Greybus

Greg Kroah-Hartman

Acknowledgments

Linaro

Leaflabs

BayLibre

Google

Linux Foundation

UniPro is easy to use

Familiar high-level semantics from Internet and other networks...

- Send and receive messages of arbitrary size
- Automatic in order delivery

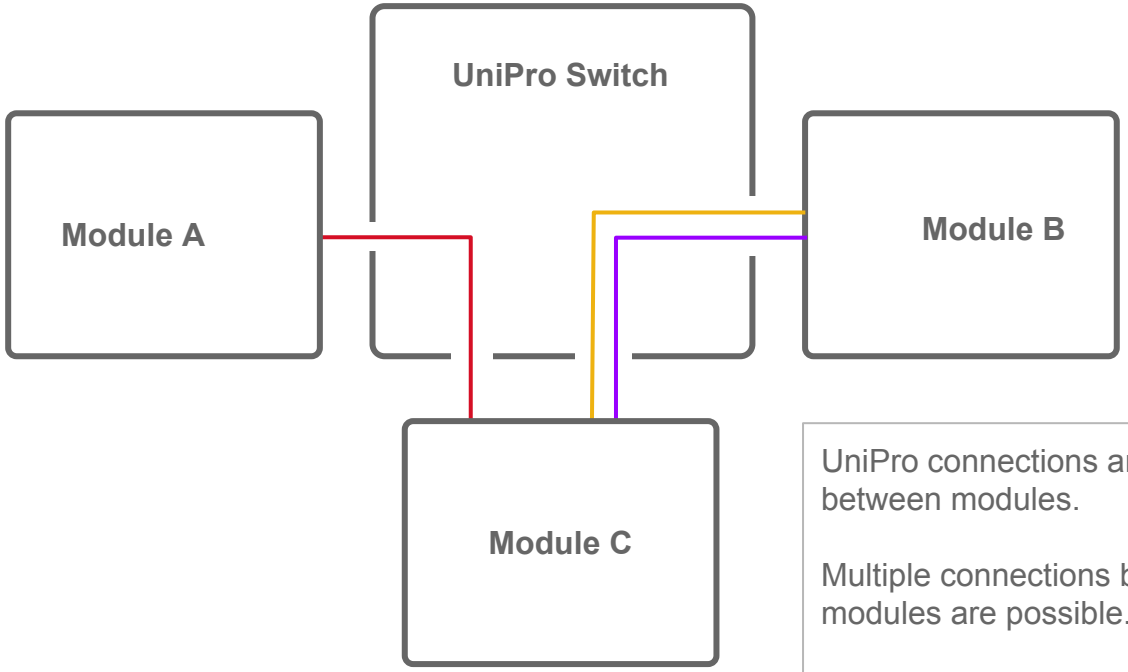
… but sophisticated underpinnings

- Mature and robust technology stack
- Leverages existing high-speed PHY
- Low-power modes
- Low latency
- QoS features

Mobile-friendly basis for other peripheral communication

- UniPro-based camera standard: CSI-3
- UniPro-based flash storage standard: UFS

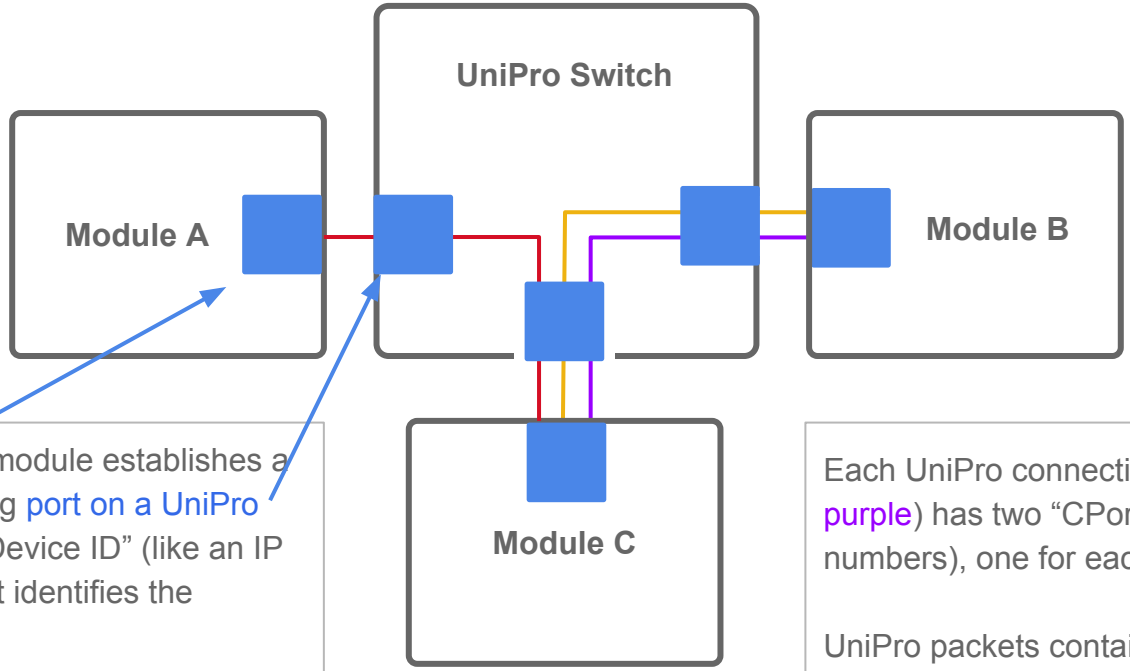
UniPro provides bidirectional connections for data transfer



