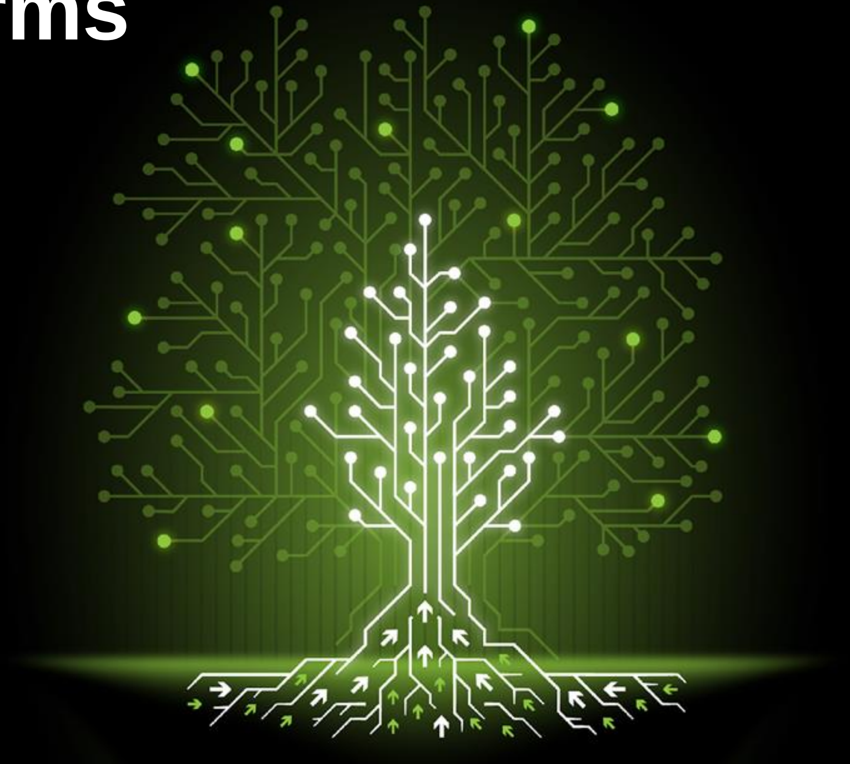
Universal Payload for Optimized Firmware Handoff in Server Platforms

Lean Sheng Tan, Lead, 9elements
Vincent Zimmer, Senior Principal Engineer, Intel



Scaling Innovation Through Collaboration

OCP GLOBAL SUMMIT

OCTOBER 17-19, 2023
SAN JOSE, CA

Open System Firmware (OSF)

OPEN SYS FW

Universal Payload for Optimized Firmware Handoff in Server Platforms

Lean Sheng Tan, Lead, 9elements
Vincent Zimmer, Senior Principal Engineer, Intel

OPEN PLATINUM™

The paradox of choice with OCP OSF

OPEN SYSTEM FIRMWARE

- Challenges of firmware development
- Growing customer demand with various open source platform code solutions and boot environments/payloads
- Closely coupled platform firmware design & code for some solutions

