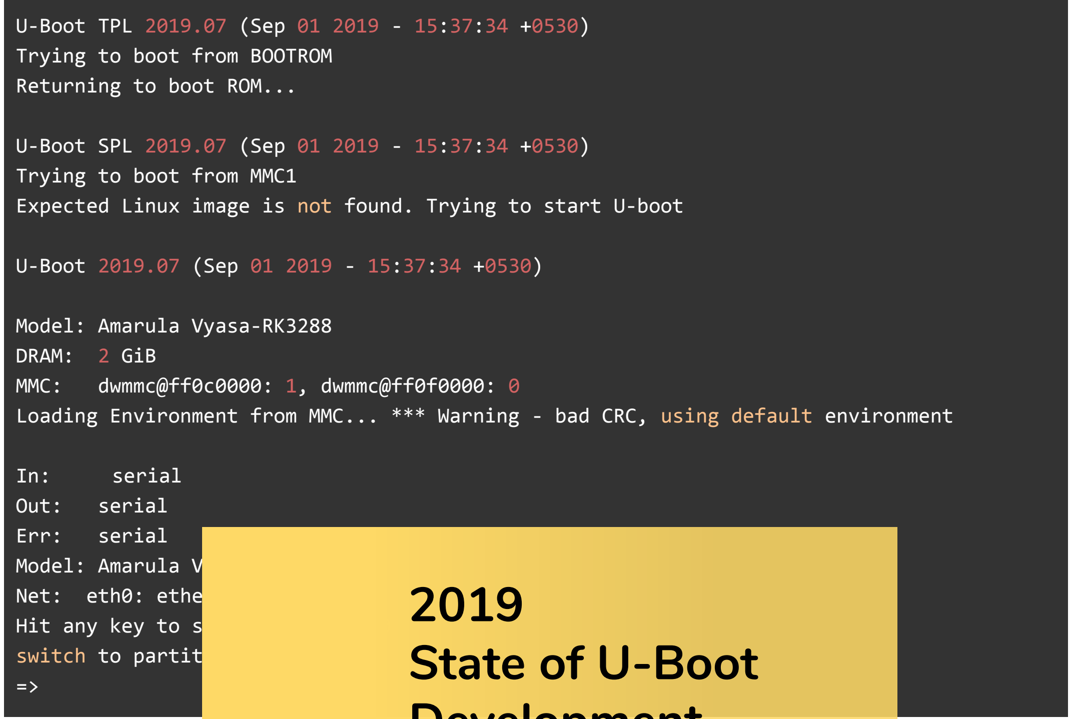


```
U-Boot TPL 2019.07 (Sep 01 2019 - 15:37:34 +0530)
Trying to boot from BOOTROM
Returning to boot ROM...

U-Boot SPL 2019.07 (Sep 01 2019 - 15:37:34 +0530)
Trying to boot from MMC1
Expected Linux image is not found. Trying to start U-boot

U-Boot 2019.07 (Sep 01 2019 - 15:37:34 +0530)

Model: Amarula Vyasa-RK3288
DRAM:   2 GiB
MMC:    dwmmc@ff0c0000: 1, dwmmc@ff0f0000: 0
Loading Environment from MMC... *** Warning - bad CRC, using default environment

In:      serial
Out:    serial
Err:    serial
Model: Amarula V
Net:  eth0: ethe
Hit any key to s
switch to partit
=>
```

2019 State of U-Boot Development Report

Jagan Teki, Amarula Solutions | OSFC 2019

Jagan Teki

- CEO and Embedded Linux Engineer at Amarula Solutions
  - *Bootloader*: BootROM, bootloaders, U-Boot, boot bsps, chip/board bring ups, devicetrees, device drivers, boottime, secure boot, atf, optee and etc.
  - *Embedded Linux*: Linux bsps, devicetrees, device drivers, multimedia, optimizations, integrations and etc.

- Mainline contributions
  - **Linux**
    - Contributor of Allwinner, Rockchip, i.MX platforms, bsps, device drivers.
    - Maintainer of few **DSI** LCD panels.
  - **U-Boot**
    - Contributor of Xilinx Zynq, Allwinner, Rockchip, i.MX platforms, bsps, device drivers.
    - Maintainer of Allwinner **sunXi** SoCs
    - Maintainer of **SPI/SPI-NOR** Subsystems
  - Contributor of **Buildroot**, **Yocto**

Agenda

U-Boot background

- ➔ Das U-Boot
- ➔ Supporting platforms/languages
- ➔ Development process
- ➔ Project statistics

Features merged till v2019.07

- ➔ Build systems, tools
- ➔ U-Boot, stages
- ➔ Image boot, complexities
- ➔ Devicetree, improvements
- ➔ Driver model
- ➔ Firmware upgrades
- ➔ Miscellaneous

Summary

- ➔ Lessons learned from 18 years of U-Boot
- ➔ Future plans

Das U-Boot

1876	**162**	**1570136**	**~50**	**Wolfgang Denk, Tom Rini**	**v2019.07**
Developers	Employers	Lines of Code	Maintainers	Head	Recent release

Source: cloc and gitgm with v1.3.0...v2019.10-rc3

Supporting platforms and languages

Architecture/SoC

ARM32: Aspeed, Altera, Allwinner, Atmel, Broadcom, Qemu, Qualcomm, Marvell, NXP, Rockchip, STM32, Tegra, TI, UniPhier, Xilinx

ARM64: Allwinner, Marvell, NXP, Rockchip, Tegra, UniPhier, Xilinx

X86 (Baytrail, Broadwell, Quark, etc)

ARC, M68K, MicroBlaze, MIPS, NDS32, NIOS2, PowerPC, RISCV, Sandbox, SuperH, Xtensa

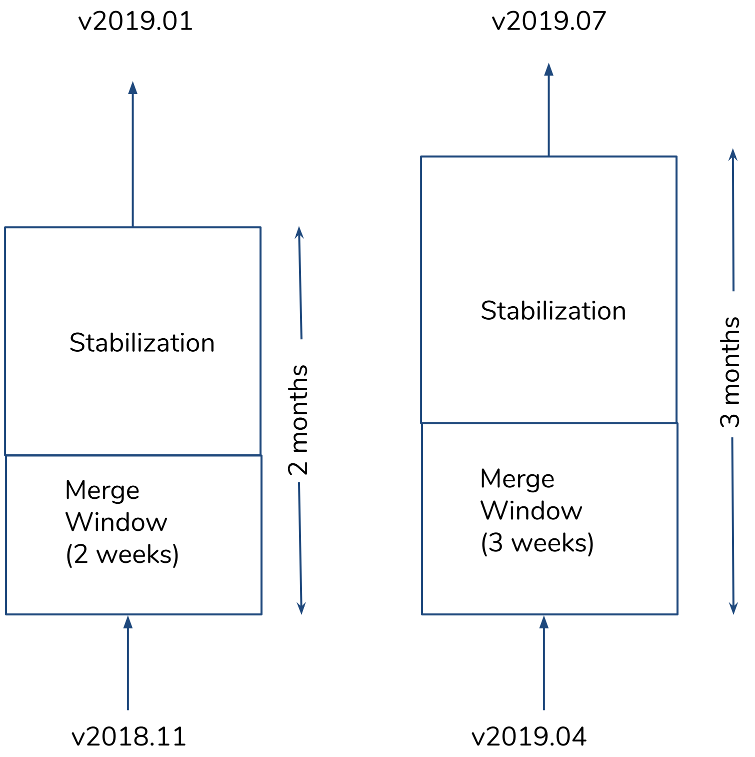
Boards

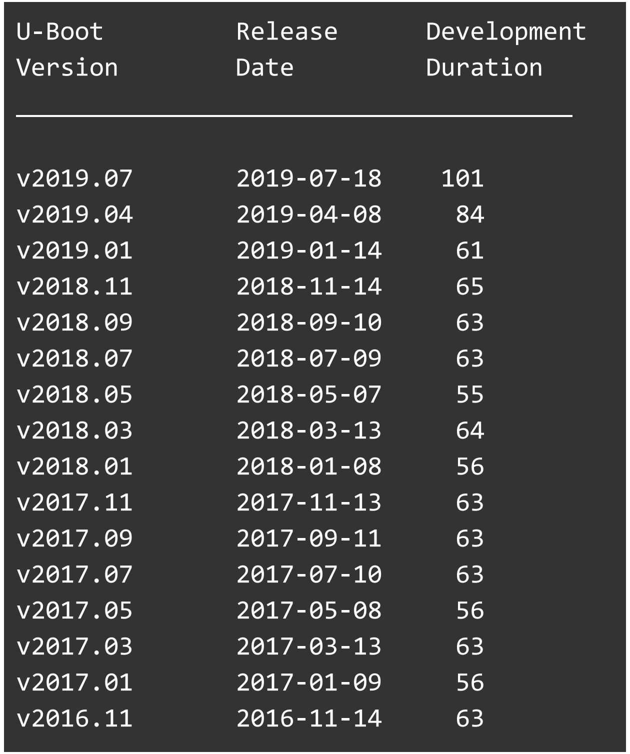
185+ different board vendors
~**1400** different boards

Language

C
C/C++ Header
Assembly
Python
make
Perl
Bourne Shell
C++
yacc
YAML
Glade
lex
NAnt script
Markdown
Bourne Again Shell
DOS Batch
CSS
Kermit
Tcl/Tk
sed
INI

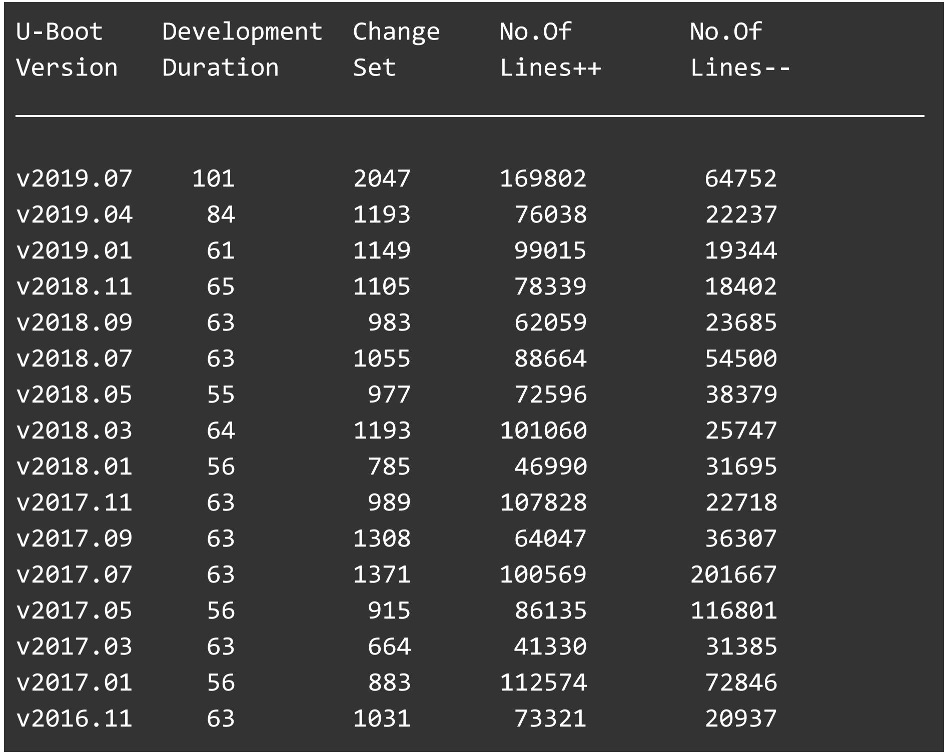
Development process





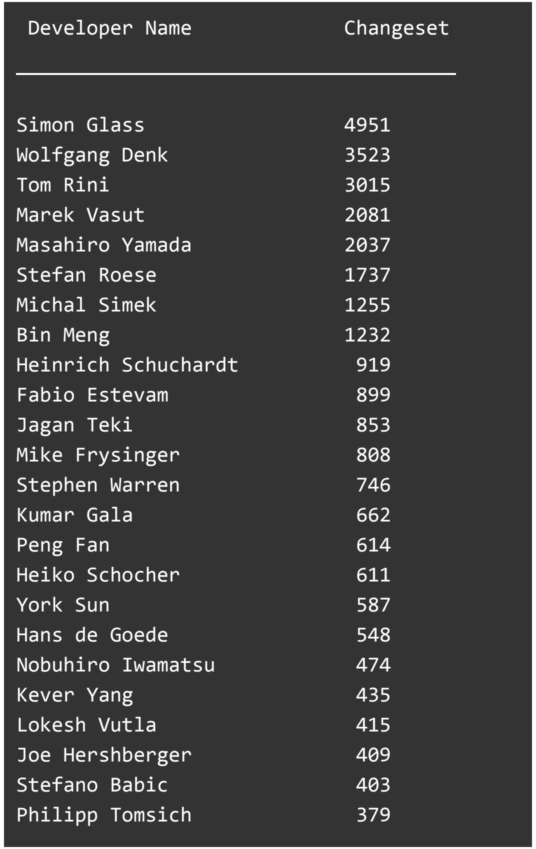
```
U-Boot           Release        Development
Version          Date           Duration
________________________________________________


v2019.07         2019-07-18      101
v2019.04         2019-04-08       84
v2019.01         2019-01-14       61
v2018.11         2018-11-14       65
v2018.09         2018-09-10       63
v2018.07         2018-07-09       63
v2018.05         2018-05-07       55
v2018.03         2018-03-13       64
v2018.01         2018-01-08       56
v2017.11         2017-11-13       63
v2017.09         2017-09-11       63
v2017.07         2017-07-10       63
v2017.05         2017-05-08       56
v2017.03         2017-03-13       63
v2017.01         2017-01-09       56
v2016.11         2016-11-14       63
```

U-Boot Changesets, Code size



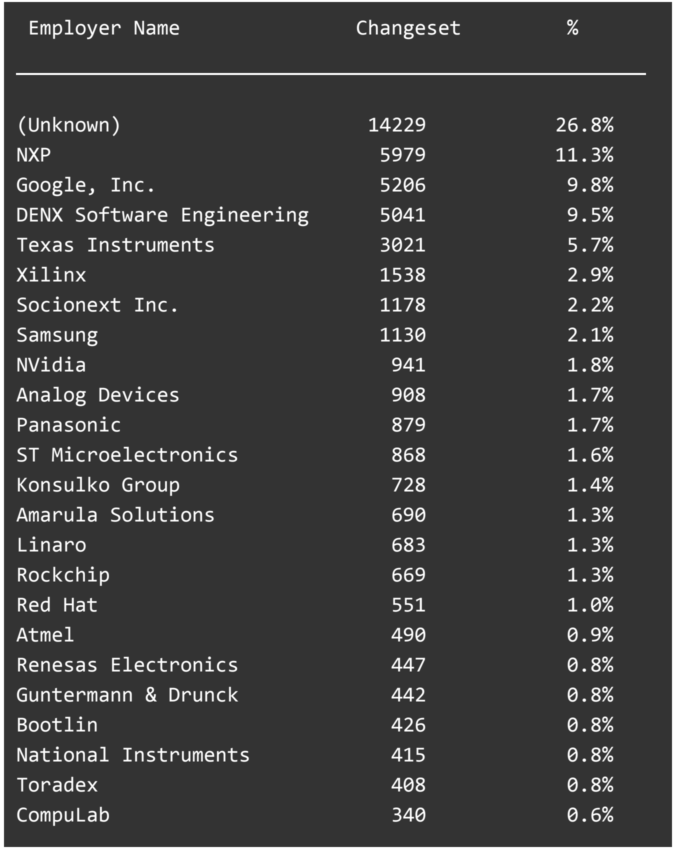
```
U-Boot     Development  Change     No.Of        No.Of
Version    Duration     Set        Lines++      Lines--

──────────────────────────────────────────────────────────────


v2019.07    101         2047       169802        64752
v2019.04     84         1193        76038        22237
v2019.01     61         1149        99015        19344
v2018.11     65         1105        78339        18402
v2018.09     63          983        62059        23685
v2018.07     63         1055        88664        54500
v2018.05     55          977        72596        38379
v2018.03     64         1193       101060        25747
v2018.01     56          785        46990        31695
v2017.11     63          989       107828        22718
v2017.09     63         1308        64047        36307
v2017.07     63         1371       100569       201667
v2017.05     56          915        86135       116801
v2017.03     63          664        41330        31385
v2017.01     56          883       112574        72846
v2016.11     63         1031        73321        20937
```

Who is Contributing the work



```
 Developer Name              Changeset
______________________________________

Simon Glass                  4951
Wolfgang Denk                3523
Tom Rini                     3015
Marek Vasut                  2081
Masahiro Yamada              2037
Stefan Roese                 1737
Michal Simek                 1255
Bin Meng                     1232
Heinrich Schuchardt           919
Fabio Estevam                 899
Jagan Teki                    853
Mike Frysinger                808
Stephen Warren                746
Kumar Gala                    662
Peng Fan                      614
Heiko Schocher                611
York Sun                      587
Hans de Goede                 548
Nobuhiro Iwamatsu             474
Kever Yang                    435
Lokesh Vutla                  415
Joe Hershberger               409
Stefano Babic                 403
Philipp Tomsich               379
```

