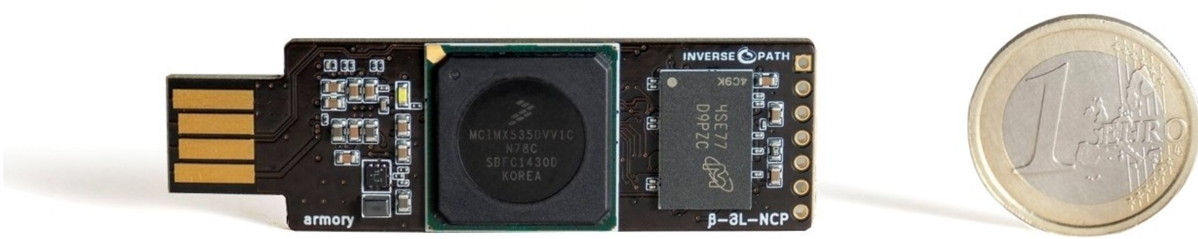
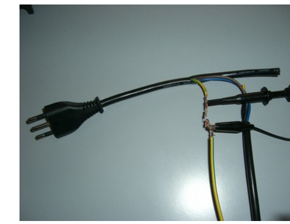
Forging the USB armory

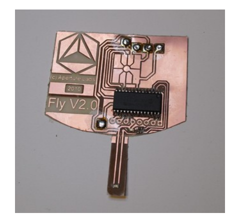
Andrea Barisani
<andrea@inversepath.com>



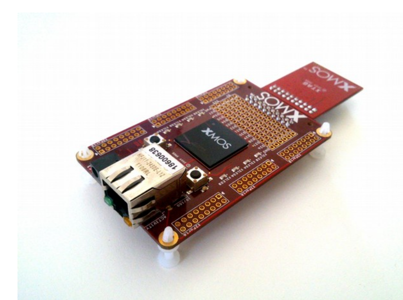
2007: Unusual Car Navigation Tricks
Injecting RDS-TMC Traffic Information Signals



2009: Sniff Keystrokes With Lasers/Voltmeters
Side Channel Attacks Using Optical Sampling Of Mechanical Energy And Power Line Leakage

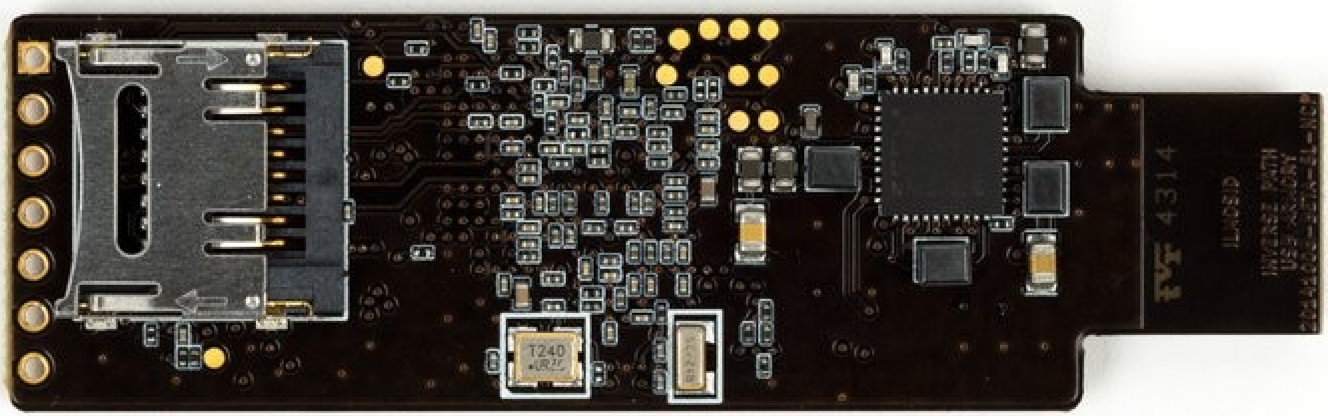


2011: Chip & PIN is definitely broken
Credit card skimming and PIN harvesting in an EMV world



2013: Fully arbitrary 802.3 packet injection
Maximizing the Ethernet attack surface

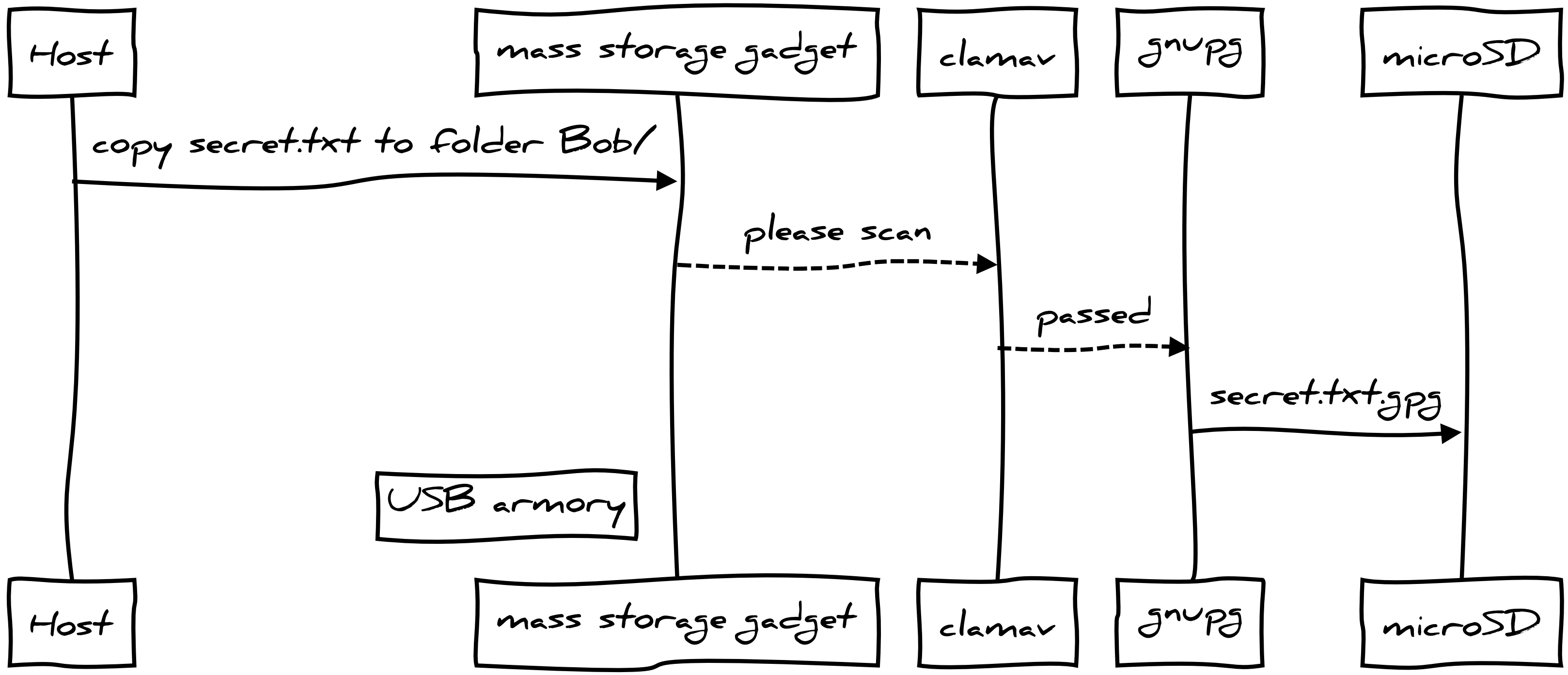
INVERSE PATH



Designed for personal security applications

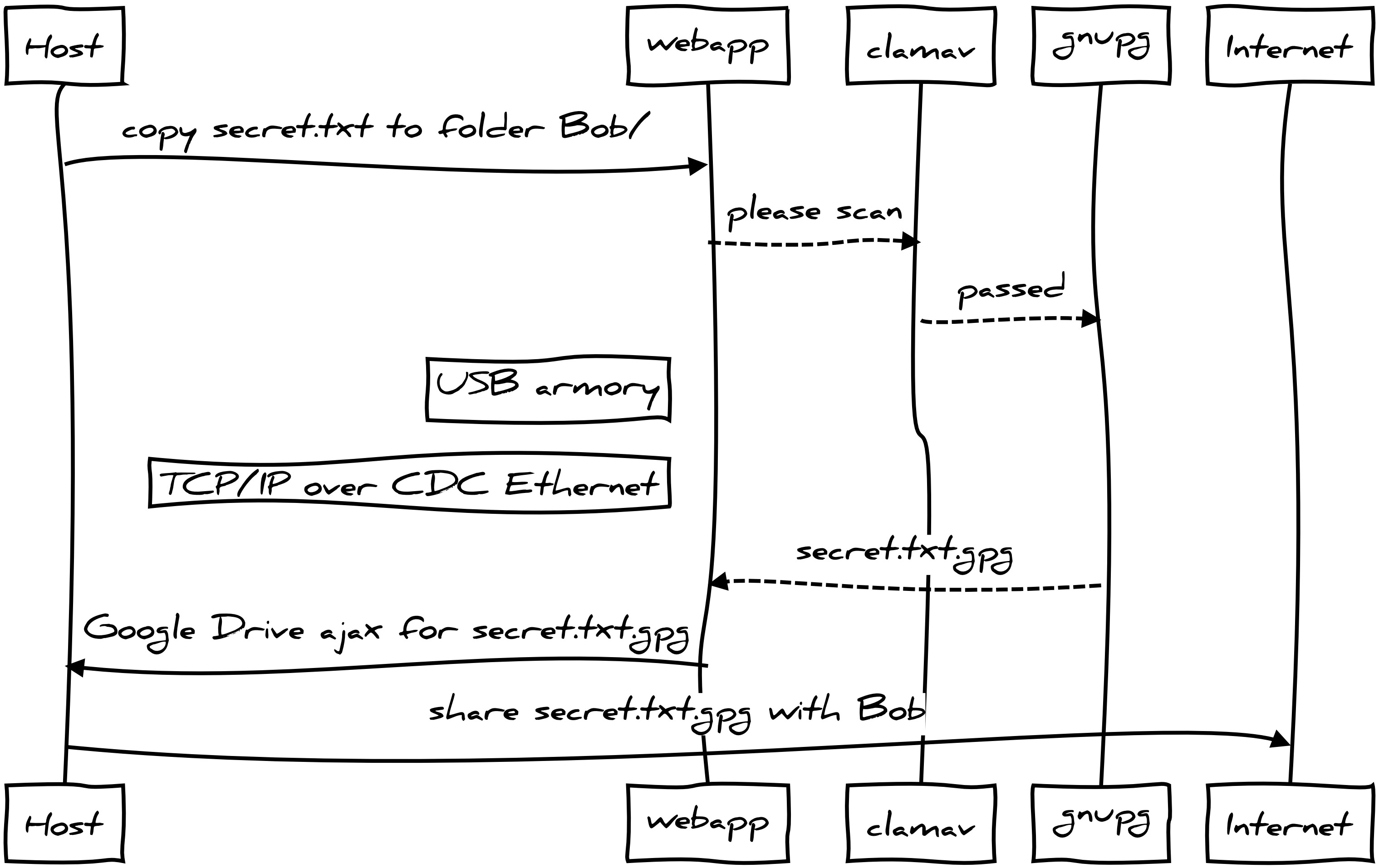
- mass storage device with advanced features such as automatic encryption, virus scanning, host authentication and data self-destruct
- OpenSSH client and agent for untrusted hosts (kiosk)
- router for end-to-end VPN tunneling, Tor
- password manager with integrated web server
- electronic wallet (e.g. pocket Bitcoin wallet)
- authentication token
- portable penetration testing platform
- low level USB security testing

