AUTOMOTIVE ETHERNET PHYS & SOFTWARE

DR. CHRISTIAN HERBER

SOFTWARE ARCHITECT ETHERNET

AMF-AUT-T2699 | JUNE 2017

SECURE CONNECTIONS
FOR A SMARTER WORLD

AGENDA

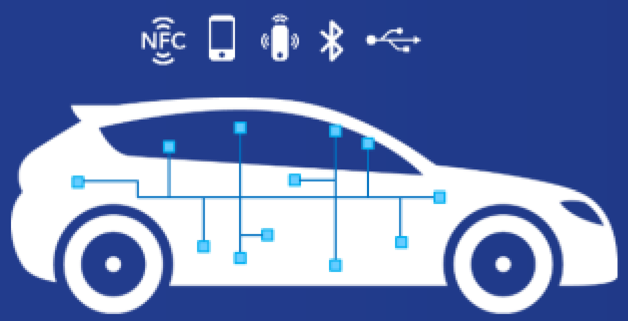
1. Introduction Automotive Ethernet
2. Generating added values through PHY SW
3. Overview of Linux’ phydev framework
4. Overview of AUTOSAR EthTrcv Driver
5. Summary

01.

Introduction Automotive Ethernet

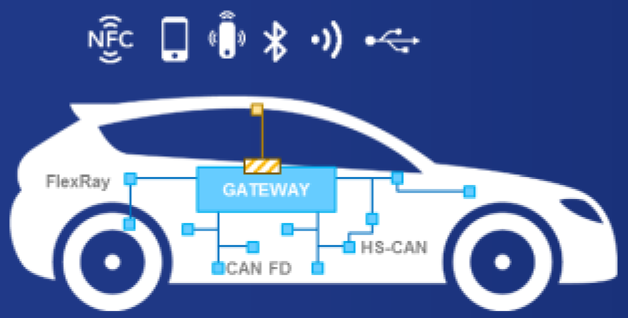
Bandwidth and Security Transform Car Networks

Today



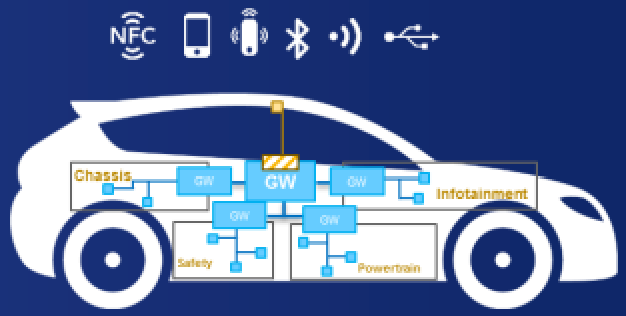
Full network exposed to attacks
(LOW protection level)

2020



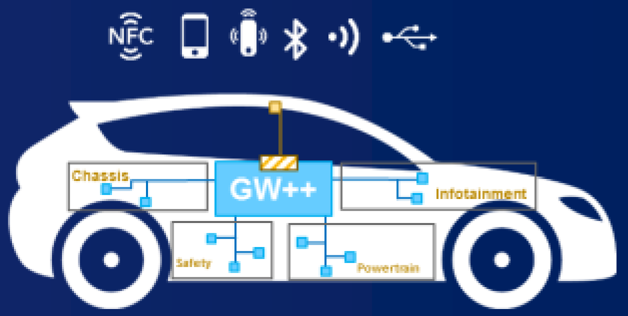
Gateway limits impact
Unprotected sub-networks
(MEDIUM protection level)

2025



Isolated systems
Domain GWs w/ controlled access
(HIGH protection level)

…after 2025



Multi-Domain Controller
Function clustering/virtualization
(full self-driving car)

IVN TODAY

- Dominated by classic CAN
- No security
- Few gateways
- Squeezed systems (bandwidth, topology, CPU, EMC)
- Simple nodes

CHALLENGES

- Major investments in network re-architecture
- Strong security not possible on CAN 2.0
- CAN FD hampered by ringing and EMC
- Lack of CAN FD and Secure MCUs
- Auto Ethernet eco-system still not mature
- Ensure the transition remains manageable

IVN TOMORROW

- CAN FD, Ethernet and more
- IDS and Crypto security
- Central and Domain gateways
- Tighter EMC specs
- Wider topology range
- Smart nodes

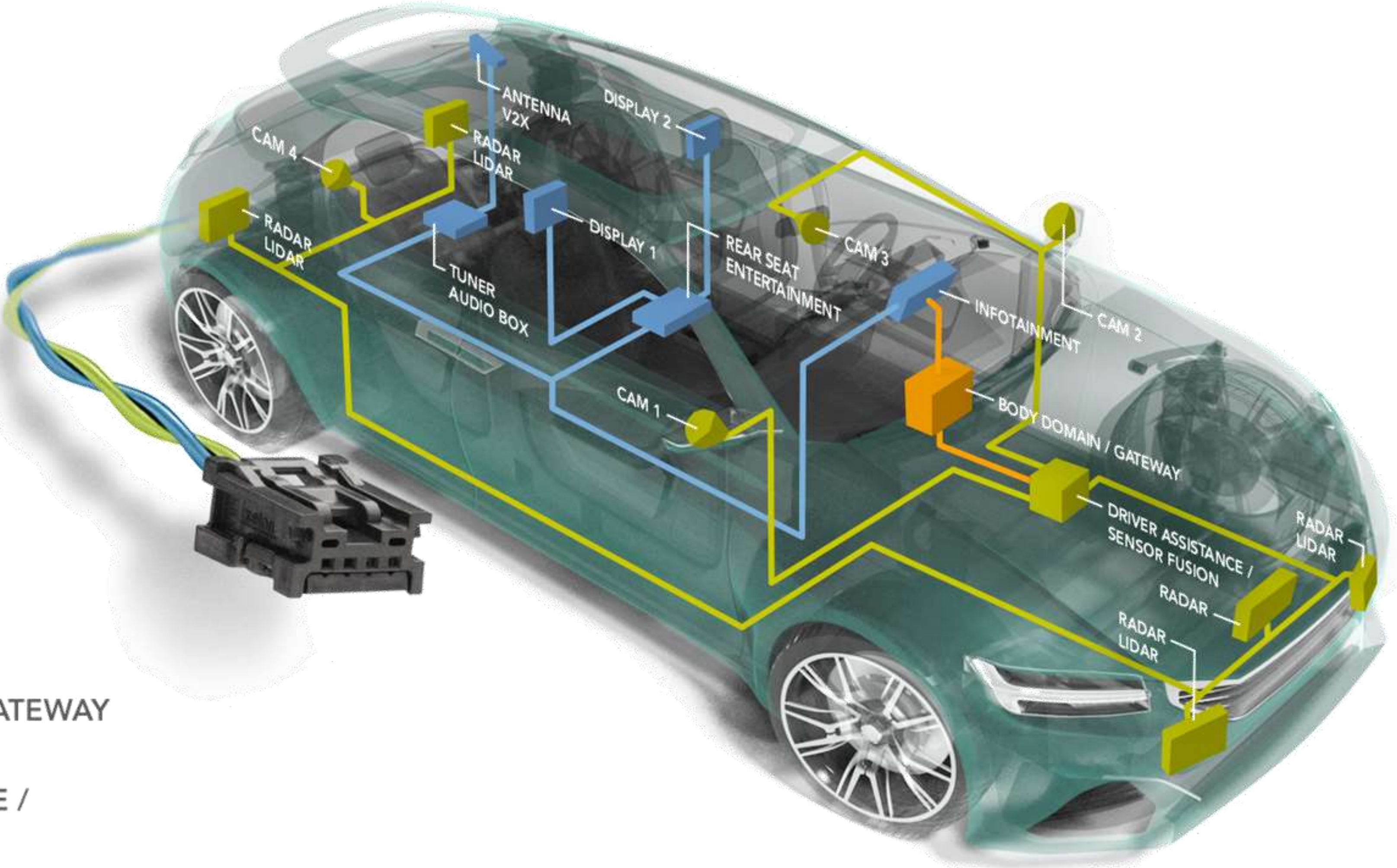
Introducing Ethernet: NXP Provides Auto-Native Portfolio
Flexible, Scalable Solution

CO-FOUNDER

MEMBER RTPGE
Reduced Twisted Pair
Gigabit Ethernet

PROMOTER

REGULAR



🟦 INFOTAINMENT

🟧 BODY DOMAIN / GATEWAY

🟨 DRIVER ASSISTANCE / SENSOR FUSION

ETHERNET BACKBONE

June 23, 2017

NXP Ethernet Portfolio: The Auto-Native Portfolio

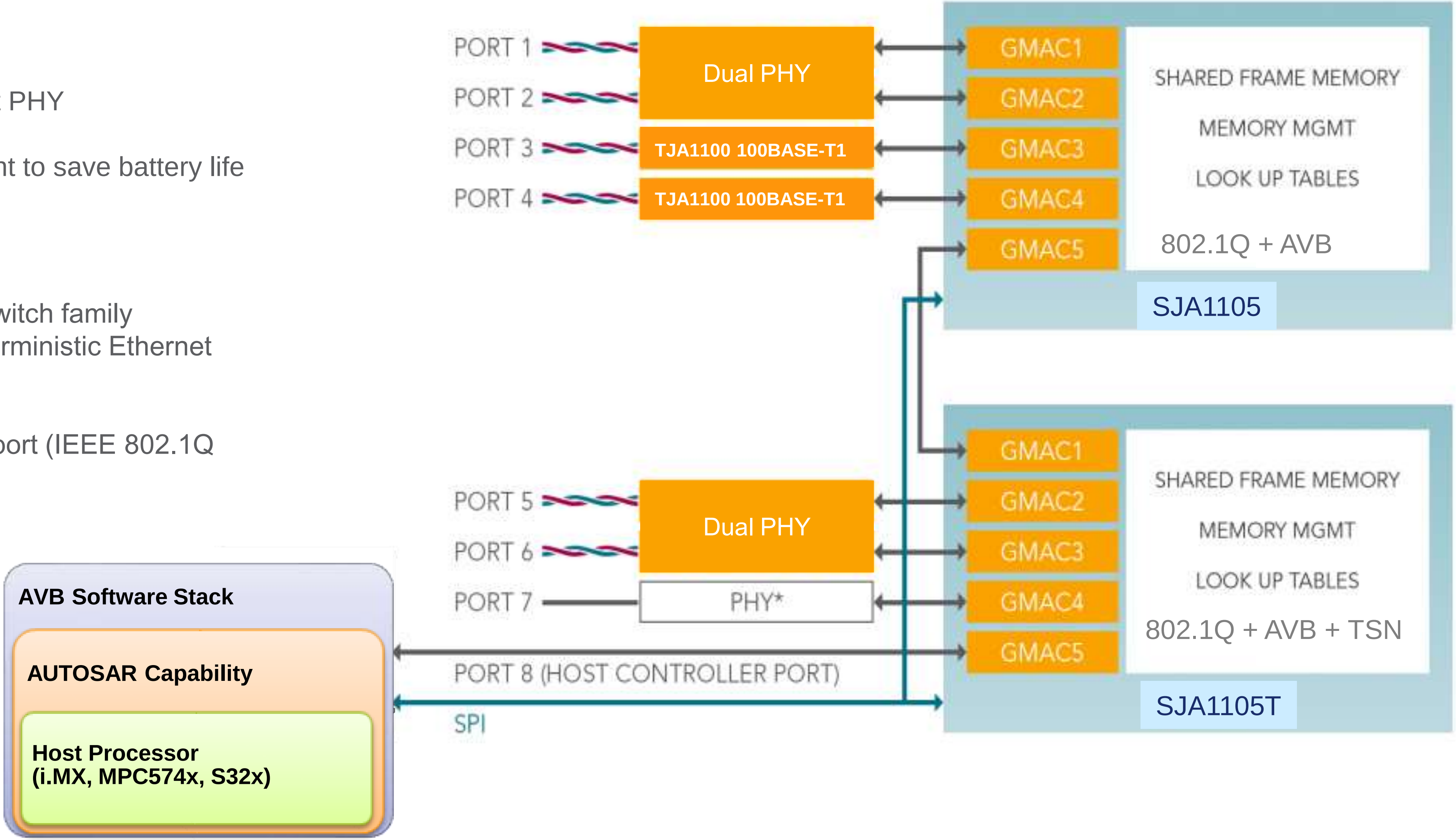
Flexible, Scalable Solution

TJA1100

- IEEE 100BASE-T1 Compliant PHY
- Fully automotive qualified
- Enhanced Power Management to save battery life

SJA1105(T)

- Layer 2 Store and Forward Switch family
- Supports AVB, TSN and Deterministic Ethernet
- 10/100/1000 Mbps interfaces
- MII/RMII/RGMII Interface
- Port Mirroring and VLAN support (IEEE 802.1Q and IEEE 802.1P)



NXP CONFIDENTIAL

02.

