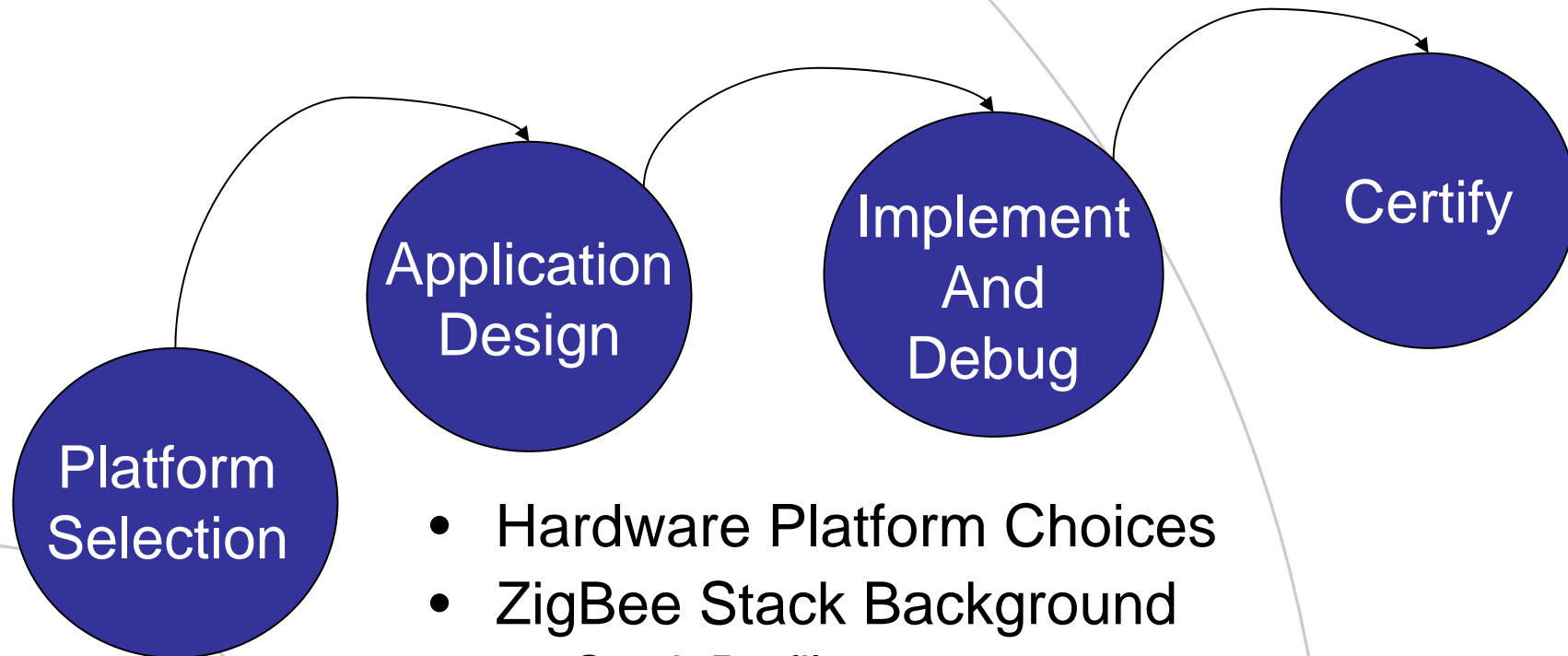


Demystifying the Building of ZigBee Applications

Andy Wheeler
CTO
Ember Corporation

Outline

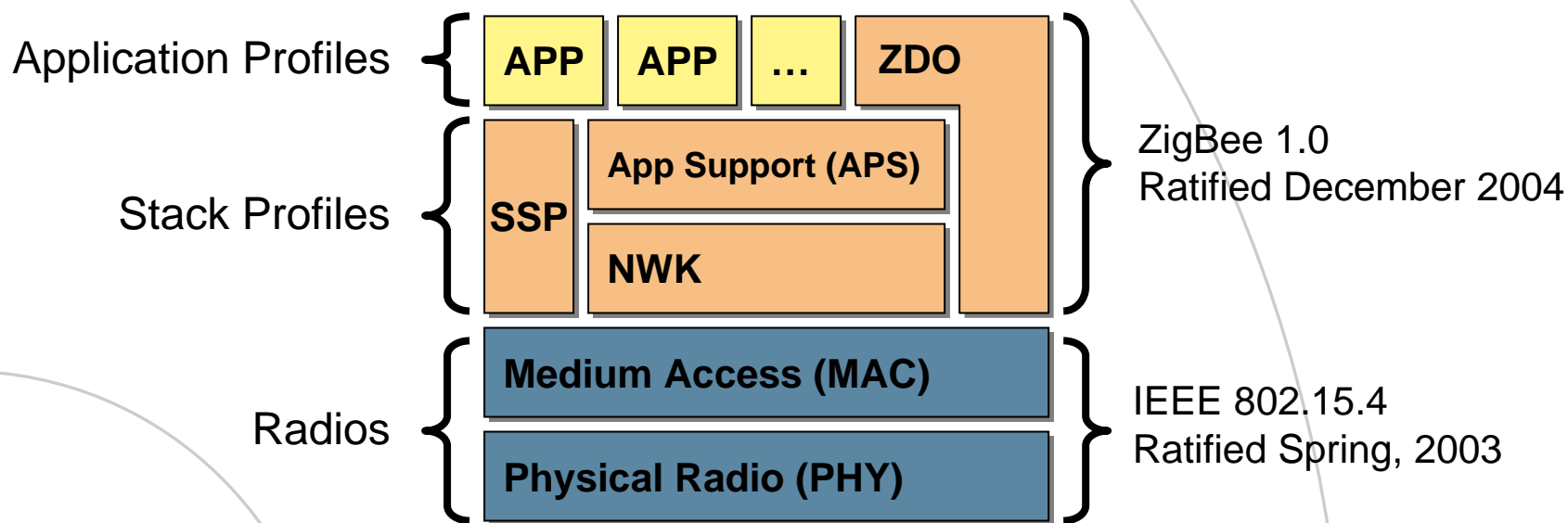


- Hardware Platform Choices
- ZigBee Stack Background
 - Stack Profiles
 - Application Profiles
- Interoperability/Compliance
- Debugging ZigBee Applications
- Vendor Selection

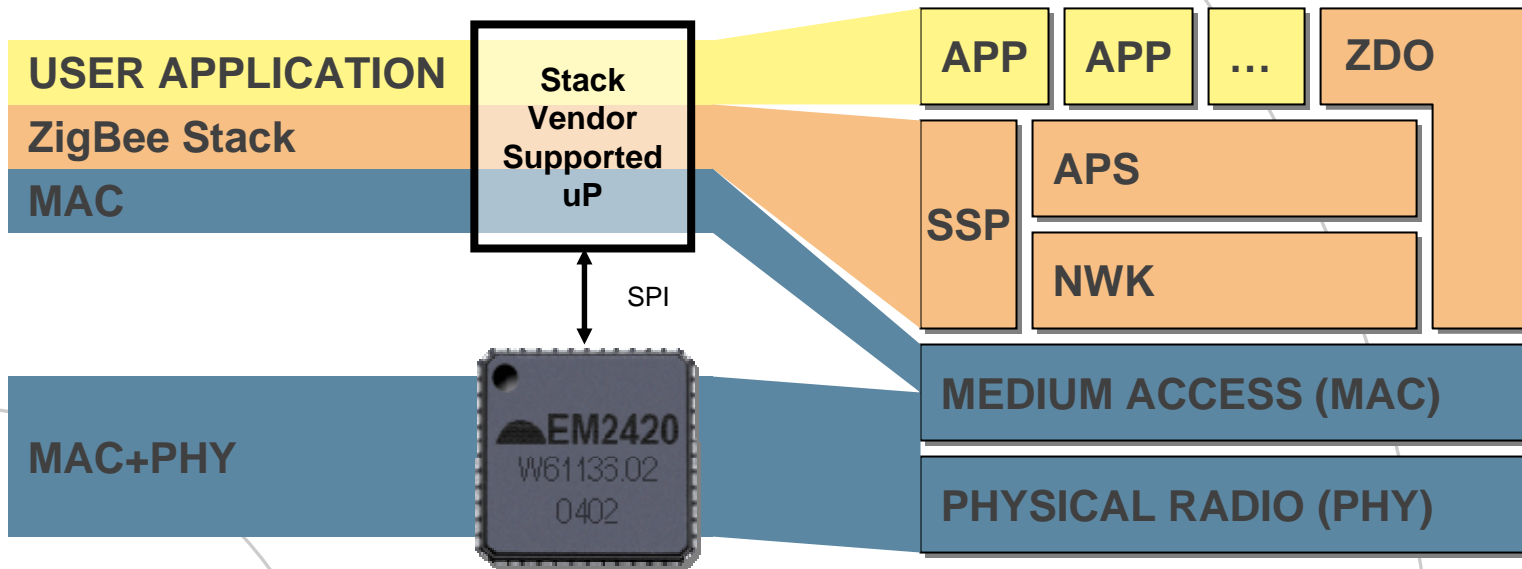
QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

