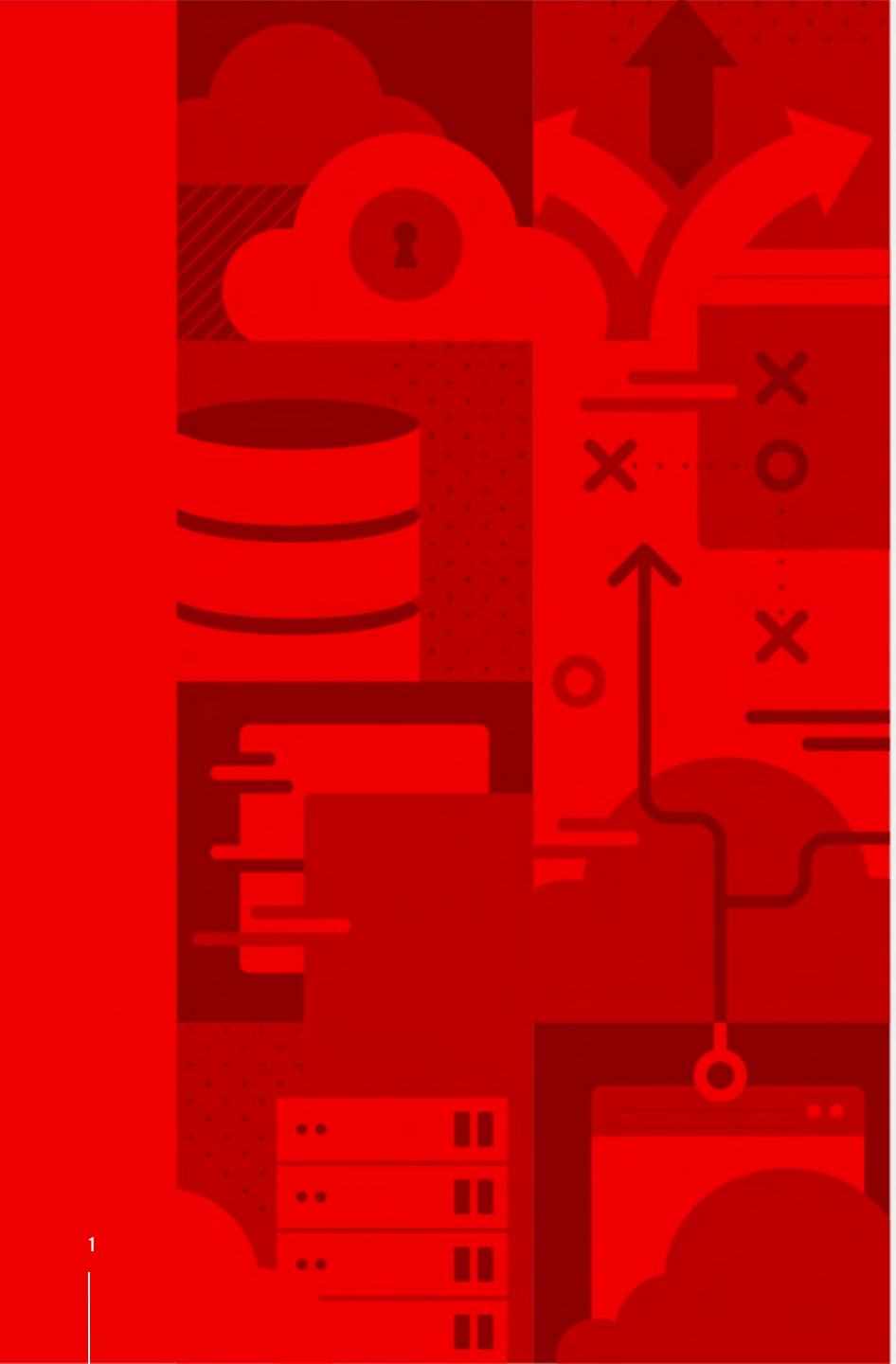


LVFS:
The next 50 million firmware updates

An overview of the ecosystem, and showing some of the cool new things we're trying to do.

Richard Hughes
Principal Engineer

Red Hat

Who am I?

Building Open Source

for **over 15 years.**

A *firmware troublemaker*

for over 6 years.

Red Hat

Users were not updating firmware

What hardware is installed?

Users don't typically know exactly what hardware they are using.

What updates are available

Users do not visit OEM websites to manually look for firmware updates.

Where do I get them from?

Many OEMs have insecure download links without any file checksums or signatures.

How to apply the update

Vendor tools often required Microsoft Windows, or unsupported Linux versions.

LVFS and fwupd work together

LVFS : Trusted Metadata Source

The hardware vendor uploads firmware to the LVFS where it is verified and signed. Users then download a shared metadata catalogue from a central server.

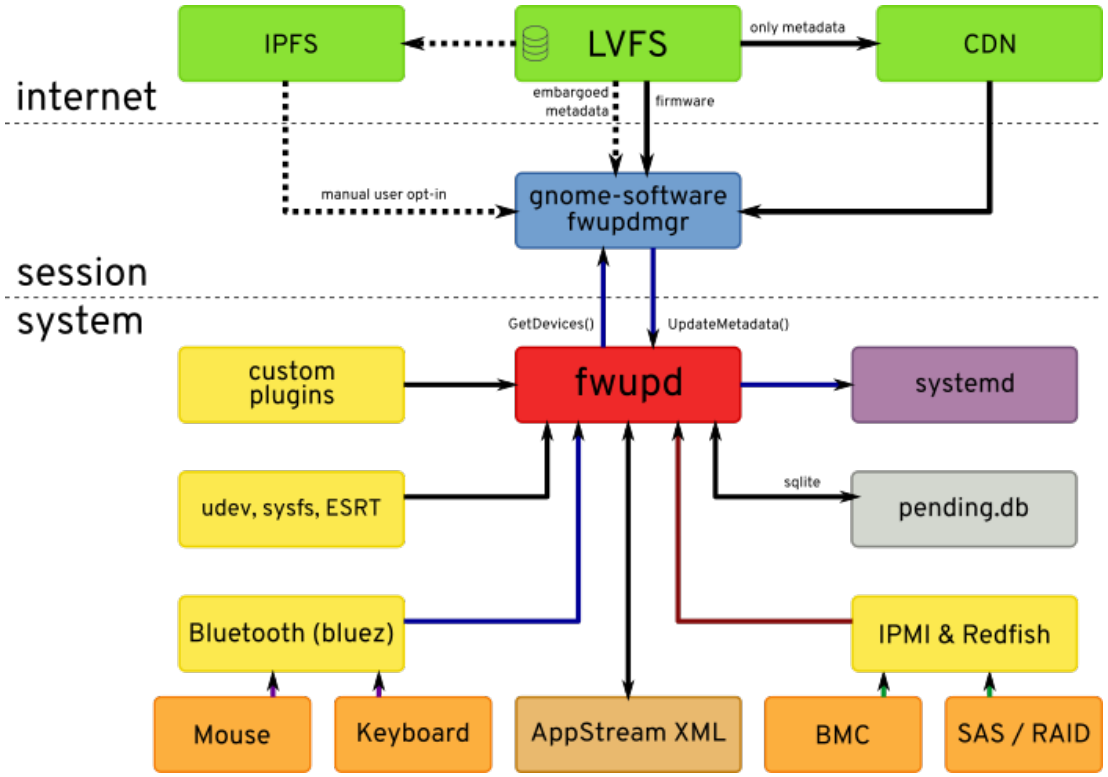
fwupd : Mechanism

The open source fwupd project deploys the update onto the Linux client machine. Over 32 update protocols are now supported and more are planned.

