Bluetooth High Speed and Virtual / Soft AMP Controller for mac80211

Andrei Emeltchenko / Intel OTC

Wireless mini-summit

Barcelona

09.11.2012

Contents

- Bluetooth High Speed and how it works
- Types of Bluetooth High Speed Controllers (AMPs)
- Current Bluetooth High Speed stack implementation
- SoftAMP implementation proposals

Bluetooth speed evolution

- Basic Rate (BR)
  - Specification versions: 1.0 – 2.1
  - Data rate up to 1 Mbit/s
- Enhanced Data Rate (EDR)
  - Specification versions: 2.0+EDR – 2.1+EDR
  - Data rate up to 3 Mbit/s
- Bluetooth High Speed
  - Specification versions: 3.0+HS, 4.0
  - Data rate up to 24 Mbit/s

High Speed main ideas

- Utilize secondary radio for fast data transfer
  - 802.11 radio is specified
  - Only data is transferred through HS, negotiation and establishment is done through BR/EDR.
- Using existing HW
  - Usually both Bluetooth and WIFI are on the same chip
  - Cheap to implement for manufacturer
- Transparent to user
  - Support existing Bluetooth profiles
  - No new paradigm

Is this that simple?

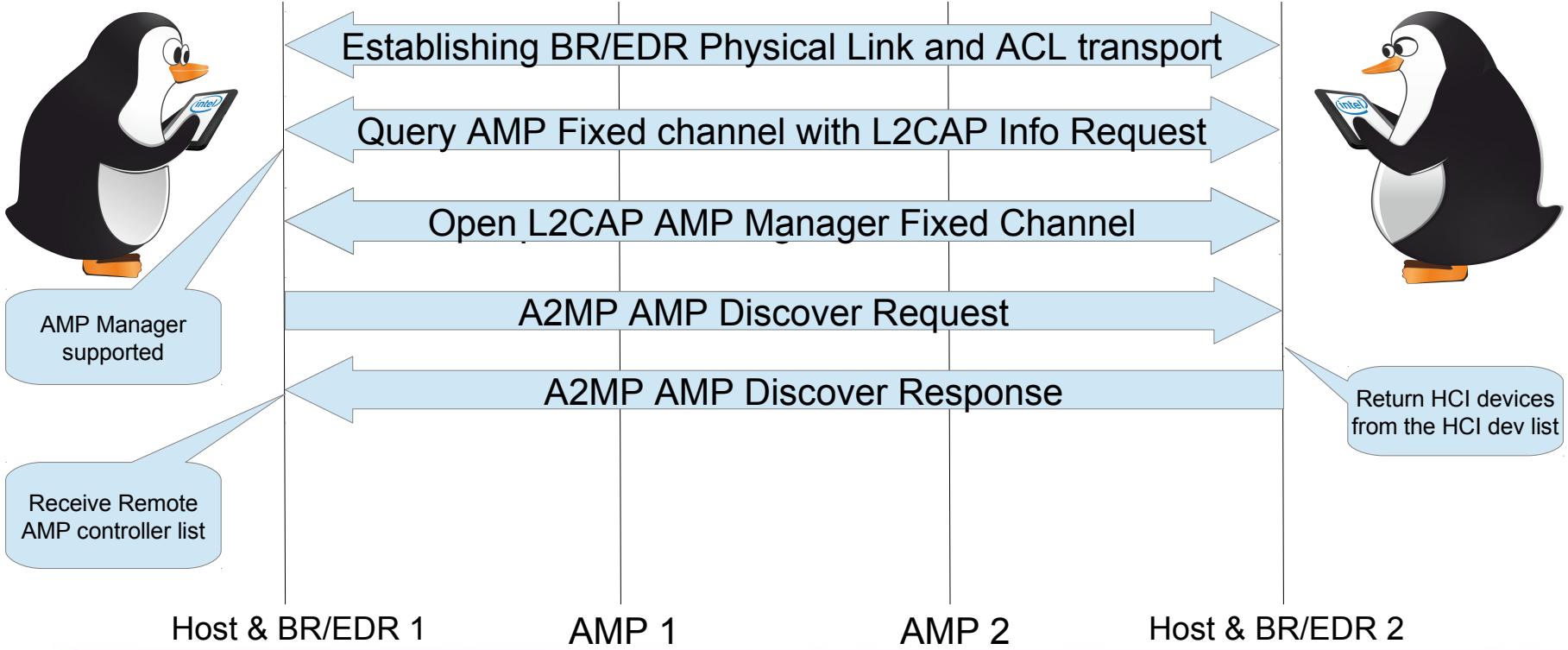
- To make Bluetooth High Speed working with obex

```
...
+                {
+                int chan_policy = BT_CHANNEL_POLICY_AMP_PREFERRED;
+
+                if (setsockopt(sock, SOL_BLUETOOTH, BT_CHANNEL_POLICY,
+                               &chan_policy, sizeof(chan_policy)) < 0)
+                        goto failed;
+                }
...
```

- Currently also need kernel option to enable High Speed

```
options bluetooth enable_hs=y
```