enhanced mass storage

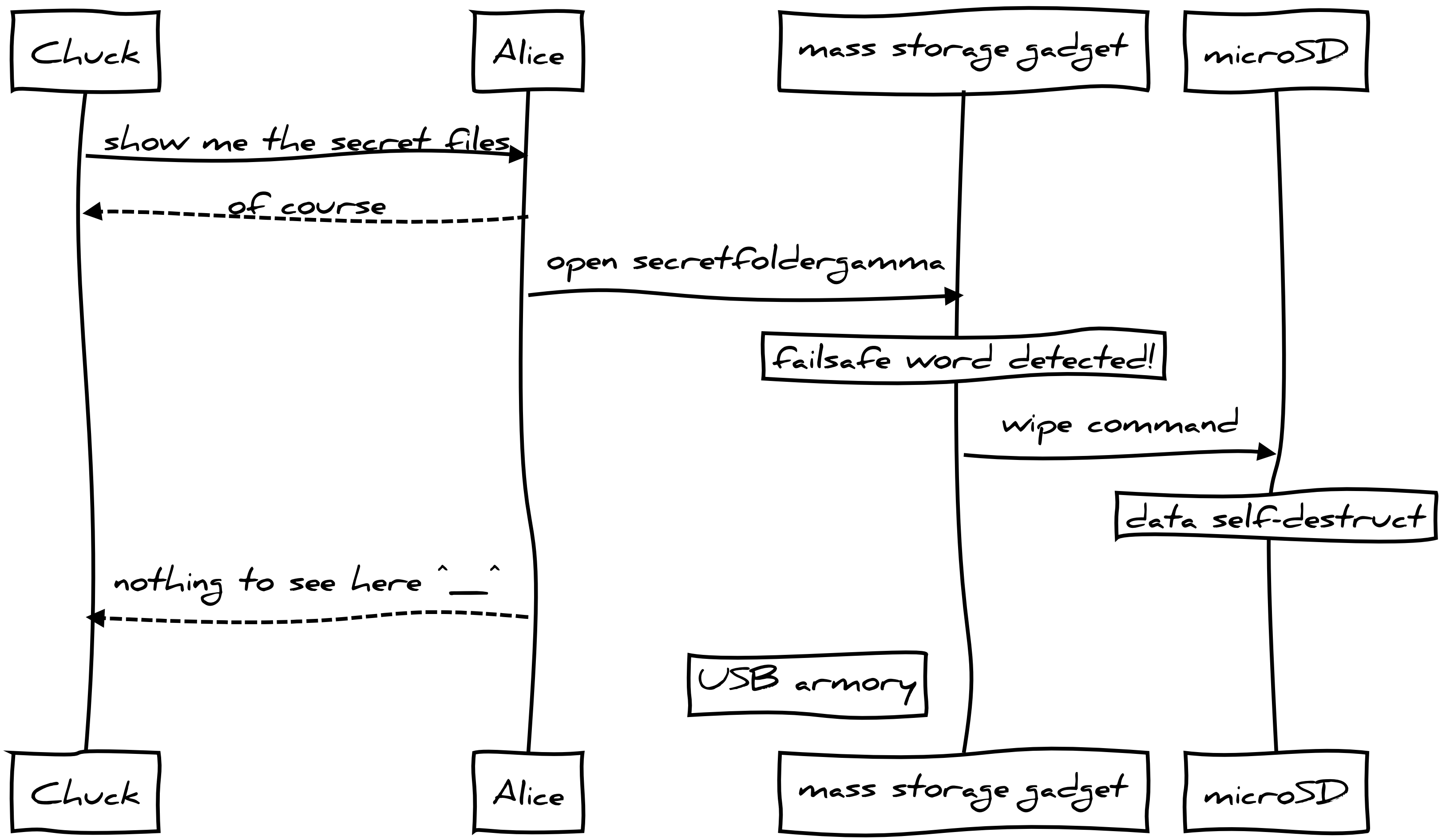


INVERSE PATH

enhanced mass storage

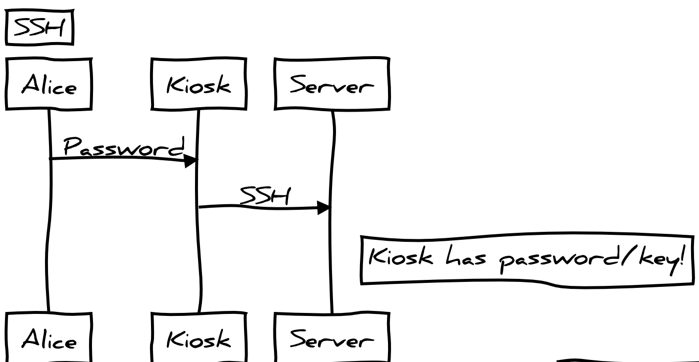


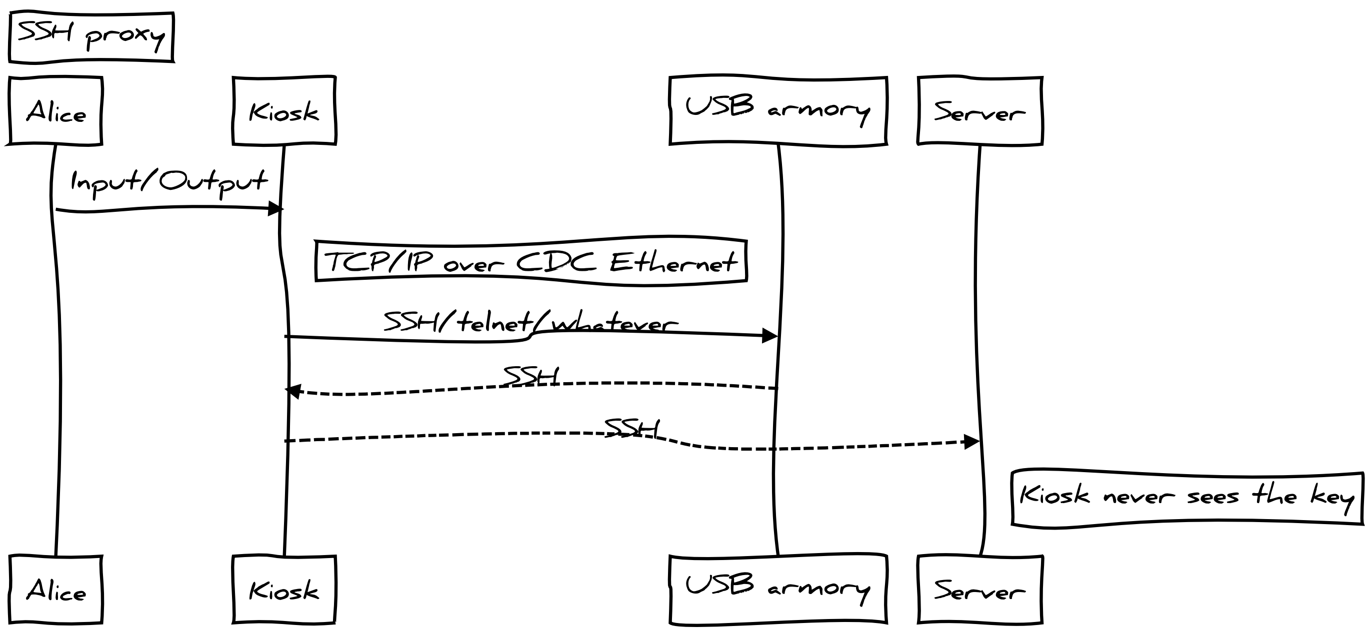
enhanced mass storage

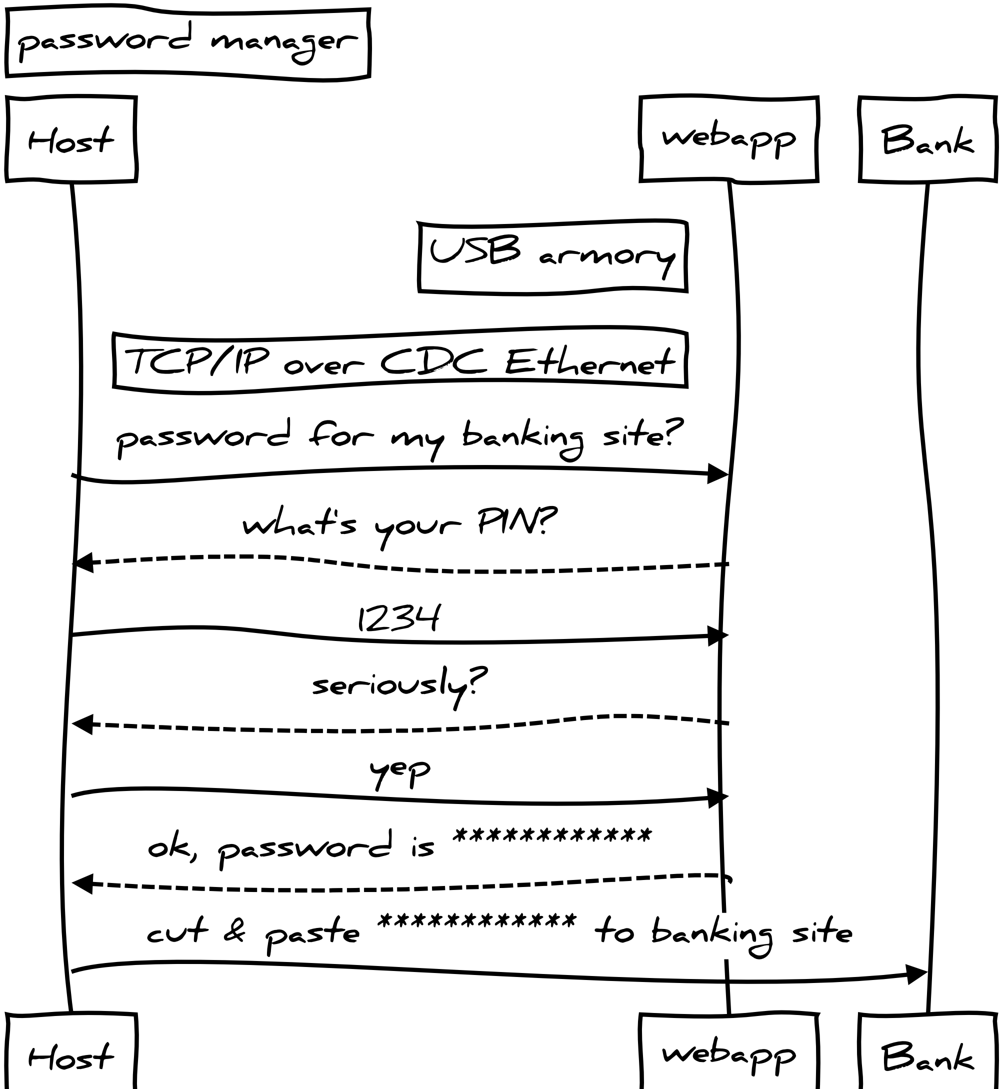


INVERSE PATH

SSH proxy



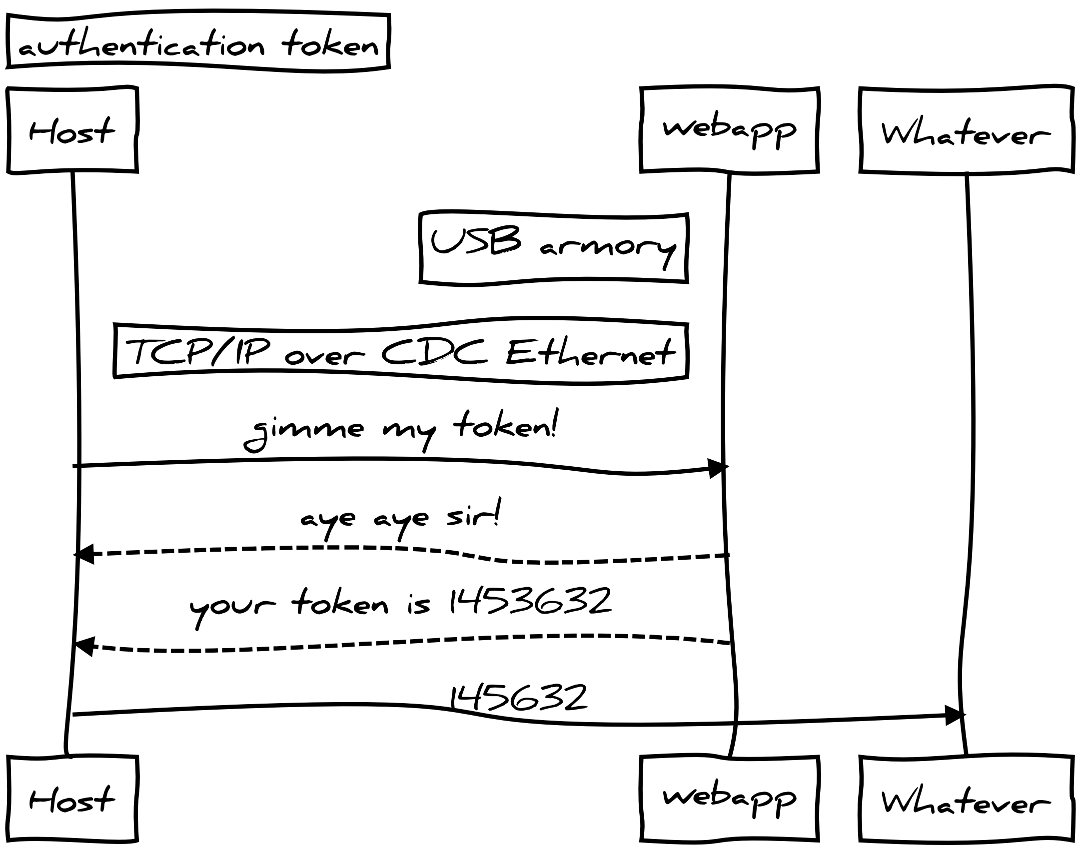




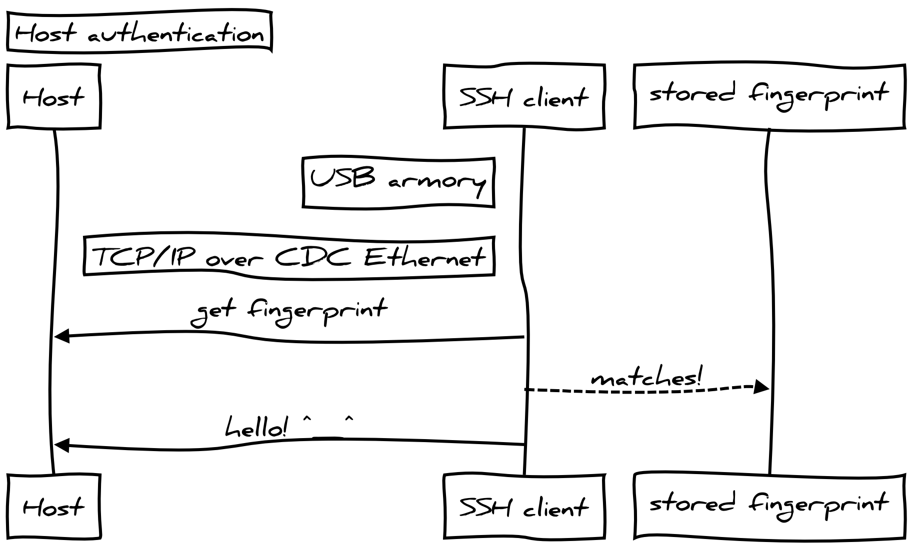
password manager

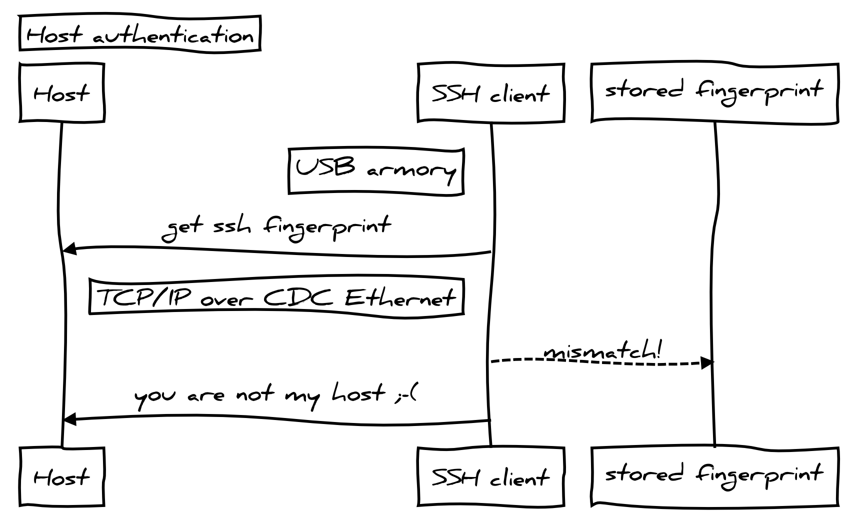
*trivial example, better options planned

authentication token



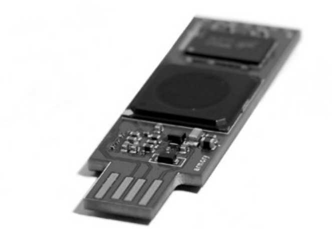
USB device authenticates host





Design goals

Compact USB powered device
Fast CPU and generous RAM
Secure boot
Standard connectivity over USB
Familiar developing/execution environment
Open design

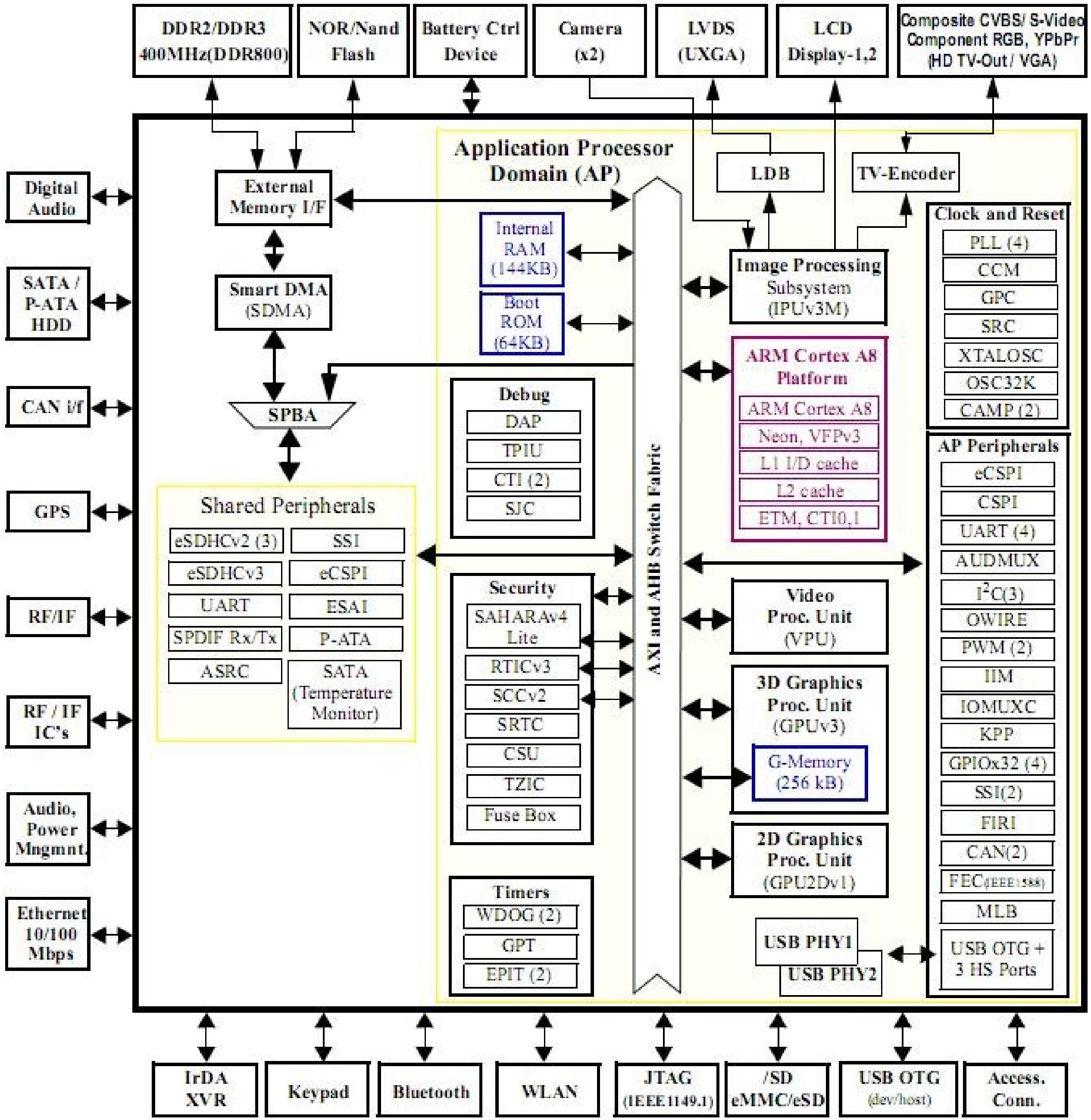


Selecting the System on Chip (SoC)

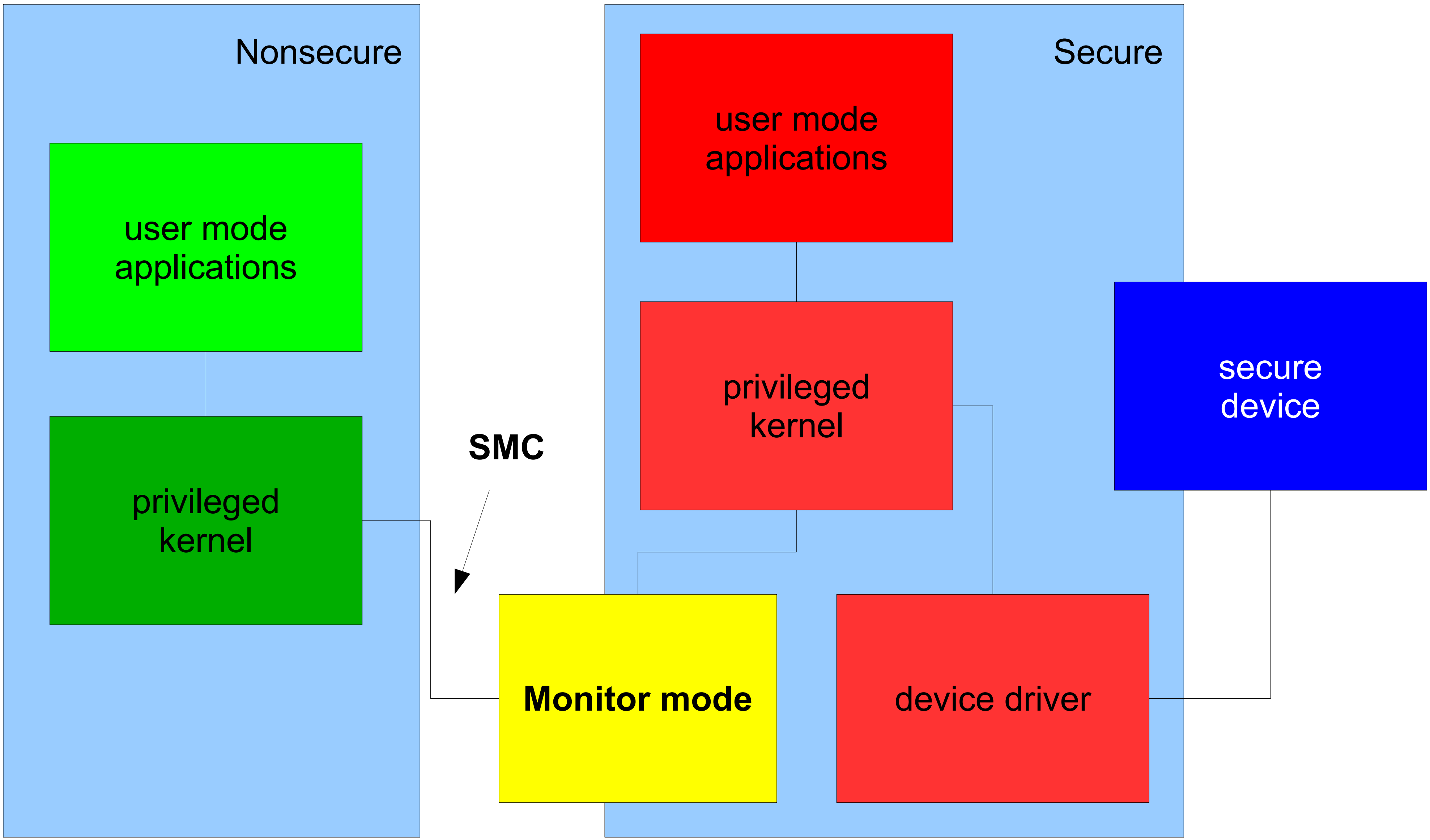
Freescale i.MX53

- ARM® Cortex™-A8 800-1200 Mhz
- almost all datasheets/manuals are public (no NDA required)
- Freescale datasheets are “ok” (far better than other vendors)
- ARM® TrustZone®, secure boot + storage + RAM
- detailed power consumption guide available
- excellent native support (Android, Debian, Ubuntu, FreeBSD)
- good stock and production support guarantee