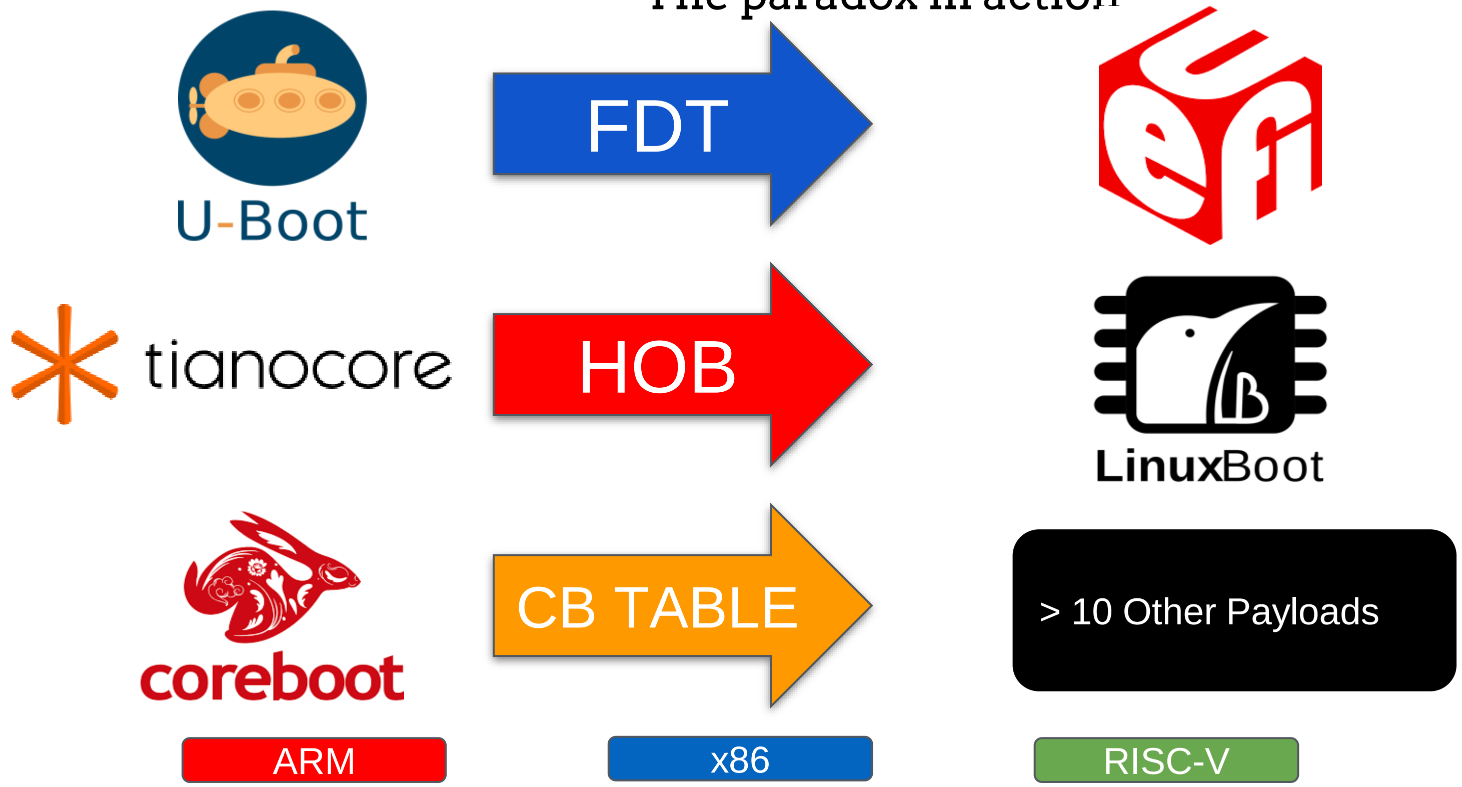
- Higher effort/cost + low reuse + longer Time-to-Market
- Porting effort from platform to platform
- Re-development effort from different bootloaders

How to address the paradox?

Overview

OPEN SYSTEM FIRMWARE

The paradox in action



Make firmware ~~great~~ SIMPLE again

U-Boot

tianocore

coreboot

UPL

LinuxBoot

> 10 Other Payloads

ARM

x86

RISC-V

Universal Payload = the "USB" of OSF world

Solutions?

OPEN SYSTEM FIRMWARE

- Aspects of a payload - how to encapsulate and how to invoke
- On the latter, what information to pass in to decouple platform init from the payload phase
- Options for information/data to pass in evaluated include:
- Use coreboot tables
- Use HOB
- Protobufs, CBOR https://www.youtube.com/@universalscalablefirmwarec534/videos
- Invent a new mechanism that
  overcomes the limitations
  of HOB & coreboot tables
- …

Problem statement on data passing

OPEN SYSTEM FIRMWARE

coreboot tables

- Very old (LinuxBIOS): LB_TAG_xxx
- Payload handoff of data that cannot be probed from the hardware
- GPL v2

UEFI HOB

- Some HOB structs contain fields which are specific for Edk2 DXE
- 64K limit - 2 byte size entry
- 16 byte GUID instead of tag number

Not self describing!