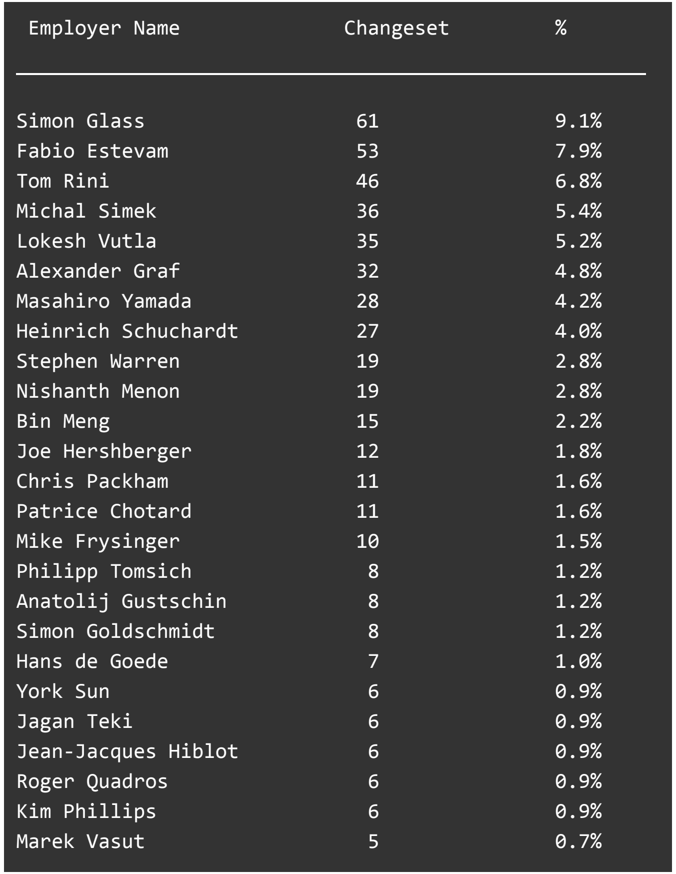
Who is Sponsoring the work



```
 Employer Name                Changeset        %
_______________________________________________________

(Unknown)                      14229          26.8%
NXP                             5979          11.3%
Google, Inc.                    5206           9.8%
DENX Software Engineering       5041           9.5%
Texas Instruments               3021           5.7%
Xilinx                          1538           2.9%
Socionext Inc.                  1178           2.2%
Samsung                         1130           2.1%
NVidia                           941           1.8%
Analog Devices                   908           1.7%
Panasonic                        879           1.7%
ST Microelectronics              868           1.6%
Konsulko Group                   728           1.4%
Amarula Solutions                690           1.3%
Linaro                           683           1.3%
Rockchip                         669           1.3%
Red Hat                          551           1.0%
Atmel                            490           0.9%
Renesas Electronics              447           0.8%
Guntermann & Drunck              442           0.8%
Bootlin                          426           0.8%
National Instruments             415           0.8%
Toradex                          408           0.8%
CompuLab                         340           0.6%
```

Note: Top 25 developers/employers till v2019.10-rc3

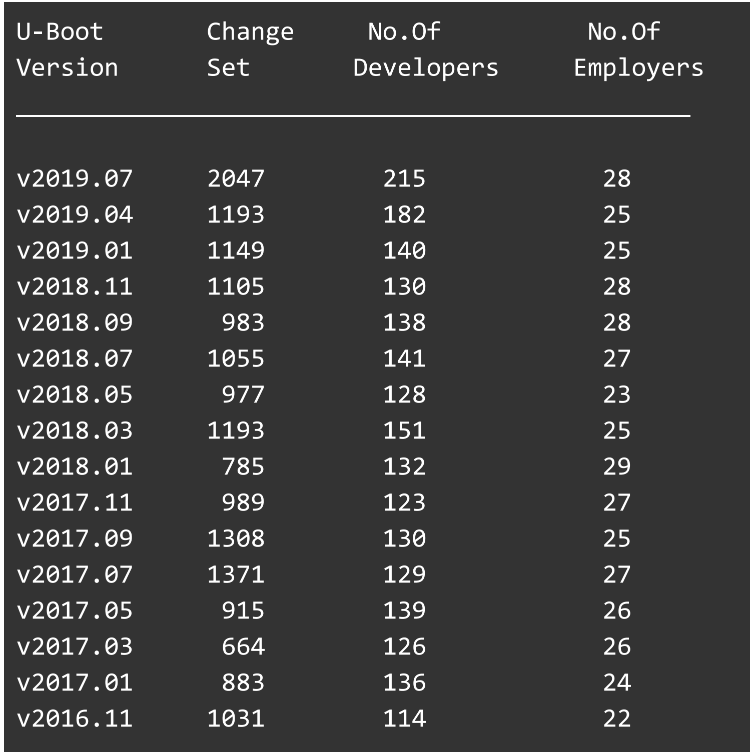
Who is Reporting the work



Employer Name	Changeset	%
Simon Glass	61	9.1%
Fabio Estevam	53	7.9%
Tom Rini	46	6.8%
Michal Simek	36	5.4%
Lokesh Vutla	35	5.2%
Alexander Graf	32	4.8%
Masahiro Yamada	28	4.2%
Heinrich Schuchardt	27	4.0%
Stephen Warren	19	2.8%
Nishanth Menon	19	2.8%
Bin Meng	15	2.2%
Joe Hershberger	12	1.8%
Chris Packham	11	1.6%
Patrice Chotard	11	1.6%
Mike Frysinger	10	1.5%
Philipp Tomsich	8	1.2%
Anatolij Gustschin	8	1.2%
Simon Goldschmidt	8	1.2%
Hans de Goede	7	1.0%
York Sun	6	0.9%
Jagan Teki	6	0.9%
Jean-Jacques Hiblot	6	0.9%
Roger Quadros	6	0.9%
Kim Phillips	6	0.9%
Marek Vasut	5	0.7%

Note: Top 25 bug reports till v2019.10-rc3

Bringing in new developers



U-Boot Version	Change Set	No.Of Developers	No.Of Employers
v2019.07	2047	215	28
v2019.04	1193	182	25
v2019.01	1149	140	25
v2018.11	1105	130	28
v2018.09	983	138	28
v2018.07	1055	141	27
v2018.05	977	128	23
v2018.03	1193	151	25
v2018.01	785	132	29
v2017.11	989	123	27
v2017.09	1308	130	25
v2017.07	1371	129	27
v2017.05	915	139	26
v2017.03	664	126	26
v2017.01	883	136	24
v2016.11	1031	114	22

U-Boot Build Systems

- Kbuild
- Kconfig
- Tools

Kbuild

Gitlab master or dev tree at
https://gitlab.com/u-boot/u-boot

Custodian's or Maintainers tree at
https://gitlab.denx.de/u-boot/custodians

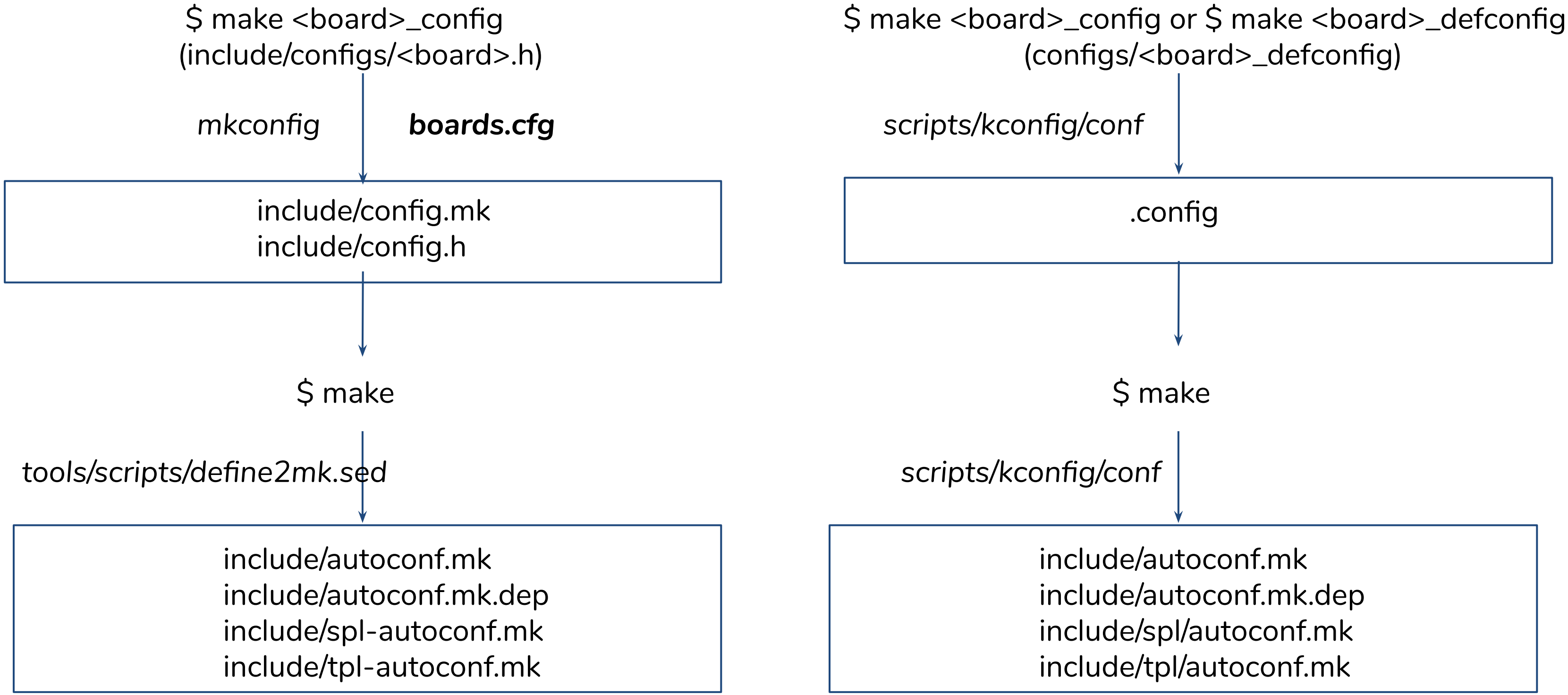
Example of building vyasa RK3288 board

```
$ git clone git://gitlab.com/u-boot/u-boot.git
$ u-boot
$ export ARCH=arm
$ export CROSS_COMPILE=arm-linux-gnueabi-
$ make vyasa-rk3288_defconfig
$ make
```

```
scripts/kconfig/conf  --syncconfig Kconfig
 CHK      include/config.h
 UPD      include/config.h
 CFG      u-boot.cfg
 GEN      include/autoconf.mk
 GEN      include/autoconf.mk.dep
 CFG      spl/u-boot.cfg
 GEN      spl/include/autoconf.mk
 CFG      tpl/u-boot.cfg
 GEN      tpl/include/autoconf.mk
 CHK      include/config/uboot.release
 UPD      include/config/uboot.release
 CHK      include/generated/version_autogenerated.h
 UPD      include/generated/version_autogenerated.h
 CHK      include/generated/timestamp_autogenerated.h

LD        arch/arm/cpu/built-in.o
CC        arch/arm/cpu/armv7/cache_v7.o
AS        arch/arm/cpu/armv7/cache_v7_asm.o
 CC       arch/arm/cpu/armv7/cpu.o
 CC       arch/arm/cpu/armv7/cp15.o
 CC       arch/arm/cpu/armv7/syslib.o
 AS       arch/arm/cpu/armv7/sctlr.o
 AS       arch/arm/cpu/armv7/lowlevel_init.o
 LD       arch/arm/cpu/armv7/built-in.o
 AS       arch/arm/cpu/armv7/start.o
 CC       arch/arm/lib/eabi_compat.o
 AS       arch/arm/lib/crt0_arm_efi.o
 CC       arch/arm/lib/reloc_arm_efi.o
 CC       arch/arm/mach-rockchip/boot_mode.o
 CC       arch/arm/mach-rockchip/rk3288-board.o
 CC       arch/arm/mach-rockchip/sdram_common.o
 CC       arch/arm/mach-rockchip/rk_timer.o
 CC       arch/arm/mach-rockchip/rk3288/clk_rk3288.o
 CC       arch/arm/mach-rockchip/rk3288/rk3288.o
 CC       arch/arm/mach-rockchip/rk3288/syscon_rk3288.o
 CC       board/amarula/vyasa-rk3288/vyasa-rk3288.o
```

```
CC        spl/arch/arm/mach-rockchip/sdram_common.o
CC        spl/arch/arm/mach-rockchip/rk_timer.o
CC        spl/arch/arm/mach-rockchip/rk3288/clk_rk3288.o
CC        spl/arch/arm/mach-rockchip/rk3288/rk3288.o
CC        spl/arch/arm/mach-rockchip/rk3288/syscon_rk3288.o
CC        spl/arch/arm/mach-rockchip/bootrom.o
CC        spl/arch/arm/mach-rockchip/rk3288-board-spl.o
CC        spl/arch/arm/cpu/armv7/cache_v7.o
CC        spl/arch/arm/cpu/armv7/cpu.o
AS        spl/arch/arm/cpu/armv7/lowlevel_init.o
AS        spl/arch/arm/cpu/armv7/start.o
LD        spl/u-boot-spl
OBJCOPY spl/u-boot-spl-nodtb.bin
COPY      spl/u-boot-spl.dtb
CAT       spl/u-boot-spl-dtb.bin
COPY      spl/u-boot-spl.bin

CC        tpl/arch/arm/mach-rockchip/sdram_common.o
CC        tpl/arch/arm/mach-rockchip/rk_timer.o
CC        tpl/arch/arm/mach-rockchip/rk3288/clk_rk3288.o
CC        tpl/arch/arm/mach-rockchip/rk3288/rk3288.o
CC        tpl/arch/arm/mach-rockchip/rk3288/syscon_rk3288.o
CC        tpl/arch/arm/mach-rockchip/bootrom.o
CC        tpl/arch/arm/mach-rockchip/rk3288-board-tpl.o
CC        tpl/arch/arm/cpu/armv7/cache_v7.o
AS        tpl/arch/arm/cpu/armv7/cache_v7_asm.o
CC        tpl/arch/arm/cpu/armv7/cpu.o
CC        tpl/arch/arm/cpu/armv7/cp15.o
CC        tpl/arch/arm/cpu/armv7/syslib.o
AS        tpl/arch/arm/cpu/armv7/lowlevel_init.o
AS        tpl/arch/arm/cpu/armv7/start.o
LDS       tpl/u-boot-spl.lds
LD        tpl/u-boot-tpl
OBJCOPY tpl/u-boot-tpl-nodtb.bin
COPY      tpl/u-boot-tpl.bin
COPY      u-boot.dtb
MKIMAGE u-boot-dtb.img
```

Kconfig



$ make <board>_config
(include/configs/<board>.h)

mkconfig **boards.cfg**

include/config.mk
include/config.h

$ make

tools/scripts/define2mk.sed

include/autoconf.mk
include/autoconf.mk.dep
include/spl-autoconf.mk
include/tpl-autoconf.mk

$ make <board>_config or $ make <board>_defconfig
(configs/<board>_defconfig)

scripts/kconfig/conf

.config

$ make

scripts/kconfig/conf

include/autoconf.mk
include/autoconf.mk.dep
include/spl/autoconf.mk
include/tpl/autoconf.mk

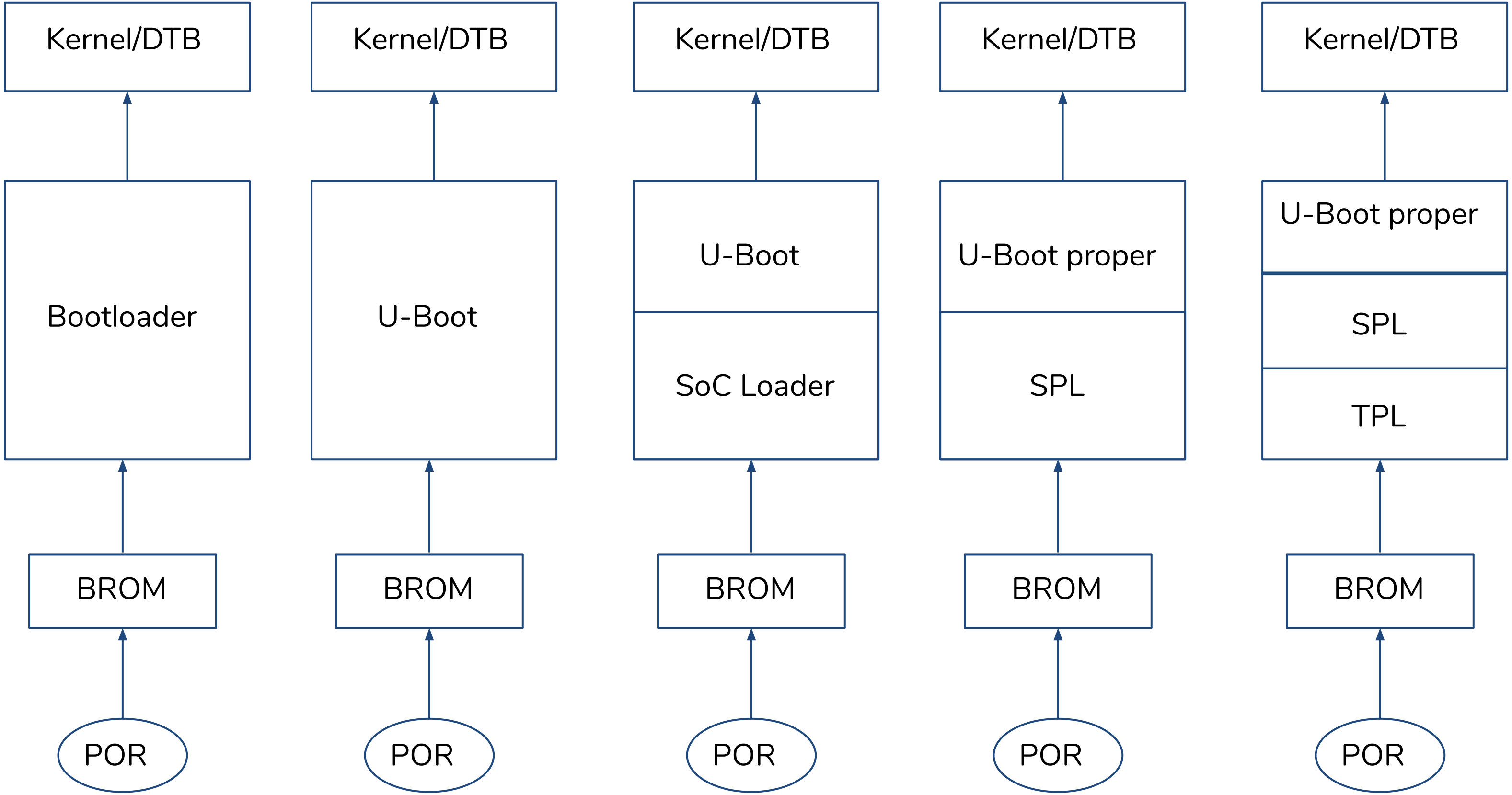
Tools: Build, Testing

- Patman
  - Manual patch creation, cover-letter, adding maintainers etc can be error-prone.
  - Create patch, insert cover-letter, add maintainer (via ~/.git-mailrc), run checkpatch.pl etc
  - How to use? tools/patman/README
- Buildman
  - U-Boot builder for multiple commits, branches etc
  - Replaced by legacy MAKEALL
  - Understandable output summary
  - tools/buildman/README

