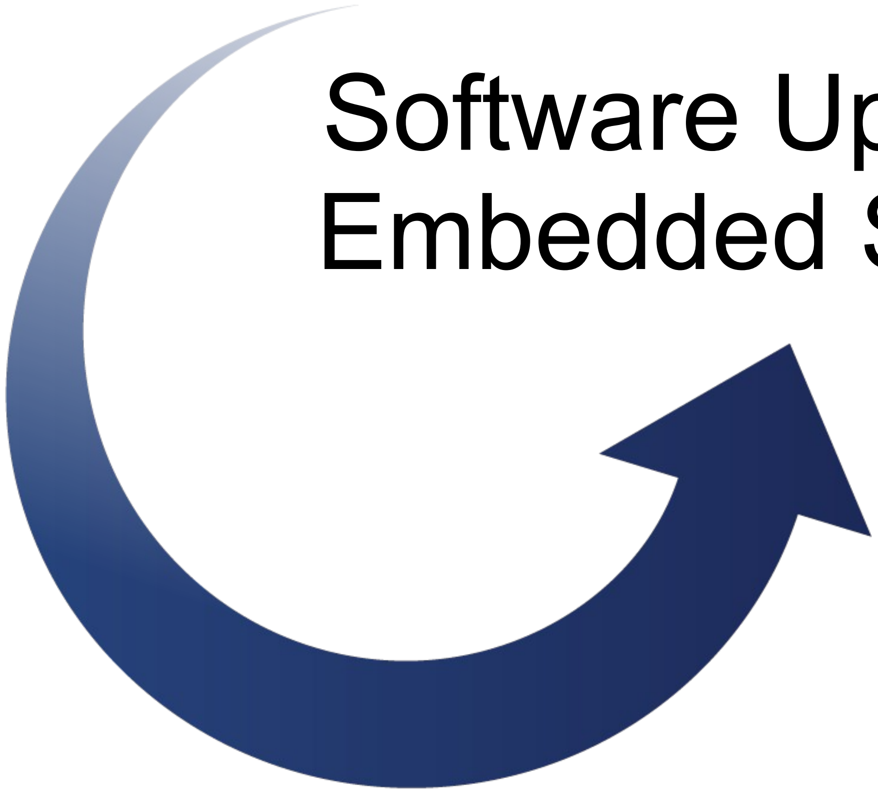
Software Update on Embedded Systems



Do not brick your device

Stefano Babic

ELCE October 2014

Introduction

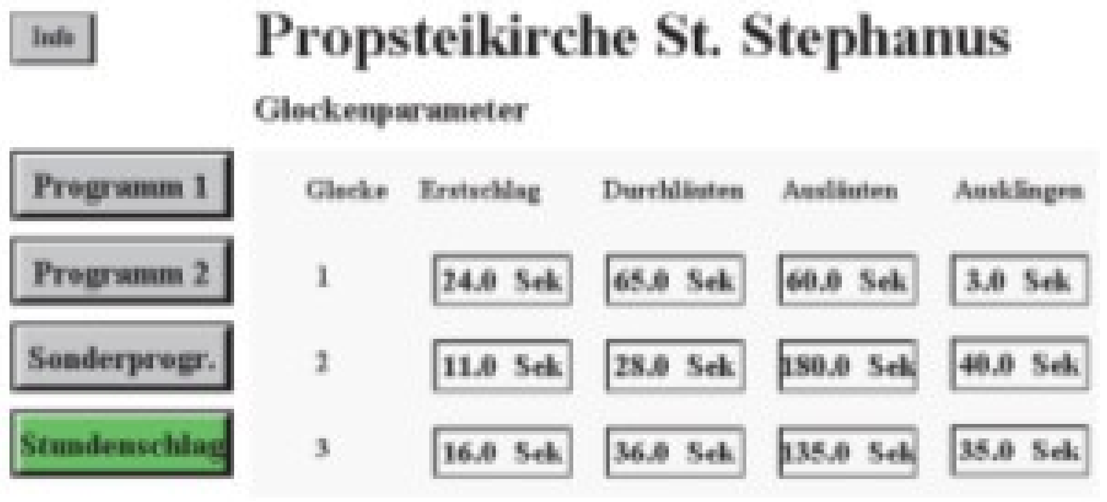
- Me:
  - Software Engineer at DENX, Gmbh
  - U-Boot Custodian for Freescale's i.MX
  - Focus on Linux embedded with PowerPC and ARM processors.

Agenda

- Why upgrade ?
- Why is it different with a Linux-PC ?
- Upgrading strategies
- Swupdate

Why do we need to update an embedded system ?

- It is not only hardware
- Bug fixes
- New features can be added
- Security issues : heartbleed, bad implementation...



Propsteikirche St. Stephanus

Glockenparameter

Glocke	Erstschlag	Durchläuten	Ausläuten	Ausklingen
1	24.0 Sek	65.0 Sek	60.0 Sek	3.0 Sek
2	11.0 Sek	28.0 Sek	180.0 Sek	40.0 Sek
3	16.0 Sek	36.0 Sek	135.0 Sek	35.0 Sek

Why is ES different ?

- Power failure
- Bad firmware
- Communication errors in case of remote update
- No access to target

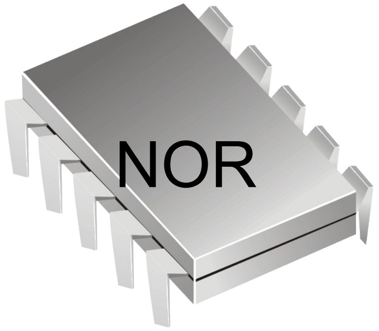
Target must recover from errors !