UniPro connections are established between modules.

Multiple connections between modules are possible.

UniPro provides bidirectional connections for data transfer



Each UniPro port on a module establishes a link with a corresponding port on a UniPro switch in the Endo. A “Device ID” (like an IP address) in each packet identifies the destination module.

Each UniPro connection (red, yellow, and purple) has two “CPort IDs” (like port numbers), one for each module.

UniPro packets contain destination CPort IDs, which identify the packet’s associated connection.

UniPro key features and non-features

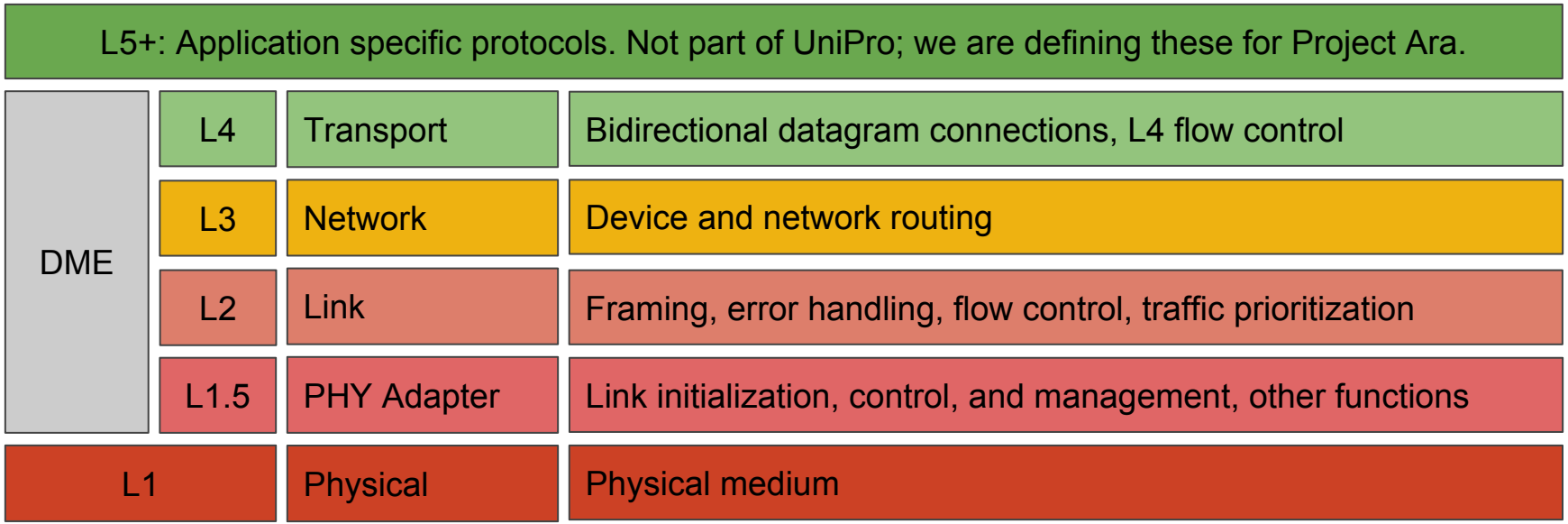
Features

- Multiple independent bidirectional connections between endpoints
- Reliable, in order transmission and reception of datagrams
- Error handling
- Credit-based flow control
- Traffic prioritization

