WIND RIVER

Education Services

Detail Architecture and Connectivity of Intel Gateway Solutions for IOT

Agenda

Architecture and Connectivity

- Architecture
- SRM Components
- Hardware Connectivity Options
- Software Connectivity Options
- Device Management

Objectives

By the end of this chapter you will be able to:

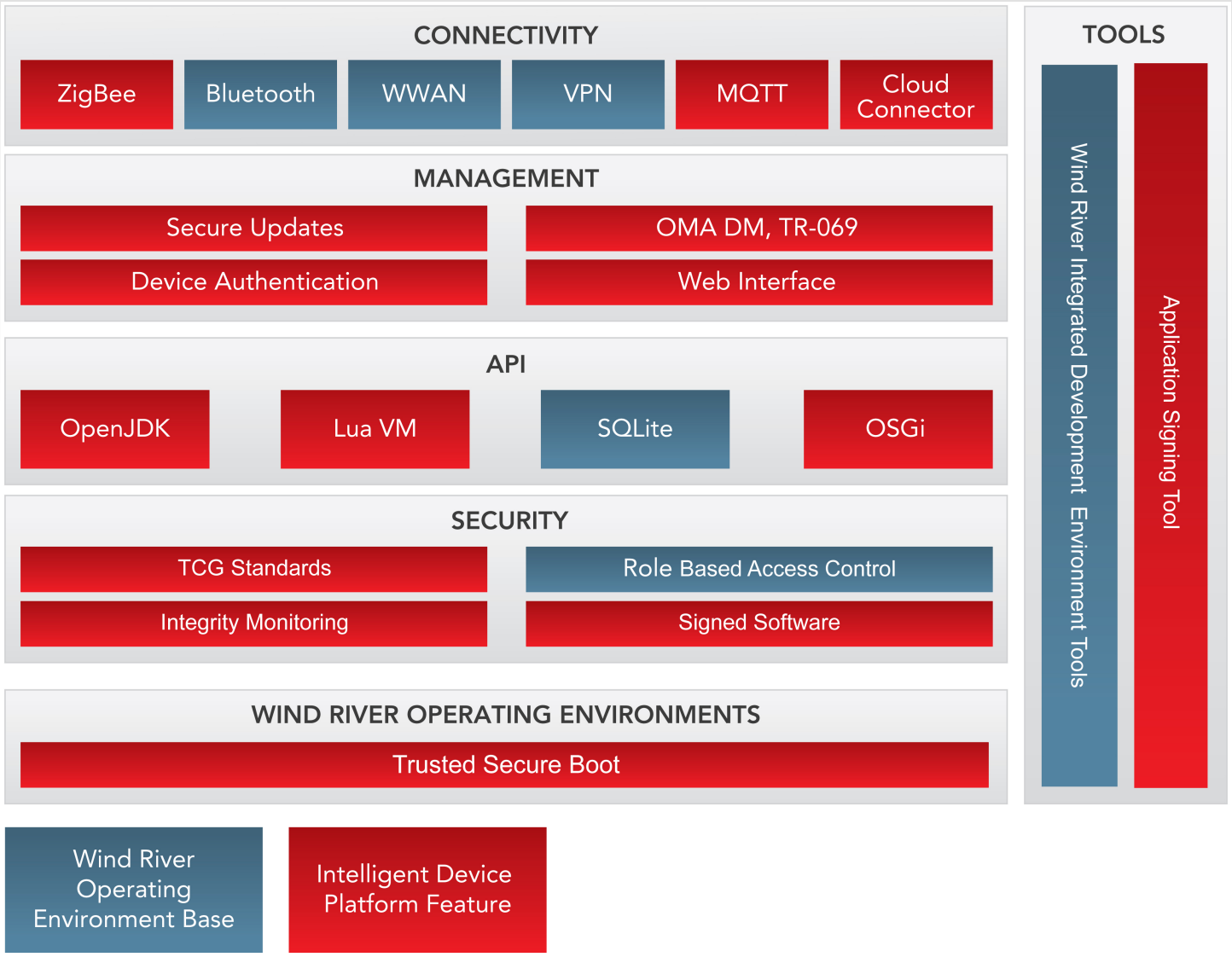
- Identify where the IDP components are located within your system
- Name hardware connectivity options supported by IDP
- Identify software connectivity options provided by IDP
- Identify device management options provided by IDP