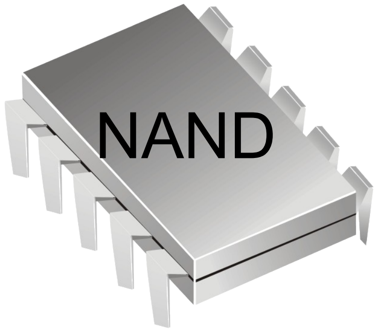
Which elements must be updated ?

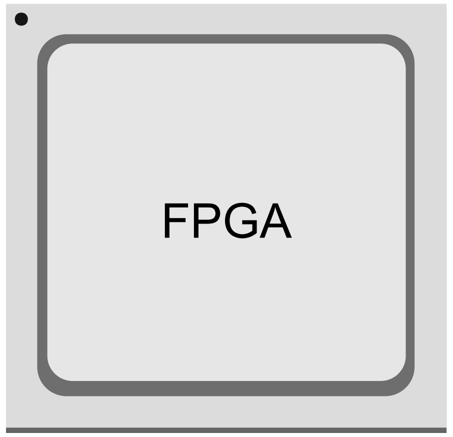
- Bootloader (dangerous !)
- Kernel + DT
- Root filesystem
- Application data, other filesystems..
- Customer data (migration)
- Specific software (FPGA bitstream,...)

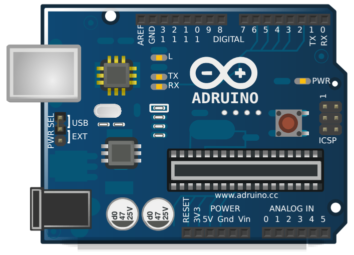
Where is a new SW installed ?

denx software engineering





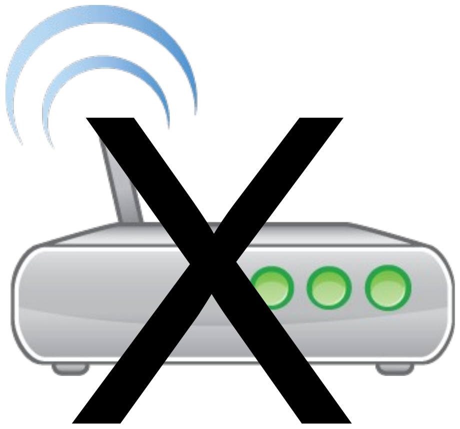




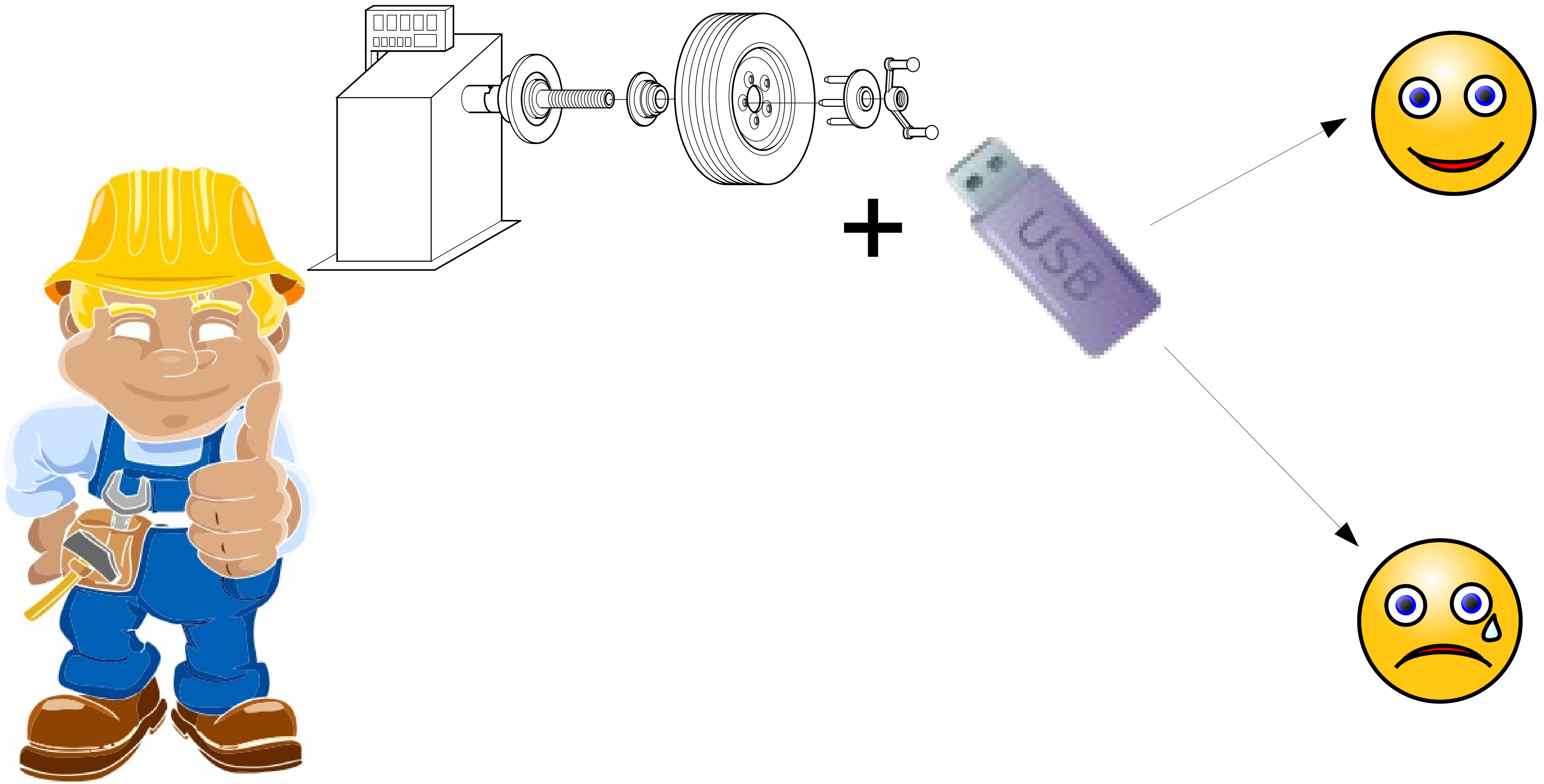
Which interface ?