Expectations of a Universal Payload Handoff Design

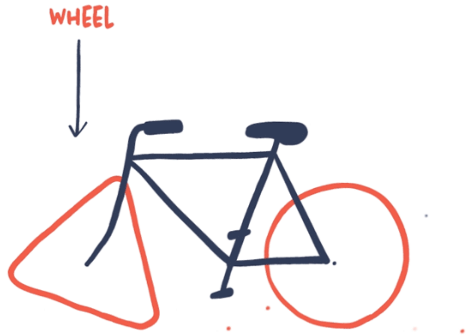
1. **self-describing**
2. **Not bounded by any single language**
3. example of data:
   - memory map
   - buildtime information like tooling, version,
   - pointer to memory console
   - uart to use for debug
   - framebuffer
4. Smooth transition/ **lightweight** introduction
   - edkII, coreboot, U-Boot, Linuxboot, SeaBIOS, etc

Broaden the scope & !reinvent the wheel

OPEN SYSTEM FIRMWARE

- This is not just limited to payload
  - Other binary interfaces (e.g. OpenSBI, FSP)
- This is not a new issue and has already been solved

Whats the best ‘wheel’ for this firmware usage?

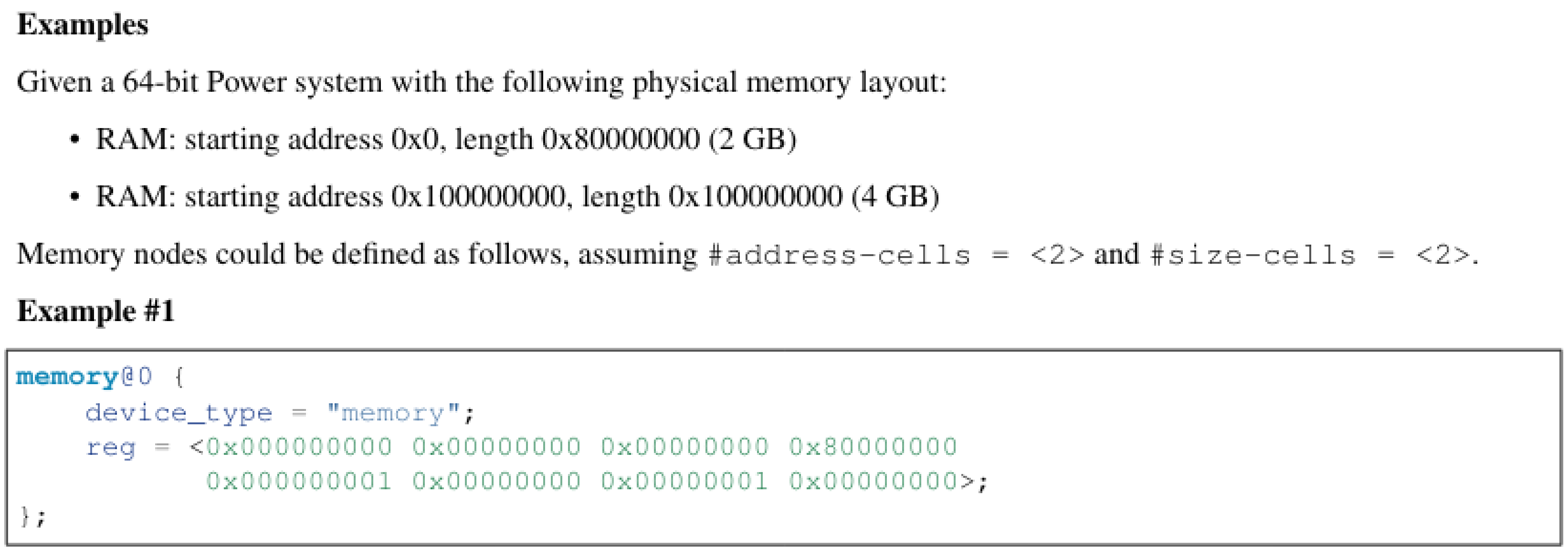


Flattened devicetree (FDT)