Agenda

Architecture and Connectivity

- **Architecture**
- SRM Components
- Hardware Connectivity Options
- Software Connectivity Options
- Device Management

Wind River Intelligent Device Platform



WIND RIVER

Architecture

- The layers, profiles and templates are installed into:
  - your Wind River Linux installation
  - under **wrlinux-addons**
  - as the **wr-idp** directory
- To get access to these, the configure command for your project must include the option

  --enable-addons=wr-idp
- Other options can follow to specify which features you want or do not want in this project.

  --with-template=feature/non_grsec

Architecture (cont'd)

- IDP leverages Wind River Linux tools and adds:
  - Security
    - McAfee Embedded Control,Verified boot (Secureboot),Tamper-proof file system (Encrypted Storage), SRM signing tool, Grsecurity
  - Connectivity
    - 3G, Wi-Fi, Ethernet, Bluetooth, Zigbee, VPN, MQTT, Multiwan
  - Management Support
    - Webif, TR-069, OMA-DM
  - Application Development
    - OpenJDK, Lua, OMA-DM, Sqlite3, OSGi, MQTT
- On top of the existing
  - Compilers and tools
  - Wind River Linux

IDP Development Environment

Software Development Environment

Wind River Workbench

- Eclipse framework (Galileo) 3.5
- Eclipse CDT project 6.0
- Wind River GNU compiler
- User space and kernel debuggers
- Linux user & kernel space configuration tools
- Run-time analysis tools:
  - System viewer
  - Memory analyzer
  - Performance profiler
  - Data monitor
  - Code coverage analyzer

Additional Tools Add-ons

- OSGi Eclipse Plug-ins

Common development environment across all supported hardware

WIND RIVER

IDP Runtime Components

<table>
<tr><th colspan="3">Base System Runtime Image</th></tr>
<tr><td>Wind River Linux
• Version 5.0
• Kernel 3.4
• i586 tool chain</td><td colspan="2">• UEFI EDK II</td></tr>
<tr><td>Runtime Environments
•Lua
•Java
•OSGi*</td><td>Connectivity
•MQTT
•WAN
•IPSec
•PPP</td><td>• L2TP
• Firewall
• Cloud Connector
• OPC-DA</td></tr>
<tr><td>Trusted Software Stack
•FIPS 140-2-ready OpenSSL libraries
•Tamper-Proof File System
•Role Based Access Control
•Integrity Measurement
•Remote Device Attestation
•Secure Boot
•Intel Security Embedded Control</td><td>Management
•Web based Interface
•OMA-DM
•TR-069
•Secure Updates</td><td></td></tr>
</table>