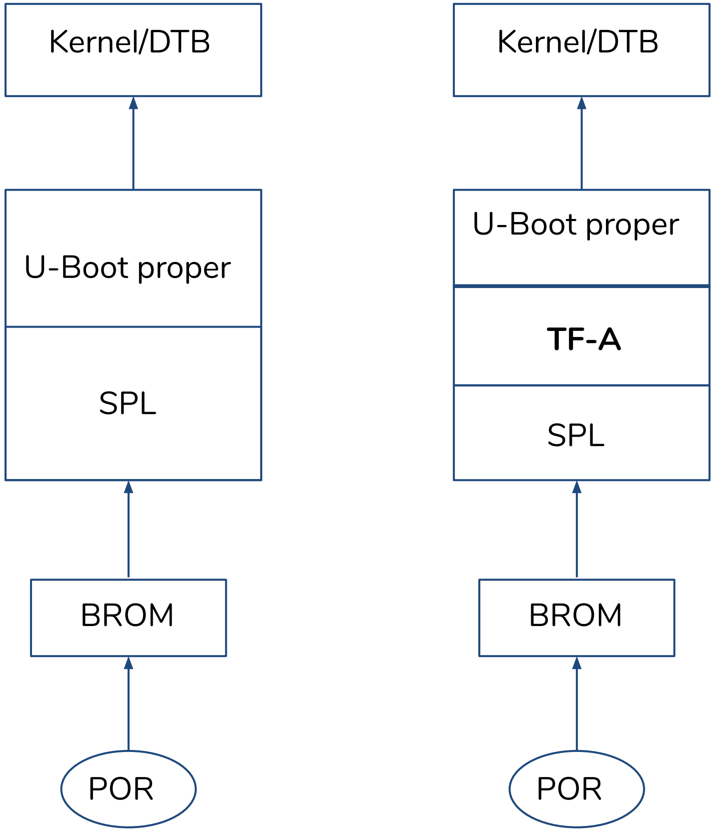
- travis-ci.org
  - Automated build environment, with limited run-time, free to use
  - May take longer duration, if more jobs are initiated
  - .travis.yml, u-boot travis build plugin
- test.py
  - Pytest framework
  - Works for sandbox, qemu, some real hardware
  - Sanity tests for dm code
  - doc/README.trace
- trace
  - Kind of Linux ftrace
  - Collect execution and sent to host for analysis

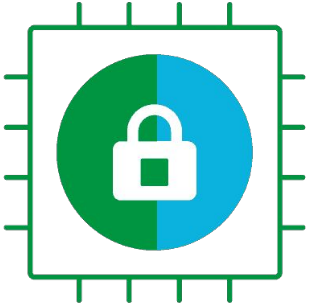
U-Boot, Stages

U-Boot

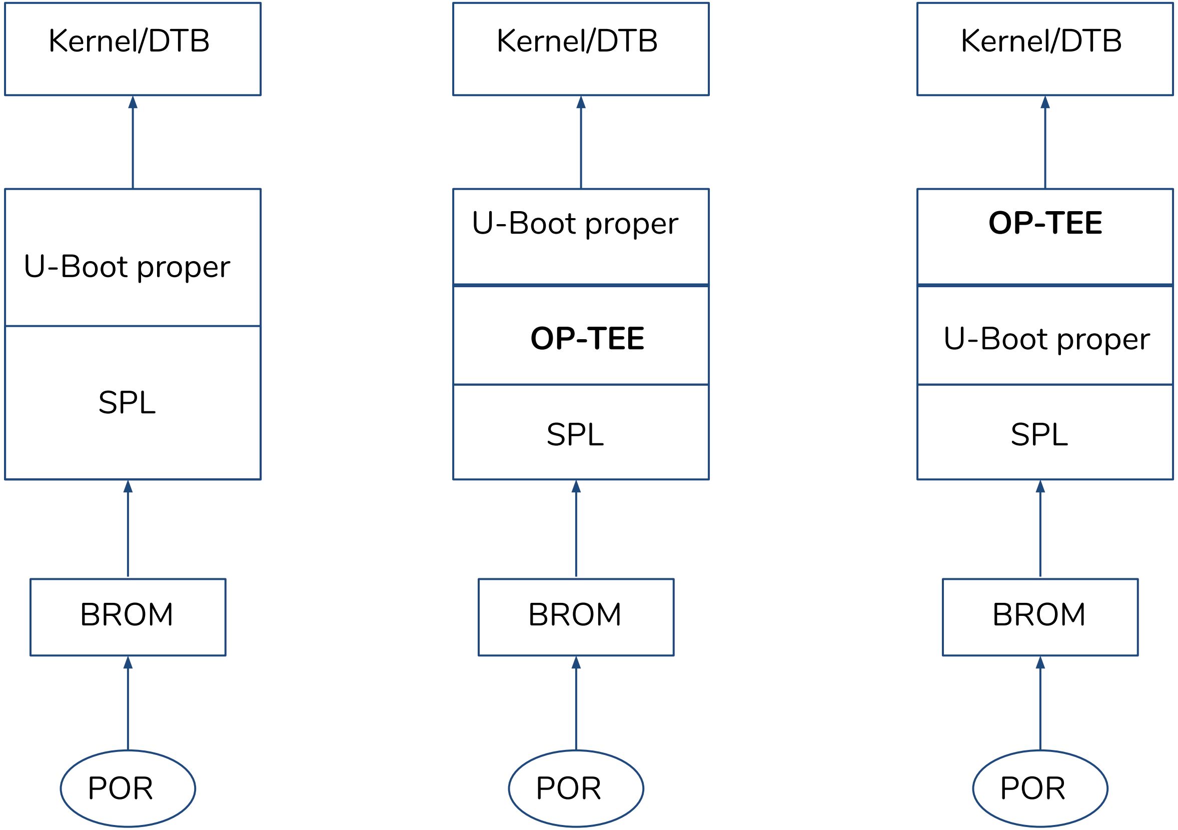

Kernel/DTB
Bootloader
BROM
POR
Kernel/DTB
U-Boot
BROM
POR
Kernel/DTB
U-Boot
SoC Loader
BROM
POR
Kernel/DTB
U-Boot proper
SPL
BROM
POR
Kernel/DTB
U-Boot proper
SPL
TPL
BROM
POR

TF-A

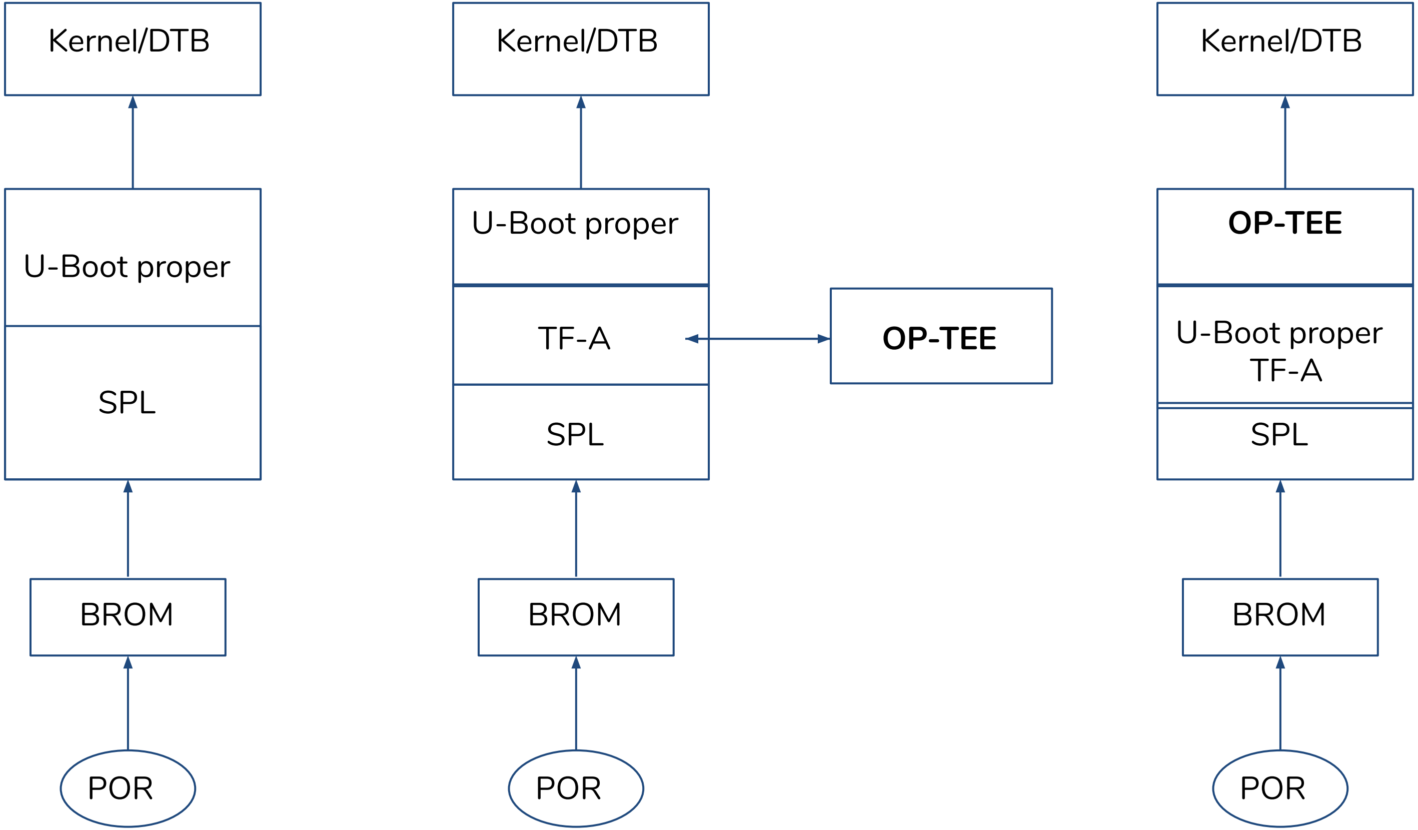

Kernel/DTB
U-Boot proper
SPL
BROM
POR
Kernel/DTB
U-Boot proper
TF-A
SPL
BROM
POR



OP-TEE, ARMv7


Kernel/DTB
U-Boot proper
SPL
BROM
POR
Kernel/DTB
U-Boot proper
OP-TEE
SPL
BROM
POR
Kernel/DTB
OP-TEE
U-Boot proper
SPL
BROM
POR

OP-TEE, ARMv8


Kernel/DTB
U-Boot proper
SPL
BROM
POR
Kernel/DTB
U-Boot proper
TF-A
SPL
BROM
POR
OP-TEE
Kernel/DTB
OP-TEE
U-Boot proper
TF-A
SPL
BROM
POR

U-Boot Sequence

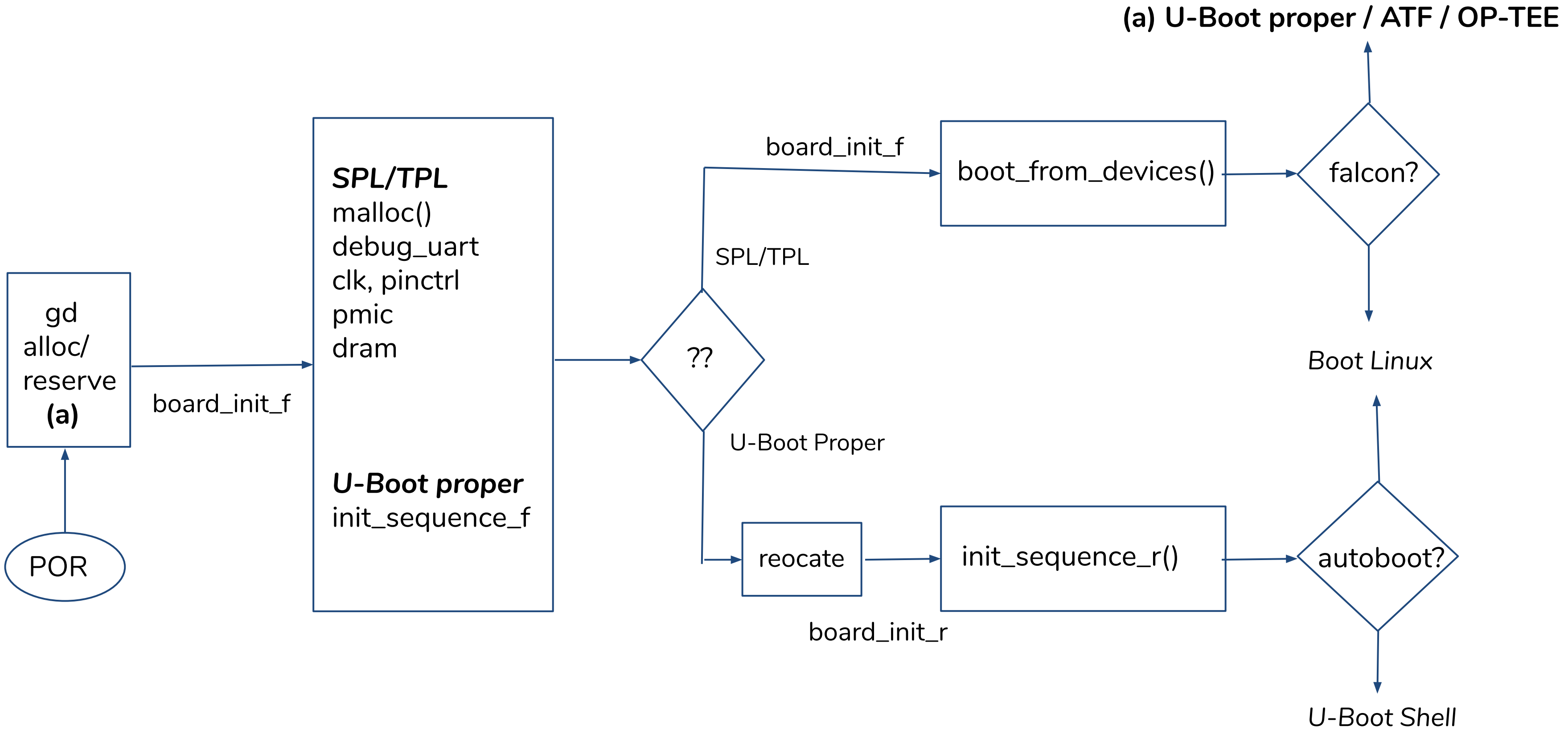


Image boot, Complexity

- ➔ Legacy image
- ➔ FIT
- ➔ Verified image
- ➔ Secure boot
- ➔ Falcon mode
- ➔ EFI boot
- ➔ Distro boot

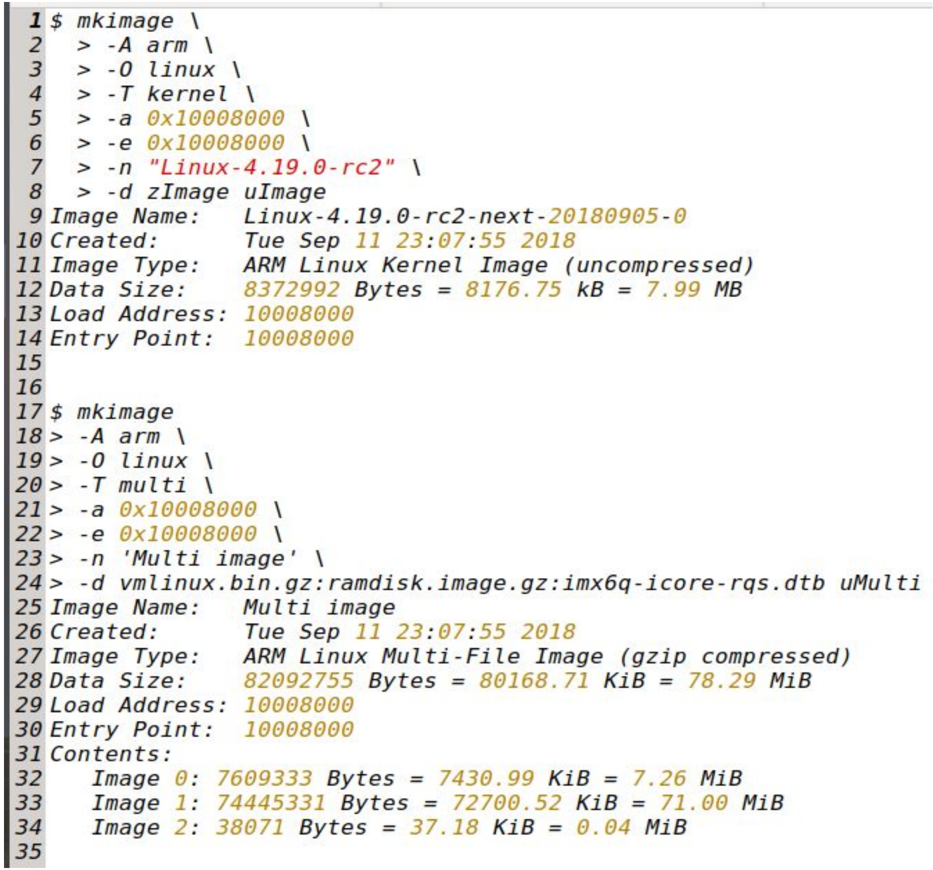
Legacy Image

- ➔ Fixed offset images - standalone, zImage binaries
- ➔ go addr [arg ...]
- ➔ u-boot Image format
- ➔ Single component uImage
- ➔ Monolithic, combination of images
- ➔ bootm [addr [arg ...]]