- Local:
  - Local storage (USB, SD,..)
  - Local peripheral (USB as device, UART,..)
- Remote:
  - HTTP / web based
  - FTP
  - Proprietary protocol
  - Many more...

Who will update ?



No expertise required



System upgrade solutions

- Bootloader upgrade
- Linux upgrade
  - Package Manager
  - Rescue image or specific application
  - From the running application

Bootloader

- ✖ Limited access to peripherals (drivers, filesystems)
- ✖ Implementation in bootloader not in sync with Linux
- ✖ Limited network support (UDP, not TCP)
- ✖ Limited UI with an operator

- ✓ Update is simpler
- ✓ Smaller footprint

Linux App

- ✖ Footprint

- ✓ Availability of all drivers used by the product
- ✓ A lot of tools/libraries

package manager as distro ?

- ✖ Upgrade is not atomic
- ✖ Nightmare for test engineers/support
- ✖ New firmware partially written
- ✖ More places where things can go wrong

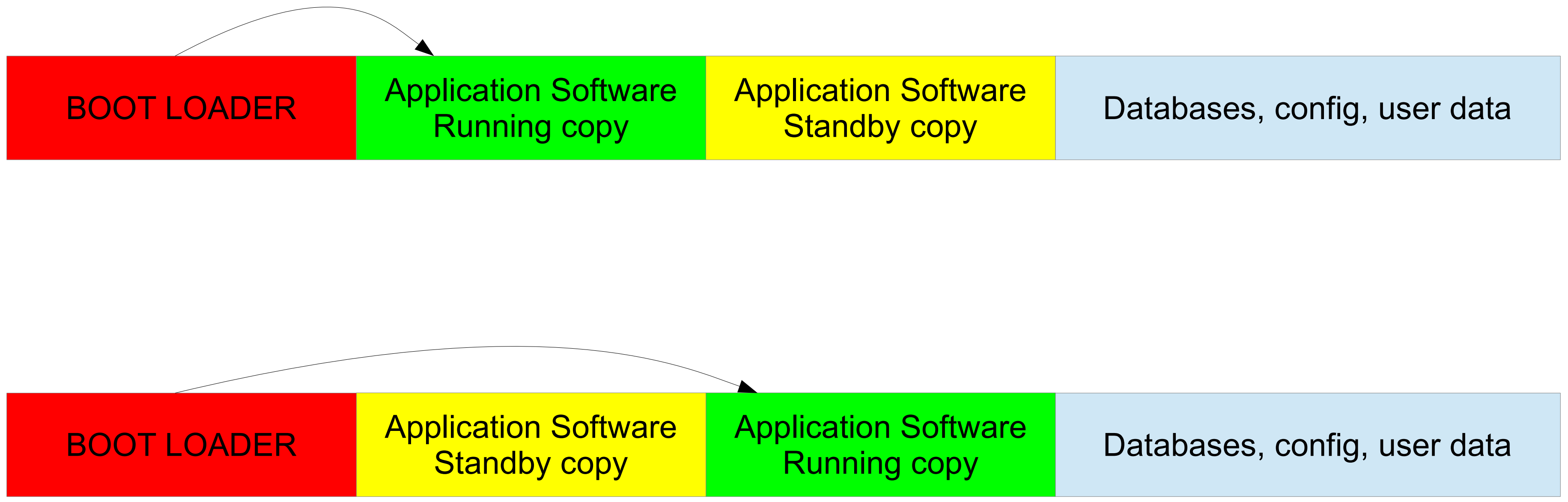
- ✓ Small update image

Full update

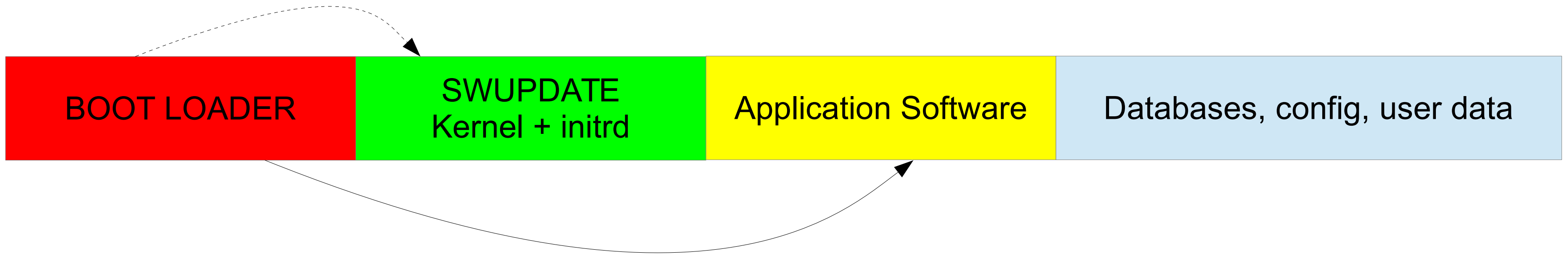
- ✖ Size, Time to transfer

- ✓ Atomic: it works or not
- ✓ Single image delivery

Double copy strategy


BOOT LOADER
Application Software Running copy
Application Software Standby copy
Databases, config, user data
BOOT LOADER
Application Software Standby copy
Application Software Running copy
Databases, config, user data

