



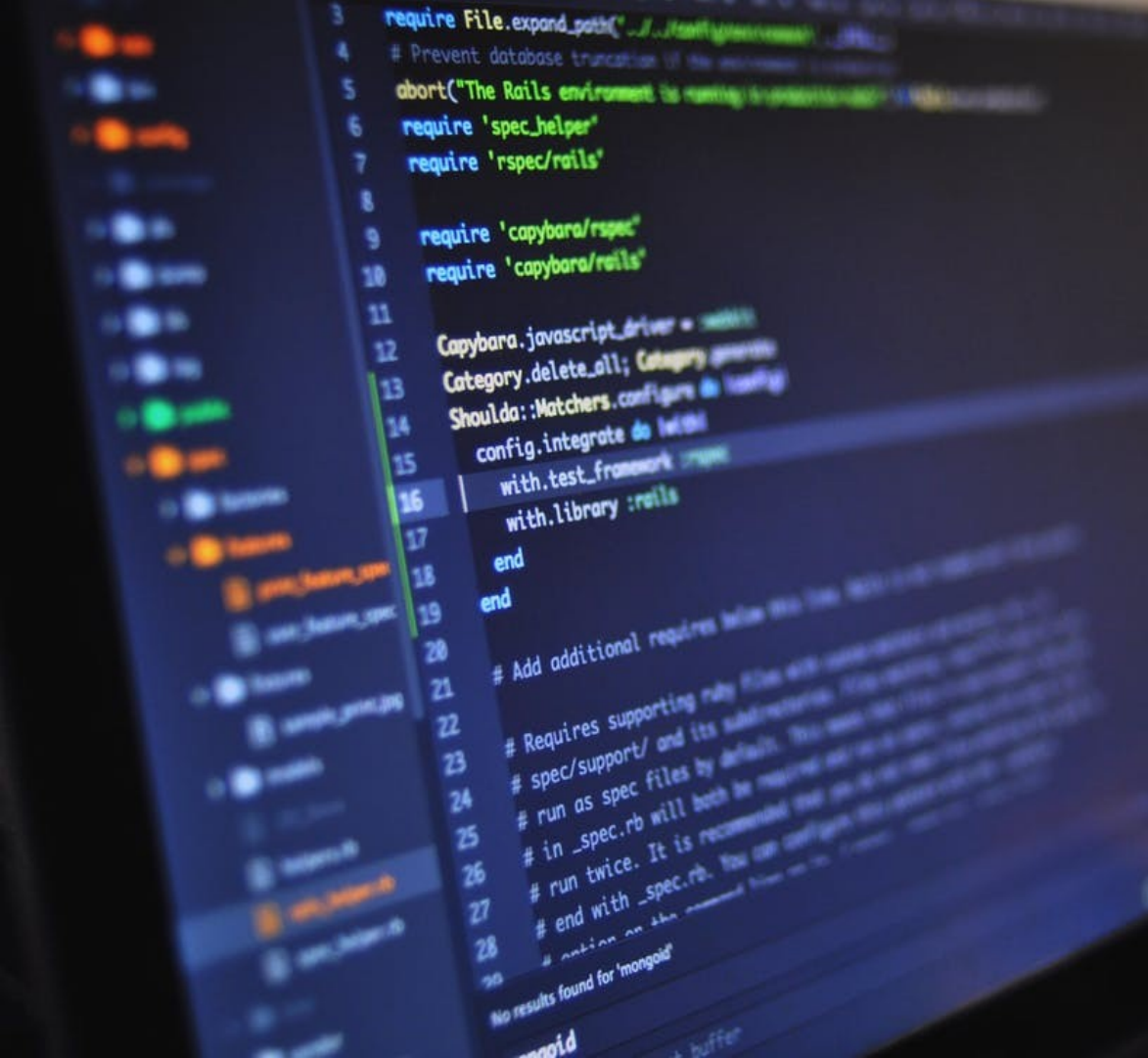
Case Study: Yocto / OpenEmbedded in All Scenarios OS