OPEN SYSTEM FIRMWARE

- devicetree.org
- Library exists in EDKII, coreboot, U-Boot
- Describe the hardware and pass control
- Libraries available in a lot of languages
- Binary & plain text format
- Rich usage across embedded firmware, RTOS, OS
- Live binding document

Flattened devicetree (FDT) example



Examples

Given a 64-bit Power system with the following physical memory layout:

- RAM: starting address 0x0, length 0x80000000 (2 GB)
- RAM: starting address 0x100000000, length 0x100000000 (4 GB)

Memory nodes could be defined as follows, assuming `#address-cells = <2>` and `#size-cells = <2>`.

Example #1

```
memory@0 {
    device_type = "memory";
    reg = <0x000000000 0x00000000 0x00000000 0x80000000
           0x000000001 0x00000000 0x00000001 0x00000000>;
};
```

POC

OPEN SYSTEM FIRMWARE

- Flow
  - Buildtime create of devicetree structure
  - Unpack devicetree
  - Apply fixup on nodes
  - Pack it again
  - Pass a pointer to the payload
  - Payload parses devicetree using existing libraries

- Fields definition for FDT: reuse Zhiguang's work

- Patches
  - https://github.com/ArthurHeymans/coreboot/tree/coreboot_fit_for_edk2
  - https://github.com/ArthurHeymans/edk2/tree/POC_FDT

Example UPL? EDKII payload

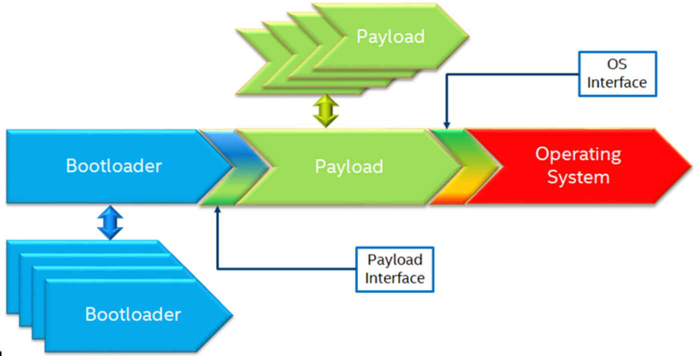
Tianocore: Support FDT library. · tianocore/edk2@10416bf (github.com)

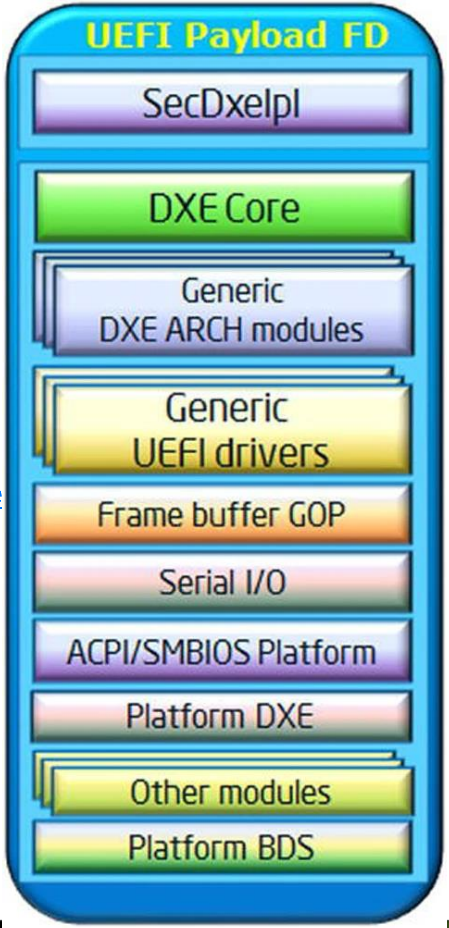
MdePkg: Support FDT library. · tianocore/edk2@5d58660 (github.com)

History for UefiPayloadPkg - tianocore/edk2 (github.com)

….

images from Intel FSP and UEFI Integration | SpringerLink &
An evolutionary approach to system firmware :: Open Source Firmware Conference 2021 :: pre





Future, replace UPDs in FSP?