Non-Features

- Stream transmission and reception (datagrams only)
- Multicast (endpoint-to-endpoint only)

UniPro strictly adheres to OSI layering



Greybus

An application layer for Unipro

Greybus protocol

- Device discovery and description
- Network routing and housekeeping
- Class protocols

Device description

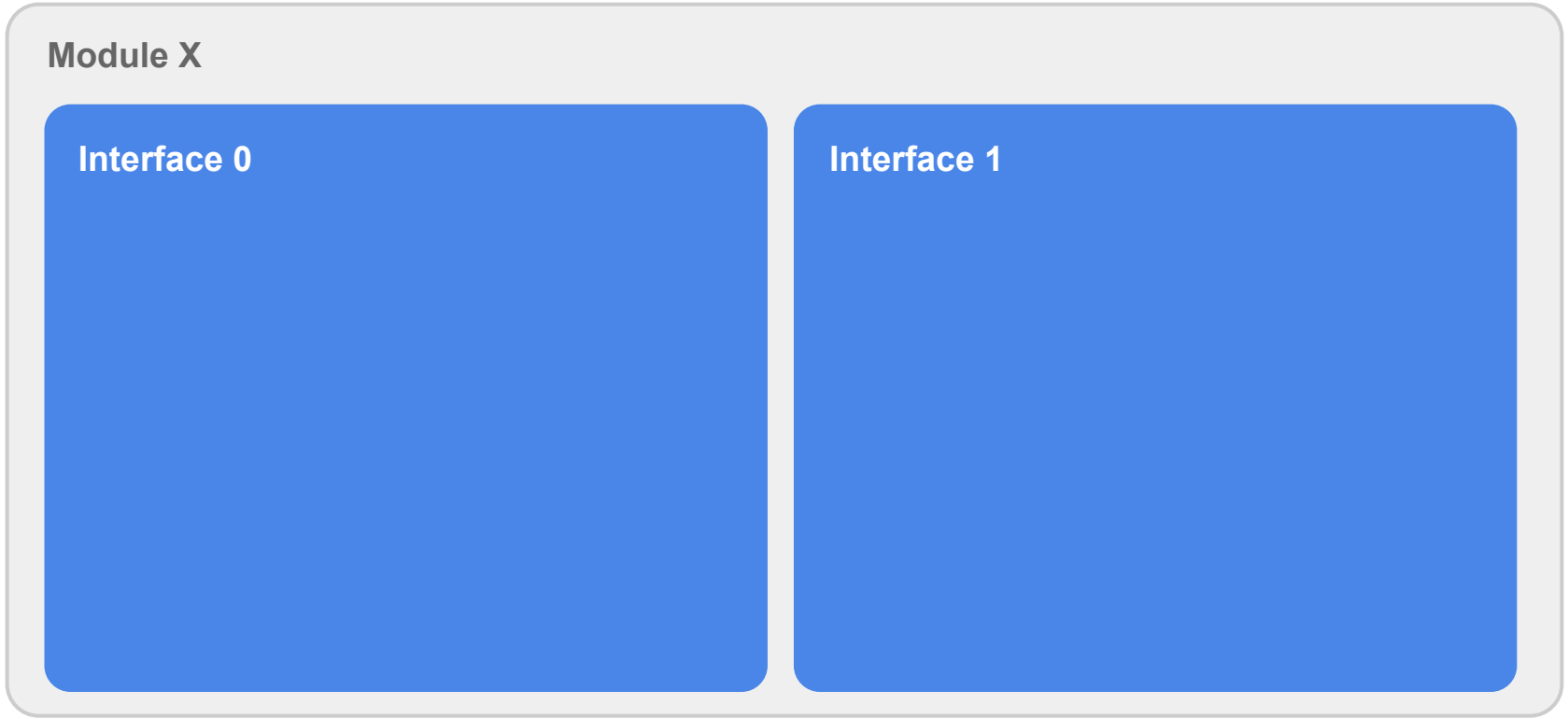
A Manifest describes the device to the system.