Andrei Gherzan <andrei.gherzan@huawei.com>

Davide Ricci <davide.ricci@huawei.com>

Stefan Schmidt <stefan.schmidt@huawei.com>

Yocto Project Summit, May 2021





Meet Philip.



headsets TVs thermostat

phones

watches

projector

lights valves

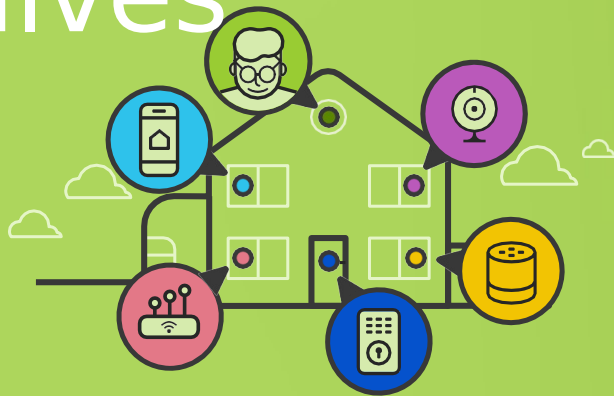
motion

doorbell

door lock

alarm

cameras











CONSUMERS

Complexity
Insecurity
Lack of privacy
Turned into products



DEVICE MAKERS/OEMs

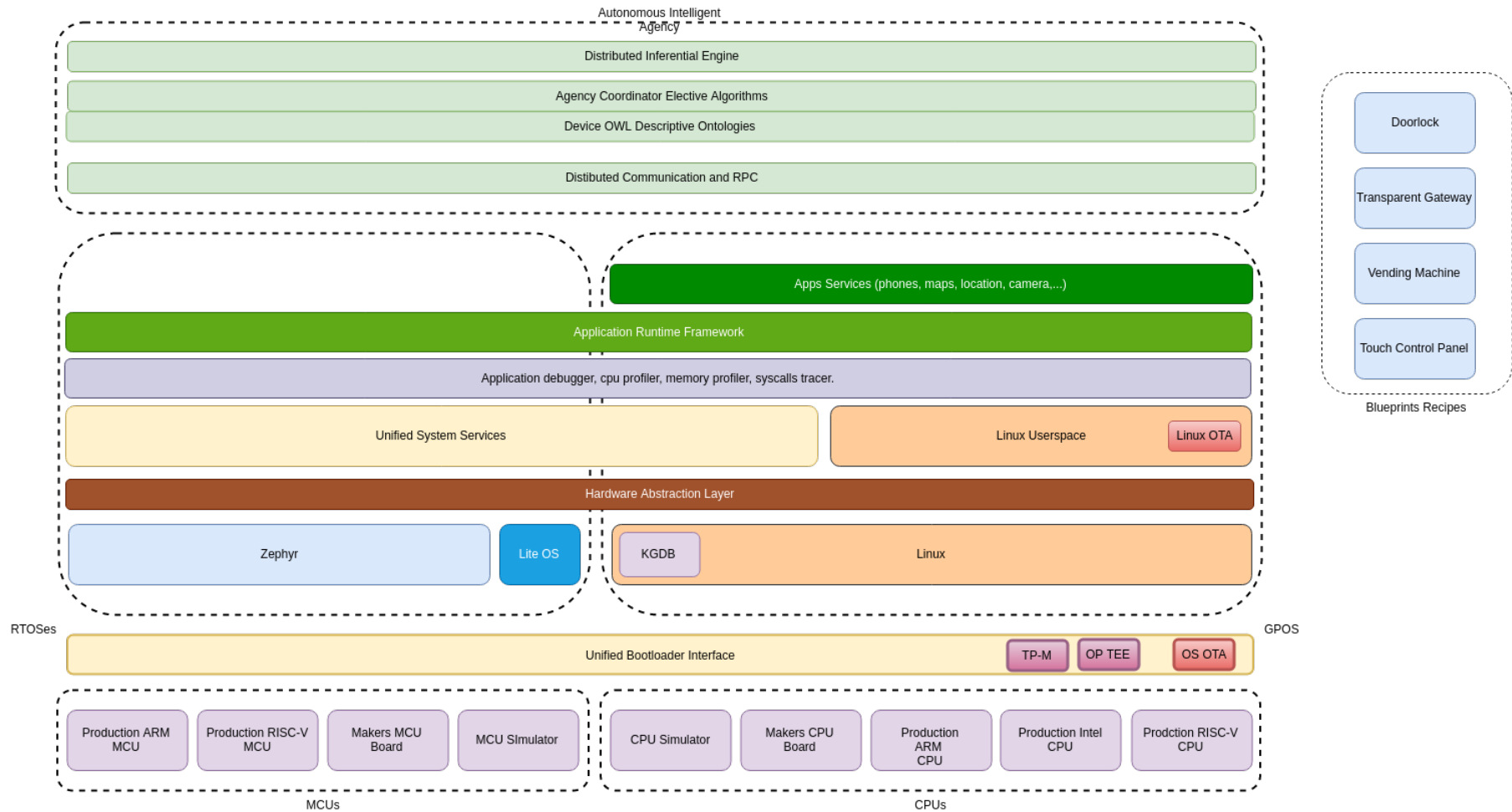
Reinventing the wheel
Sub-optimal choices
Becoming Device Dealers
Cheap Disposable Devices
Monetize consumer's data