Start Bluetooth BR/EDR connection

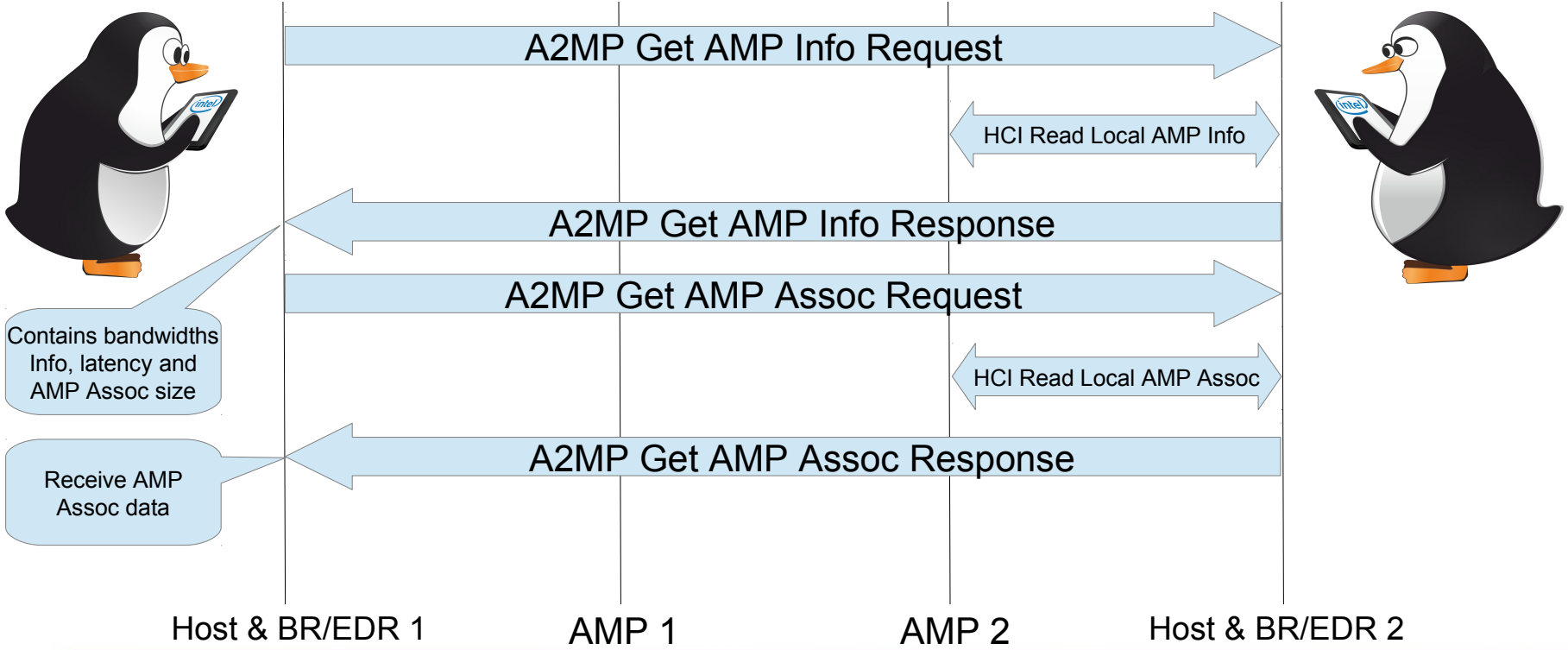


Remote Controller list

- **Available remote controllers**
- **Contains:**
  - **Controller id**
  - **Controller type**
    - **BR/EDR**
    - **802.11**
  - **Controller status**
    - **AMP status and capacity information**

```
Controller list:
        id 0 type 0 (BR-EDR) status 0x01 (Bluetooth only)
        id 1 type 1 (802.11 AMP) status 0x01 (Bluetooth only)
```

Get remote AMP controller Info and Assoc



AMP Assoc structure

- **Most important data in High Speed communication**
- **Contains:**
  - **MAC Address**
  - **Preferred Channel List**
  - **Connected Channel List**
  - **PAL Capabilities**
  - **PAL version**

```
Assoc data [len 36]:
        MAC: 00:50:43:21:30:F9
        Preferred Chan List (number of triplets 2)
          Country code: US
          Reg ext id 201 reg class 12 coverage class 0
          Channels 1 - 12 max power 20
        PAL CAP: 03 00 00 00
        PAL VER: 01 Comp ID: 0048 SubVer: 0001
```