LVFS : Anonymous Reporting

After updating firmware, fwupd optionally sends success or failure information back to the LVFS to ensure updates are being deployed without problems

Architecture



D-Bus is used to interact with fwupd

- Desktop neutral interface with binding for every language

Updates not applied without an agent

- Full integration with GNOME and KDE, with CLI interface
- Work on Cockpit and CoreOS integration for server

Scalable architecture designed to continue to grow

- LVFS hosted on AWS

Designed to the decentralised

- Can easily be mirrored on a private network and puts privacy first by matching hardware client side

The fwupd daemon will not run non-free code

Efficiency

Plugins enumerate and flash hardware, abstracting functionality as reusable modules. Typically ~1000 lines of code and easy to write and audit.

Maintenance

Hardware vendors do not need to build update binaries for many different Linux distributions.

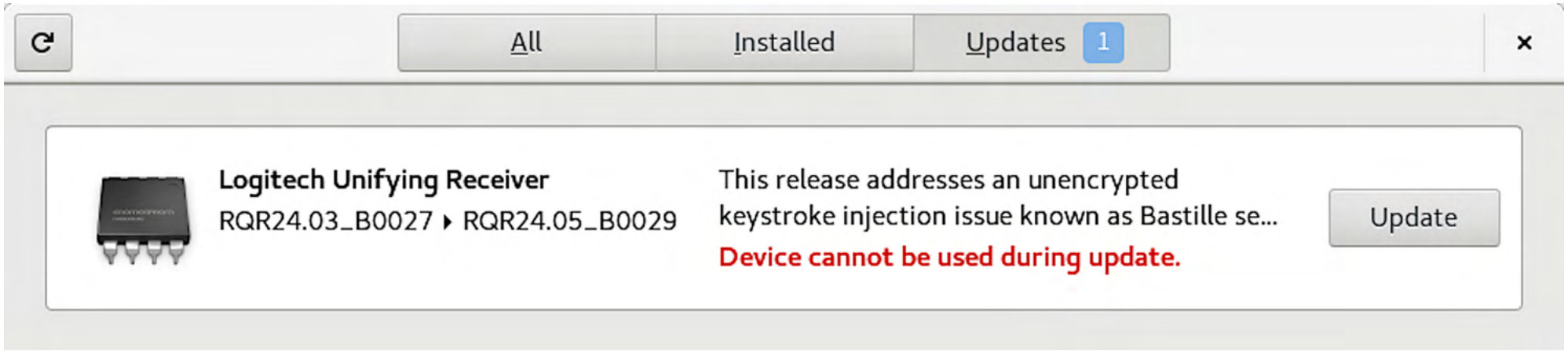
Update protocol

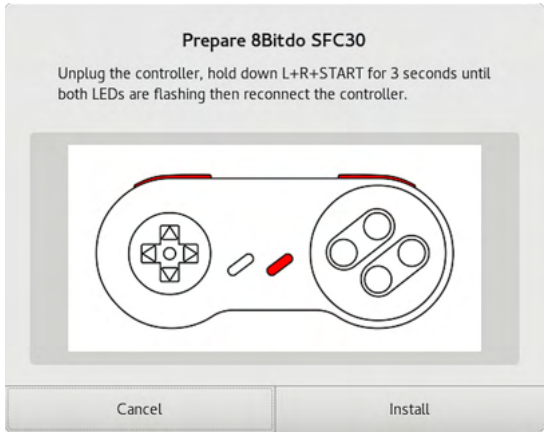
Not be part of the device security protection. Use strong cryptography to prevent modification.

Compliance

Various customers are unable to run non-free static binaries from hardware vendors.

We have to make this beautiful





99% of updates are applied using the GUI tools

- GNOME Software supporting fwupd since RHEL 7
- Release notes have to be understandable
- Firmware updates treated as 1st class citizen

Red Hat

The LVFS grows every year, as new vendors join and as more firmware is uploaded

Companies and agencies are free to mirror the LVFS for privacy or scalability reasons and so we don't actually know the real number of downloads.

52.1M

Firmware files supplied to end users

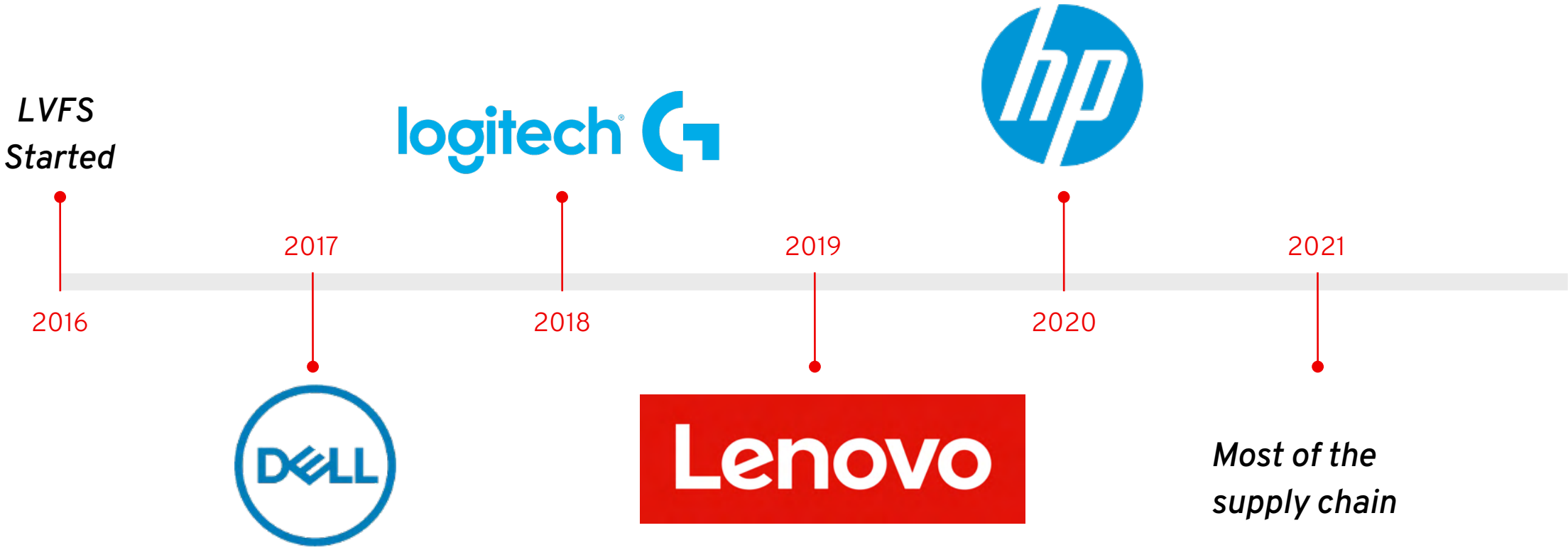
Since the LVFS started the official server has supplied millions of firmware updates for over 200 different devices.