Examples of this information:

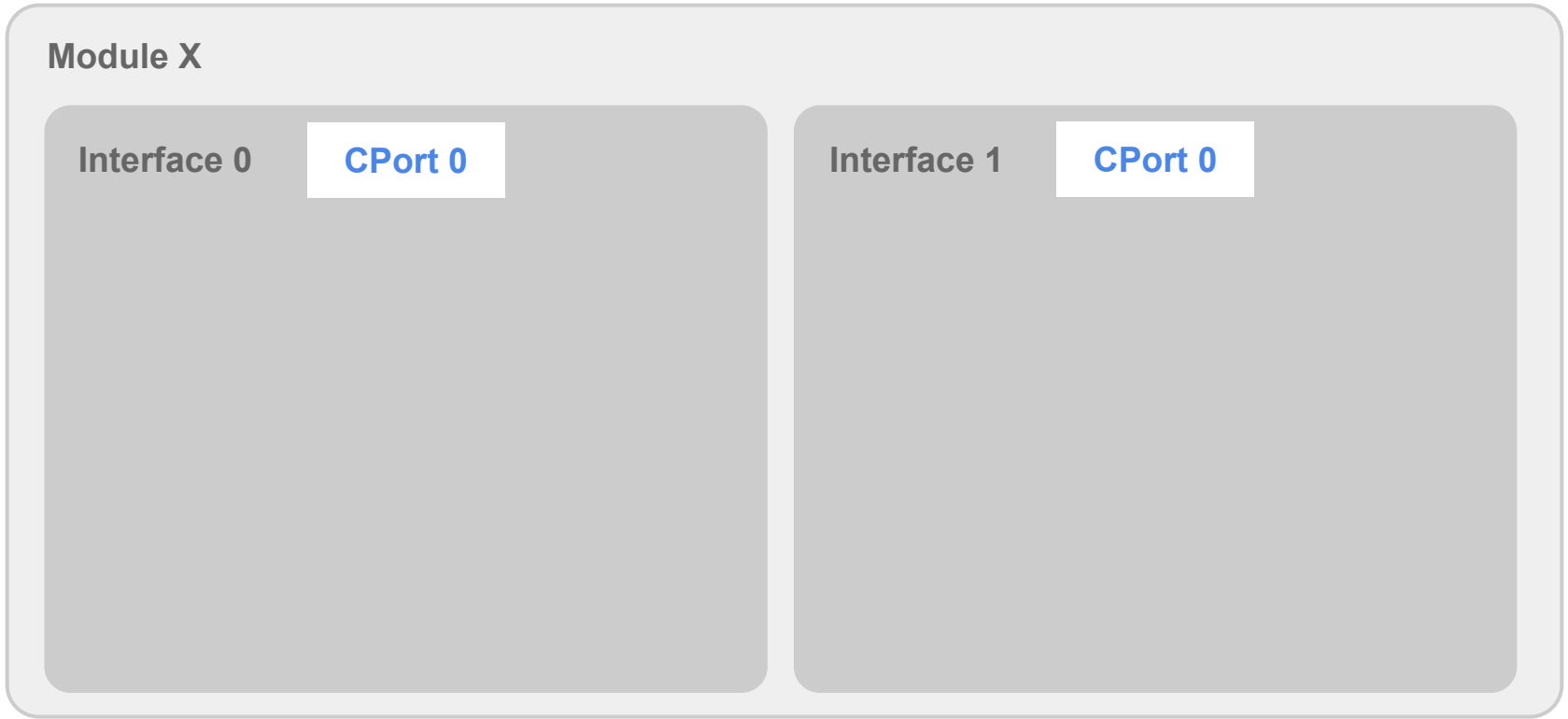
- Vendor and Product ID
- Serial Number
- “Bundles”
- Protocols used