INVERSE PATH

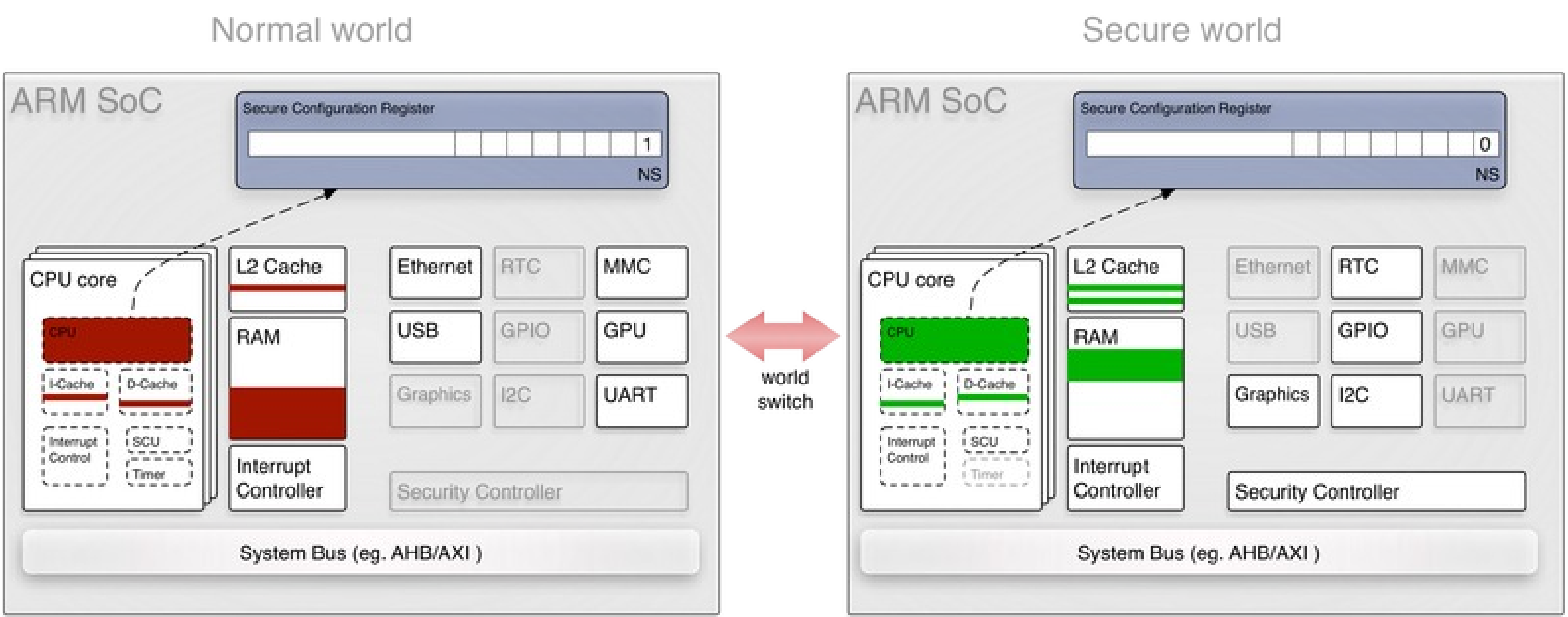


Domain Separation



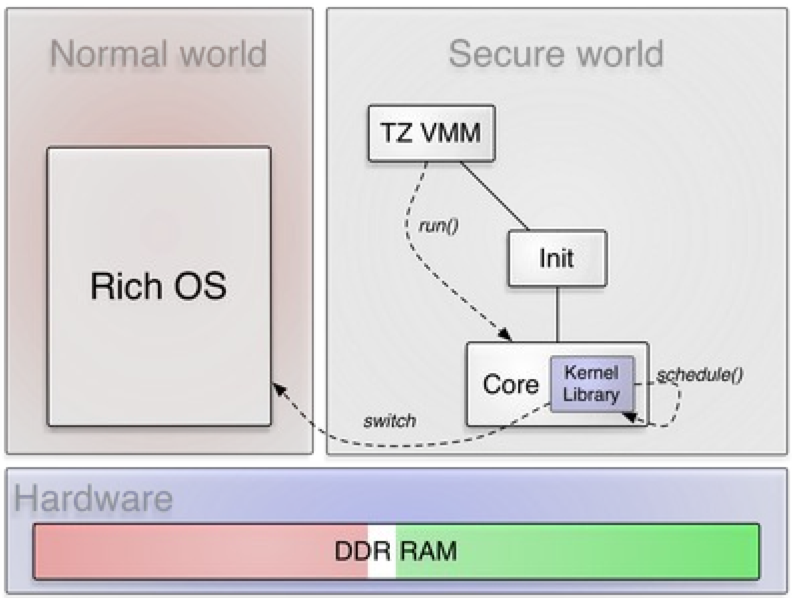
INVERSE PATH

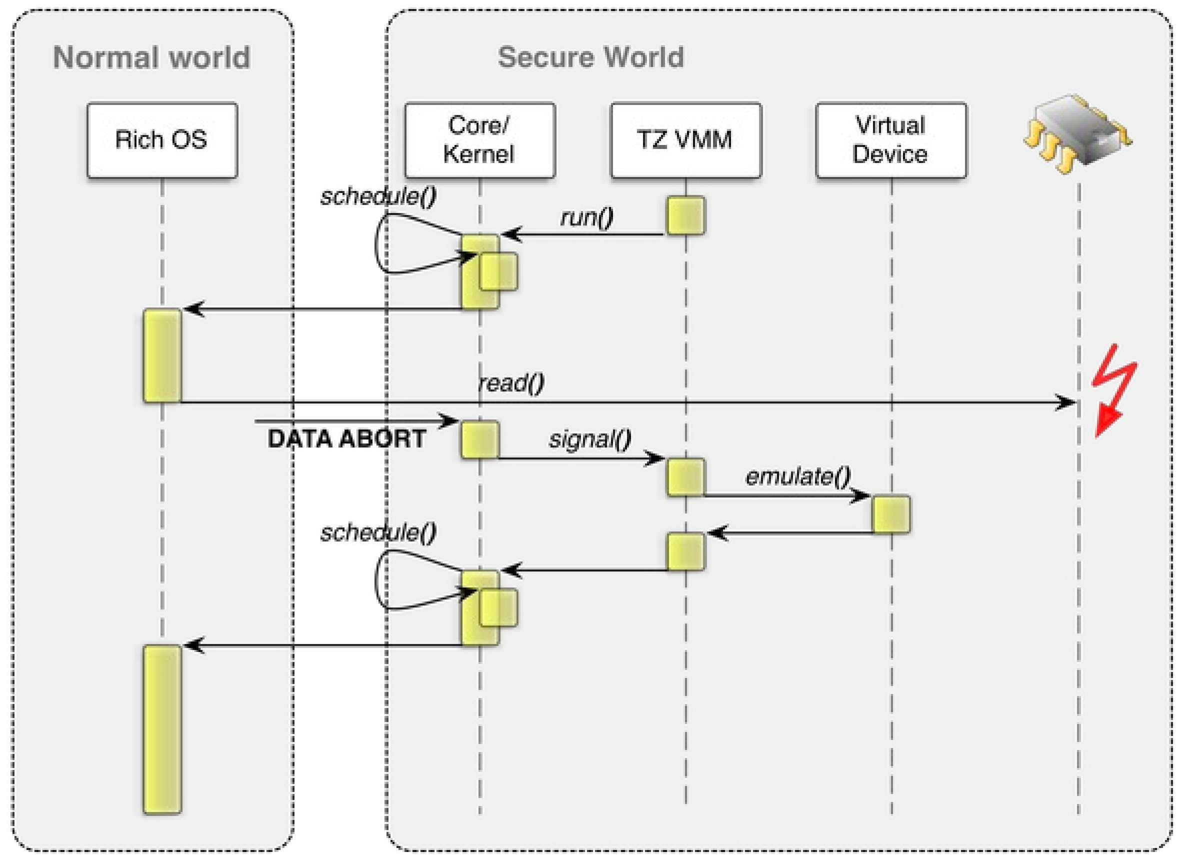
ARM® TrustZone®



http://genode.org/documentation/articles/trustzone

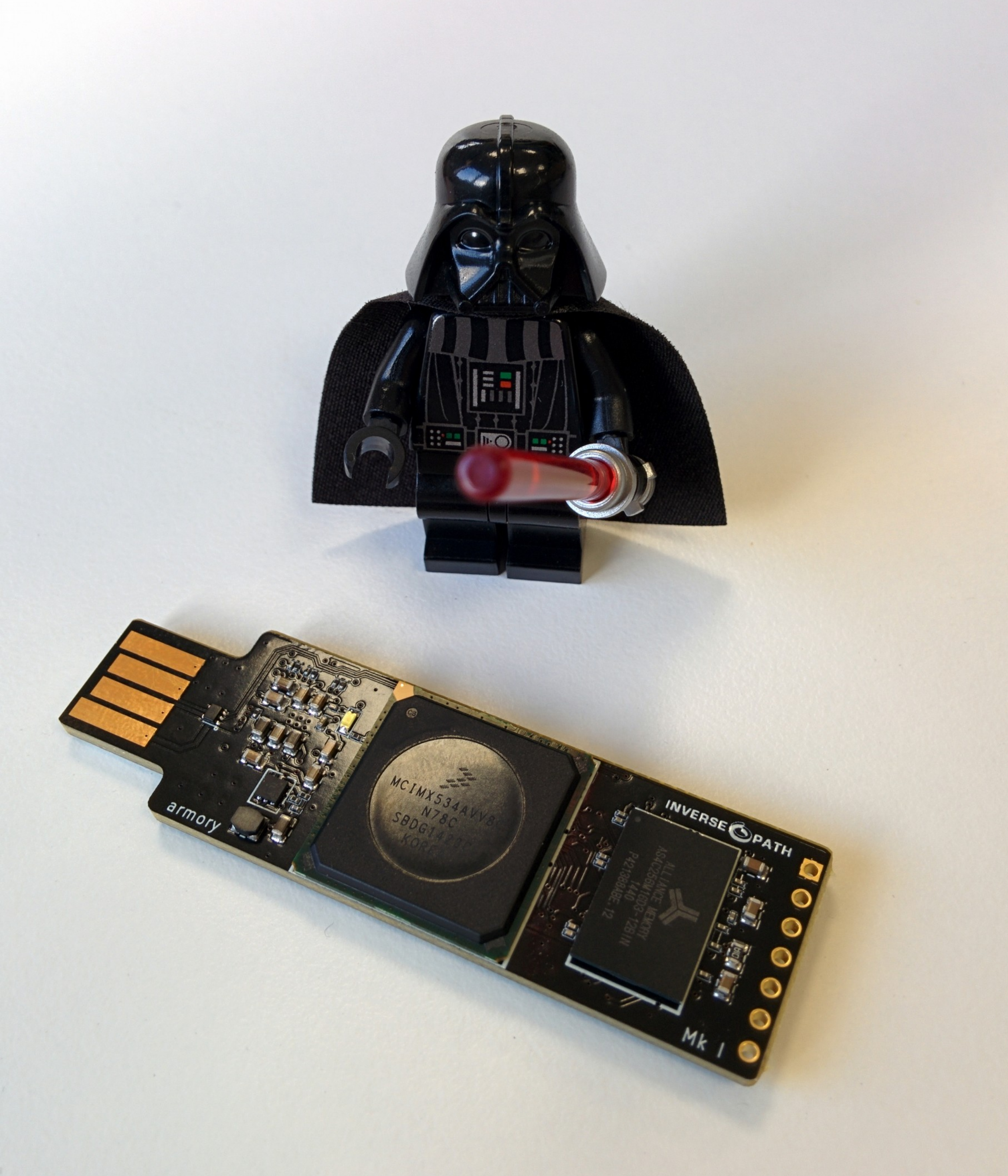
ARM® TrustZone®





http://genode.org/documentation/articles/trustzone

INVERSE PATH



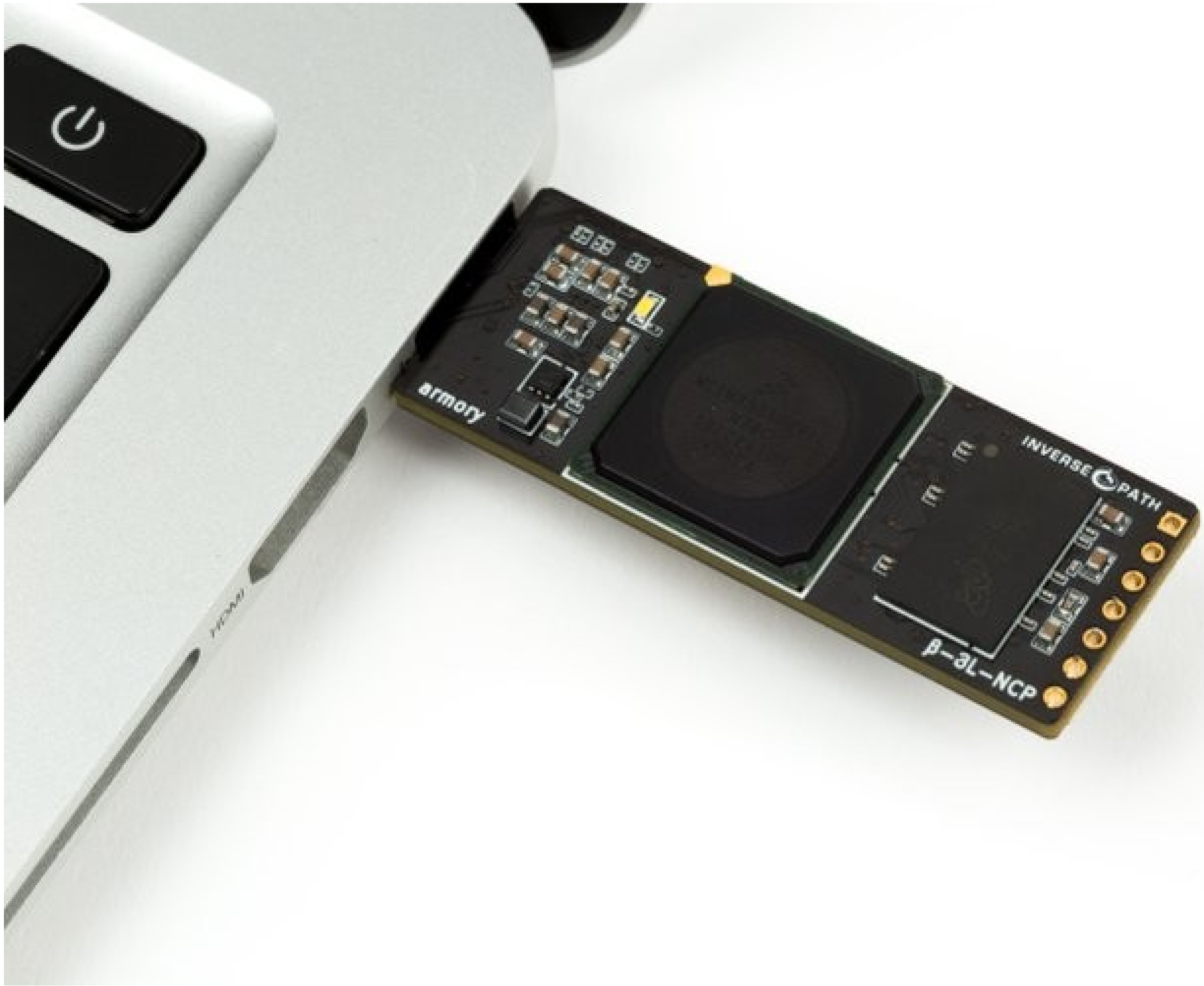
open source hardware

open source™

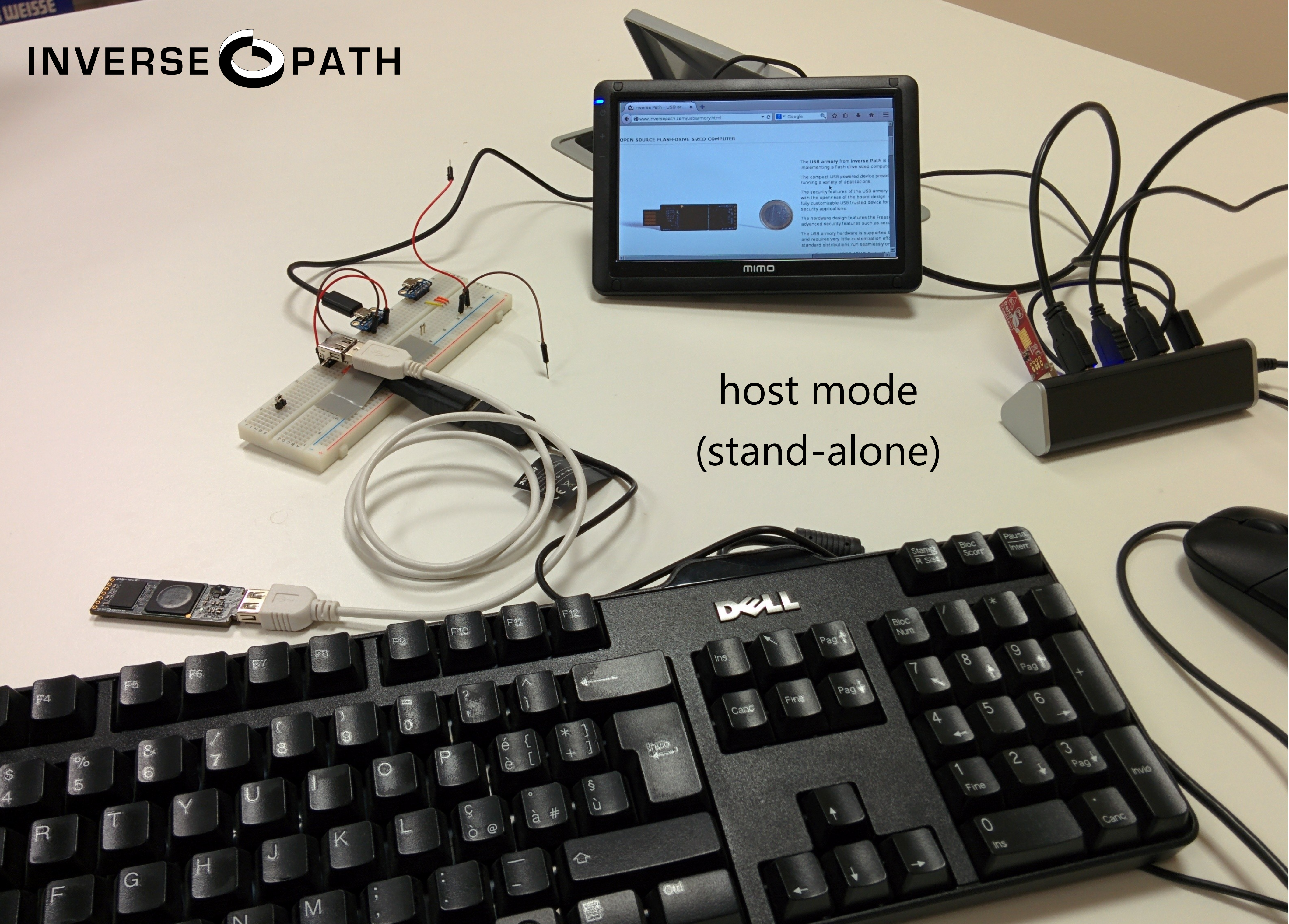
http://inversepath.com/usbarmory

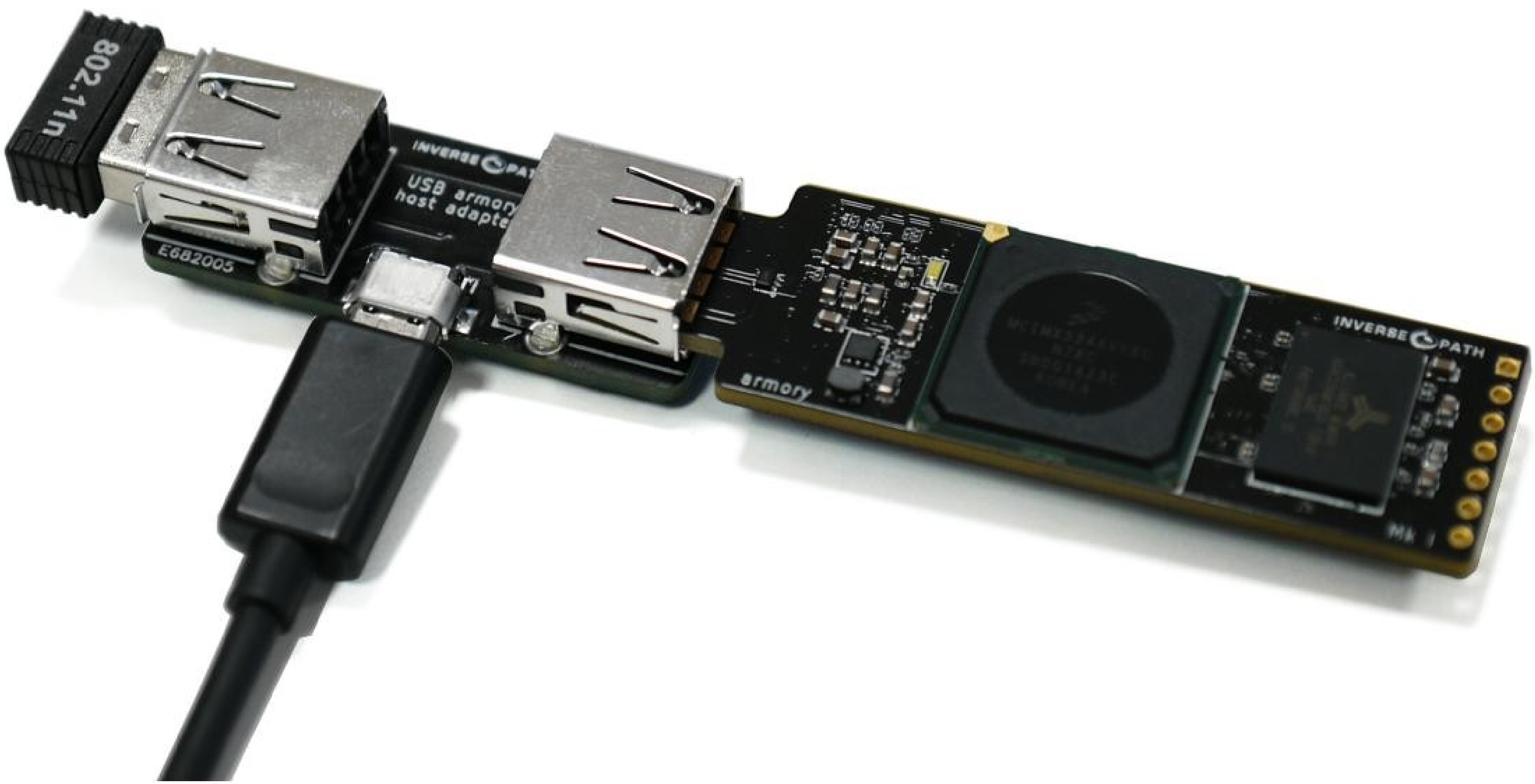
USB armory - Open source flash-drive-sized computer

