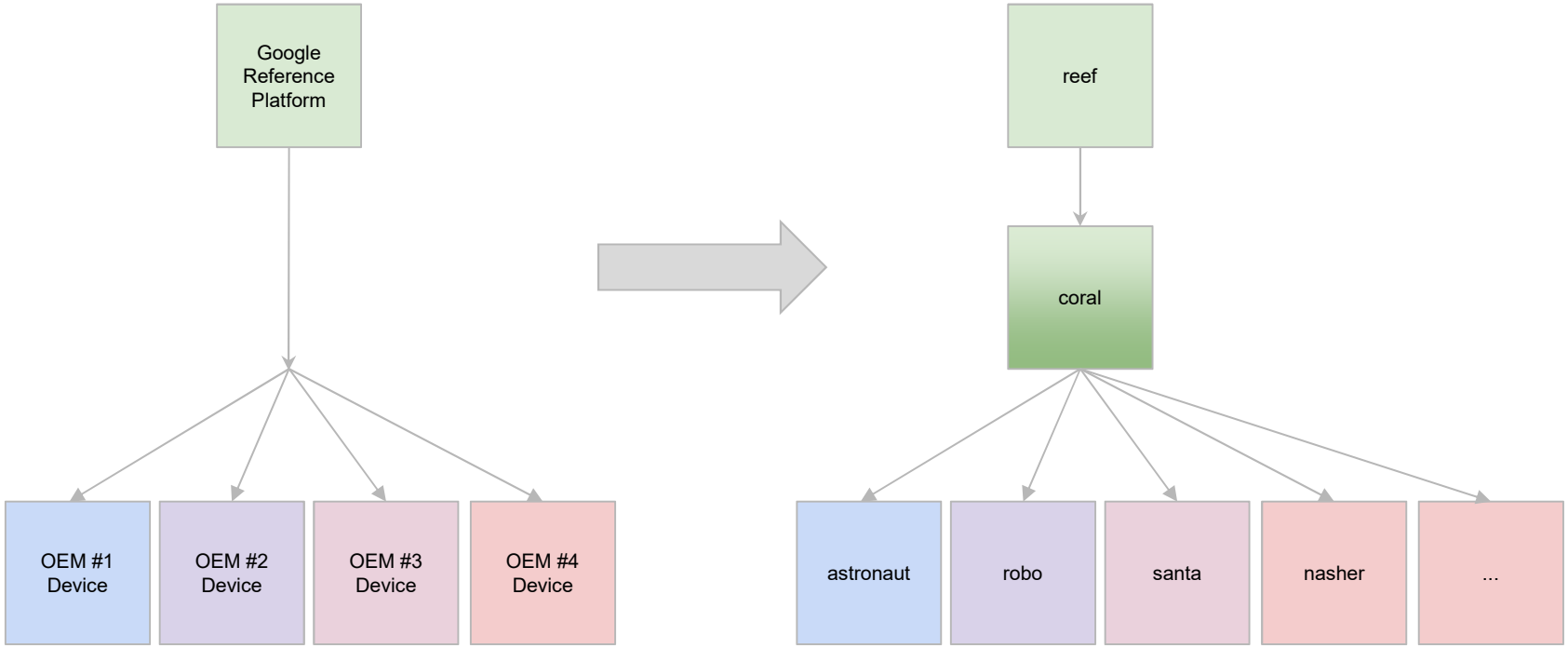
Traditional Chrome OS Platform Model



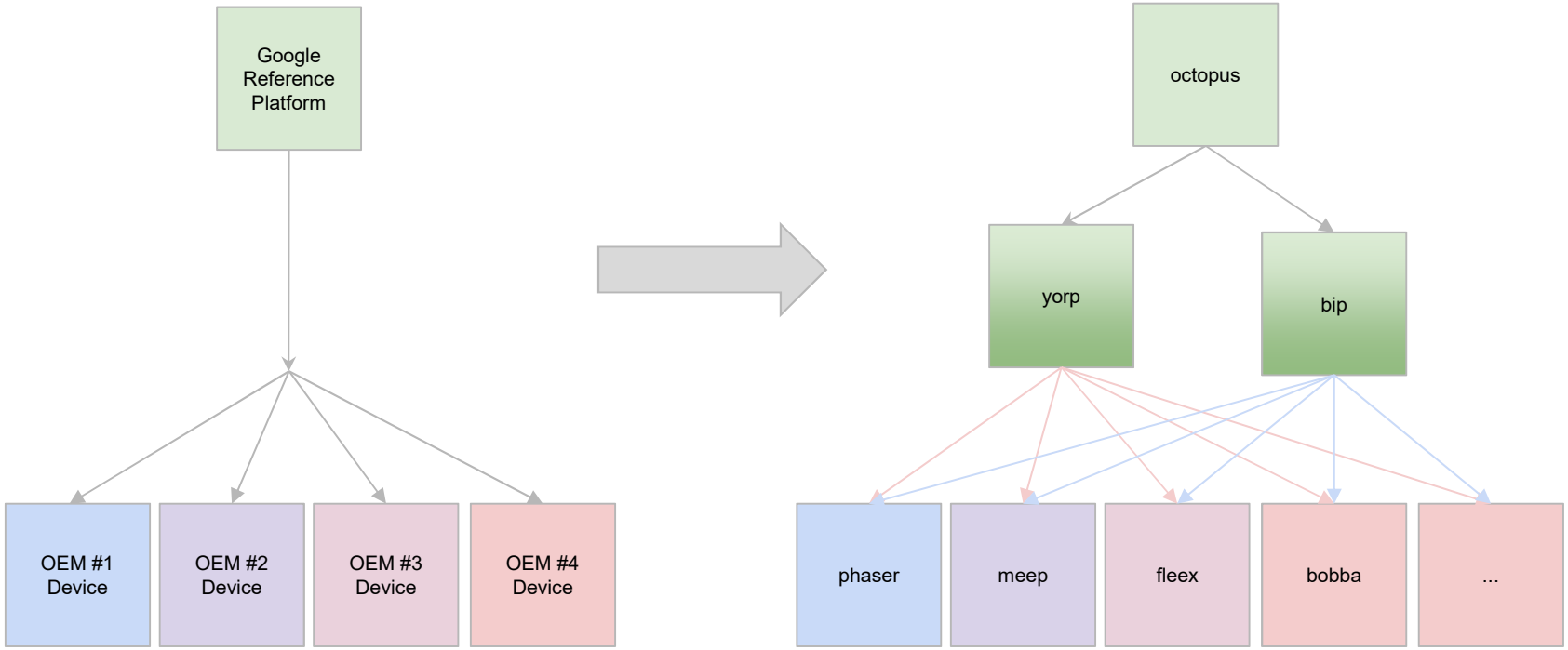
Traditional Chrome OS Platform Model