CONTENT CREATORS

Lack of choice
Lack of standards
Drive o.s. / cloud stickiness
Influence device makers
Help monetize consumer's data

Device Functionalities	Kernel	Zephyr/LiteOS				Linux					
	CPU type	MCU				CPU					
	Number of CPUs	1		2		2	4		8		
	Display	Headless	Display				Headless	Display			
	GPU Acceleration	Simple Graphics						Accelerated Graphics			
	Application Runtime Engine	Javascript / C-C++						Javascript /C-C++ /Java			
	Application Framework	GN, Jar, ...									
Applications	Application Framework	Phone, Maps, Location, ...									
	Applications										
Distributed Functionalities	Discovery	Distributed Communications									
	Sensor	Distributed Sensors									
	Actuator	Actuator		Actuator		Actuator		Actuator			
	HMI	HMI		HMI		Dist. HMI		Dist. HMI			
	Distributed compute and storage					Distributed Compute and Storage					
	Edge AI	Autonomous Agents				Autonomous Agents and Orchestrators					
Device Performance	Communication Range - meters	10		100		1000					
	Energy Consumption	uWatts		mWatts		Watts					
	Memory footprint	kB		MB			GB				
	Processor speed - MIPS	100		500		1000		50000		100000	
Brand	OpenHarmony Device Brand Name	Things				Gateway		Mobile			
Device Type	OpenHarmony Powered Devices	Speakers, Earbud, Light Bulbs, Doorlocks, Appliances, Watches, Thermostats,...				Transparent GWs		Phones, Tables, In-car			