Modules Interfaces CPorts Bundles

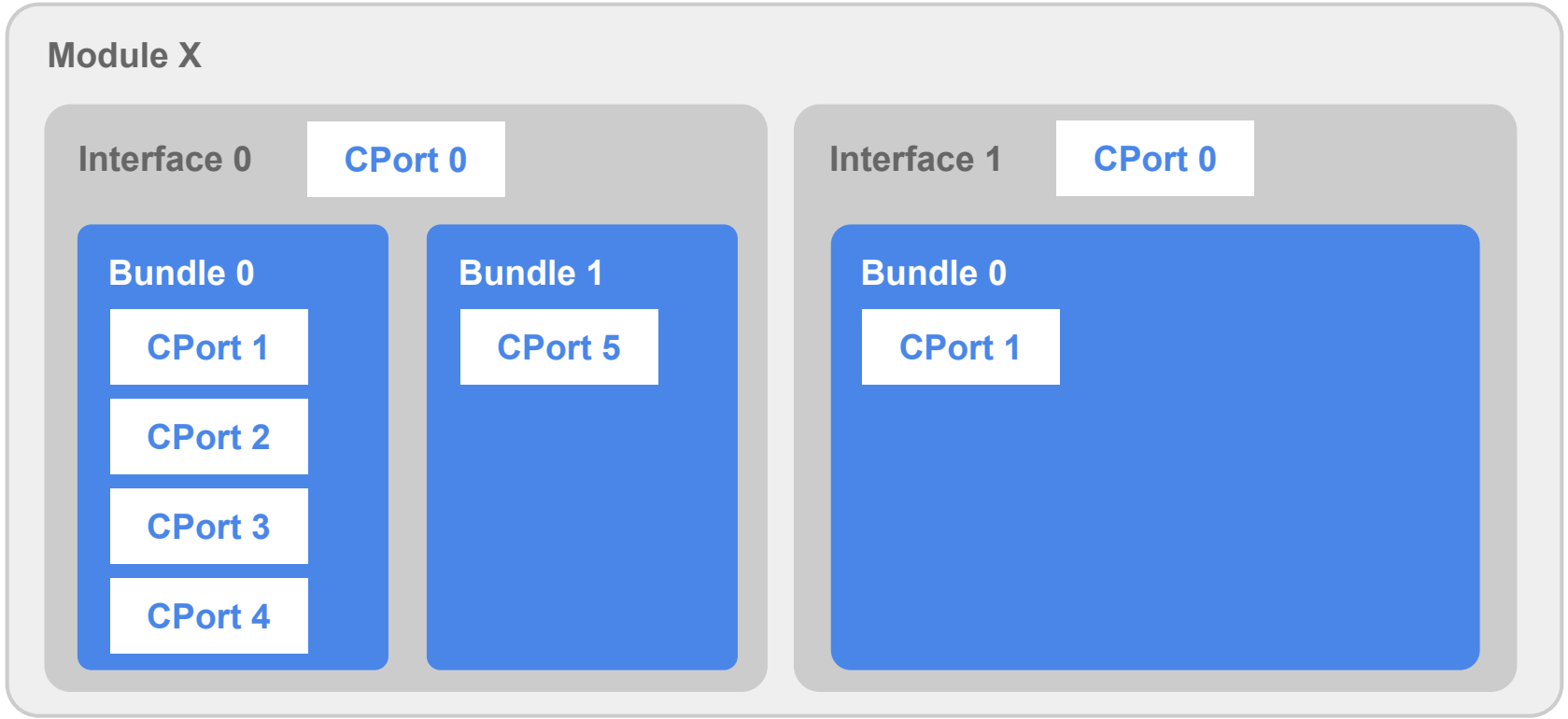
Modules Interfaces CPorts Bundles



Modules Interfaces CPorts Bundles



Modules Interfaces CPorts Bundles



Operations

- RPC-like way to talk to an interface
- Request / Response communication

Operations

- RPC-like way to talk to an interface
- Request / Response communication

```
int get_version(char *major, char *minor);

int vibrator_on(short timeout_ms);
```