~~Traditional~~ Current Chrome OS Platform Model



~~Traditional~~ Current Chrome OS Platform Model

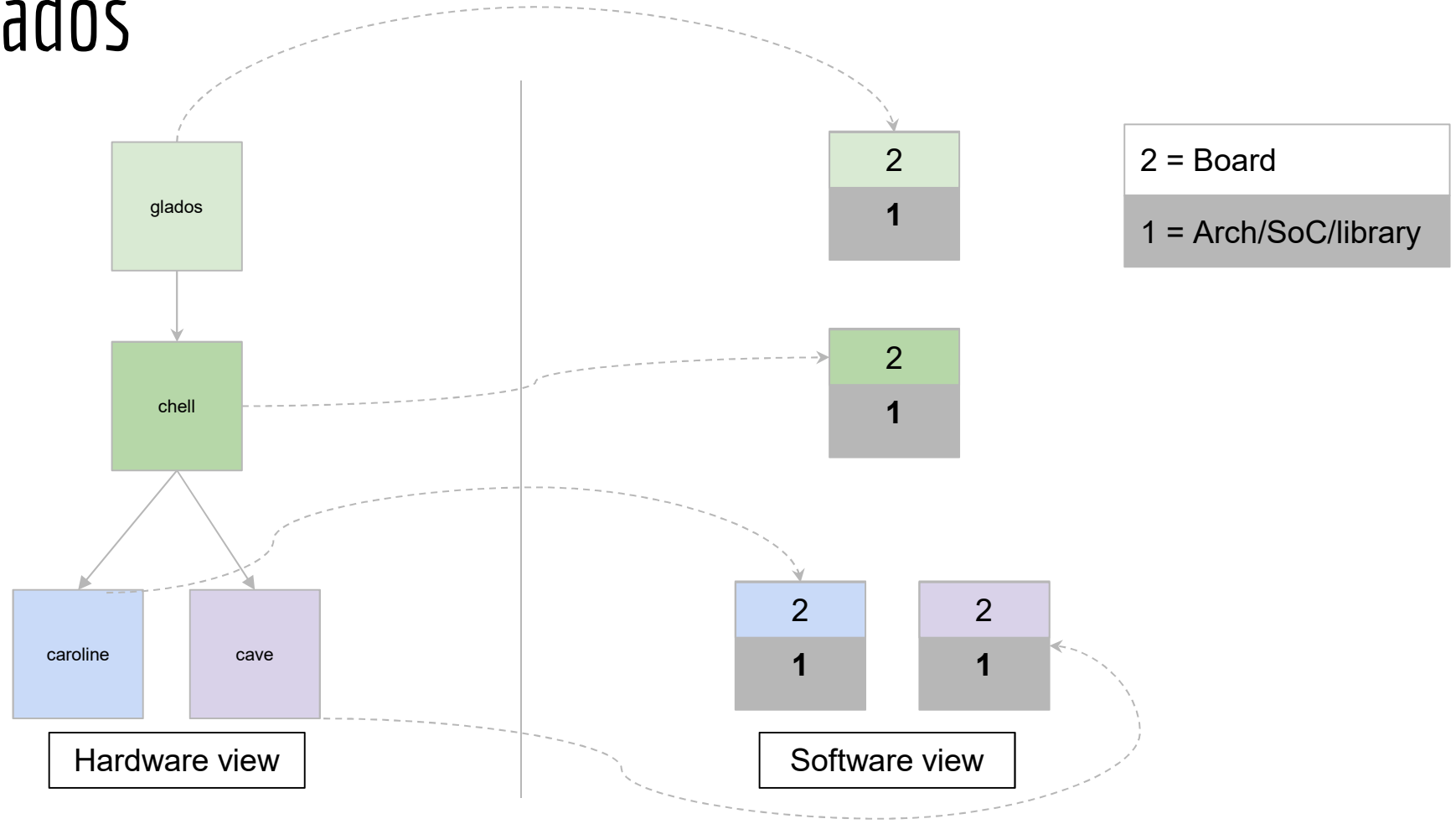


but.....

