Implementing LoRa, FSK & Further LPWAN Interfaces

Linux Plumbers Conference 2019

You, Me and IoT Microconference

Andreas Färber

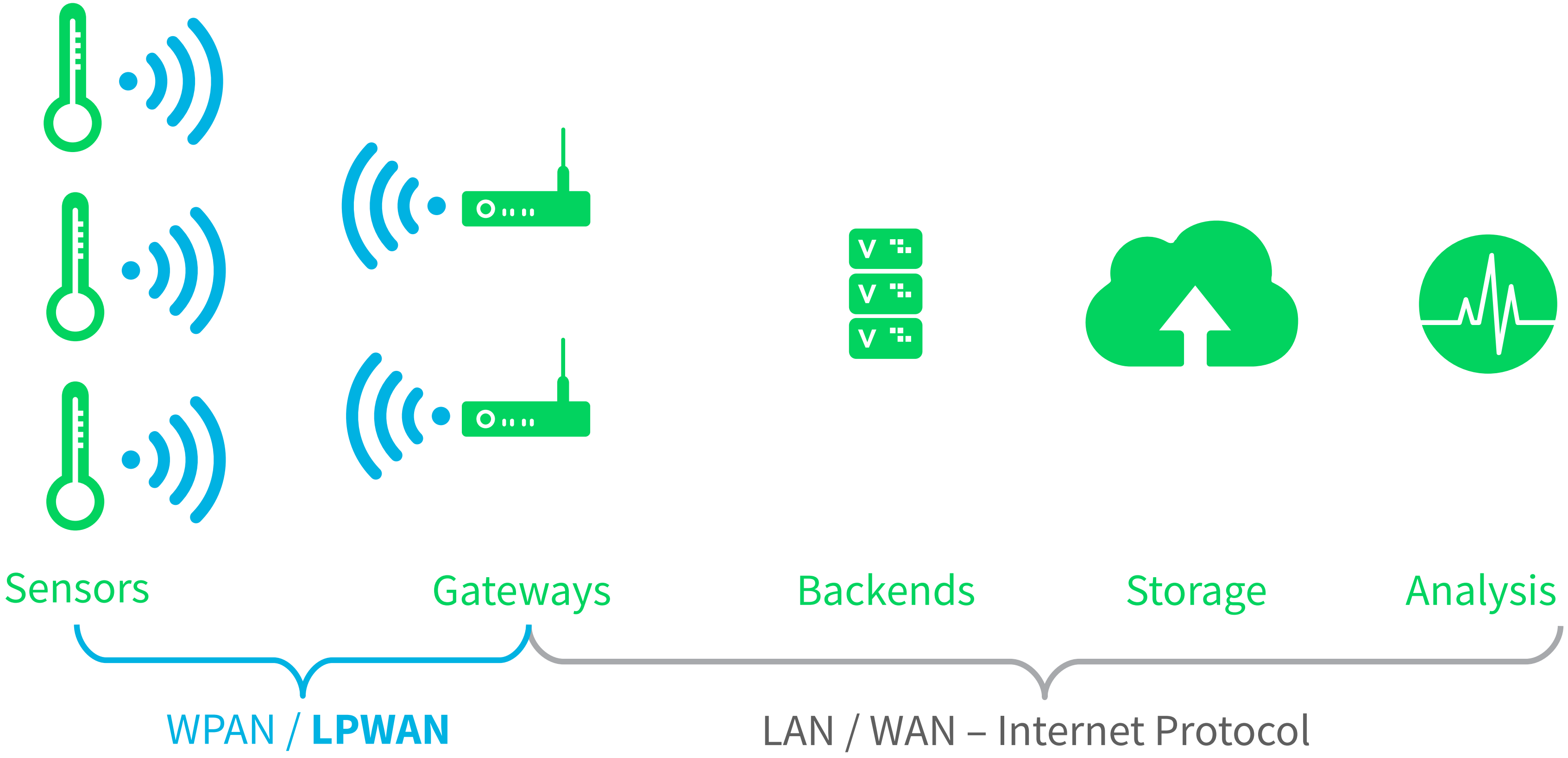
ProjMgr arm64, SUSE Labs

✉ afaerber@suse.com

afaerber on Freenode.net

IoT Architecture and Properties

IoT Tiers With Low-Power WAN



Sensors

Gateways

Backends

Storage

Analysis

WPAN / **LPWAN**

LAN / WAN – Internet Protocol

LPWAN Characteristics (1/2)

