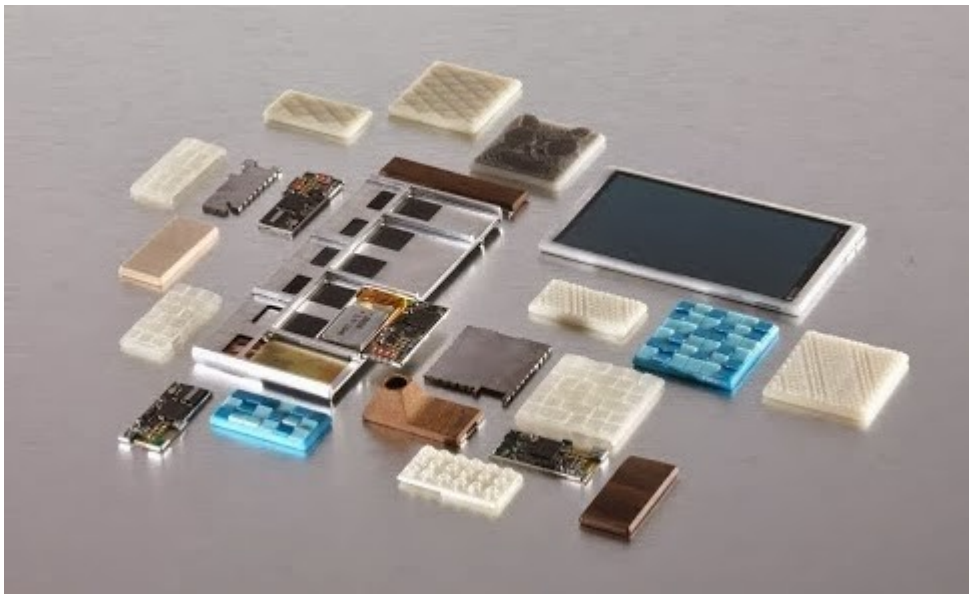
Project Ara: Redefining Handset and Android Architecture



Android MC / LPC 2015

Karim Yaghmour

>OPERSYS™

DISCLAIMER:

I do **NOT** speak for Google

I do **NOT** speak for Project Ara

Project Ara = lots of teams from lots of organizations

My role = help with Android architecture

I know **nothing** of:

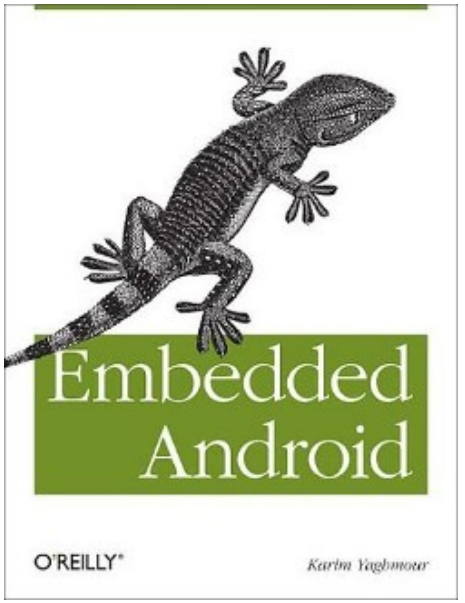
Release dates, Pricing, Business plans, etc.

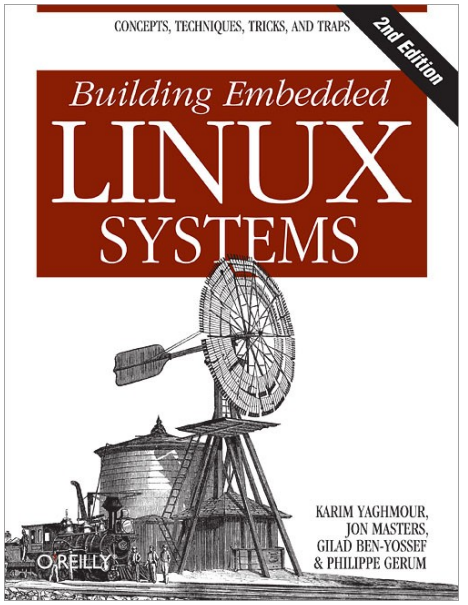
In short:

1. File this presentation under “creative artwork”
2. All opinions are mine :-)

About

- Author of:





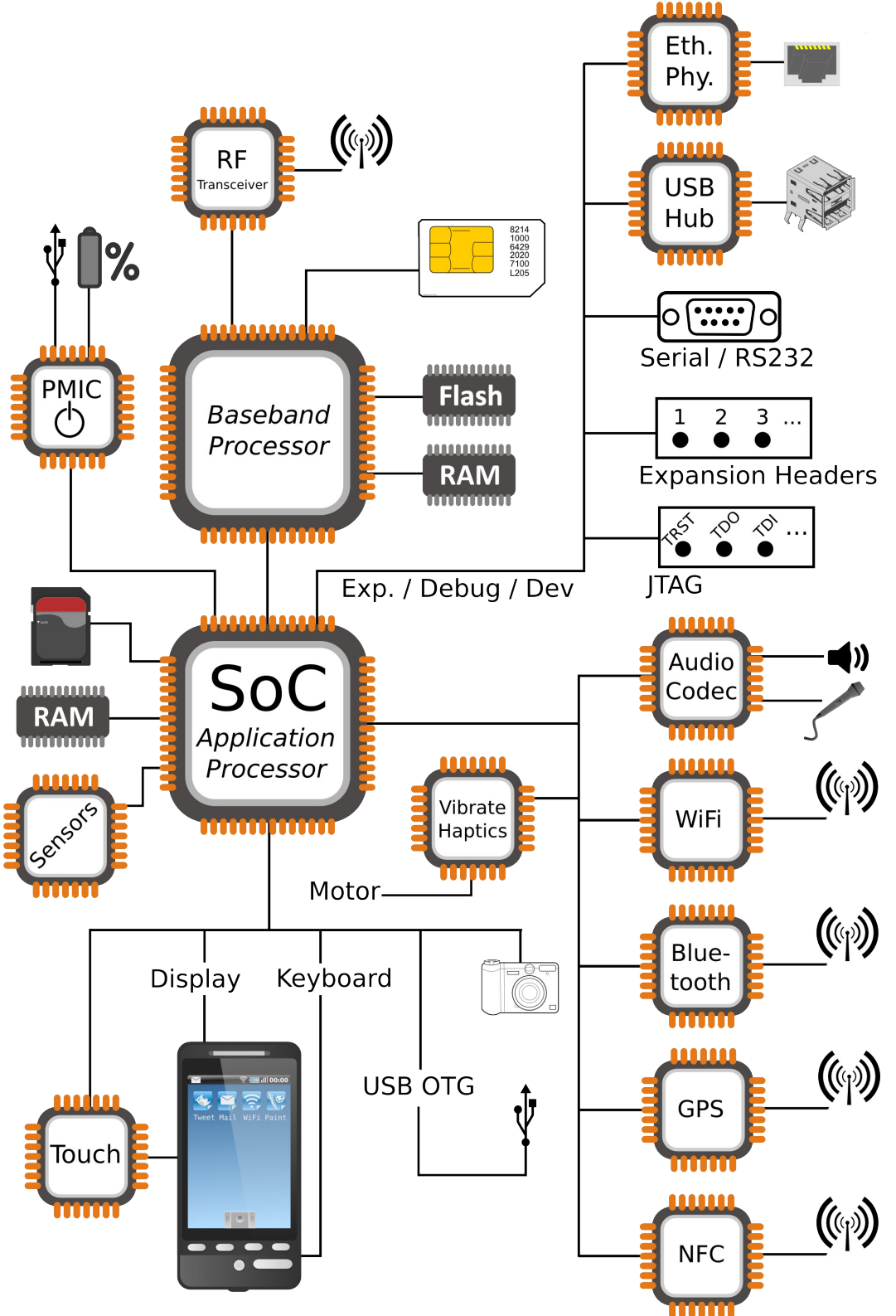
- Introduced Linux Trace Toolkit in 1999
- Originated Adeos and relayfs (kernel/relay.c)
- Ara Android Arch Oversight
- Training, Custom Dev, Consulting, ...