Objectives

- Faster time to market for all OEMs
- More flexibility to OEMs w.r.t components
- Avoid duplication of work within software stack
- Make it manageable for everyone
- Make the whole process scalable

Glados


glados
chell
caroline
cave
Hardware view
2
1
2
1
2
1
2
1
Software view
2 = Board
1 = Arch/SoC/library

mainboard.asl file

https://chromium.googlesource.com/chromiumos/third_party/coreboot/+/firmware-glados-7820.B/src/mainboard/google/glados/acpi/mainboard.asl

```
Scope (\_SB.PCI0.I2C0)
{
    /* Touchscreen */
    Device (ELTS)
    {
        Name (_HID, "ELAN0001")
        Name (_DDN, "Elan Touchscreen")
        Name (_UID, 1)
        Name (_S0W, 4)

        Name (_CRS, ResourceTemplate ()
        {
            I2cSerialBus (
                BOARD_TOUCHSCREEN_I2C_ADDR,
                ControllerInitiated,
                400000,
                AddressingMode7Bit,
                "\\_SB.PCI0.I2C0",
            )
            Interrupt (ResourceConsumer, Edge,
ActiveLow)
            {
                BOARD_TOUCHSCREEN_IRQ
            }
        })

        Method (_STA)
        {
            Return (0xF)
        }
    }
}
```

https://chromium.googlesource.com/chromiumos/third_party/coreboot/+/firmware-glados-7820.B/src/mainboard/google/chell/acpi/mainboard.asl

```
Scope (\_SB.PCI0.I2C0)
{
    Name (FMCN, Package () { 72, 160, 30 })

    /* Touchscreen */
    Device (ELTS)
    {
        Name (_HID, "ELAN0001")
        Name (_DDN, "Elan Touchscreen")
        Name (_UID, 1)
        Name (_S0W, 4)

        Name (_CRS, ResourceTemplate ()
        {
            I2cSerialBus (
                BOARD_TOUCHSCREEN_I2C_ADDR,
                ControllerInitiated,
                400000,
                AddressingMode7Bit,
                "\\_SB.PCI0.I2C0",
            )
            Interrupt (ResourceConsumer, Edge,
ActiveLow)
            {
                BOARD_TOUCHSCREEN_IRQ
            }
        })

        Method (_STA)
        {
            Return (0xF)
        }
    }
}
```

https://chromium.googlesource.com/chromiumos/third_party/coreboot/+/firmware-glados-7820.B/src/mainboard/google/caroline/acpi/mainboard.asl

```
Scope (\_SB.PCI0.I2C0)
{
    Name (FMCN, Package () { 87, 197, 26 })

    Device (ATSA)
    {
        Name (_HID, "ATML0001")
        Name (_DDN, "Atmel Touchscreen")
        Name (_UID, 1)
        Name (_S0W, 4)

        Name (_CRS, ResourceTemplate ()
        {
            I2cSerialBus (
                BOARD_TOUCHSCREEN_I2C_ADDR,
                ControllerInitiated,
                400000,
                AddressingMode7Bit,
                "\\_SB.PCI0.I2C0",
            )
            Interrupt (ResourceConsumer, Edge,
ActiveLow)
            {
                BOARD_TOUCHSCREEN_IRQ
            }
        })

        Method (_STA)
        {
            Return (0xF)
        }
    }
}
```

SSDT generation in coreboot

- Runtime addition of device nodes to ACPI SSDT[1] based on device tree properties.
- Relieves developers from the hectic task of defining device nodes in ASL.
- Driver in ramstage responsible for generation of the nodes at runtime.
- Examples:
  - src/drivers/i2c/generic
  - src/drivers/spi/acpi
  - src/drivers/generic/gpio_keys
  - ...
  - soc/.../.../chip.c
  - ...
- git grep “acpi_fill_ssdt_generator”

1. SSDT - Secondary System Description Table (https://www.uefi.org/acpi/specs)

SSDT generation in coreboot

https://chromium.googlesource.com/chromiumos/third_party/coreboot/+/firmware-glados-7820.B/src/mainboard/google/glados/acpi/mainboard.asl

