PUNCH BOOT

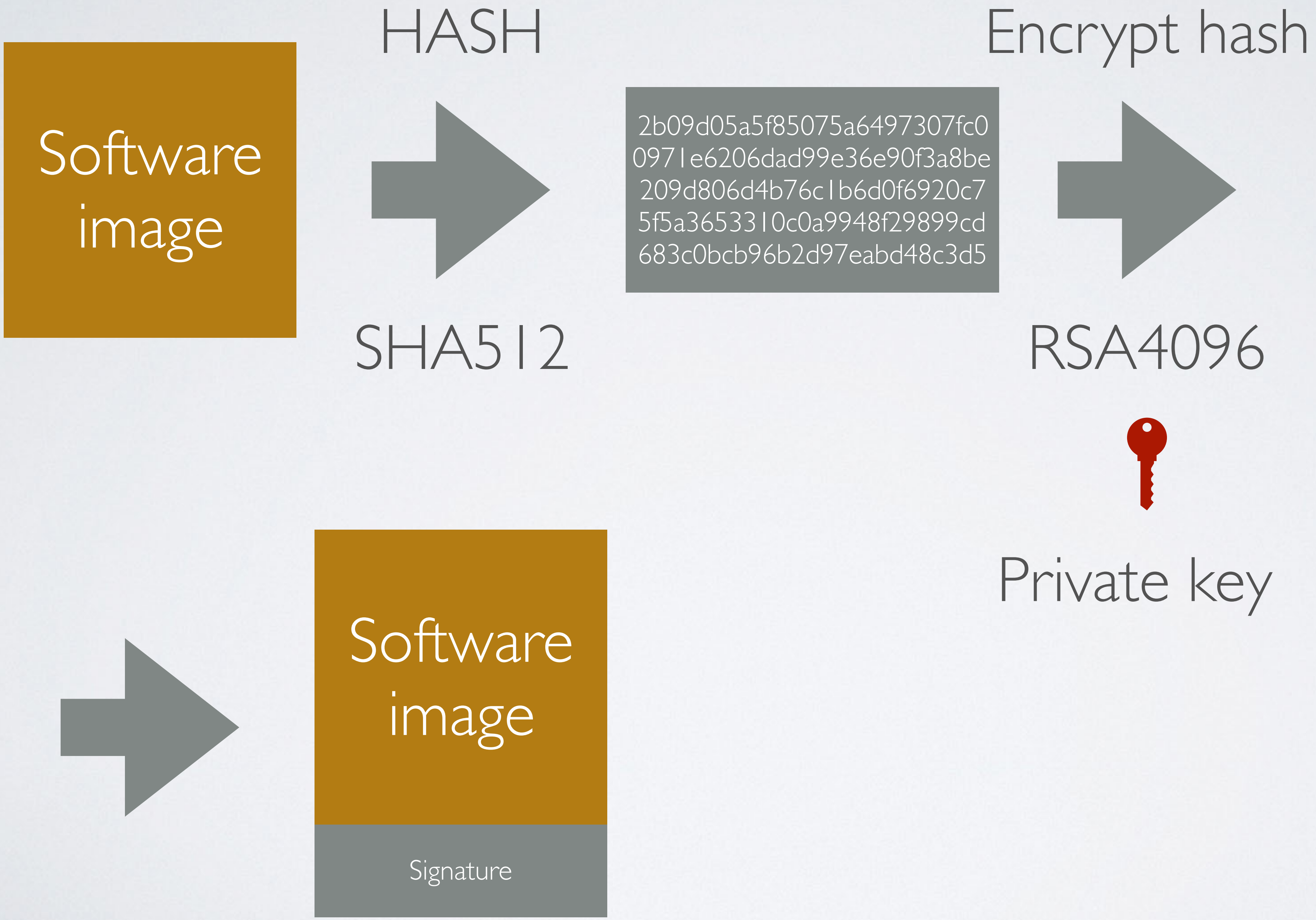
INTRODUCTION

- Boot loader for embedded systems
  - No run time configuration
  - < 2 kLOC
- Focus on security and boot time
- Production software download
  - USB HS transfer speeds of 20 MBytes/s
- Software update primitives
  - A / B system switching to support atomic updates

SECURE BOOT - BASICS

- Why secure boot?
  - Prevent malicious software from running
  - Supply chain integrity