126K

Success reports from end users

Over 99% of firmware was deployed correctly, with 1% of "known failures" identified using a built-in rule engine.

Over 120 OEMs, ODMs and IHVs use the LVFS



Red Hat

It’s actually hard to not support the LVFS.

OEMs are free to choose whatever criteria they like for hardware suppliers, and they are choosing these rules for various business reasons.

Lenovo

All suppliers for Lenovo ThinkPad, ThinkStation and ThinkCentre have to have working fwupd plugins and be able to upload to the LVFS. Failure to meet either criteria causes the “preferred vendor” status to be lost.

Dell

All approved ODMs and ISVs being used by Dell must have firmware that can be updated using fwupd and have updates available on the LVFS.

Google

Firmware must be updatable using fwupd to get the “Designed for Chrome” compliance sticker. Google are shipping parts of fwupd in nearly every Chromebook now sold.

Red Hat

Server vendors are racing to get firmware on the LVFS

Lenovo ThinkSystem

The SR630v2 system has passed validation and the first firmware will be available on the LVFS 2022Q3 which puts Lenovo on several preferred supplier lists. More SKUs are expected by 2023.

Dell Server

One of the biggest customers has told Dell to “**Get on the LVFS**”. Dell is now certifying the Redfish plugin on 15th generation PowerEdge servers.

What the vendors are saying...

> LVFS is strategically important for Dell to be able to provide secure firmware updates in a standards-compliant way.

Mario Limonciello
Sr. Principal Software Engineer, Dell

> Standardizing on LVFS has helped Lenovo seamlessly distribute our firmware updates to our customers

Rob Herman
Executive Director, Lenovo

There is no cost to use the LVFS or to contribute to fwupd

The Linux Vendor Firmware Service is sponsored by the Linux Foundation and most development work is provided by Red Hat.

Independent consulting companies provide technical help and training.

2019

LVFS analyses uploaded firmware

Firmware is checked and scanned for known issues. Headers and footers are checked against the provided metadata values.

2020

LVFS helps secure the ecosystem

UEFI firmware is decompressed and analysed. Researchers can scan for vulnerabilities using Yara. Notification of microcode downgrade.

2021

LVFS launches HSI specification

The Host Security ID indicates the level of platform security. Results are uploaded to LVFS for analysis. HSI will be used for purchasing decisions.

2022

LVFS launches fwupd friendly firmware specification

We want to make it easy for ODMs and OEMs to choose components that already have fwupd plugin support.

Firmware Analysis : UpdateCapsule

UEFI Capsule 2019-07-02 01:35:14

Check the UEFI capsule header and file structure

```
GUID: 5ffdbc0d-f340-441c-a803-8439c8c0ae10
HeaderSize: 0x1000
Flags: 0x70000
CapsuleImageSize: 0xab6dda
```

Retry

Firmware Analysis : Raising the Bar

Blocklist

Use a simple blocklist to check firmware for problems

Enabled

Values

DO NOT TRUST::IBV example certificate being used
DO NOT SHIP::IBV example certificate being used
To Be Defined By O.E.M::IBV example DMI data being used
c97445f45cdef9f0d3e05e1e585fc297235b82b5be8ff3efca67c59852018192::Contains the Dual EC backdoor for the NSA
Do not trust::IBV example certificate being used

Modify

Using FwHunt we remind vendors about the embargo

```
hex_strings:
  - 56e8.........593c01....80be....000000
    # 56                                    push    esi
    # E8 .. .. .. ..                       call    x_BiosSsaEnabled
    # 59                                    pop     ecx
    # 3C 01                                 cmp     al, 1
    # .. ..                                 jnz     short loc_FFDE86FD
    # 80 BE .. .. 00 00 00                  cmp     byte ptr [esi+81h], 0
    # .. ..                                 jz      short loc_FFDE86FD
  - 6a006a0268be00000056e8
    # 6A 00                                 push    0
    # 6A 02                                 push    2
    # 68 BE 00 00 00                        push    0BEh
    # 56                                    push    esi
    # E8 .. .. .. ..                       call    SsaApi
```

Making firmware platform security simple

Assigning weights

We assign weights to various protections, e.g. `BIOSWE` (HSI:1) more important than TME (HSI:3)

Allow overrides

Security protections are allowed to obsolete other failures, for example BiosGuard obsoletes PRx register configuration

Secure by default

HSI forces vendors to turn on security by default out of the box as users do not manually run tests.

Test Specificacy

HSI tests can be silicon vendor or platform specific as requried. Higher HSI levels must pass **all** lower HSI tests.

Host Security ID provides clear and unambigious validation of firmware platform security

The HSI tests are performed at runtime during every system boot with no extra tools or configuration required.

By the OEM

The OEM can use the HSI tests to verify the claims of the hardware vendor or the independant silicon vendor.

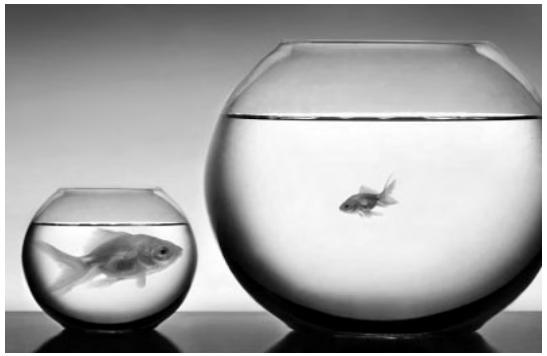
By the corporate security team

The company or government security team can use the HSI specification to verify all hardware is running with the apropriate HSI value for the appropriate threat level.

By the user

The end customer can test the hardware in the field to test the OEM claims, and also check for firmware regressions after each upgrade.

Publishing the results make vendors aim higher

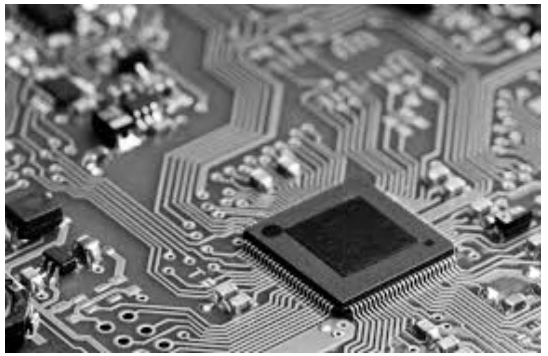


Public Scoreboard

A per-vendor and per-model public scoreboard allows consumers to check hardware before purchase and also compare OEMs and modes.

Purchase Requirements

A minimum HSI level should be part of purchasing or bidding requirements for large contracts.