? Not flexible, indexing

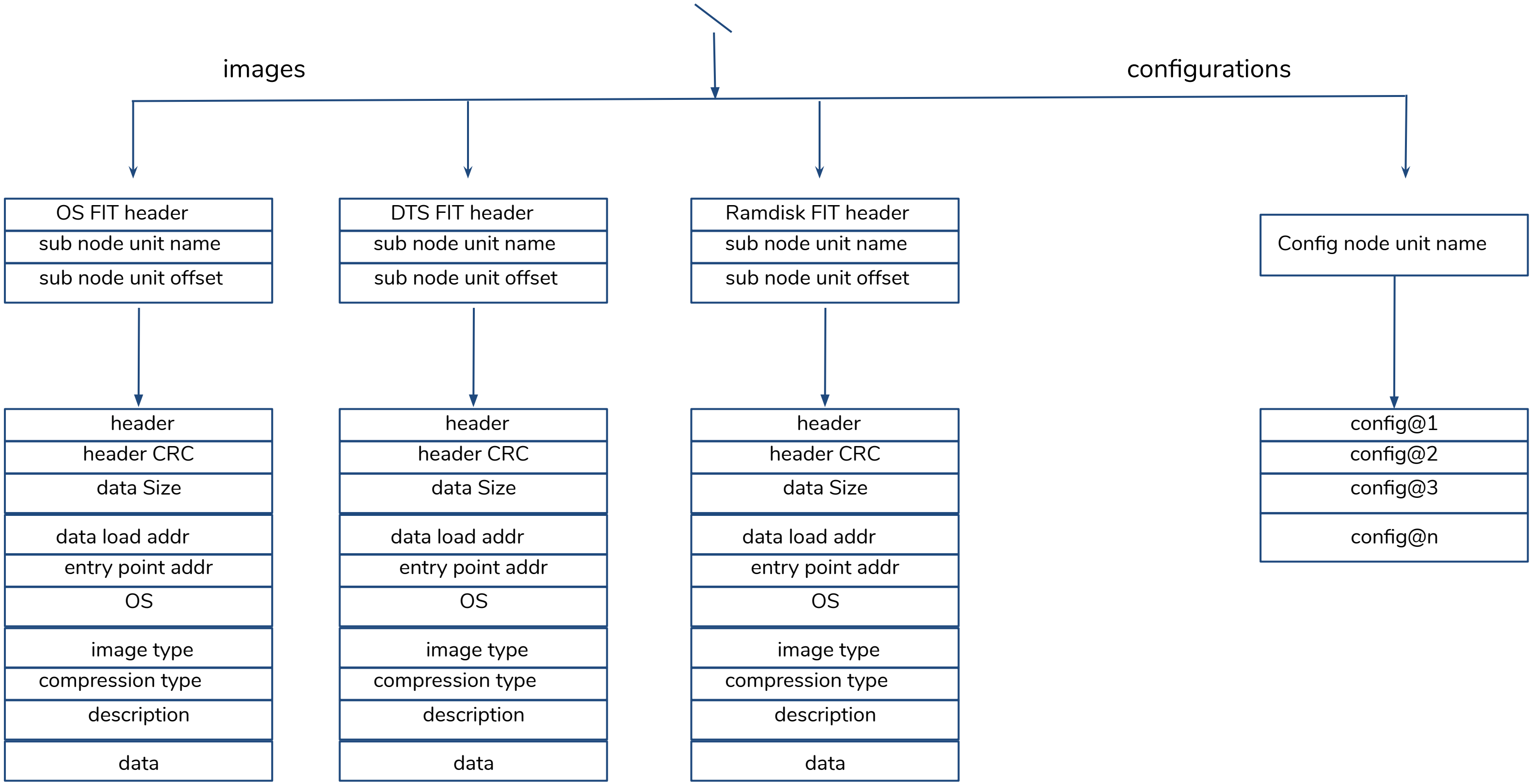
? No hash integrity

? No scope of security addition

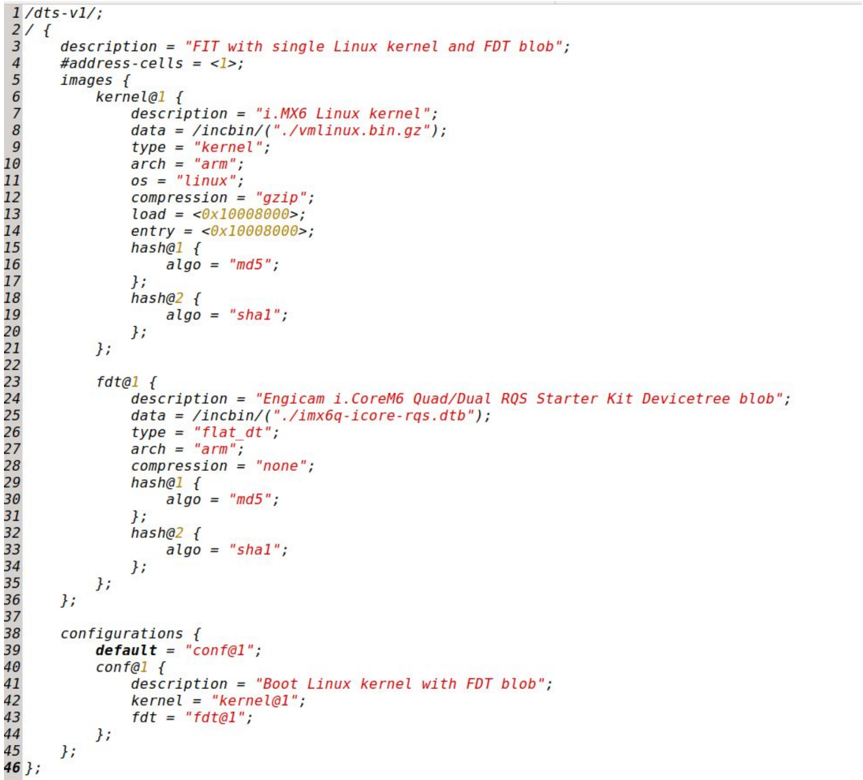


```
$ mkimage \
  > -A arm \
  > -O linux \
  > -T kernel \
  > -a 0x10008000 \
  > -e 0x10008000 \
  > -n "Linux-4.19.0-rc2" \
  > -d zImage uImage
Image Name:   Linux-4.19.0-rc2-next-20180905-0
Created:      Tue Sep 11 23:07:55 2018
Image Type:   ARM Linux Kernel Image (uncompressed)
Data Size:    8372992 Bytes = 8176.75 kB = 7.99 MB
Load Address: 10008000
Entry Point:  10008000


$ mkimage
> -A arm \
> -O linux \
> -T multi \
> -a 0x10008000 \
> -e 0x10008000 \
> -n 'Multi image' \
> -d vmlinux.bin.gz:ramdisk.image.gz:imx6q-icore-rqs.dtb uMulti
Image Name:   Multi image
Created:      Tue Sep 11 23:07:55 2018
Image Type:   ARM Linux Multi-File Image (gzip compressed)
Data Size:    82092755 Bytes = 80168.71 KiB = 78.29 MiB
Load Address: 10008000
Entry Point:  10008000
Contents:
   Image 0: 7609333 Bytes = 7430.99 KiB = 7.26 MiB
   Image 1: 74445331 Bytes = 72700.52 KiB = 71.00 MiB
   Image 2: 38071 Bytes = 37.18 KiB = 0.04 MiB

```

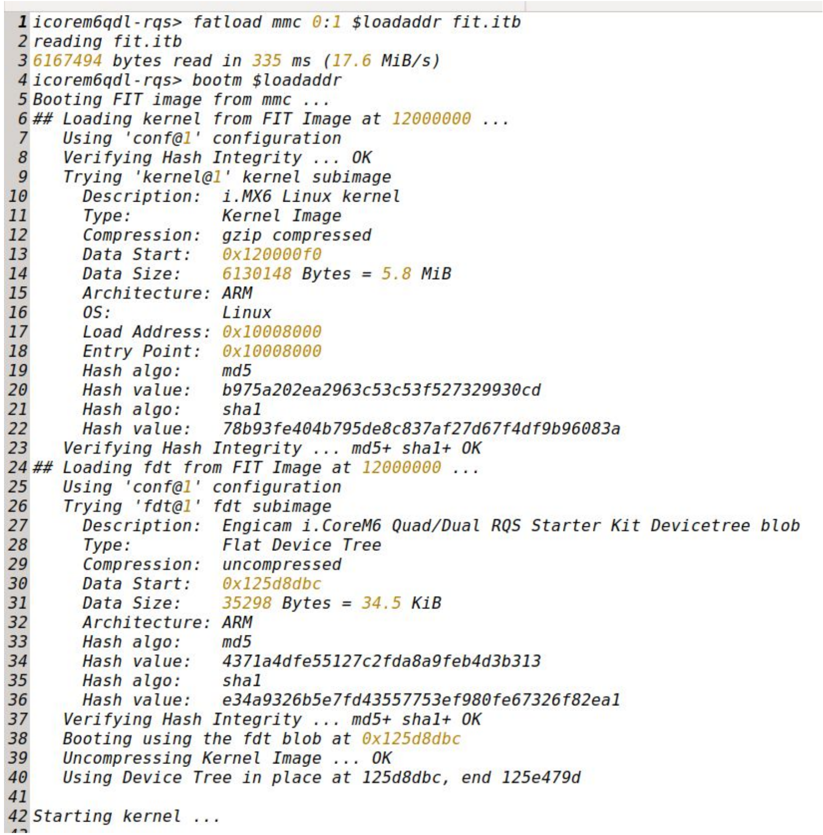
FIT (Flattened uImage Tree)



FIT, contd



```
/dts-v1/;
/ {
    description = "FIT with single Linux kernel and FDT blob";
    #address-cells = <1>;
    images {
        kernel@1 {
            description = "i.MX6 Linux kernel";
            data = /incbin/("./vmlinux.bin.gz");
            type = "kernel";
            arch = "arm";
            os = "linux";
            compression = "gzip";
            load = <0x10008000>;
            entry = <0x10008000>;
            hash@1 {
                algo = "md5";
            };
            hash@2 {
                algo = "sha1";
            };
        };

        fdt@1 {
            description = "Engicam i.CoreM6 Quad/Dual RQS Starter Kit Devicetree blob";
            data = /incbin/("./imx6q-icore-rqs.dtb");
            type = "flat_dt";
            arch = "arm";
            compression = "none";
            hash@1 {
                algo = "md5";
            };
            hash@2 {
                algo = "sha1";
            };
        };
    };

    configurations {
        default = "conf@1";
        conf@1 {
            description = "Boot Linux kernel with FDT blob";
            kernel = "kernel@1";
            fdt = "fdt@1";
        };
    };
};
```



```
icorem6qdl-rqs> fatload mmc 0:1 $loadaddr fit.itb
reading fit.itb
6167494 bytes read in 335 ms (17.6 MiB/s)
icorem6qdl-rqs> bootm $loadaddr
Booting FIT image from mmc ...
## Loading kernel from FIT Image at 12000000 ...
   Using 'conf@1' configuration
   Verifying Hash Integrity ... OK
   Trying 'kernel@1' kernel subimage
     Description:  i.MX6 Linux kernel
     Type:         Kernel Image
     Compression:  gzip compressed
     Data Start:   0x120000f0
     Data Size:    6130148 Bytes = 5.8 MiB
     Architecture: ARM
     OS:           Linux
     Load Address: 0x10008000
     Entry Point:  0x10008000
     Hash algo:    md5
     Hash value:   b975a202ea2963c53c53f527329930cd
     Hash algo:    sha1
     Hash value:   78b93fe404b795de8c837af27d67f4df9b96083a
   Verifying Hash Integrity ... md5+ sha1+ OK
## Loading fdt from FIT Image at 12000000 ...
   Using 'conf@1' configuration
   Trying 'fdt@1' fdt subimage
     Description:  Engicam i.CoreM6 Quad/Dual RQS Starter Kit Devicetree blob
     Type:         Flat Device Tree
     Compression:  uncompressed
     Data Start:   0x125d8dbc
     Data Size:    35298 Bytes = 34.5 KiB
     Architecture: ARM
     Hash algo:    md5
     Hash value:   4371a4dfe55127c2fda8a9feb4d3b313
     Hash algo:    sha1
     Hash value:   e34a9326b5e7fd43557753ef980fe67326f82ea1
   Verifying Hash Integrity ... md5+ sha1+ OK
   Booting using the fdt blob at 0x125d8dbc
   Uncompressing Kernel Image ... OK
   Using Device Tree in place at 125d8dbc, end 125e479d

Starting kernel ...
```

FIT Complexity ?

Multiple kernel, dtb, ramdisk

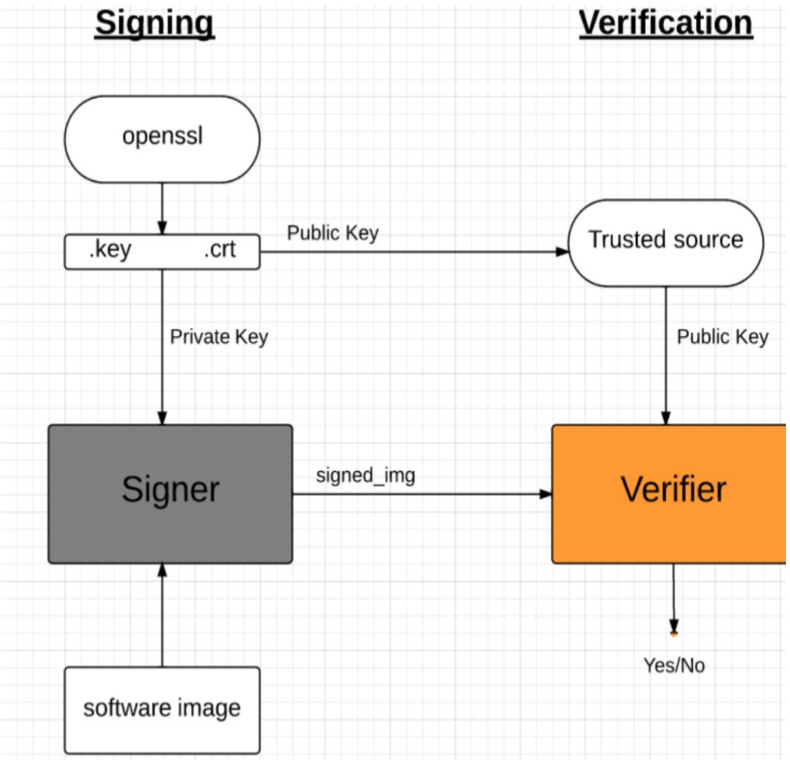
bit file, fpga

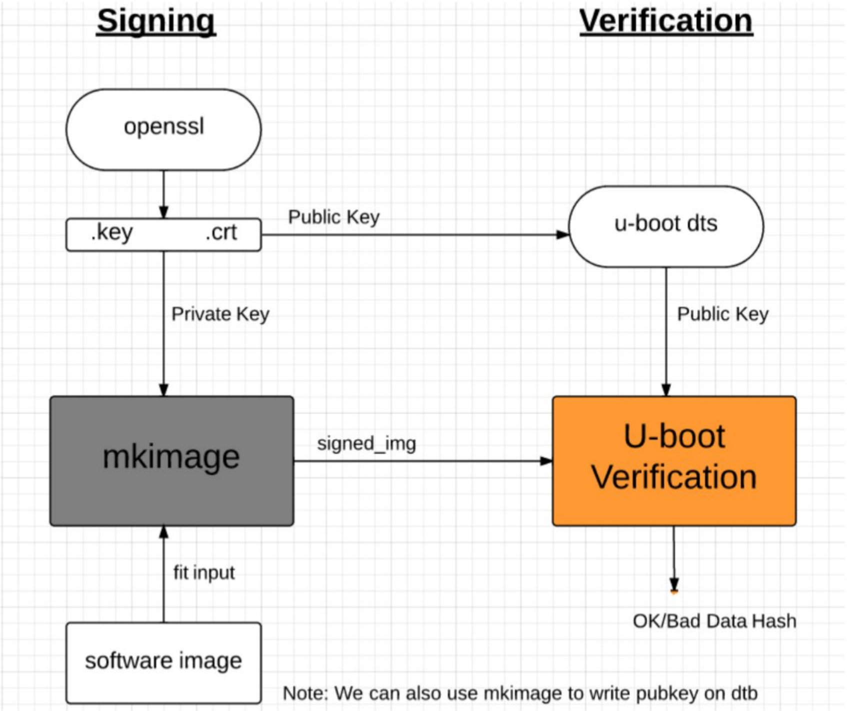
ATF

OPTEE

new image type?