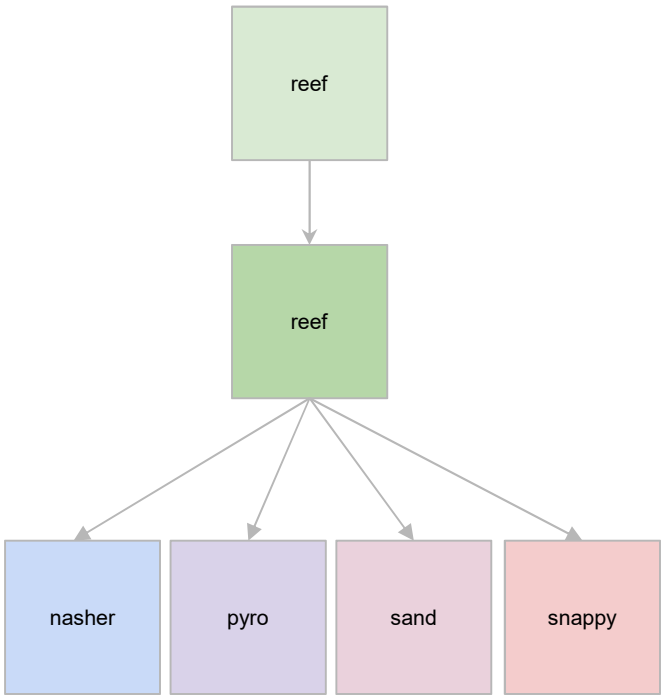
```
Scope (\_SB.PCI0.I2C0)
{
    /* Touchscreen */
    Device (ELTS)
    {
        Name (_HID, "ELAN0001")
        Name (_DDN, "Elan Touchscreen")
        Name (_UID, 1)
        Name (_S0W, 4)

        Name (_CRS, ResourceTemplate ()
        {
            I2cSerialBus (
                BOARD_TOUCHSCREEN_I2C_ADDR,
                ControllerInitiated,
                400000,
                AddressingMode7Bit,
                "\\_SB.PCI0.I2C0",
            )
            Interrupt (ResourceConsumer, Edge,
ActiveLow)
            {
                BOARD_TOUCHSCREEN_IRQ
            }
        })

        Method (_STA)
        {
            Return (0xF)
        }
    }
}
```

https://review.coreboot.org/cgit/coreboot.git/tree/src/mainboard/google/glados/variants/glados/devicetree.cb?id=refs/heads/master

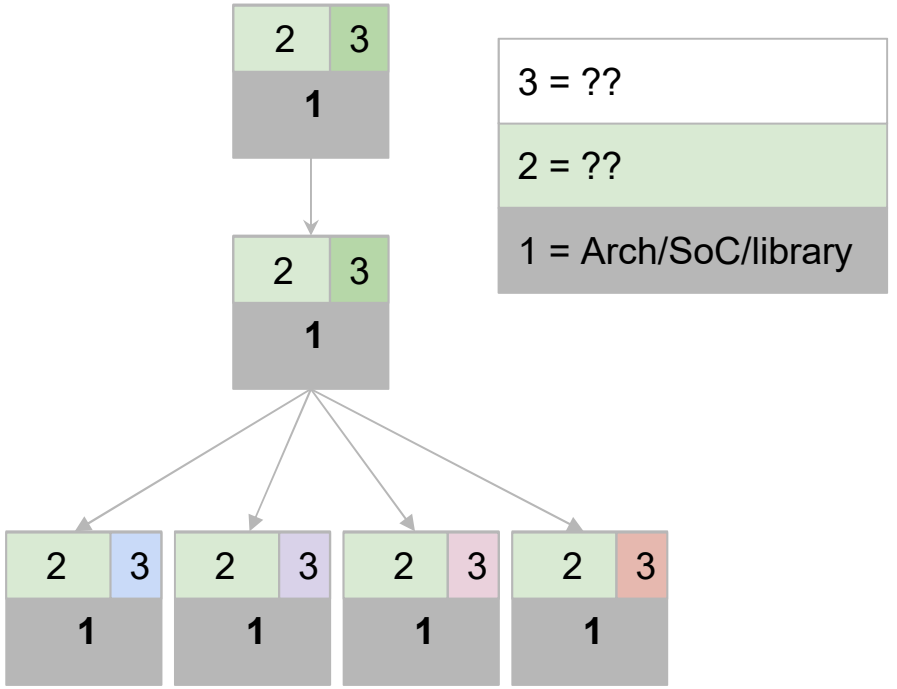
```
device pci 15.0 on
    chip drivers/i2c/generic
        register "hid" = ""ELAN0001""
        register "desc" = ""ELAN Touchscreen""
        register "irq" = "ACPI_IRQ_EDGE_LOW(GPP_E7_IRQ)"
        device i2c 10 on end
    end
end # I2C #0
```