Establish High Speed Physical Link

HCI Create Physical Link

HCI Write Remote AMP Assoc

Scanning & Start Beaconing

HCI Channel Selected

HCI Read Local AMP Assoc

Read selected channel and other Assoc info

A2MP Create Physical Link Request

HCI Accept Physical Link

HCI Write Remote AMP Assoc

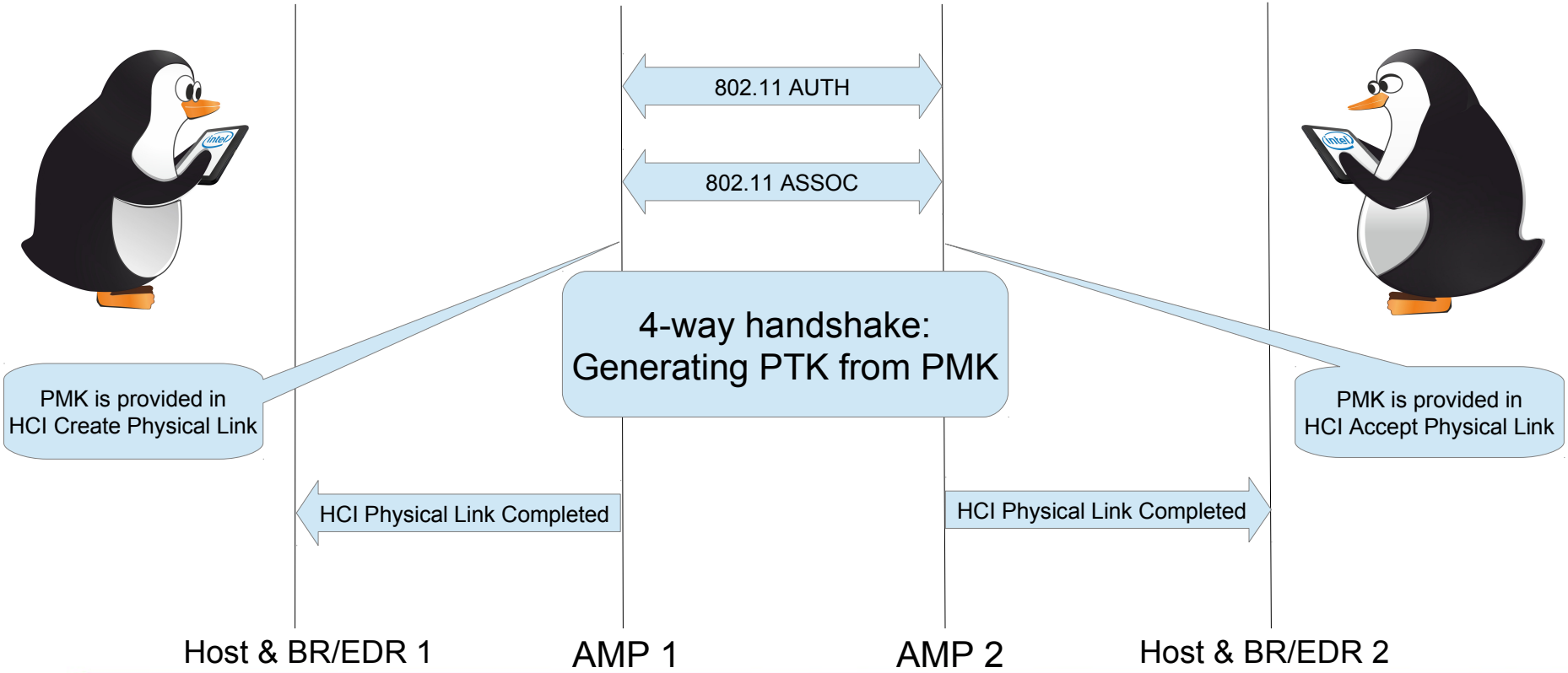
Include Local AMP Assoc in Request

Start Beaconing

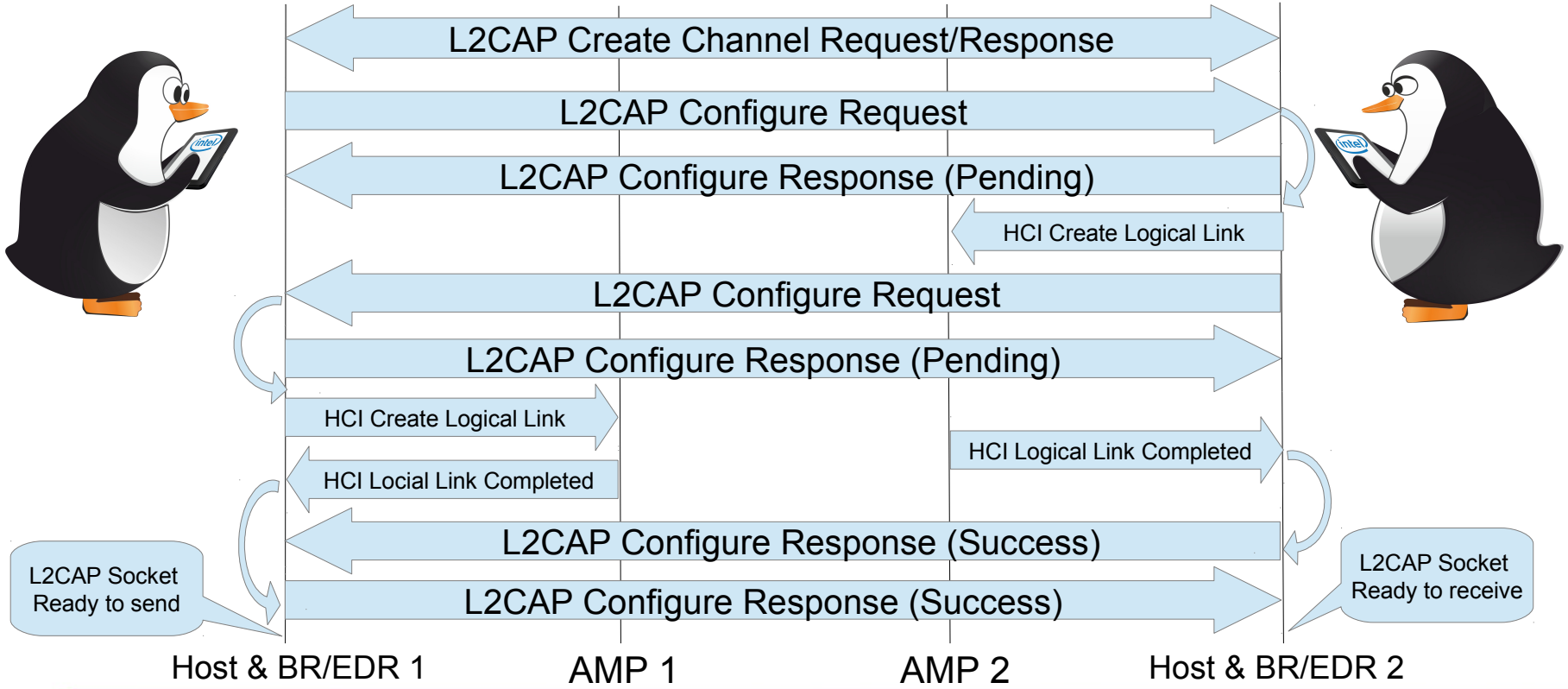
A2MP Create Physical Link Response

Host & BR/EDR 1 | AMP 1 | AMP 2 | Host & BR/EDR 2

High Speed Link Security



Create High Speed L2CAP channel and logical link



Send data over High Speed Link

HCI send

ACL header	L2CAP header	DATA

wireless send

IEEE802.11 header	LLC SNAP header	L2CAP header	DATA

HCI receive

ACL header	L2CAP header	DATA

Host & BR/EDR 1 | AMP 1 | AMP 2 | Host & BR/EDR 2

High Speed data packet

wireless-llc-filtered.cap [Wireshark 1.9.0 (SVN Rev 32dca73 from (no)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