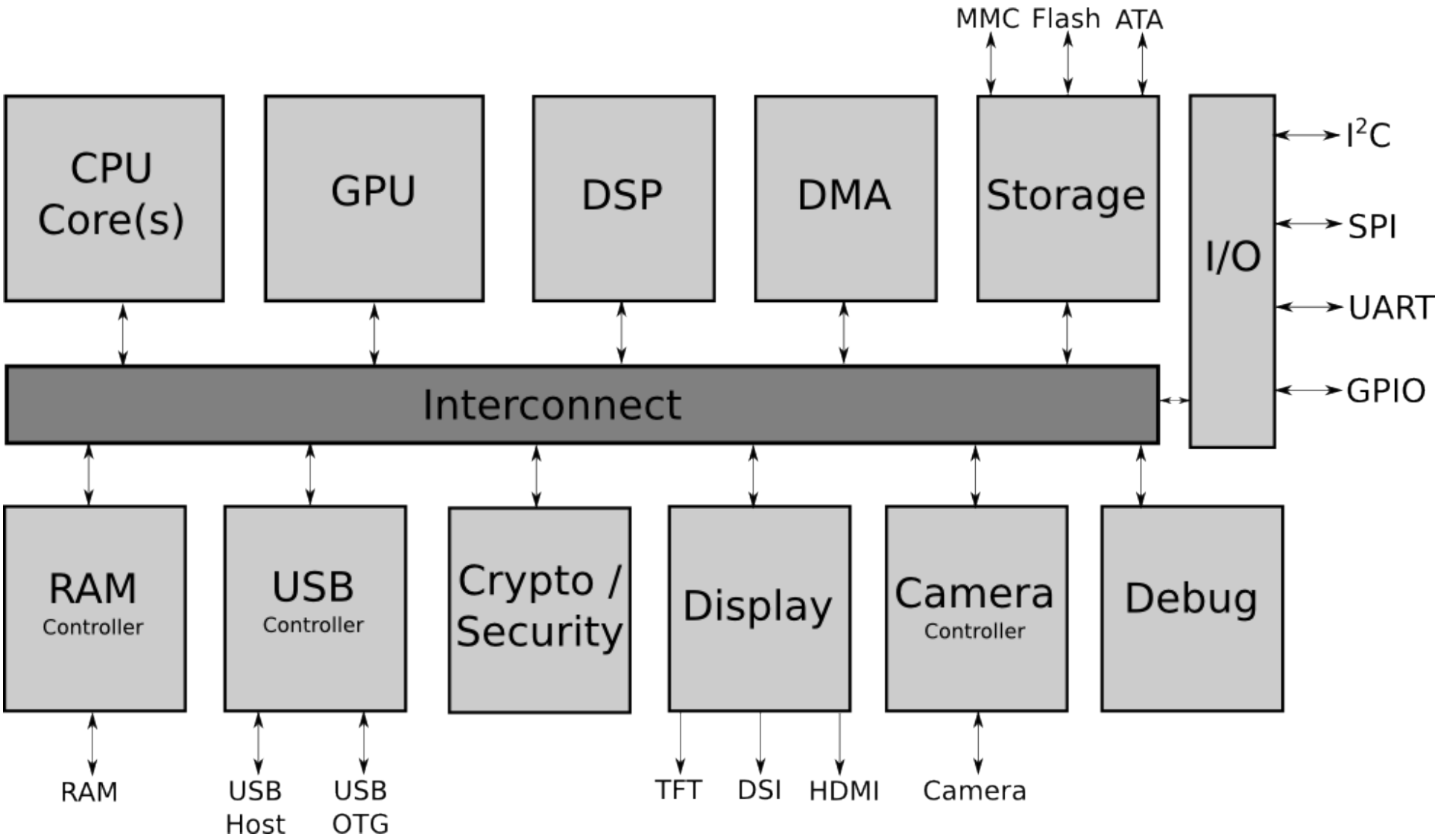
Project Ara: Redefining Handset and Android Architecture

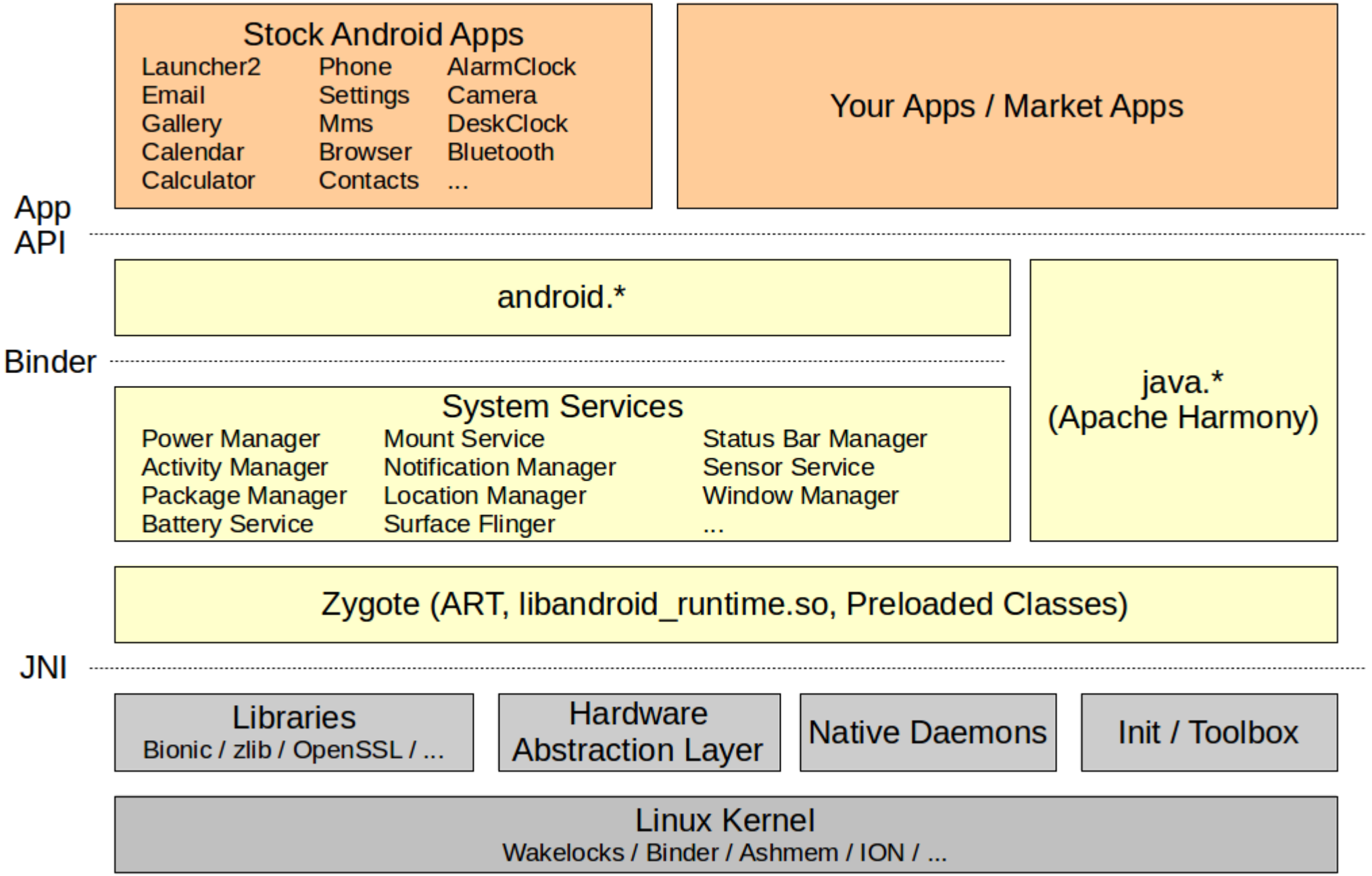
1. Traditional Phone Architecture
2. Origins and Goals of Project Ara
3. Module Ecosystem
4. Cool Technologies
5. Hardware Architecture
6. Software Architecture
7. Challenges
8. What's next?

1. Traditional Phone Architecture

- Hardware Schematic
- SoC
- Android


Eth.
Phy.
RF
Transceiver
USB
Hub
8214
1000
6429
2020
7100
L205
Serial / RS232
PMIC
Baseband
Processor
Flash
RAM
1 2 3 ...
Expansion Headers
TRST TDO TDI ...
JTAG
Exp. / Debug / Dev
SoC
Application
Processor
RAM
Audio
Codec
Sensors
Vibrate
Haptics
WiFi
Motor
Blue-
tooth
Display
Keyboard
GPS
Touch
USB OTG
NFC


MMC
Flash
ATA
CPU Core(s)
GPU
DSP
DMA
Storage
I/O
I²C
SPI
UART
GPIO
Interconnect
RAM Controller
USB Controller
Crypto / Security
Display
Camera Controller
Debug
RAM
USB Host
USB OTG
TFT
DSI
HDMI
Camera


Stock Android Apps
Launcher2
Email
Gallery
Calendar
Calculator
Phone
Settings
Mms
Browser
Contacts
AlarmClock
Camera
DeskClock
Bluetooth
...
Your Apps / Market Apps
App API
android.*
java.* (Apache Harmony)
Binder
System Services
Power Manager
Activity Manager
Package Manager
Battery Service
Mount Service
Notification Manager
Location Manager
Surface Flinger
Status Bar Manager
Sensor Service
Window Manager
...
Zygote (ART, libandroid_runtime.so, Preloaded Classes)
JNI
Libraries
Bionic / zlib / OpenSSL / ...
Hardware Abstraction Layer
Native Daemons
Init / Toolbox
Linux Kernel
Wakelocks / Binder / Ashmem / ION / ...