Low Power

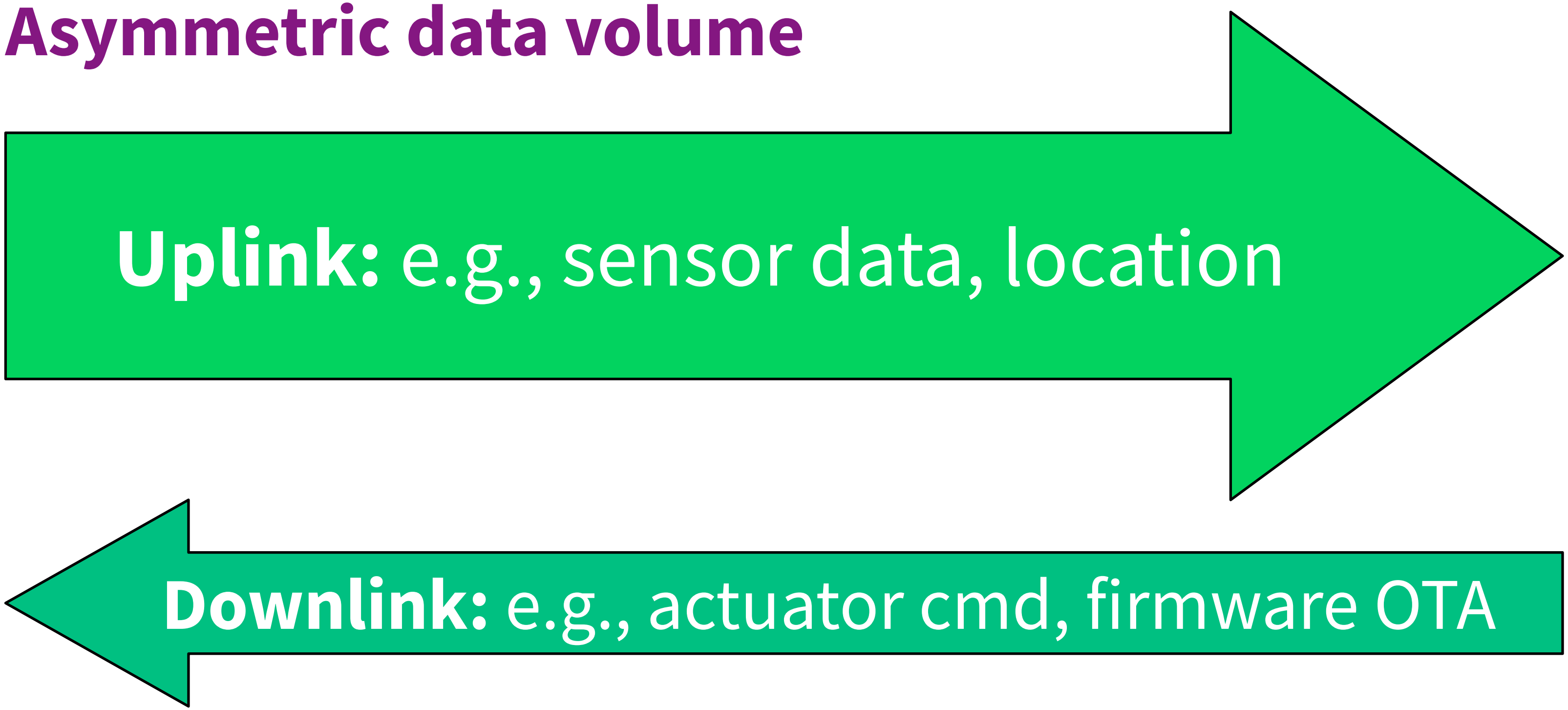
01101 up to 10 years

Wide Area

up to 48 km (30 miles)

LPWAN Characteristics (2/2)