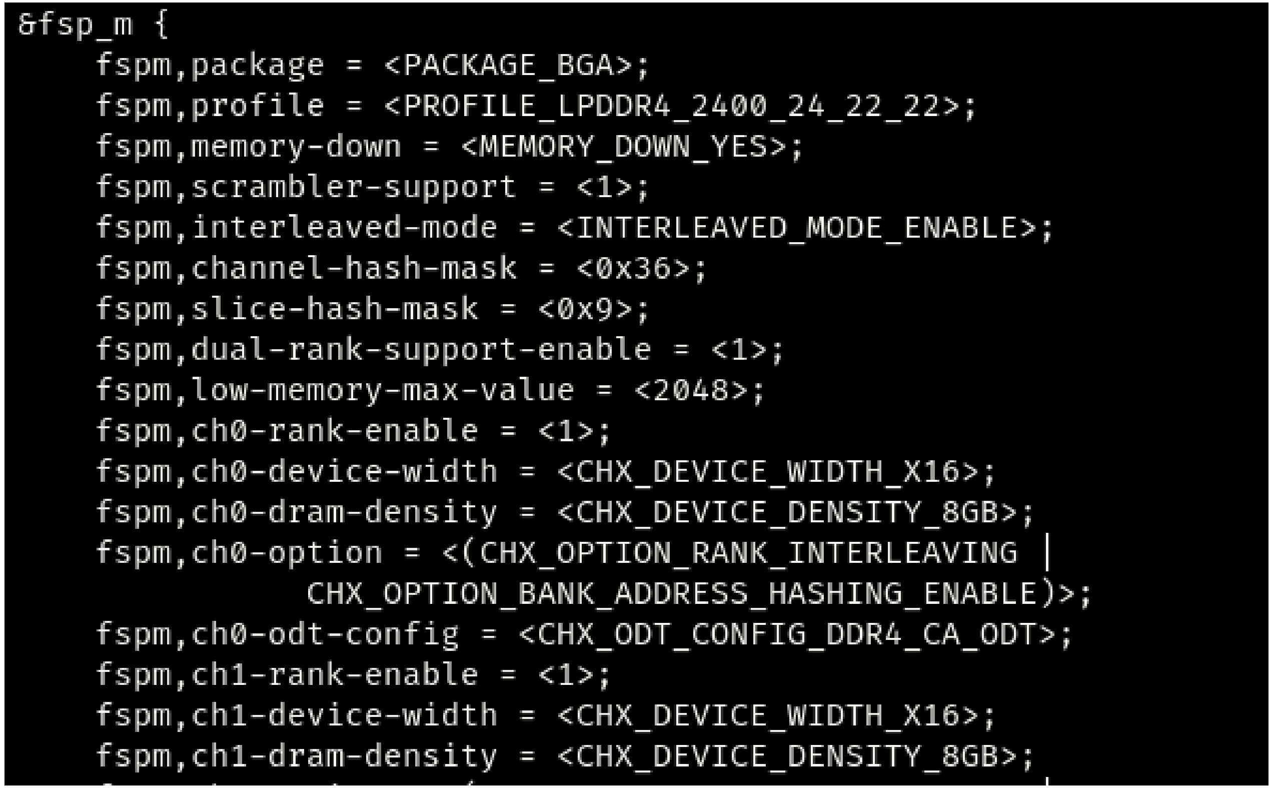
OPEN SYSTEM FIRMWARE

```
/** Offset 0x0B26 - Frequency Limit for Mixed 2DPC DDR5 1 Rank 8GB and 8GB
  Frequency Limit for 2DPC Mixed or non-POR Config. 0: Auto, otherwise a frequency
  in MT/s, default is 2000
**/
  UINT16                                    FreqLimitMixedConfig_1R1R_8GB;
```