Generating added values through PHY SW

Network management, monitoring/diagnosis, safety, security

Motivation

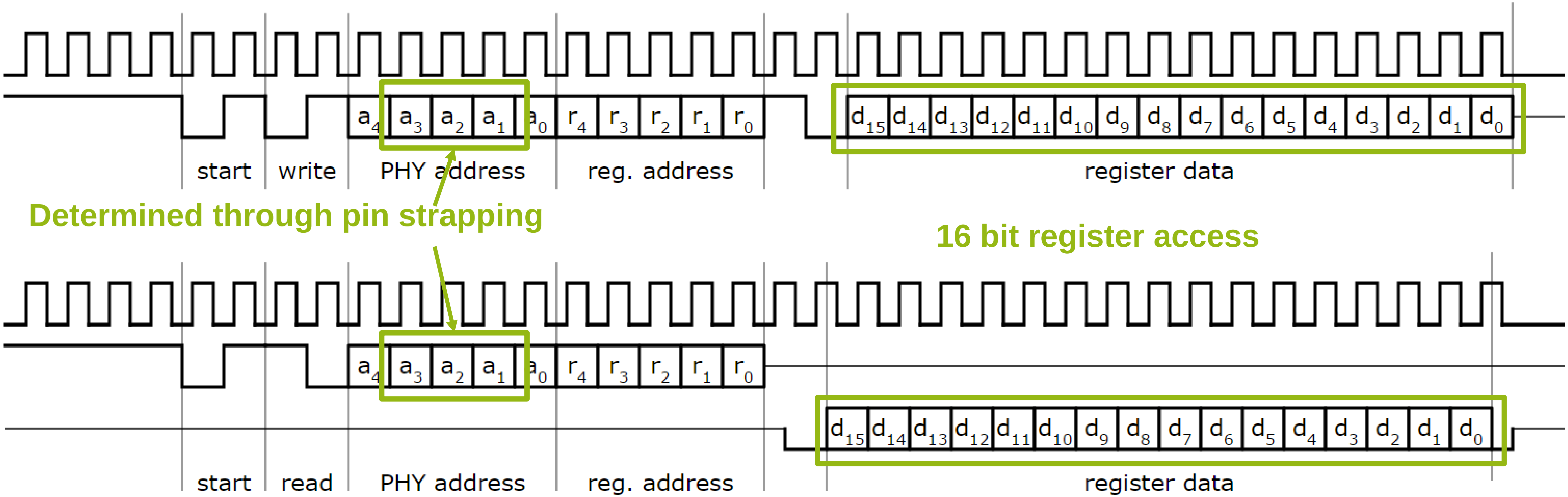
- An Ethernet PHY is functional without any software, so why bother?

- Tasks related to network management, diagnosis, fault handling require SW involvement
- Added values can be generated through e.g.:
  - Reduced wiring cost by leveraging advanced network management like wake/sleep
  - Fault prevention/ageing detection through SNR, symbol errors, cable test
  - Enablement of ASIL x functional safety designs
  - Detection of tampering for enhanced security, e.g. for PNAC

How to configure/control TJA1100

- Reset and enable pin, wake-up pin
- Pin strapping (Resistor strapping)
  - Master/Slave + enable/disable of PHY
  - Autonomous mode/managed mode
  - PHY Address (used for MDIO access), three bits configurable
- Register read/write through MDIO bus
  - Status information
  - Link control, loopback modes, test modes
  - Sleep/wake

Management Data Input/Output (MDIO) Serial Management Interface (SMI)



Determined through pin strapping

16 bit register access

PHY Diagnosis

Permanent

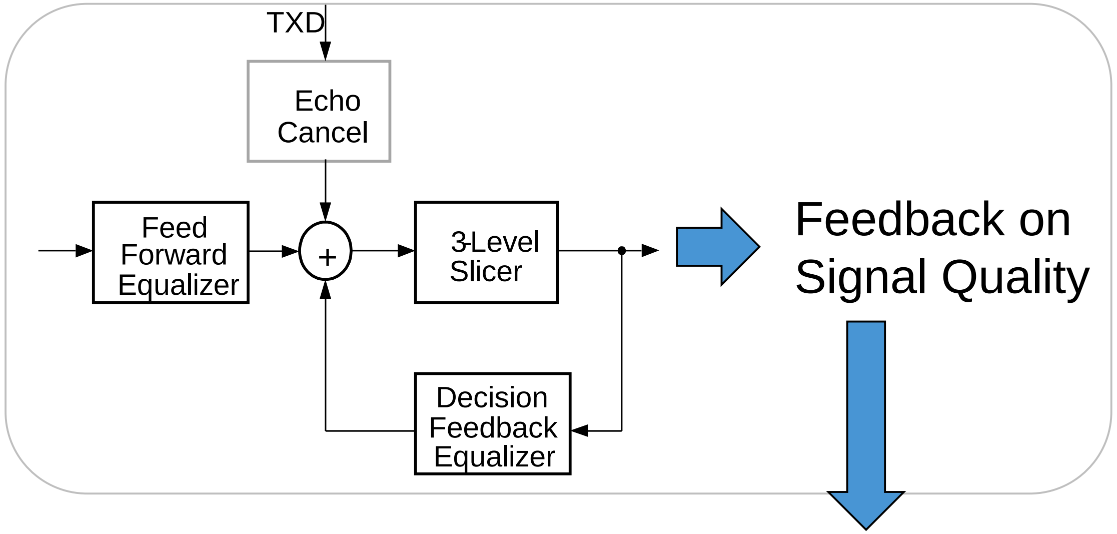
- Link up information
  - Scrambler/Clock sync/PLL… feedback
  - Receiver status (local and remote)
- Feedback from Equalizer – Signal Quality + Warning when limits are exceeded
- Symbol error detection
- Under-Voltage and Over-Temperature status

On Request

e.g. during start-up or if link failure detected

- Cable diagnostics: Open/Short detection
  - e.g. during start-up or if link failure detected
- Loop back modes – to check integrity along data flow
  - Internal, external, remote (see next slides)

- Usage of additional PHY feedback for channel quality under discussion with OEMs

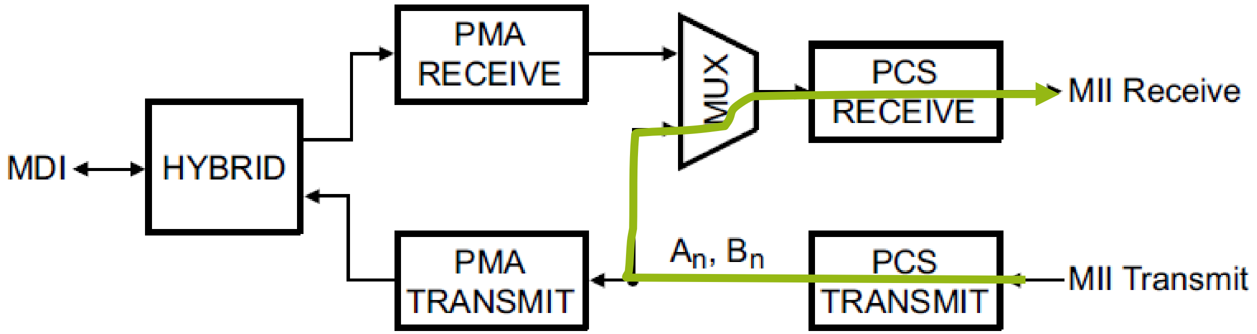


Feedback on Signal Quality

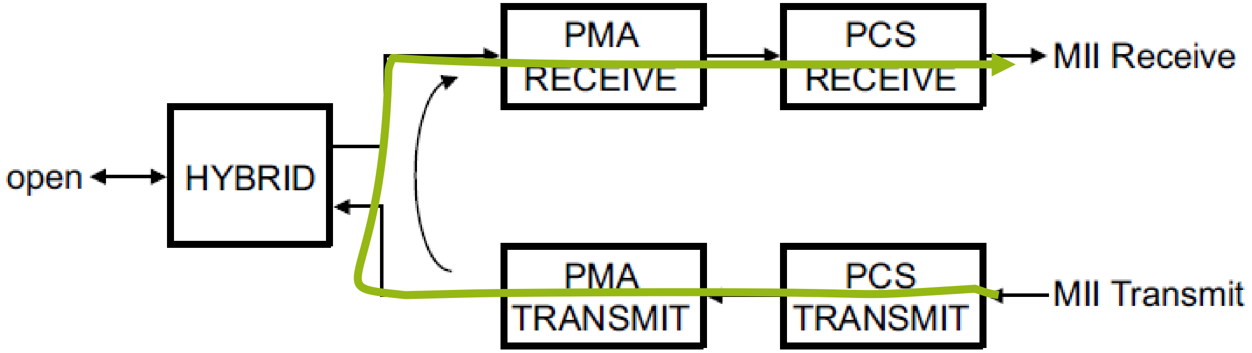
000 worse than class A (unstable link)
001 class A (unstable link)
010 class B (unstable link)
011 class C (unstable link)
100 class D (poor link; potential bit error)
101 class E (good link)
110 class F (very good link)
111 class G (very good link)

Loopback Modes (1)

Internal Loopback



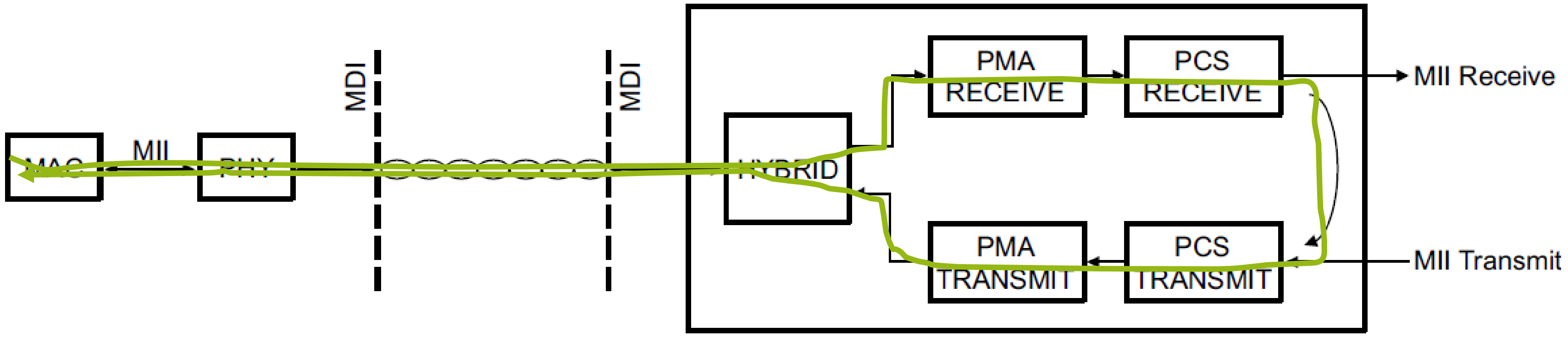
External Loopback



- Both will loop back traffic to the host connected over MII
- No physical medium needs to be attached
- Can be used for diagnosis of PHY and for (software) testing