Our usage of Yocto/OE

Our usage of Yocto/OE

S TABILITY

LTS/dunfell

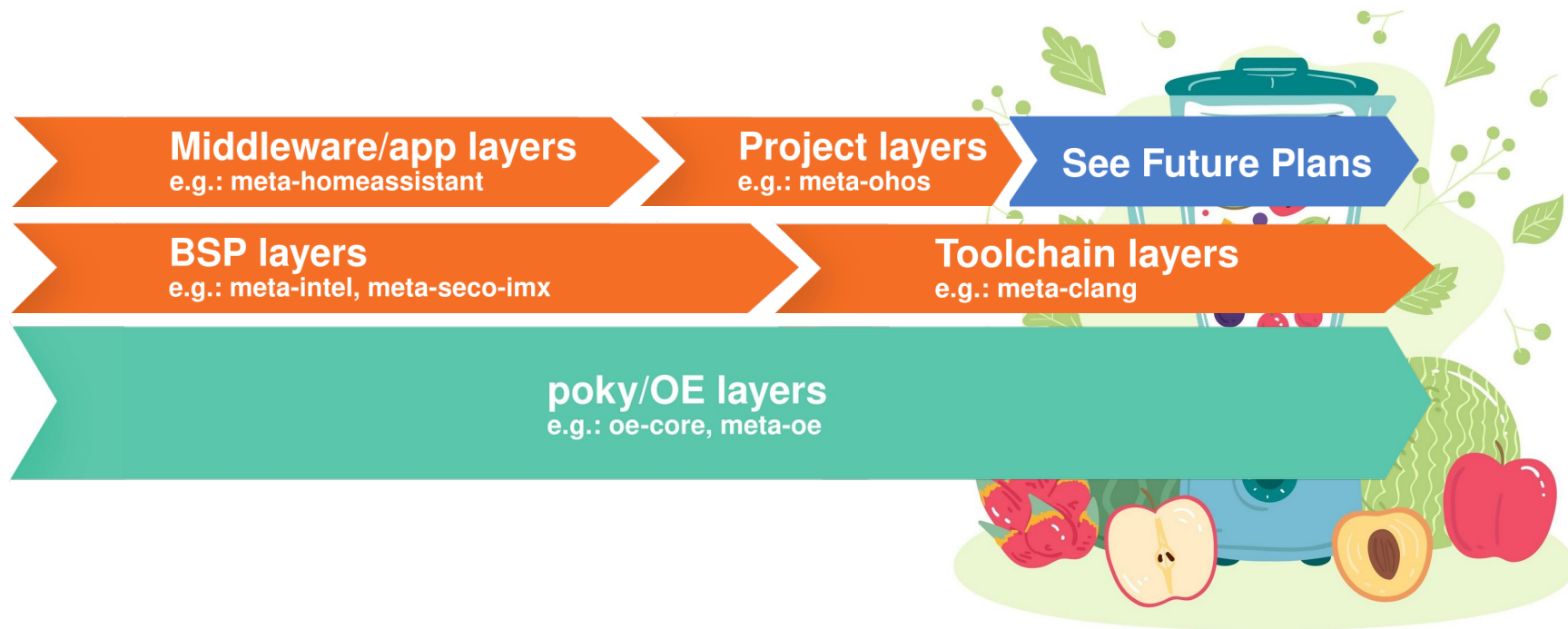


U NIFICATION



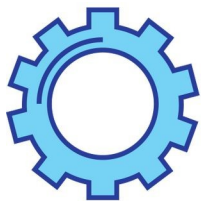
Fixed Layers Setup

Our usage of Yocto/OE: Build Setup



Our usage of Yocto/OE: Toolchain Selection

- **musl**
 - Binary footprint
 - Improvements on resource exhaustion
 - Similar level of runtime performance (compared to glibc)