No.	Time	Source	Destination	Protocol	Length	Info
3	0.008804	MarvellS_21:3a:	XaviTech_62:a1:bd	EAPOL	217	Key (Message 3 of 4)
4	0.012028	XaviTech_62:a1:	MarvellS_21:3a:e8	EAPOL	161	Key (Message 4 of 4)
5	0.411641	MarvellS_21:3a:	XaviTech_62:a1:bd	L2CAP	93	Connection oriented channel

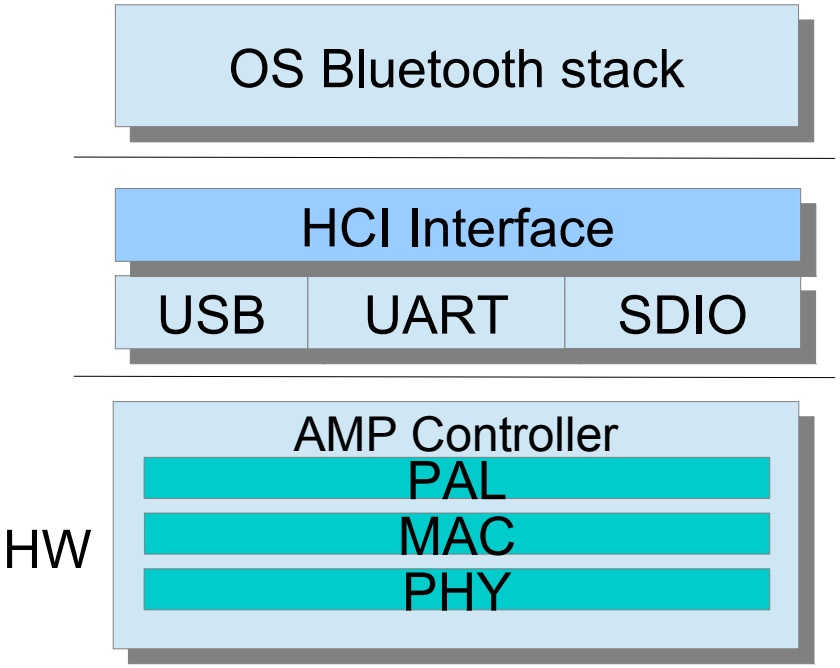
```
▶ Frame 5: 93 bytes on wire (744 bits), 93 bytes captured (744 bits)
▶ Radiotap Header v0, Length 18
▶ IEEE 802.11 QoS Data, Flags: .p....FTC
▼ Logical-Link Control
    DSAP: SNAP (0xaa)
    IG Bit: Individual
    SSAP: SNAP (0xaa)
    CR Bit: Command
  ▶ Control field: U, func=UI (0x03)
    Organization Code: Bluetooth (0x001958)
    PID: AMP_U L2CAP ACL data (0x0001)
▼ Bluetooth L2CAP Protocol
    Length: 11
    CID: 0x0041
    Payload: 00008000071000100076be

0000  00 00 12 00 2e 48 00 00  10 6c 80 09 c0 00 c2 05   .....H.. .l......
0010  00 00 88 43 2c 00 e0 91  53 62 a1 bd 00 50 43 21   ...C,... Sb...PC!
```

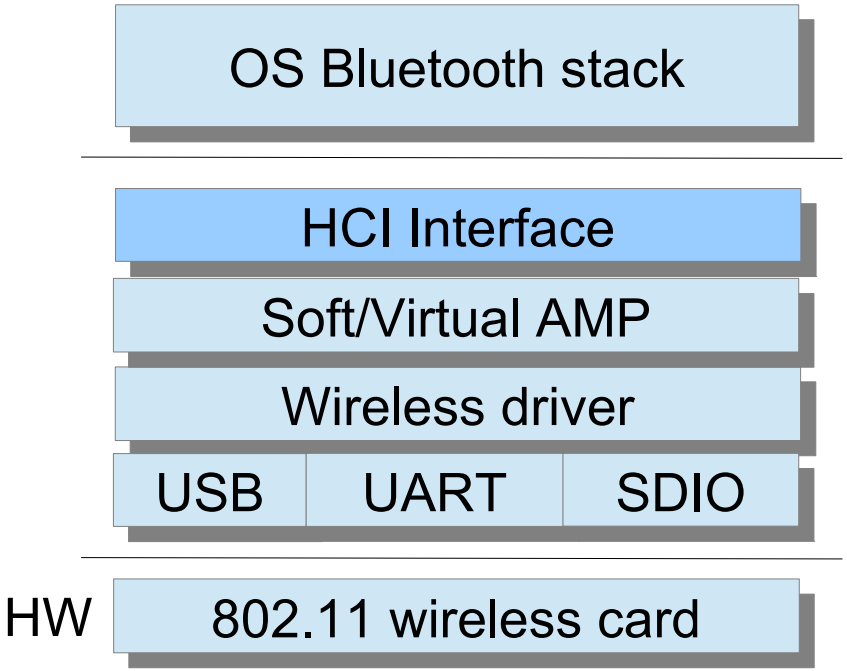
Frame (93 bytes) | Decrypted CCMP data (23 bytes)