Protocol classes

Protocol classes

Implemented

- Battery
- Vibrator
- NFC

Protocol classes

Implemented

- Battery
- Vibrator
- NFC

In progress

- Audio
- Input (HID)
- Sensors
- Camera

Protocol classes

Implemented

- Battery
- Vibrator
- NFC

In progress

- Audio
- Input (HID)
- Sensors
- Camera

Later

- Wifi
- Bluetooth
- Cellular modem
- GPS
- Lights
- Display

Bridged PHY protocols

- USB
- I^2C
- I^2S
- GPIO
- SPI
- SDIO
- UART
- PWM

“Tunneled” protocols

- CSI
- DSI

Linux kernel implementation

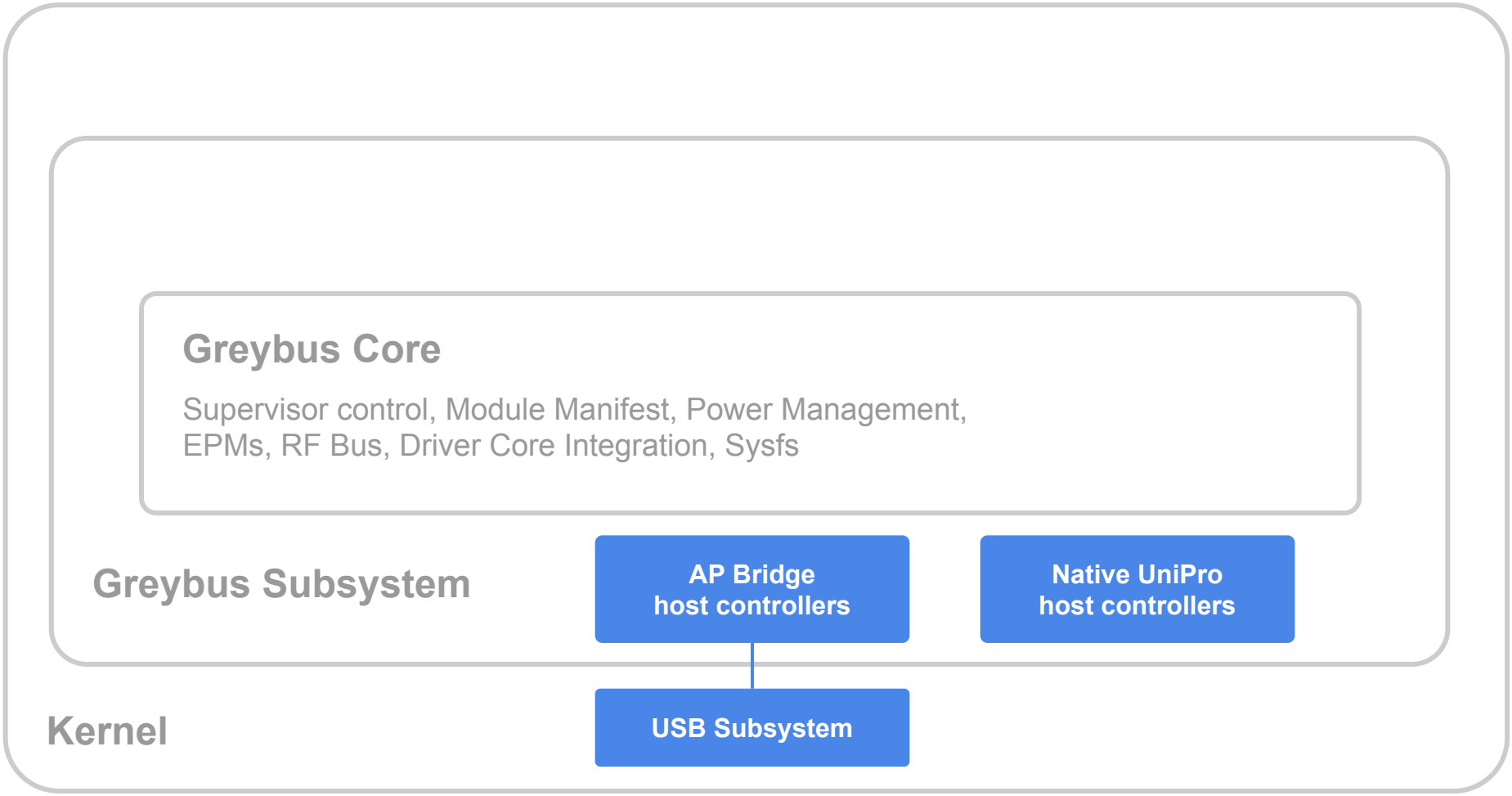
```
github.com/projectara/greybus
```