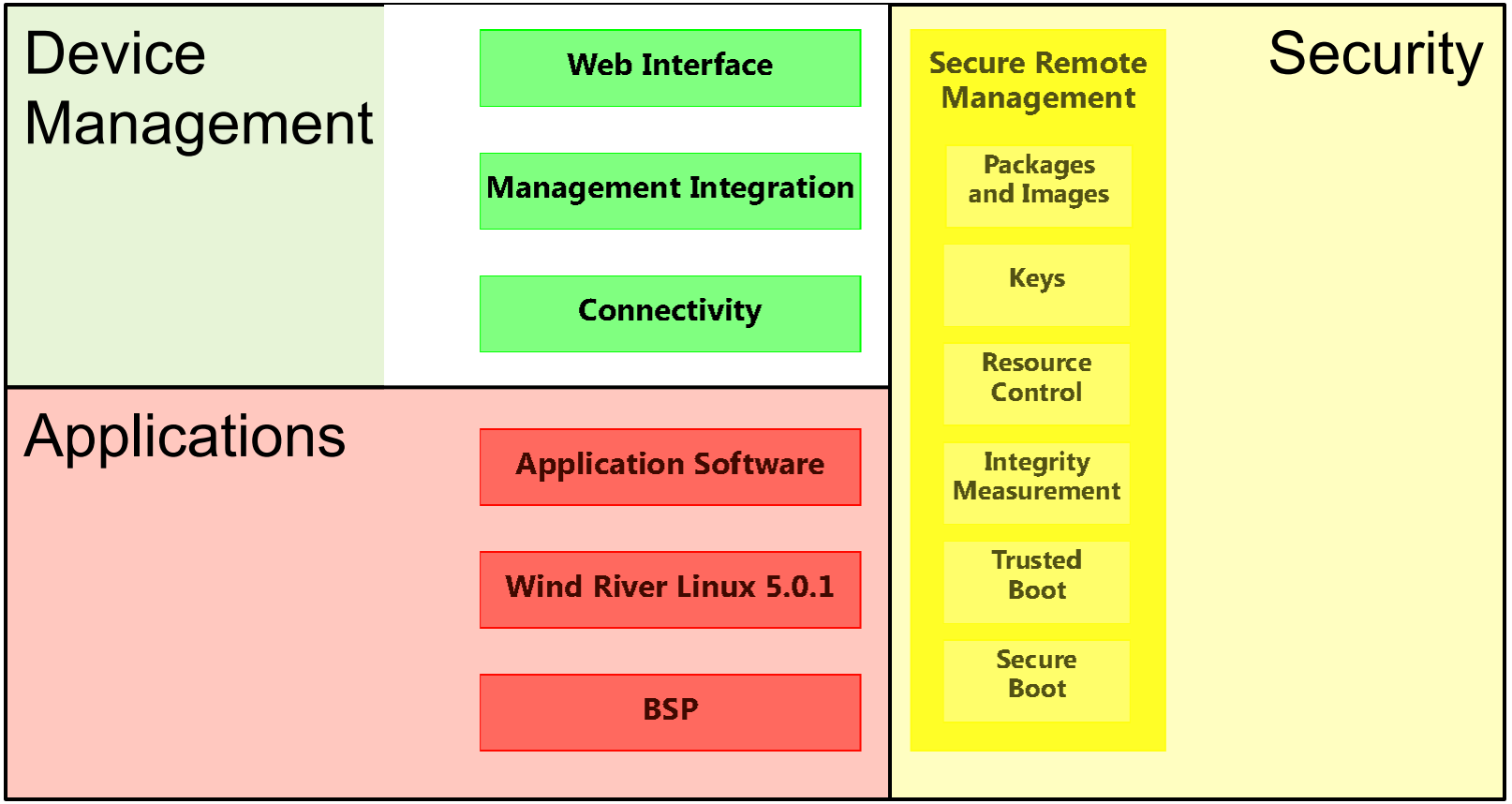
WIND RIVER

IDP 3rd Party Components

Add On	Partner	Description	Distribution
OSGi	ProSyst	mBS SmartHome SDK based on OSGi	Binary
OMA-DM	WORKS SYSTEMS	OneAgent OMA-DM	Binary
TR-069		OneAgent TR-069	Binary
802.15.4	exegin	802.154 MAC Layer + Interface Library	Binary
ZigBee		Native ZigBee stack	Binary
iDigi	Digi	Cloud Connector	Source

- All 3rd party components come fully licensed
- Runtime licensing is included as part of the customers Runtime License purchase, there are no additional deployment costs.
- Source code licensing for Add-ons may be available from the various partners.

Device View



WIND RIVER

Agenda

Architecture and Connectivity

- Architecture
- **SRM Components**
- Hardware Connectivity Options
- Software Connectivity Options
- Device Management

IDP Layers

- IDP is an addon to Wind River Linux
- It comes as layers you include in your platform project.
- The layers require the argument **--enable-addon=wr-idp** when configuring the platform project.
- The layers contain the specific features.
  - You can include these individually (without the rest of the layer) **--with-template=feature/xxx**
  - More on this later.