Asymmetric data volume



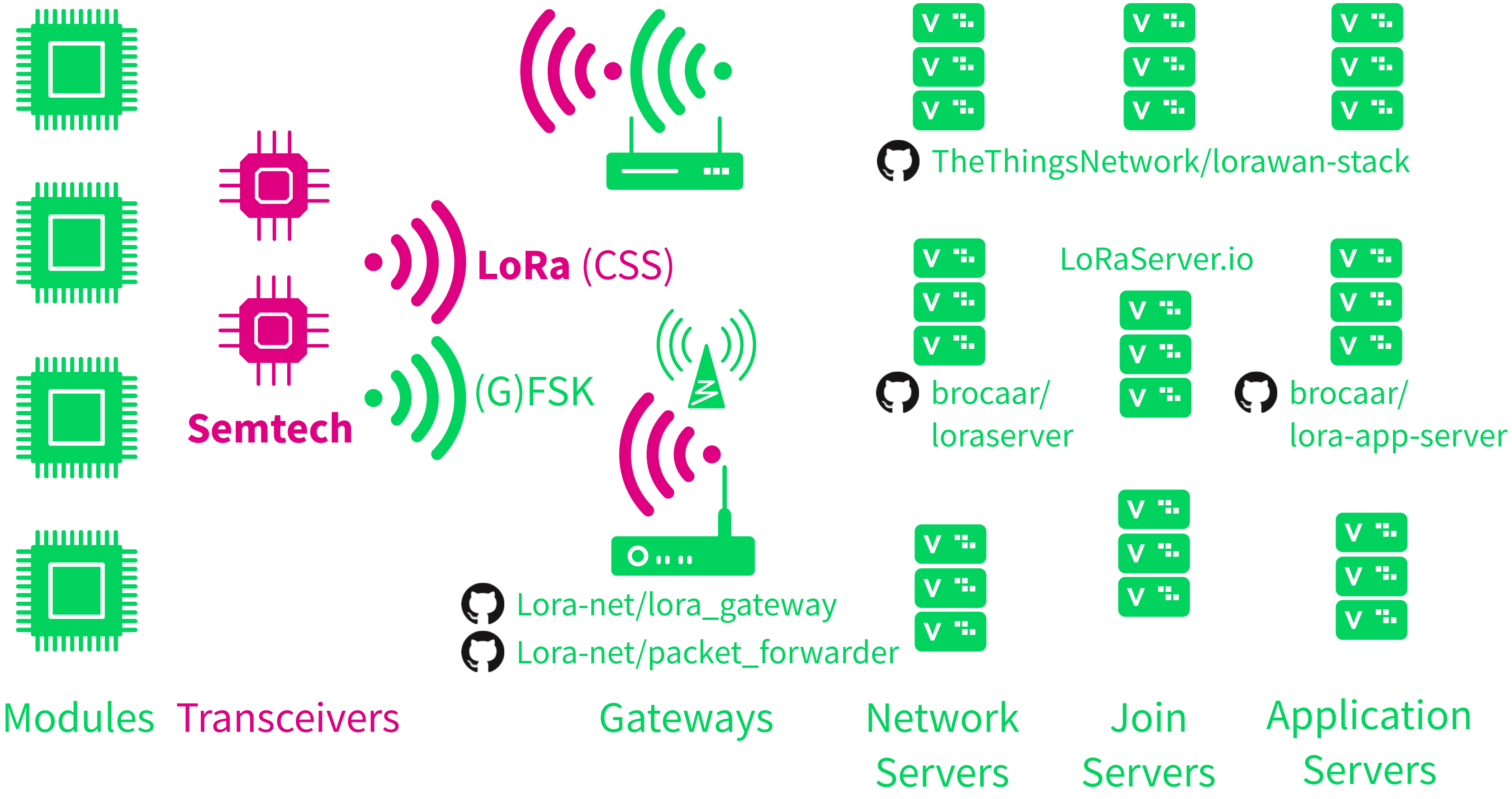
Uplink: e.g., sensor data, location

Downlink: e.g., actuator cmd, firmware OTA

LPWAN Classification

- **Unlicensed** (U-LPWA) – ISM/SRD
  - Sub-GHz (433 / 470 / 868 / 915 / 923 MHz)
  - 2.4 GHz
  - Regulatory restrictions: duty cycle, dwell time
- **Licensed**
  - LTE

LoRaWAN Architecture



Semtech

LoRa (CSS)