File: "wireless-llc-filtered.cap" 1310 bytes 00:00:10 ... Profile: Default

Types of AMP



FullAMP



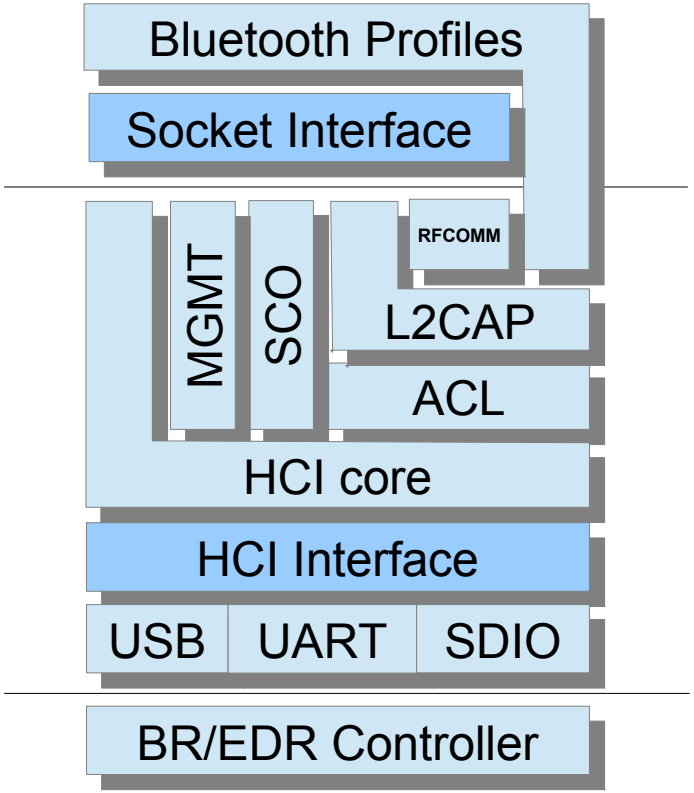
SoftAMP

Bluez stack

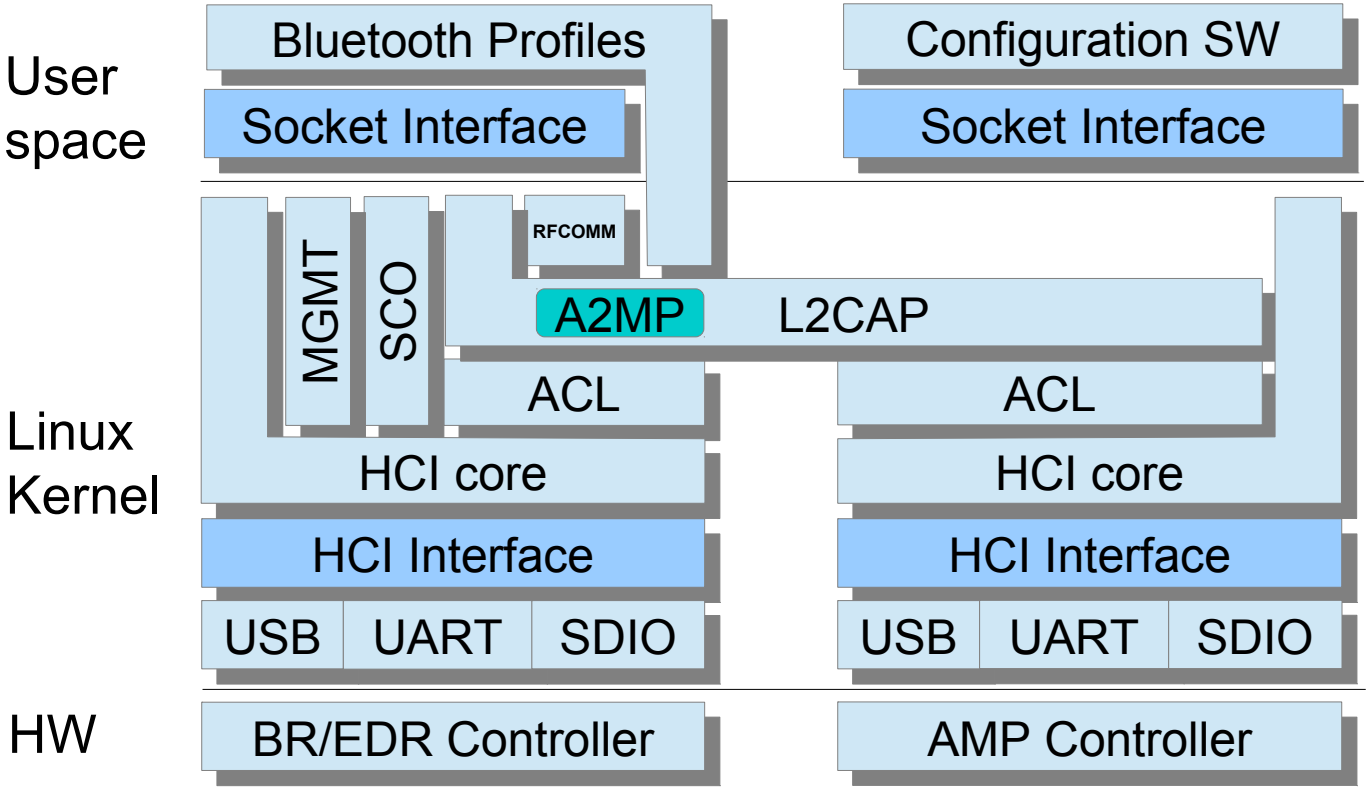
User space

Linux Kernel

Firmware & HW



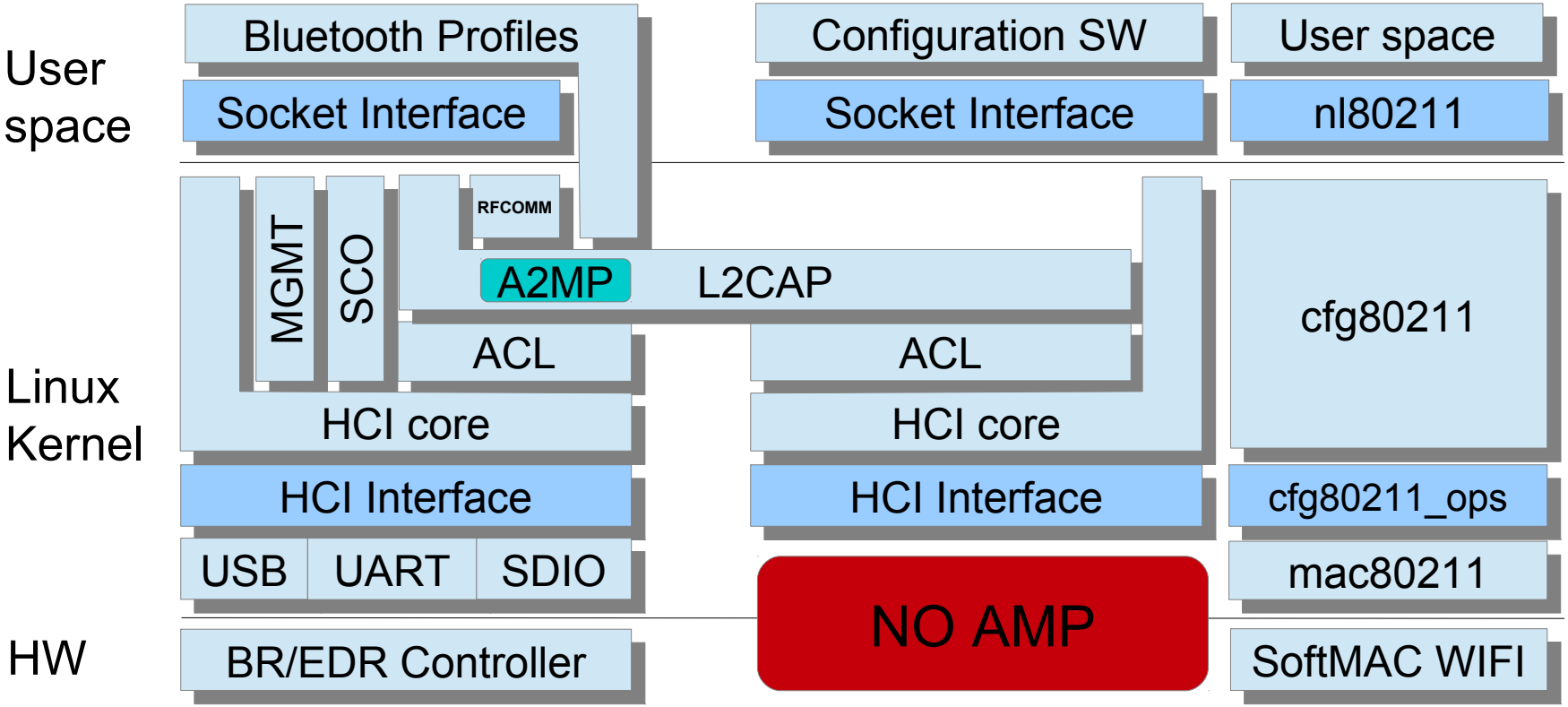
BlueZ stack with High Speed support



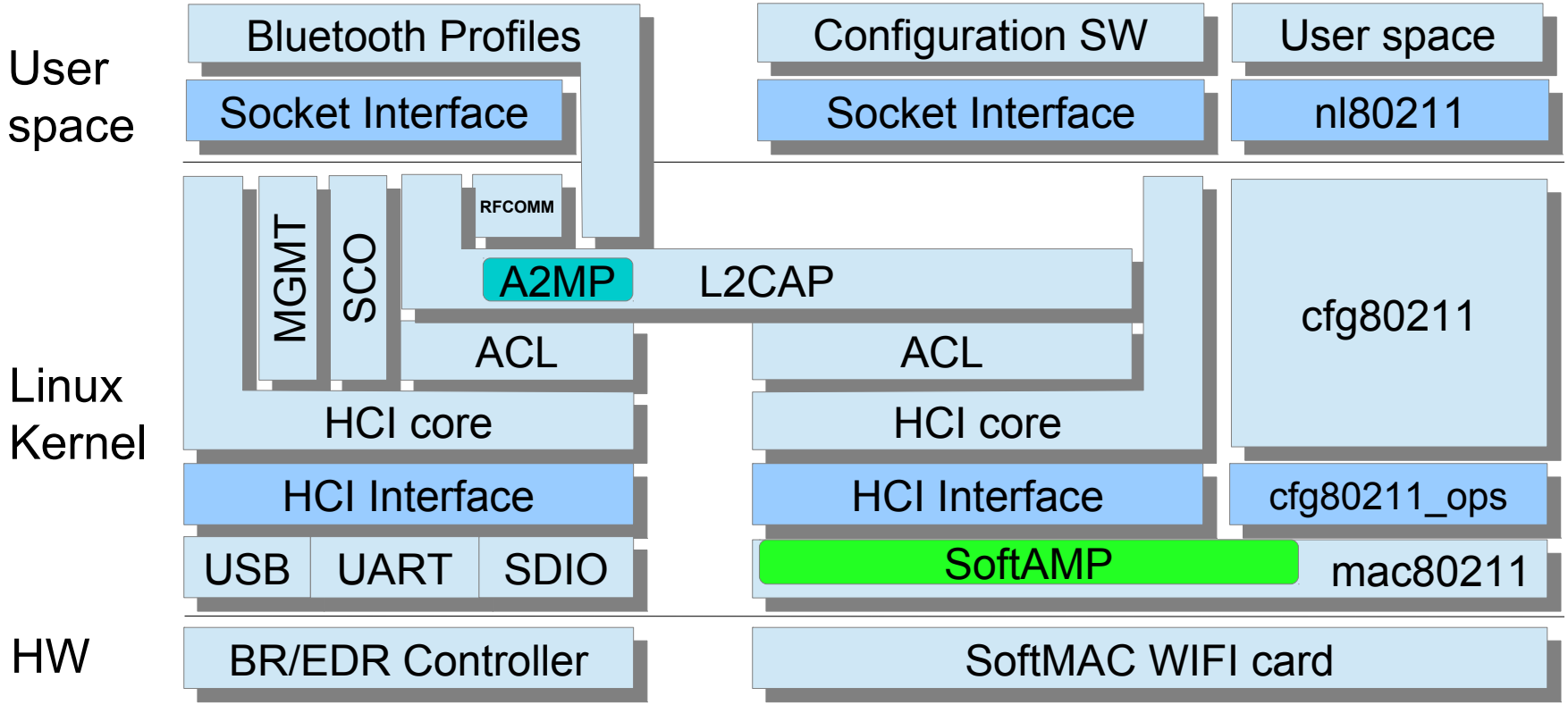
Upstream support

- A2MP – kernel 3.5
- Basic AMP configuration – kernel 3.7
- HCI core and L2CAP – kernel 3.8 (currently in bluetooth-next)
- L2CAP channel move full support – kernel 3.8 – 3.9