OEMs choose secure hardware

Vendors should be chosing hardware based on price and how it affects the HSI value.

fwupd friendly firmware

Synaptics

- **FS7600** — MIS Touch Fingerprint Reader
- **VMM6xxx** — DisplayPort Alt Mode 2.0 protocol converter
- **VMM7100** — DisplayPort USB Type-C-to-HDMI 2.1 protocol converter

Wacom

- **G14** — EMR based technology pen
- **G14T** — A panel with finger touch and AES technology pen version 1.0, 2.0, and 2.a
- **G14TL** — A panel with finger touch and AES technology pen version 1.0, 2.0, and 2.a
- **G14TS** — A panel with finger touch and AES technology pen version 1.0, 2.0, and 2.a

Using eSPI for verification

```
[hughsie@hughsie-work build (wip/hughsie/intel-spi %)]$ sudo ./src/fwupdtool --plugins intel_spi get-devices --show-all
Loading…                [***************************************]
WARNING: This package has not been validated, it may not work properly.
20EQS64N0C
│
└─CM236 Chipset LPC/eSPI Controller:
    │   Device ID:          71b31258b13a4b2793e529856a190f8fb02ad151
    │   Current version:    31
    │   Vendor:             Intel Corporation (PCI:0x8086)
    │   GUIDs:              2a27aec1-f32e-5b68-ab14-8c3cb563fdfe ← PCI\VEN_8086&DEV_A150&SUBSYS_17AA222E&REV_31
    │                       461c3a89-a297-581f-a30e-631e0d53d056 ← PCI\VEN_8086&DEV_A150&SUBSYS_17AA222E
    │                       d3f0e5be-48ac-5e84-b953-25e291652927 ← PCI\VEN_8086&DEV_A150&REV_31
    │                       c60968d9-9a1d-5ae3-97b5-c0d3fadb56ae ← PCI\VEN_8086&DEV_A150
    │                       d1e04049-182d-5523-a947-b458eece3a76 ← INTEL_SPI_CHIPSET\PCH100
    │   Device Flags:       • Internal device
    │                       • Cryptographic hash verification is available
    │
    ├─BIOS:
    │       Device ID:      ff7dbf2f6e354a5727c6ce1c466230f38bd26ff0
    │       Vendor:         Intel Corporation (PCI:0x8086)
    │       GUID:           6da44464-8748-5379-8c0c-204396ee49a7 ← IFD\BIOS
    │       Device Flags:   • Internal device
    │                       • Cryptographic hash verification is available
    │
    ├─Gigabit Ethernet:
    │       Device ID:      11188287b93230d58f85f059dfab93e1d59724bb
    │       Vendor:         Intel Corporation (PCI:0x8086)
    │       GUID:           2767029e-0944-520d-b835-25e35ed25740 ← IFD\GBE
    │       Device Flags:   • Internal device
    │                       • Cryptographic hash verification is available
    │
    └─Intel Management Engine:
            Device ID:      7ef8a531d2413174034556f12dff8aa3bb4a8c30
            Vendor:         Intel Corporation (PCI:0x8086)
            GUID:           486c866f-42ce-5b87-9309-9df7929e2dd9 ← IFD\ME
            Device Flags:   • Internal device
                            • Cryptographic hash verification is available
```

Red Hat

Import and export of complete IFD

```
<firmware gtype="FuIfdFirmware">
  <descriptor_map0>0x40003</descriptor_map0>
  <descriptor_map1>0x58100208</descriptor_map1>
  <descriptor_map2>0x310330</descriptor_map2>
  <components_rcd>0x325c00f5</components_rcd>
  <illegal_jedec>0x42</illegal_jedec>
  <firmware gtype="FuIfdBios">
    <id>bios</id>
    <idx>0x1</idx>
    <addr>0x1000</addr>
    <firmware gtype="FuEfiFirmwareVolume">
      <id>8c8ce578-8a3d-4f1c-9935-896185c32dd3</id>
      <alignment>0xB</alignment>
      <firmware gtype="FuEfiFirmwareFilesystem">
        <alignment>0x3</alignment>
        <firmware gtype="FuEfiFirmwareFile">
          <id>ced4eac6-49f3-4c12-a597-fc8c33447691</id>
          <type>0x0B</type>
          <firmware gtype="FuEfiFirmwareSection">
            <type>0x02</type>
            <id>ced4eac6-49f3-4c12-a597-fc8c33447691</id>
            <data>aGVsbG8gd29ybGQ=</data>
          </firmware>
          <firmware gtype="FuEfiFirmwareSection">
            <data>aGVsbG8gd29ybGQ=</data>
          </firmware>
        </firmware>
        <firmware gtype="FuEfiFirmwareFile">
          <id>ced4eac6-49f3-4c12-a597-fc8c33447691</id>
          <data>aGVsbG8gd29ybGQ=</data>
        </firmware>
      </firmware>
    </firmware>
    <firmware gtype="FuEfiFirmwareVolume">
      <id>fff12b8d-7696-4c8b-a985-2747075b4f50</id>
      <alignment>0xB</alignment>
      <data>aGVsbG8gd29ybGQ=</data>
    </firmware>
  </firmware>
  <firmware gtype="FuIfdImage">
    <id>me</id>
    <idx>0x2</idx>
    <addr>0x2000</addr>
    <data>V29ybGQh</data>
  </firmware>
</firmware>
```

```
$ ./src/fwupdtool firmware-parse lenovo-p50.bin ifd
<firmware gtype="FuEfiFirmwareSection">
  <offset>0xc</offset>
  <size>0x1a0004</size>
  <data size="0x1a0000">
  </data>
  <type>0x17</type>
  <type_name>volume-image</type_name>
  <firmware gtype="FuEfiFirmwareVolume">
    <id>8c8ce578-8a3d-4f1c-9935-896185c32dd3</id>
    <size>0x1a0000</size>
    <alignment>0x4</alignment>
    <attrs>0xfeff</attrs>
    <name>Volume:Ffs2</name>
    <firmware gtype="FuEfiFirmwareFilesystem">
      <offset>0x187830</offset>
      <alignment>0x4</alignment>
      <firmware gtype="FuEfiFirmwareFile">
        <id>ffffffff-ffff-ffff-ffff-ffffffffffff</id>
        <size>0x30</size>
        <data size="0x14">&amp;quot;.,....J.F</data>
        <alignment>0x3</alignment>
        <type>0xf0</type>
        <type_name>ffs-pad</type_name>
      </firmware>
      <firmware gtype="FuEfiFirmwareFile">
        <id>cdc11ae9-01e7-42cb-88eb-fdffd8819893</id>
        <size>0x2d08</size>
        <data size="0x2cea">^</data>
        <alignment>0x3</alignment>
        <type>0xa</type>
        <type_name>mm</type_name>
      </firmware>
```


Red Hat

Updating with fwupd as a proxy to a BMC

Talking to the BMC using Redfish or legacy IPMI

- As well as local devices like ATA, NVMe, DFU etc.
- Leverage fwupd as part of host OS insights
- fwupd acts as a “proxy” using the internal USB NIC
- Authentication to BMC via EFI or IMPI KCS.