(G)FSK

Lora-net/lora_gateway

Lora-net/packet_forwarder

TheThingsNetwork/lorawan-stack

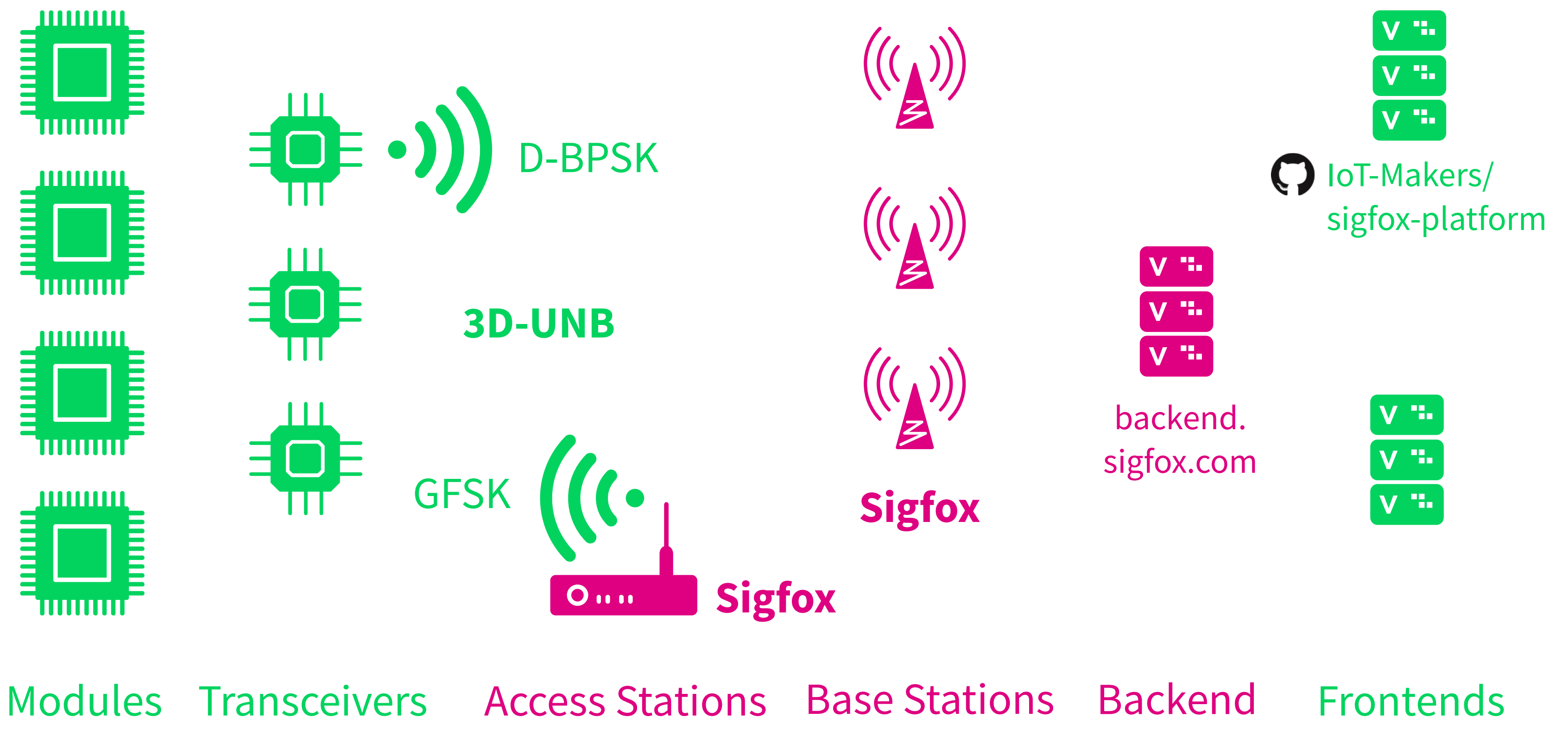
LoRaServer.io

brocaar/loraserver

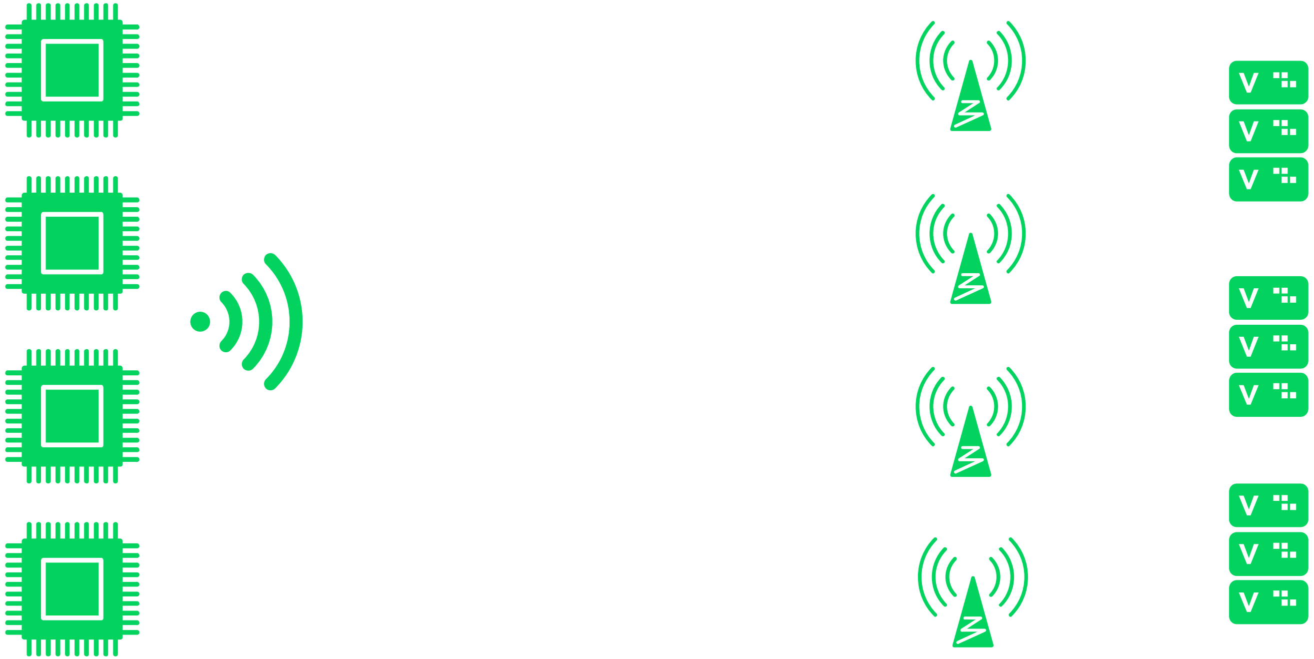
brocaar/lora-app-server

Modules Transceivers Gateways Network Servers Join Servers Application Servers

Sigfox Network Architecture



NB-IoT Network Architecture