Hardware Platform Types

ZigBee Stack Architecture

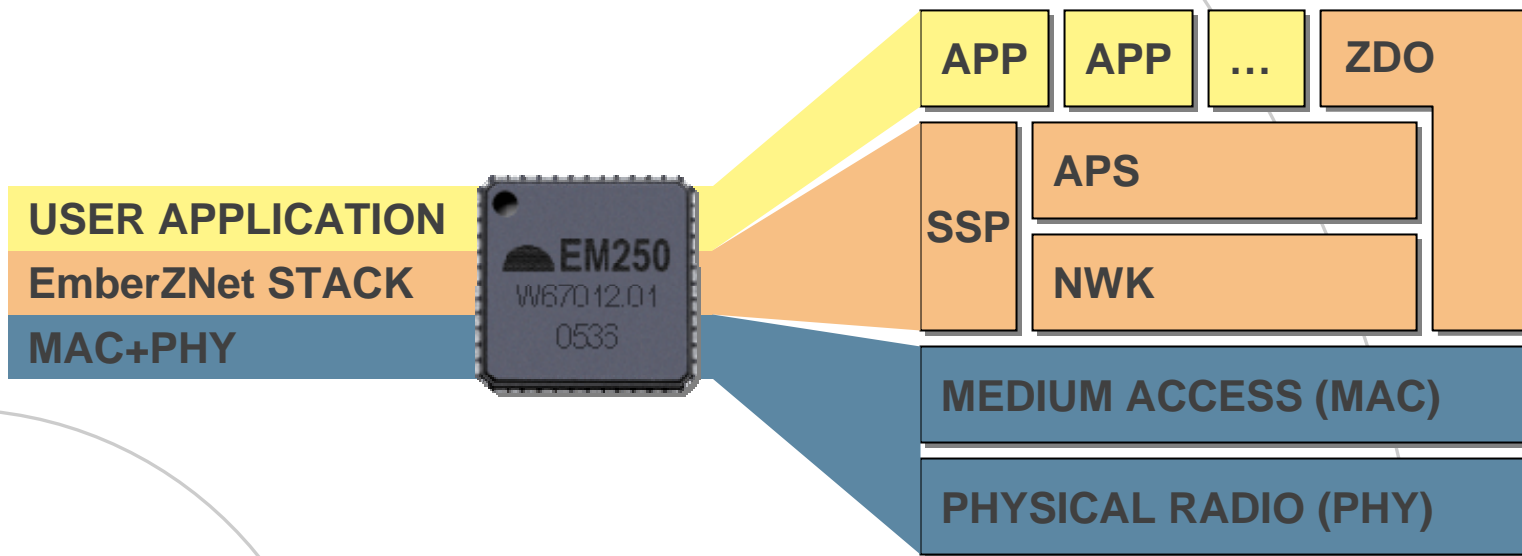


RF Only 802.15.4 Radio



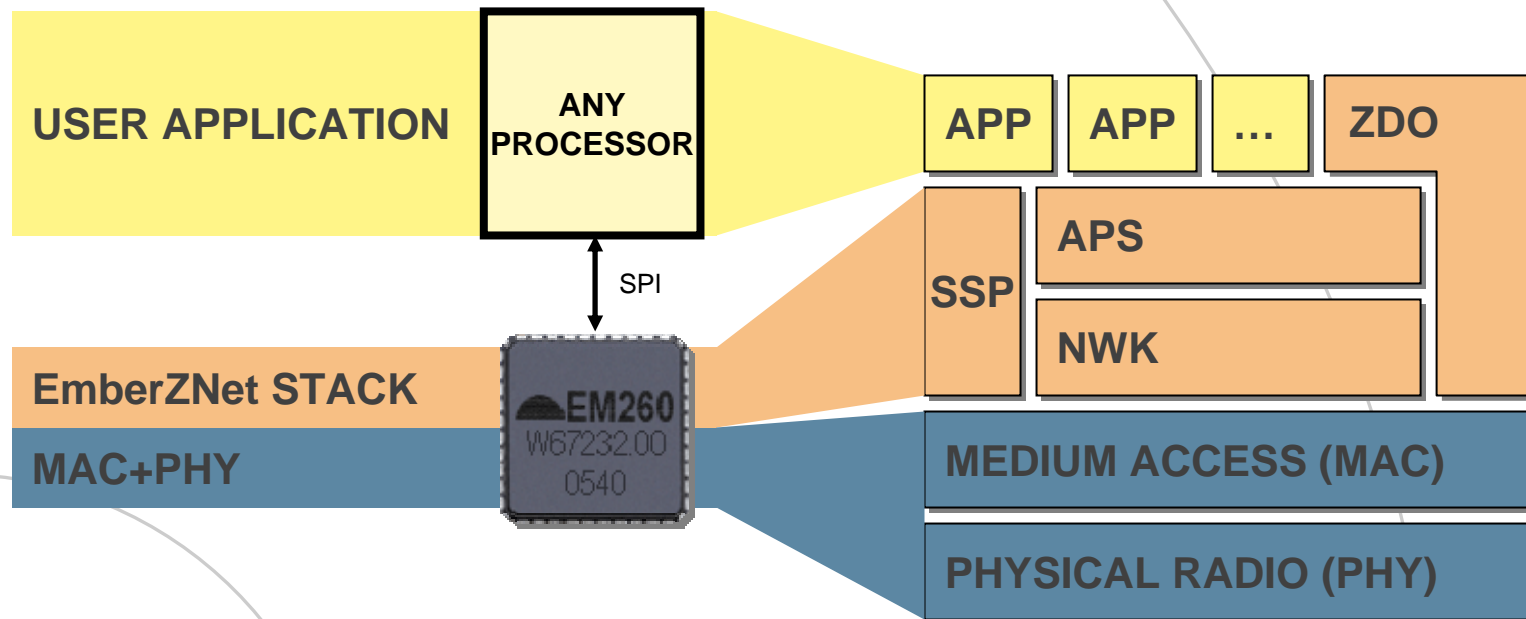
- Most common solution on the market today
 - Many vendors (not with common radio interfaces however)
- Attach to any micro supported by stack vendors
- Single micro must handle both application and stack - development challenge

System-On-A-Chip Approach



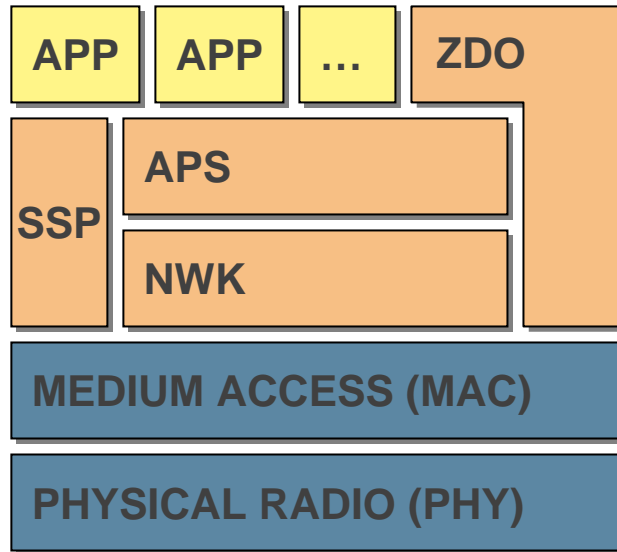
- All functions for a ZigBee device in a single chip
- Highly integrated, lower component count, micro/peripherals optimized for ZigBee, lowest BoM
- Requires use of the processor core and peripheral mix chosen by vendor

Network Processor Approach



- Separates the application from the stack/radio - most flexible approach
- Allows processor to be selected independent of ZigBee stack/radio
- Fewer integration problems/real-time problems
- Increases component count and current consumption over SoC approach