Layers

- Specific layers included with IDP for Wind River Linux:
  - **wr-srm**
  - **wr-idp-devkit**
  - **meta-java-dl**
  - third party related:
    - **wr-digi-idigiconnector**
    - **wr-exegin-zigbee-ia**
    - **wr-prosyst-mbs-smarthome-sdk-ia**
    - **wr-wks-oneagent-oma-dm-ia**
    - **wr-wks-oneagent-tr069**

wr-srm

- Provides secure remote management components.
- Requires:
  - **oe-core** part of Wind River Linux (must use this one!)
  - **wr-base** part of base Wind River Linux
  - **wr-features** part of base Wind River Linux
  - **wr-kernel** part of base Wind River Linux
  - **meta-networking** part of base Wind River Linux
  - **wr-idp-devkit** from **wr-idp** add-on layer
- Recommend including these layers:
  - **wrlcompat** part of base Wind River Linux
  - **wrlinux** part of base Wind River Linux

wr-srm Features

- Default template gives you:
  - SRM enabled initramfs
  - Kernel security enhancements
  - Grub-ima
    - Trusted boot
  - Grsecurity
  - Secure file system
  - Secure Package Management (Signed RPM)
  - Openssl-fips support for application development
- All of that comes when configure contains either **--with-layer=wr-srm**
  or
  --enable-roofs=glibc-idp

wr-srm Feature Names

- **grsec_std**
  - Grsecurity and related tools
- **non_grsec**
  - Virtual feature that removes the **grsec_std** feature provided by default.
- **openssl-fips**
  - Provides FIPS 140-2 ready OpenSSL libs for applications

wr-idp-devkit

- Provides the components of IDP.
- Requires these layers:
  - **oe-core** part of base Wind River Linux
  - **wr-base** part of base Wind River Linux
  - **wr-features** part of base Wind River Linux
  - **meta-networking** part of base Wind River Linux
  - **wr-srm** part of **wr-idp** add-on layer

- Recommend including these layers:
  - **wrlcompat**
  - **wrlinux**

wr-idp-devkit Features

- Default gives you:
  - Extra kernel files from
    ${LAYER_PATH_wr-idp}/wr-idp/templates/default
  - Wireless firmware specific to machine
  - Board specific features defined elsewhere…
- Feature **wr-idp-devkit-full** will get everything available in this layer except the **min_footprint** feature.

wr-idp-devkit Feature Names

- firewall
- graphics_qt
- ipsec_vpn
- l2tp
- min_footprint
- mqtt
- netifd
- online_updates
- wwan-sierra
- openjdk-bin
- pppoe
- pptp_vpn
- recovery
- upnp
- vlan
- webif
- wrs_qt_demo
- lua app. development

wr-idp-devkit Features

- Main IDP layer, contains most features and packages:
  - default
    - Default system configuration for each supported board
  - firewall
    - Provides Linux Firewall
  - graphics_qt
    - Add Wind River QT demo
  - idp_devkit_full
    - Convientient way to include all board-independent features at once
  - ipsec_vpn
    - Adds strongSwan Ipsec VPN implementation to the project.

wr-idp-devkit Features (cont'd)

- l2tp
  - Adds L2TP VPN implementation to the project
- min_footprint
  - Decreases the footprint of image by removing packages and kernel options.
- mqtt
  - Provides client/server tools for the MQTT protocol
- netifd
  - Provides a port of the Network Interface Daemon from OpenWRT
- online_updates
  - Provides ability to update target binary RPMs from an online repository.
- openjdk-bin
  - Provides the OpenJDK binary
- pppoe
  - Provides the point-to-point connectivity over Ethernet

Wr-idp-devkit Features (cont'd)

- pptp_vpn
  - Provides point-to-point tunneling protocol (pptp) for VPN connections
- recovery
  - Provides ability to create bootable recovery media for project.
- upnp
  - Provides Universal Plug aNd Play support to the project.
- vlan
  - Adds 802.1Q protocol and support to the project
- webif
  - Adds Webif, web browser based interface for configuring target services
- wrs_qt_demo
  - Add the Wind River QT demo of QT development capability.
- wwan-sierra
  - Adds Sierra modem management apps for MC8355 & MC7750.