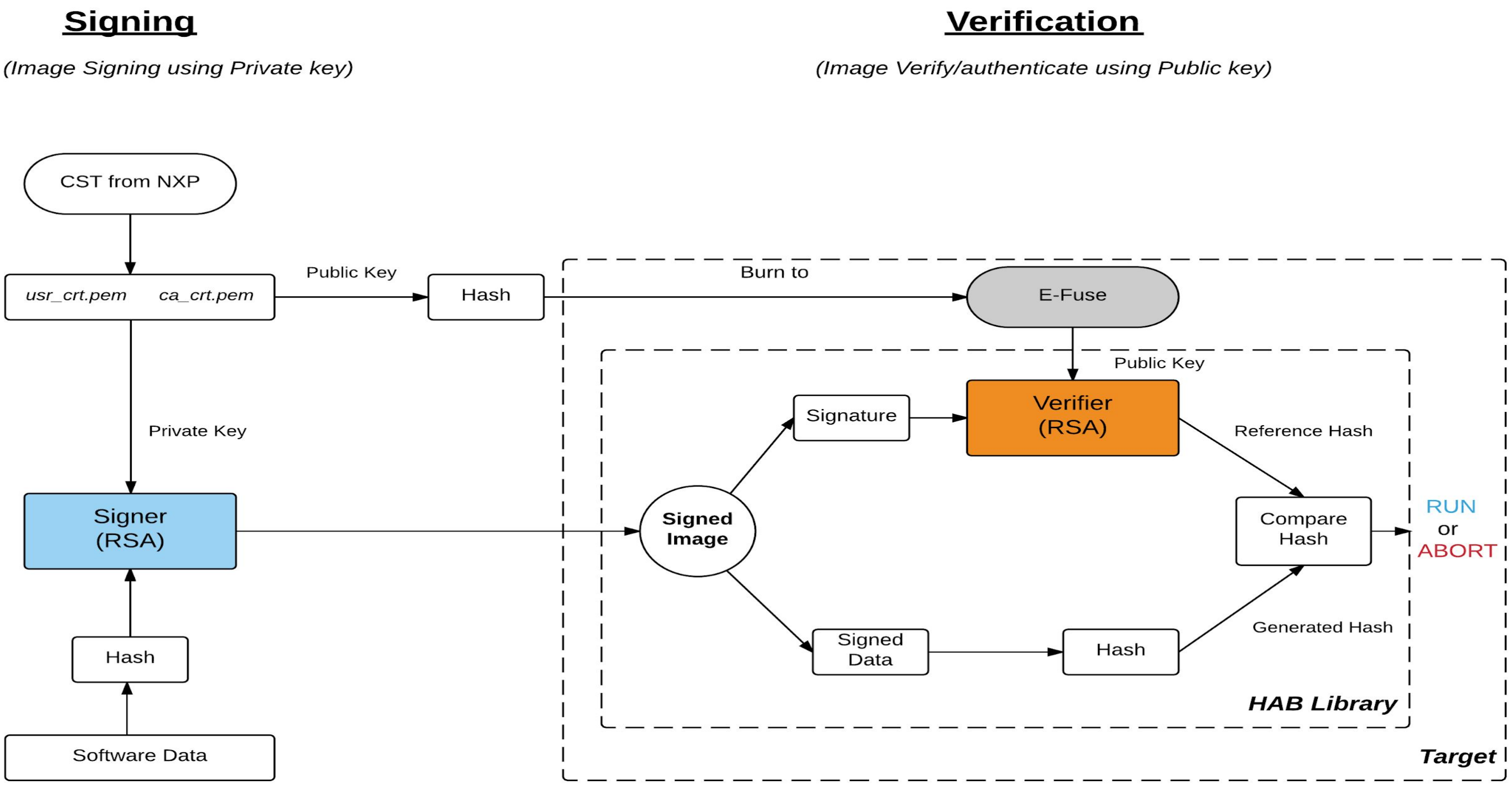
Verified boot (Software)



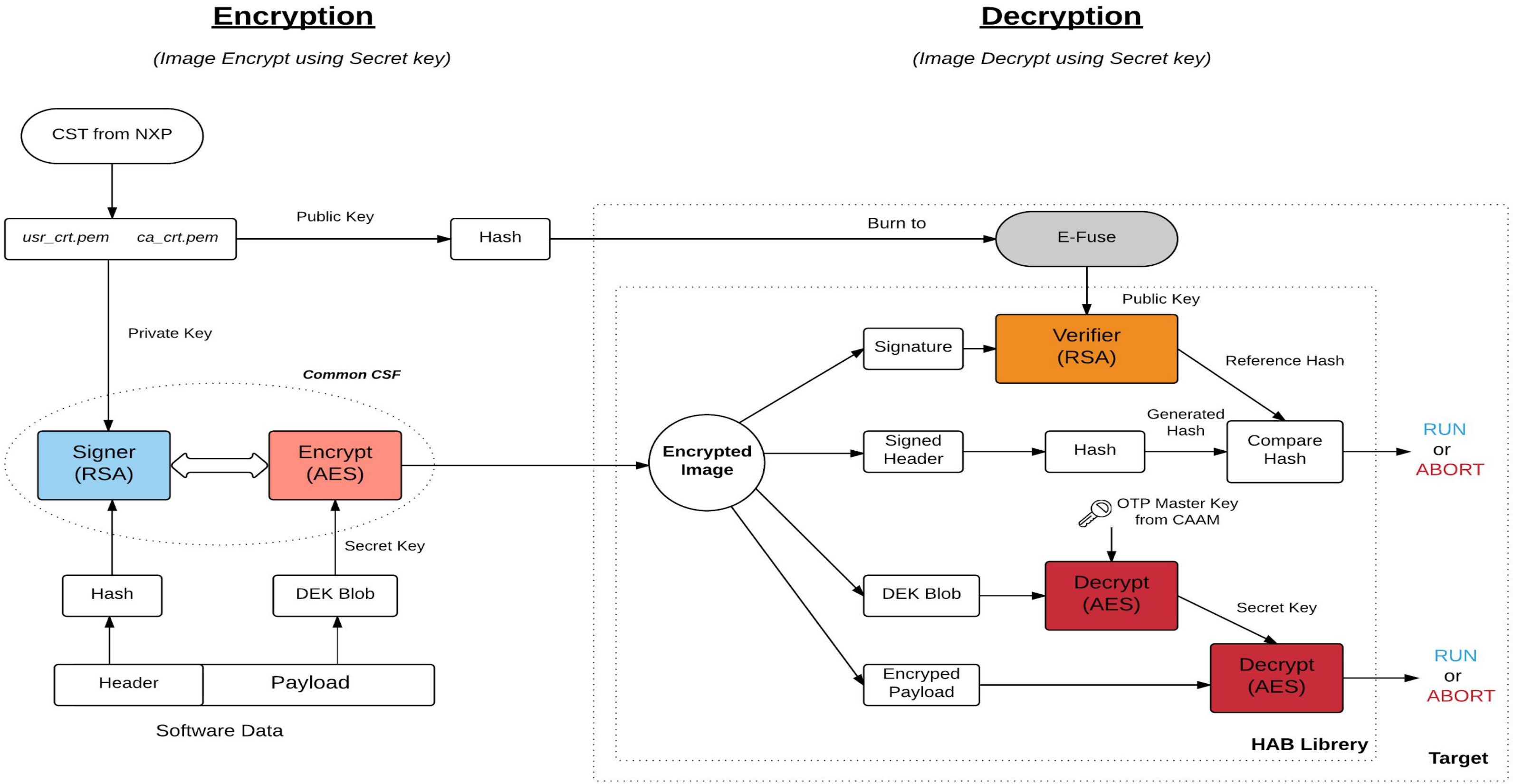


Note: We can also use mkimage to write pubkey on dtb

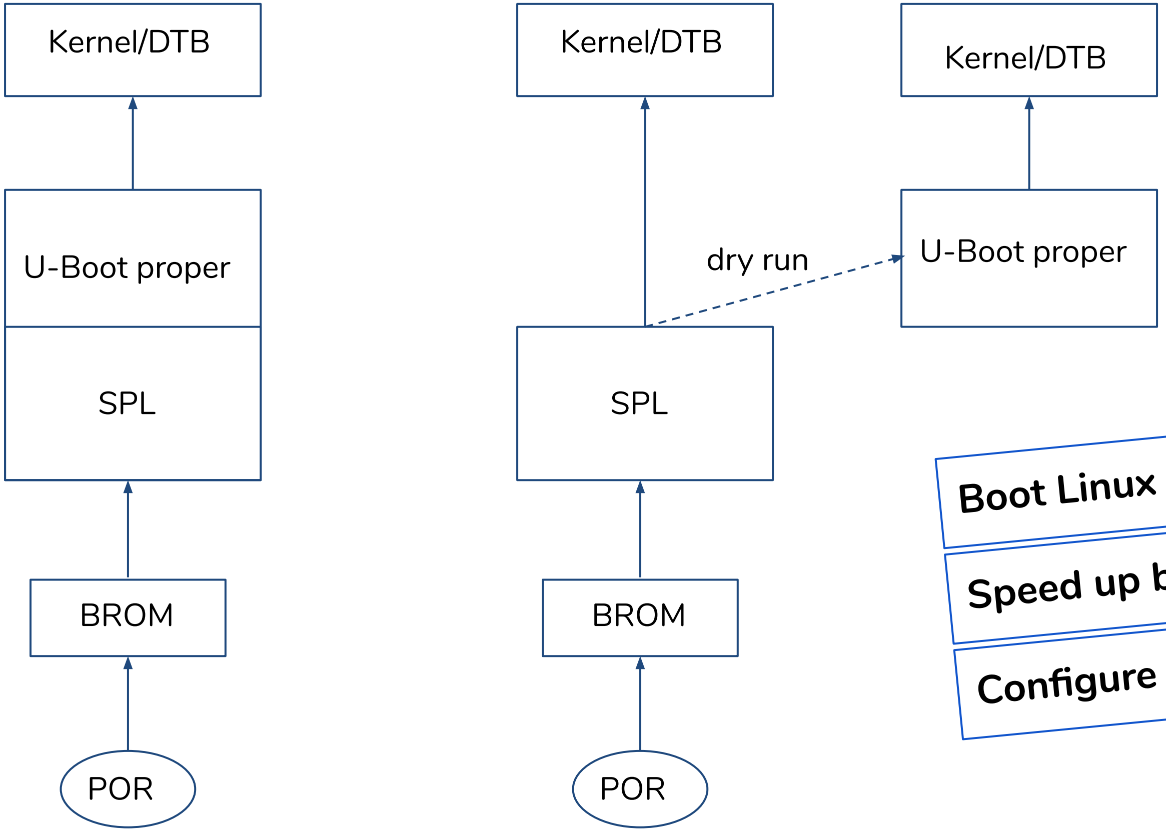
Secure boot, Signed



Secure boot, Encrypted



Falcon mode

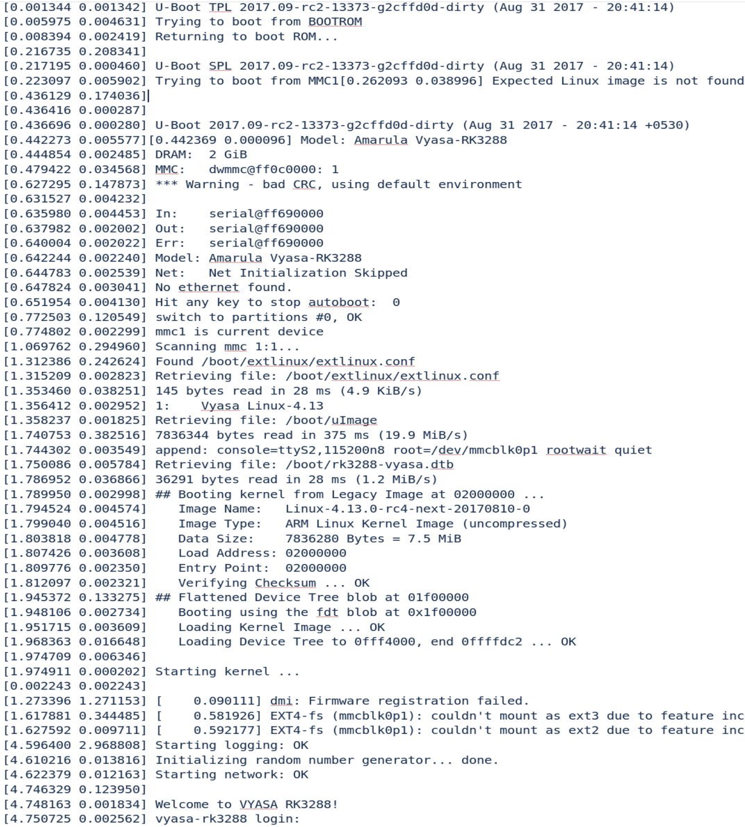


Boot Linux from SPL

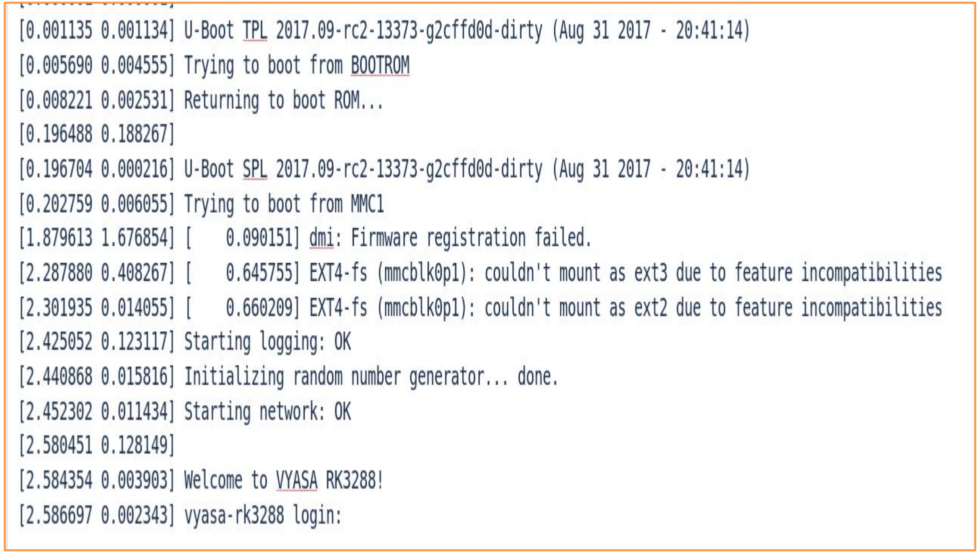
Speed up boot

Configure falcon, during dryrun

Falcon, is useful?



```
[0.001344 0.001342] U-Boot TPL 2017.09-rc2-13373-g2cffd0d-dirty (Aug 31 2017 - 20:41:14)
[0.005975 0.004631] Trying to boot from BOOTROM
[0.008394 0.002419] Returning to boot ROM...
[0.216735 0.208341]
[0.217195 0.000460] U-Boot SPL 2017.09-rc2-13373-g2cffd0d-dirty (Aug 31 2017 - 20:41:14)
[0.223097 0.005902] Trying to boot from MMC1[0.262093 0.038996] Expected Linux image is not found. Trying to start U-boot
[0.436129 0.174036]|
[0.436416 0.000287]
[0.436696 0.000280] U-Boot 2017.09-rc2-13373-g2cffd0d-dirty (Aug 31 2017 - 20:41:14 +0530)
[0.442273 0.005577][0.442369 0.000096] Model: Amarula Vyasa-RK3288
[0.444854 0.002485] DRAM:  2 GiB
[0.479422 0.034568] MMC:   dwmmc@ff0c0000: 1
[0.627295 0.147873] *** Warning - bad CRC, using default environment
[0.631527 0.004232]
[0.635980 0.004453] In:    serial@ff690000
[0.637982 0.002002] Out:   serial@ff690000
[0.640004 0.002022] Err:   serial@ff690000
[0.642244 0.002240] Model: Amarula Vyasa-RK3288
[0.644783 0.002539] Net:   Net Initialization Skipped
[0.647824 0.003041] No ethernet found.
[0.651954 0.004130] Hit any key to stop autoboot:  0
[0.772503 0.120549] switch to partitions #0, OK
[0.774802 0.002299] mmc1 is current device
[1.069762 0.294960] Scanning mmc 1:1...
[1.312386 0.242624] Found /boot/extlinux/extlinux.conf
[1.315209 0.002823] Retrieving file: /boot/extlinux/extlinux.conf
[1.353460 0.038251] 145 bytes read in 28 ms (4.9 KiB/s)
[1.356412 0.002952] 1:    Vyasa Linux-4.13
[1.358237 0.001825] Retrieving file: /boot/uImage
[1.740753 0.382516] 7836344 bytes read in 375 ms (19.9 MiB/s)
[1.744302 0.003549] append: console=ttyS2,115200n8 root=/dev/mmcblk0p1 rootwait quiet
[1.750086 0.005784] Retrieving file: /boot/rk3288-vyasa.dtb
[1.786952 0.036866] 36291 bytes read in 28 ms (1.2 MiB/s)
[1.789950 0.002998] ## Booting kernel from Legacy Image at 02000000 ...
[1.794524 0.004574]    Image Name:   Linux-4.13.0-rc4-next-20170810-0
[1.799040 0.004516]    Image Type:   ARM Linux Kernel Image (uncompressed)
[1.803818 0.004778]    Data Size:    7836280 Bytes = 7.5 MiB
[1.807426 0.003608]    Load Address: 02000000
[1.809776 0.002350]    Entry Point:  02000000
[1.812097 0.002321]    Verifying Checksum ... OK
[1.945372 0.133275] ## Flattened Device Tree blob at 01f00000
[1.948106 0.002734]    Booting using the fdt blob at 0x1f00000
[1.951715 0.003609]    Loading Kernel Image ... OK
[1.968363 0.016648]    Loading Device Tree to 0fff4000, end 0ffffdc2 ... OK
[1.974709 0.006346]
[1.974911 0.000202] Starting kernel ...
[0.002243 0.002243]
[1.273396 1.271153] [    0.090111] dmi: Firmware registration failed.
[1.617881 0.344485] [    0.581926] EXT4-fs (mmcblk0p1): couldn't mount as ext3 due to feature incompatibilities
[1.627592 0.009711] [    0.592177] EXT4-fs (mmcblk0p1): couldn't mount as ext2 due to feature incompatibilities
[4.596400 2.968808] Starting logging: OK
[4.610216 0.013816] Initializing random number generator... done.
[4.622379 0.012163] Starting network: OK
[4.746329 0.123950]
[4.748163 0.001834] Welcome to VYASA RK3288!
[4.750725 0.002562] vyasa-rk3288 login:
```



```
[0.001135 0.001134] U-Boot TPL 2017.09-rc2-13373-g2cffd0d-dirty (Aug 31 2017 - 20:41:14)
[0.005690 0.004555] Trying to boot from BOOTROM
[0.008221 0.002531] Returning to boot ROM...
[0.196488 0.188267]
[0.196704 0.000216] U-Boot SPL 2017.09-rc2-13373-g2cffd0d-dirty (Aug 31 2017 - 20:41:14)
[0.202759 0.006055] Trying to boot from MMC1
[1.879613 1.676854] [    0.090151] dmi: Firmware registration failed.
[2.287880 0.408267] [    0.645755] EXT4-fs (mmcblk0p1): couldn't mount as ext3 due to feature incompatibilities
[2.301935 0.014055] [    0.660209] EXT4-fs (mmcblk0p1): couldn't mount as ext2 due to feature incompatibilities
[2.425052 0.123117] Starting logging: OK
[2.440868 0.015816] Initializing random number generator... done.
[2.452302 0.011434] Starting network: OK
[2.580451 0.128149]
[2.584354 0.003903] Welcome to VYASA RK3288!
[2.586697 0.002343] vyasa-rk3288 login:
```