2. Origins and Goals of Project Ara

- DARPA
- ATAP
- MAKEwithMOTO
- Modu
- Phonebloks
- Launch of Project Ara
- Project Ara Today

2.1. DARPA

- Defense Advanced Research Projects Agency
- Launched as a reaction to Sputnik in 1958
- Prevent and preempt technological surprise
- Birthplace of the Internet

- Key Tenets
  - Small/Flexible
  - Flat
  - Autonomous
  - Short mandates
  - World-class teams

2.2. ATAP

- Advanced Technologies and Projects
- Founded in Motorola in 2012
- Started by DARPA veterans
- Based on DARPA model
- Part of Google's Motorola acquisition
- Retained by Google after Lenovo acquisition

2.3. MAKEwithMOTO

- 5 month tour in 2013
- 16 stops: 12 universities & 4 Maker Faires
- Short creative marathons
- Mobile device-centric projects

2.4. Modu

- Israeli startup started in 2007
- Modular cell phone to be used in other devices
- Customizable look and feel
- Ceased to operate in February 2011
- Patents purchased by Google

2.5. Phonebloks

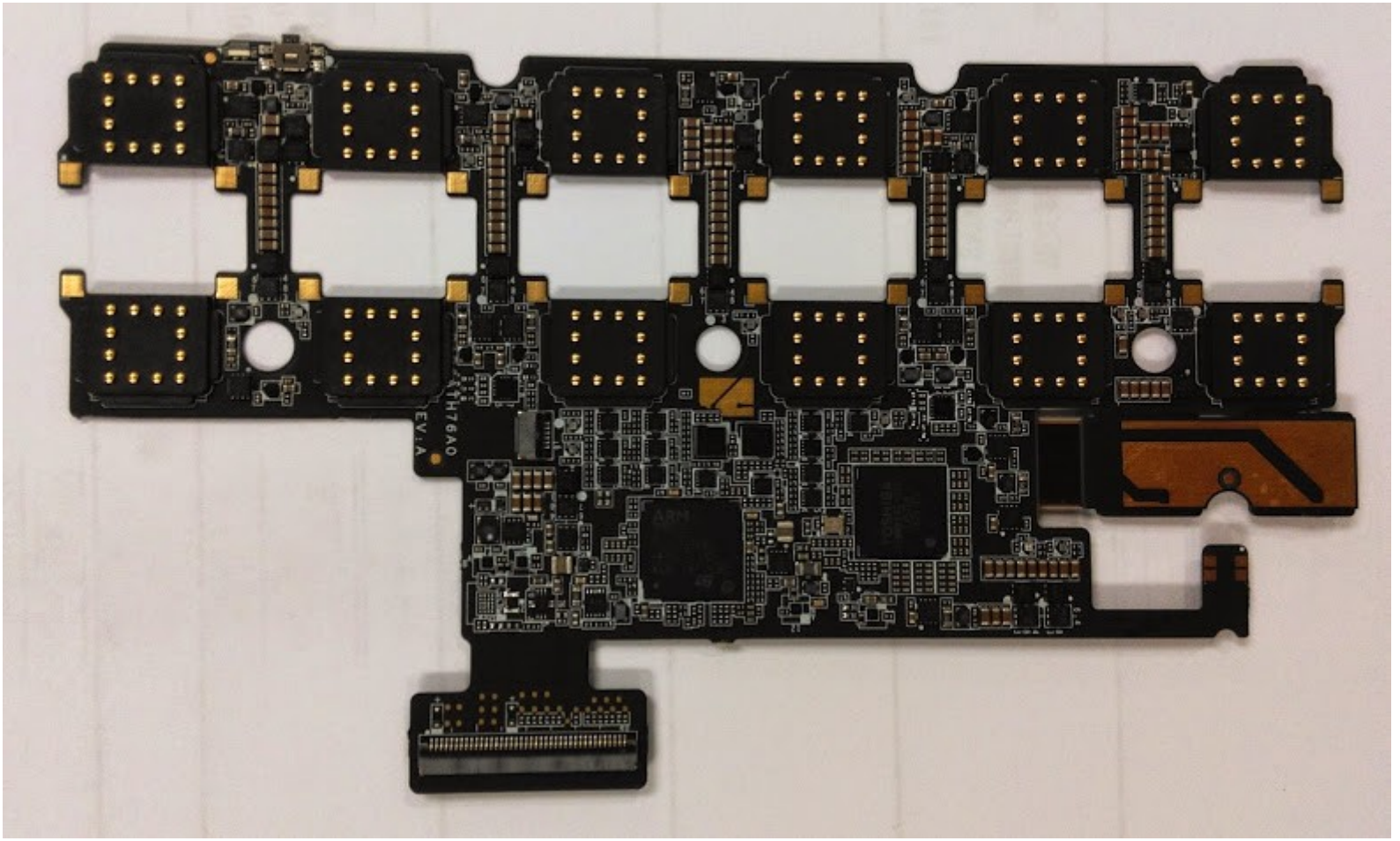
- Initially just a concept video
- Internet hit coincides with Project Ara's creation
- Invited to Collaborate with Project Ara (https://www.youtube.com/watch?v=BaPf4ZIbDVM)

2.6. Launch of Project Ara

- Announced by Motorola in October 2013
- Led by Paul Eremenko, from MAKEwithMOTO
- First developer conference April 15-16, 2014
- Announcement of the Module Development Kit

Demo at convincing scale

2.7. Project Ara Today


EV1.A

3. Module Ecosystem

- What's a module:
  - Ara building block
  - Much like an app in the app ecosystem
- Module developers are independent players
- Module market / ecosystem for getting/provisioning modules
- Rules on how to build modules -- MDK:
  - Form-factors
  - Design language
  - Software
  - etc.
- Get the MDK from projectara.com

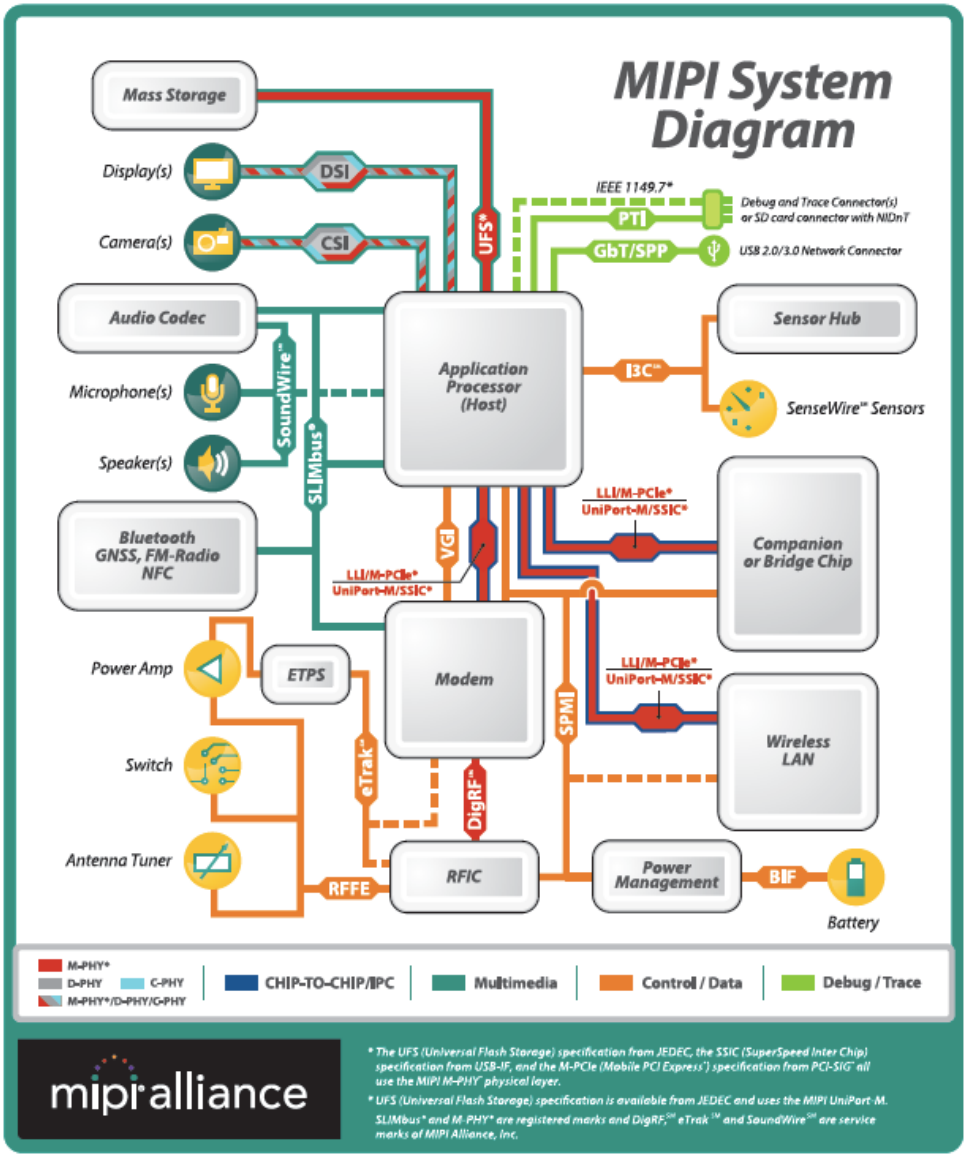
- What to do for a module -- typically:
  - Hardware (actual pluggable module)
  - Firmware (to communicate with other modules)
  - App (to run on Android)

- Obvious modules:
  - AP
  - Modem
  - Wifi
  - LCD
  - Speaker
  - Camera
  - Batteries,
  - etc.
- Other modules:
  - Whatever you can think of ...