ZigBee Modules



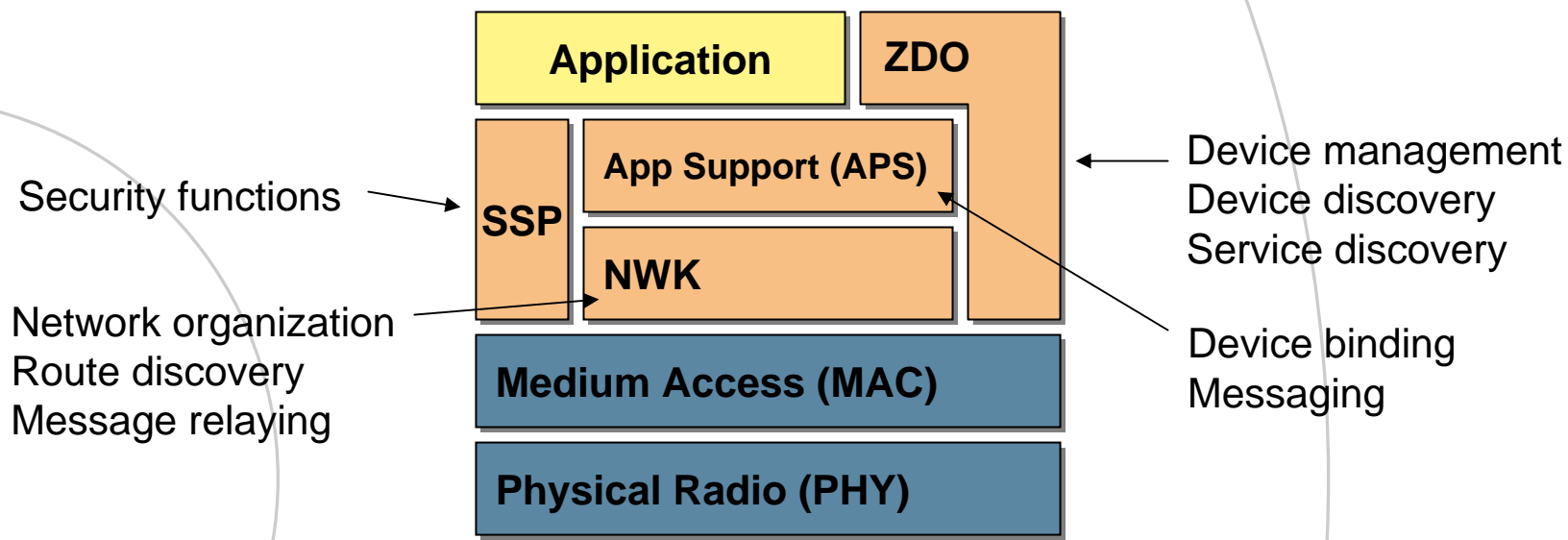
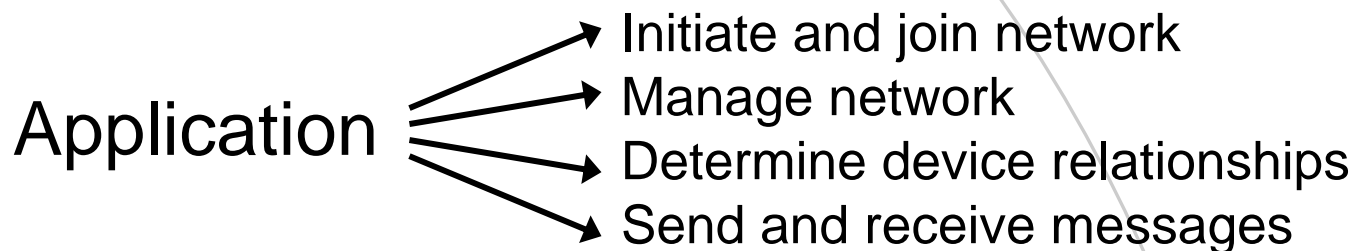
- Wide variety of module manufactures now offering SoC, RF-only, and network processor modules
- Reduce design time and RF skills needed
- Higher cost and often less flexible due to integrated antennas
- Appropriate when the volume is lower or when RF test is prohibitive

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

What's in the ZigBee Stack?

ZigBee Functional Overview



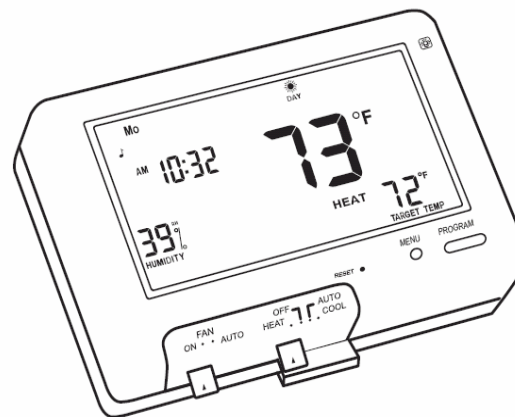
ZigBee Concepts: Device Types

ZigBee Type	Notes
ZigBee Coordinator (ZC)	Exactly 1 per network
ZigBee Router (ZR)	No duty cycling available
ZigBee End Device (ZED)	Does not relay packets



ZigBee Concepts: Endpoints

- An endpoint is a virtual address within a ZigBee device – up to 255 endpoints per device
- Endpoint 0, and endpoints 240-255 are reserved for special functions; endpoints 1-239 are available for user applications
- Each endpoint supports exactly one application profile – but different endpoints can support different application profiles



Example device endpoints:

- 0 – Network management (ZDO)
- 1 – Home Automation/HCL
- 100 – Vendor extensions

ZigBee Concepts: Cluster IDs

- Cluster IDs are used to define message types
- Cluster IDs and their associated message structure are defined by the application profile



Example Cluster IDs for a private application profile:

ID 0 : SWITCH_ON

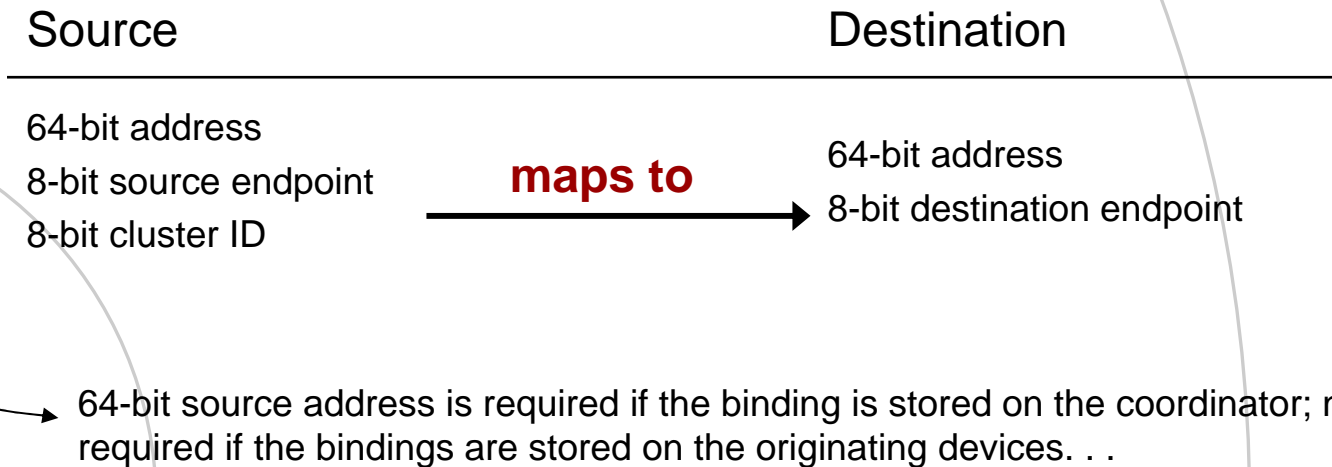
ID 1 : SWITCH_OFF

ID 2: SWITCH_DIMMER_SETTING

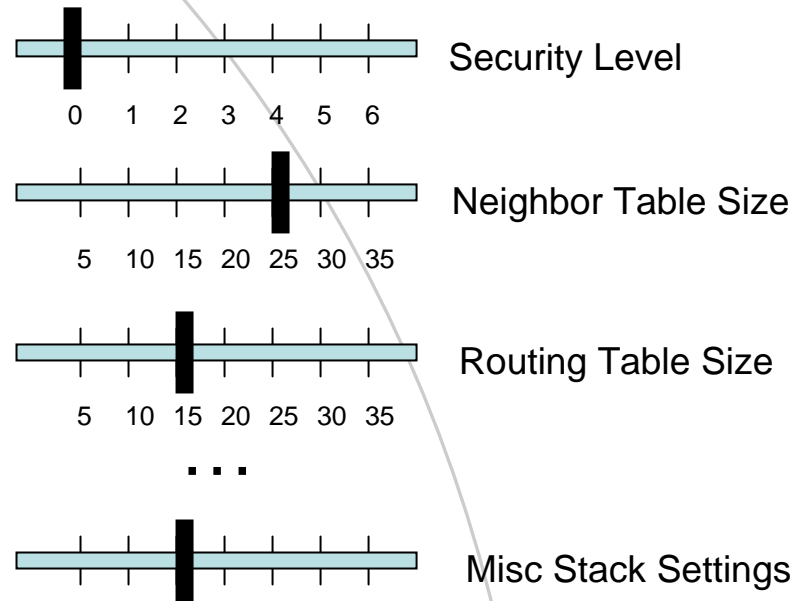
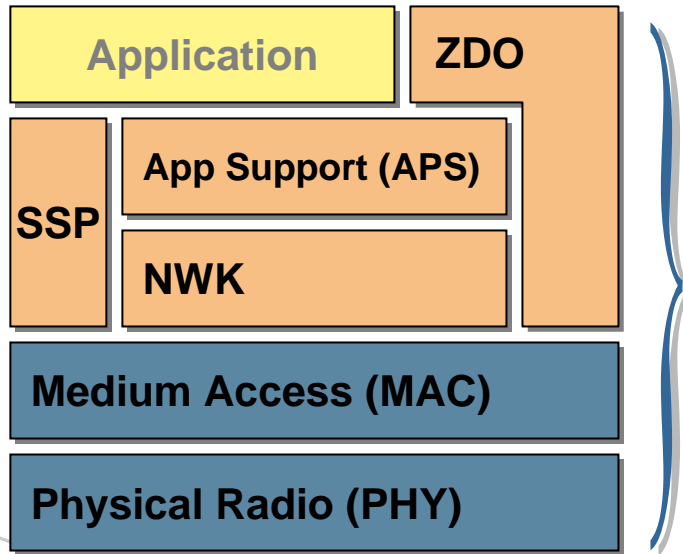
And so on...