Red Hat

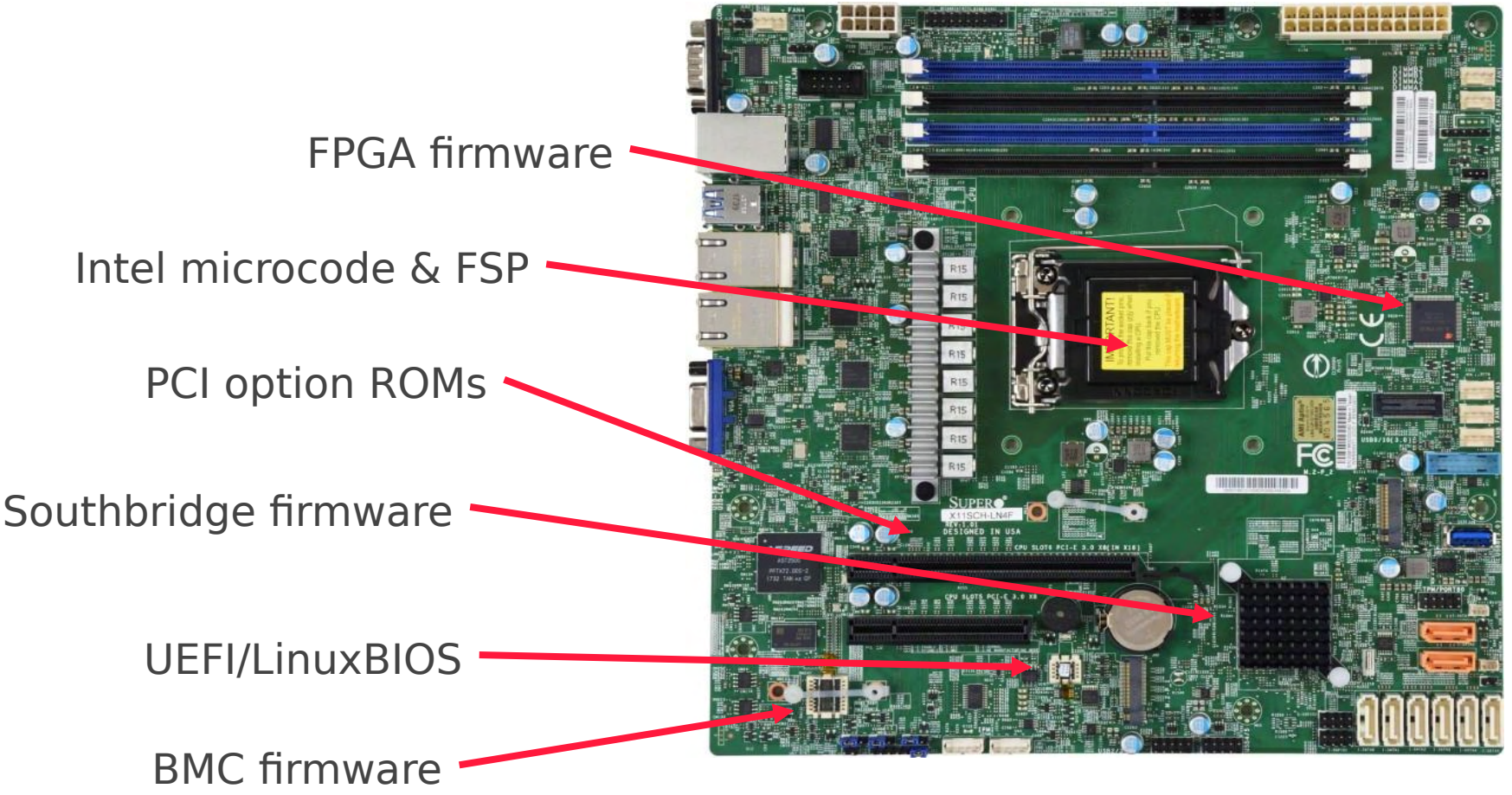
Updating with fwupd **running on** the BMC

```
root@evb-ast2500:~# fwupdmgr get-devices
WARNING: This package has not been validated, it may not work properly.
AST2500 EVB
?
??bmc:
?     Device ID:          4ab0e7e8286c726c4572bd7ee9e5ee7749a6221e
?     Summary:            Memory Technology Device
?     Vendor:             DMI:ASPEED
?     GUID:               484fcb1d-2f5b-527f-8b12-41536294f524 ? MTD\NAME_bmc
?     Device Flags:       ? Internal device
?                         ? Updatable
?                         ? Needs a reboot after installation
?
??kernel:
?     Device ID:          9cc4118aa0d6706e10ffdf87eeccec184817b9e1
?     Summary:            Memory Technology Device
?     Vendor:             DMI:ASPEED
?     GUID:               e9c923c5-9809-5276-ab14-e1c1cd697f7d ? MTD\NAME_kernel
?     Device Flags:       ? Internal device
?                         ? Updatable
?                         ? Needs a reboot after installation
?
```

U.S. DoC says we have to care about SBoM

Red Hat

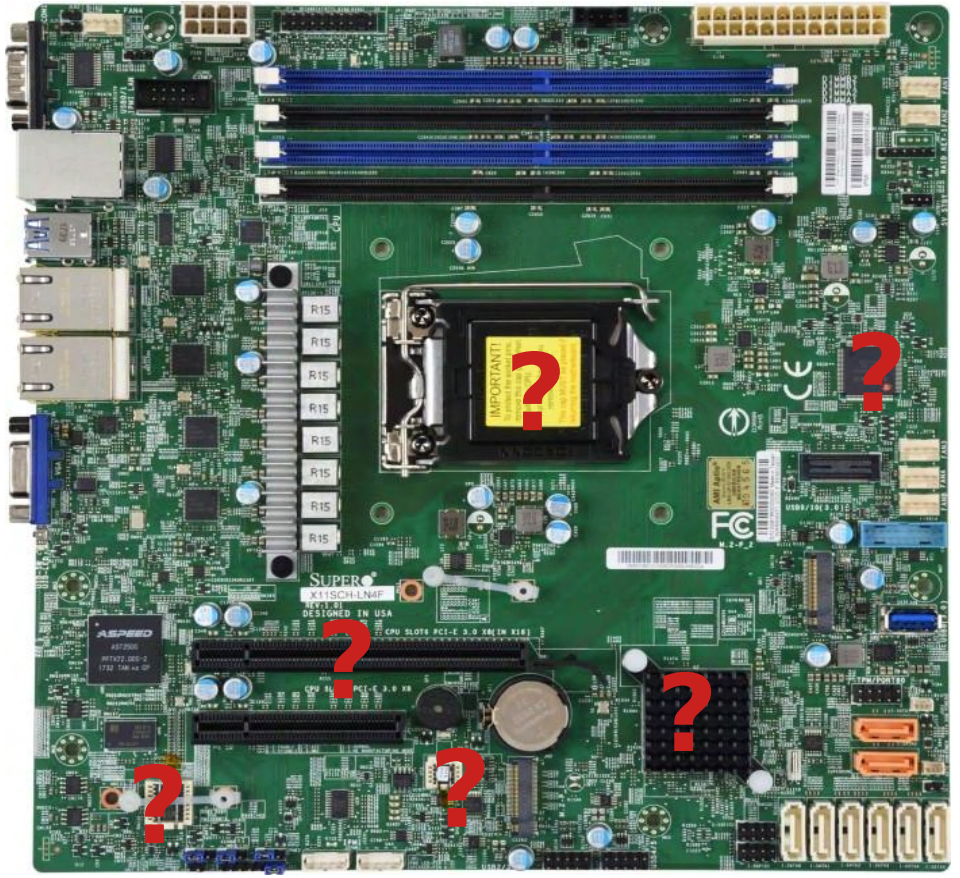
We have more than one blob?