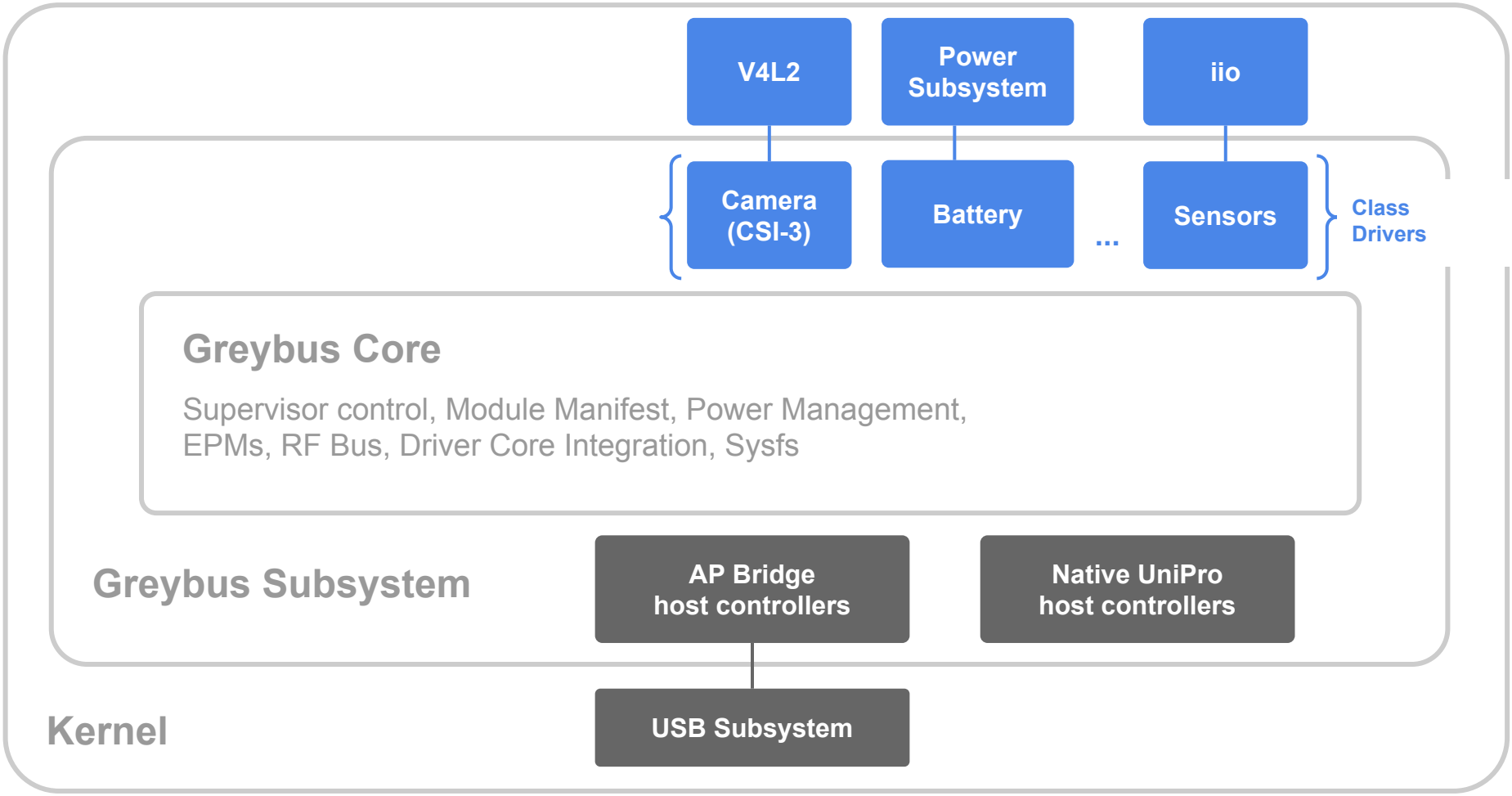
Greybus Core

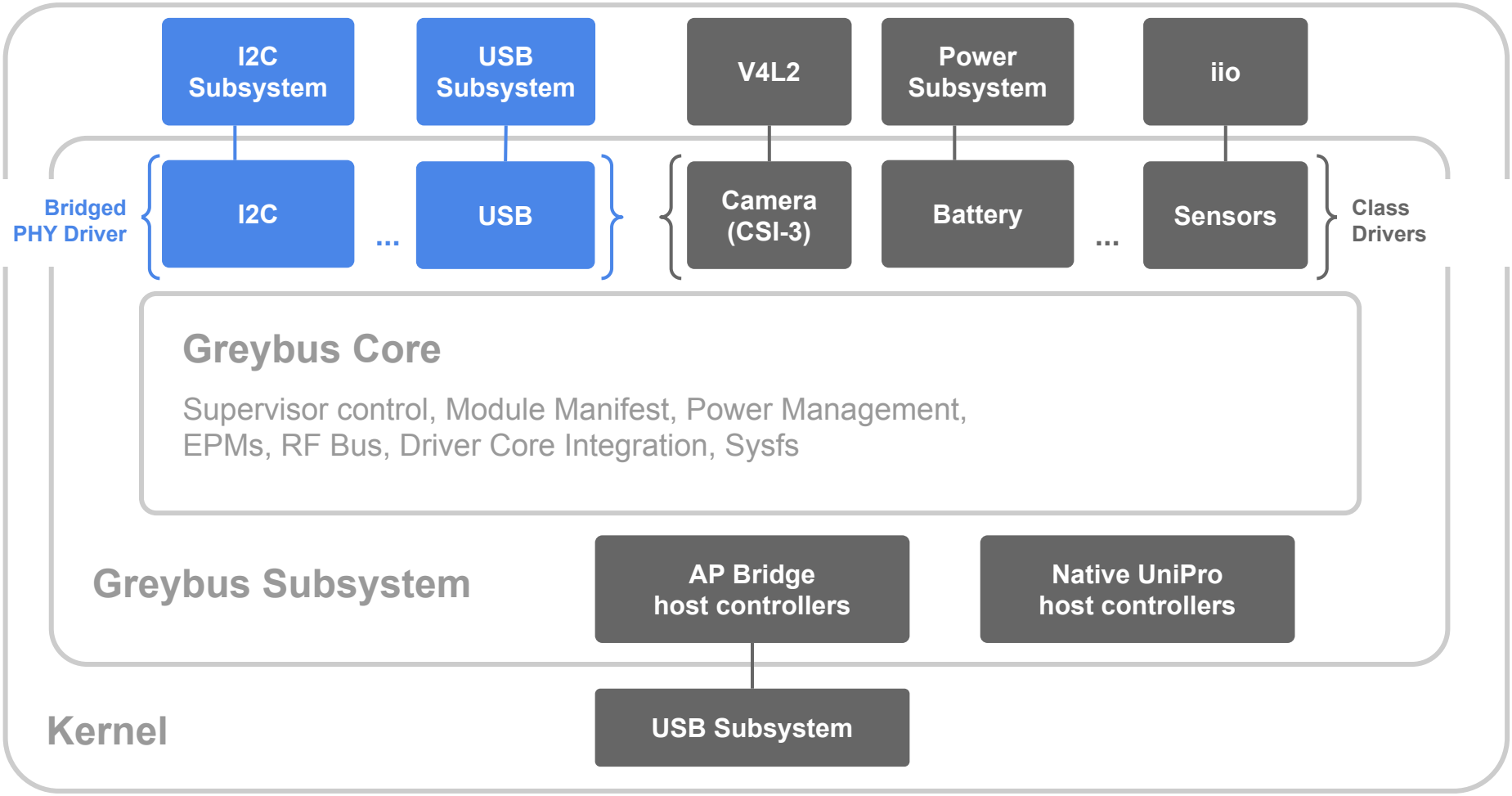
Supervisor control, Module Manifest, Power Management,
EPMs, RF Bus, Driver Core Integration, Sysfs

Greybus Subsystem

Kernel


Greybus Core
Supervisor control, Module Manifest, Power Management,
EPMs, RF Bus, Driver Core Integration, Sysfs
Greybus Subsystem
AP Bridge
host controllers
Native UniPro
host controllers
USB Subsystem
Kernel


V4L2
Power Subsystem
iio
Camera (CSI-3)
Battery
...
Sensors
Class Drivers
Greybus Core
Supervisor control, Module Manifest, Power Management,
EPMs, RF Bus, Driver Core Integration, Sysfs
Greybus Subsystem
AP Bridge host controllers
Native UniPro host controllers
USB Subsystem
Kernel


I2C Subsystem
USB Subsystem
V4L2
Power Subsystem
iio
Bridged PHY Driver
I2C
...
USB
Camera (CSI-3)
Battery
...
Sensors
Class Drivers
Greybus Core
Supervisor control, Module Manifest, Power Management,
EPMs, RF Bus, Driver Core Integration, Sysfs
Greybus Subsystem
AP Bridge host controllers
Native UniPro host controllers
USB Subsystem
Kernel

Linux kernel - Bridged-PHY support

- USB devices “just work”
- UART devices “just work”
- Userspace access provided for rest
- Existing kernel drivers need 20 lines added