ZigBee Concepts: Bindings

- Bindings define device relationships by matching an incoming endpoint and cluster id to an outgoing endpoint and cluster id. Bindings may be stored in the routing device or in the coordinator. Bindings are used for *indirect addressing* of messages:



ZigBee Stack Profiles



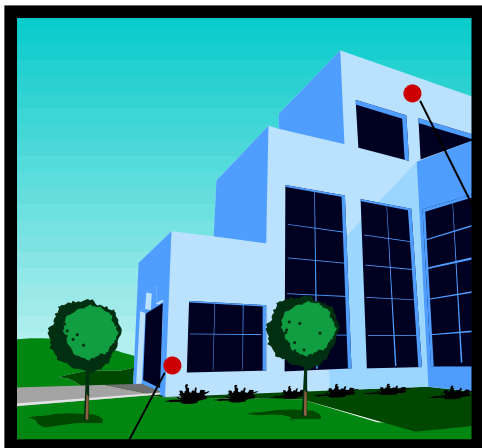
- Stack Profiles define the settings of various “knobs” in the ZigBee stack
- Allows tradeoffs to be made between stack performance/characteristics and resource use
- Luckily, only a few stack profiles exist...

HC Stack Profile



- Home control oriented stack, aimed at simpler, smaller networks
- Key Characteristics
 - Optimized for small networks
 - Security uses a single key for the network, installed out of band or unsecured over the air
 - Bindings in a coordinator device (saves memory on other devices)

CII Stack Profile



- Commercial and industrial targeted stack
- Key characteristics
 - Much more scalable addressing and routing than HA stack
 - Aggregation and multicast
 - More robust security
 - Distributed bindings and less reliance on network coordinator

Application Profiles

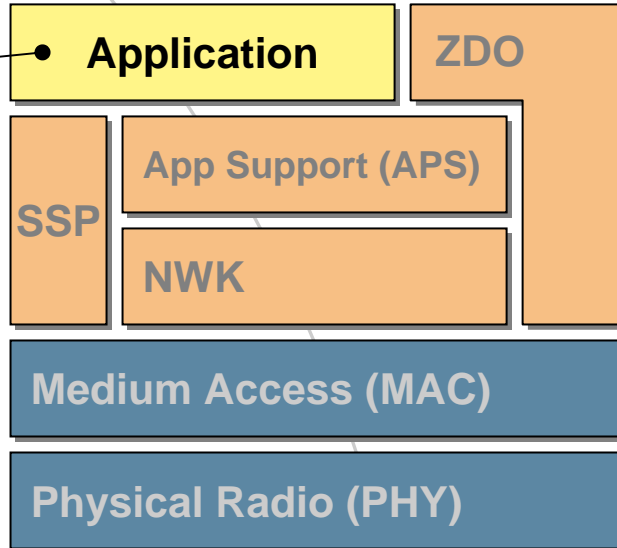


Clusters

0: off
1: on
2: scene 1
3: scene 2

Clusters

0: fan off
1: fan on
2: temp set
3: time set

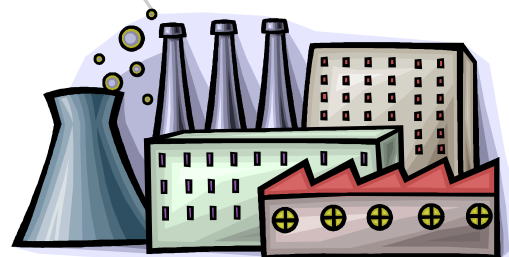


- Application profiles define what messages are sent over the air for a given application
- Devices with the same app profiles interoperate
- ZigBee publishes a set of public profiles, but vendors may create private ones as well (at the expense of interoperability)

Some Application Profiles



- Home Automation
 - Built on HC stack profile
 - Defines set of devices used in home automation
 - Lightswitches
 - Thermostats
 - Window shade
 - Heating unit
 - etc

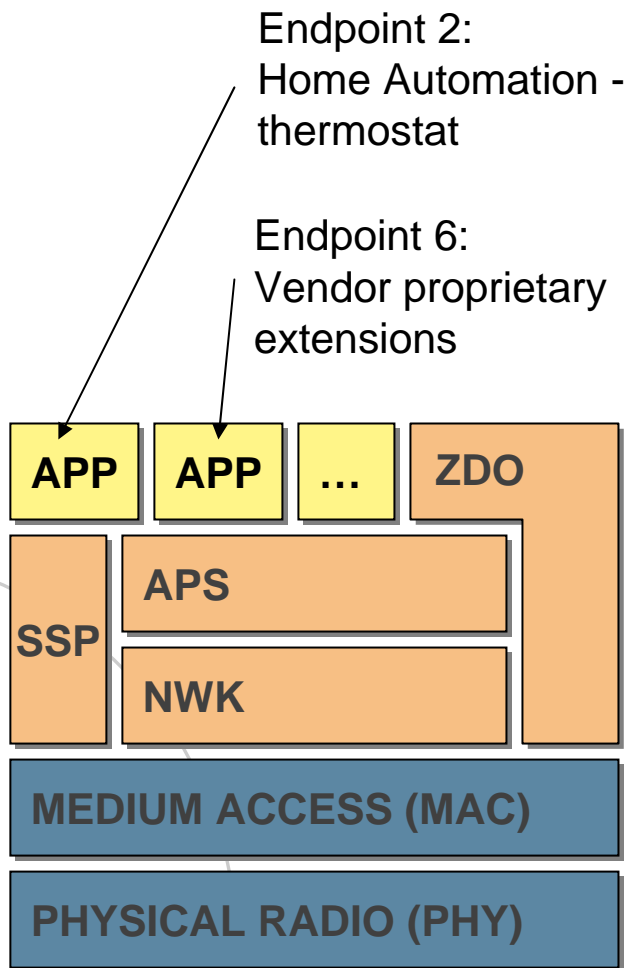


- Industrial Plant Monitoring
 - Built on CII stack profile
 - Consists of device definitions for sensors used in industrial control
 - Temperature
 - Pressure sensors
 - Infrared
 - etc

More Application Profiles

- Multiple profiles at various stages of progress
 - Commercial Building Automation
 - Building control, management, and monitoring
 - Heating, Ventilation, and Air Conditioning
 - Cross-market HVAC control
 - Automated Meter Reading
 - Addresses utility meter reading
 - Wireless Sensor Networks
 - Very low power unattended networks
- Vendors may form new profile groups within ZigBee and/or propose private profiles for consideration