4. Cool Technologies

- MIPI UniPro
- Capacitive (contactless) connectors
- Endoskeleton
- EPMs
- Battery charge/recharge
- Printable covers
- Greybus
- gbsim

4.1. MIPI UniPro



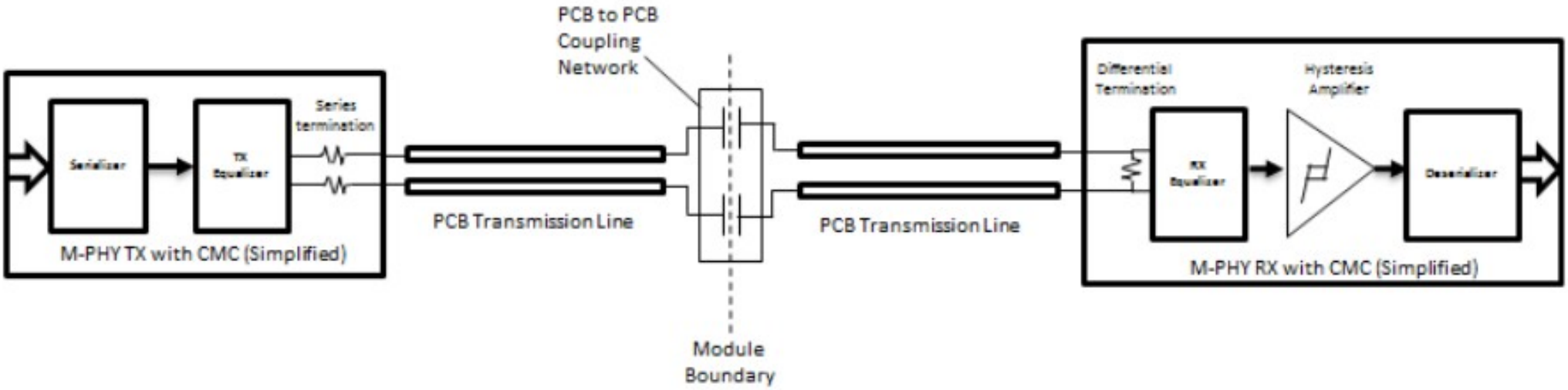
4.1.1 UniPro = Unified Protocol

UniPro protocol stack (this color-coding is a long-standing UniPro tradition)

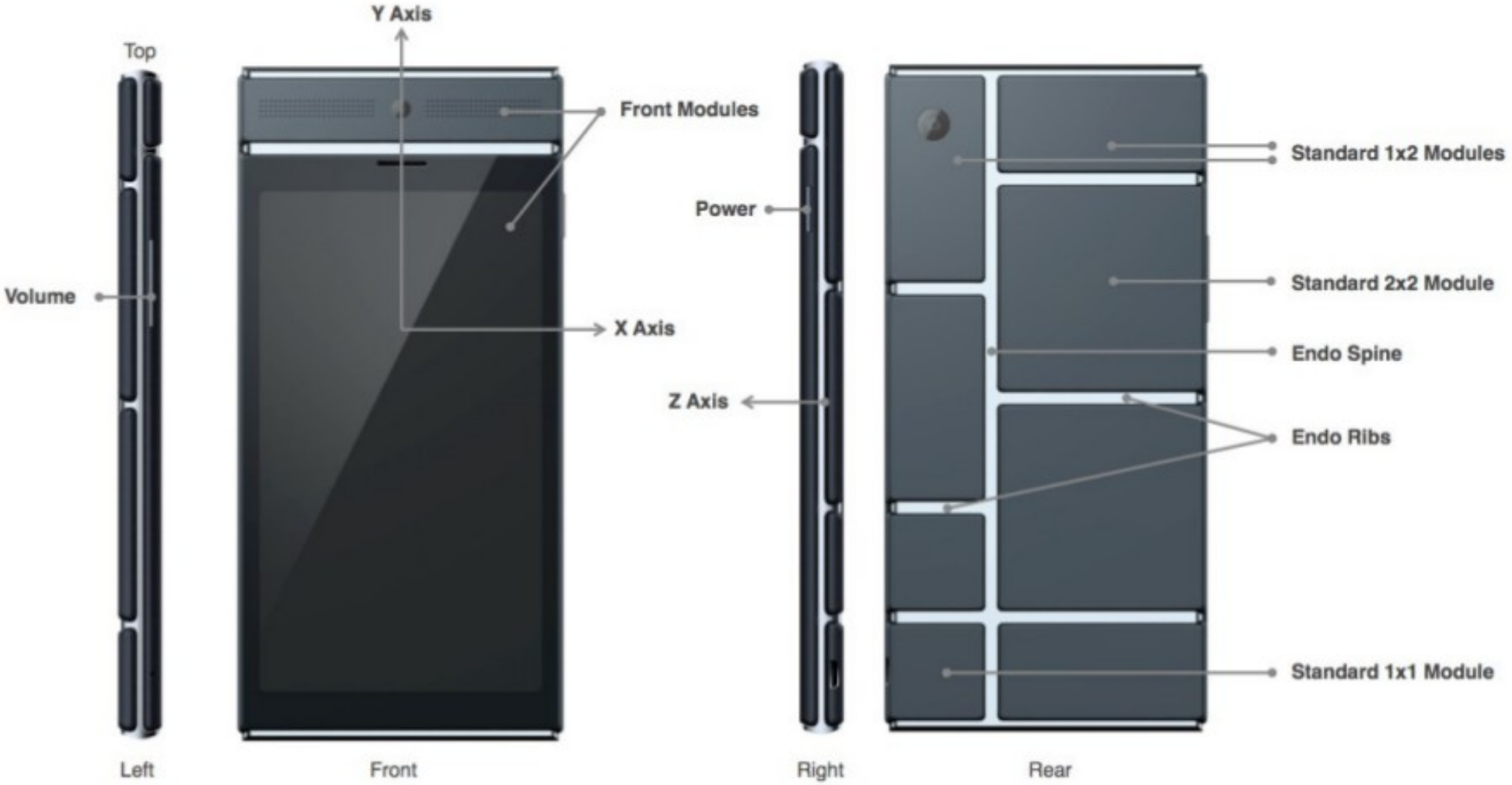
Layer #		Layer name	Functionality	Data unit name
LA		Application	Payload and transaction semantics	Message
DME	Layer 4	Transport	Ports, multiplexing, flow control	Segment
	Layer 3	Network	Addressing, routing	Packet
	Layer 2	Data link	Single-hop reliability and priority-based arbitration	Frame
	Layer 1.5	PHY adapter	Physical layer abstraction and multi-lane support	UniPro symbol
Layer 1		Physical layer (PHY)	Signaling, clocking, line encoding, power modes	PHY symbol

- UFS
- CSI-3
- DSI-2
- GBT
- UniPort-M (Ara -- UniPro with M-PHY)

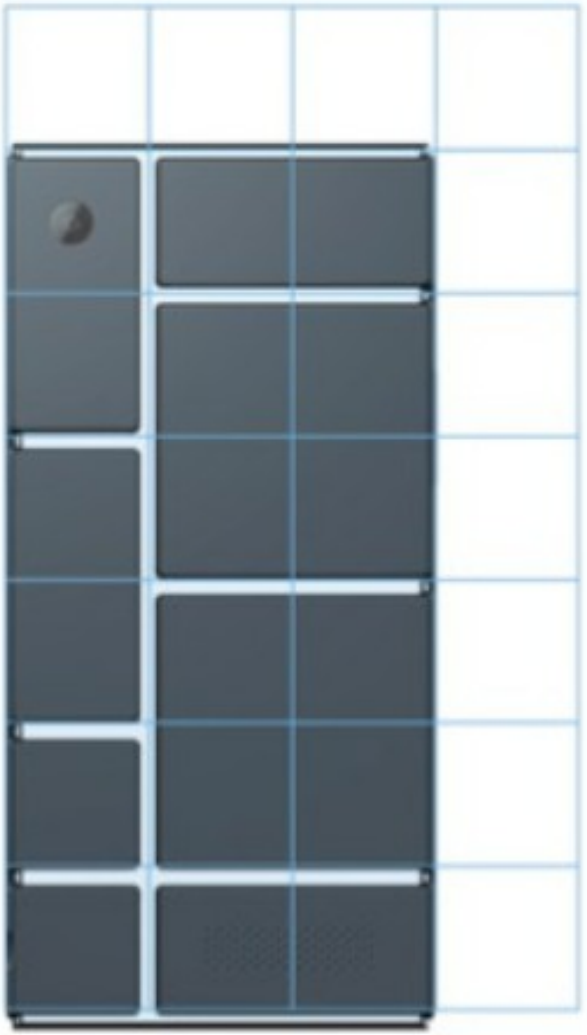
4.2. Capacitive (contactless) connectors