Loopback Modes (2)

Remote Loopback



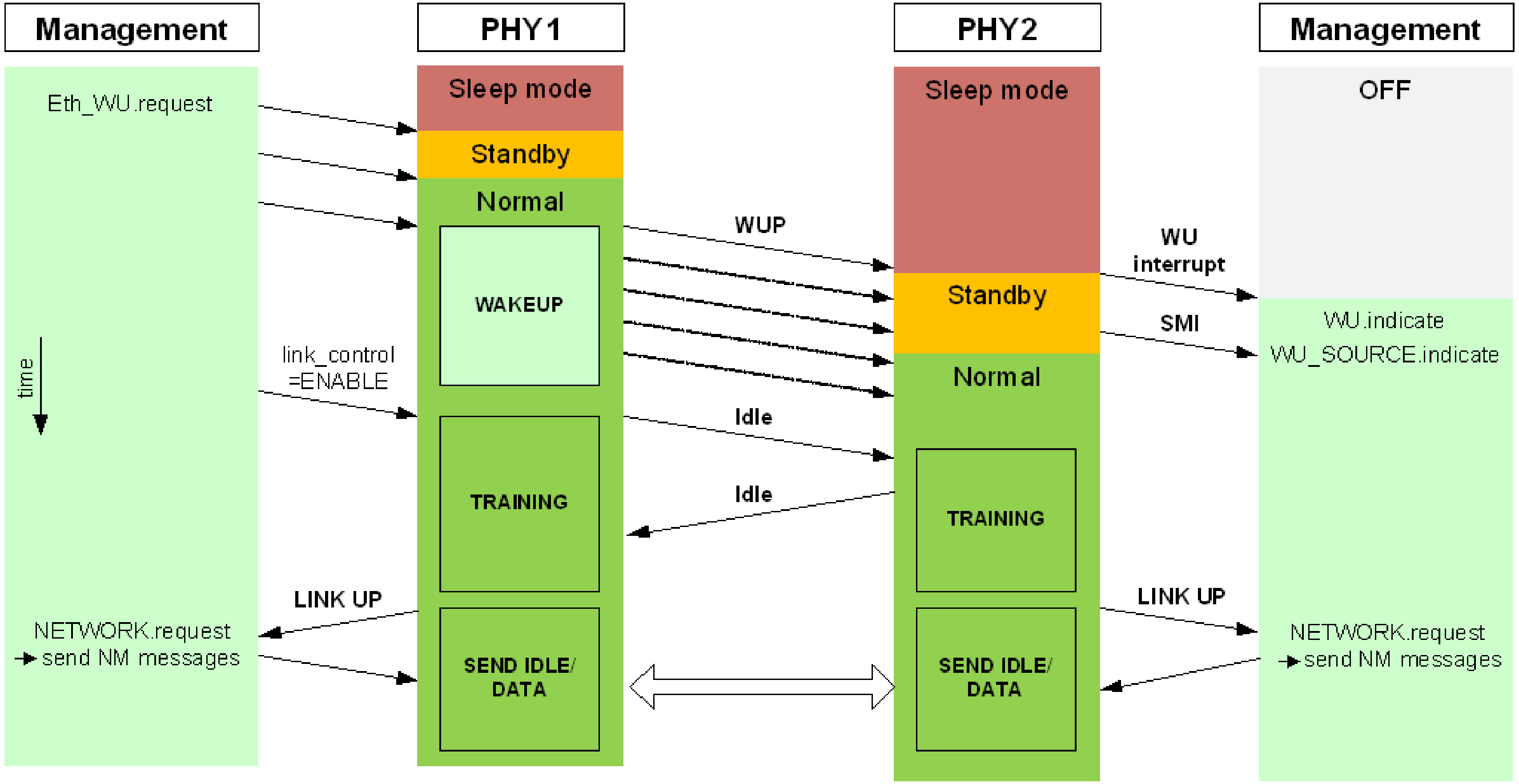
- Complete diagnosis of PHY and cable
- Requires a second PHY to be connected via a cable

Sleep/Wake-Up: Objectives

- Enables partial networking
- No dedicated wake-up line needed
- Node/cluster wakeup within less than 250 ms
  - No microcontroller involvement in forwarding wake requests
- Sleep current consumption per port less than 10 µA
  - PHYs are directly powered through battery supply V_BAT