SSDT generator driver

Reef



Hardware view



Software view

Baseboard and Variant Structure

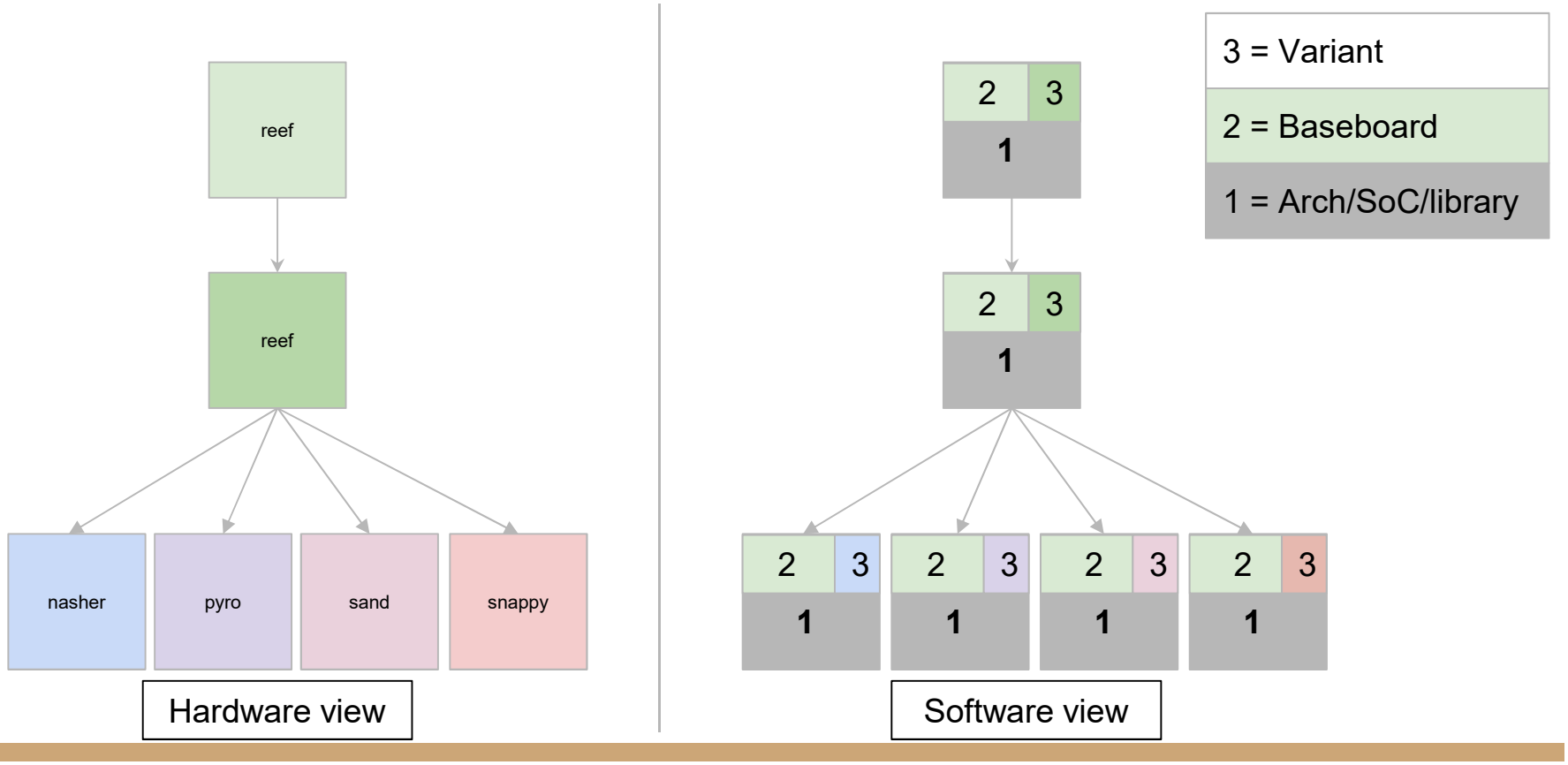
Glados and followers

```
src/
└ mainboard/
        └ google/
            ├ glados/
            ├ chell/
            ├ caroline/
            └ cave/
```

Reef and followers

```
src/
└ mainboard/
        └ google/
            └ reef/
                └ variants/
                    ├ baseboard/
                    ├ nasher/
                    ├ pyro/
                    ├ reef/
                    ├ sand/
                    └ snappy/
```

Baseboard and Variant Structure



Baseboard and Variant Structure

 # wc -l `find variants/baseboard/ -type f`
48 variants/baseboard/nhlt.c
234 variants/baseboard/devicetree.cb
396 variants/baseboard/gpio.c
54 variants/baseboard/include/baseboard/variants.h
81 variants/baseboard/include/baseboard/acpi/dptf.asl
80 variants/baseboard/include/baseboard/ec.h
53 variants/baseboard/include/baseboard/gpio.h
170 variants/baseboard/memory.c
8 variants/baseboard/Makefile.inc
1124 total

 # wc -l `find variants/reef/ -type f`
16 variants/reef/include/variant/acpi/dptf.asl
21 variants/reef/include/variant/ec.h
21 variants/reef/include/variant/gpio.h
58 total

 # wc -l `find variants/nasher/ -type f`
18 variants/nasher/include/variant/acpi/dptf.asl
17 variants/nasher/include/variant/ec.h
17 variants/nasher/include/variant/gpio.h
52 total