BlueZ and SoftMAC wireless card



BlueZ and SoftAMP



SoftAMP tasks

- Implement registration of Bluetooth HCI AMP device
  - Need to provide convenient way to start/stop SoftAMP.
- Handle HCI commands and events.
  - Process HCI commands and send events.
- Security – auth, assoc, 4 way handshake
- Data management
  - Send: remove HCI header, add LLC/SNAP, add IEEE802.11 header.
  - Receive: remove IEEE802.11 header, remove LLC/SNAP header, add HCI header.
- Channel management and bandwidth sharing

Security

- 4 way handshake derives PTK from PMK
- PMK is derived from Bluetooth Link Key (LK)
  - Derive Generic AMP Key (GAMP) from LK
    - GAMP_LK = HMAC-SHA-256(LK||LK, 'gamp', 32)
  - Derive Dedicated AMP Link Key which is PMK
    - PMK = HMAC-SHA-256(GAMP_LK, ‘802b’, 32)

Methods of implementing SoftAMP

- In-kernel
- User-space driven

Common parts of SoftAMP

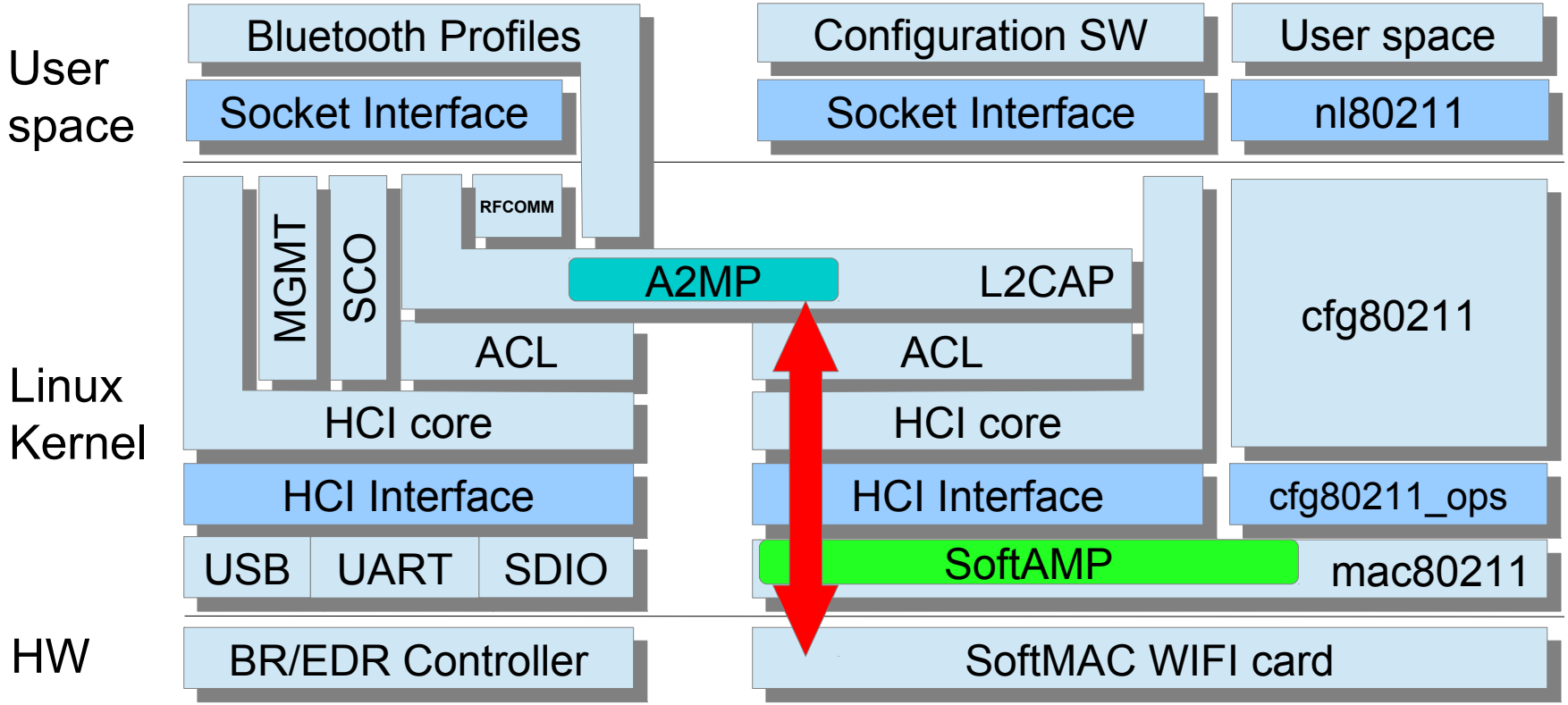
- Make SoftAMP a part of mac80211 (selectable?)
  - Any SoftMAC card might be used
- Implemented as a new virtual interface type in mac80211
  - SoftMAC drivers shall support this new type
- Data packets goes directly between bluetooth and wireless stacks