→ Standardized in Open Alliance TC10, moving to ISO standard

Sleep/Wake States



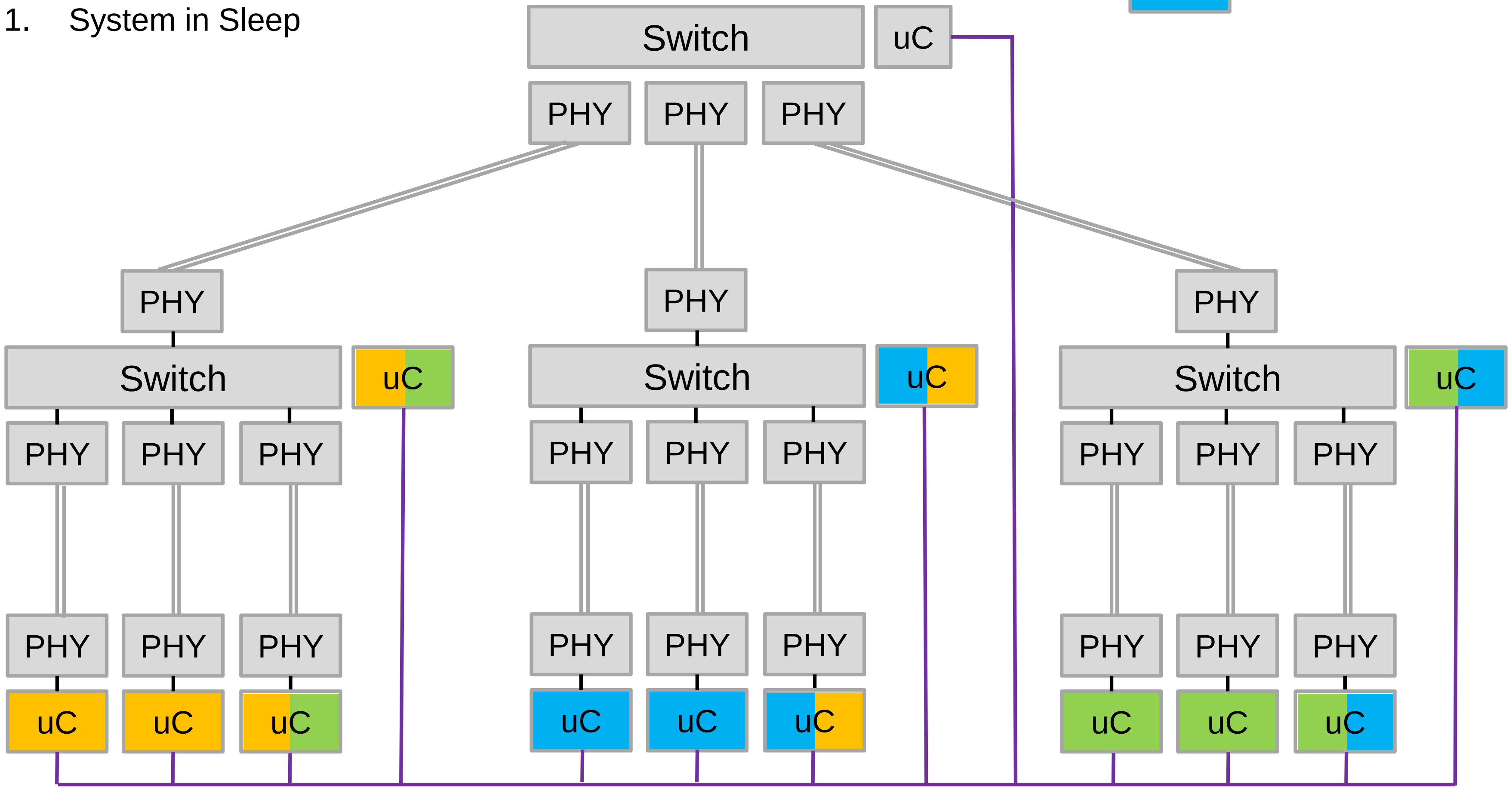
Global Wake-up with Activation Line

uC Funct. Cluster A

uC Funct. Cluster B

uC Funct. Cluster C

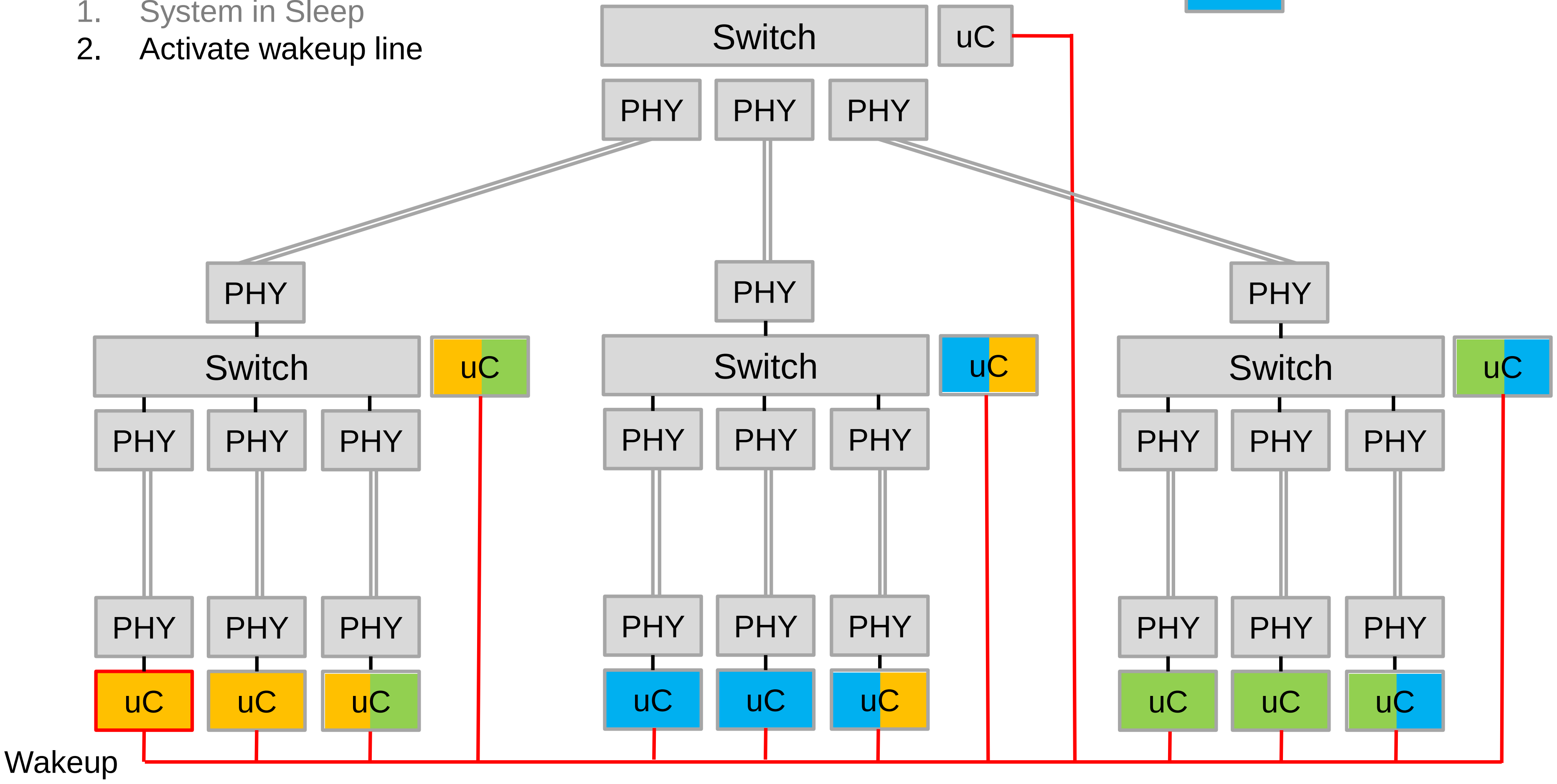
1. System in Sleep



Global Wake-up with Activation Line

uC 🟨	Funct. Cluster A
uC 🟩	Funct. Cluster B
uC 🟦	Funct. Cluster C

1. System in Sleep
2. Activate wakeup line



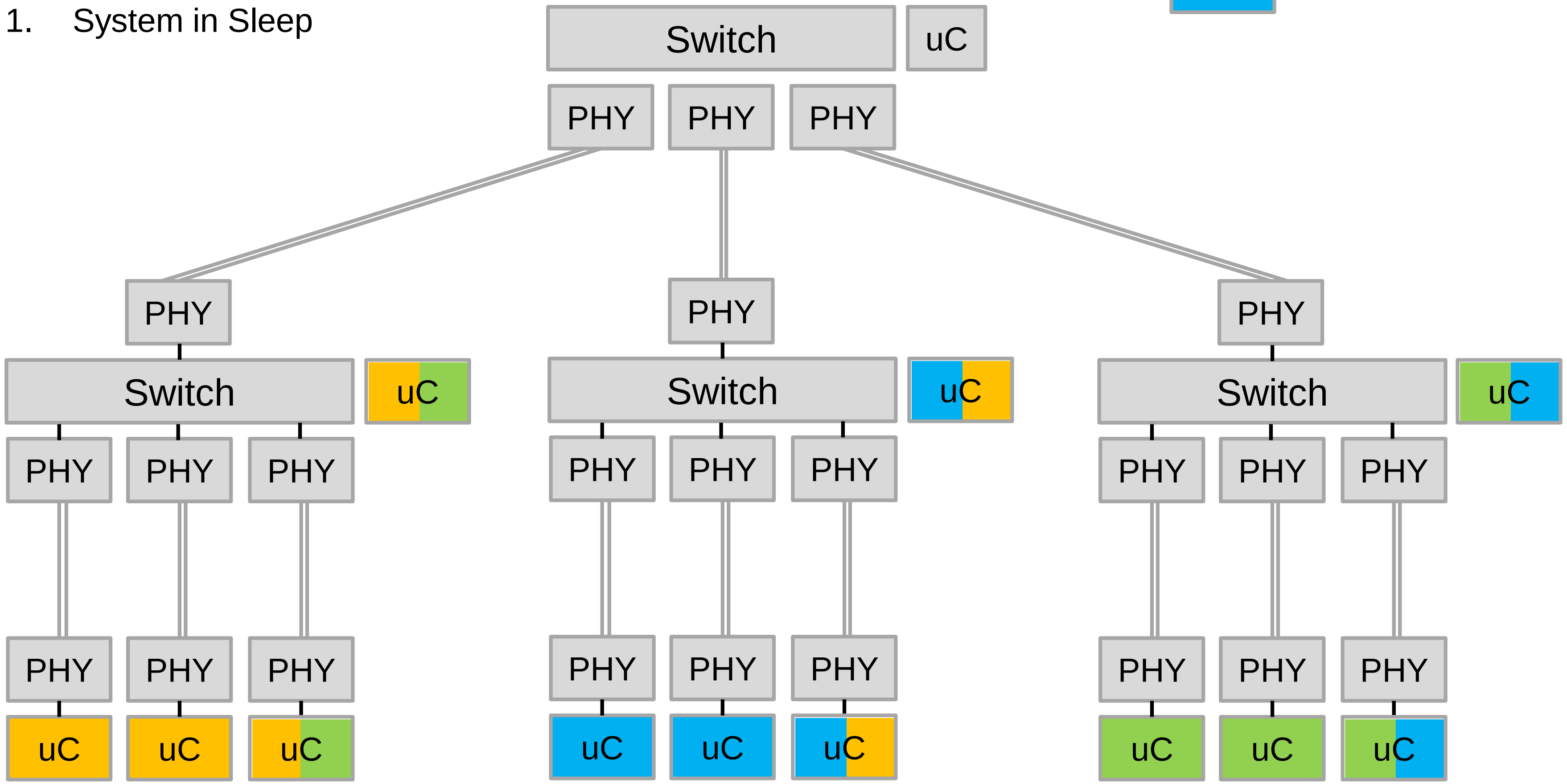
Wake-up over Ethernet

🟨 uC Funct. Cluster A

🟩 uC Funct. Cluster B

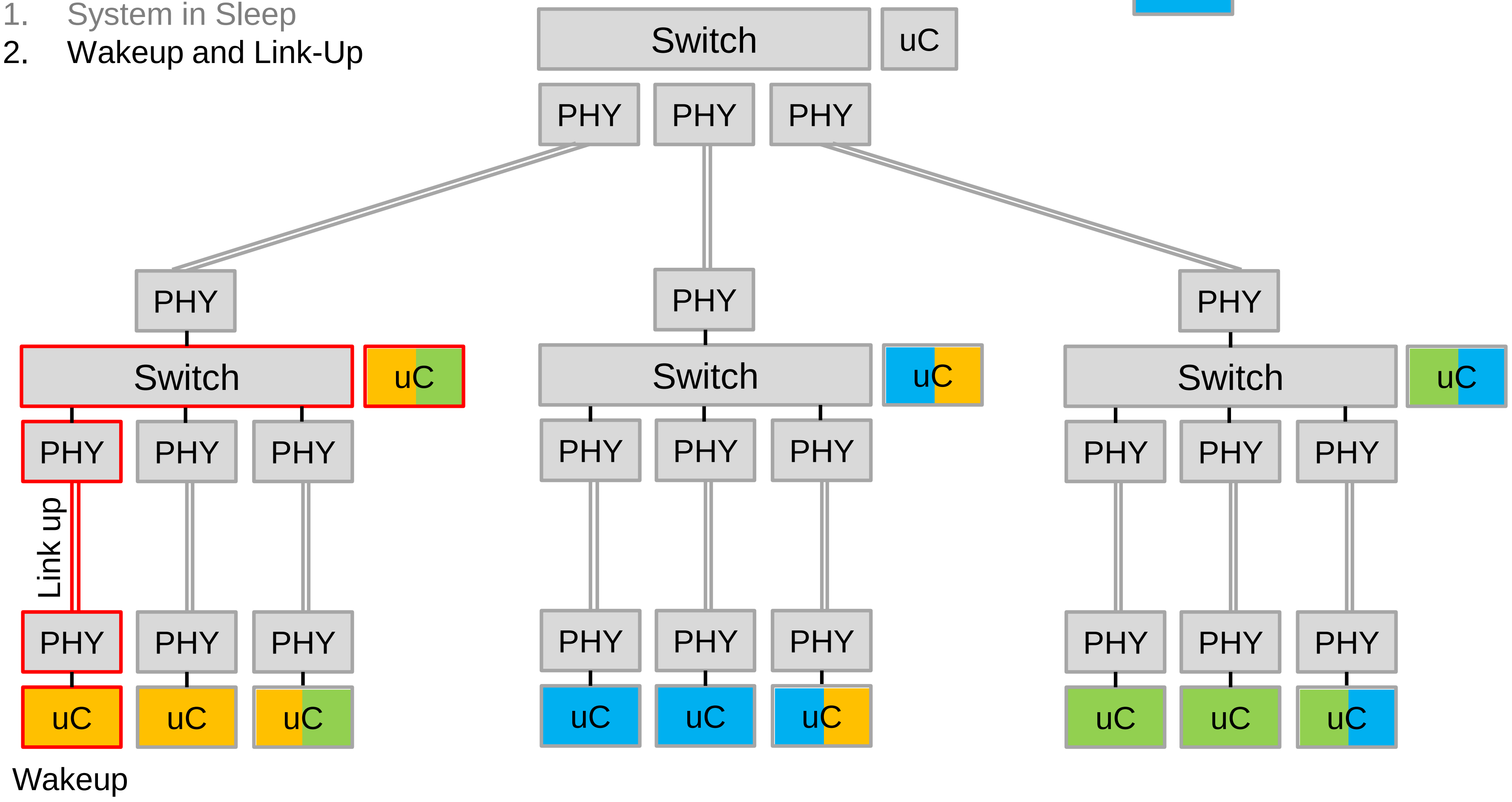
🟦 uC Funct. Cluster C