CRYPTOGRAPHIC SIGNATURE



Software image

HASH

SHA512

2b09d05a5f85075a6497307fc0
0971e6206dad99e36e90f3a8be
209d806d4b76c1b6d0f6920c7
5f5a365331 0c0a9948f29899cd
683c0bcb96b2d97eabd48c3d5

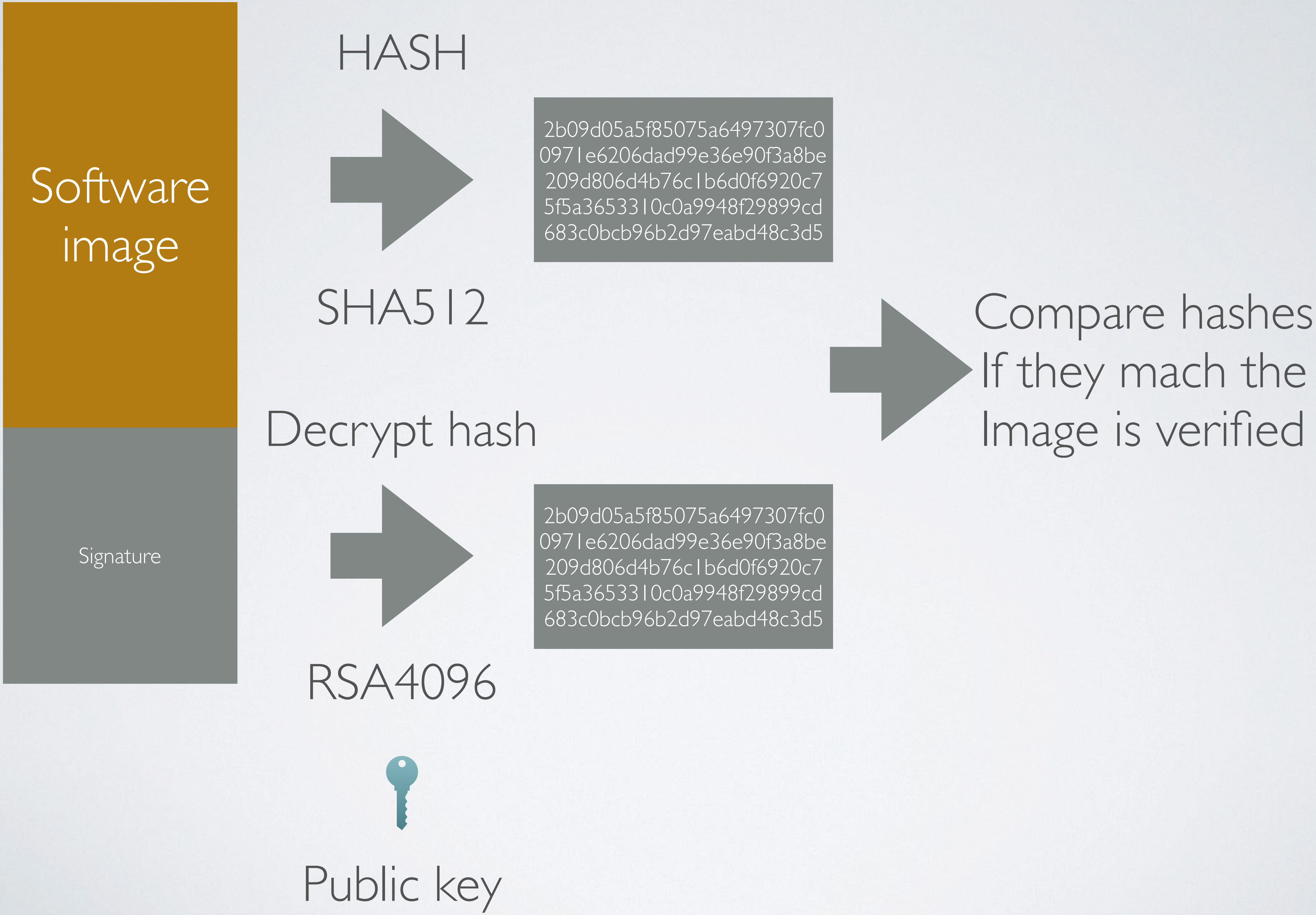
Encrypt hash

RSA4096

Private key

Software image

Signature



Software image

Signature

HASH

SHA512

2b09d05a5f85075a6497307fc0
0971e6206dad99e36e90f3a8be
209d806d4b76c1b6d0f6920c7
5f5a3653310c0a9948f29899cd
683c0bcb96b2d97eabd48c3d5