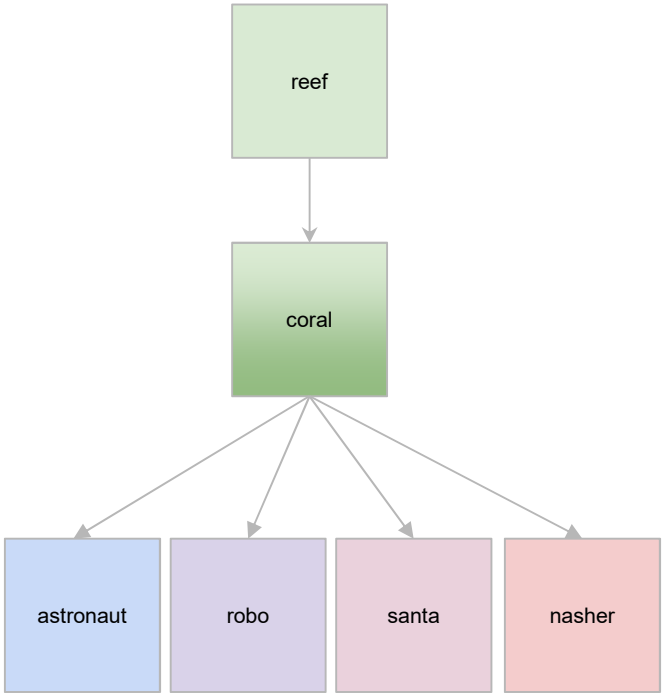
 # wc -l `find variants/sand/ -type f`
230 variants/sand/devicetree.cb
81 variants/sand/include/variant/acpi/dptf.asl
81 variants/sand/include/variant/ec.h
17 variants/sand/include/variant/gpio.h
409 total

 # wc -l `find variants/pyro/ -type f`
249 variants/pyro/devicetree.cb
81 variants/pyro/include/variant/acpi/dptf.asl
21 variants/pyro/include/variant/ec.h
21 variants/pyro/include/variant/gpio.h
114 variants/pyro/memory.c
1 variants/pyro/Makefile.inc
487 total

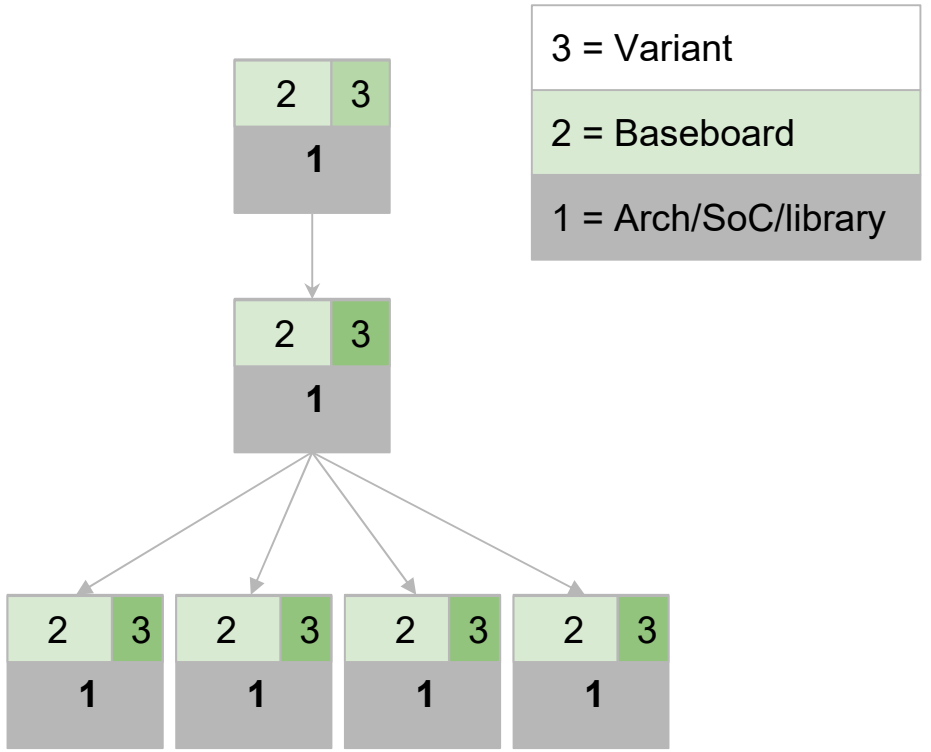
 # wc -l `find variants/snappy/ -type f`
123 variants/snappy/mainboard.c
286 variants/snappy/devicetree.cb
82 variants/snappy/include/variant/acpi/dptf.asl
24 variants/snappy/include/variant/ec.h
21 variants/snappy/include/variant/gpio.h
1 variants/snappy/Makefile.inc
537 total

 # wc -l `find . -not -path "./variants/*" -type f`
6 ./board_info.txt
68 ./dsdt.asl
148 ./mainboard.c
112 ./Kconfig
61 ./ec.c
54 ./chromeos.fmd
52 ./smihandler.c
14 ./acpi_tables.c
31 ./Kconfig.name
32 ./bootblock.c
50 ./chromeos.c
29 ./romstage.c
17 ./Makefile.inc
674 total