1. System in Sleep



Wake-up over Ethernet

1. System in Sleep
2. Wakeup and Link-Up

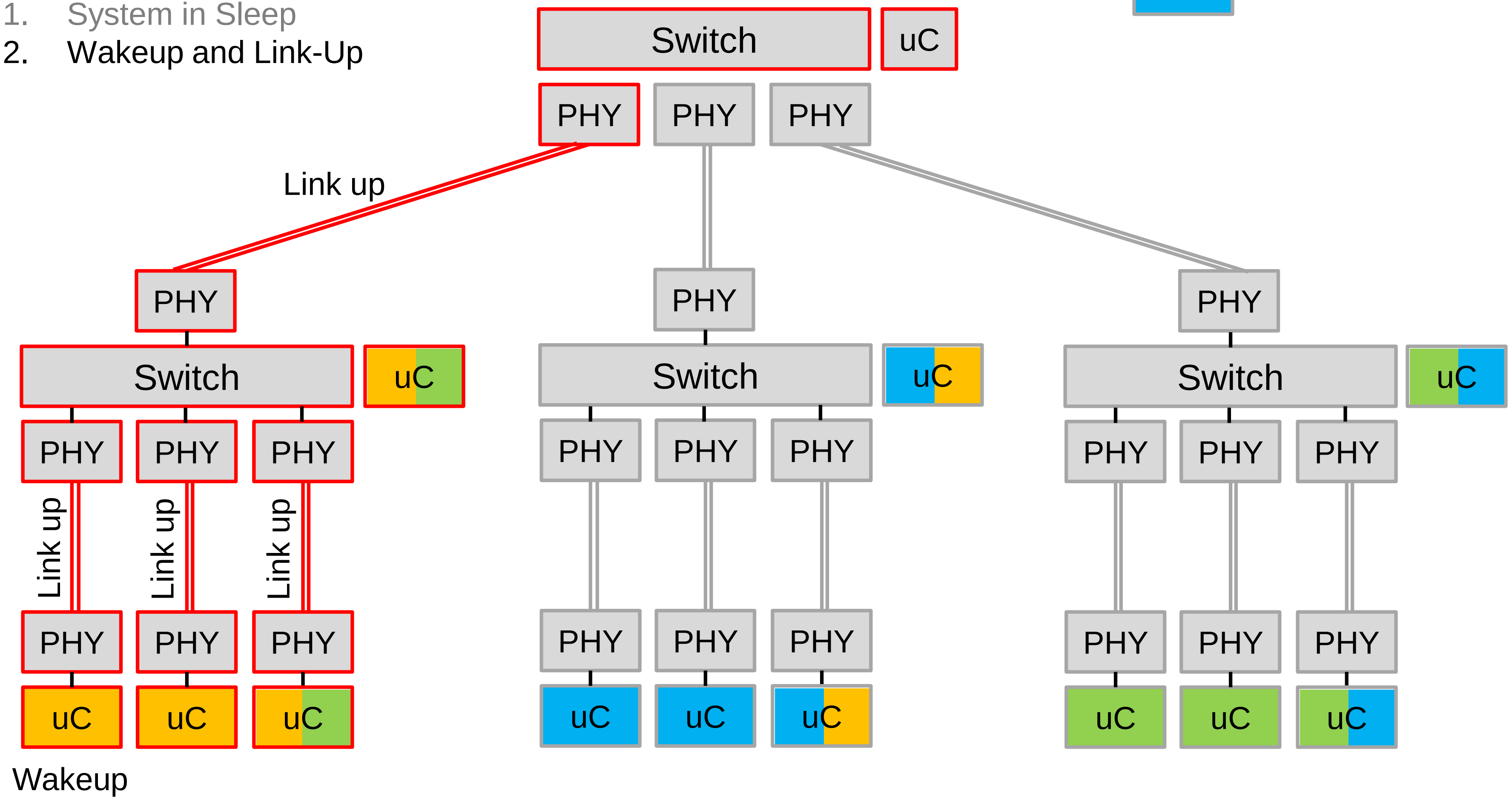


uC	Funct. Cluster A
uC	Funct. Cluster B
uC	Funct. Cluster C

Wake-up over Ethernet

uC	Funct. Cluster A
uC	Funct. Cluster B
uC	Funct. Cluster C

1. System in Sleep
2. Wakeup and Link-Up



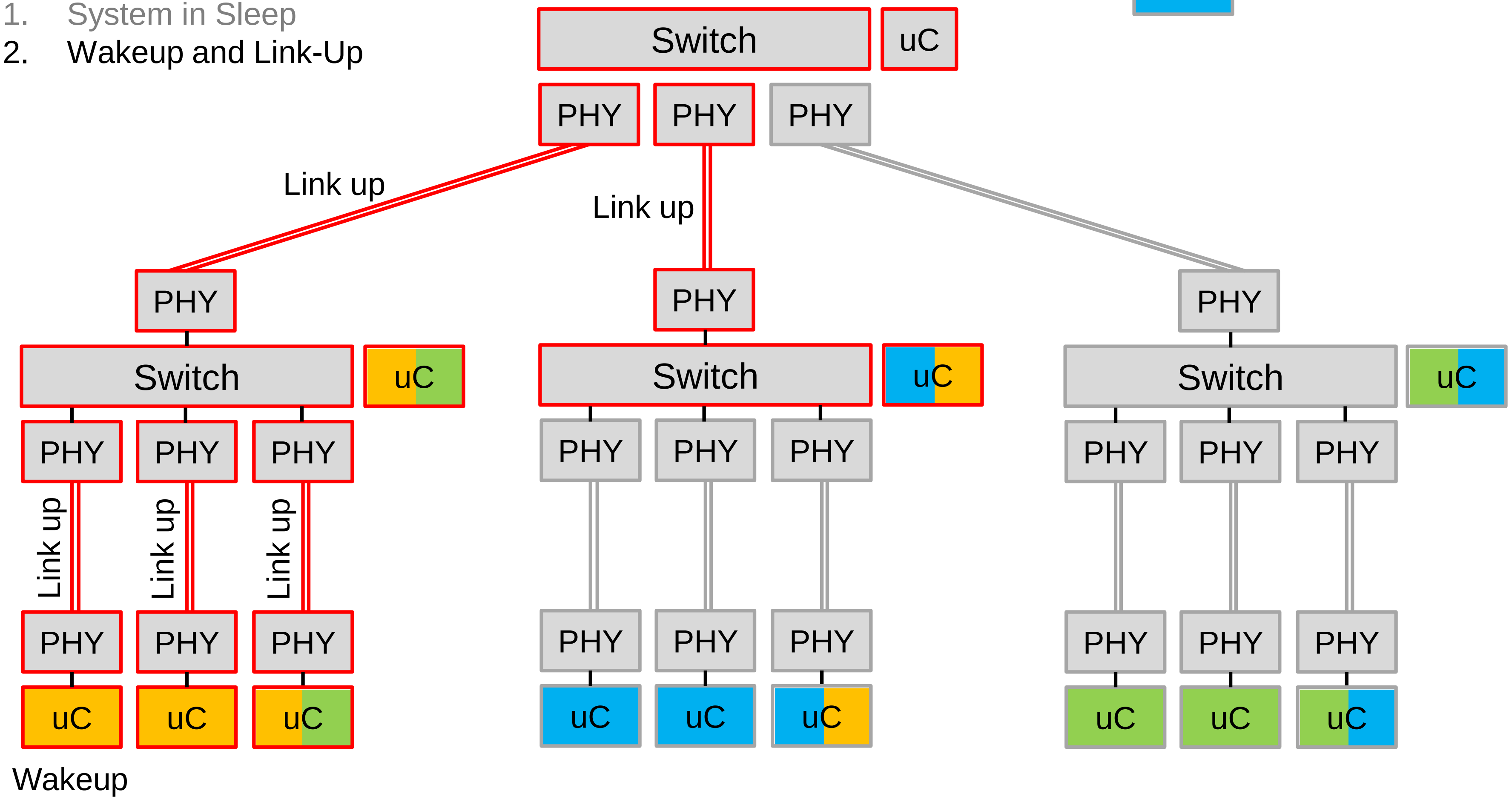
Wake-up over Ethernet

🟨 uC Funct. Cluster A

🟩 uC Funct. Cluster B

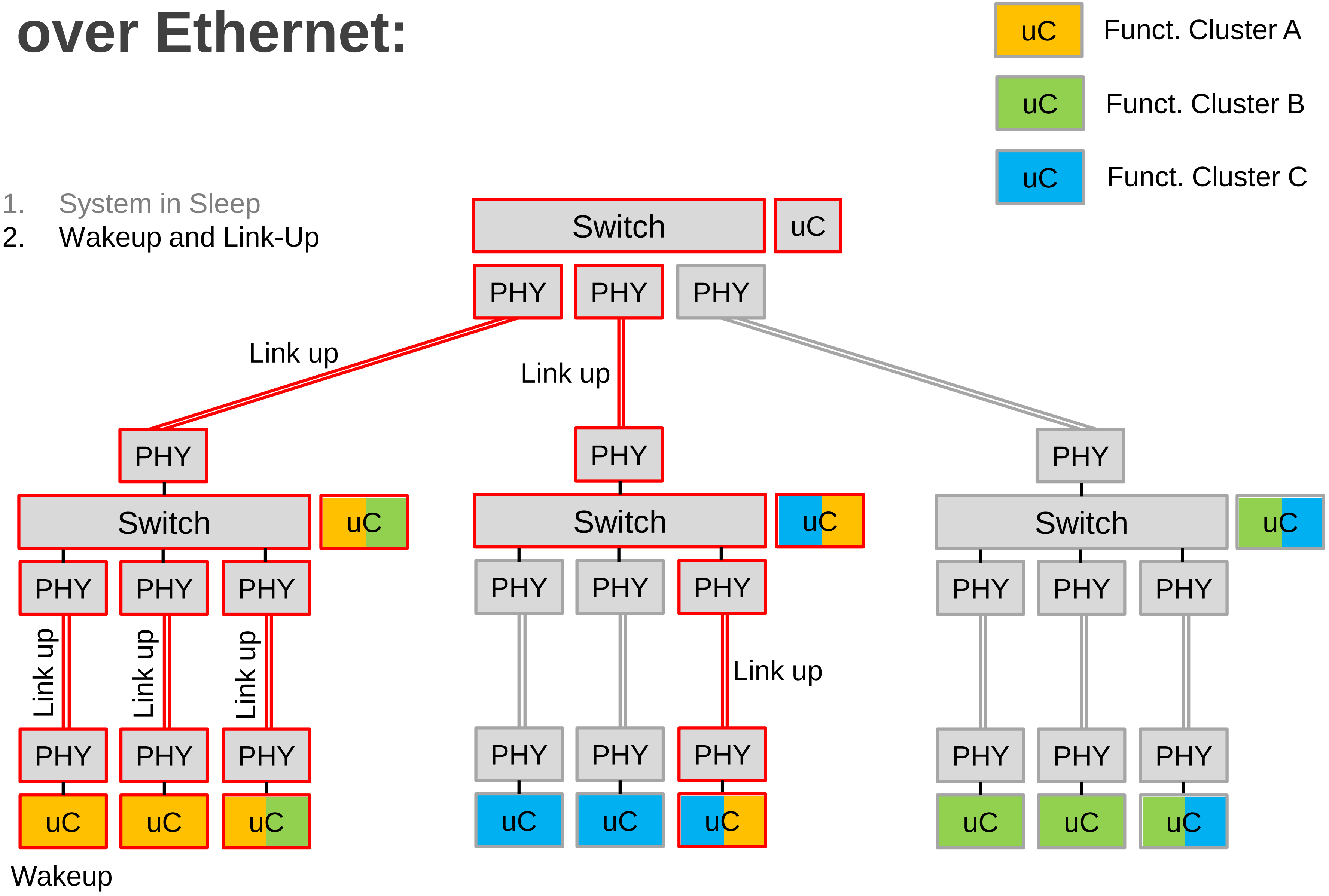
🟦 uC Funct. Cluster C

1. System in Sleep
2. Wakeup and Link-Up

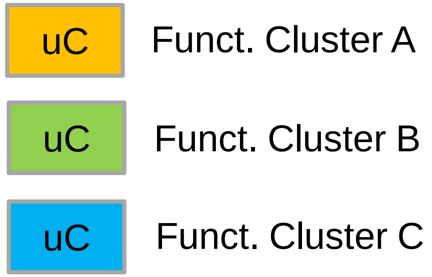


Wake-up over Ethernet:

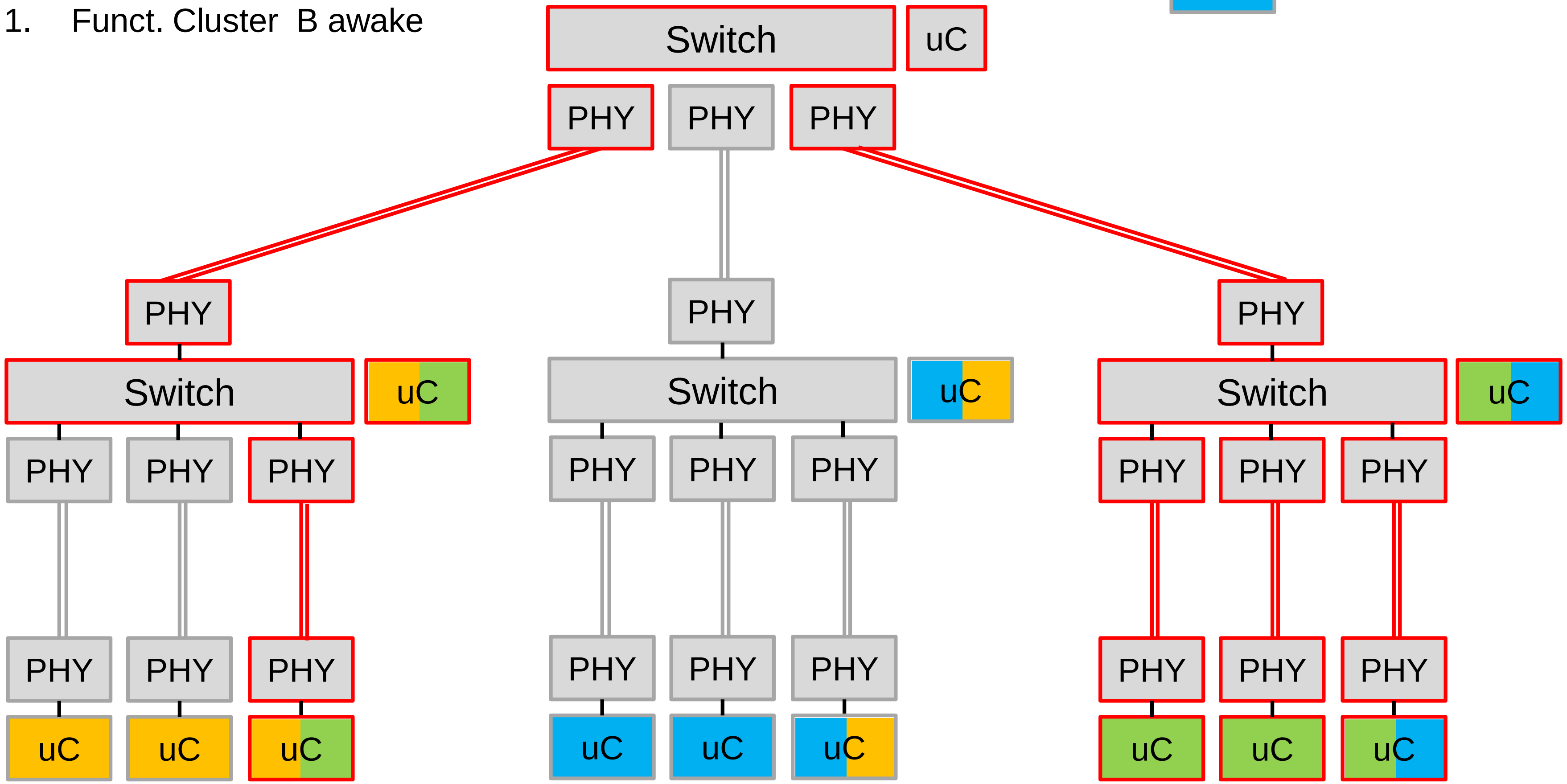
1. System in Sleep
2. Wakeup and Link-Up