Single copy (rescue)



swupdate: FLOSS upgrade sw

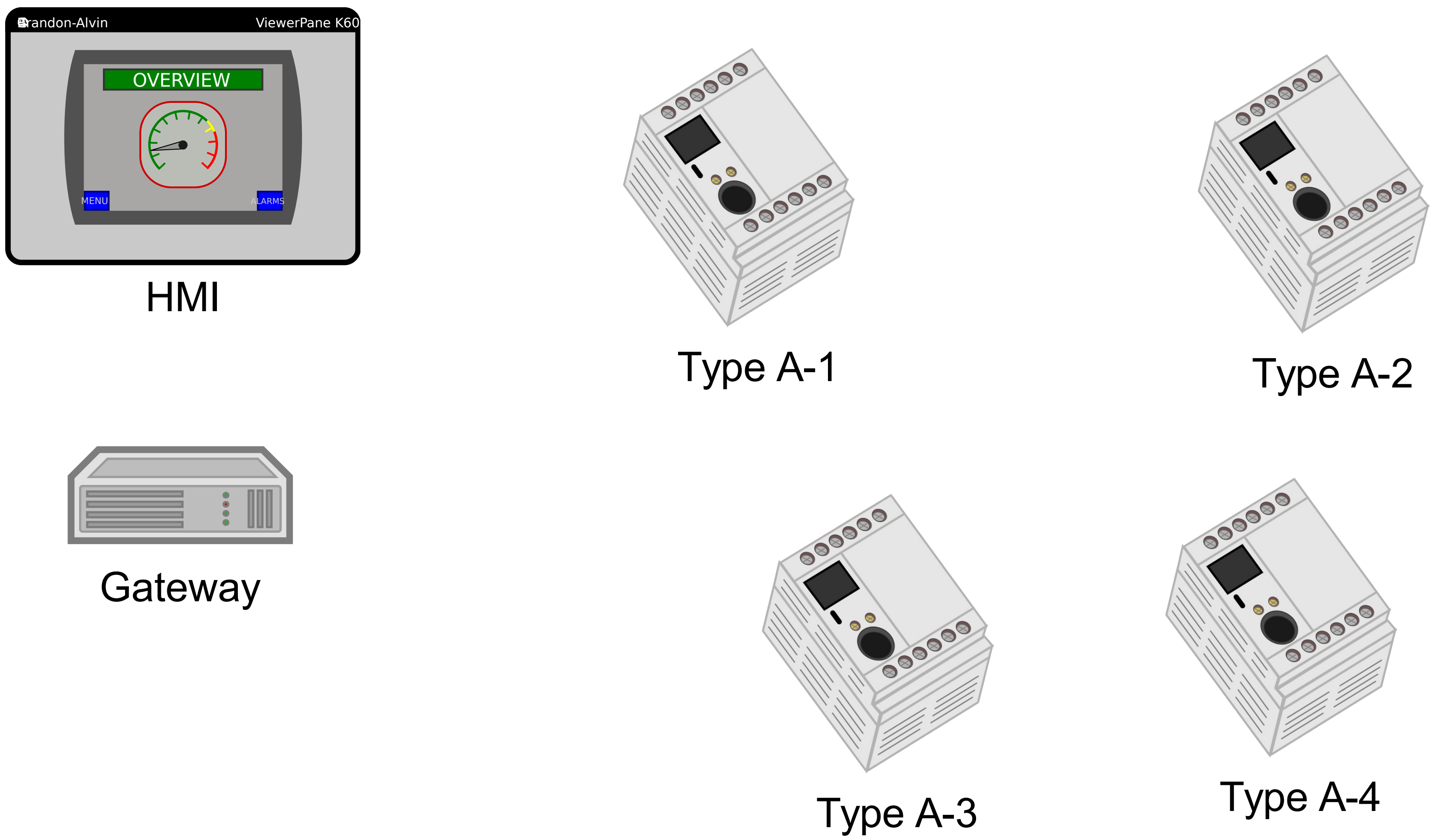
denx software engineering

- Missing an open source upgrade software for ES
- Take care of failure mechanism
- Hardware / software compatibility
- Proof correctness images to be installed (chksum,..)
- Partitioning storage
- Local or remote install