Agenda

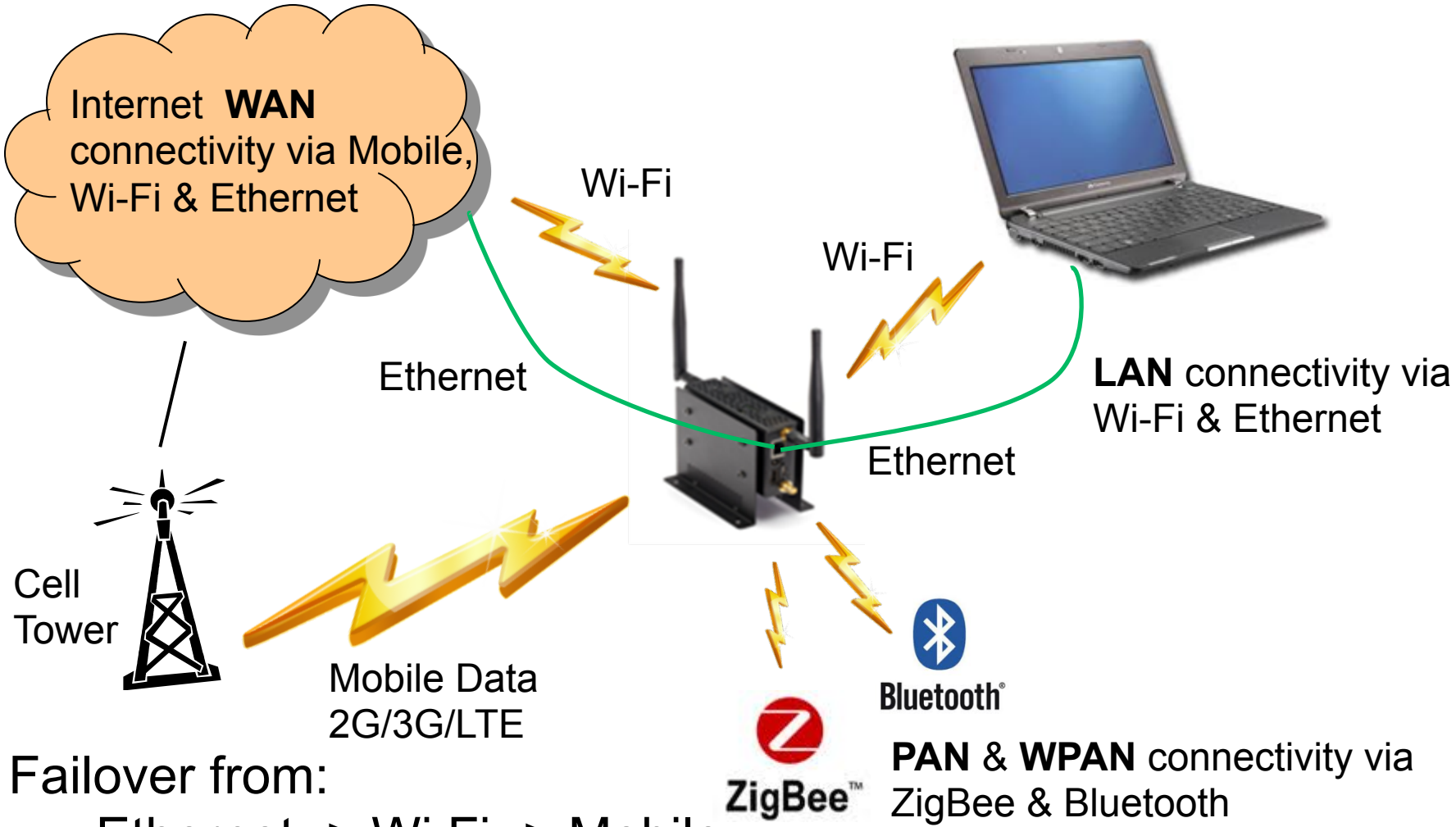
Architecture and Connectivity

- Architecture
- SRM Components
- **Hardware Connectivity Options**
- Software Connectivity Options
- Device Management