- Freescale i.MX53 ARM® Cortex™-A8 800Mhz, 512MB DDR3 RAM
- USB host powered (<500 mA) device with compact form factor (65 x 19 x 6 mm)
- ARM® TrustZone®, secure boot + storage + RAM
- microSD card slot
- 5-pin breakout header with GPIOs and UART
- customizable LED, including secure mode detection
- excellent native support (Debian, Ubuntu, Arch Linux ARM)
- USB device emulation (CDC Ethernet, mass storage, HID, etc.)
- Open Hardware & Software



device mode

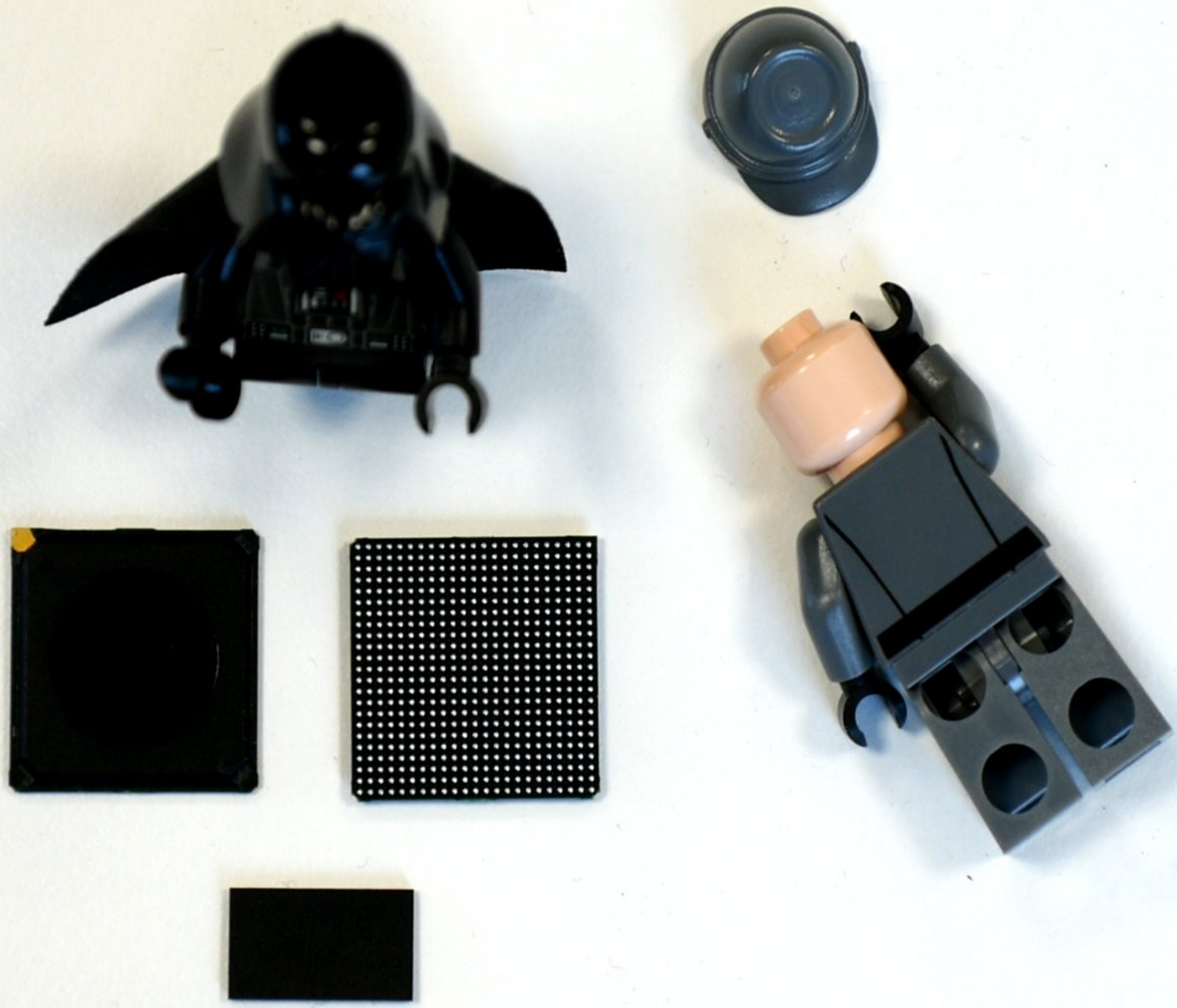

INVERSE PATH
host mode
(stand-alone)
OPEN SOURCE FLASH-DRIVE SIZED COMPUTER
mimo
DELL