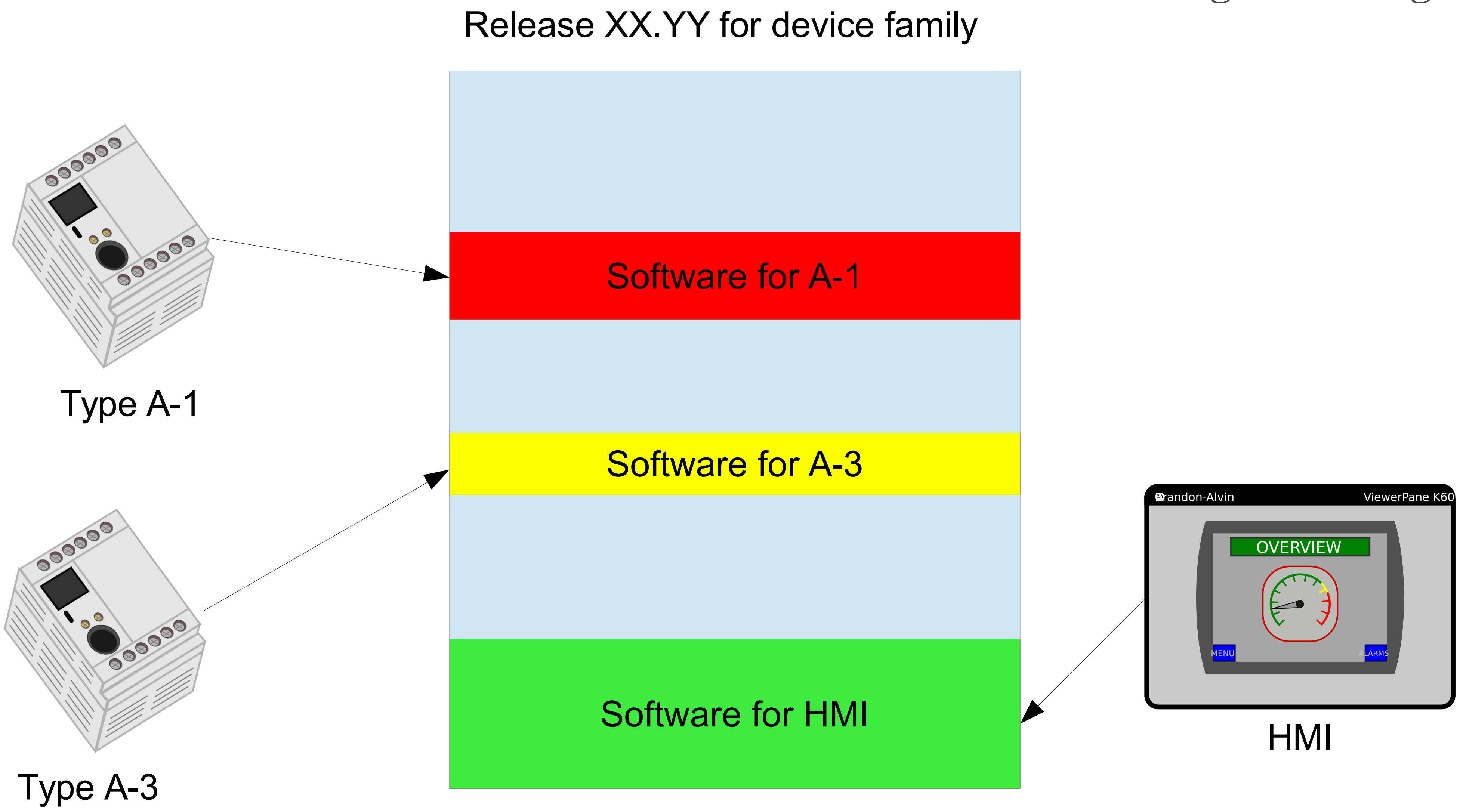
Swupdate-2

- Scriptable (LUA), pre- and postinstall scripts
- Single image for multiple devices
- Easy for users to perform update
- Missing : signed images !