- Not Flexible: Header mismatch?
- Not Scalable: 1 struct for all IPs/ configurations
- Not Reusable: each FSP has a new UPD

FSP usage with FDT

OPEN SYSTEM FIRMWARE



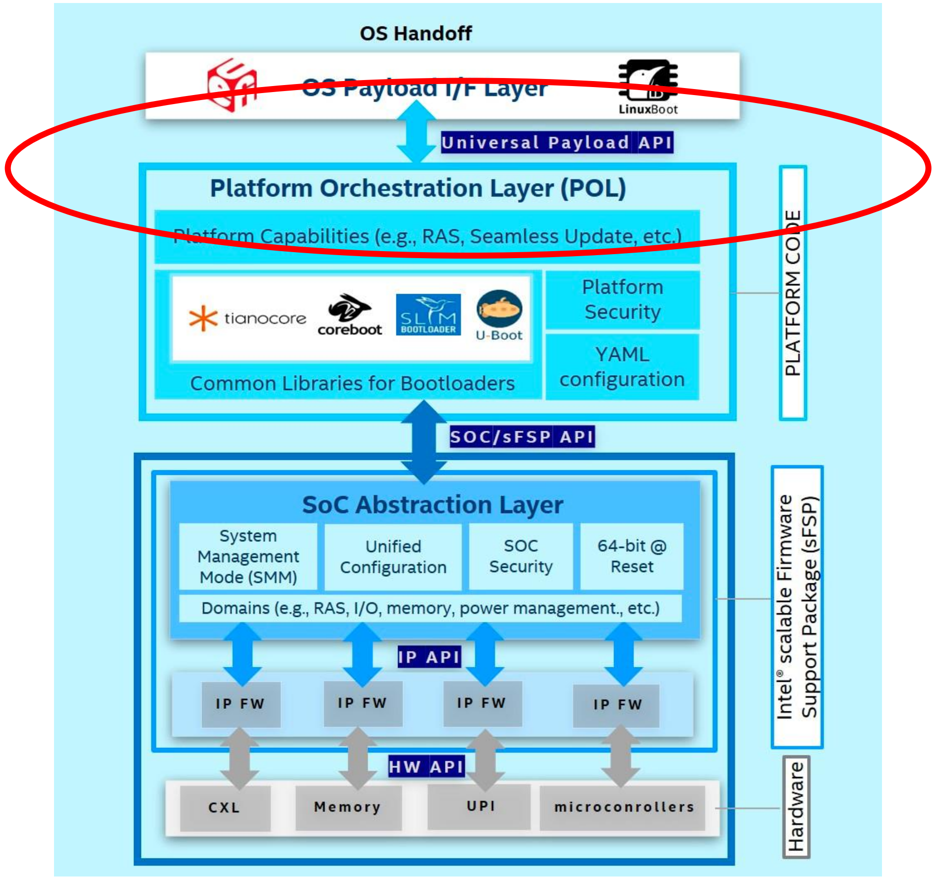
```
&fsp_m {
    fspm,package = <PACKAGE_BGA>;
    fspm,profile = <PROFILE_LPDDR4_2400_24_22_22>;
    fspm,memory-down = <MEMORY_DOWN_YES>;
    fspm,scrambler-support = <1>;
    fspm,interleaved-mode = <INTERLEAVED_MODE_ENABLE>;
    fspm,channel-hash-mask = <0x36>;
    fspm,slice-hash-mask = <0x9>;
    fspm,dual-rank-support-enable = <1>;
    fspm,low-memory-max-value = <2048>;
    fspm,ch0-rank-enable = <1>;
    fspm,ch0-device-width = <CHX_DEVICE_WIDTH_X16>;
    fspm,ch0-dram-density = <CHX_DEVICE_DENSITY_8GB>;
    fspm,ch0-option = <(CHX_OPTION_RANK_INTERLEAVING |
               CHX_OPTION_BANK_ADDRESS_HASHING_ENABLE)>;
    fspm,ch0-odt-config = <CHX_ODT_CONFIG_DDR4_CA_ODT>;
    fspm,ch1-rank-enable = <1>;
    fspm,ch1-device-width = <CHX_DEVICE_WIDTH_X16>;
    fspm,ch1-dram-density = <CHX_DEVICE_DENSITY_8GB>;
```