EFI boot

- ➔ Most of AArch64 and x86 UEFI is default booting system.
- ➔ U-Boot support via EFI_LOADER, CMD_BOOTEFI configs.

```
=> load mmc 0:2 ${fdt_addr_r} boot/dtb
29830 bytes read in 14 ms (2 MiB/s)
=> load mmc 0:1 ${kernel_addr_r} efi/debian/grubaa64.efi
reading efi/debian/grubaa64.efi
120832 bytes read in 7 ms (16.5 MiB/s)
=> bootefi ${kernel_addr_r} ${fdt_addr_r}
```

doc/README.uefi

ELCE 2017: Marrying U-Boot, UEFI and grub, Alexander Graf

Distro boot

- ➔ *bootargs* - system boot environment commands
- ➔ board-specific logic with respect to storage mechanism
- ➔ DISTRO_DEFAULTS, generic distribution configuration

```
# u-boot, extlinux.conf
label linux-5.0.0-rc3
  kernel /Image
  devicetree /sun50i-a64-amarula-relic.dtb
  append console=ttyS0,115200 earlyprintk root=/dev/mmcblk0p2 rootwait rw
```

Devicetree, improvements

- Kconfig
- FDT
- OF livetree
- FDT Overlay
- OF platdata

FDT (Flat Device Tree)

- Run-time hardware configuration
- Single U-Boot binary for multiple boards (with board controlled dts)
- Handle via libfdt
- Enabled via CONFIG_OF_CONTROL

- FDT supported U-Boot can build
  - with default dts, CONFIG_DEFAULT_DEVICE_TREE=<dts-file-name> in defconfig
    - $ make
  - with user-specified dts
    - $ make DEVICE_TREE=<dts-file-name>

- Sample, UniPhier Pro4 reference, Pro4 Ace, Pro4 Sanji boards
  - $ make uniphier_v7_defconfig (single configuration)
  - $ make DEVICE_TREE=uniphier-pro4-ref
  - $ make DEVICE_TREE=uniphier-pro4-ace
  - $ make DEVICE_TREE=uniphier-pro4-sanji

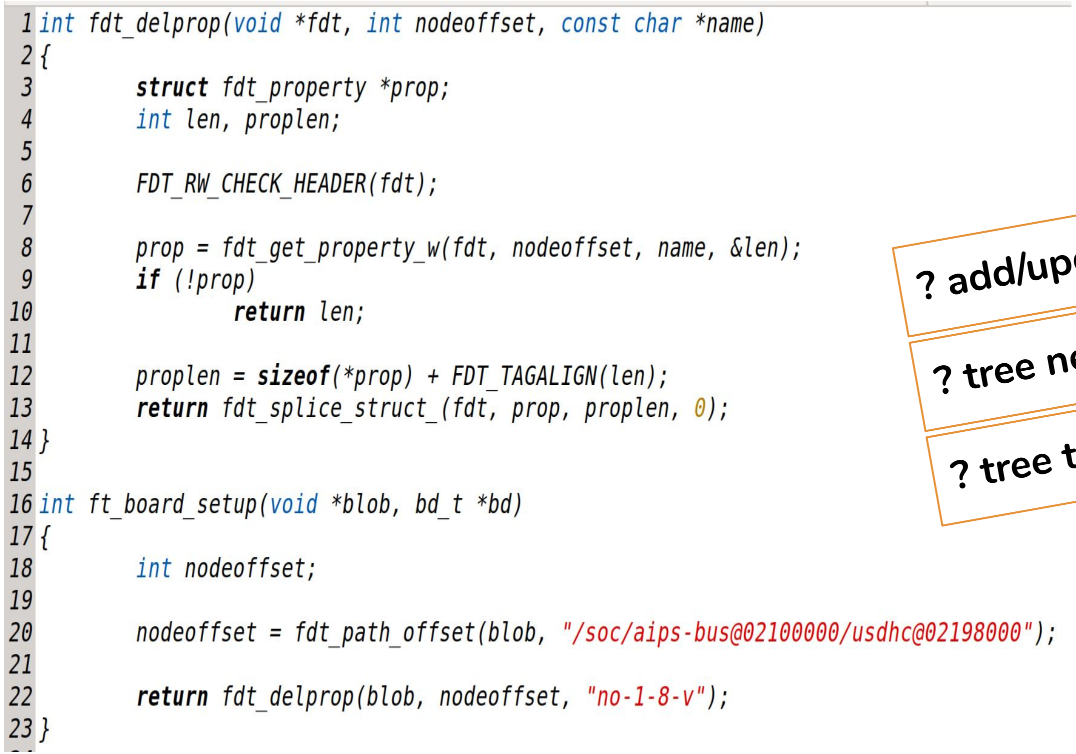
- DTB packing during build
  - CONFIG_OF_EMBED
  - CONFIG_OF_SEPARATE

FDT, u-boot

- ➔ Maintain U-Boot specific node definitions in separate file
- ➔ Useful for DT allocation in SPL
- ➔ u-boot,dm-pre-alloc, u-boot,dm-spl

```
/* arch/arm/dts/imx6qdl-u-boot.dtsi */
/ {
   soc {
        u-boot,dm-spl;
        aips-bus@02000000 {
           u-boot,dm-spl;
        };
   };
};

&gpio1 {
   u-boot,dm-spl;
};
```

FDT, libfdt



```c
int fdt_delprop(void *fdt, int nodeoffset, const char *name)
{
        struct fdt_property *prop;
        int len, proplen;

        FDT_RW_CHECK_HEADER(fdt);

        prop = fdt_get_property_w(fdt, nodeoffset, name, &len);
        if (!prop)
                return len;

        proplen = sizeof(*prop) + FDT_TAGALIGN(len);
        return fdt_splice_struct_(fdt, prop, proplen, 0);
}

int ft_board_setup(void *blob, bd_t *bd)
{
        int nodeoffset;

        nodeoffset = fdt_path_offset(blob, "/soc/aips-bus@02100000/usdhc@02198000");

        return fdt_delprop(blob, nodeoffset, "no-1-8-v");
}
```

? add/update, copy large amount

? tree need to rebuilt

? tree traversing is slow

Livetree (Live Device Tree)

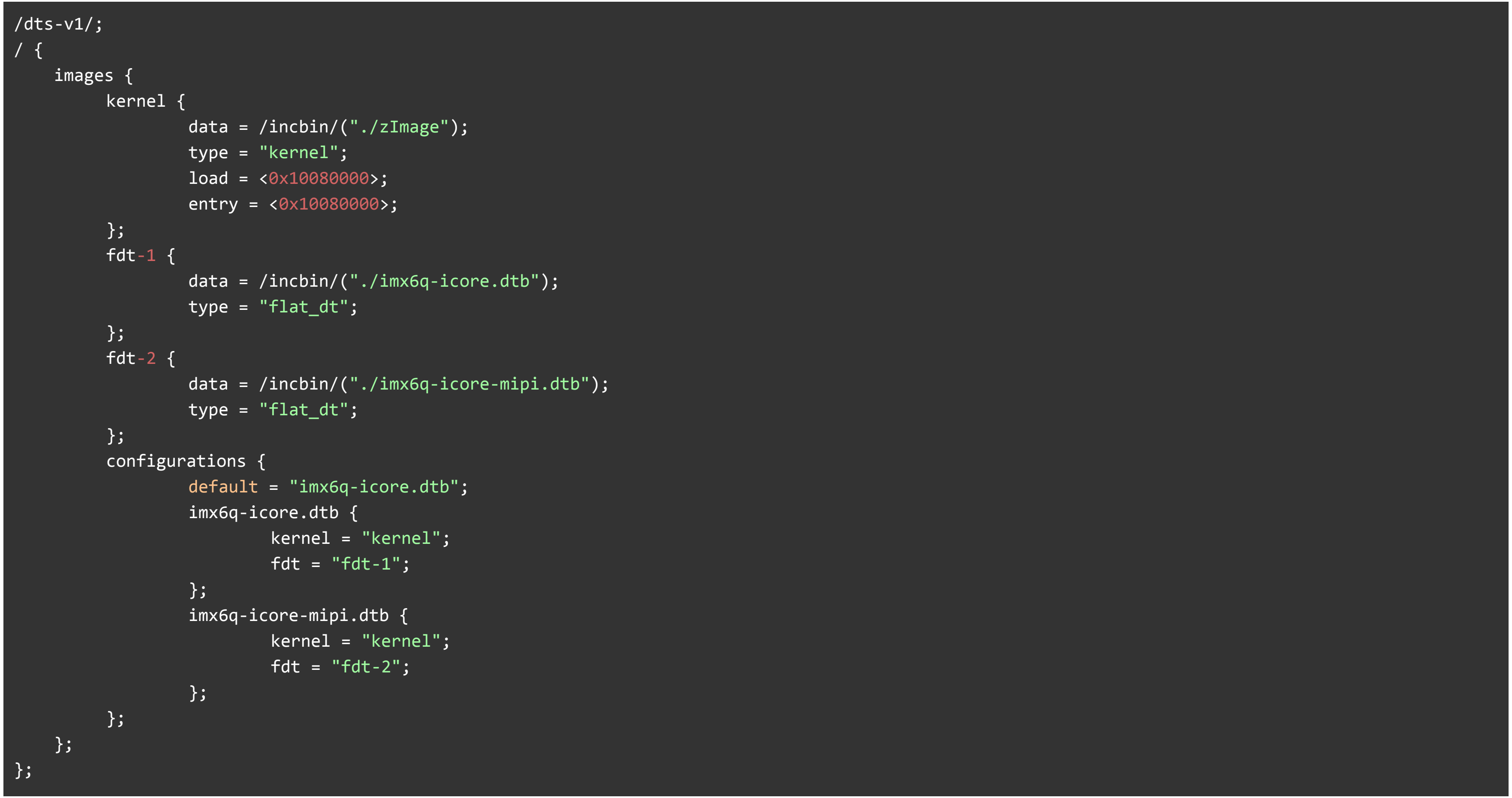
- ➔ Pointer-based *hierarchical structures*
- ➔ Support after relocation
- ➔ ofnode , point to either flat tree or livetree
- ➔ Enabled via CONFIG_OF_LIVE

```
static int zynq_spi_ofdata_to_platdata(struct udevice *bus)
{
     struct zynq_spi_platdata *plat = bus->platdata;

     /* old code */
     plat->regs = (struct zynq_spi_regs *)devfdt_get_addr(bus);
     plat->frequency = fdtdec_get_int(blob, node, "spi-max-frequency", 250000000);

     /* new code */
     plat->regs = (struct zynq_spi_regs *)dev_read_addr(bus);
     plat->frequency = dev_read_u32_default(bus, "spi-max-frequency", 250000000);

    return 0;
}
```

FDT Overlay

- DTO, enable centralize DTB to be overlaid on the device tree.
- Single image of multitude of similar boards and their expansion options
- **HAT** - Rasberry PI, Tinker board
- DTO can load U-Boot via
  - FIT image
  - Manual load

Pure class A amplifier with True Native DSD format and Analog volume control. Compatible with Raspberry PI and Tinker Board.



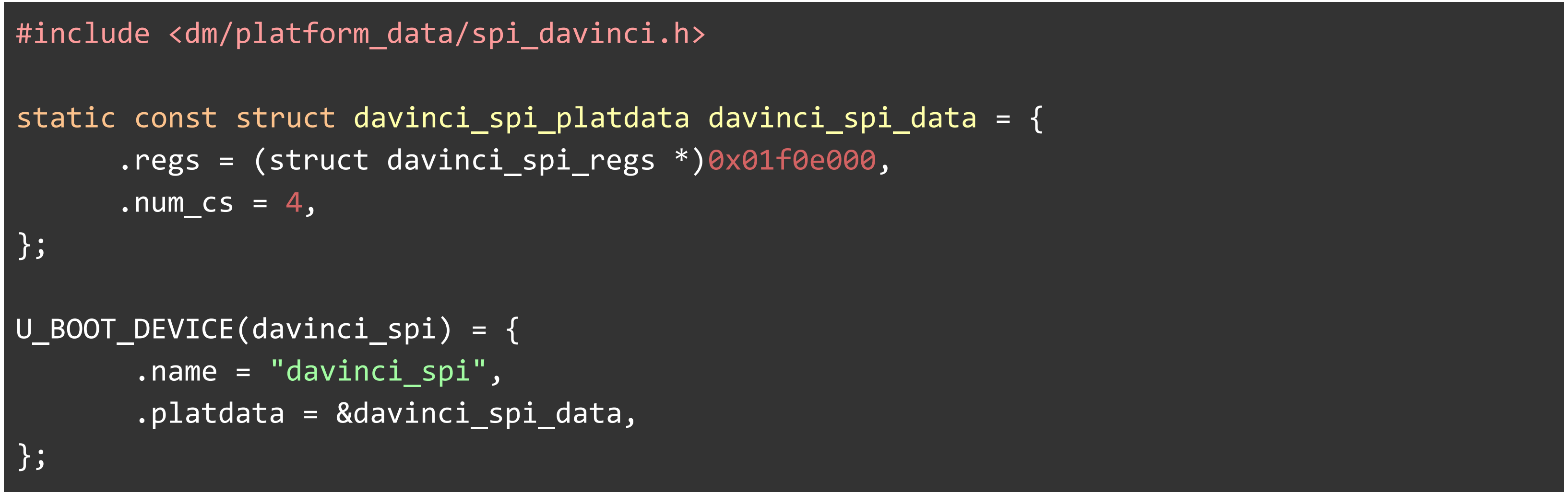
```
/dts-v1/;
/ {
    images {
        kernel {
                data = /incbin/("./zImage");
                type = "kernel";
                load = <0x10080000>;
                entry = <0x10080000>;
        };
        fdt-1 {
                data = /incbin/("./imx6q-icore.dtb");
                type = "flat_dt";
        };
        fdt-2 {
                data = /incbin/("./imx6q-icore-mipi.dtb");
                type = "flat_dt";
        };
        configurations {
                default = "imx6q-icore.dtb";
                imx6q-icore.dtb {
                        kernel = "kernel";
                        fdt = "fdt-1";
                };
                imx6q-icore-mipi.dtb {
                        kernel = "kernel";
                        fdt = "fdt-2";
                };
        };
    };
};
```

FDT Overlay, loading

```
/* via FIT */
=> bootm $loadaddr#imx6q-icore.dtb#imx6q-icore-mipi

/* Manual load */
=> setenv fdt_addr 0x18000000
=> setenv fdt_ovaddr 0x180c0000

=> load mmc 0:1 ${fdt_addr} ${bootdir}/base.dtb
=> load mmc 0:1 ${fdt_ovaddr} ${bootdir}/overlay.dtb

=> fdt resize 8192

=> fdt apply $fdt_ovaddr

=> bootm ${loadaddr} - ${fdt_addr}
```

OF Platdata

- SPL size increases with FDT
- Enabled via CONFIG_SPL_OF_PLATDATA
- Explicitly define the device details Like legacy platform_device in Linux



```
#include <dm/platform_data/spi_davinci.h>

static const struct davinci_spi_platdata davinci_spi_data = {
      .regs = (struct davinci_spi_regs *)0x01f0e000,
      .num_cs = 4,
};

U_BOOT_DEVICE(davinci_spi) = {
       .name = "davinci_spi",
       .platdata = &davinci_spi_data,
};
```

Driver model

- ➔ Driver model
- ➔ DM, CPU
- ➔ DM, Power
- ➔ DM, Core
- ➔ DM, Peripherals
- ➔ Block layer
- ➔ USB layer

Driver model

? ad-hoc model, direct functions call

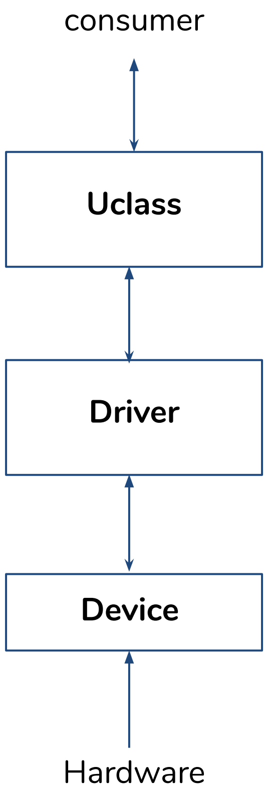
? multiple controllers can’t fit same driver