Red Hat

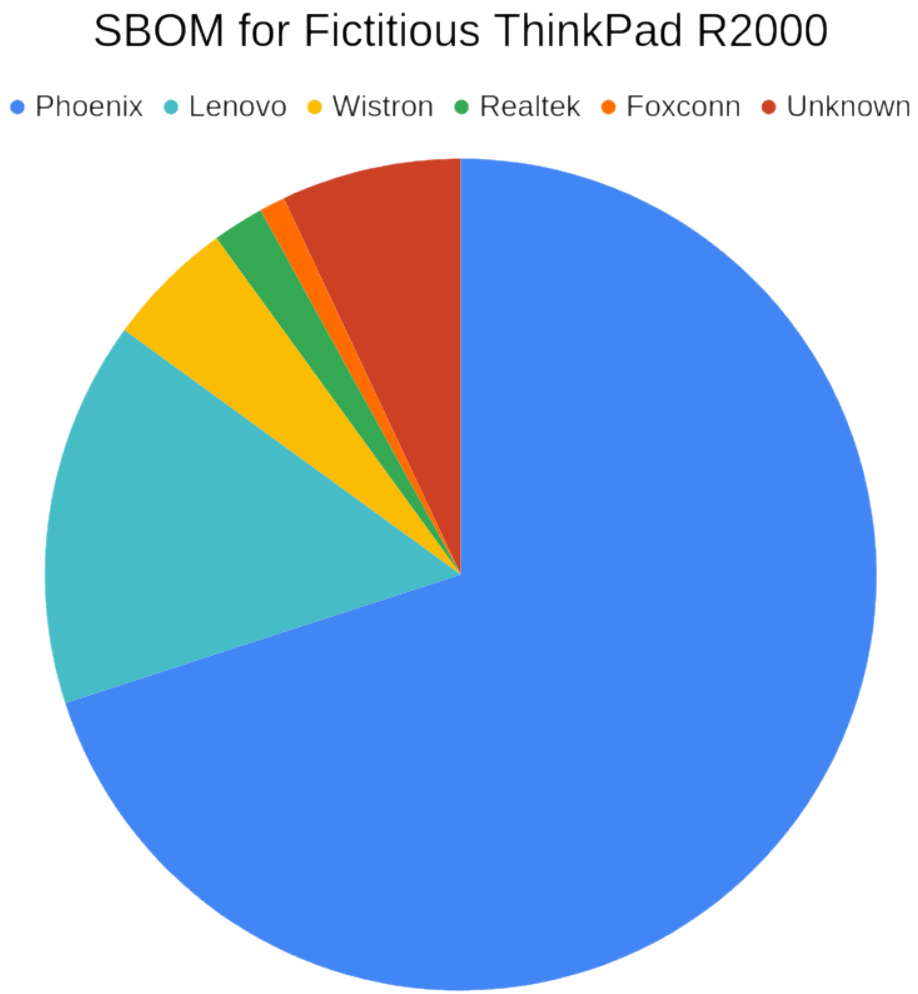
Who supplied each firmware?

- Who built them?
- When did they build it?
- What OpenSSL did they use?
- What is the licence?
- What is the version?



Red Hat

SBOM via uSWID



Embed the SBOM data into a SBOM COFF section

- Means it doesn't get stripped
- Which allows the LVFS to extract from FVs

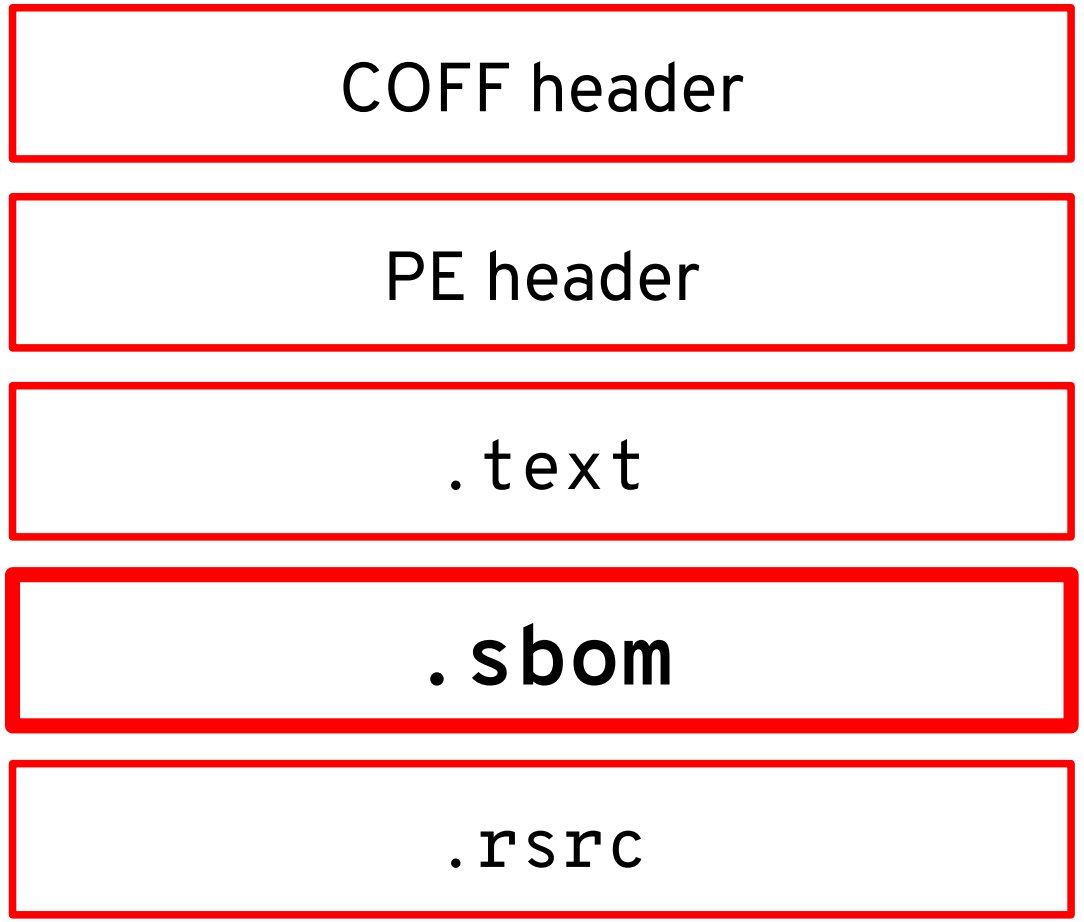
Allow entity "patching" using a simple .ini format

```
[uSWID-Entity:Distributor]
name = OEM Vendor
```