Future Aspect

OPEN SYSTEM FIRMWARE

one full firmware stack example with UPL in practice

- OS Handoff Interface
- SoC Universal API design for Silicon code



Timeline - where are we now

OPEN SYSTEM FIRMWARE

Time	Agenda
March '22	Initial Discussion with community members (EDK2, coreboot, oreboot, etc)
April' 22	Meetings with different communities & industries exploring CBOR, collect feedbacks
July '22	Exploring FDT as a better solution
Aug '22	Consensus reached among key contributors from each communities to proceed with FDT
Sep '22	OSFC presentation for broader feedback
Aug '23	Initial spec draft for feedback; planning for code patches land on edk2, coreboot, LinuxBoot
Nov '23	Finalize Handoff Spec 1.0

Conclusion

- Open development!

- cross communities collaboration
  - good willing people, gesture of trust
  - u-boot, edkII, coreboot
  - By community, for community

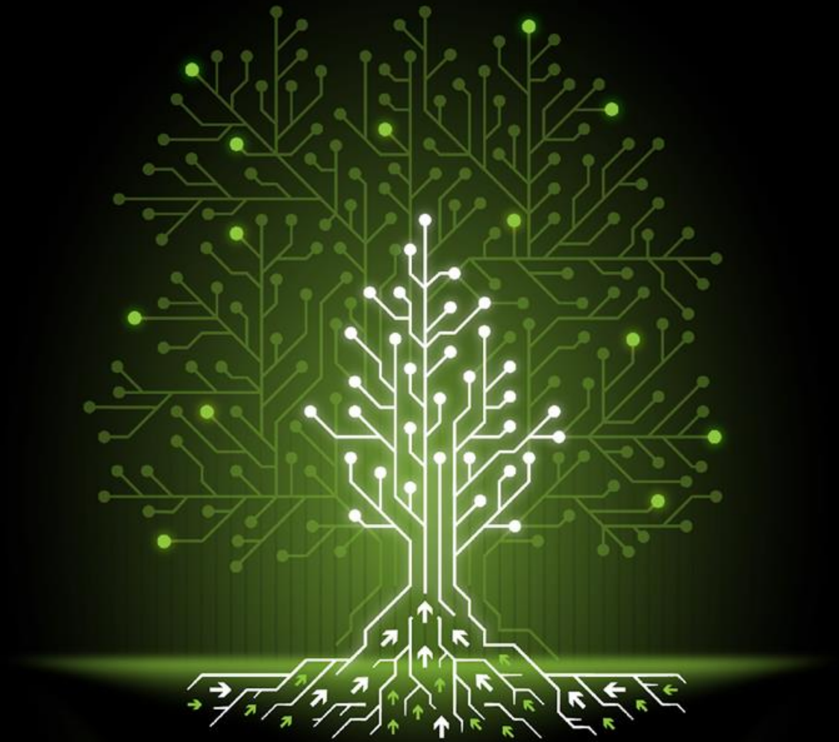
- light-weight industrial wide solution
  - small integration ‘fees’
  - reusable & adoptable
  - self-describing!

Call to Action

- Join upcoming OSF OCP call on this topic

- Check out and contribute to the specification
  - Latest specification link: https://github.com/UniversalPayload/spec

Open Discussion



Scaling Innovation Through Collaboration

OCP GLOBAL SUMMIT

OCTOBER 17-19, 2023
SAN JOSE, CA