Coral

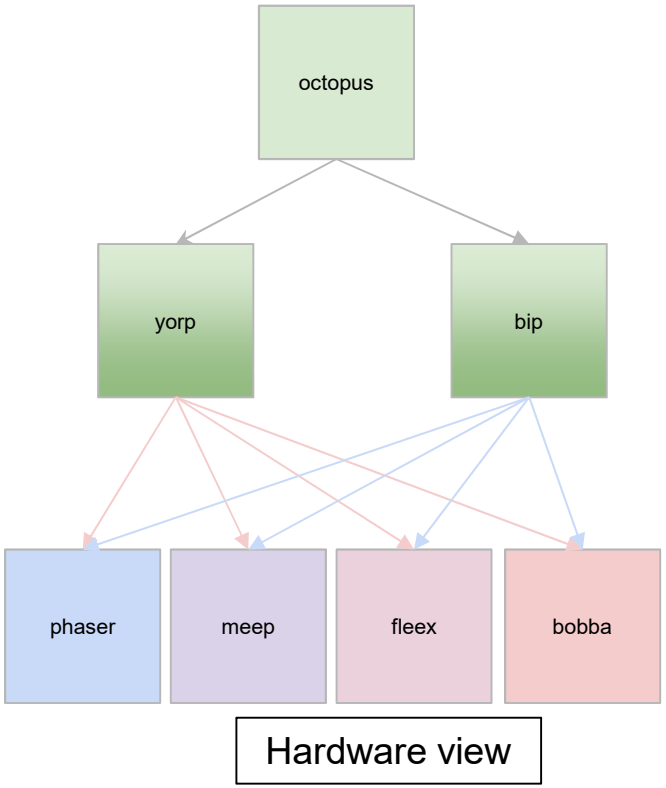


Hardware view

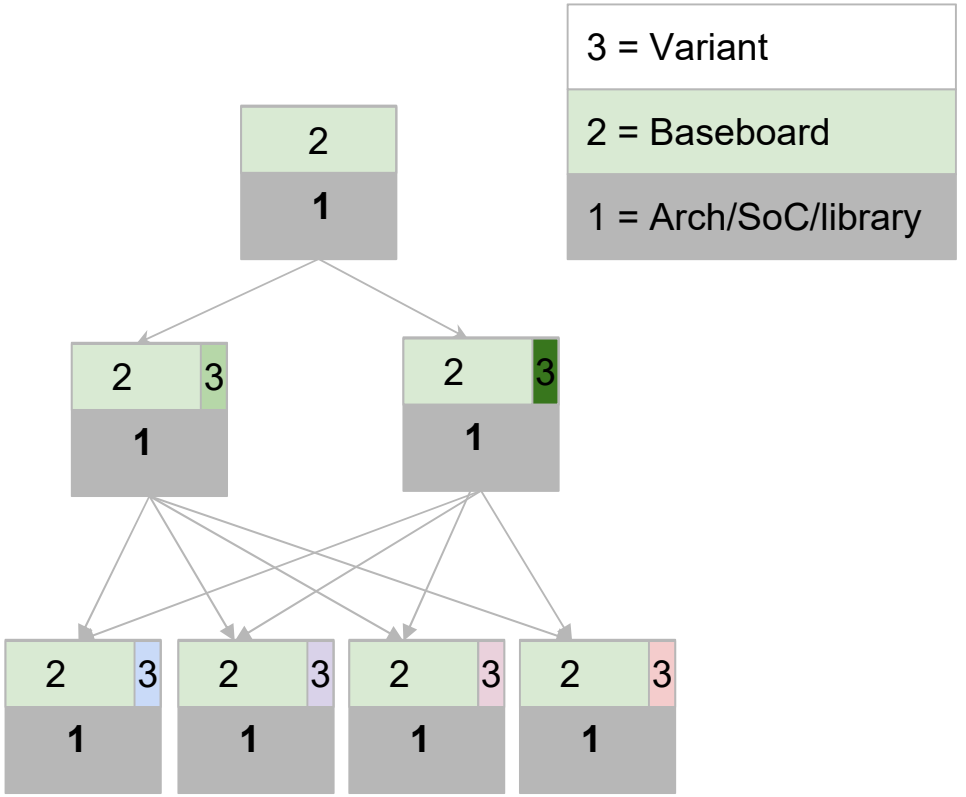


Software view

Octopus



Hardware view



Software view

Devicetree duplication

# diff sand/devicetree.cb baseboard/devicetree.cb | wc -l **16** **# wc -l sand/devicetree.cb baseboard/devicetree.cb** 230 sand/devicetree.cb 234 baseboard/devicetree.cb	**# wc -l `find variants/sand/ -type f`** **230 variants/sand/devicetree.cb** 81 variants/sand/include/variant/acpi/dptf.asl 81 variants/sand/include/variant/ec.h 17 variants/sand/include/variant/gpio.h **409 total**
# diff pyro/devicetree.cb baseboard/devicetree.cb | wc -l **63** **# wc -l pyro/devicetree.cb baseboard/devicetree.cb** 249 pyro/devicetree.cb 234 baseboard/devicetree.cb	**# wc -l `find variants/pyro/ -type f`** **249 variants/pyro/devicetree.cb** 81 variants/pyro/include/variant/acpi/dptf.asl 21 variants/pyro/include/variant/ec.h 21 variants/pyro/include/variant/gpio.h 114 variants/pyro/memory.c 1 variants/pyro/Makefile.inc **487 total**
# diff snappy/devicetree.cb baseboard/devicetree.cb | wc -l **77** **# wc -l snappy/devicetree.cb baseboard/devicetree.cb** 286 snappy/devicetree.cb 234 baseboard/devicetree.cb	**# wc -l `find variants/snappy/ -type f`** 123 variants/snappy/mainboard.c **286 variants/snappy/devicetree.cb** 82 variants/snappy/include/variant/acpi/dptf.asl 24 variants/snappy/include/variant/ec.h 21 variants/snappy/include/variant/gpio.h 1 variants/snappy/Makefile.inc **537 total**