https://github.com/hughsie/python-uswid

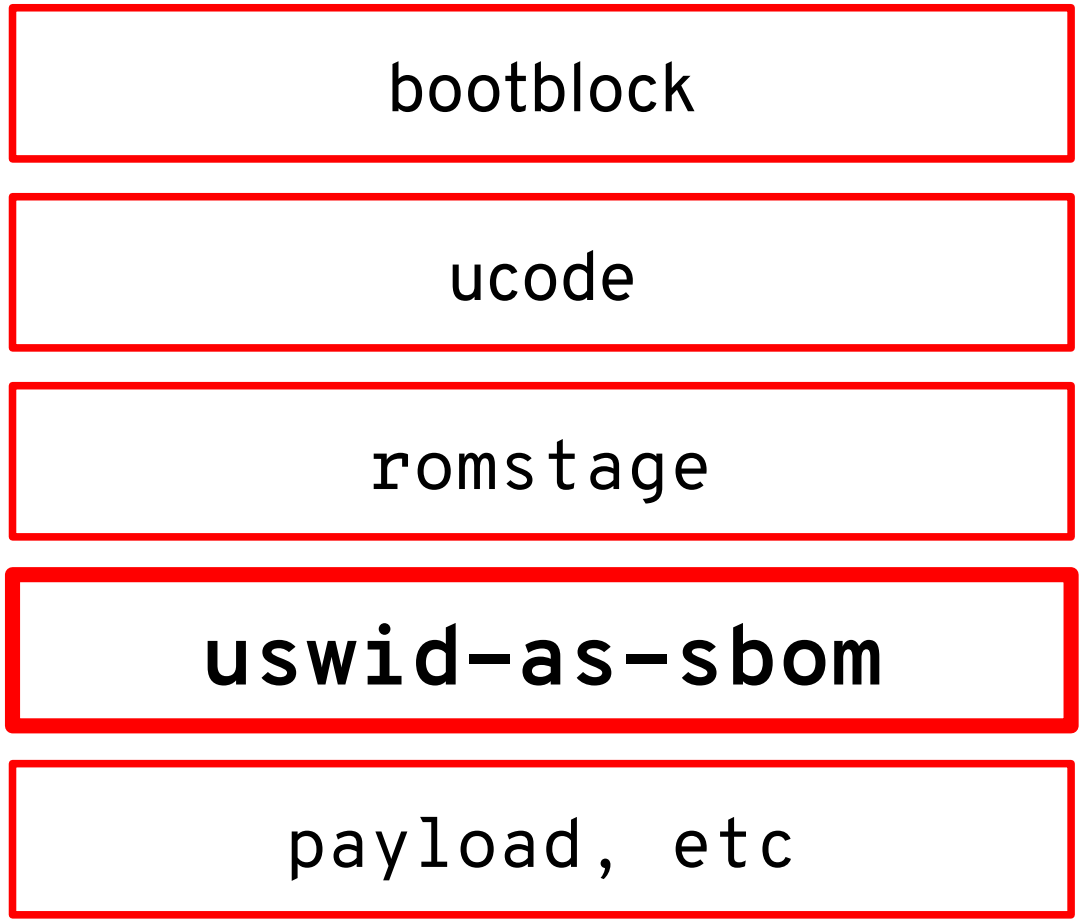
Red Hat

A New COFF Section for EDKish



Red Hat

A New CBFS section for coreboot



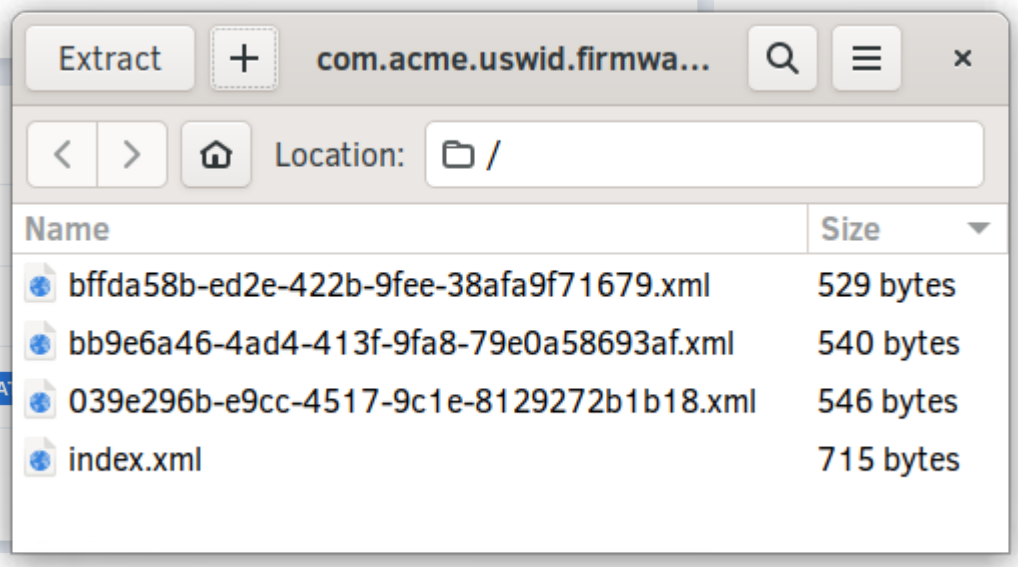
LVFS end-to-end with SWID export

coreboot — vf490ec2adc210907e3f27599c2c6fed2f1505e63 `a9032c9d-2aaa-5a25-a0e6-6d865b24e6d2`