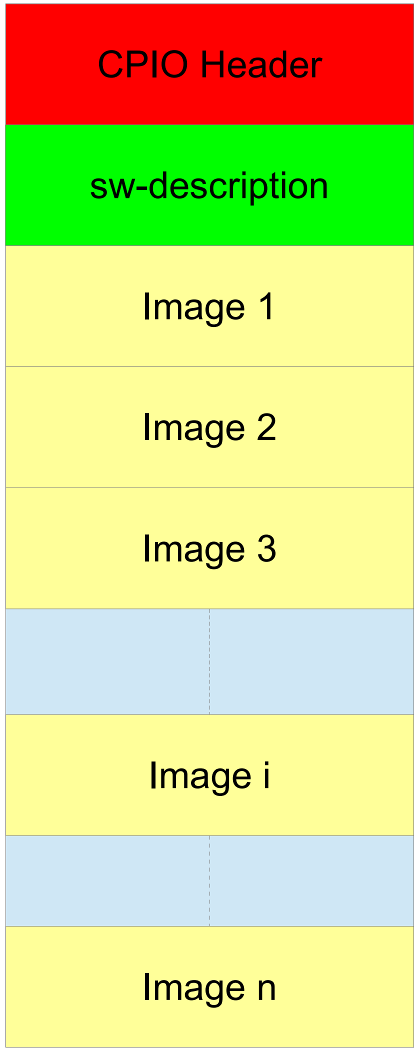
Handling hardware differences



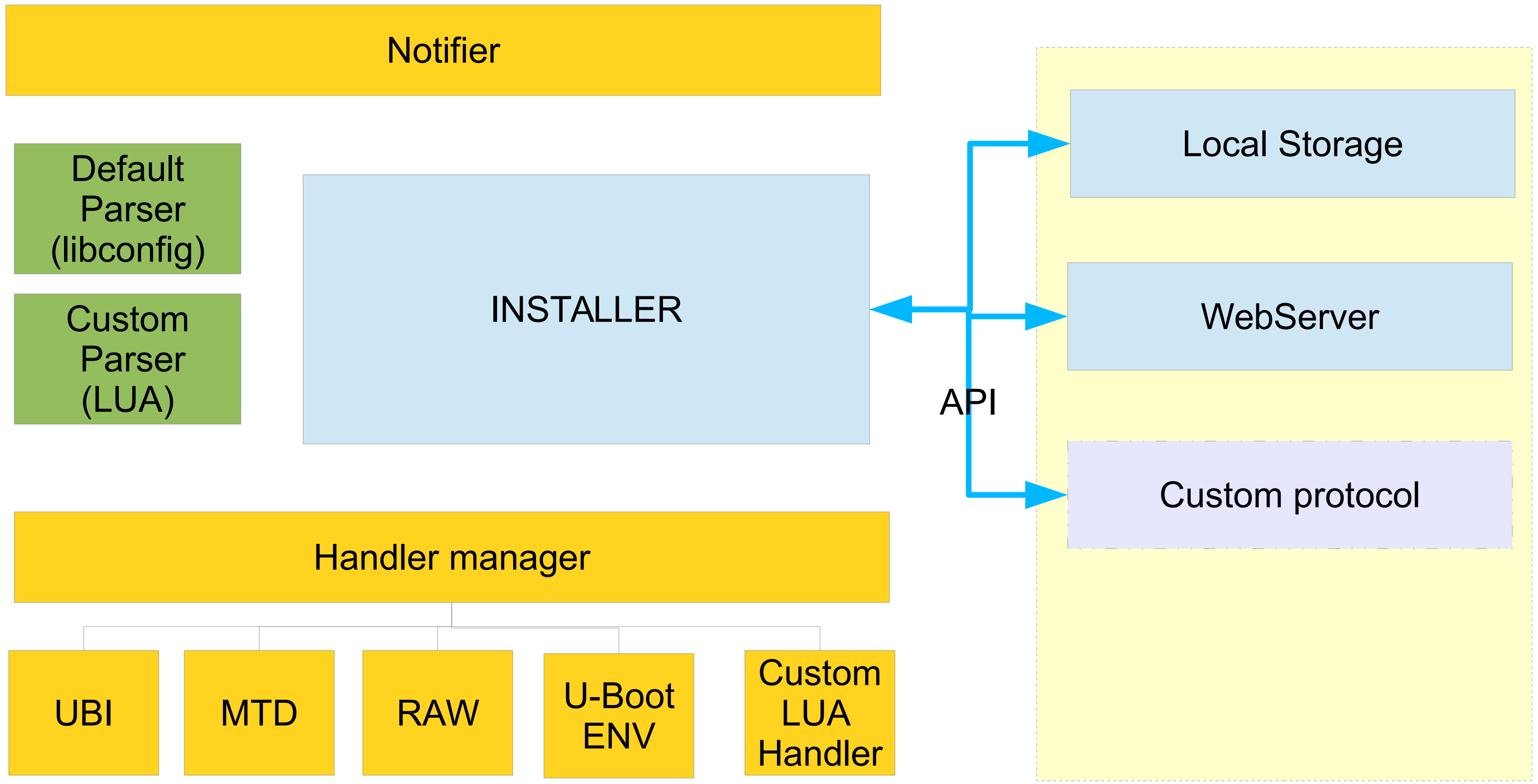
One release, multiple devices



Single image structure



Swupdate architecture



Handling HW differences

```
software =
{
    version = "0.1.0";

    target-1 = {
        images: (
            {
                ...
            }
        );
    };

    target-2 = {
        images: (
            {
                ...
            }
        );
    };
}
```

sw-description

```
software =
{
	version = "0.1.0";

	myboard = {

		hardware-compatibility: [ "1.2", "1.3", "18#010071"];

		partitions: ( /* UBI Volumes */
			{
				name = "rootfs";
				device = "mtd10";
				size = 104896512; /* in bytes */
			},
			{
				name = "kernel";
				device = "mtd9";
				size = 4194304; /* in bytes */
			}
		);
```

sw-description

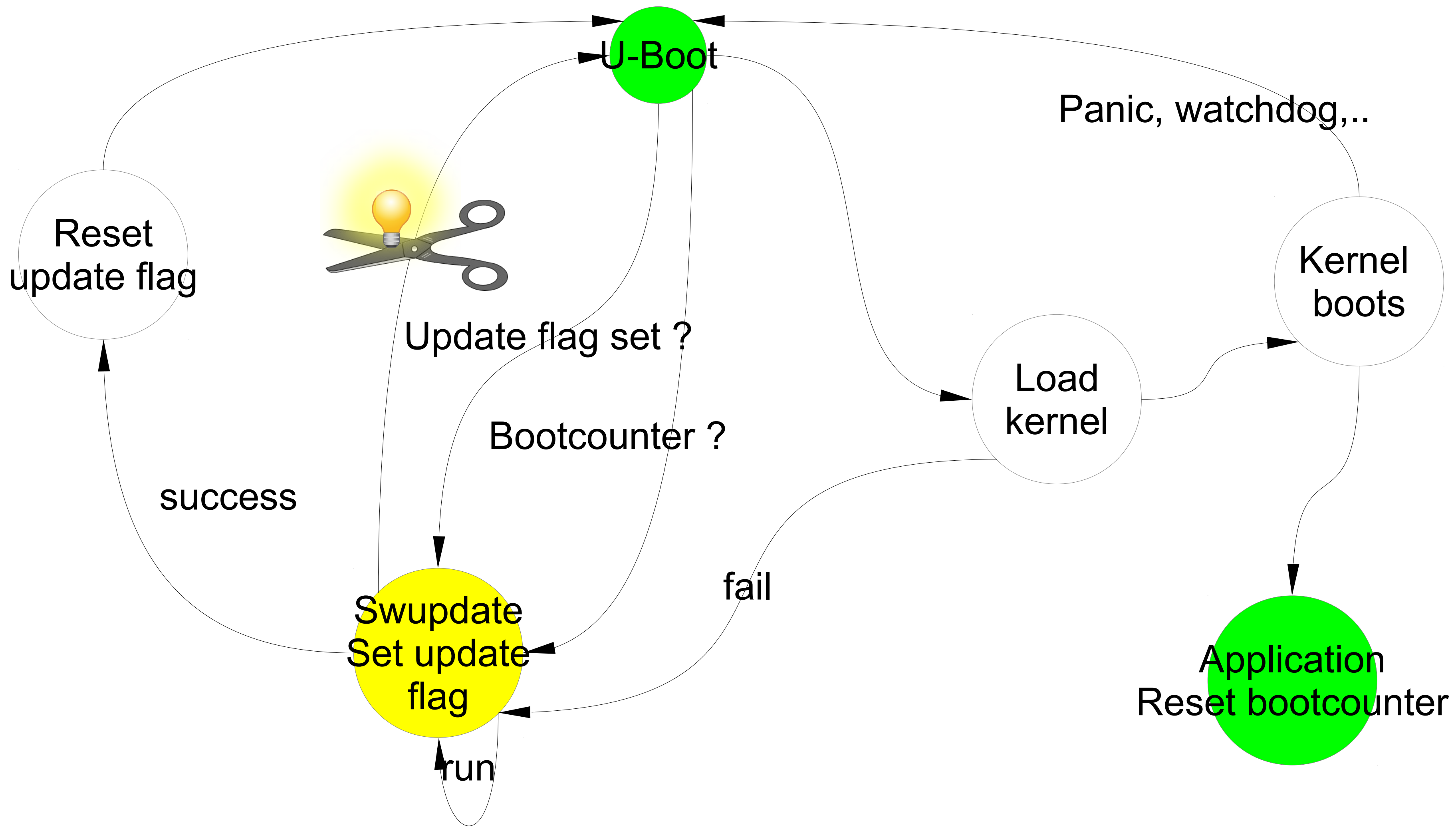
```
images: (
	{
		filename = "core-image-base-myboard.ubifs";
		volume = "rootfs";
	},
	{
		filename = "uboot-env";
		type = "uboot";
	},
	{
		filename = "uImage";
		volume = "kernel";
	},
	{
		filename = "fpga.bin";
		type = "fpga";
	}
);
```