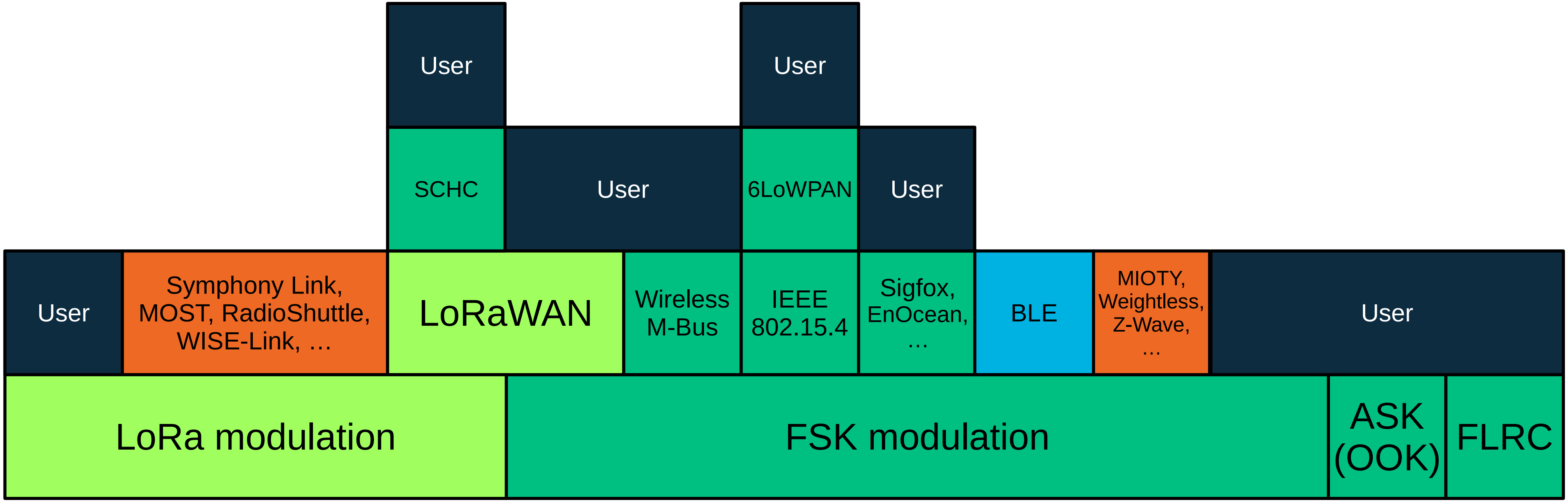
Modems

Base Stations

Servers

Representing IoT in Linux

Protocol Layers Around LoRa



User

User

SCHC

User

6LoWPAN

User

User

Symphony Link, MOST, RadioShuttle, WISE-Link, …

LoRaWAN

Wireless M-Bus

IEEE 802.15.4

Sigfox, EnOcean, …

BLE

MIOTY, Weightless, Z-Wave, …

User

LoRa modulation

FSK modulation

ASK (OOK)