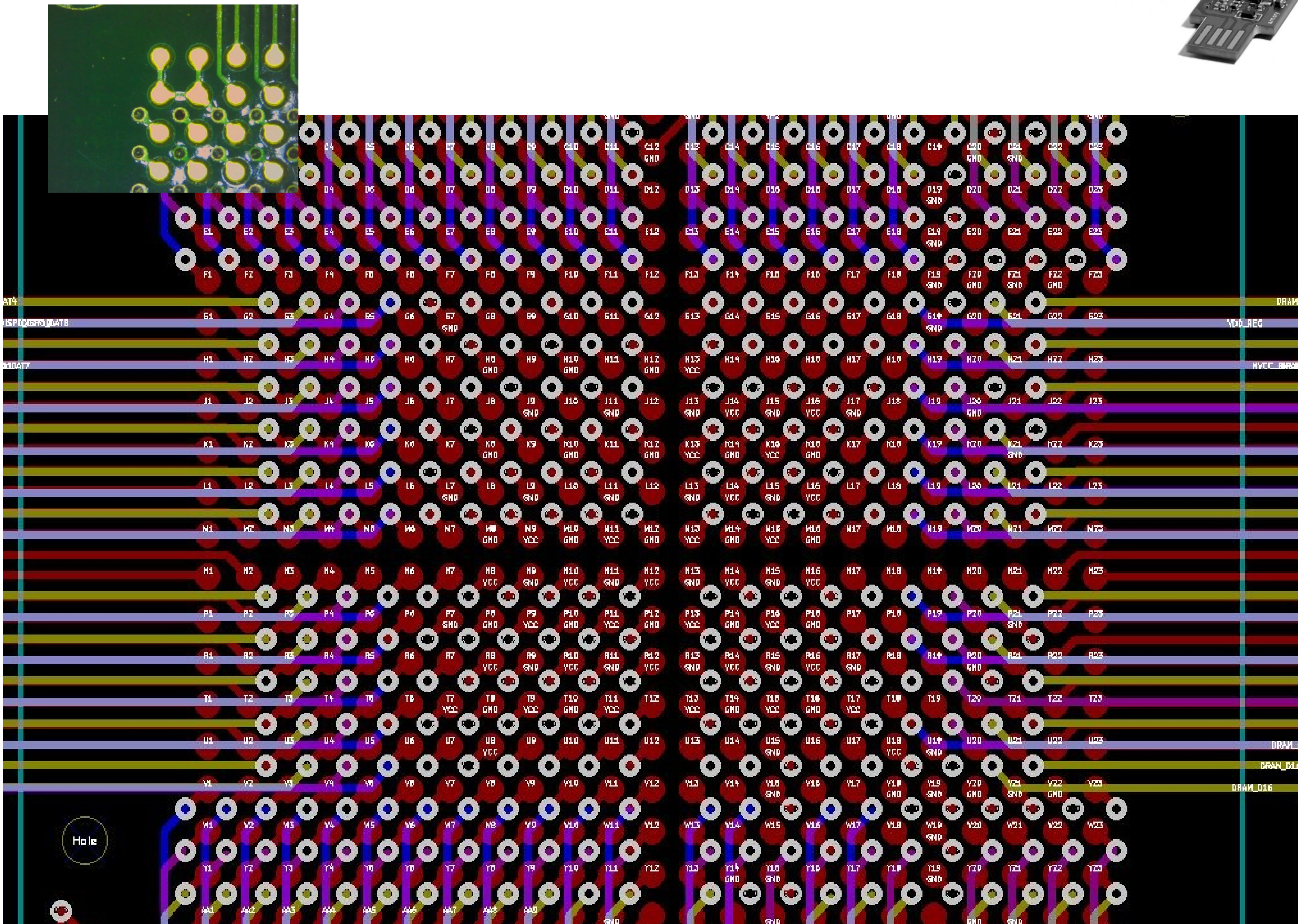


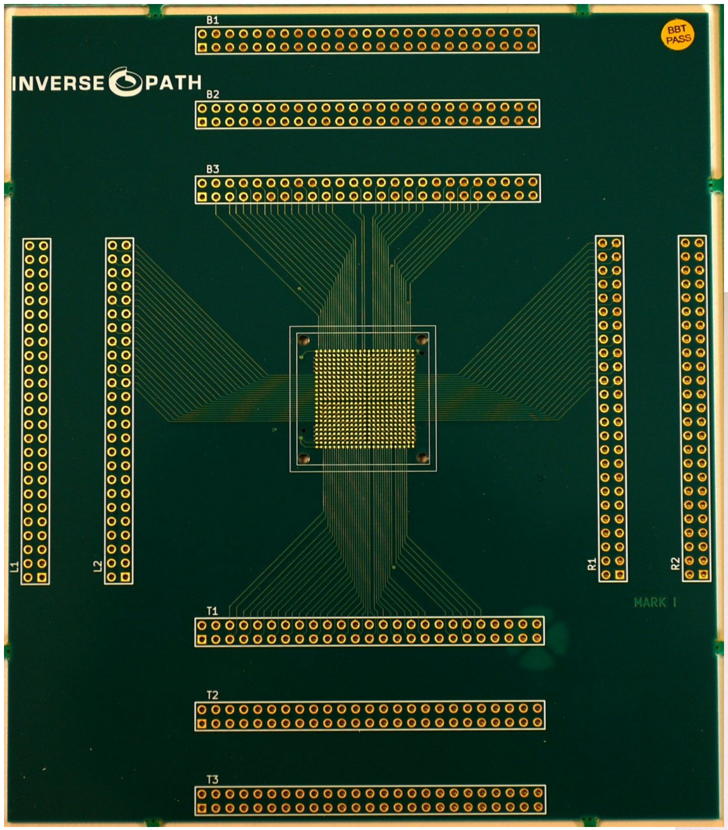
INVERSE PATH

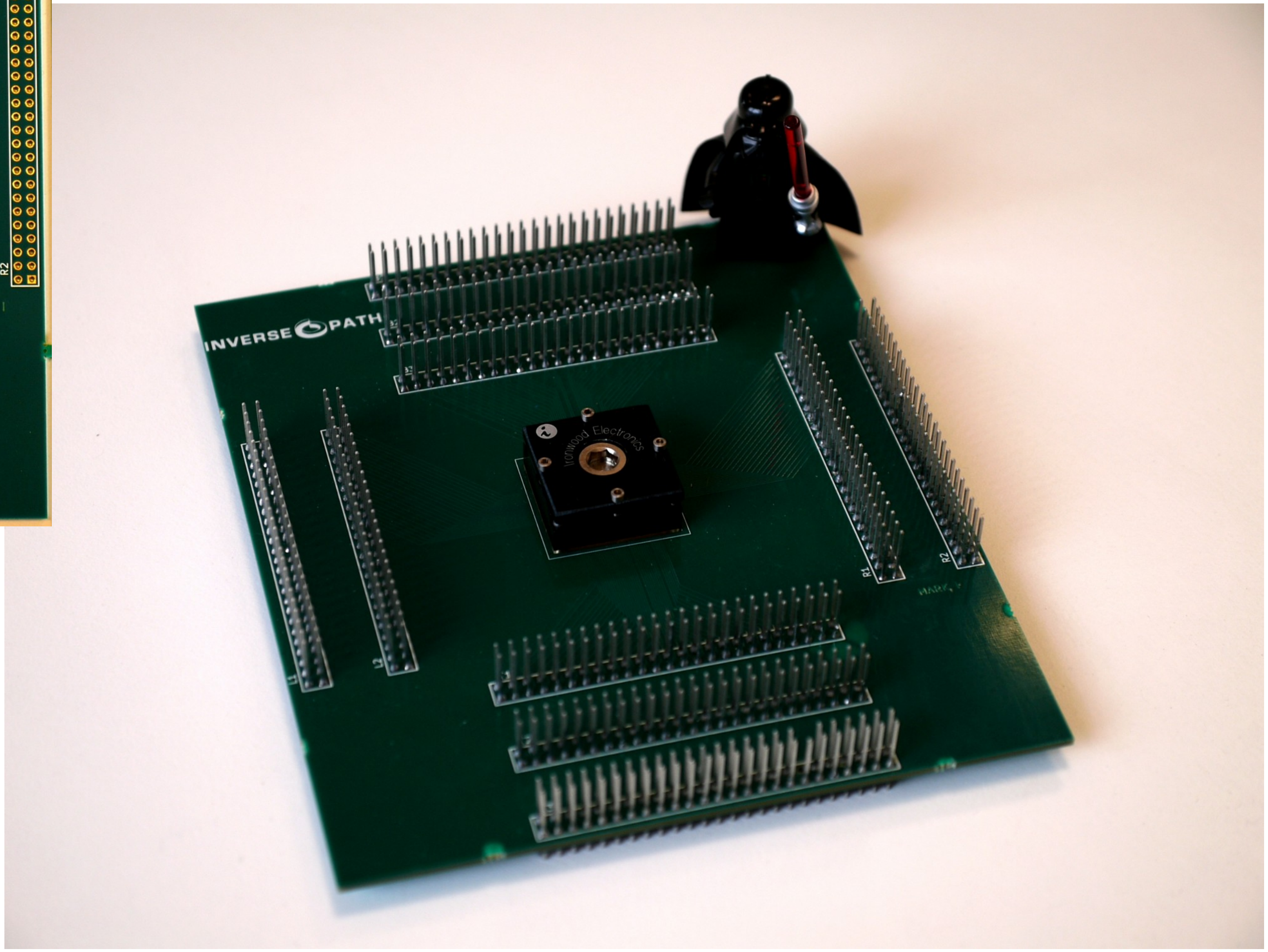
inversepath.com/usbarmory

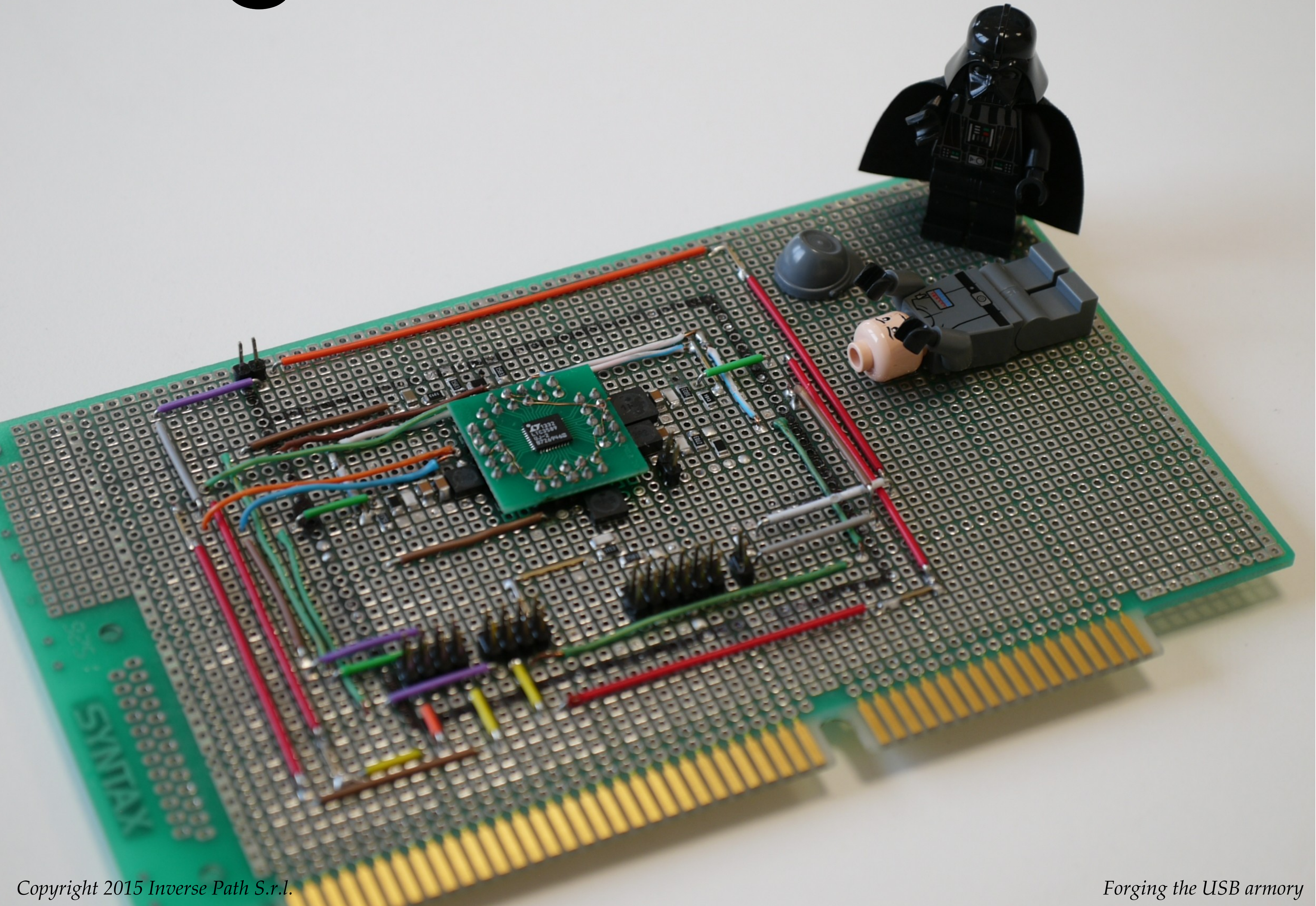
INVERSE PATH

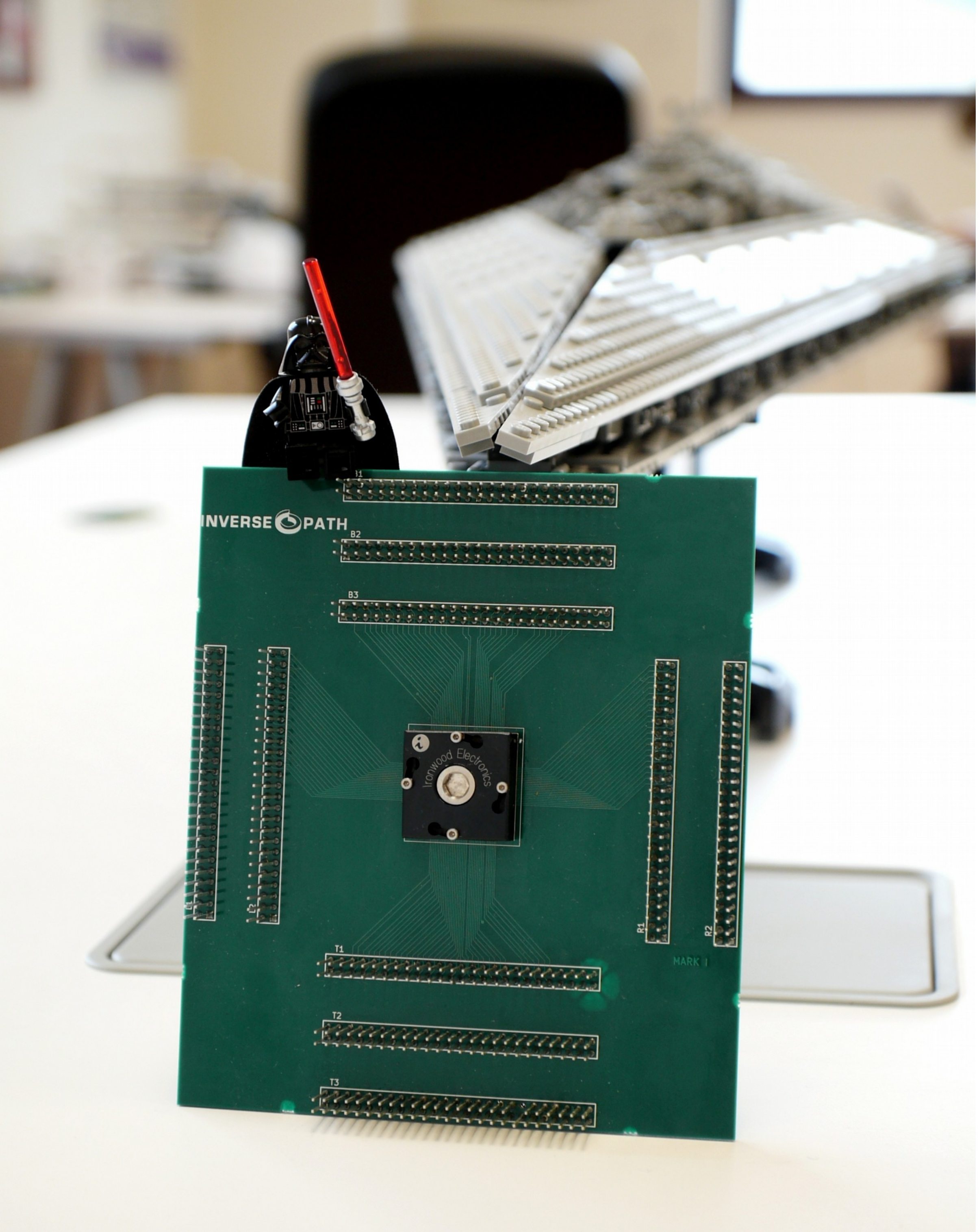



Hole
VDD_REG
DRAM_D16

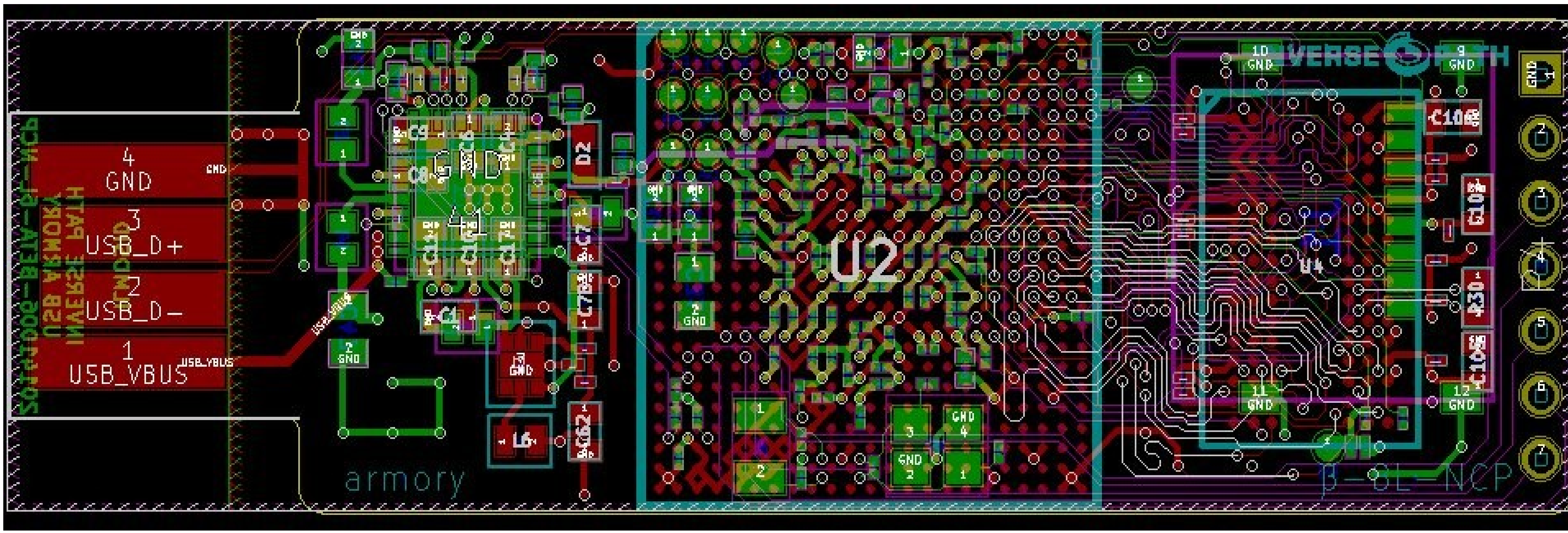

INVERSE PATH
BBT PASS
B1
B2
B3
L1
L2
R1
R2
T1
T2
T3
MARK I

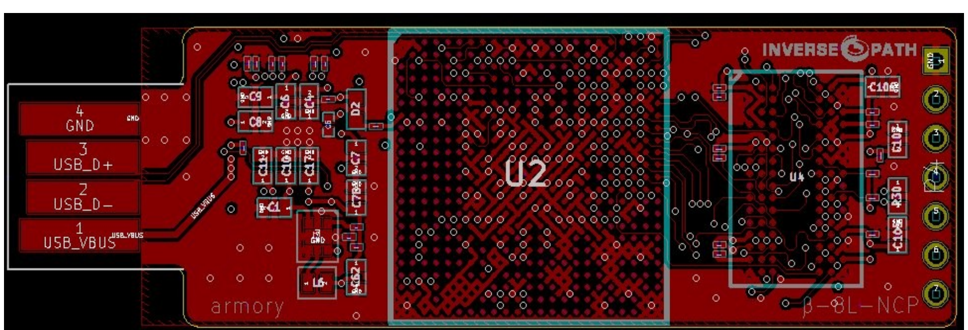

INVERSE PATH

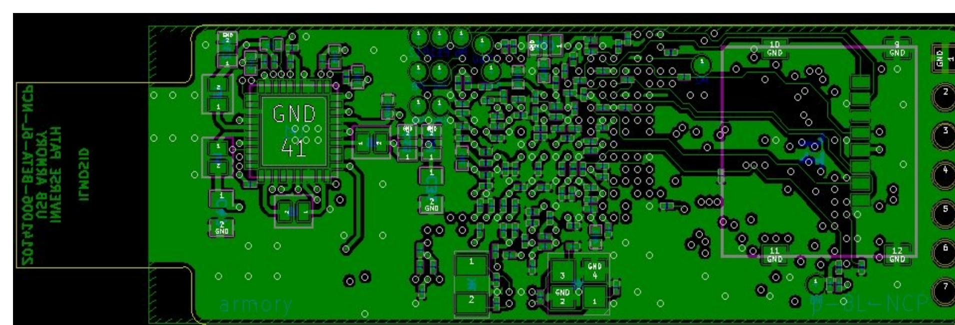