Decrypt hash

RSA4096

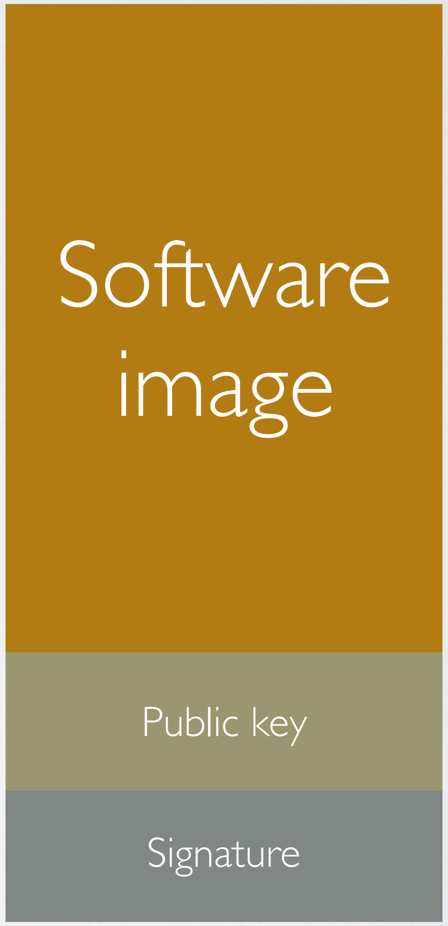
Public key

2b09d05a5f85075a6497307fc0
0971e6206dad99e36e90f3a8be
209d806d4b76c1b6d0f6920c7
5f5a3653310c0a9948f29899cd
683c0bcb96b2d97eabd48c3d5

Compare hashes
If they mach the
Image is verified

ROOT OF TRUST

- Public keys used for image verification must be fused into the CPU
- Size of the keys are unpractical to store in OTP fuses due to size
- Hash of public keys are stored in OTP fuses which can not be changed
- Every boot the mask rom compares stored public keys hash to the stored OTP hash



WHAT PROBLEMS CAN PUNCHBOOT SOLVE

- Secure boot
  - Load and authenticate next software image
  - Cryptographic accelerators for computing hash’es and RSA signatures
  - One hash and one signature for the complete image which might contain several images
- Production software download
  - Recovery mode allows high speed USB transfers which saves time in software download cell
  - Directly download boot loader image, kernel image and root filesystems
- Day-to-day development
  - Recovery mode can load images into RAM and execute them

DESIGN

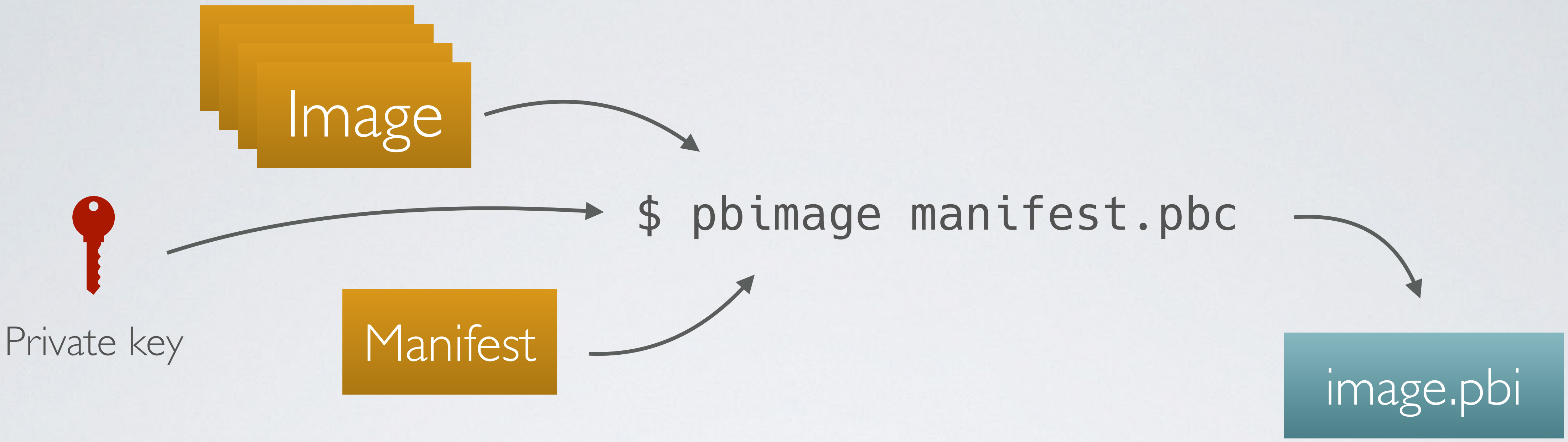
- C99
- Supports ARMv7a and ARMv8 architectures
- GUID Partition Table (GPT) support
- Platform support for IMX6UL, IMX8M, IMX8X
- Released under BSD - 3