sw-description: scripts, u-boot

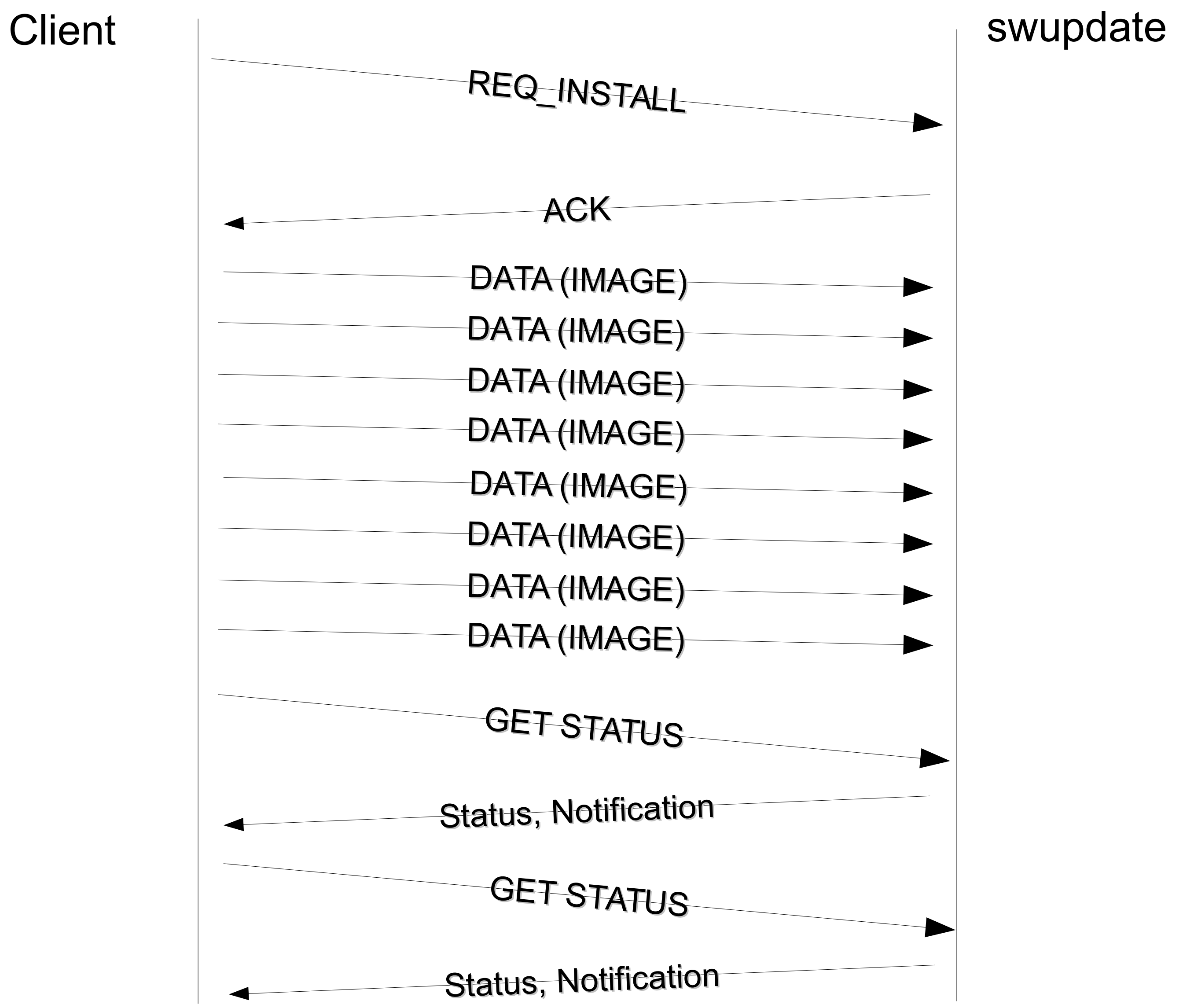
```
scripts: (
		{
			filename = "test.lua";
			type = "lua";
		},
		{
			filename = "sdcard.lua";
			type = "lua";
		},
		{
			filename = "test_shell.sh";
			type = "shellscript";
		}
);

uboot: (
		{
			name = "vram";
			value = "4M";
		}
```