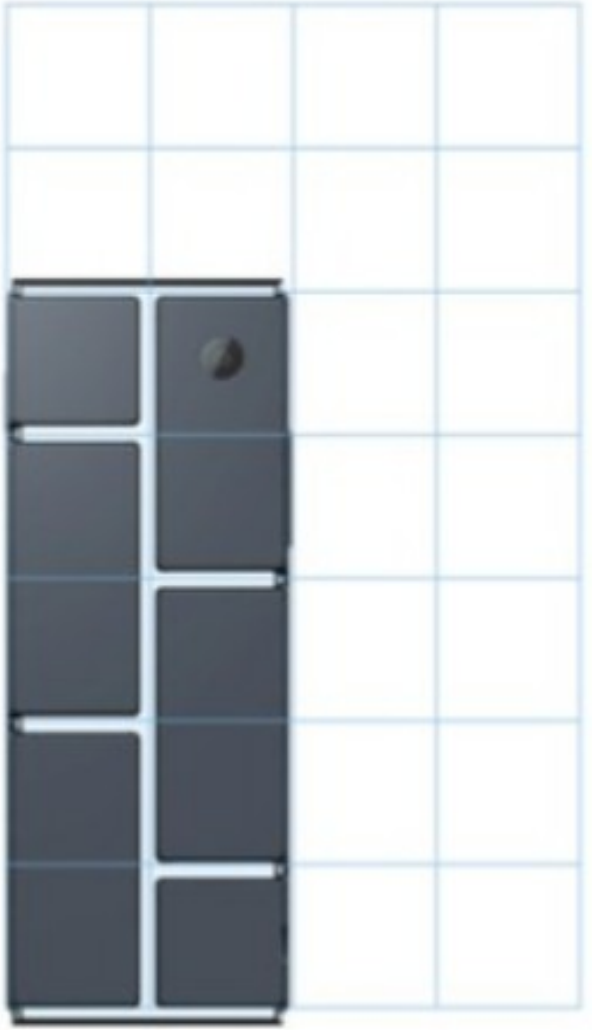
4.3. Endoskeleton



4x7
(TBD, Future Release)