PUNCHBOOT CLI

- Supports different communication backends
  - USB
  - Domain socket (for testing)
- Can easily be integrated into other tools

```
--- Punch BOOT 3c0e ---

Bootloader:
 punchboot boot -w -f <fn>                - Install bootloader
 punchboot boot -r                        - Reset device
 punchboot boot -b -s A or B              - BOOT System A or B
 punchboot boot -x -f <fn> [-s A or B]    - Load image to RAM and execute it
 punchboot boot -a -s A, B or none        - Activate system partition

Device:
 punchboot dev -l                         - Display device information
 punchboot dev -i [-f <fn>] [-y]          - Perform device setup
 punchboot dev -w [-y]                    - Lock device setup

Partition Management:
 punchboot part -l                        - List partitions
 punchboot part -w -n <n> -f <fn>         - Write 'fn' to partition 'n'
 punchboot part -i
```

PBIMAGE TOOL

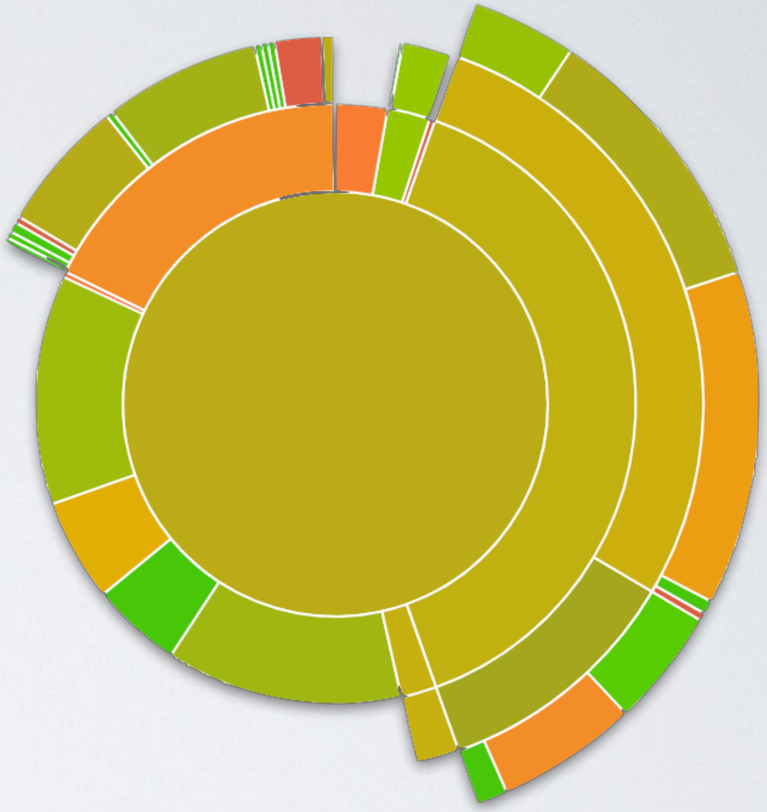


```
$ pbimage manifest.pbc
```

PB Image manifest

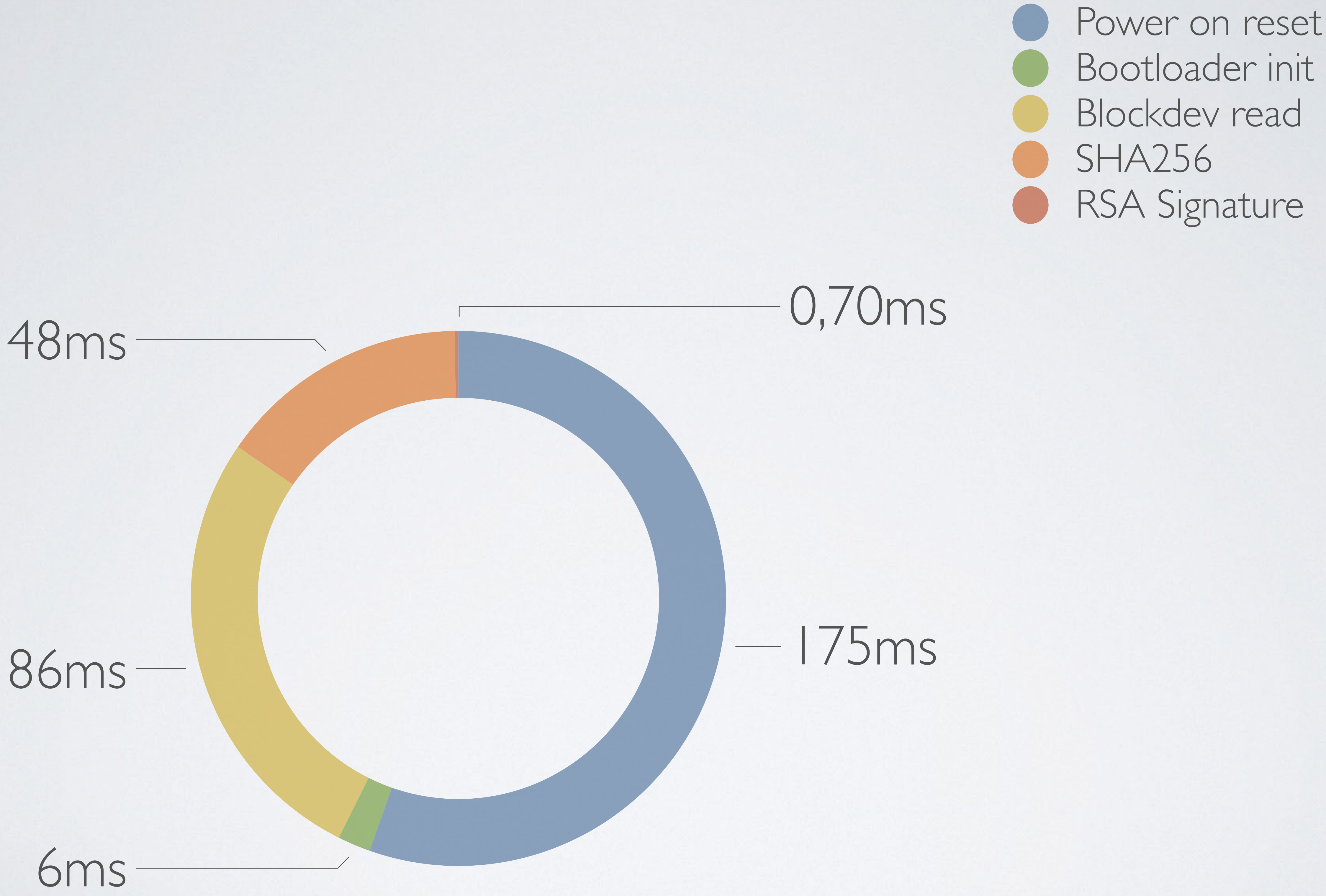
```
[pbimage]
key_index = 1
key_source = ../pki/prod_rsa_private.der
output = jiffy.pbi

[component]
type = ATF
load_addr = 0x80000000
file = /work/imx-atf/build/imx8qxp/release/bl31.bin

[component]
type = DT
load_addr = 0x82000000
file = /work/linux-imx/arch/arm64/boot/dts/freescale/jiffy.dtb

[component]
type = LINUX
load_addr = 0x82020000
file = /work/linux-imx/arch/arm64/boot/Image
```

MODULE AND INTEGRATION TESTS



- Test suite runs in QEMU
- 85 % coverage
- Integration tests also cover support tools
- Static code analysis performed with synopsys coverity

15 MByte boot image on IMX8X


Power on reset
Bootloader init
Blockdev read
SHA256
RSA Signature
0,70ms
48ms
175ms
86ms
6ms