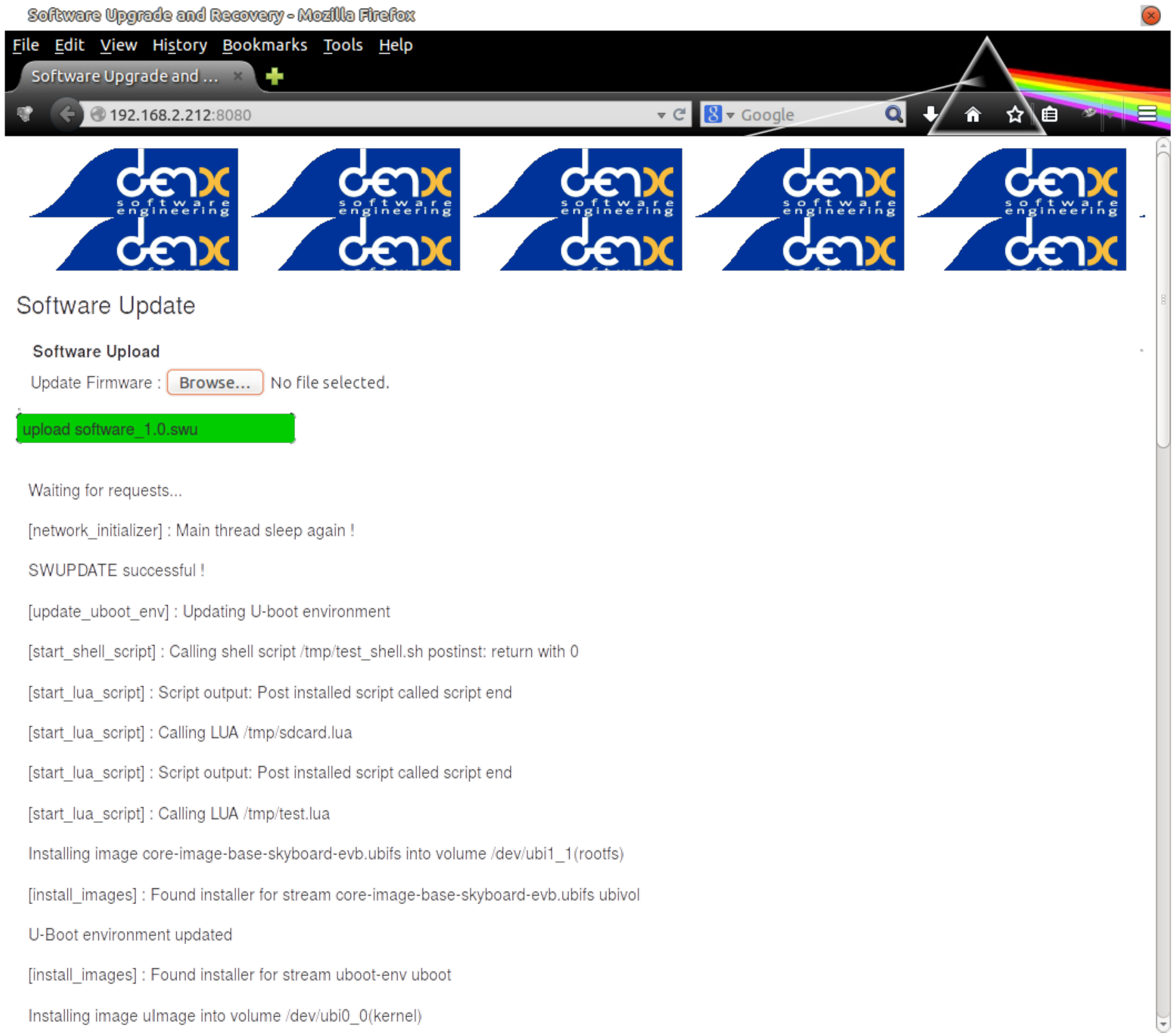
Recovery from failures



API for external client



Updating from browser


Software Upgrade and Recovery - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Software Upgrade and ...
192.168.2.212:8080
Google

Software Update

Software Upload

Update Firmware : Browse... No file selected.

upload software_1.0.swu

```
Waiting for requests...
[network_initializer] : Main thread sleep again !
SWUPDATE successful !
[update_uboot_env] : Updating U-boot environment
[start_shell_script] : Calling shell script /tmp/test_shell.sh postinst: return with 0
[start_lua_script] : Script output: Post installed script called script end
[start_lua_script] : Calling LUA /tmp/sdcard.lua
[start_lua_script] : Script output: Post installed script called script end
[start_lua_script] : Calling LUA /tmp/test.lua
Installing image core-image-base-skyboard-evb.ubifs into volume /dev/ubi1_1(rootfs)
[install_images] : Found installer for stream core-image-base-skyboard-evb.ubifs ubivol
U-Boot environment updated
[install_images] : Found installer for stream uboot-env uboot
Installing image uImage into volume /dev/ubi0_0(kernel)
```

Using with Yocto

- Meta-swupdate
- It generates a ramdisk suitable for u-boot (.uboot.gz)
- “dora” and “daisy” branches
- Footprint RAMDISK (gzipped) : 2.6 – 7 MB
  - Typical: ~4MB

Handler in LUA

```
require ("swupdate")

fpga_handler = function(image)
     print("Install FPGA Software ")

       for k,l in pairs(image) do
              print("image[" .. tostring(k) .. "] = " .. tostring(l) )
              swupdate.notify(swupdate.RECOVERY_STATUS.RUN,0,
               "image[" .. tostring(k) .. "] = " .. tostring(l))
       end

       return 0
end

swupdate.register_handler("fpga",fpga_handler)
```

swupdate todo list

- Create a community around the project
- Security: add support for signed images !
- Low resources: support for full streamable image
- New handlers

Links

- Swupdate sources at https://github.com/sbabic/swupdate
- Documentation at http://sbabic.github.io/swupdate
- Mailing list: swupdate@googlegroups.com
- http://www.denx.de/

Questions ...

- It's your turn now...