Multi-Profile Devices

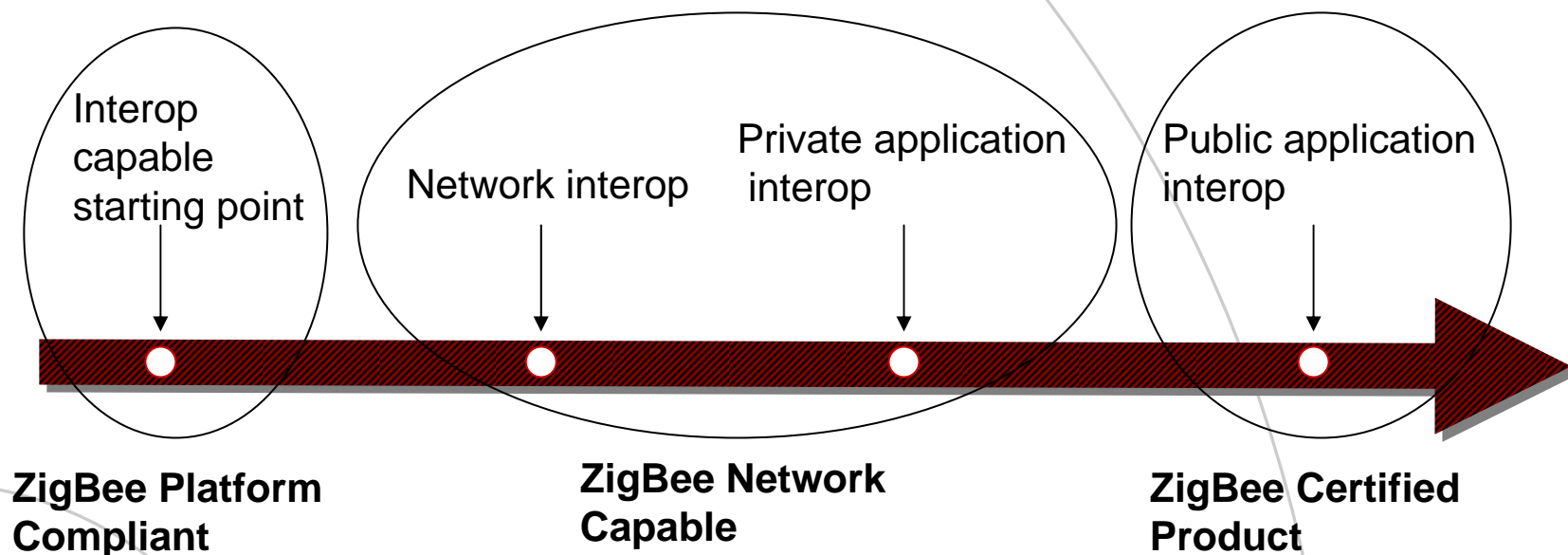


- Vendor devices may implement multiple profiles
- Additional application profiles live on different endpoints within the device
- Allows creation of vendor specific extensions
- All endpoints on a single device must use of the same stack profile

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

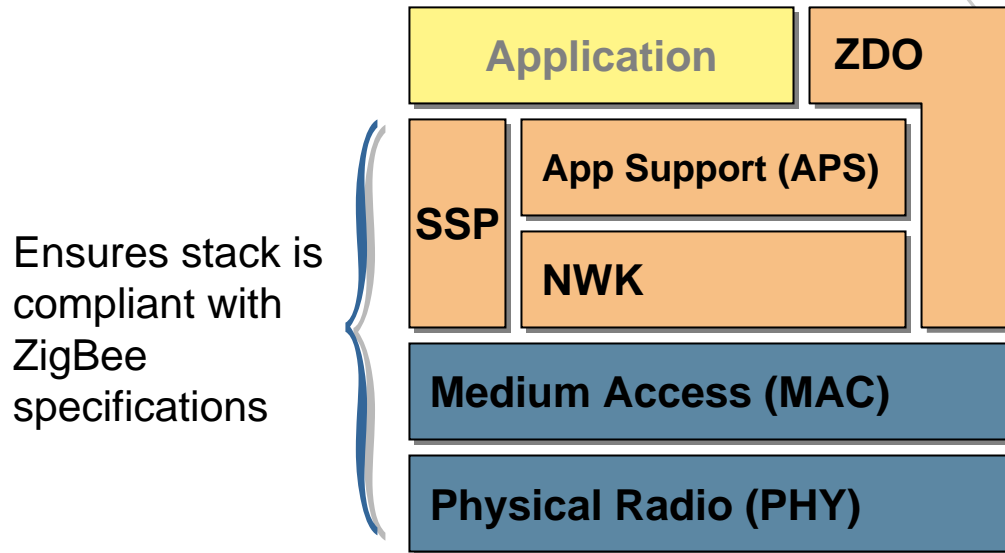
Understanding ZigBee Interoperability and Compliance

Interoperability Levels



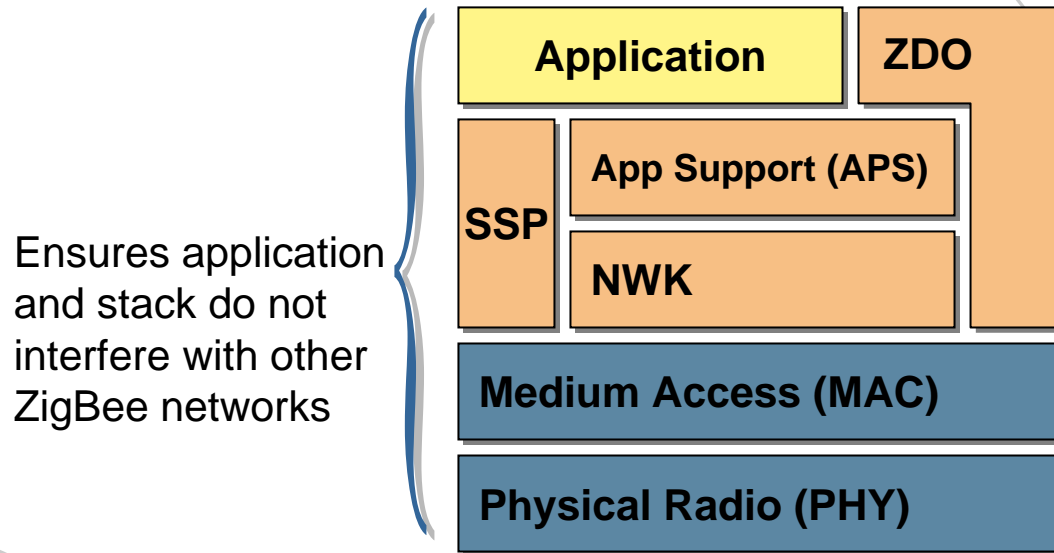
- Devices built on ZigBee do not necessarily interoperate
- Wide spectrum of interop choices
- It's a designer choice on level of vendor interoperability to support
- No matter which level you choose, you must be at least an Adopter class member to use ZigBee IP in a product- even if you don't use the ZigBee name

ZigBee Compliant Platform



- Platform certification - ensures all parts of the stack other than the application are compliant with a particular stack profile
- Developers should be sure any platform they consider has this
- Does not imply interoperability at the application layer, only ensures that you can build interoperable devices if you choose
- Obtained by platform (stack or module) providers

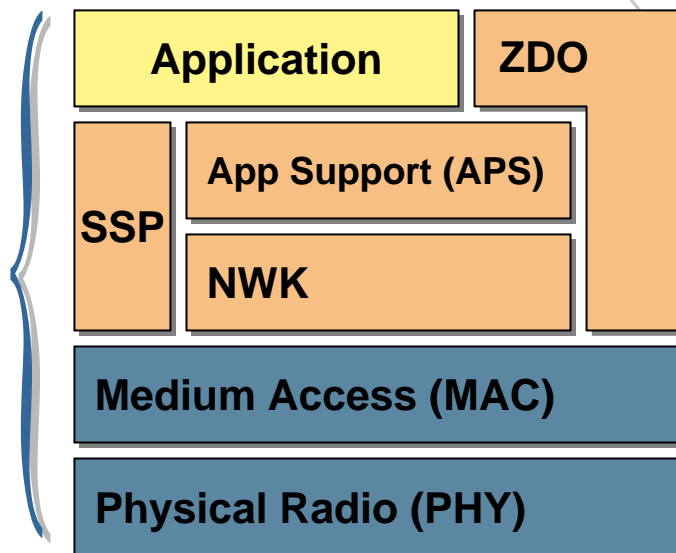
ZigBee Network Capable



- Allows a vendor who builds a product with a ZigBee stack, but who does not follow a published application profile to ensure their product does not harm other ZigBee networks
- Permits the use of ZigBee in literature
- Does not imply any interop between multiple vendors devices