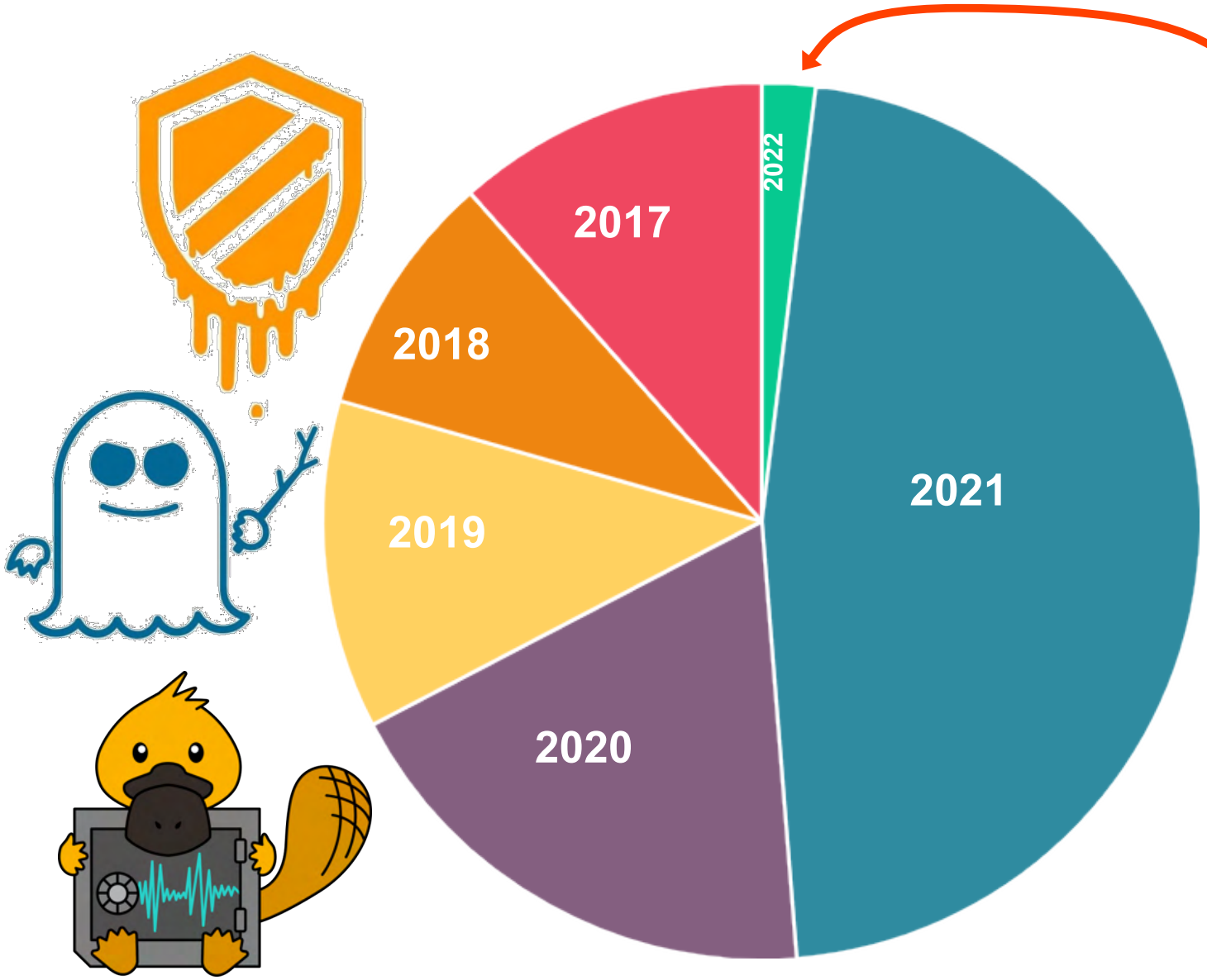
Summary	coreboot is a project to develop open source boot firmware for various architectures
Product	coreboot
Colloquial Version	63c440f4e9a2466dd4a6f8c750621341a2c5ec79
Entity	9elements TAG_CREATOR SOFTWARE_CREATOR
Generator	uSWID

Intel-Microcode — v2021-04-28

Summary	Micrcode Updates for Intel Processors
Product	Intel-Microcode
Entity	9elements TAG_CREA
Generator	uSWID

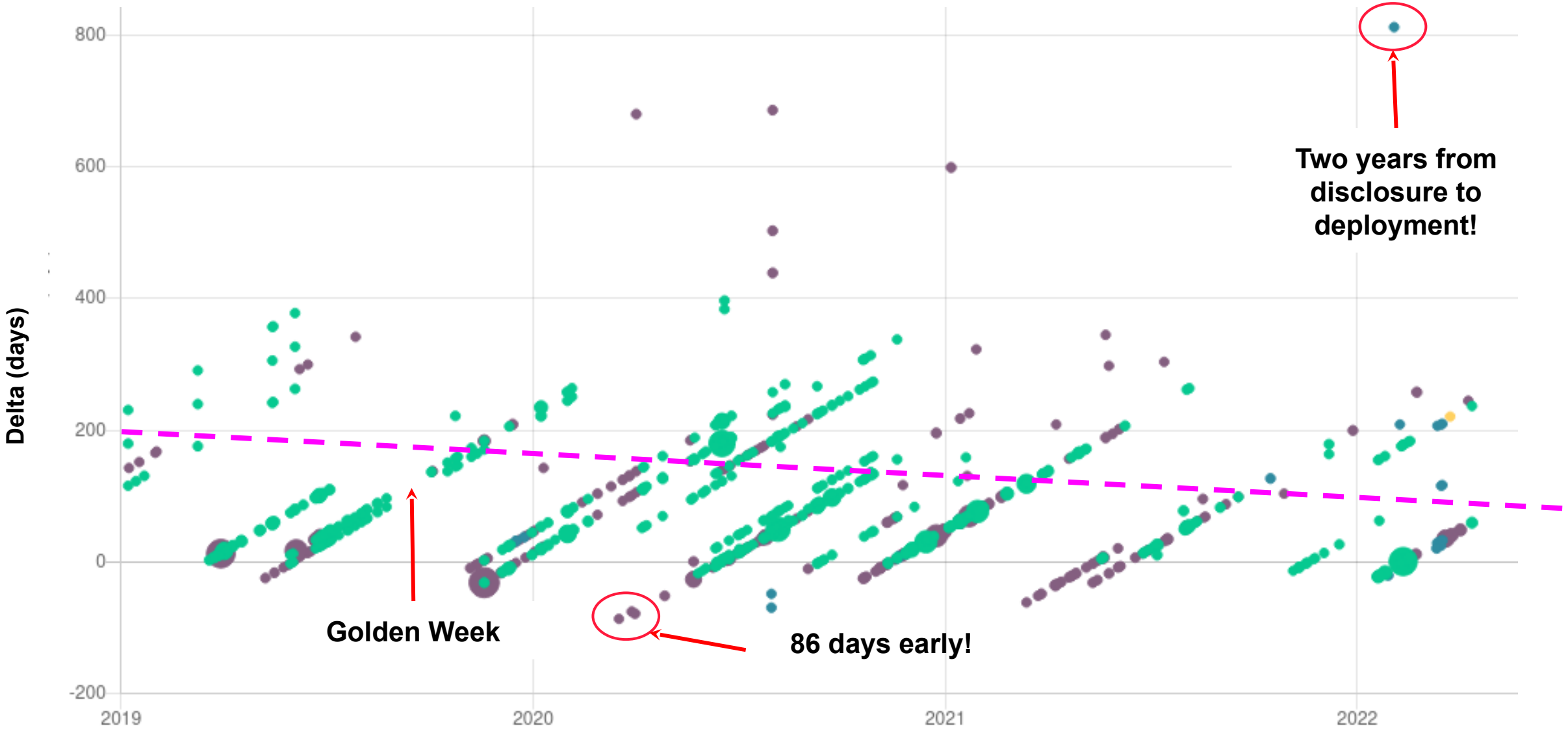


The newest versions of Intel Microcode



CVE-2022-21151
Processor Speculative Cross Store Bypass Advisory

Vendors take a long time to roll out fixes



Call to action

https://fwupd.org/
https://fwupd.github.io/
https://github.com/fwupd/fwupd
https://gitlab.com/fwupd/lvfs-website
https://lists.linuxfoundation.org/pipermail/lvfs-general/

REALLY HARD PROBLEMS

- Attestation of the firmware on all kinds of devices.
- Speeding up distribution of security fixes, and to avoid breaking embargos.
- Making vendors care about the LVFS when making inexpensive user devices.