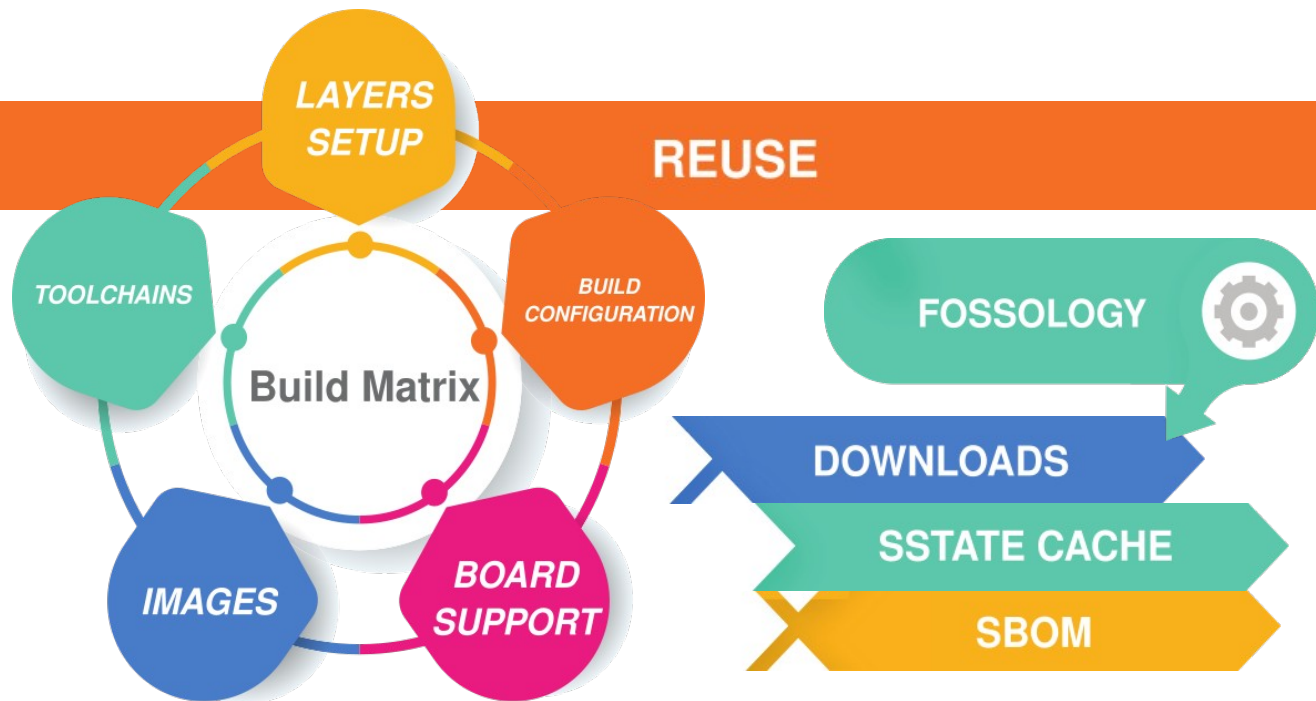


Our usage of Yocto/OE: Toolchain Selection

- **clang/LLVM**
 - Modern compiler frontend/backend
 - Support for emerging languages
 - Static analysis tools included
 - WIP:
 - dunfell support for meta-clang
 - Blacklist of unsupported packages



Our usage of Yocto/OE: CI



Our usage of Yocto/OE: Upstream Contribution





Problems We Ran Into

Problems We Ran Into: Newcomer Feedback

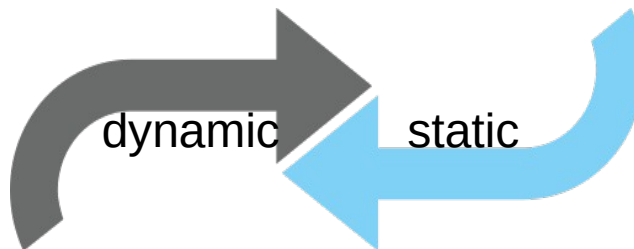
- **Debugging is not easy**
 - Build system
 - Build packages
- **Layering system**
- Build system is **too flexible**



Problems We Ran Into: meta-zephyr

west build tool

- 200+ machines
- Extra support for shield



Yocto

- 5 machines

Problems We Ran Into: BSP layers mix

Define primitives

- *Configurable*
- *Subscribable*

```
sh-5.1$ cat base...  
FILESEXTRAPATH...  
sh-5.1$ ca...  
tail -n1 layer.conf
```