FLRC

Socket Address/Protocol Families

- AF_BLUETOOTH, AF_IEEE802154
- *AF_LORA?* → PF_PACKET
- *AF_**LORAWAN***
- *AF_3DUNB?*

- **AF_MAX** prohibits dynamic additions

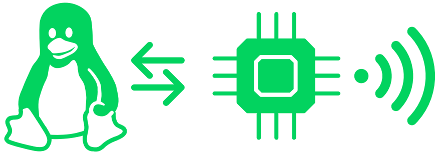
Sockets: PF_PACKET Types

- ETH_P_IEEE802154 (0x00F6)
- *ETH_P_**LORA**, ETH_P_FLRC*
  - *ETH_P_LORAWAN*
- *ETH_P_OOK(?), ETH_P_**FSK***
  - *ERP2? ZWAVE? …*
- *ETH_P_3DUNB[_{DL,UL}]?*

LoRa/FSK driver project

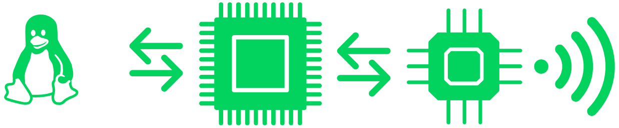
Getting Started With LoRa Chipsets

Types Of LoRa Radio Modules



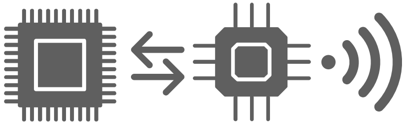
Plain transceiver

- SPI / UART / USB
- Volatile register settings
- Software MAC needed



MCU w/firmware + transceiver

- UART / USB Serial
- Firmware determines chip features exposed
- Optional certified MAC



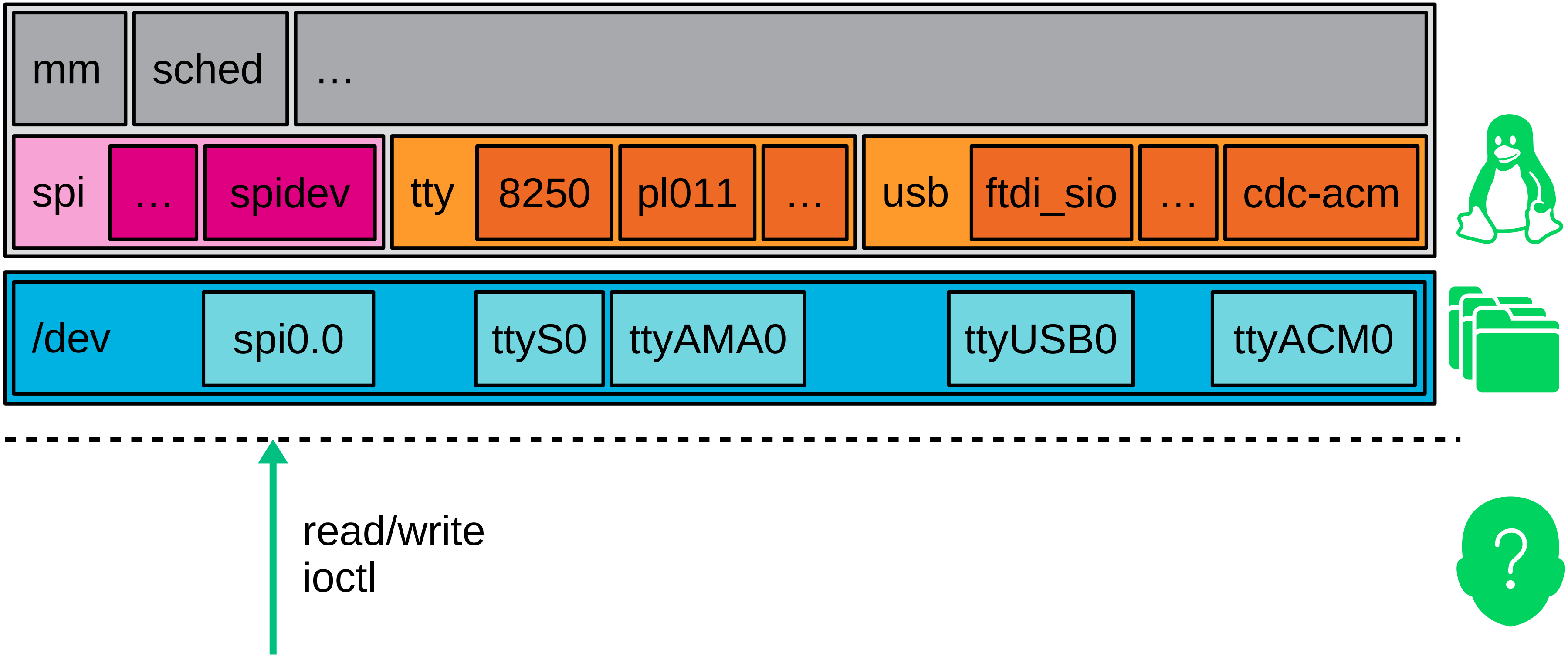
Plain MCU + transceiver

- n/a – no fixed API
- Custom MCU code for sending / receiving
- Optional MAC

LoRa Use Cases On Linux

- Prototyping of sensor/actuator node
  - Well-documented transceiver chips
  - Simple sending of raw LoRa packets
  - LoRaWAN client
- LoRaWAN gateway
  - Complex multi-channel “concentrator” chip