Wake-up over Ethernet: Wake forwarding

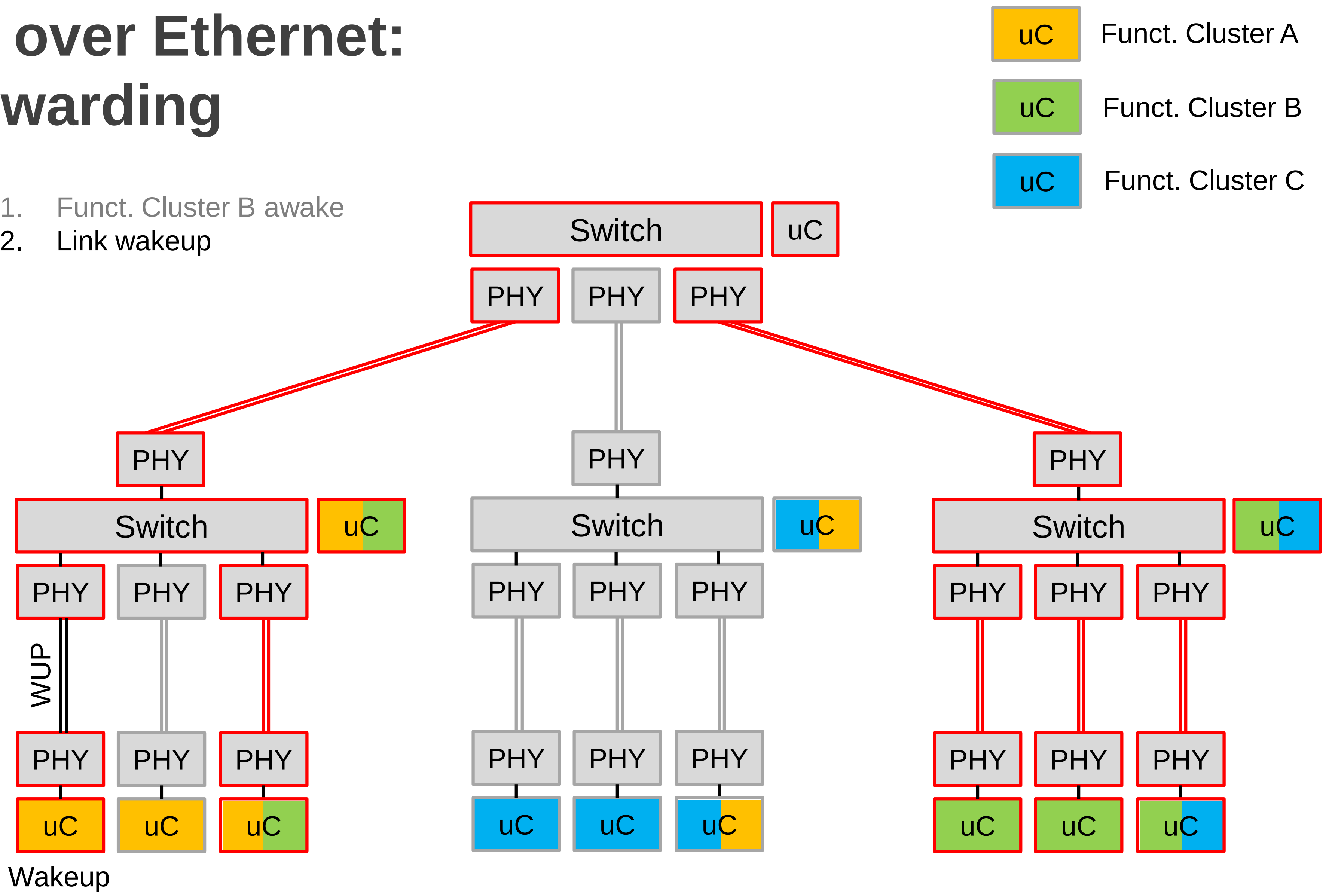


1. Funct. Cluster B awake



Wake-up over Ethernet: Wake forwarding

1. Funct. Cluster B awake
2. Link wakeup



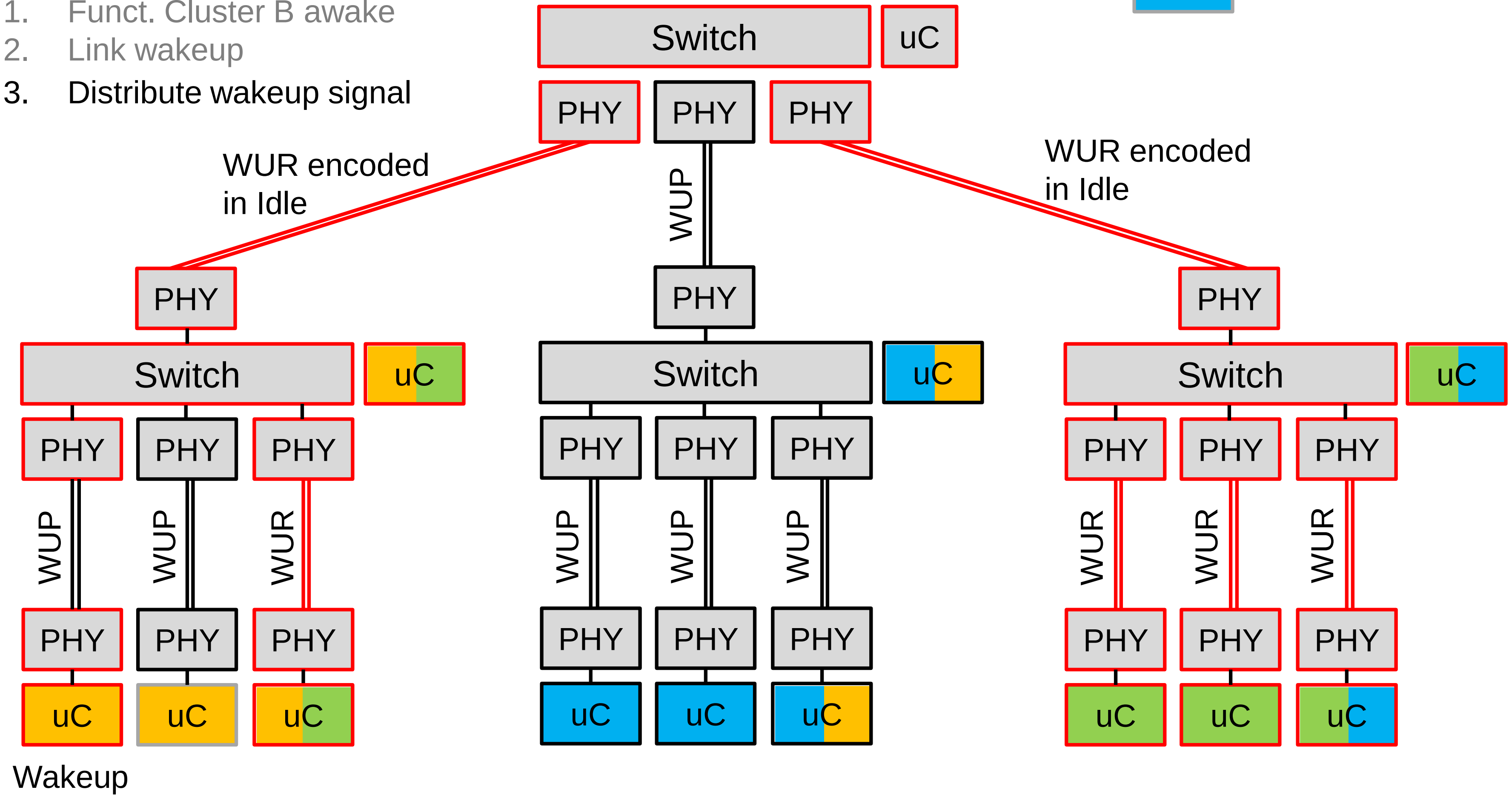
Wake-up over Ethernet: Wake forwarding

uC Funct. Cluster A

uC Funct. Cluster B

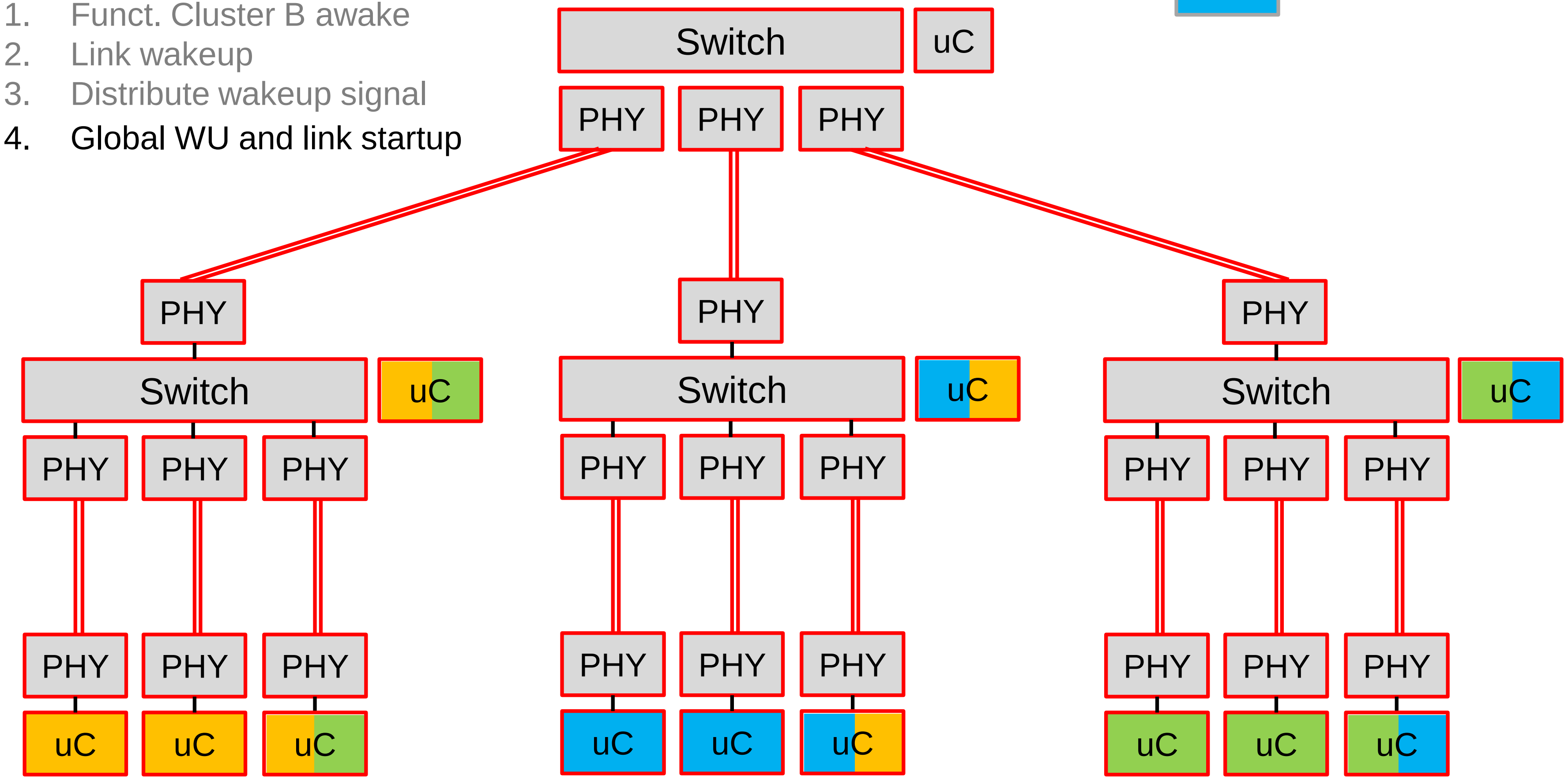
uC Funct. Cluster C

1. Funct. Cluster B awake
2. Link wakeup
3. Distribute wakeup signal



Wake-up over Ethernet: Wake forwarding

1. Funct. Cluster B awake
2. Link wakeup
3. Distribute wakeup signal
4. Global WU and link startup



uC	Funct. Cluster A
uC	Funct. Cluster B
uC	Funct. Cluster C

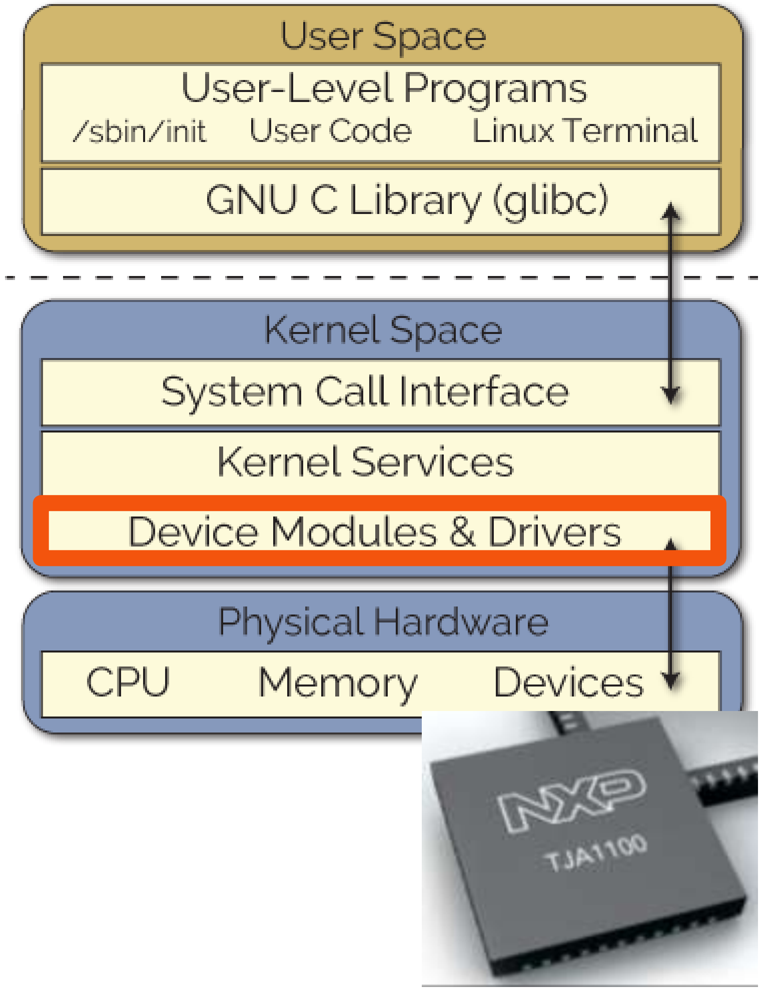
03.

Overview of Linux' phydev framework

Using Linux standardized PHY Abstraction Layer in an automotive context

Linux Kernel & Modules, Device Drivers

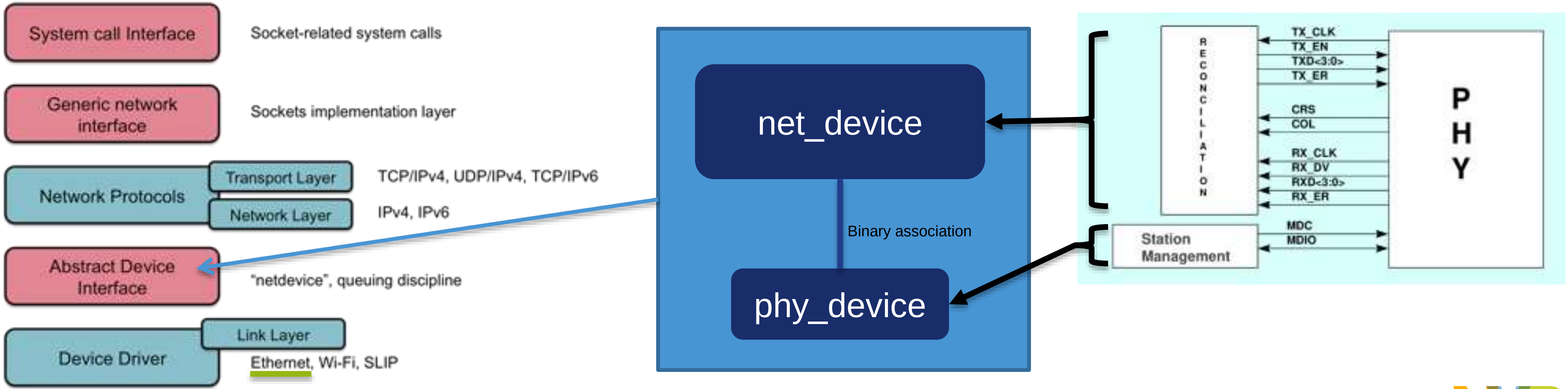
- **Linux Kernel**
  - Core of a computer's operating system
  - Provide abstraction layer to user space software
- **Kernel Module**
  - Can be loaded/unloaded at runtime to add functionality on demand
- **Device Driver**
  - Special kind of Kernel module
  - Software interface to Hardware device
  - Control a device that is attached to the computer



- Other software can communicate with the device
  - Via a standardized interface
  - Without knowledge about hardware specifics

Phy Device in the Linux Network Stack

- Each network interface is described by a struct **net_device** item
  - abstract device interface, core of the network driver layer (rx, tx functions etc.)
- Ethernet interface is special **net_device**
  - Ethernet specific settings, e.g. MTU, queue length, header length
- **phy_device** represents connected Phy
  - accessed during initialization and configuration, calls back on link change
  - not involved in actual data transfer

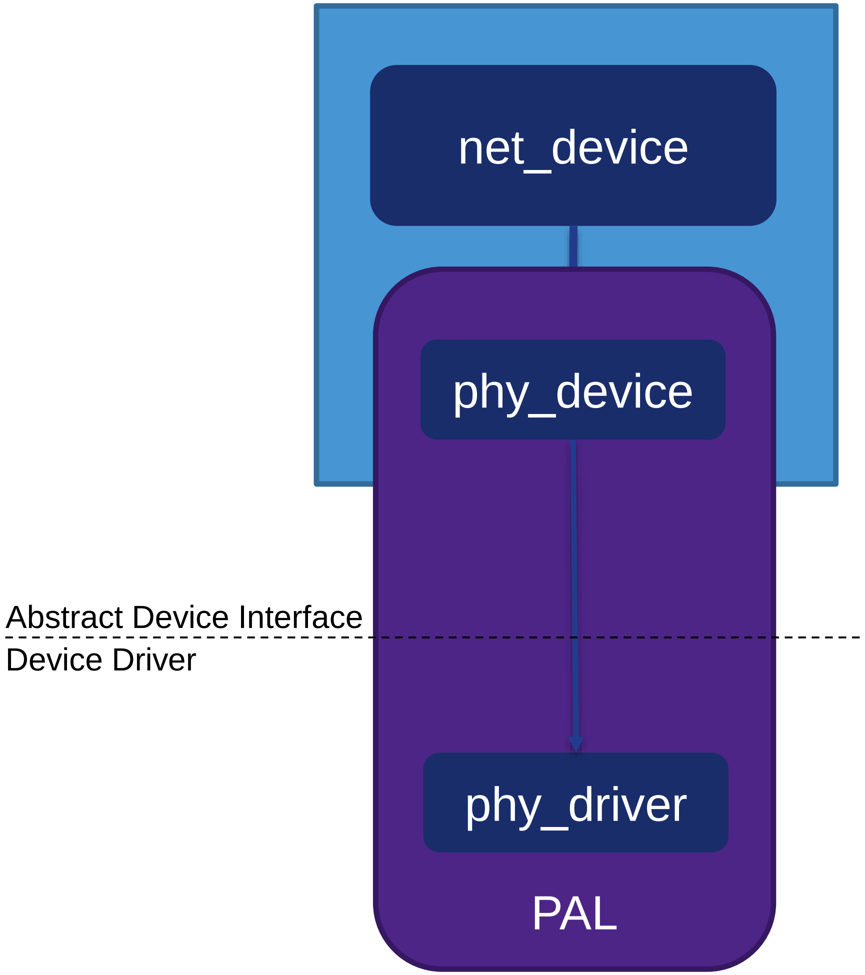


Phy Abstraction Layer (since ca 2005)