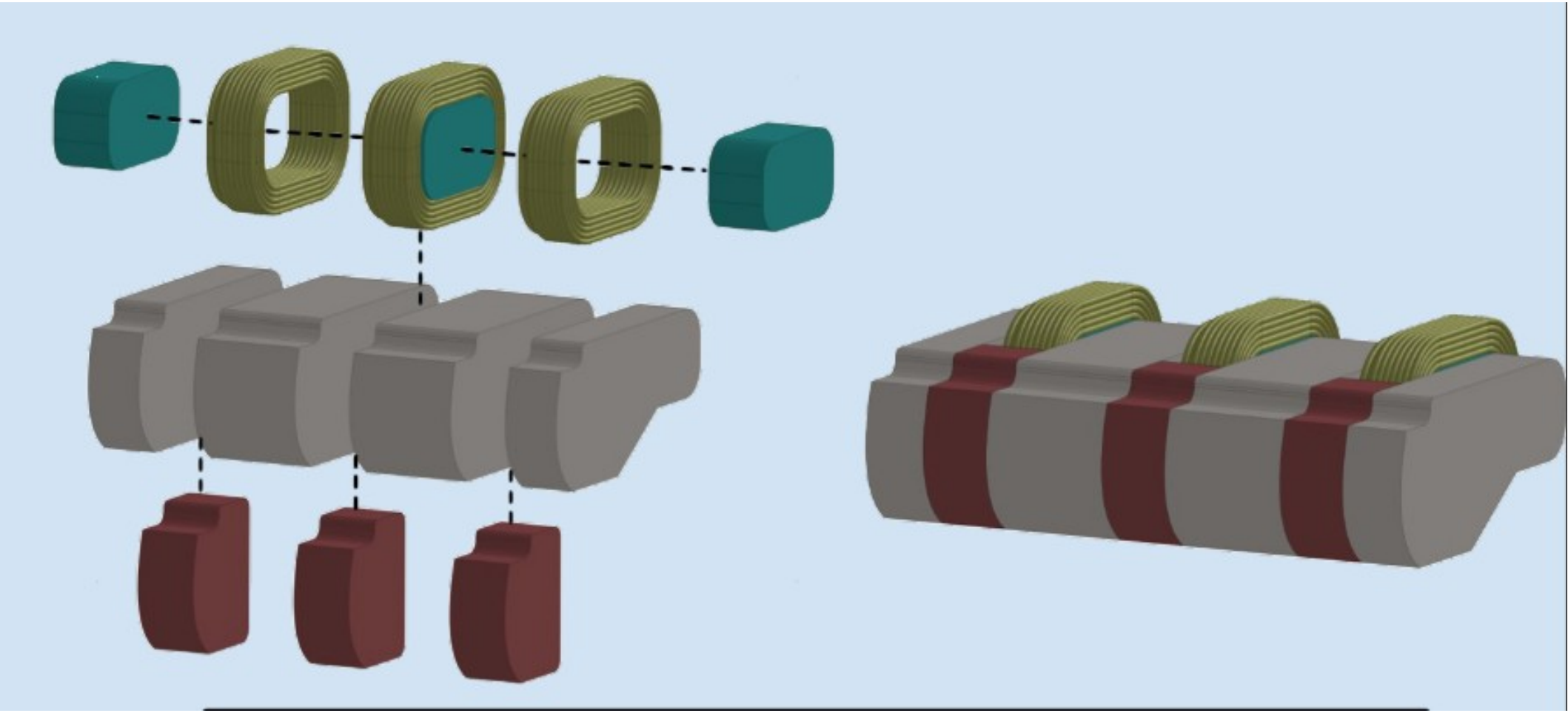


3x6

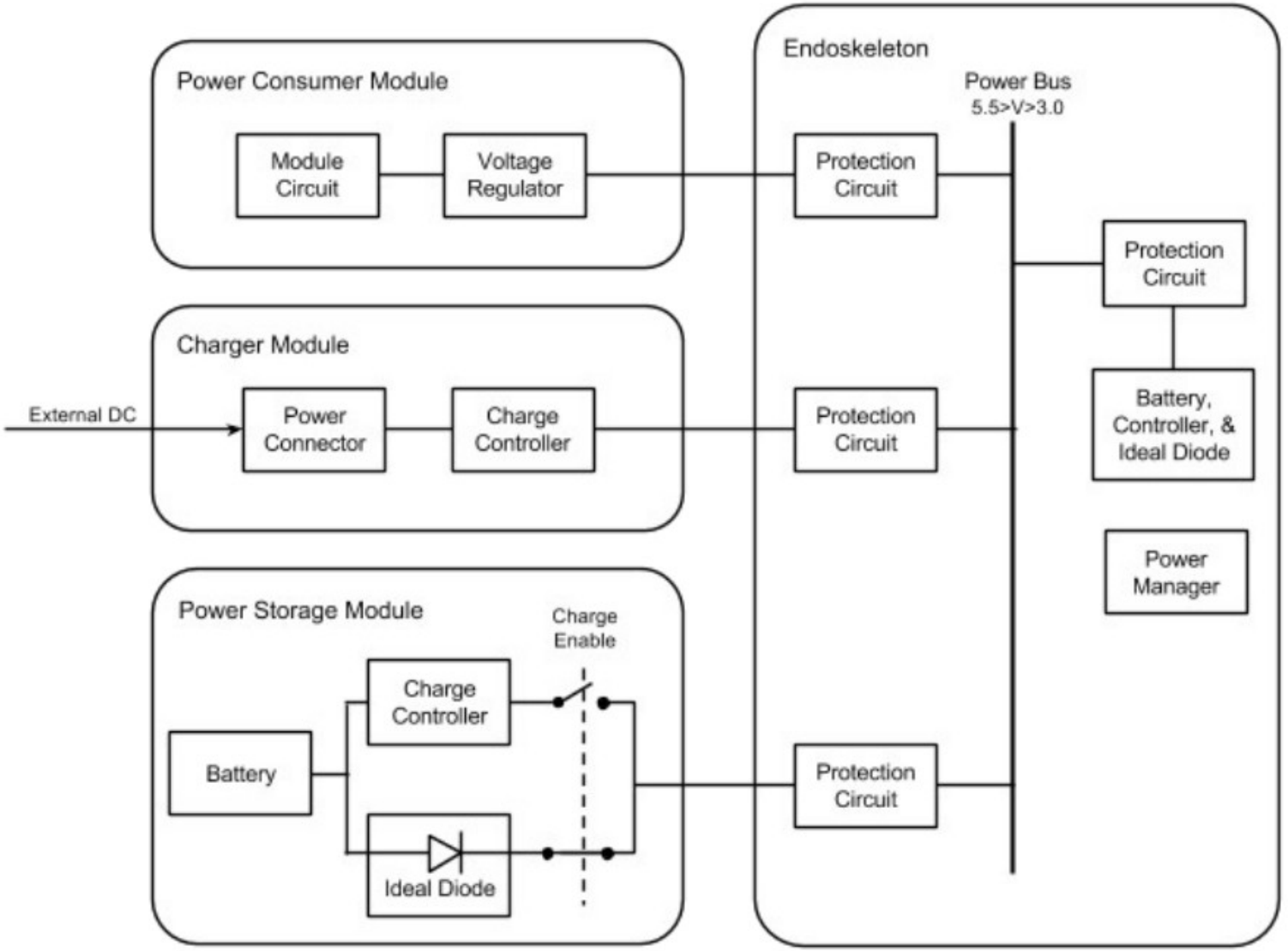


2x5

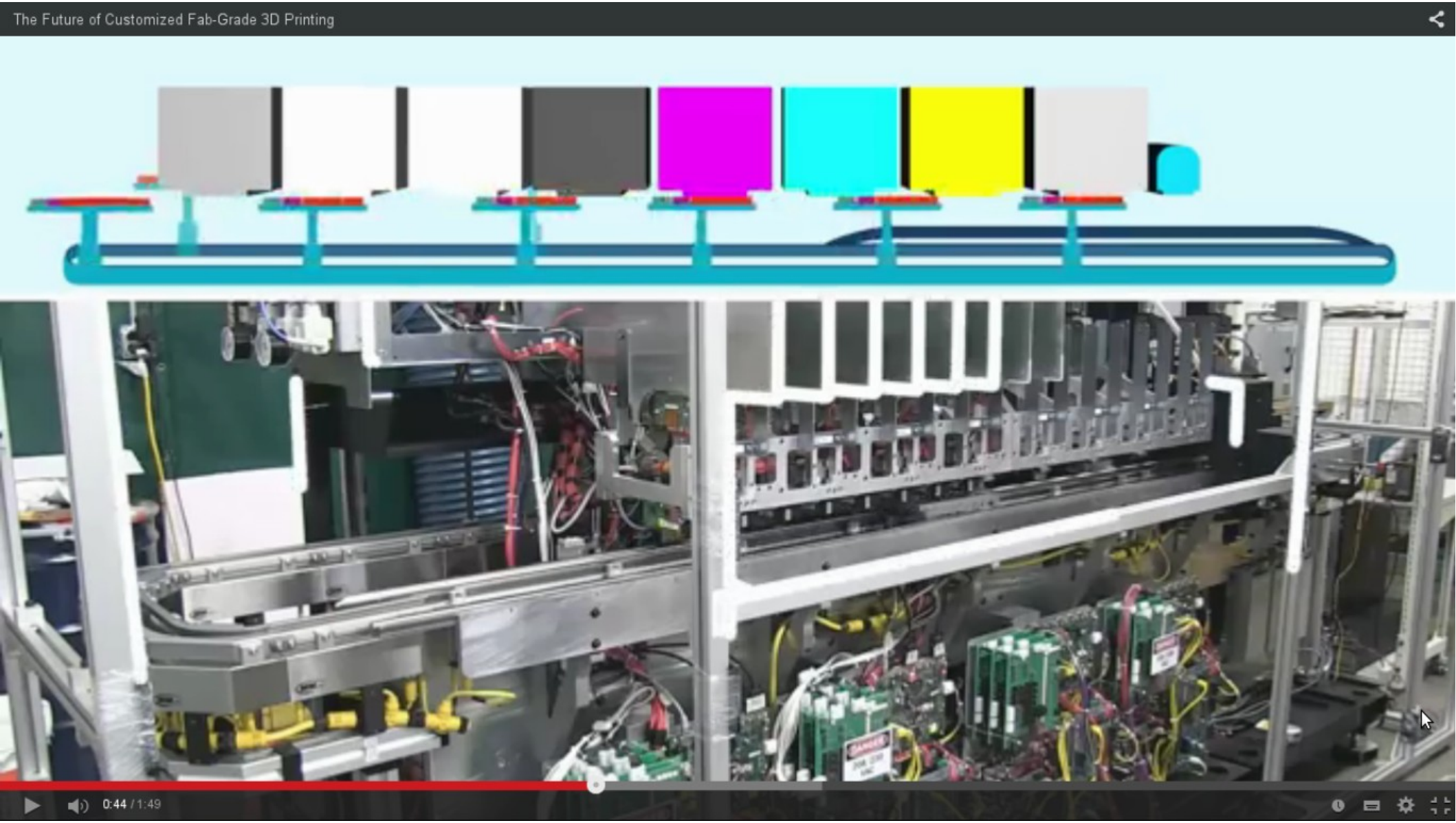
4.4. EPMs



4.5. Battery charge/recharge



4.6. Printable covers

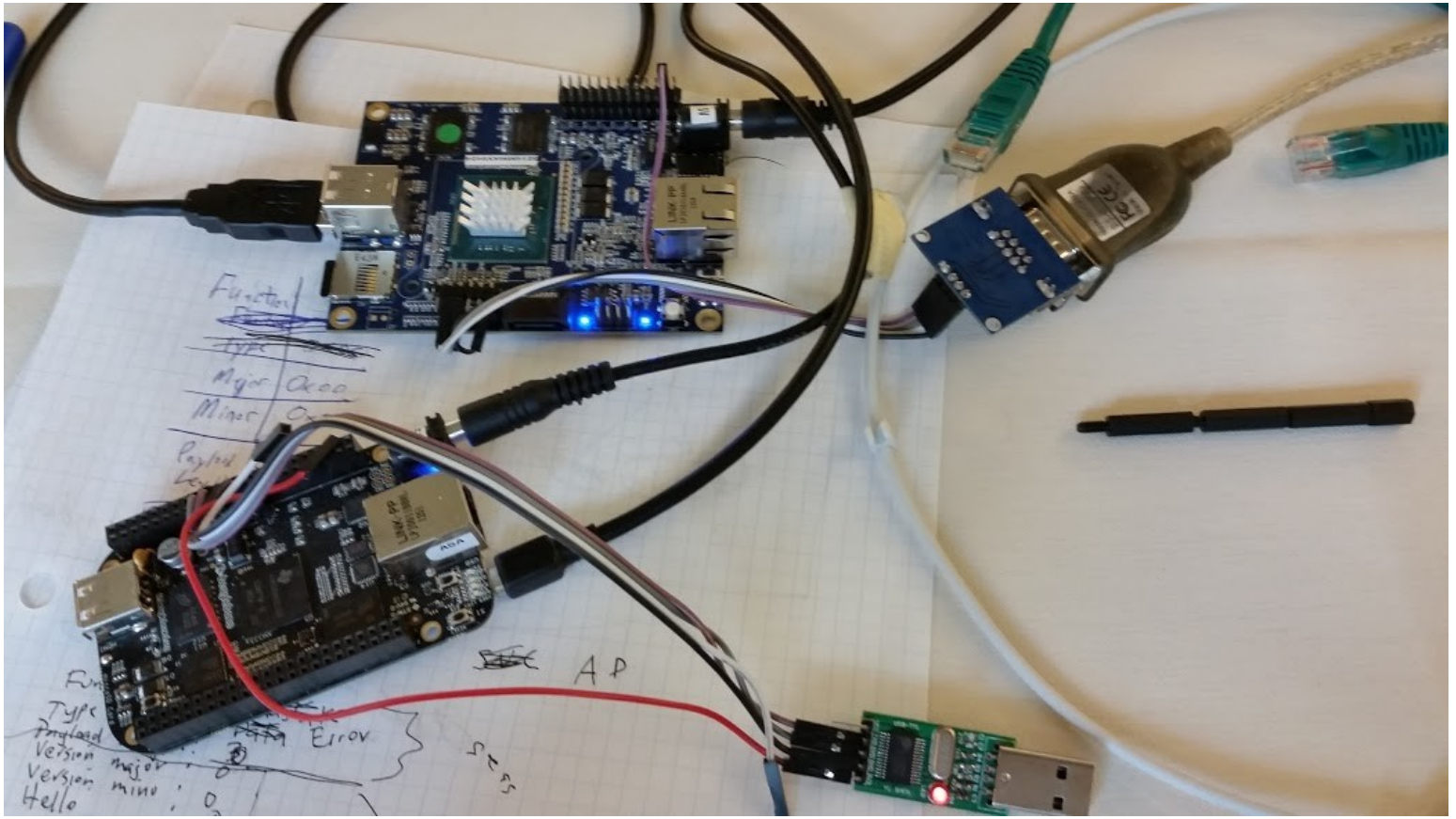


4.7. Greybus

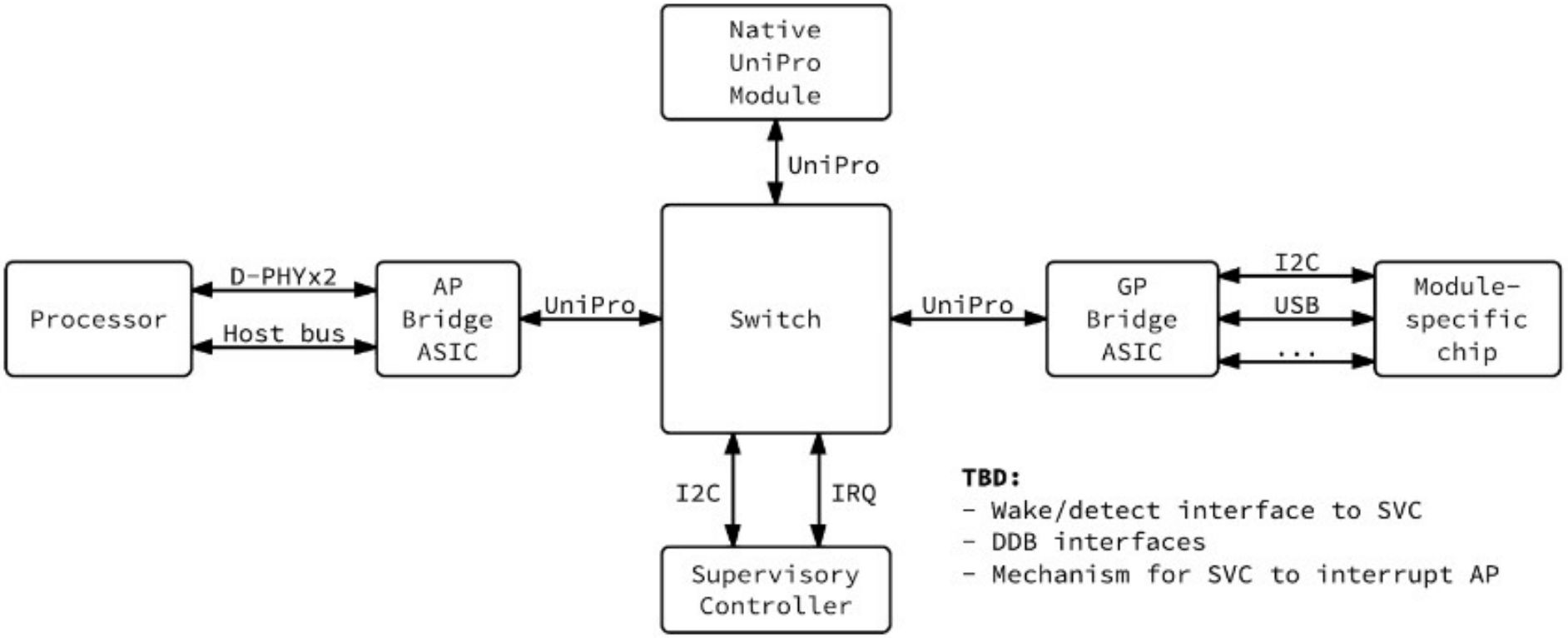
- In-kernel infrastructure for Ara
- Support for UniPro
- Support for Ara-specific interfaces:
  - Power bus / Power management
  - Hotplug support