ZigBee Certified Product

Ensures application
conforms to a
specific public
application profile



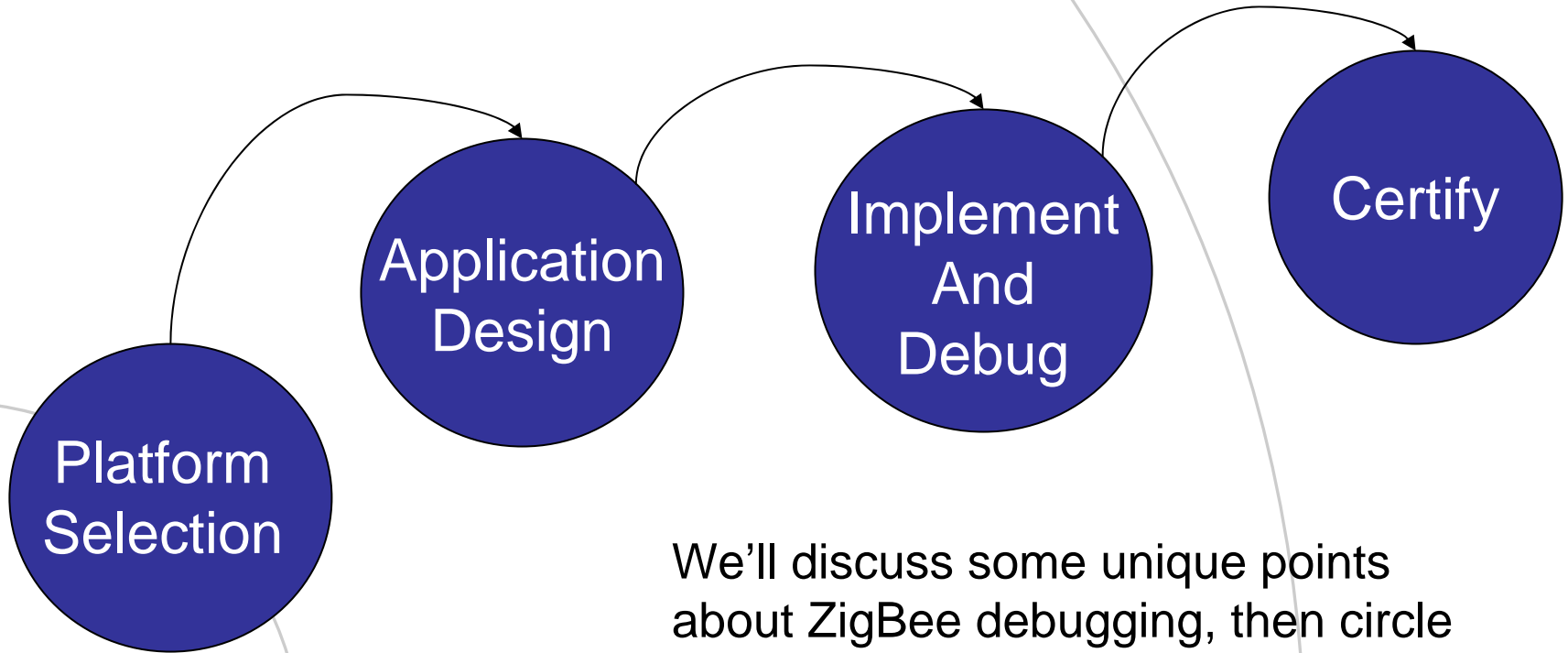
QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

- Guarantees interoperability between products all running a specific application profile
- Allows product vendor to use ZigBee language and logos on their product
- Does not preclude vendor extensions from being present - only ensures a given profile is followed

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

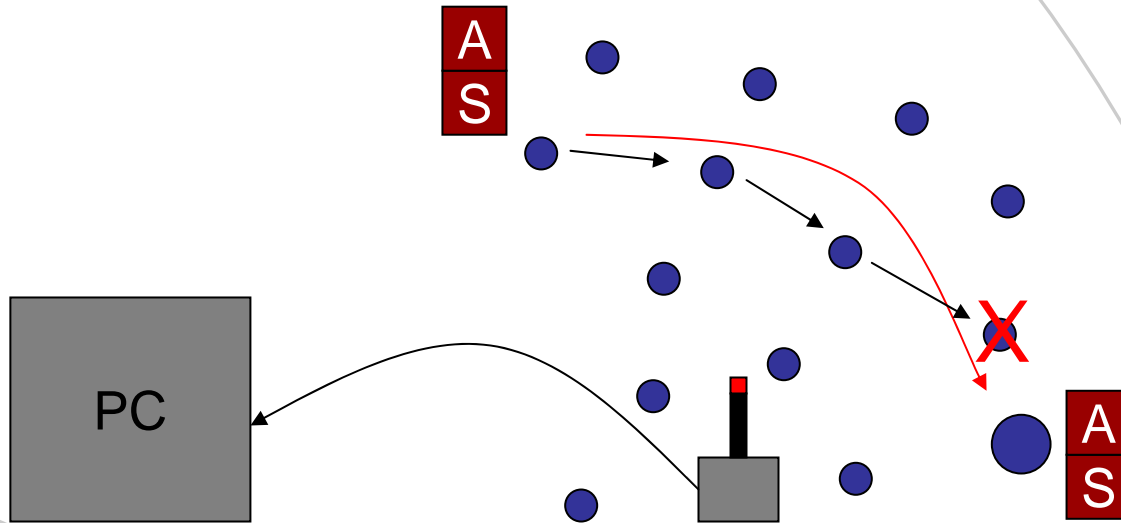
Designing Your Own Application

ZigBee Development



We'll discuss some unique points about ZigBee debugging, then circle back and close with some thoughts on platform selection

Wireless Application Debugging



- Unique challenges in debugging wireless applications
 - Resource constraints
 - Halt/Step style of debugging doesn't always work
- Common tool is a packet sniffer

Typical Packet Sniffer

Frame Display - Bookmarked Home Controls.cfa

File Edit View Summary Decode Radix Character Event Filter Options Window Help

Filter: Include frames where prob

Frame 281: (Host) Len=27

- MAC:
 - MHR:
 - Radio Supplied Information:
- NWK:
 - Frame Control:
 - Routing Fields:
 - Destination Address: 0x0000
 - Source Address: 0x030e
- APS:
 - Frame control:
 - Cluster Identifier: 0x02
 - Source Endpoint: Application 0x01
 - Application Framework:
 - Transaction Count: 1
 - Frame Type: Key Value Pair (KVP)
 - Home Controls and Lighting:
 - Transaction Sequence Number: 1
 - Command Type: SET
 - Attribute Data Type: Unsigned Integer 8-bit
 - Attribute ID: Dim Bright
 - Fade Time: 1 Second(s)
 - Brightness State: 88

All Protocols		MAC	NWK	APS	
Application Framework		ZDP	Home Controls and Lighting		
B...	Frame#	Trans Seq #	Cmd Type	Data Type	Attrib ID
C	268	0	SET	UINT8	On Off
273	0	SET	UINT8	On Off	
S	281	1	SET	UINT8	Dim Bright

01100001 10001000 00010010 11110000 00101001
 00000000 00000000 00001110 00000011 00000100
 00000000 00000000 00000000 00001110 00000011
 00010100 00000010 00000001 00010001 00000001

61 88 12 f0 29 00 00 0e 03 04 00 00 00 0e 03 14
 02 01 11 01 11 00 01 58 ff 07 58