In-kernel SoftAMP

- As much as possible is done in the kernel
  - Even 4-way handshake?
- User-space knows very little that High Speed is used
  - Used mostly for enabling / disabling.
- SoftAMP can be created with any standard tool like:

```
$ iw phy phy0 interface add softamp type softamp
```

In-kernel SoftAMP interfaces



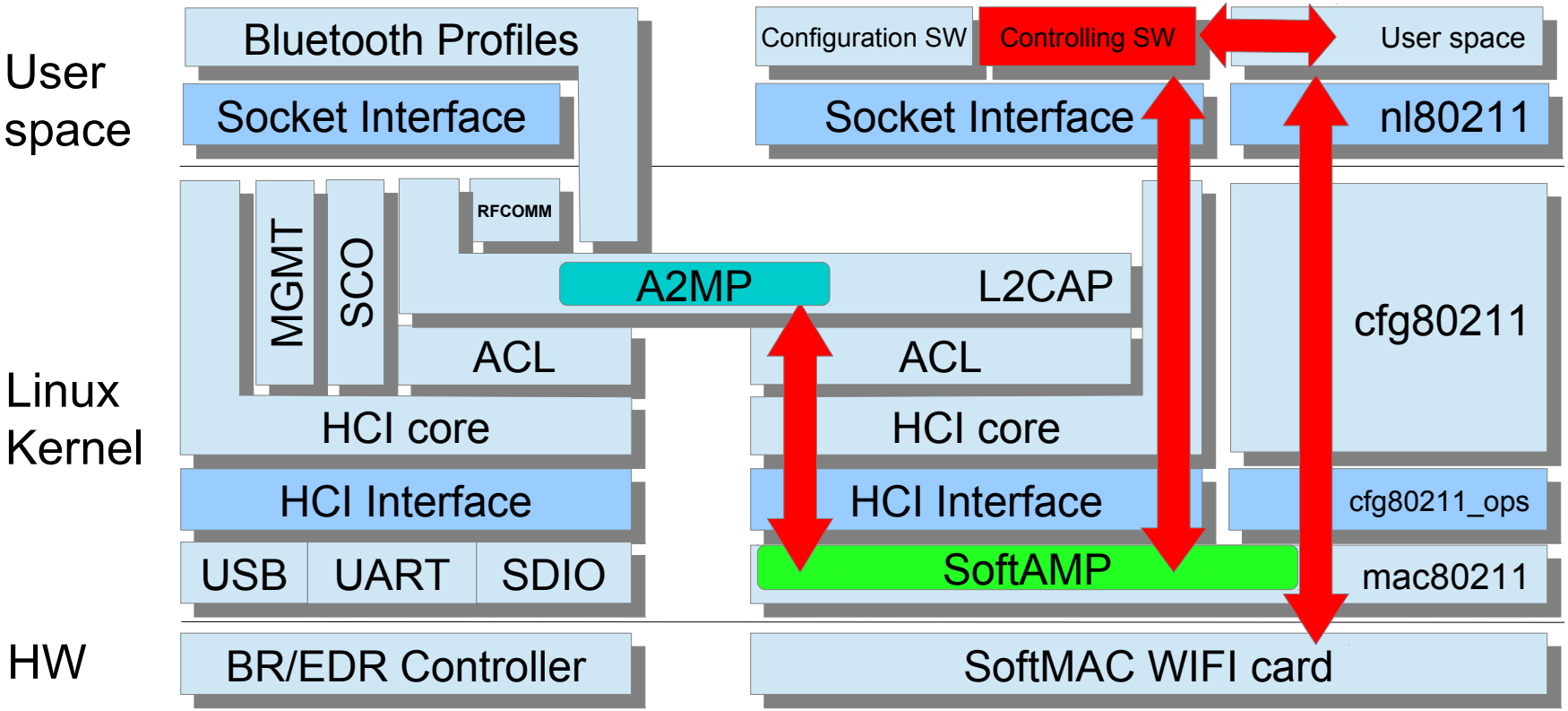
User-space driven SoftAMP

- User space is controlling many aspects of High Speed setup
- SoftAMP is created and managed through special nl80211 API messages.
- All data except for data frames are sent to wireless user space components (wpa_supplicant).

User-space driven SoftAMP. Johannes Berg ideas:

- cfg80211 add/remove interface can get a new interface type, to also create/remove the hci
- ifup/down will be hci up/down
- new nl80211 API to start/stop the AMP (start/stop beaconing)
- new nl80211 API to RX/TX HCI messages so userspace can handle parts of the HCI protocol
- new nl80211 API to RX/TX AMP PDUs, with the given AMP_PROTO, so userspace can handle 4-way-handshake etc.

User space driven SoftAMP interfaces



Abbreviation

- A2MP – AMP Manager Protocol
- ACL – Asynchronous Connection-oriented link
- AMP – Alternate MAC / PHY Controller
- L2CAP – Logical Link Control and Adaptation Protocol
- PAL – Protocol Adaptation Layer
- SCO – Synchronous Connection-Oriented link

Questions?


TURE INTEL LINUX WIRELESS
OP
CS
YOCTO
CONNMAN
XEN
GUPNP
KVM
POKY
OFONO
LINUX KERNEL
SYNCEVOLUTION
SIMPLE FIRMWARE INTERFACE (SFI)
ENTERPRISE SECURITY INFRASTRUCTURE
INTEL OPEN SOURCE
TECHNOLOGY CENTER