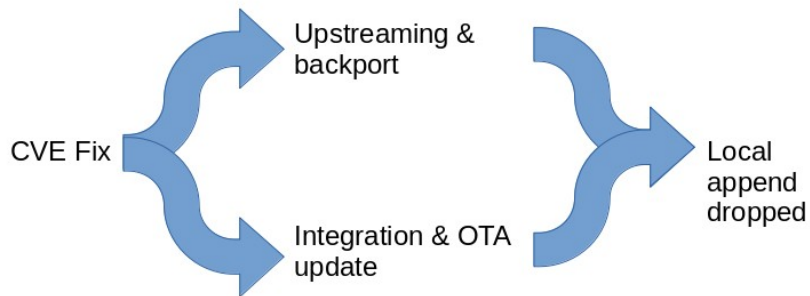
```
sh-5.1$ cat west...  
FILESEXTRAPATH...  
IMAGE_INSTALL append = " pm-utils"  
prepend_machineA := "${THISDIR}/${PN}:"  
Distro) \n \l
```

Problems We Ran Into: GN-based Projects



CVE's on Dunfell

- CVE checker in Yocto and reporting helps to keep track
- Huge thanks to Steve Sakoman
- Work in progress on our CI to run for our target images and machines
- CVE fixes important to
- upstream



Firmware Redistribution Rights

- Open Source experienced lawyers within the dev team

LegalOps



- Real world example:
- Enabling a BSP caused the automatic inclusion of firmware files into the images
- Aim: good ootb experience - Result: legal risk

SPDX Headers

- Headers have been added to scripts already
- Next step would be headers in the meta-data itself
- **oe-core LICENSE file:**
 - “All metadata is MIT licensed unless otherwise stated. Source code included in tree for individual recipes (e.g. patches) are under the LICENSE stated in the associated recipe (.bb file) unless otherwise stated.
 - License information for any other files is either explicitly stated or defaults to GPL version 2 only.”
- Start with adding MIT license header to bb files
- Gradually adding license headers to patches



Future Plans

Cross-kernel OTA Story

Linux & RTOS
images

Orchestration
system service

meta-rauc

A/B partition table
for all devices

Read-only
filesystem plus
state and data



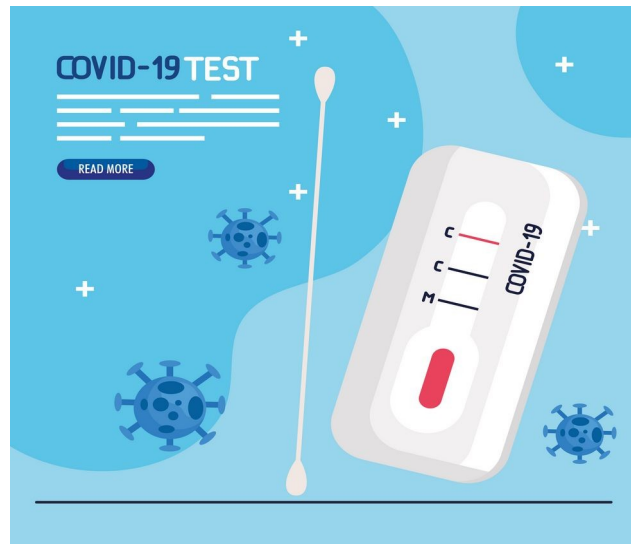
meta-liteos

Multiple kernel state:

- Linux based meta layers for Arm and Intel boards
- Meta-zephyr and meta-freertos for RTOS on MCU
- LiteOS as an additional RTOS kernel under discussion
- meta-liteos could be the result for hardware support and integration

Cross-kernel Testing Story

- Seeking a way to test across the multiple kernels
- Application Certification Test Suite, ACTS
 - A few thousand test cases
 - Linux Test Project
 - Libc testing
 - Filesystems
 - APIs



Summary

- Yocto / OpenEmbedded chosen as one of the core building blocks for All Scenarios OS
- Mixed bag of layers, kernel, IP compliance and more, but with a clear upstream policy
- Discussion on more SPDX headers in meta-data



Thank You!



HUAWEI



Open Source Technology Center

Visit us: <https://www.ostc-eu.org/>
<https://git.ostc-eu.org/>
<https://chat.ostc-eu.org/>



yocto ·
PROJECT

THE
LINUX
FOUNDATION