Override devicetree

- sconfig utility updated to allow variants to specify an override tree.
- Applies variant specific overrides on top the properties specified by the base devicetree.
- Rules for overriding:
  - If a property is present only in the override tree, then it takes effect.
  - If a property is present in base tree as well as override tree, then the value in override tree takes effect.
  - If a device has a chip associated with it in the override tree, then that device gets added to the appropriate parent in base tree.
  - If the parent device in base tree already has some children, then the device in override tree is added as a sibling of the children in base tree.

Override devicetree

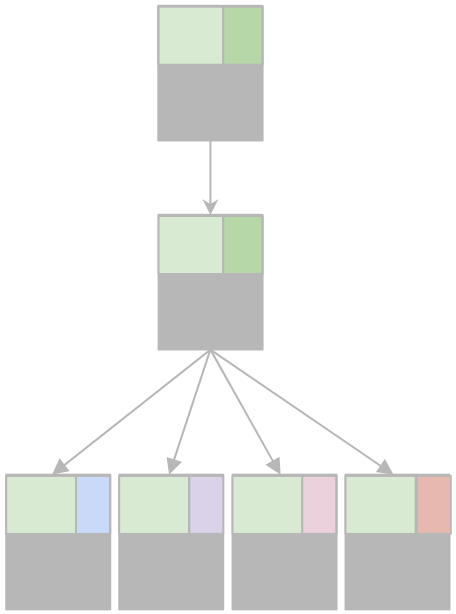
- Enabling override tree is as simple as:
  - Setting OVERRIDE_DEVICETREE to point to variants tree
  - Setting properties and device info in variants tree specific to the variant

- sconfig will walk through the base tree and override tree in lockstep fashion applying properties and device nodes as appropriate.

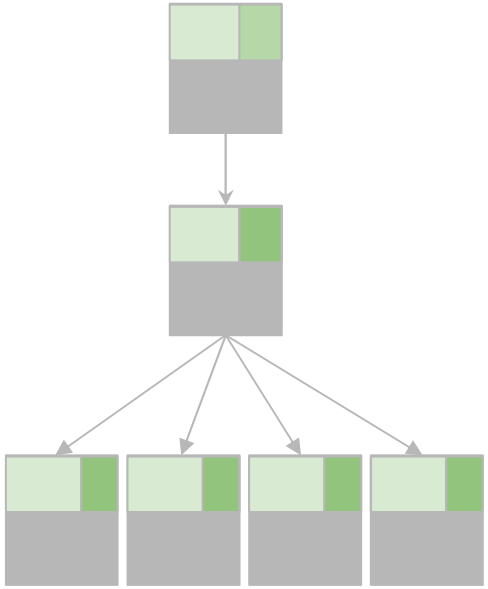
- A good example of override devicetree usage is octopus: https://review.coreboot.org/c/coreboot/+/27219/

Summary

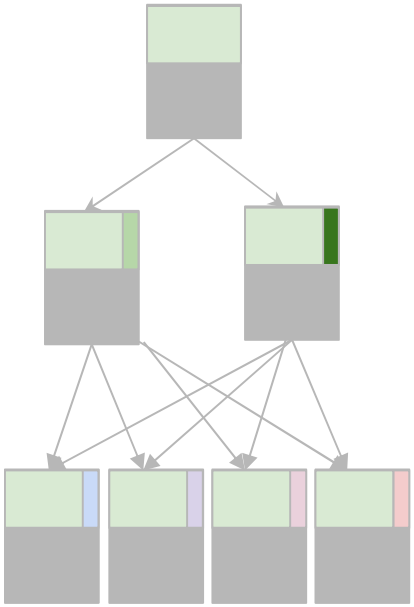
glados



reef



coral



octopus

What's next?

Current challenges - GPIO config table duplication

- Baseboard provides one set of GPIO configs.
- Variant can either:
  - Use the one provided by the baseboard, or
  - Supply its own table of GPIO configs.
- Pros:
  - No need to copy the same table over and over if variants have the GPIOs routed the same way in hardware as the baseboard.
- Cons:
  - If even one GPIO is different, then either variant needs to:
    - Provide a copy of baseboard table and change that one GPIO.
    - Call GPIO configuration after baseboard configuration is done
    - None of the above options are optimal

Current challenges - GPIO config table duplication

- Current workaround - gpio_configure_pads_with_override
  - Baseboard provides its default GPIO config table.
  - Variant provides an override table with only the GPIOs that are different from the baseboard.
  - gpio_configure_pads_with_override loops through all the GPIOs in base table and checks if there is an override present for it in the override table.
    - Yes: Applies the configuration from override table.
    - No: Applies the configuration from base table.
- What else can be done?

The End

Questions?