SYNTAX


INVERSE PATH
B2
B3
T1
T2
T3
R1
R2
MARK 1
Ironwood Electronics

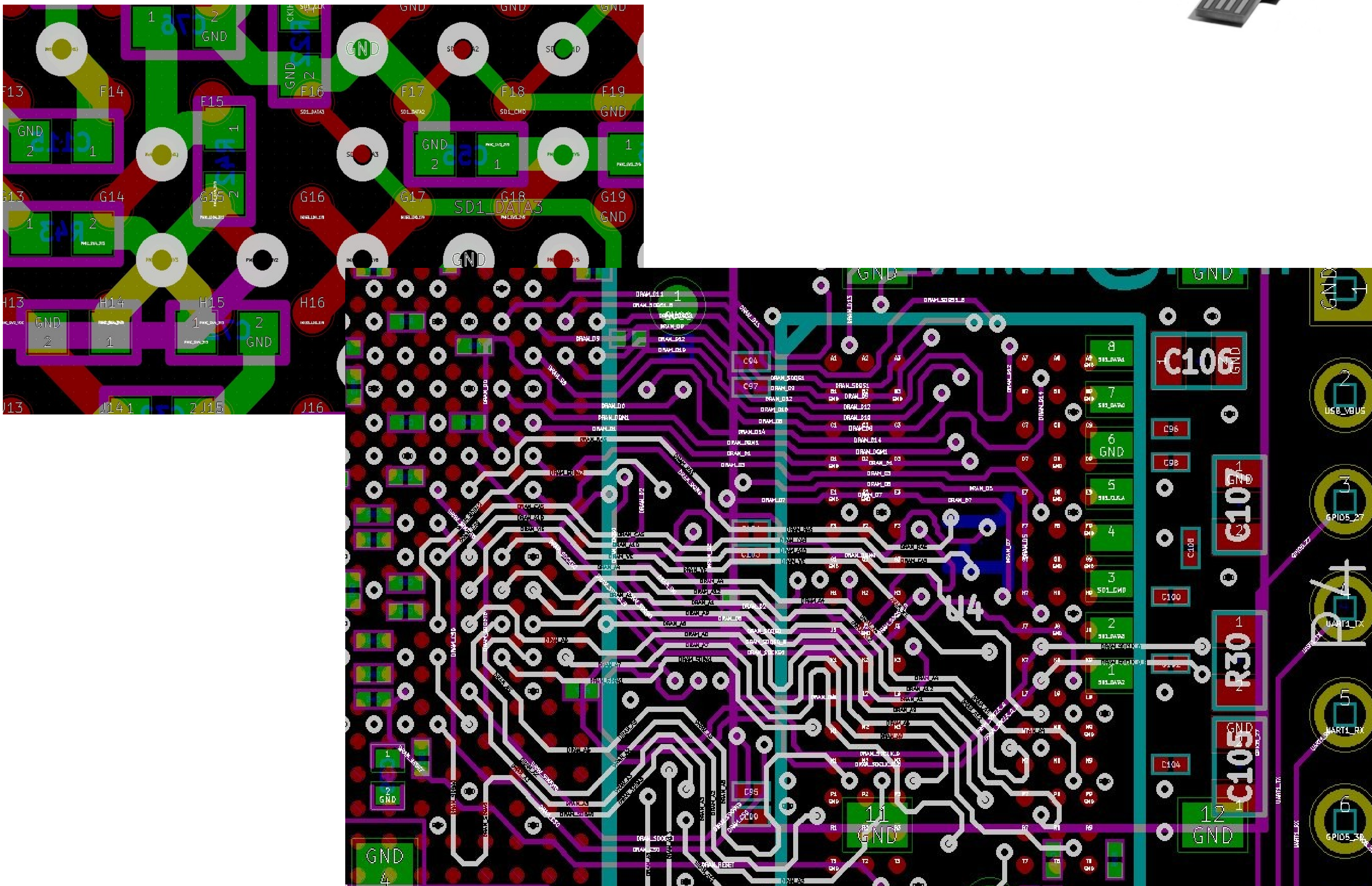
INVERSE PATH

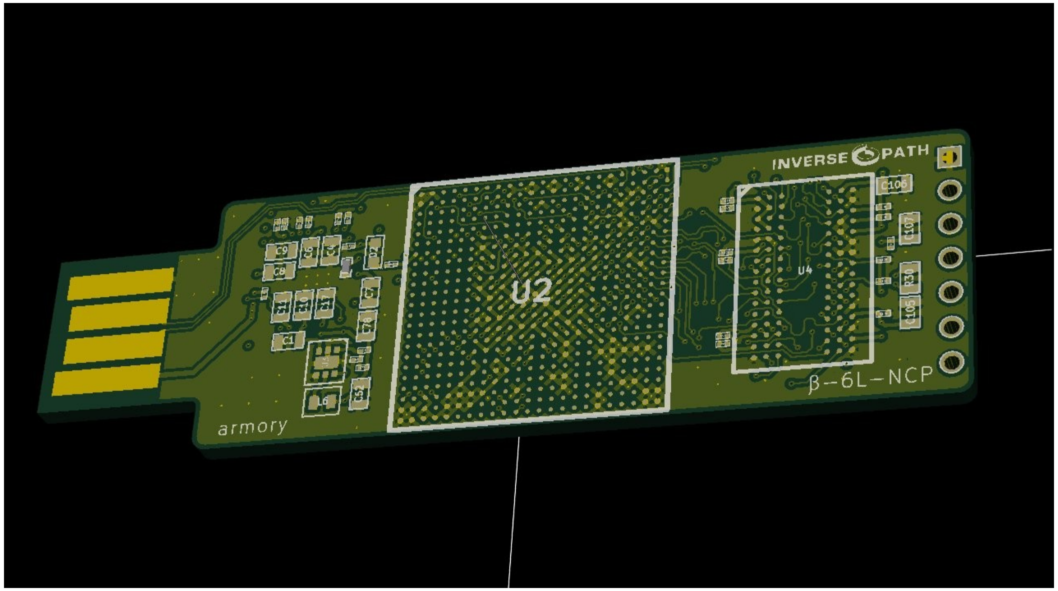


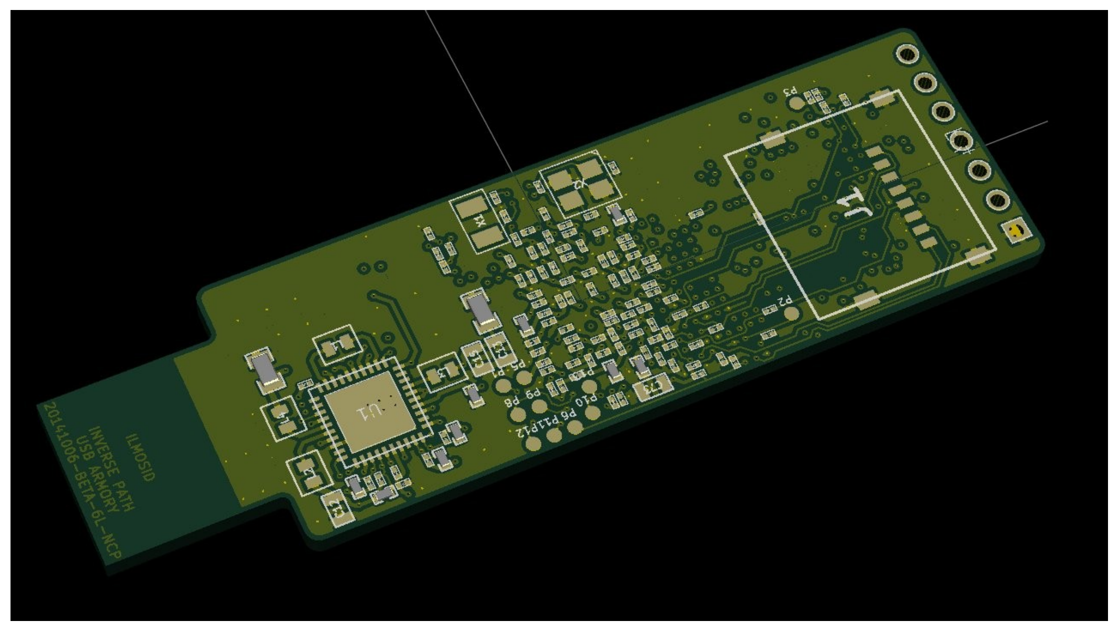




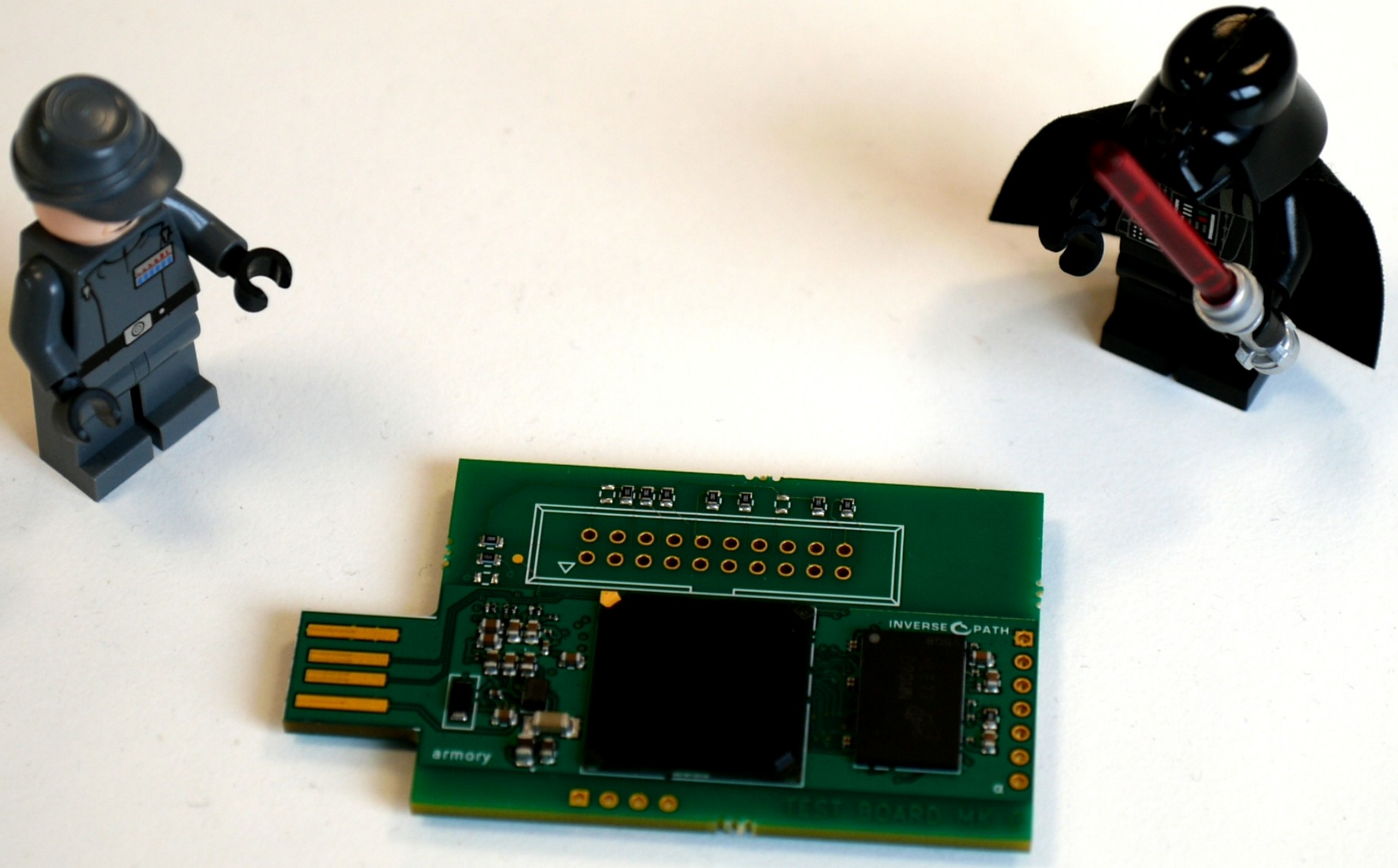
INVERSE PATH

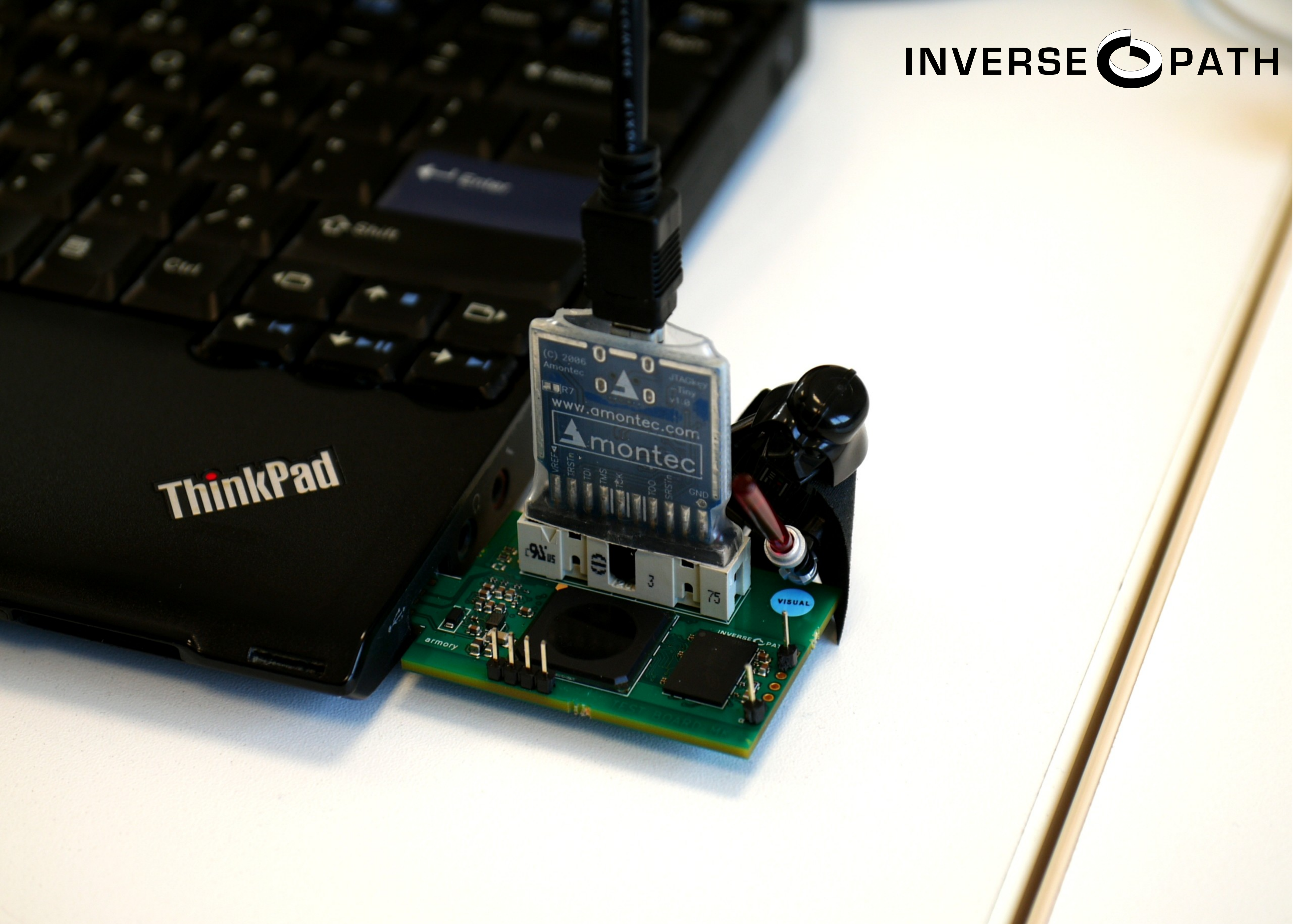



INVERSE PATH
U2
β--6L-NCP
armory


J1

INVERSE PATH




INVERSE PATH
ThinkPad
(C) 2006
Amontec
www.amontec.com
Amontec
VREF
TRSTn
TDI
TMS
TCK
TDO
SRSTn
GND
VISUAL
armory