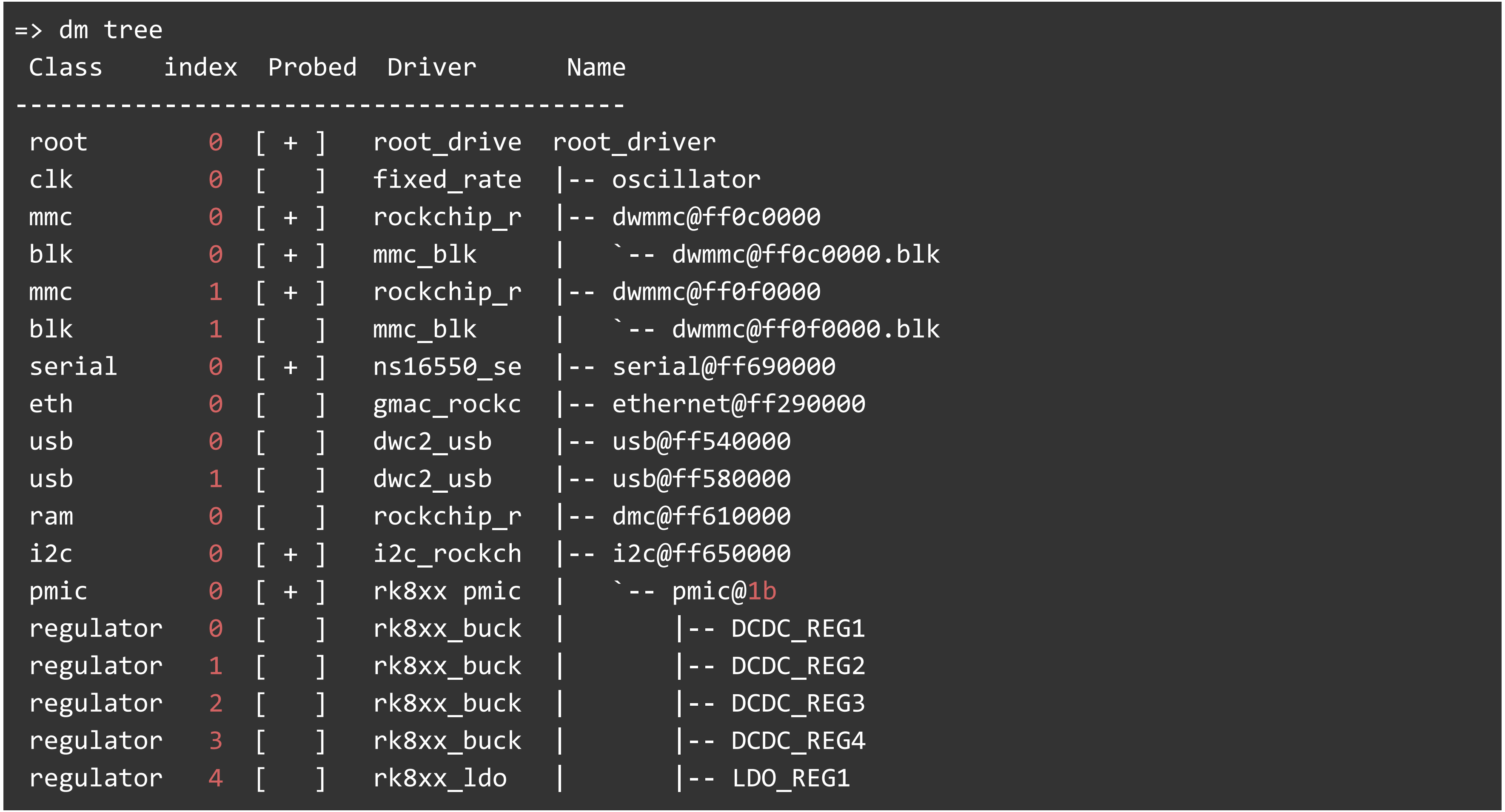
? not scalable, difficult to maintain

Simple, scalable, modular, homogeneous

Lazy initialization, but bounded

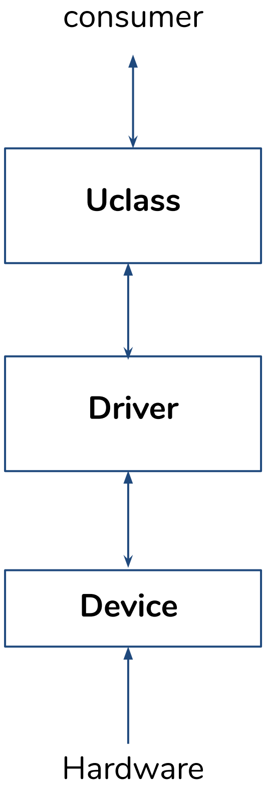
Small overhead, SPL



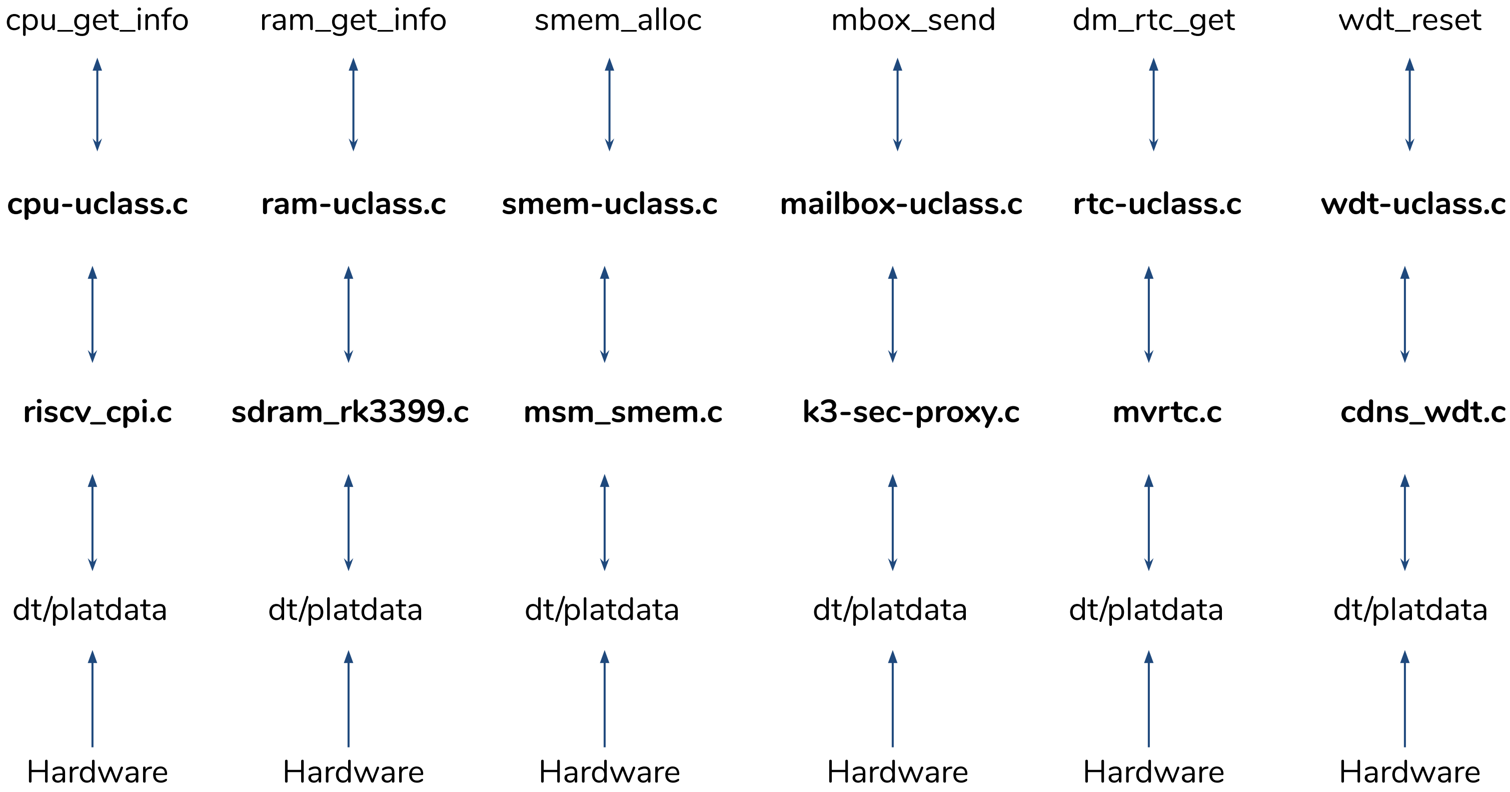


```
=> dm tree
 Class     index  Probed  Driver      Name
-----------------------------------------
 root         0  [ + ]   root_drive  root_driver
 clk          0  [   ]   fixed_rate  |-- oscillator
 mmc          0  [ + ]   rockchip_r  |-- dwmmc@ff0c0000
 blk          0  [ + ]   mmc_blk     |   `-- dwmmc@ff0c0000.blk
 mmc          1  [ + ]   rockchip_r  |-- dwmmc@ff0f0000
 blk          1  [   ]   mmc_blk     |   `-- dwmmc@ff0f0000.blk
 serial       0  [ + ]   ns16550_se  |-- serial@ff690000
 eth          0  [   ]   gmac_rockc  |-- ethernet@ff290000
 usb          0  [   ]   dwc2_usb    |-- usb@ff540000
 usb          1  [   ]   dwc2_usb    |-- usb@ff580000
 ram          0  [   ]   rockchip_r  |-- dmc@ff610000
 i2c          0  [ + ]   i2c_rockch  |-- i2c@ff650000
 pmic         0  [ + ]   rk8xx pmic  |   `-- pmic@1b
 regulator    0  [   ]   rk8xx_buck  |       |-- DCDC_REG1
 regulator    1  [   ]   rk8xx_buck  |       |-- DCDC_REG2
 regulator    2  [   ]   rk8xx_buck  |       |-- DCDC_REG3
 regulator    3  [   ]   rk8xx_buck  |       |-- DCDC_REG4
 regulator    4  [   ]   rk8xx_ldo   |       |-- LDO_REG1
```

Driver model



DM, CPU



cpu_get_info
cpu-uclass.c
riscv_cpi.c
dt/platdata
Hardware

ram_get_info
ram-uclass.c
sdram_rk3399.c
dt/platdata
Hardware

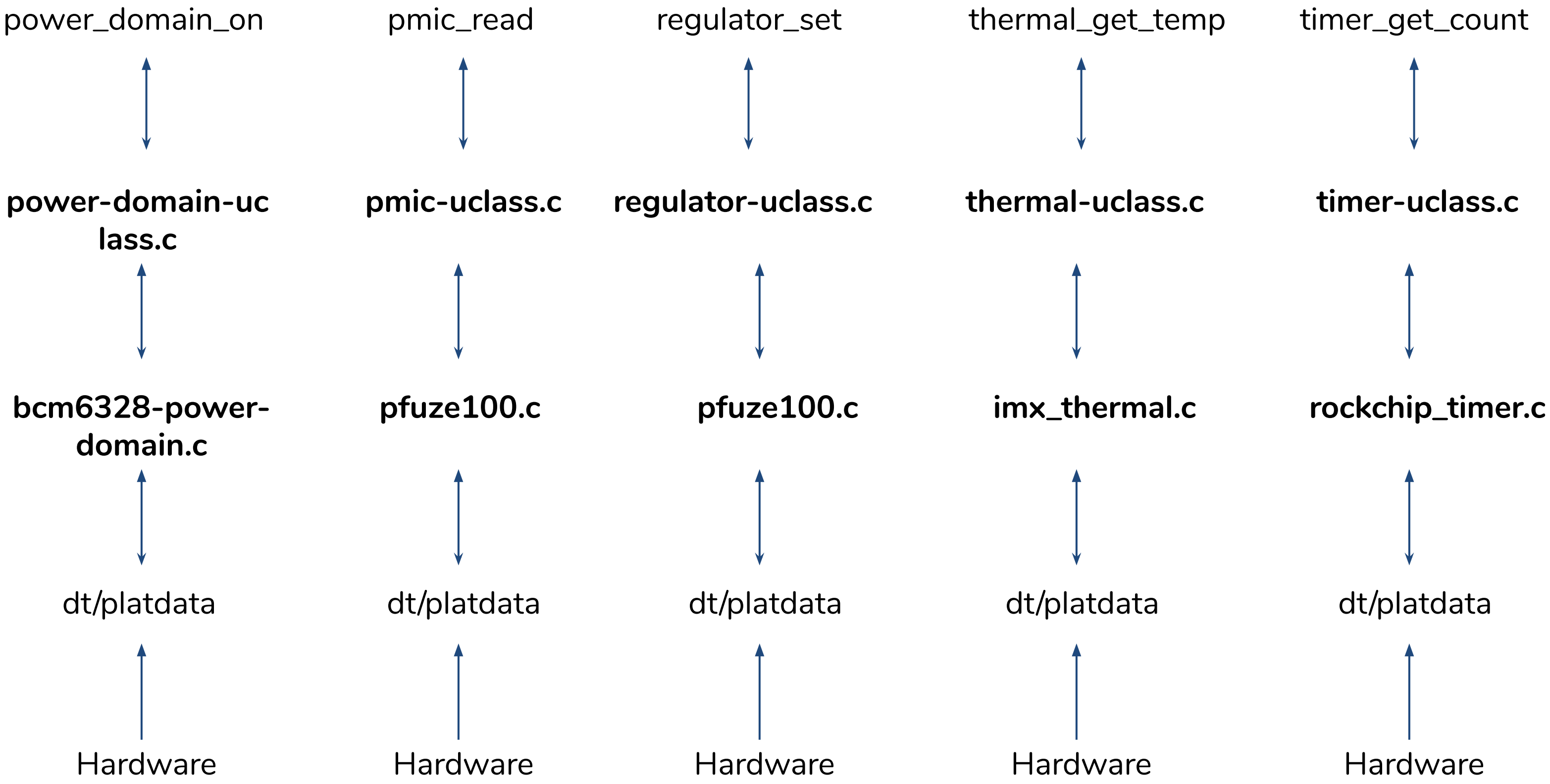
smem_alloc
smem-uclass.c
msm_smem.c
dt/platdata
Hardware

mbox_send
mailbox-uclass.c
k3-sec-proxy.c
dt/platdata
Hardware

dm_rtc_get
rtc-uclass.c
mvrtc.c
dt/platdata
Hardware

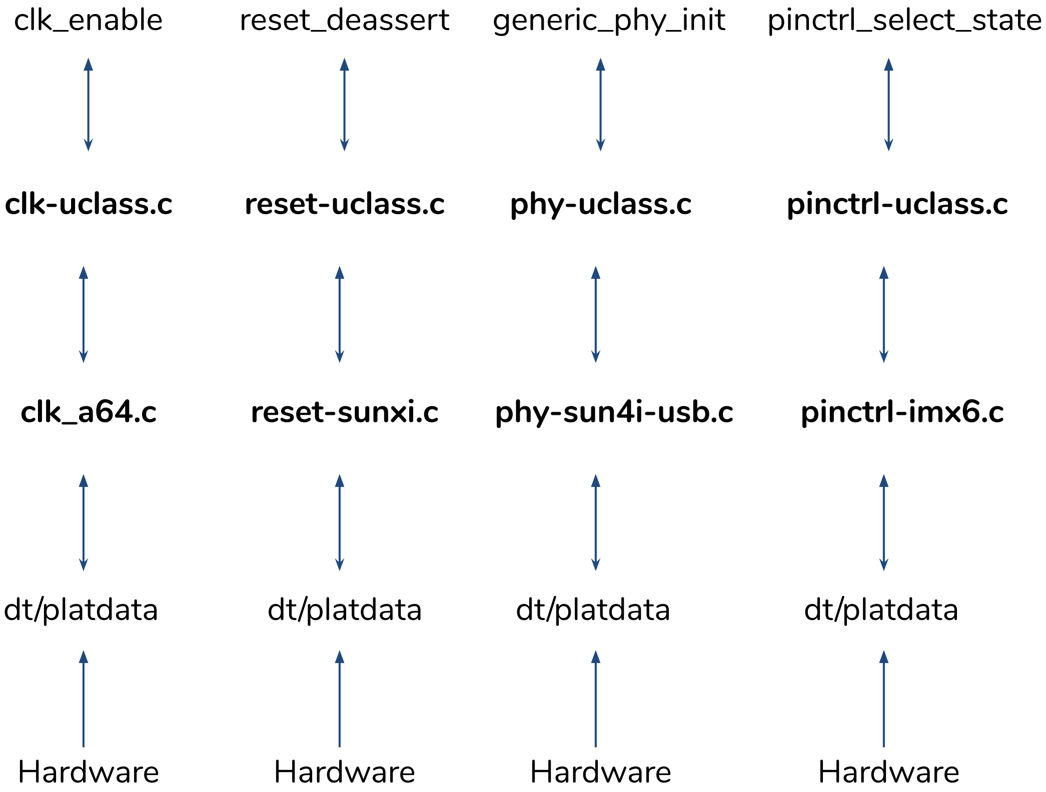
wdt_reset
wdt-uclass.c
cdns_wdt.c
dt/platdata
Hardware

DM, Power



power_domain_on	pmic_read	regulator_set	thermal_get_temp	timer_get_count
power-domain-uclass.c	**pmic-uclass.c**	**regulator-uclass.c**	**thermal-uclass.c**	**timer-uclass.c**
bcm6328-power-domain.c	**pfuze100.c**	**pfuze100.c**	**imx_thermal.c**	**rockchip_timer.c**
dt/platdata	dt/platdata	dt/platdata	dt/platdata	dt/platdata
Hardware	Hardware	Hardware	Hardware	Hardware