- Before the PAL the Phy management code was integrated into the network driver
  - ➢Separate **net_device** and **phy_device**
  - ➢Provide Framework that offers standard functionality

- Struct **phy_device**: An instance of a Phy
  - – ID, bus address, callback to **net_device** (e.g. Ethernet Driver)
  - – Device info: speed, duplex, pause, auto-negotiation support, current link state
  - – Interrupt infrastructure: irq, workqueues
  - – Vendor specific private data

  - ➢Pointer to struct **phy_driver**



Phy Abstraction Layer (since ca 2005)

- Struct **phy_driver**: Driver structure for a particular Phy type

  - Contains function pointers to *required* functions
    - **config_aneg**: configure & initiate auto-negotiation (if not available uses static configuration)
    - **read_status**: Determines the negotiated speed, duplex, pause frames

  - Contains function pointers to *optional* functions

De-/Initialization
- **config_init**
- **probe**
- **Remove**

Power Management
- **suspend**
- **resume**

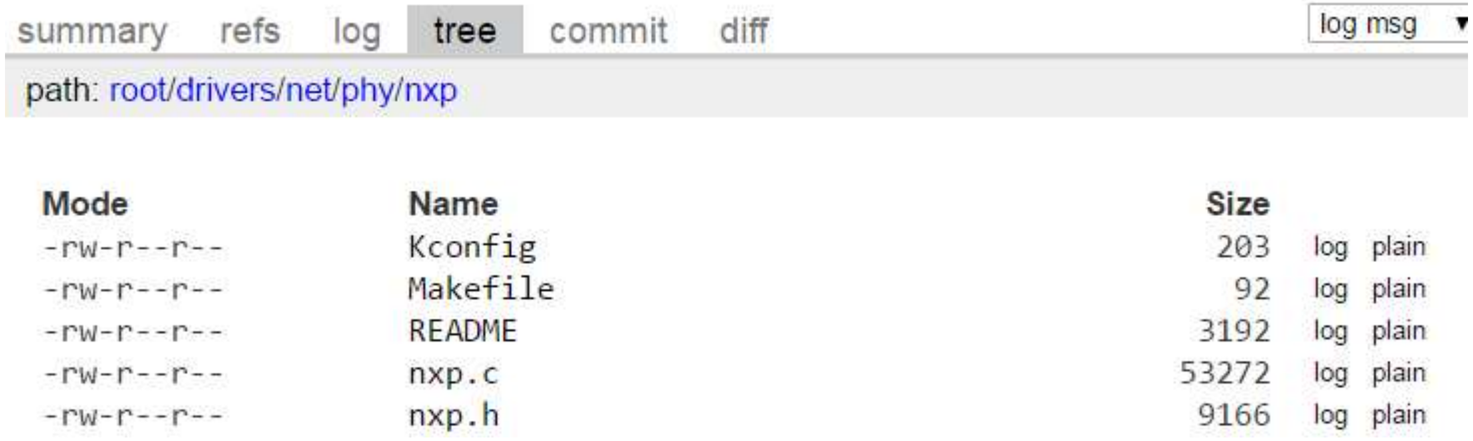
Interrupt handling
- **config_intr**
- **ack_interrupt**
- **did_interrupt**

– PAL provides generic version for some of the functions

TJA110x Linux PHY Driver

- Single Linux driver for TJA1100 and follow up devices integrated into Linux' PHY Abstraction Layer (PAL)
- Extended with automotive features
  - Support for Managed and Autonomous Mode
  - Master/Slave configuration
  - Cable Test
  - LED, Loopback and Test Modes
  - Sleep and Wakeup

Download: http://bit.ly/2lrIZxz



path: root/drivers/net/phy/nxp

Mode	Name	Size		
-rw-r--r--	Kconfig	203	log	plain
-rw-r--r--	Makefile	92	log	plain
-rw-r--r--	README	3192	log	plain
-rw-r--r--	nxp.c	53272	log	plain
-rw-r--r--	nxp.h	9166	log	plain

04.

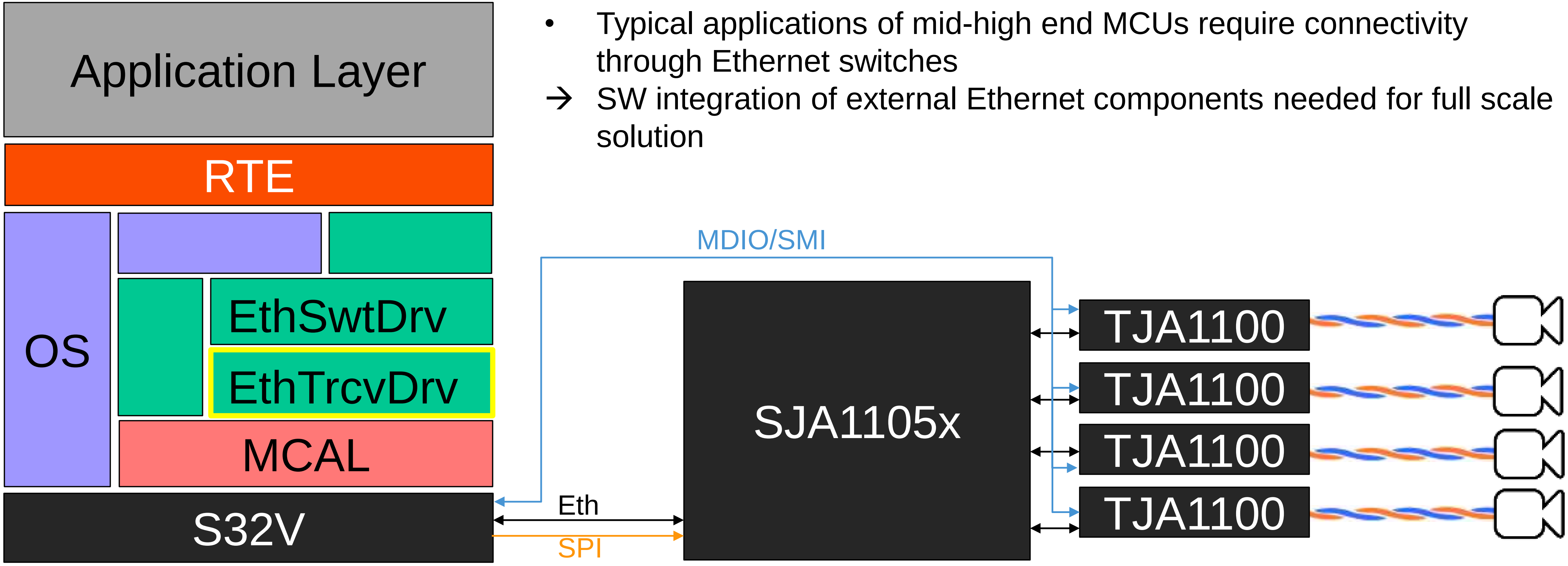
Overview of AUTOSAR EthTrcv Driver

Architecture and features of Ethernet Transceiver Drivers in AUTOSAR 4.3.0

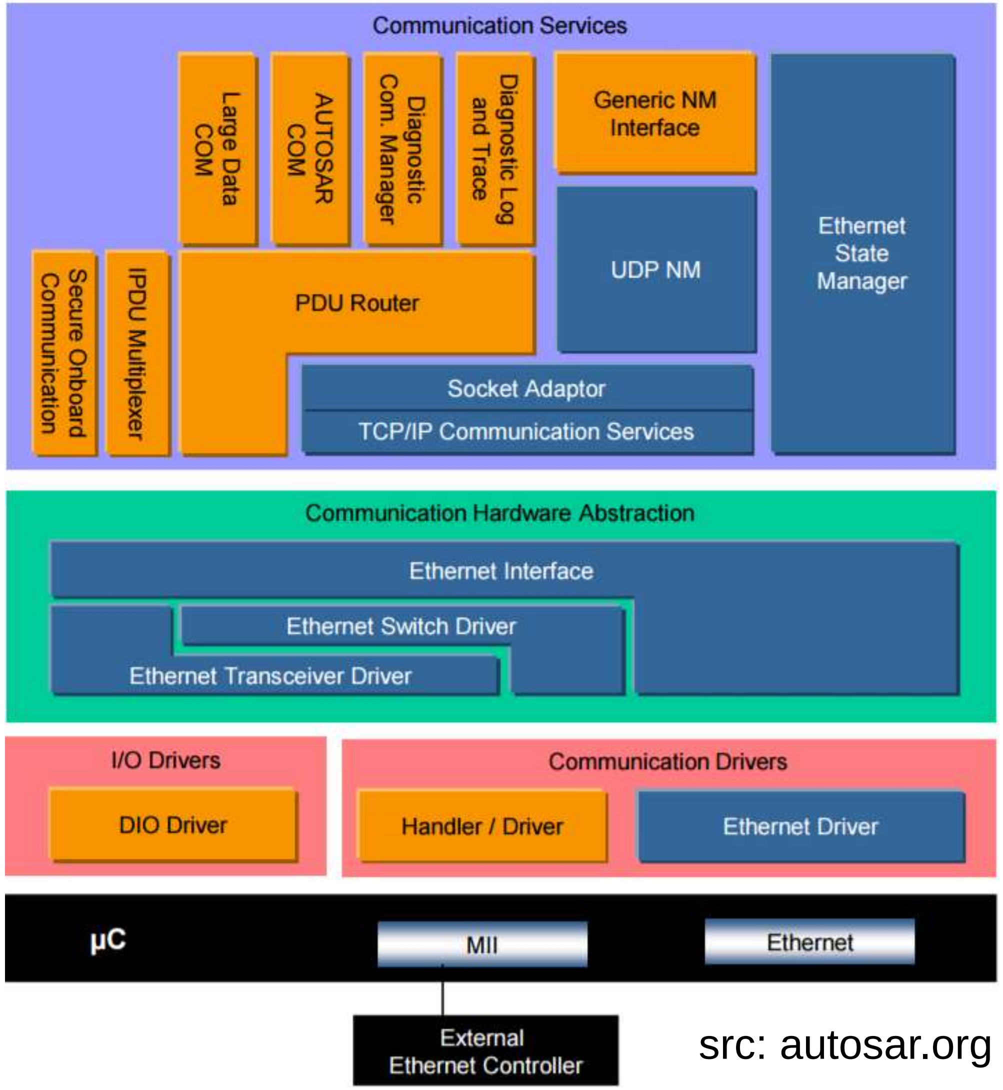
AUTOSAR Ethernet System Solution Example

- Typical applications of mid-high end MCUs require connectivity through Ethernet switches

→ SW integration of external Ethernet components needed for full scale solution



AUTOSAR Ethernet Stack



src: autosar.org

- Transceiver Driver is part of ECU Abstraction Layer (Communication Hardware Abstraction)
- Addressed by EthIf on higher layer (through EthSwt in case of a switch)
- Hardware access to PHY through Eth driver
- Used at run-time for initialization, mode changes from EthSM, notification of link state changes
- EthTrcv: Not involved in data path

Differentiating Features of the Ethernet Transceiver Driver

- Static configuration (Master/Slave, MII mode etc.)
- Support for sleep/wakeup
  - Managed by/interfacing with Ethernet State Manager
  - Configurable callout function
  - Detailed wakeup reason
- Advanced diagnostics including
  - Internal, External, Remote Loopback mode
  - Cable Test
  - 100Base-T1 Test Modes
  - SNR

→Highly optimized for automotive needs, feature set well aligned with capabilities of automotive PHYs

05.

Summary

SECURE CONNECTIONS
FOR A SMARTER WORLD