INVERSE PATH

INVERSE PATH

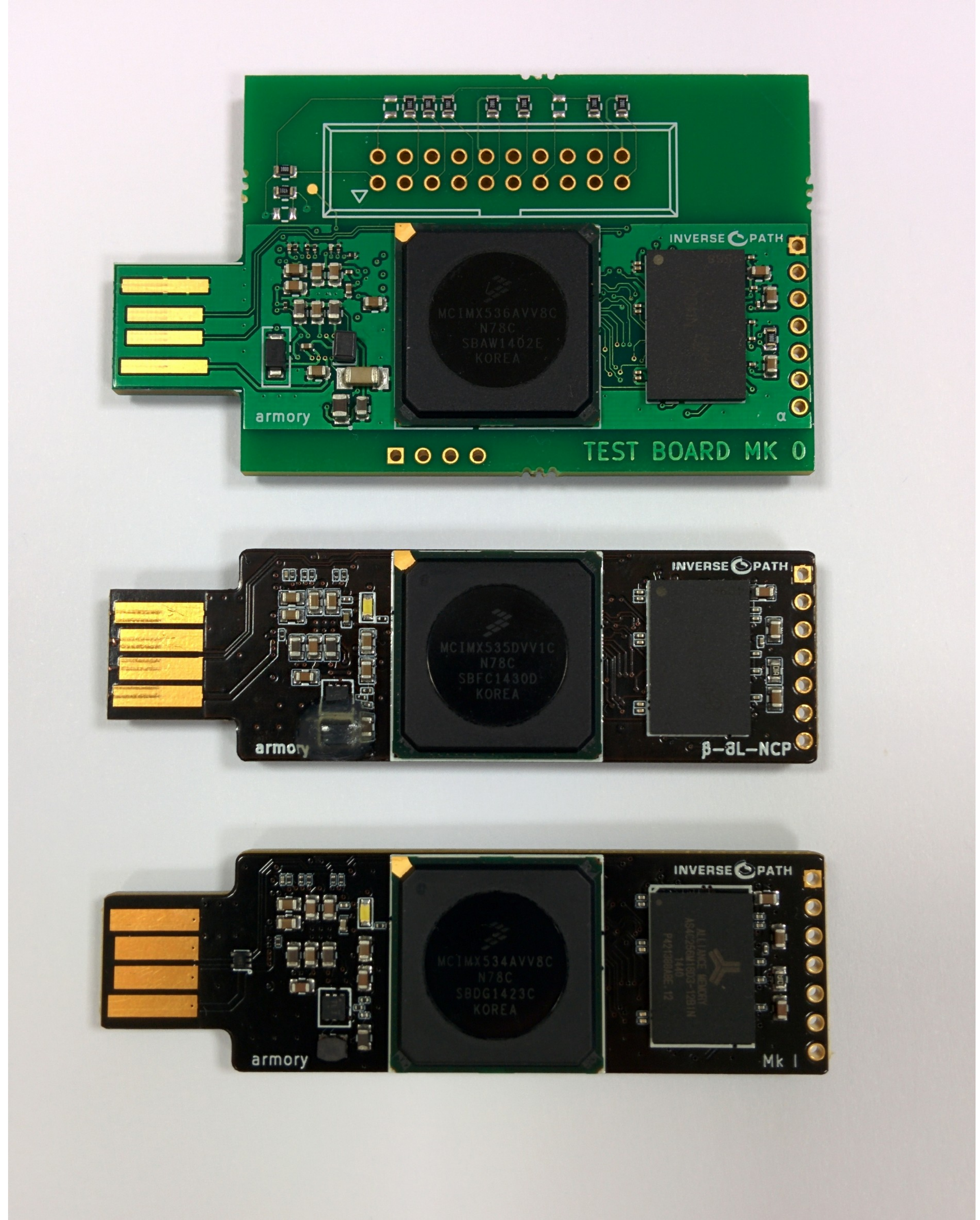
α

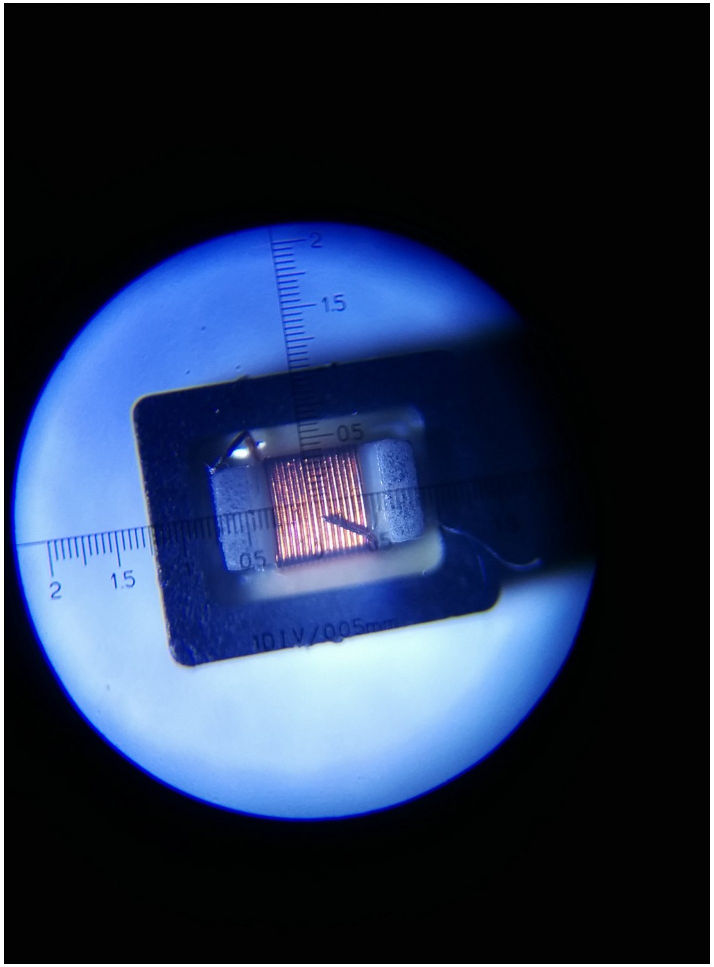
βs

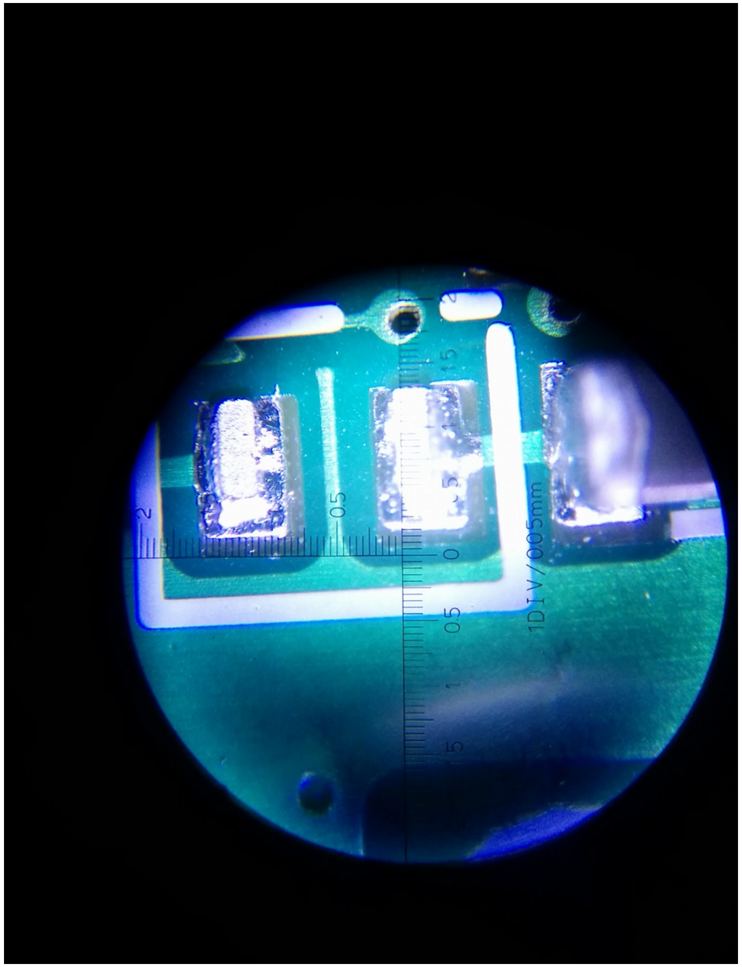
8L-NOUSBH,8L, 8L-DDR-LDO, 8L-DDR-NCP

6L, 6L-DDR-LDO, 6L-DDR-NCP

Mk I

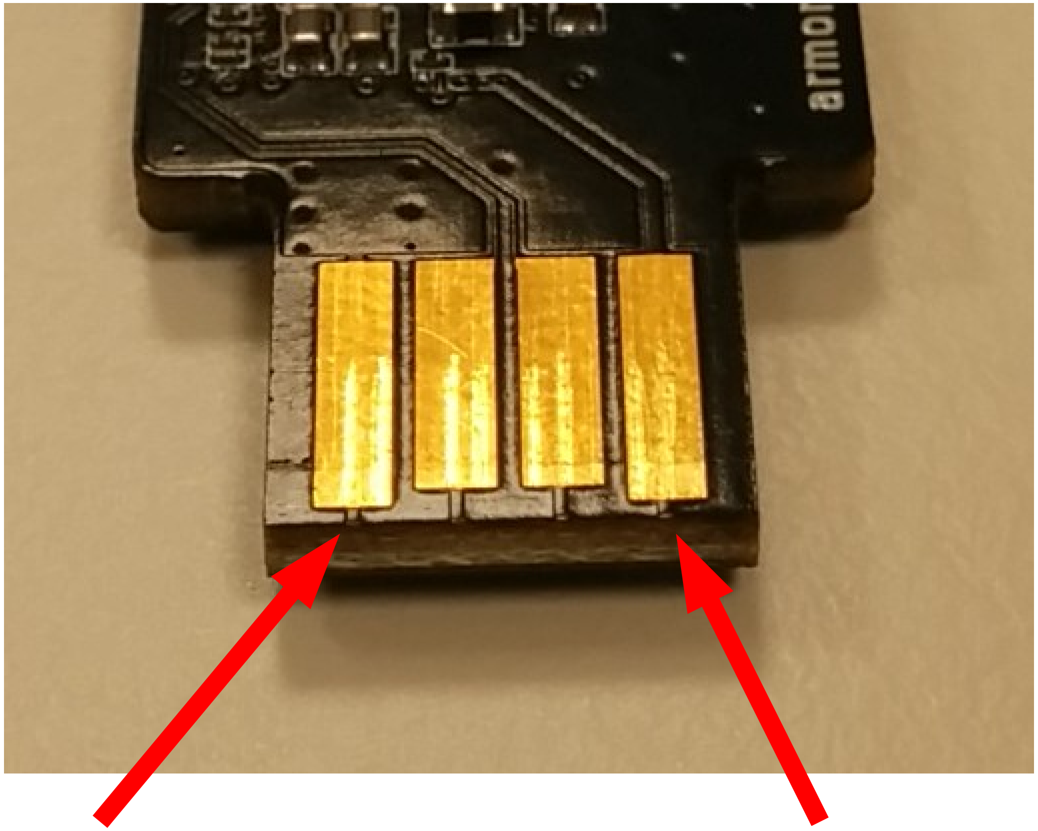




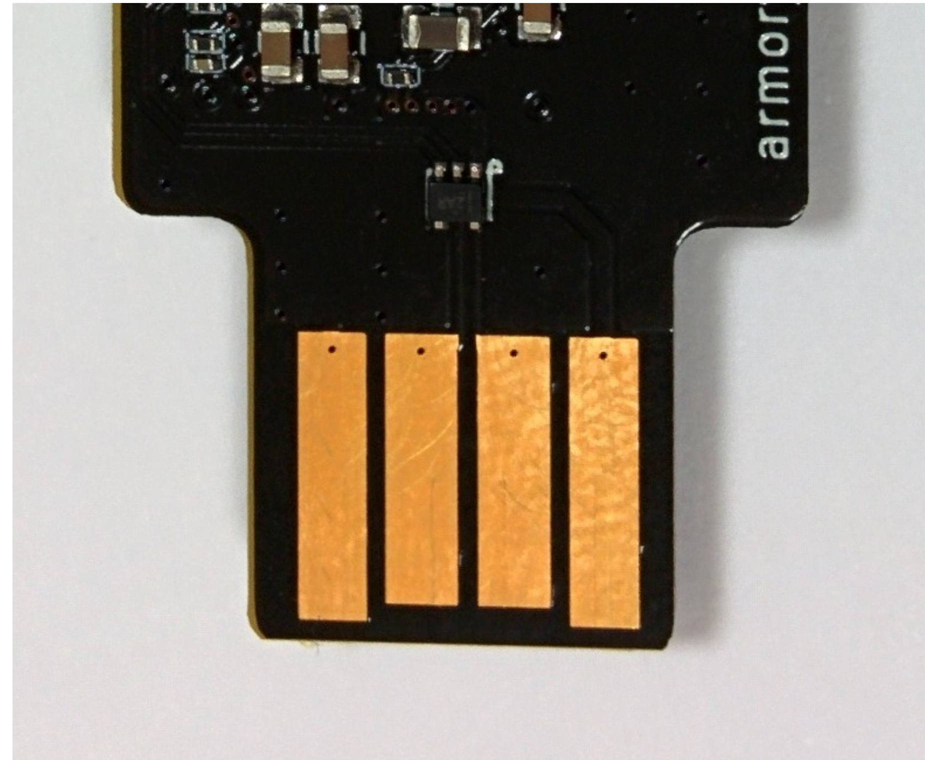


lessons learned #1

tiny inductors are fragile



evil

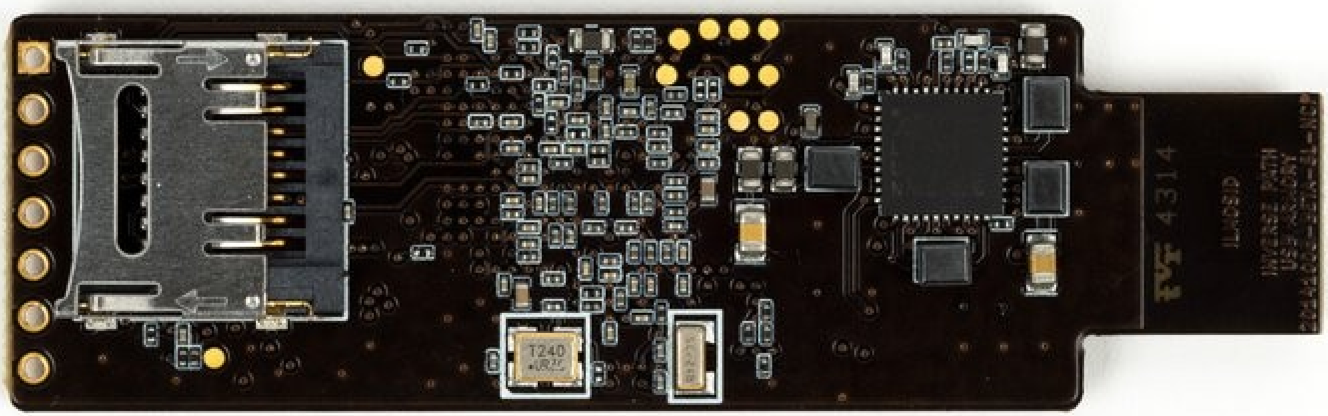


good

lessons learned #2 (the five-second rule)

gold plating traces cause under-voltage on hot swap


INVERSE PATH
MCIMX535DVV1C
N78C
SBEC1430D
KOREA
armory


T240
4314

Compiling and running Genode OS (>= 15.02):

```
git clone https://github.com/genodelabs/genode
cd genode

./tool/create_builddir hw_usb_armory
cd build/hw_usb_armory

# in etc/build.conf add "--include image/uboot" to RUN_OPT

make run/tz_vmm
cp var/run/tz_vmm/uImage $SD_CARD_MNT

uboot> ext2load mmc 0:1 0x70200000 /boot/uImage-genode; bootm 0x70200000
```

Requires minimally patch Normal world kernel compiled as follows:

```
make ARCH=arm zImage LOADADDR=0x80008000 modules
```

Secure Mode Monitor (LED example)

```
@ set GPIO4 to SECURE
        movw    r0, #0x33
        movt    r0, #0xff
        ldr     r1, =CSU_CSL
        add     r1, r1, #4      @ CSL1
        str     r0, [r1]

@ set IOMUXC to SECURE
        movw    r0, #0x33
        movt    r0, #0xff
        ldr     r1, =CSU_CSL
        add     r1, r1, #20     @ CSL5
        str     r0, [r1]

@ set OCRAM to SECURE
...
_secure_monitor:
        mov     r10, #0xcafe
        cmp     r0, r10
        beq     smc_handler
        beq     to_nonsecure
```

Secure Mode Monitor (LED example)

```
smc_handler:

        ldr     r10, =IOMUX_LED
        mov     r0, #1
        movt    r0, #0
        str     r0, [r10]               @ set the pad to GPIO

        ldr     r10, =GPIO4_DIR
        movw    r0, #0xffff
        movt    r0, #0xffff
        str     r0, [r10]               @ set direction to output

        ldr     r10, =GPIO4_DR
        ldr     r0, [r10]
        mvn     r0, r0
        str     r0, [r10]               @ toggle LED output

        movs    pc, lr
```

Secure Mode Monitor (LED example)

```
static int beg_for_led_switch(void)
{
        printk("dear smc, kindly switch the LED\n");

        /* give control to the secure monitor */
        asm volatile ("movw r0, #0xcafe");
        asm volatile ("smc #0");

        return 0;
}
```

The LED is hardware restricted via TrustZone to Secure monitor control.

A trivial interface implementation between Nonsecure Linux and Secure monitor illustrates a simple request for LED switching.

The USB armory SoC supports High Assurance Boot (HAB), enabling boot image verification.

Up to four public keys (SRK) are used to generate a SHA256 hash for verification, the hash is fused on the SoC with a permanent, irreversible operation.

Unlike Secure Boot on modern PCs the activation can not be reset. This is a feature, not a bug.

Up to 3 keys out of 4 can be revoked.