Hardware Connectivity Options

- IDP provides connectivity options beyond those included in Wind River Linux 5.0.1.
  - Wi-Fi
    - Iwlwifi (Intel IPW2100, IPW2200, 3945ABG, 4065AGN)
    - Rt2x00 (Ralink USB devices RT2770, RT2870, RT3070 RT3071, RT3072, RT3572)
    - Rt73usb (Ralink USB devices RT2571W, RT2573 & RT2671)
  - Bluetooth
    - BlueZ Bluetooth stack
      - documented online at *http://www.bluez.org*
      - adds the software capability and functionality to support Bluetooth
      - requires an external Bluetooth adapter

IDP 2.0 – Communications Capabilities



Failover from:

Ethernet -> Wi Fi -> Mobile

IDP WAN Communications Support

Specific to Cross Hill Industry / Energy Reference Design

Ethernet
- Dual 10/100

Wi-Fi – Supported via mini PCIe
- Intel Centrino Advanced-N 6205
  - 802.11 a / b / g / n
  - Client Mode

Mobile - Supported via mini PCIe
- Telit HE910 PCI

Automatic Failover / Failback capability available between WAN interfaces (ie. Ethernet -> Wi-Fi -> Mobile)

LAN / WPAN Communications Support

Specific to Cross Hill Industry / Energy Reference Design

Ethernet
- Dual 10/100

Wi-Fi
- 802.11 a / b / g / n
- Access Point or Ad-Hoc Mode

WPAN
- Bluetooth with LE
- Dual 802.15.4
- ZigBee

Serial
- RS-485 up to 2.7 Mbps
- RS-232

IEEE 802.15.4

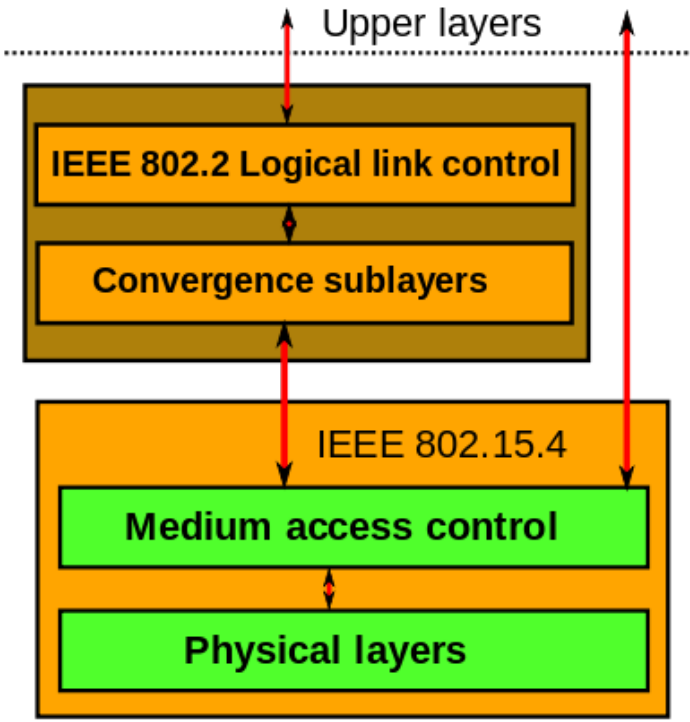
Basis of 6LowPAN (IPv6 Low power Wireless Personal Area Network), ZigBee, and others

Offers the lower network layers of a type of wireless personal area network (WPAN)

- Focused on low-cost, low-speed ubiquitous communication between devices (in contrast with other, more end-user oriented approaches, such as Wi-Fi).