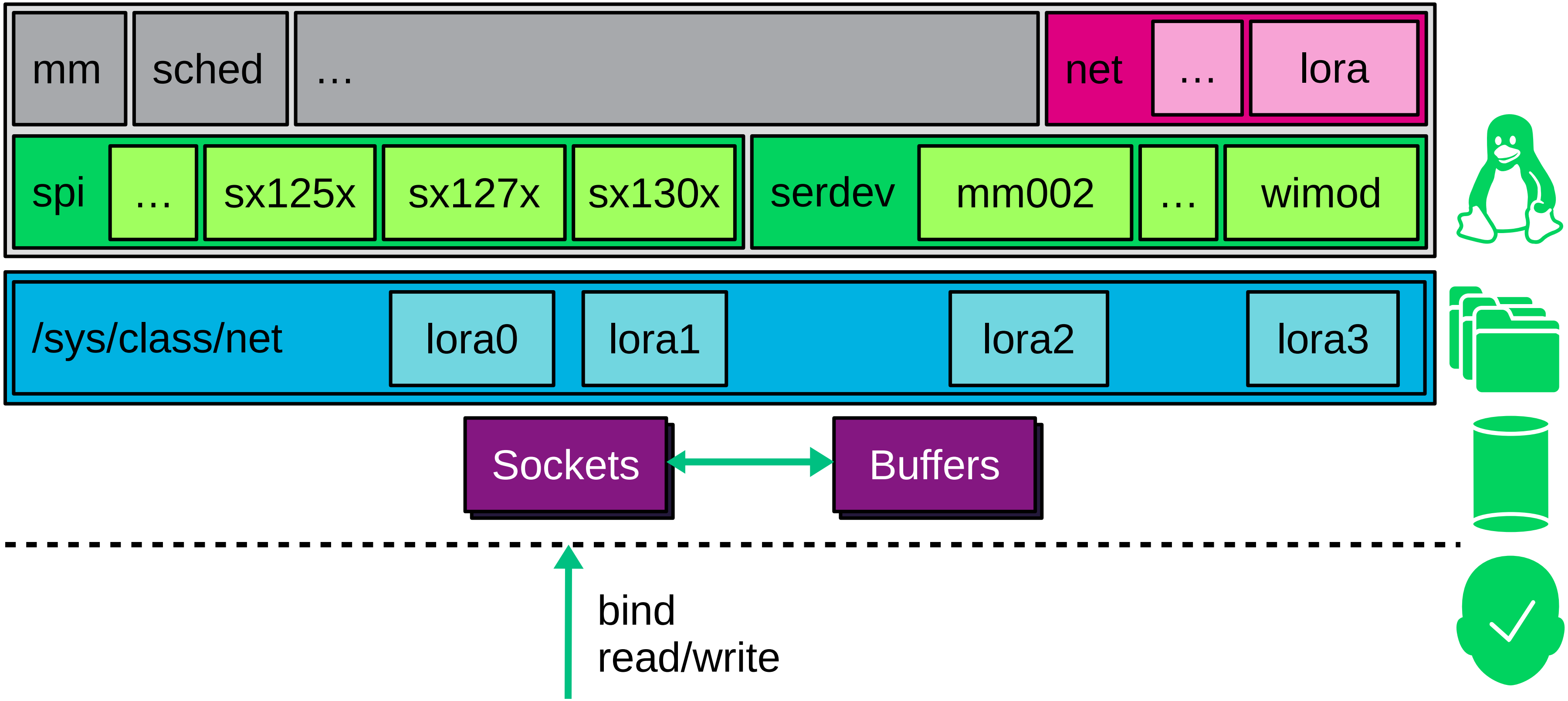
Accessing LoRa Hardware Today



Goals For LoRa On Linux

- Get vendor-independent interface
  - Hardware support via kernel, not via forks
  - Generic Open Source packages via distros
- Allow to containerize the userspace part
- Enable Kubernetes for gateway & backend