- Communication over UniPro CPorts
- Device classes
- Bridged PHY connection protocols
- https://github.com/gregkh/greybus

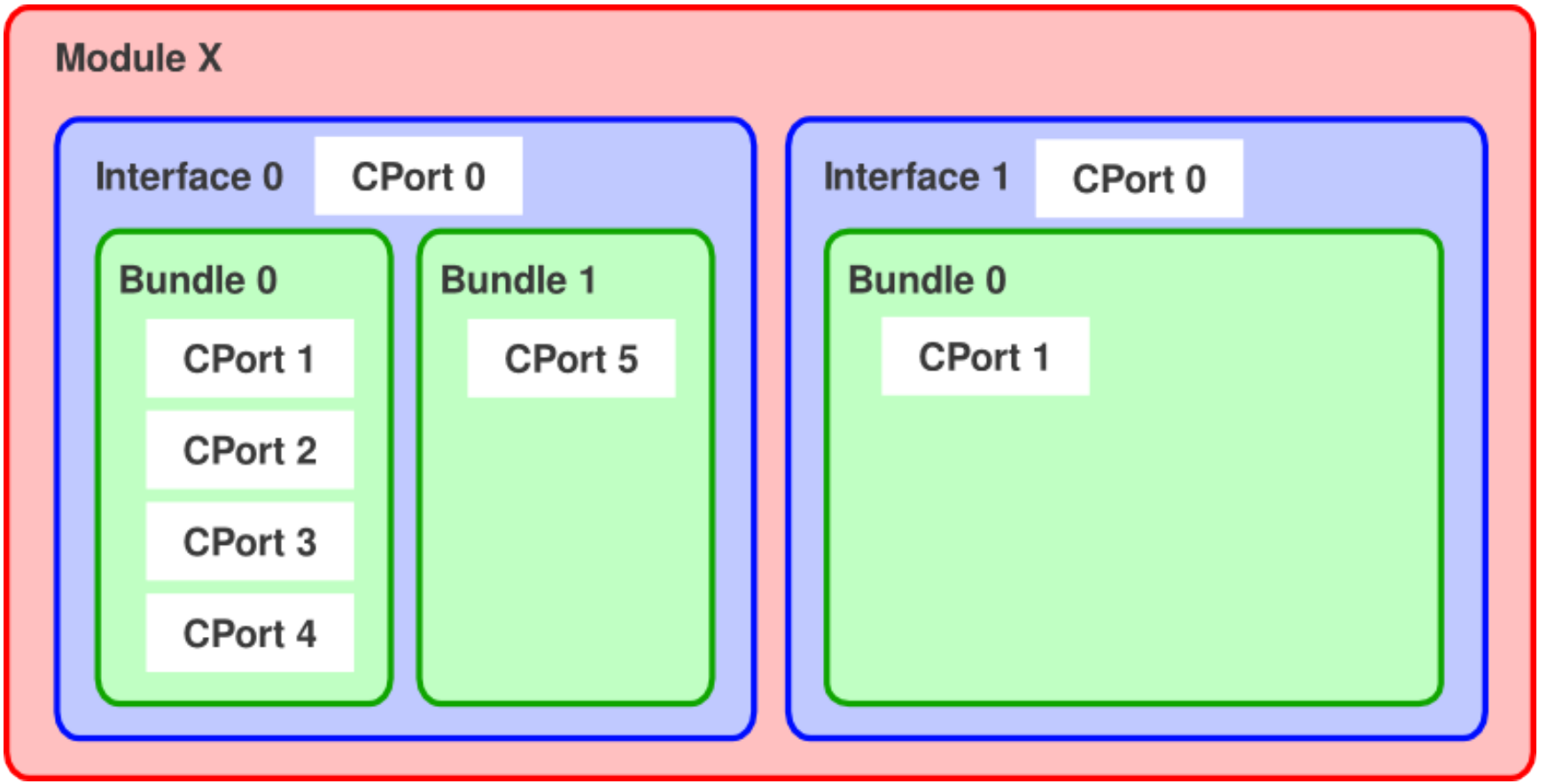
4.8. gbsim

- Greybus Simulator:
- https://github.com/ohporter/gbsim

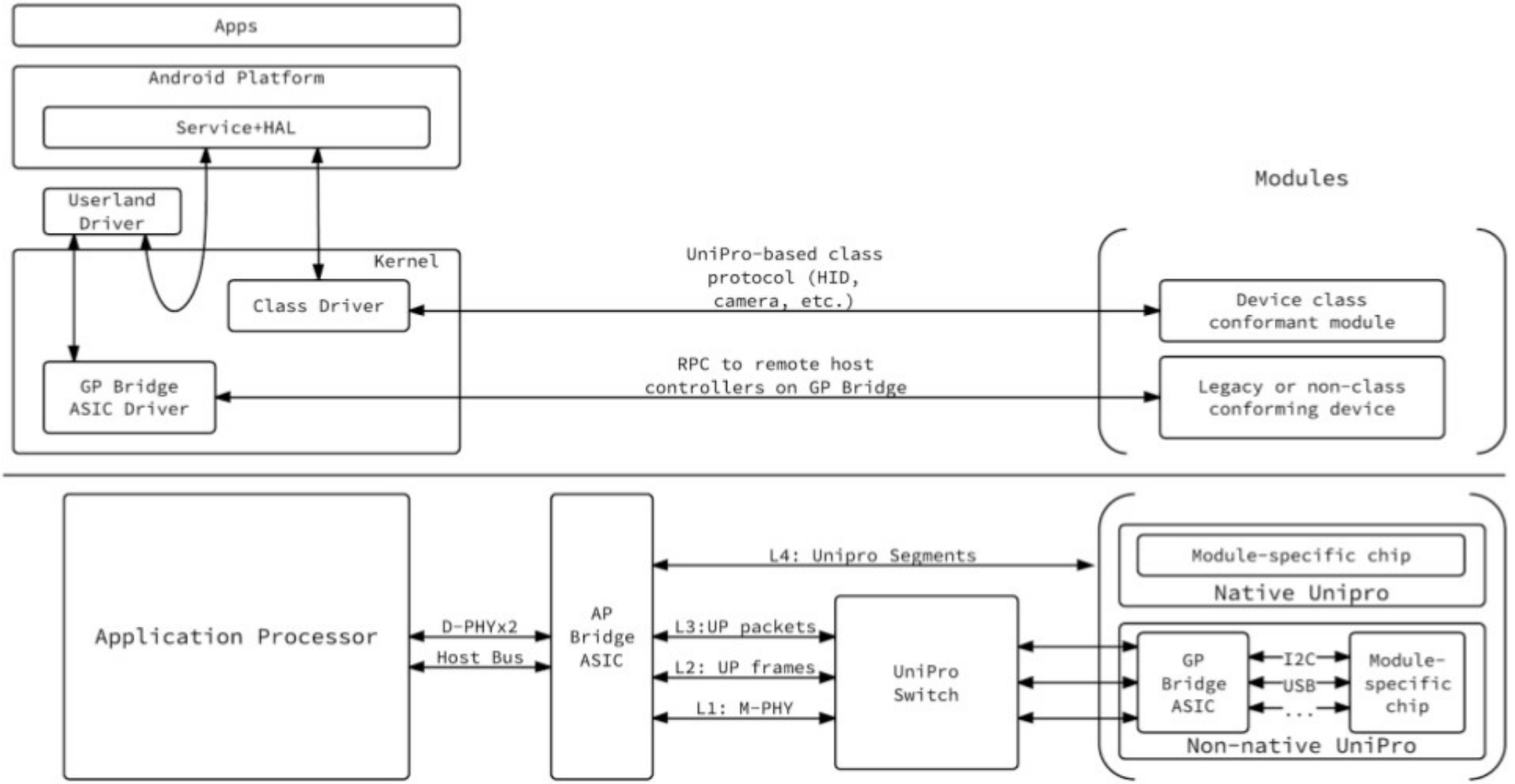


5. Hardware Architecture




Module X
Interface 0
CPort 0
Bundle 0
CPort 1
CPort 2
CPort 3
CPort 4
Bundle 1
CPort 5
Interface 1
CPort 0
Bundle 0
CPort 1