DM, Core



clk_enable

clk-uclass.c

clk_a64.c

dt/platdata

Hardware

reset_deassert

reset-uclass.c

reset-sunxi.c

dt/platdata

Hardware

generic_phy_init

phy-uclass.c

phy-sun4i-usb.c

dt/platdata

Hardware

pinctrl_select_state

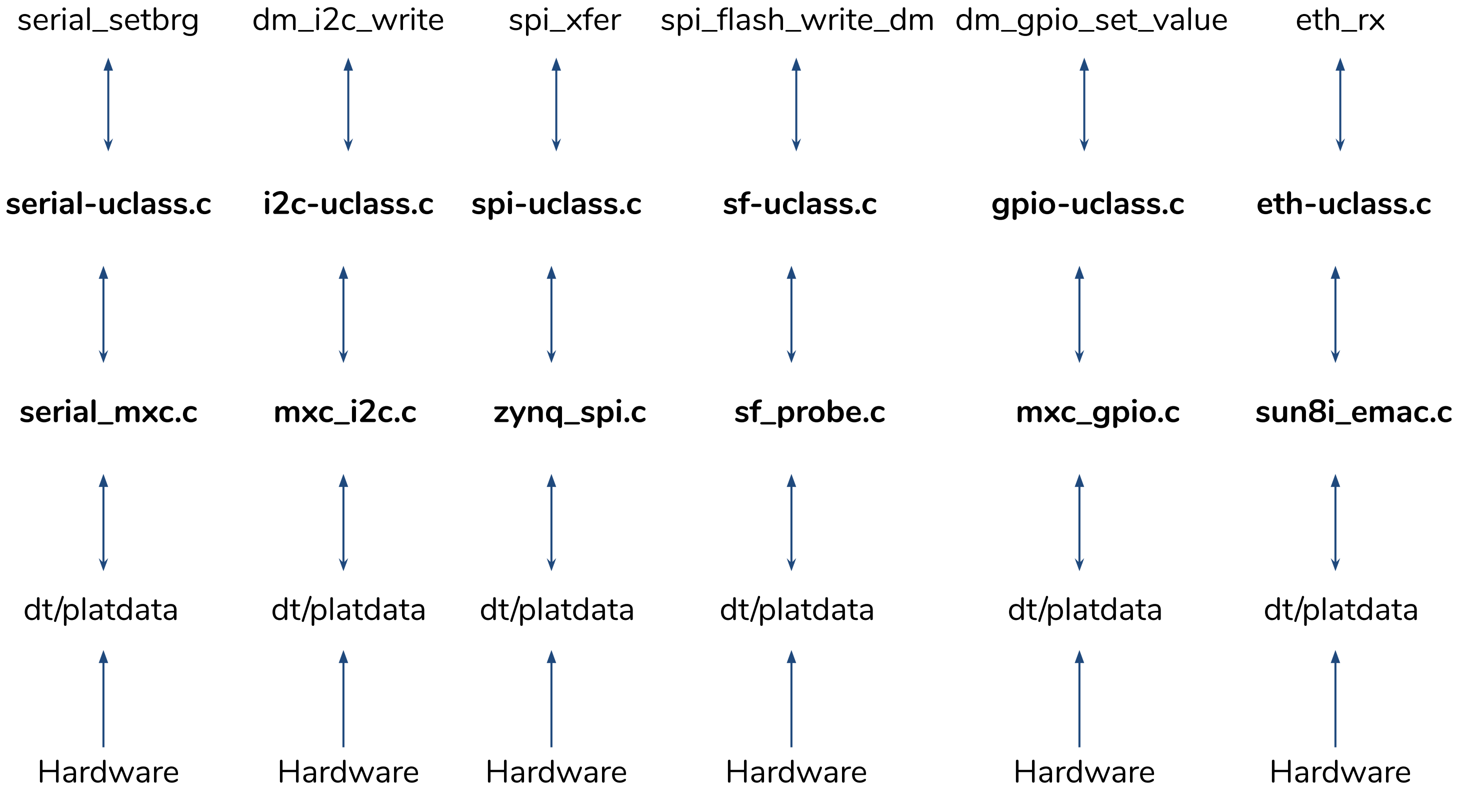
pinctrl-uclass.c

pinctrl-imx6.c

dt/platdata

Hardware

DM, Peripherals



serial_setbrg
serial-uclass.c
serial_mxc.c
dt/platdata
Hardware

dm_i2c_write
i2c-uclass.c
mxc_i2c.c
dt/platdata
Hardware

spi_xfer
spi-uclass.c
zynq_spi.c
dt/platdata
Hardware

spi_flash_write_dm
sf-uclass.c
sf_probe.c
dt/platdata
Hardware

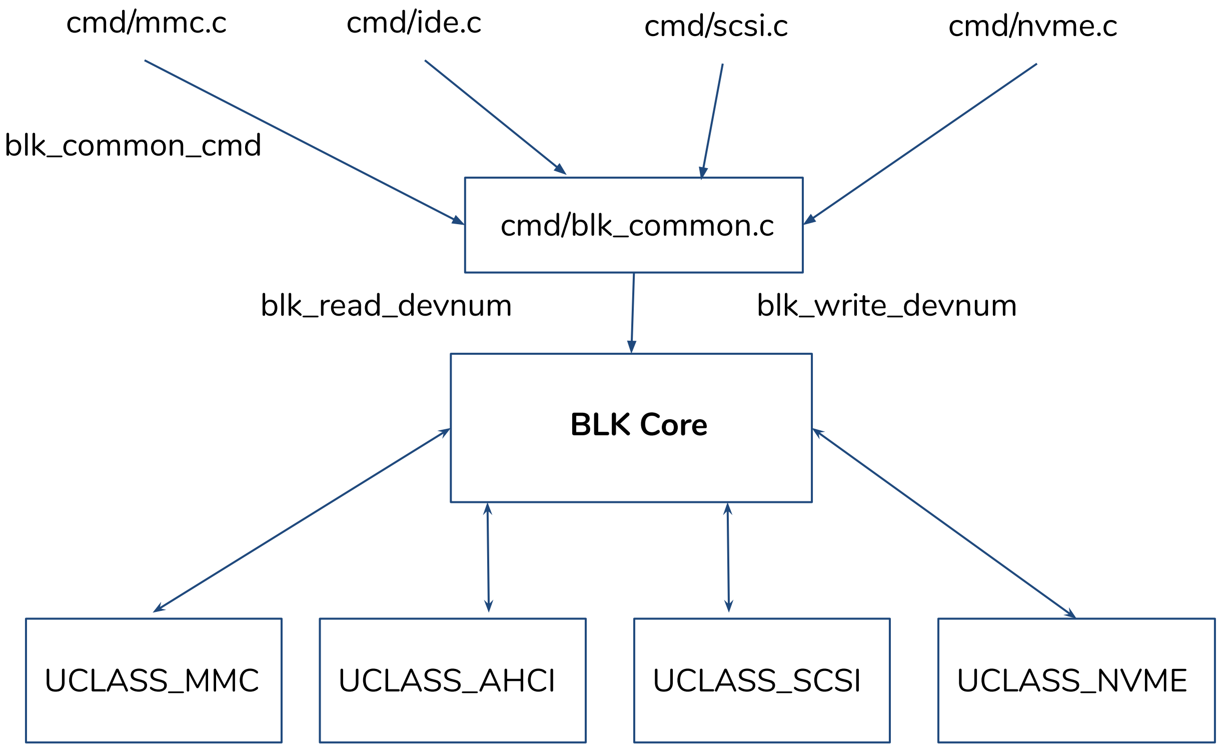
dm_gpio_set_value
gpio-uclass.c
mxc_gpio.c
dt/platdata
Hardware

eth_rx
eth-uclass.c
sun8i_emac.c
dt/platdata
Hardware

```
UCLASS_DRIVER(spi) = {
      .id             = UCLASS_SPI,
      .name           = "spi",
      .flags          = DM_UC_FLAG_SEQ_ALIAS,
      .post_bind      = dm_scan_fdt_dev,
      .post_probe     = spi_post_probe,
      .child_pre_probe = spi_child_pre_probe,
      .per_device_auto_alloc_size = sizeof(struct dm_spi_bus),
      .per_child_auto_alloc_size = sizeof(struct spi_slave),
      .per_child_platdata_auto_alloc_size = sizeof(struct dm_spi_slave_platdata),
      .child_post_bind = spi_child_post_bind,
};
U_BOOT_DRIVER(zynq_qspi) = {
      .name   = "zynq_qspi",
      .id   = UCLASS_SPI,
      .of_match = zynq_qspi_ids,
      .ops  = &zynq_qspi_ops,
      .ofdata_to_platdata = zynq_qspi_ofdata_to_platdata,
      .platdata_auto_alloc_size = sizeof(struct zynq_qspi_platdata),
      .priv_auto_alloc_size = sizeof(struct zynq_qspi_priv),
      .probe  = zynq_qspi_probe,
};
```

```c
static int sun4i_usb_phy_exit(struct phy *phy)
{
    clk_disable(&usb_phy->clocks);
    reset_assert(&usb_phy->resets);
}
static int sun4i_usb_phy_init(struct phy *phy)
{
    clk_enable(&usb_phy->clocks);
    reset_deassert(&usb_phy->resets);
}
static struct phy_ops sun4i_usb_phy_ops = {
    .init = sun4i_usb_phy_init,
    .exit = sun4i_usb_phy_exit,
};
static int sun4i_usb_phy_probe(struct udevice *dev)
{
    clk_get_by_name(dev, "usb0_phy", &phy->clocks);
    reset_get_by_name(dev, "usb0_reset", &phy->resets);
}
U_BOOT_DRIVER(sun4i_usb_phy) = {
    .id    = UCLASS_PHY,
    .ops = &sun4i_usb_phy_ops,
    .probe = sun4i_usb_phy_probe,
};
```

Generic Block Layer



cmd/mmc.c

cmd/ide.c

cmd/scsi.c

cmd/nvme.c

blk_common_cmd

cmd/blk_common.c

blk_read_devnum

blk_write_devnum

BLK Core

UCLASS_MMC

UCLASS_AHCI

UCLASS_SCSI

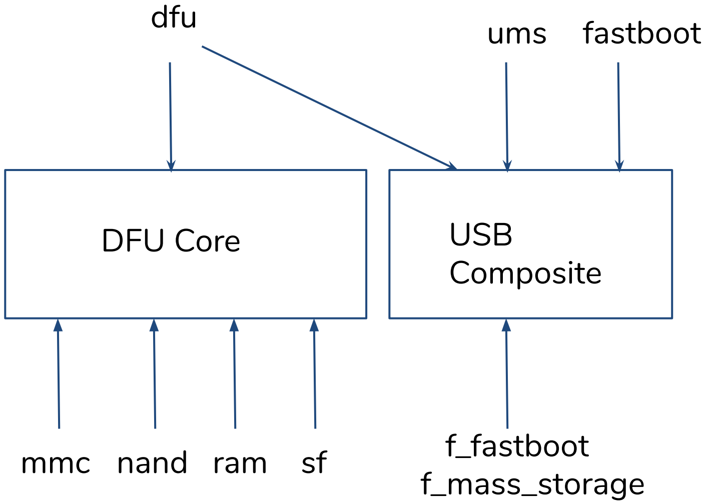
UCLASS_NVME

USB framework

- via UCLASS_USB
- uclass core: drivers/usb/host/usb-uclass.c
- platform specific uboot driver: drivers/usb/host/ehci-generic.c
- USB control, bulk, interrupt, create_int_queue etc via dm_usb_ops
- include/usb.h
- cmd/usb.c
- USB Gadgets can probe via Gadget UCLASS
- drivers/usb/gadget/ether.c, USB ETH Gadget
- MUSB can operate Host and Peripheral
- MUSB Host access via UCLASS_USB
- MUSB Peripheral access via
  - UCLASS_USB_DEV_GENERIC - host devices
  - UCLASS_USB_GADGET_GENERIC - gadget devices
- drivers/usb/musb-new/sunxi.c, SunXi MUSB driver

Firmware Upgrade

- ➔ Upgrade firmware images on running U-Boot
- ➔ DFU
- ➔ DFU via tftp
- ➔ UMS
- ➔ Fastboot

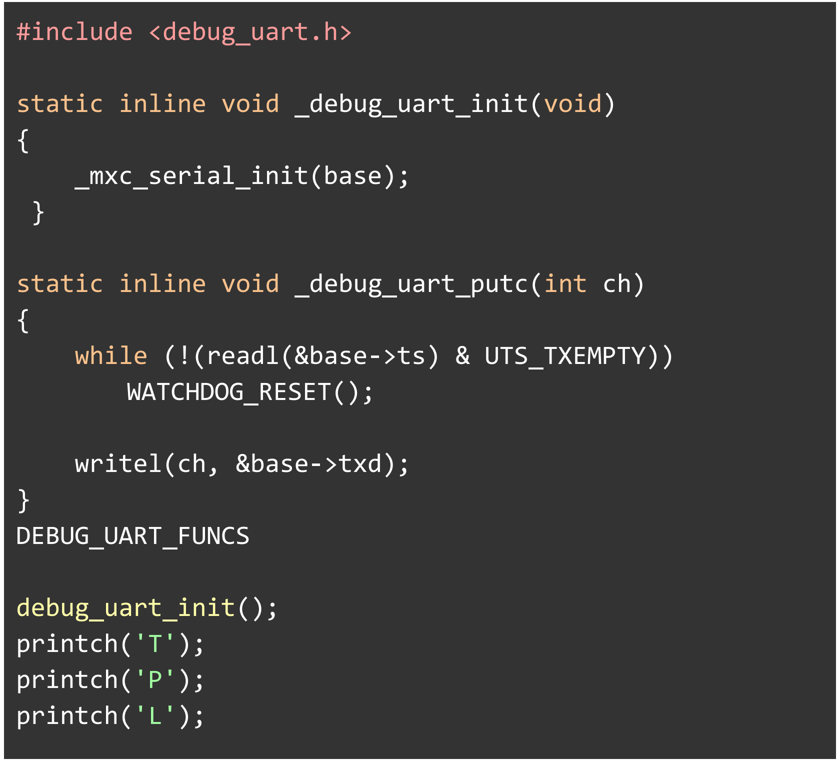


Miscellaneous

- Debug
- AVB
- x86, Slimboot
- Binman
- CI, tboot

Debug

- ➔ printf
- ➔ CONFIG_DEBUG
- ➔ GDB
- ➔ Early UART (CONFIG_DEBUG_UART)



```
#include <debug_uart.h>

static inline void _debug_uart_init(void)
{
    _mxc_serial_init(base);
 }

static inline void _debug_uart_putc(int ch)
{
    while (!(readl(&base->ts) & UTS_TXEMPTY))
        WATCHDOG_RESET();

    writel(ch, &base->txd);
}
DEBUG_UART_FUNCS

debug_uart_init();
printch('T');
printch('P');
printch('L');
```

AVB 2.0

➔ Verified Boot establishes a chain of trust from the bootloader to system images

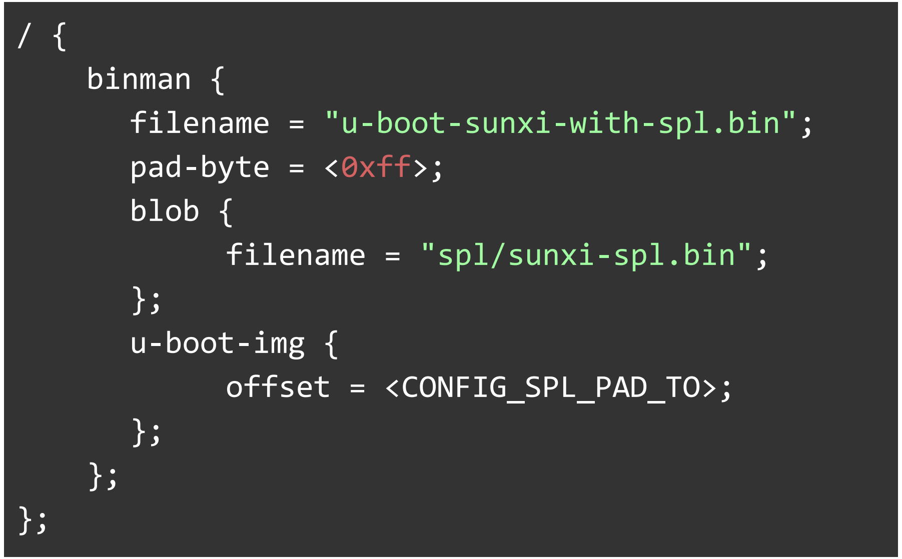
```
avb init <dev> - initialize avb 2.0 for <dev>
avb verify - run verification process using hash data from vbmeta structure
avb read_rb <num> - read rollback index at location <num>
avb write_rb <num> <rb> - write rollback index <rb> to <num>
avb is_unlocked - returns unlock status of the device
avb get_uuid <partname> - read and print uuid of partition <partname>
avb read_part <partname> <offset> <num> <addr> - read <num> bytes from
partition <partname> to buffer <addr>
avb write_part <partname> <offset> <num> <addr> - write <num> bytes to
<partname> by <offset> using data from <addr>
```

x86, Slimboot

- ➔ Support Broadwell, Apollolake bare-metals
- ➔ Intel FSP, FSP2?
- ➔ x86 boot can support coreboot as payload for U-Boot
- ➔ SBL as payload for U-boot
- ➔ SBL supports QEMU, Apollolake, Whiskeylake, Coffeelake-R platforms.

Binman

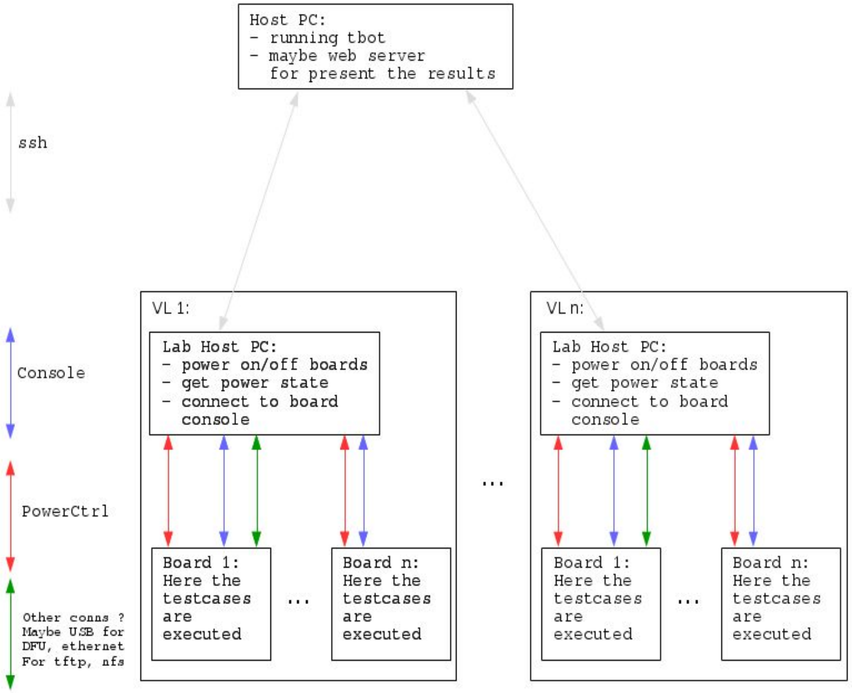
➔ Binman - Packaging multiple image components



```
/ {
    binman {
       filename = "u-boot-sunxi-with-spl.bin";
       pad-byte = <0xff>;
       blob {
             filename = "spl/sunxi-spl.bin";
       };
       u-boot-img {
             offset = <CONFIG_SPL_PAD_TO>;
       };
    };
};
```

tbot

- ➔ Execute test cases on boards
- ➔ Heiko Schocher, page https://www.tbot.tools/overview.html



Summary

- ➔ Lessons learned
- ➔ Future plans

Lessons learned from 18 years of U-Boot

- ➔ Release cycles - short and long does matters
- ➔ Strong distributed, hierarchical development model
- ➔ Supporting of new tools, make proper code validation
- ➔ More hands of developers, employers and maintainers
- ➔ Started with PPC, ARM - now support most possible architecture
- ➔ x86 development
- ➔ Drivers no longer, direct functions calls - DM

Future plans

- Kconfig migration
- Driver model migrations
  - BLK, DM_MMC, DM_SCSI, DM_USB
  - DM_SPI, DM_SPI_FLASH, DM_VIDEO, DM_PCI
- MTD driver model
- X86 development

Conclusion

- ➔ Use DT and DM for new ports
- ➔ Hands on with DM conversion
- ➔ ML: u-boot@lists.denx.de
- ➔ IRC: #u-boot
- ➔ LIn: https://www.linkedin.com/in/jaganteki/
- ➔ HABv4: https://wiki.amarulasolutions.com/uboot/secure_boot/imx6_habv4.html
- ➔ OP-TEE case study: https://wiki.amarulasolutions.com/optee/index.html
- ➔ Wiki: https://wiki.amarulasolutions.com

Questions??

Thank you

Jagan Teki <jagan@amarulasolutions.com>