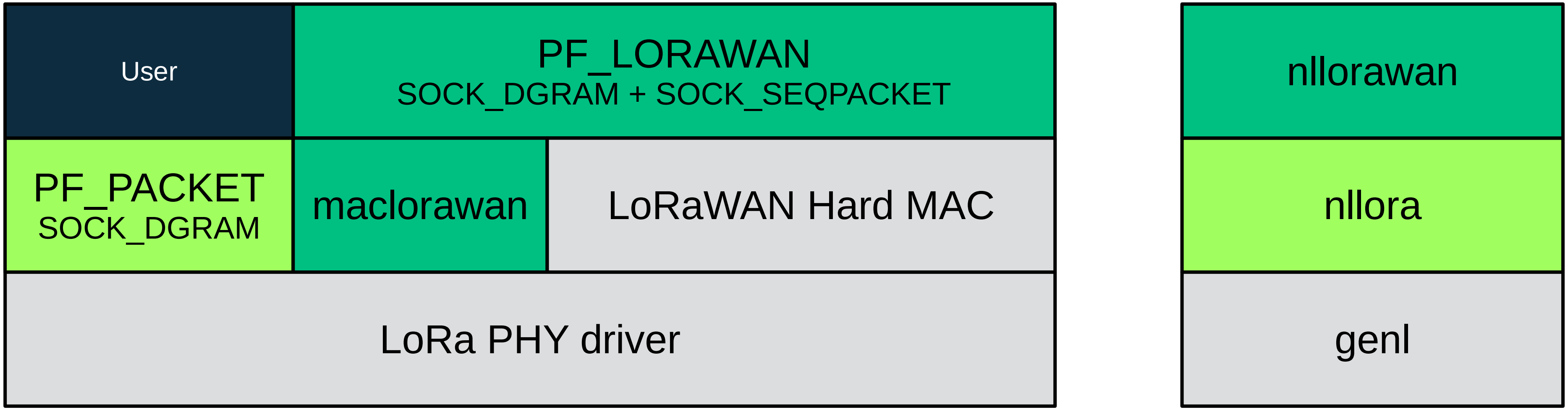
LoRa Sockets Concept



Netlink Commands / Attributes

NLLORA_CMD_SET_	**NLFSK_CMD_SET_**
• U32 FREQ	• U32 FREQ
	• U32 FREQ_DEV
• S32 TX_POWER	• S32 TX_POWER
• *u32 bandwidth*	
• *u8 sf, cr, sync_word*	

LoRa Socket Layers (Proposed)



User

PF_LORAWAN
SOCK_DGRAM + SOCK_SEQPACKET

PF_PACKET
SOCK_DGRAM

maclorawan

LoRaWAN Hard MAC

LoRa PHY driver

nllorawan

nllora

genl

Userspace Code

- test.c
  - Trivial code sending a packet
- nltest.c
  - Evolving into a dual LoRa/FSK config tool

Regmap Workaround

- Writing to FIFOs is broken in 5.3 and 5.2
  - Breaks bringing sx130x interfaces up
- Patch in progress by Ben Whitten
- Workaround: bump .max_register = 0xffff

Netdev 0x13 Workshop Outcome

- Use module param for hard- vs. soft-MAC
- Expose two devices, use carrier up/down
- Not all modes (BLE) need to be exposed
- LoRaWAN soft-MAC like 802.15.4 6LoWPAN

Help Needed!

- Netlink operations need to be defined
- Complete/add driver implementations
- Prepare and document DT snippets
- Testing
- Design discussions
- Solve various unrelated problems

Resources

- linux-lpwan mailing list
- #lora on Freenode IRC

- ELCE 2018 video and slides
- Netdev 0x13 paper, slides and video
- oSC 2019 video

Kernel development on openSUSE

Kernel Development Options

- linux.git (cross-)compile → manual deploy
- kernel-source.git → OBS → package install
- kernel-{default,lpae}-devel package
  → local module(s) or KMP package in OBS

My Interop Testing Setup

- Various Arm, MIPS boards (C-Sky TBD) with expansion boards or cables
    - No Intel or Arm ACPI yet – hardware needed
- GitHub afaerber/lora-modules.git
- kernel.org afaerber/linux-lora.git
- One sends; monitor others’ dmesg output

Credits – LoRa driver project

Industry Contributors – Code

Laird™

Industry Supporters – Hardware

AppconWireless

DRAGINO

Embedded MICRO TECHNOLOGY

Gimasi

HELTEC™ AUTOMATION

IMST

Laird™

MIPOT HI-TECH COMPANY

NEMEUS

n|fuse

openSUSE®