Fuse name	IIM bank	IIM addr[bits]	Function
SRK_HASH[255:248]	1	0x0c04	SRK table hash (part 1)
SRK_HASH[247:160]	3	0x1404-0x142c	SRK table hash (part 2)
SRK_HASH[159:0]	3	0x1430-0x147c	SRK table hash (part 3)
SRK_LOCK	1	0x0c00[2]	lock for SRK_HASH[255:248]
SRK_LOCK88	3	0x1400[1]	lock for SRK_HASH[247:160]
SRK_LOCK160	3	0x1400[0]	lock for SRK_HASH[159:0]
SRK_REVOKE[2:0]	4	0x1810[2:0]	SRK keys revocation
SEC_CONFIG[1:0]	0	0x0810[1:0]	Security configuration
DIR_BT_DIS[1:0]	0	0x0814[0]	Direct external memory

```
# syntax: fuse prog [-y] <bank> <word> <hexval> [<hexval>...]
          program 1 or several fuse words, starting at 'word'
         (PERMANENT)

=> fuse prog -y 1 0x1 0xaa
=> fuse prog -y 3 0x1 0xbb 0xcc 0xdd 0xee 0xff 0xaa 0xbb 0xcc 0xdd 0xee 0xff
=> fuse prog -y 3 0xc 0xaa 0xbb 0xcc 0xdd 0xee 0xff 0xaa 0xbb 0xcc 0xdd 0xee
=> fuse prog -y 3 0x17 0xff 0xaa 0xbb 0xcc 0xdd 0xee 0xff 0xaa 0xbb
```

Secure Boot (success)

```
U-Boot 2015.07 (Sep 10 2015 - 14:26:37 +0200)

CPU:   Freescale i.MX53 rev2.1 at 800 MHz
Reset cause: POR
Board: Inverse Path USB armory MkI
I2C:   ready
DRAM:  512 MiB
MMC:   FSL_SDHC: 0
In:    serial
Out:   serial
Err:   serial
Net:   CPU Net Initialization Failed
No ethernet found.
Hit any key to stop autoboot:  2
=> hab_status.

Secure boot enabled

HAB Configuration: 0xcc, HAB State: 0x99
No HAB Events Found!

=> boot
2301352 bytes read in 300 ms (7.3 MiB/s)
16670 bytes read in 178 ms (90.8 KiB/s)
## Booting kernel from Legacy Image at 70800000 ...
   Image Name:   Linux-4.2.0
```

Secure Boot (failure)

```
U-Boot 2015.07 (Sep 10 2015 - 14:26:37 +0200)

CPU:   Freescale i.MX53 rev2.1 at 800 MHz
Reset cause: POR
Board: Inverse Path USB armory MkI
I2C:   ready
DRAM:  512 MiB
MMC:   FSL_SDHC: 0
In:    serial
Out:   serial
Err:   serial
Net:   CPU Net Initialization Failed
No ethernet found.
Hit any key to stop autoboot:  2
=> hab_status.

Secure boot enabled

HAB Configuration: 0xf0, HAB State: 0x66

--------- HAB Event 1 -----------------
event data: ...

STS = HAB_FAILURE (0x33)
RSN = HAB_INV_SIGNATURE (0x18)
CTX = HAB_CTX_COMMAND (0xC0)
ENG = HAB_ENG_ANY (0x00)
```

INTERLOCK

http://github.com/inversepath/interlock

Open source file encryption front-end developed, but not limited to, usage with the USB armory.

Provides a web accessible file manager to unlock/lock LUKS encrypted partition and perform additional symmetric/asymmetric encryption on stored files.

Take advantage of disposable passwords, “nuking” option.

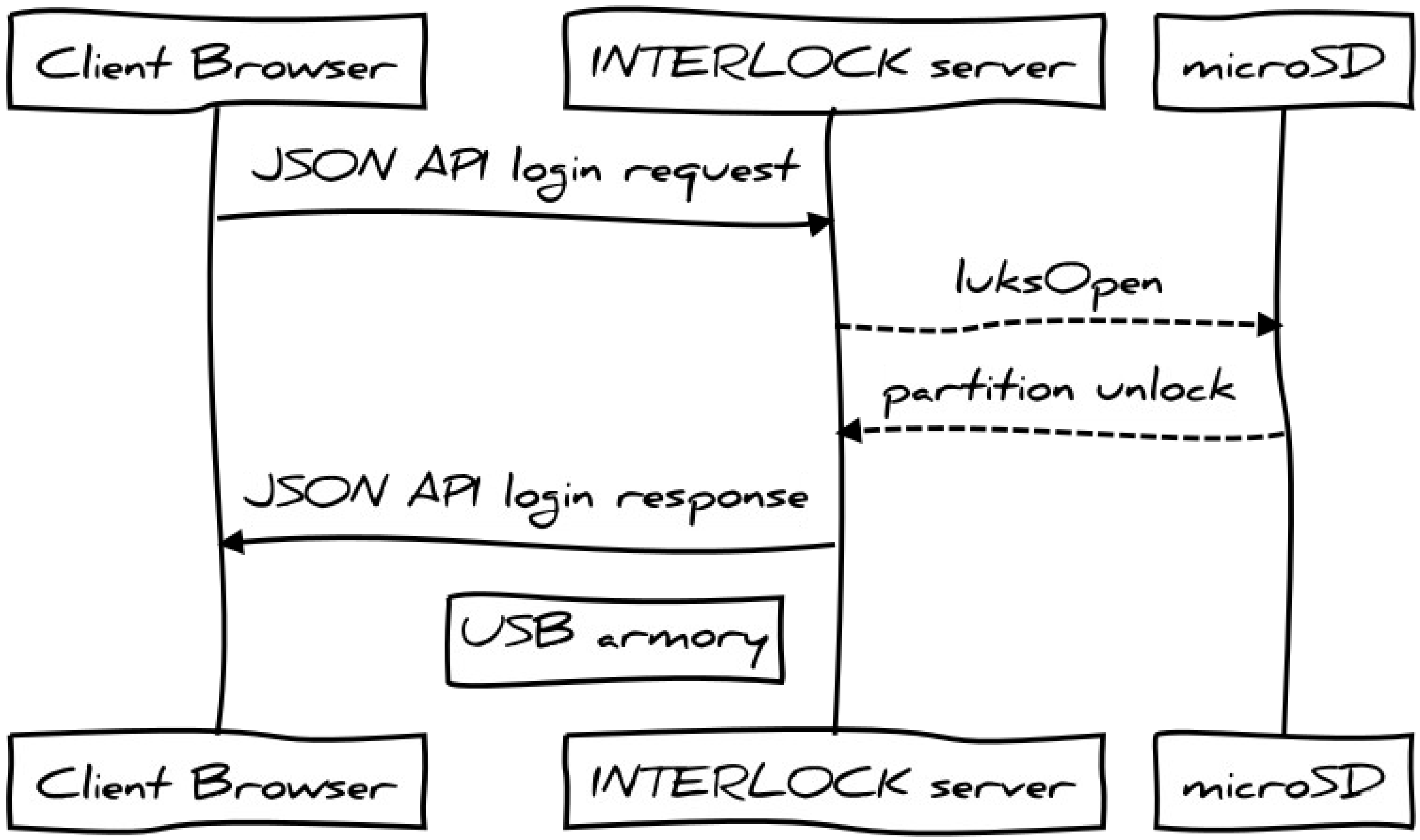
Design Goals

Clear separation between presentation and server layer to ease auditability and integration.

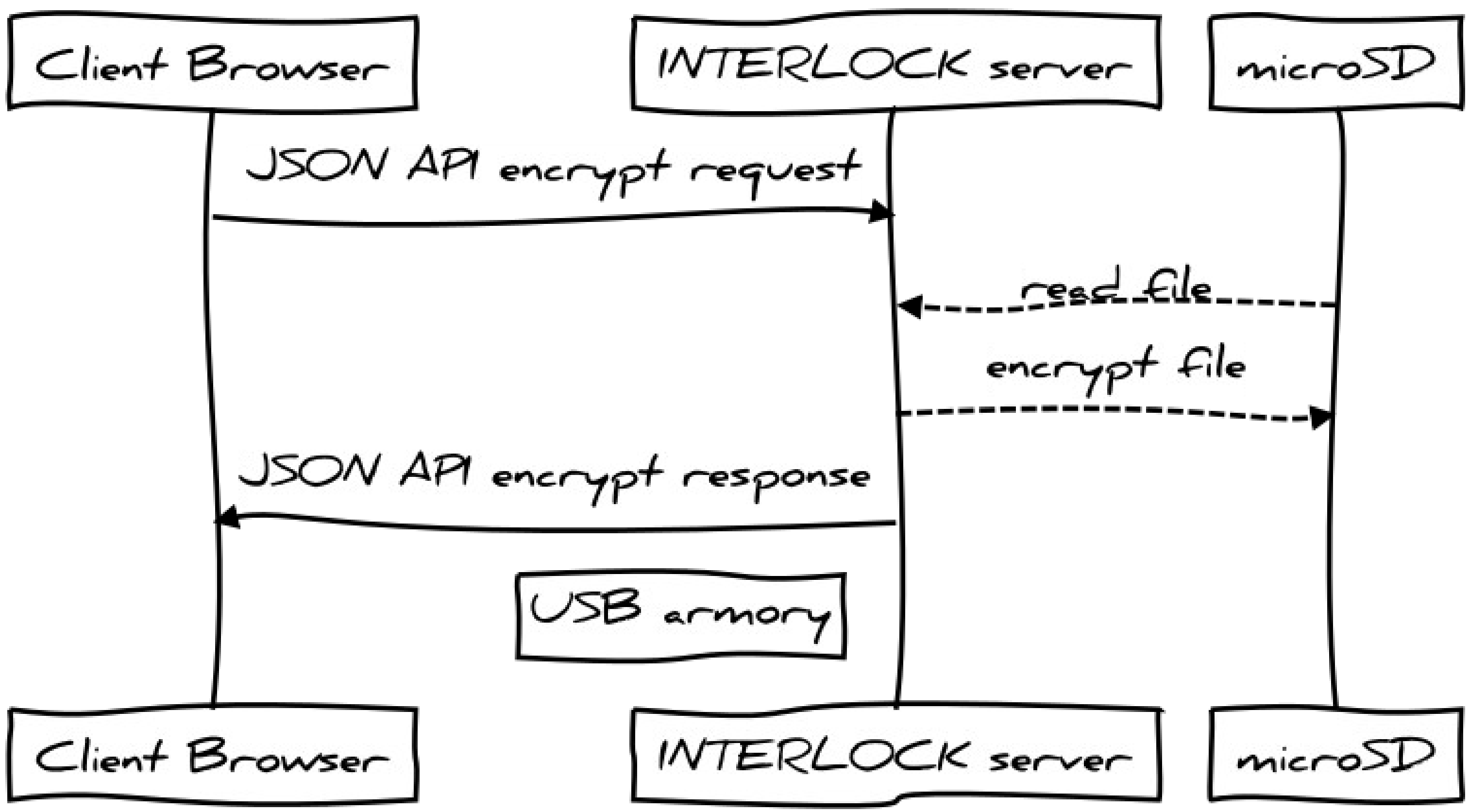
Minimum amount of external dependencies and footprint.

Encrypted volumes:	LUKS encrypted partitions
Asymmetric ciphers:	OpenPGP
Symmetric ciphers:	AES-256-OFB w/ PBKDF2 + HMAC
Security tokens:	Time-based One-Time Password (Google Authenticator)
Messaging:	TextSecure/Signal

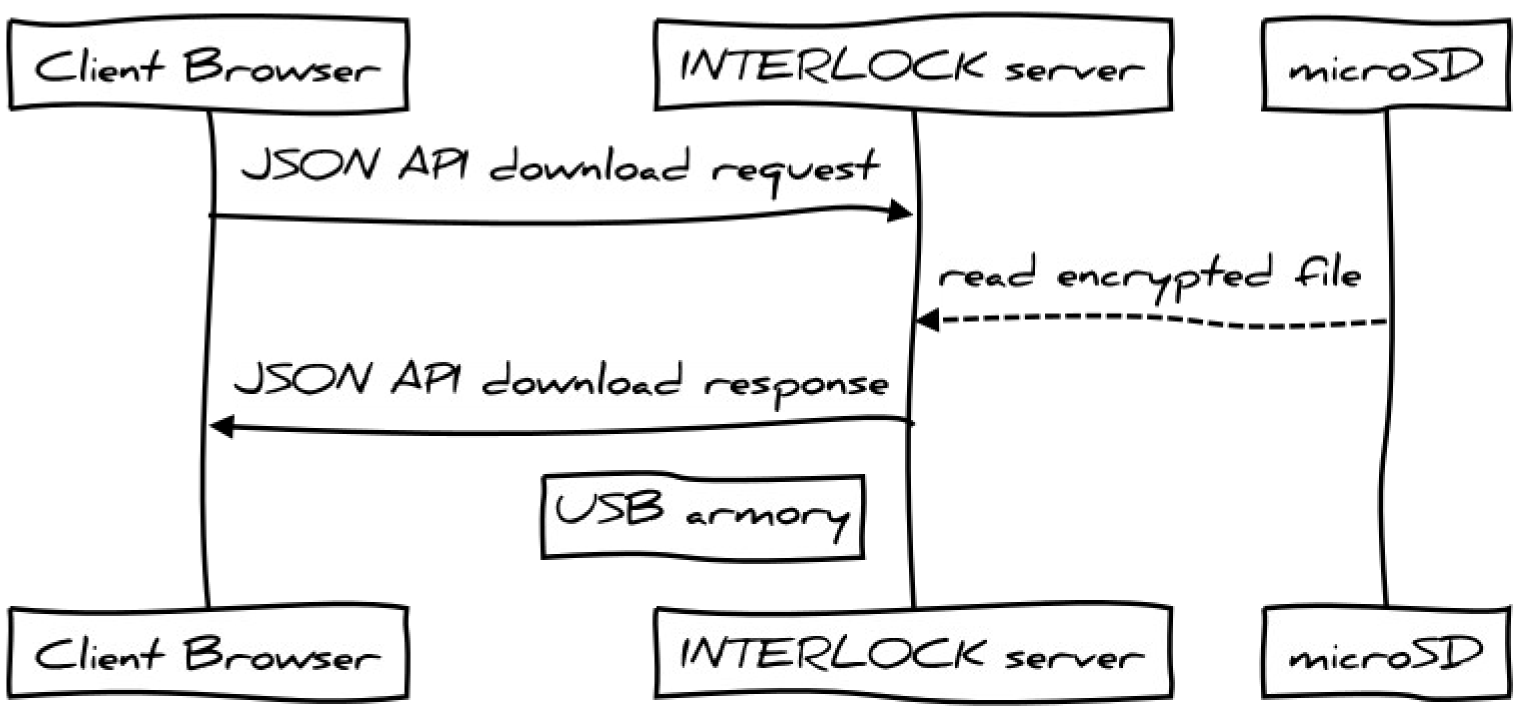
Authentication credentials are directly tied to LUKS partition.



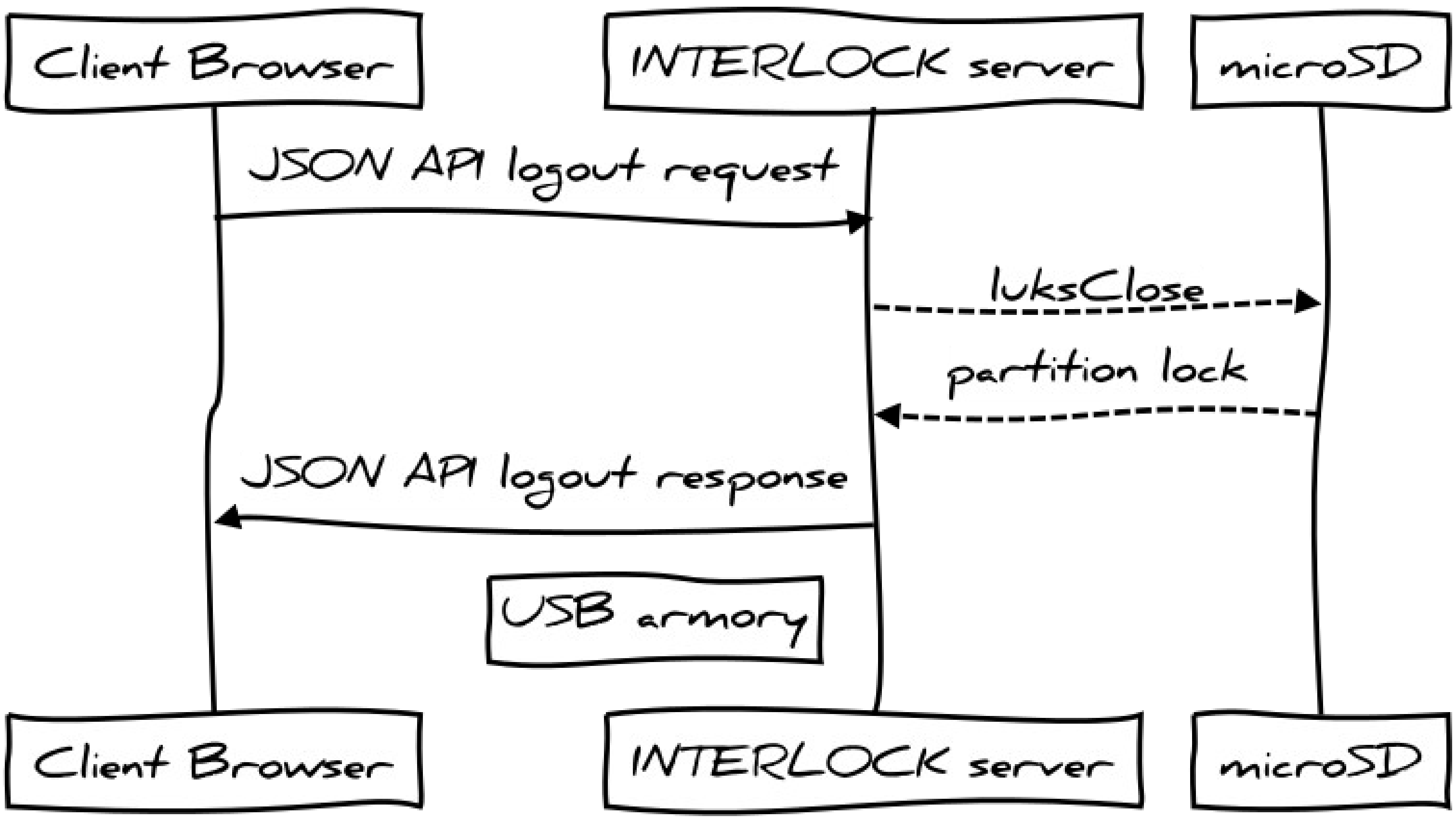
Files can be further encrypted on the USB armory…



…and later downloaded.



Logging out locks the encrypted partition.



INVERSE PATH

Thank you!

Q & A

Andrea Barisani

<andrea@inversepath.com>