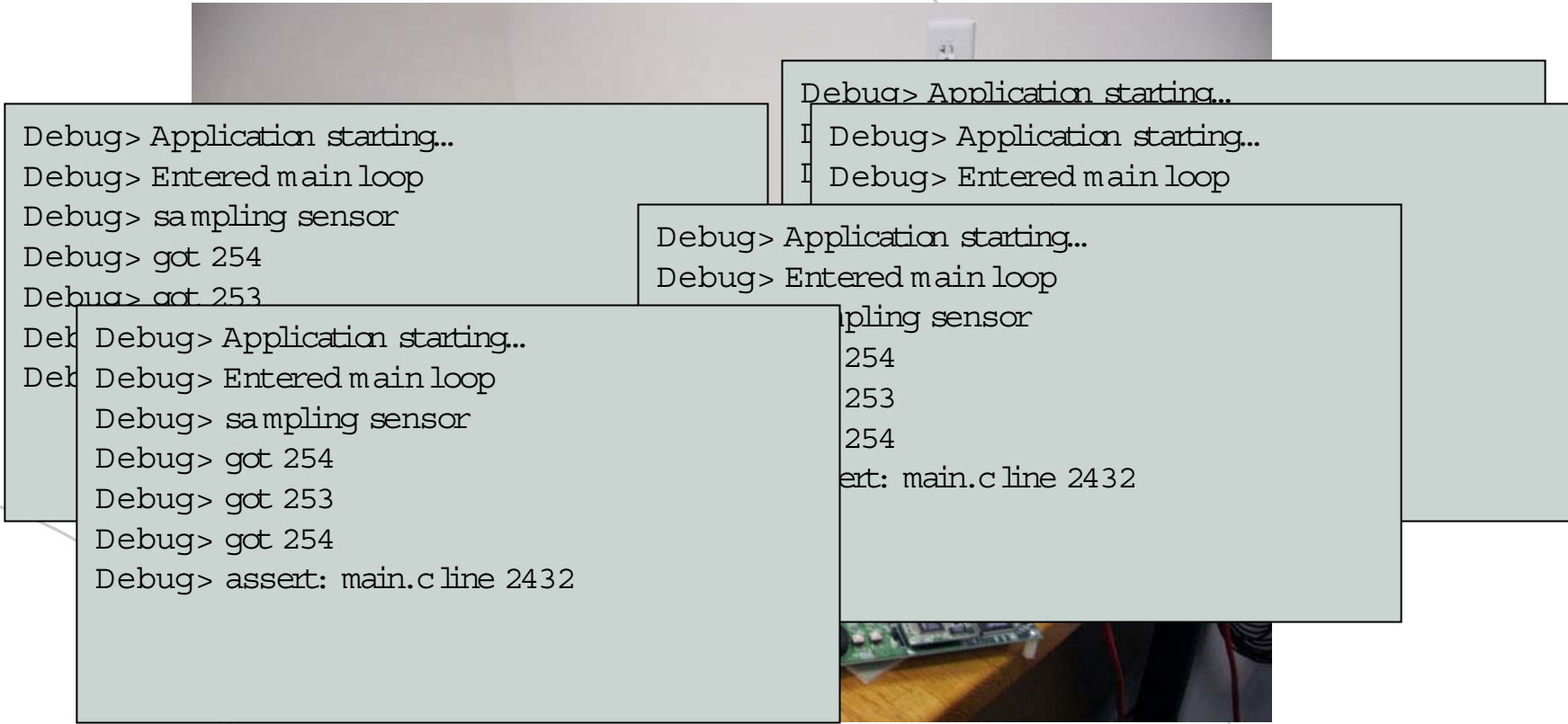
CHARACTER

Event 5,412 of 26,563 (Frame 281) 11/24/2004 4:03:31.3457 PM

Source	ASCII	Hex	Dec	Oct	Binary	Errors
Host	SOH	01	1	1	00000001	

For Help Press F1

Scaling Debugging is a Challenge



```
Debug> Application starting...
Debug> Entered main loop
Debug> sampling sensor
Debug> got 254
Debug> got 253
Debug> Application starting...
Debug> Entered main loop
Debug> sampling sensor
Debug> got 254
Debug> got 253
Debug> got 254
Debug> assert: main.c line 2432
```

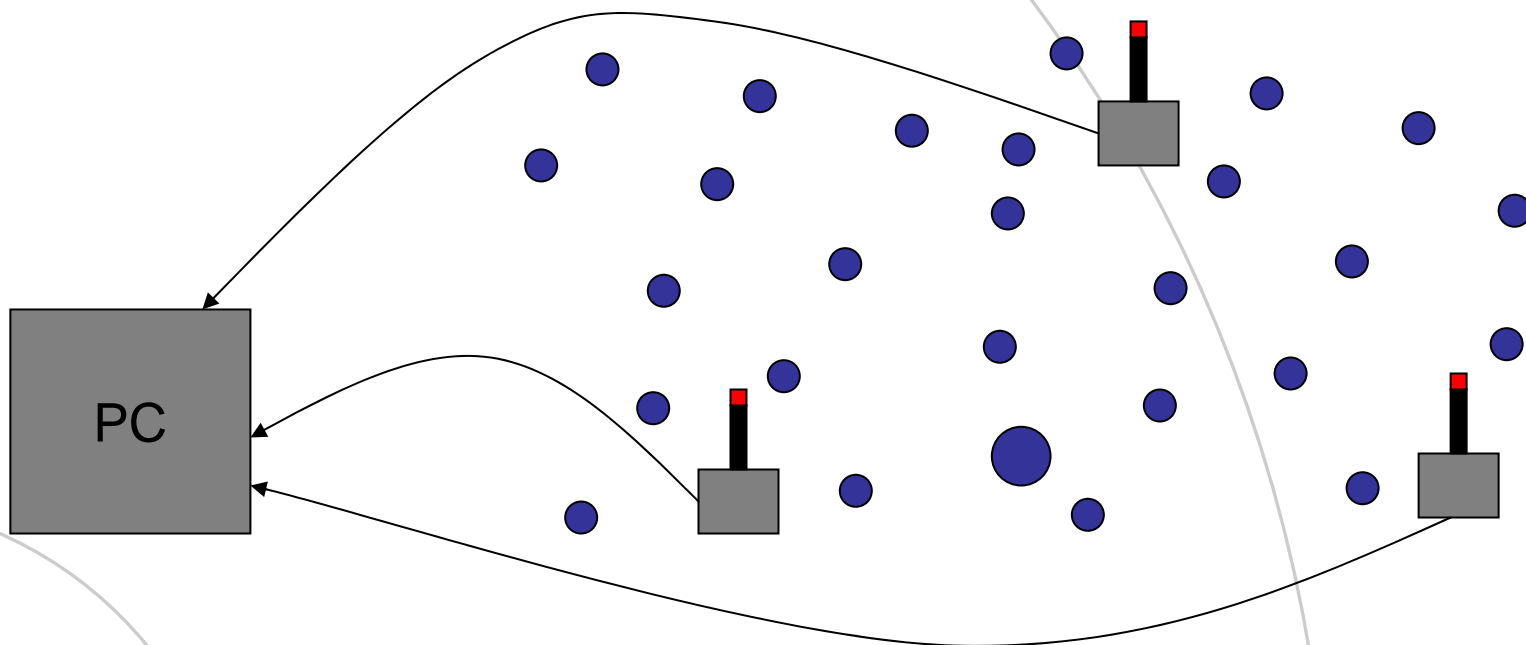
```
Debug> Application starting...
Debug> Application starting...
Debug> Entered main loop
```

```
Debug> Application starting...
Debug> Entered main loop
```

```
sampling sensor
254
253
254
ert: main.c line 2432
```

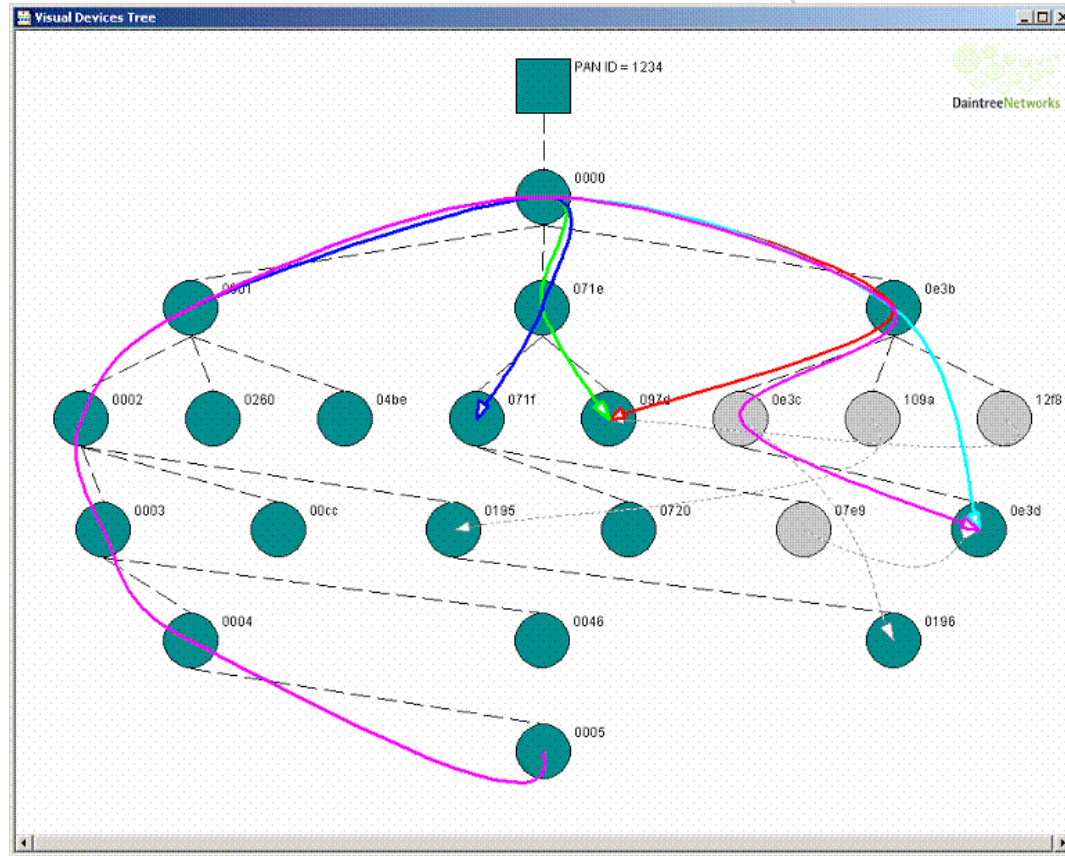
- Standard practices don't scale
- Physical separation is a challenge
- Information overload

Scaling up packet sniffing



- Add multiple sniffers (some software supports)
- Start to reach information overload