6. Software Architecture

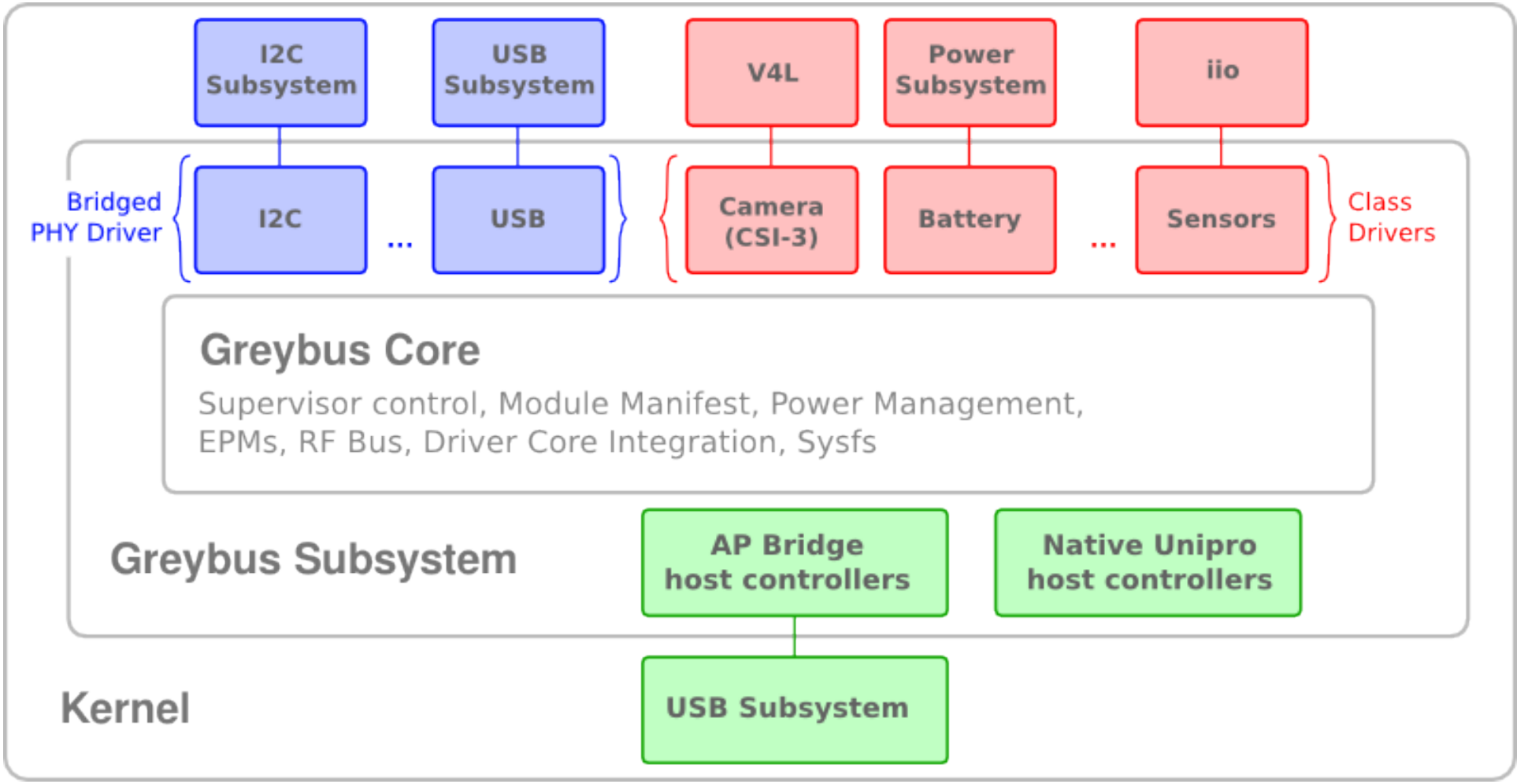


6.1. Greybus Device Classes

- Vibrator
- Battery
- Audio
- Baseband modem
- Bluetooth
- Camera
- Consumer IR
- Display
- GPS
- Keymaster
- Lights
- NFC
- Sensors
- Wifi

6.2. Bridged PHY Connection Protocols

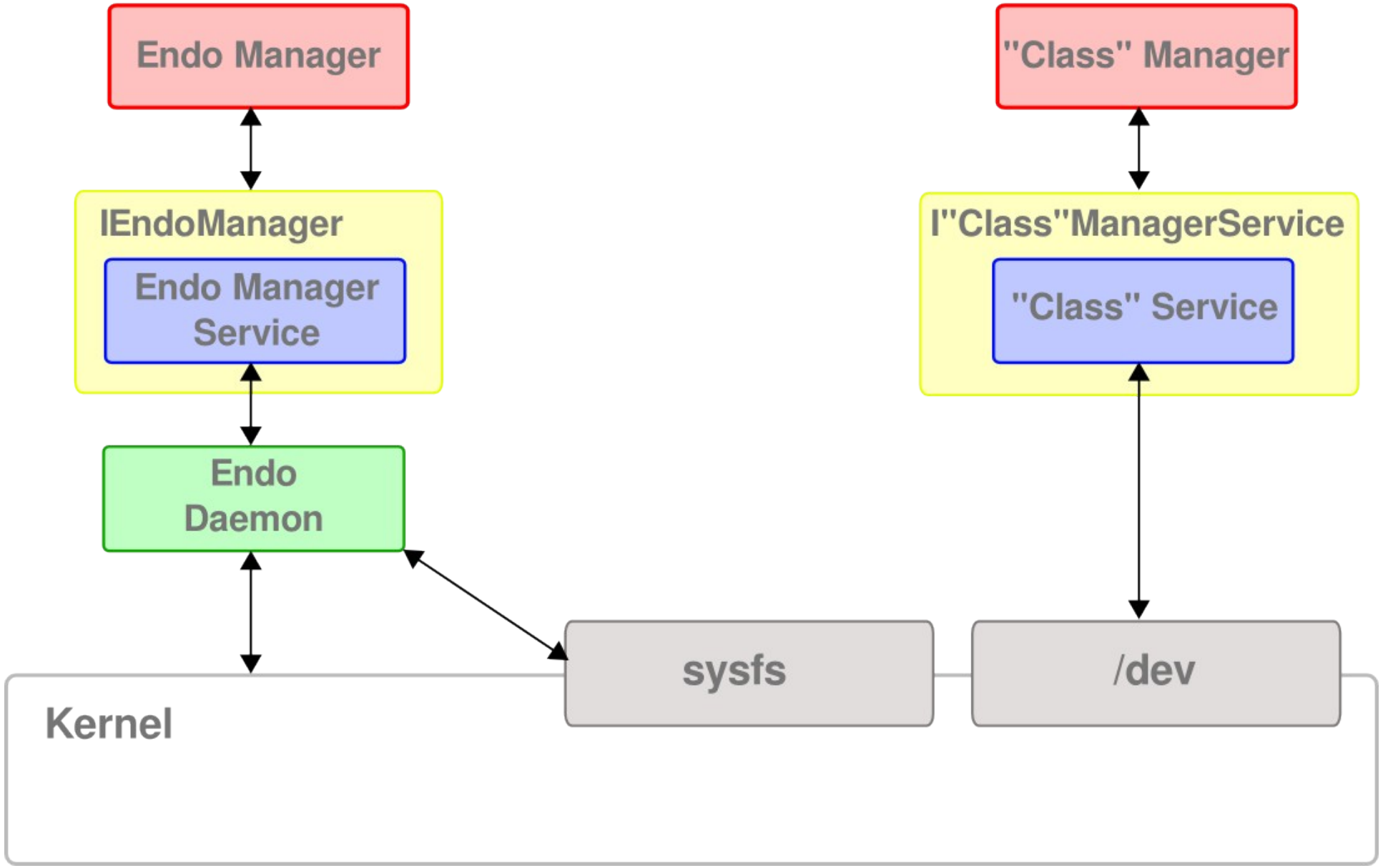
- USB
- GPIO
- SPI
- UART
- PWM
- I2C
- SDIO


I2C Subsystem
USB Subsystem
V4L
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
Kernel
USB Subsystem

6.3. Android core

- “Endo” System service
- Endo daemon
- Class-specific additions

Framework



7. Challenges

- Across the entire stack and at every step:
  - SW
  - HW
  - Manufacturing
  - Regulatory (FCC)
  - Compliance (security)

8. What's next?

Stay tuned at www.projectara.com

Thank you ...

karim.yaghmour@opersys.com