Network Analyzers



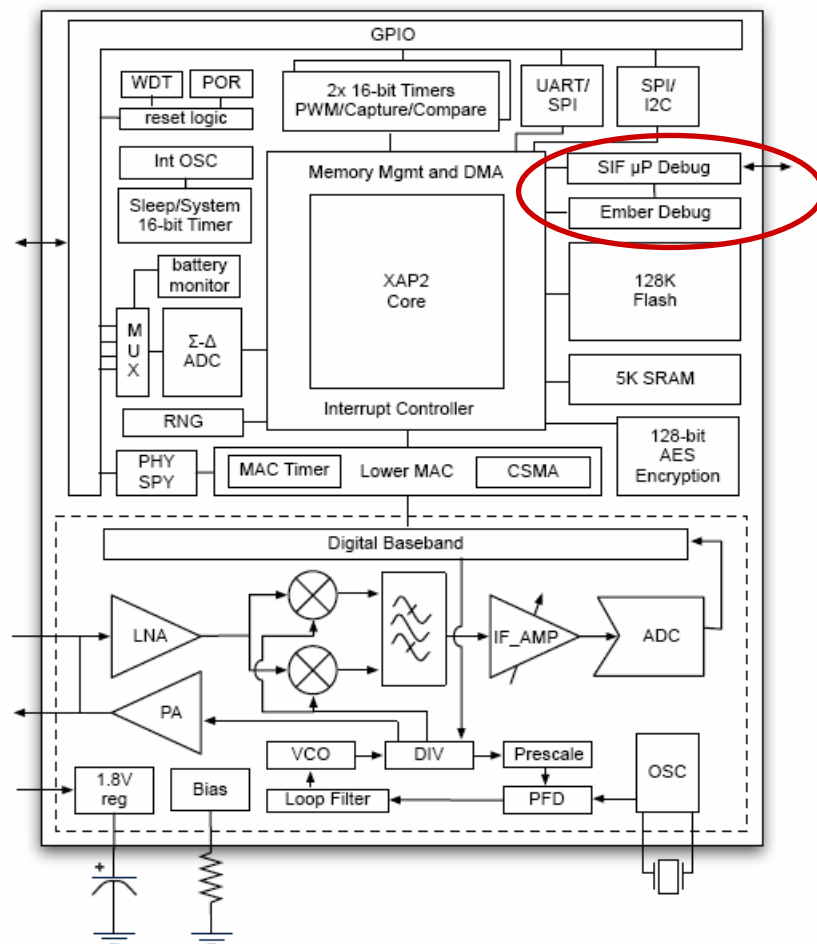
- Advanced capabilities built on packet sniffers

Comprehensive Dev System

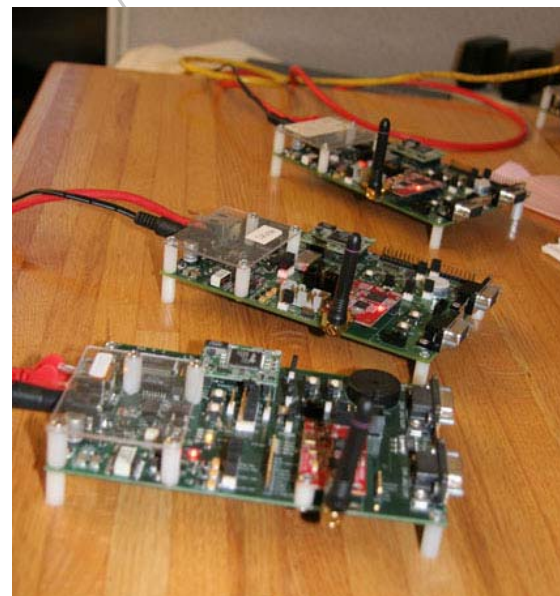
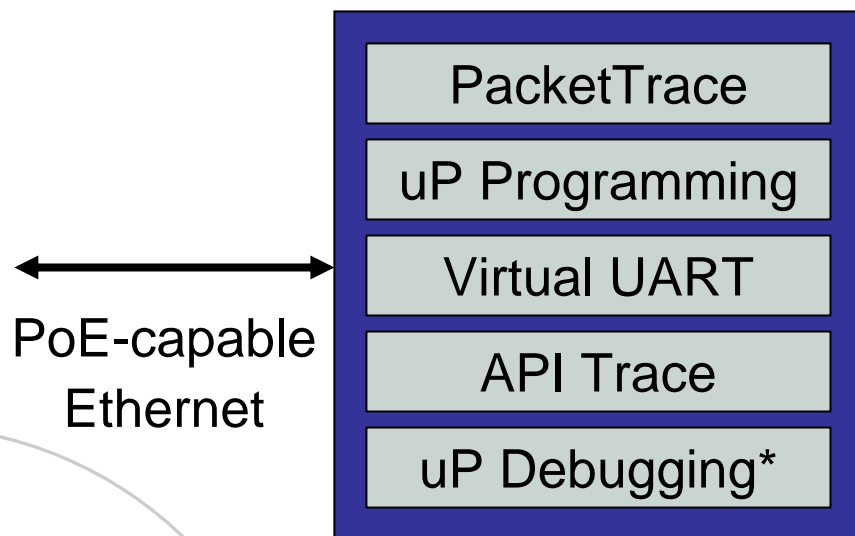
- Need a scalable method of extracting information from nodes... with minimal intrusion
 - Not just packets, but all debug information
- Need a set of tools to analyze and manage that data

On-chip Development Support

- InSight Port
 - High speed backchannel
 - Virtual UART
 - PacketTrace Module
- Protection Features
 - Memory Protection
 - Stack Overflow Detection



Example Debugger: InSight Adapter



- InSight Adapter provides comprehensive microprocessor and network debugging over a single cable

InSight Desktop

The screenshot displays the InSight Desktop application window titled "Live capture*". The interface includes a menu bar (File, Edit, Window, Help) and a toolbar with icons for Filters, Stream, Nodes, State, and Devices. A "Filter expression:" field is present with "Insert field..." and "Insert operator..." buttons, and "Apply" and "Clear" buttons below. A list of filter options is shown, with several checked: "Show group: APS Group", "Show group: Transport Datagram", "Show group: 802.15.4 Association", "Show group: Broadcast", "Show group: Unicast", "Show group: 15.4 MAC group", "(fifteenFour.source == 0)", and "IsPresentapitrace.apiTraceType".

The main display area shows a network map with nodes labeled with MAC addresses: 0x0000 000D6F000004BA14, 0xBF7E No EUI164, and 000D6F000005A0A1. A status bar at the top right indicates "Map: T=2,718.265741s".

Below the map is a tabbed interface with "Progress", "Shell", and "Event Log" tabs. The "Event Log" tab is active, showing "Event count: 845, captured: 9/30/05 2:15 PM". A table of events is displayed:

Time	Du...	Group	Summary	Source	D...
2,7...	0.0...	BRO	APS Ack Indirect	0000	FF
2,7...	0.0...	BRO	APS Ack Broadcast	BF7E	FF
2,7...	0.0...	API Trace	emberIncomingMessageHandler	ember10	
2,7...	0.0...	API Trace	emberSendLimitedMulticast	ember10	
2,7...	0.0...	API Trace	emberIncomingMessageHandler	ember10	
2,7...	0.0...	BRO	APS Data Broadcast	0000	FF
2,7...	0.0...	BRO	APS Data	BF7E	FF
2,7...	0.0...	API Trace	emberIncomingMessageHandler	ember10	
2,7...	0.0...	API Trace	emberSendLimitedMulticast	ember10	
2,7...	0.0...	API Trace	emberIncomingMessageHandler	ember10	
2,7...	0.0...	BRO	APS Command Indirect	0000	FF
2,7...	0.0...	BRO	APS Data Indirect	BF7E	FF
2,7...	0.0...	API Trace	emberIncomindMessageHandler	ember10	

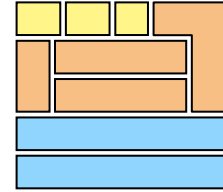
Below the event log, the "Event Detail" tab is active, showing details for the selected event: "Event: emberIncomingMessageHandler". The details include: "API Trace", "API Trace Type: emberIncomingMessageHandler (17)", "Message Type: 0x03", "Application Profile Id: 0xC00F", "Cluster Id: 0x11", "Source Endpoint: 0x01", "Destination Endpoint: 0x01", and "Unicast Options: 0x00".

Platform Vendor Selection

- Hardware
 - Platform type (RF-only, SoC, Network Processor, Module vs. Chip)
 - Power Consumption
 - RF Performance
 - 802.11/other 2.4GHz interference suppression
 - Output power/receive sensitivity
 - Debugging features supported by hardware



Platform Vendor Selection



- Software

- Top-level API

- ZigBee does not define an API- only logical behavior and over-the-air behavior

- Testing strategy

- Testing large-scale networking requires different QA mechanisms than standard embedded code

- Debugging Tools

- Management Interfaces

- Support for deployment/enterprise-connection software

- Documentation/Training