- Targets a 10-meter communications range @ 250 kbit/s.

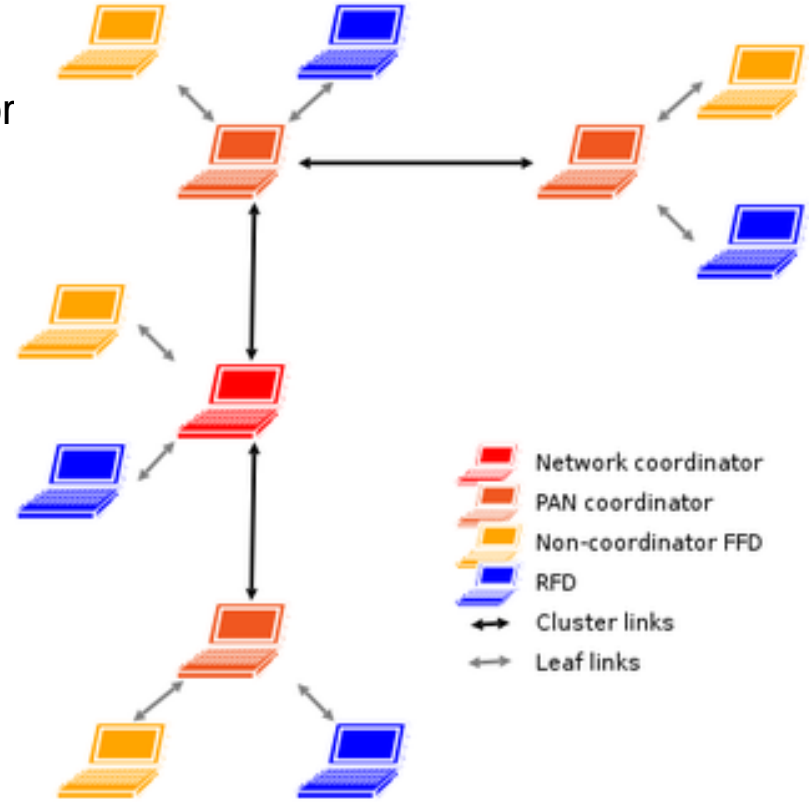
- Important features include
  - real-time suitability by reservation of guaranteed time slots,
  - collision avoidance through CSMA/CA
  - Integrated support for secure communications (AES128) handled at MAC layer, key management provided by upper layers.
  - Can also include power management functions such as link quality and energy detection.
  - 127 byte frames
  - Capable of running in unlicensed frequencies, including the 2.4 -GHz band in the U.S.
  - Mesh networking built in

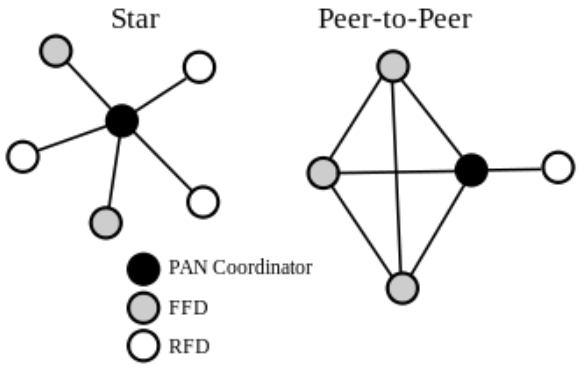


802.15.4 – Node Types

Full-function device (FFD). Can serve as the coordinator of a PAN or as a common node. Implements a general model of communication which allows it to talk to any other device: it may also relay messages, in which case it is dubbed a coordinator (or PAN coordinator when it is in charge of the whole network).

Reduced-function devices (RFD). Meant for extremely simple devices with very modest resource and communication requirements; due to this, they can only communicate with FFDs and can never act as coordinators.





What is ZigBee?

A specification for a suite of high level communication protocols used to create personal area networks built from small, low-power digital radios

- Used in applications that require a low data rate, long battery life, and secure networking

- Based on IEEE 802.15.4 protocol

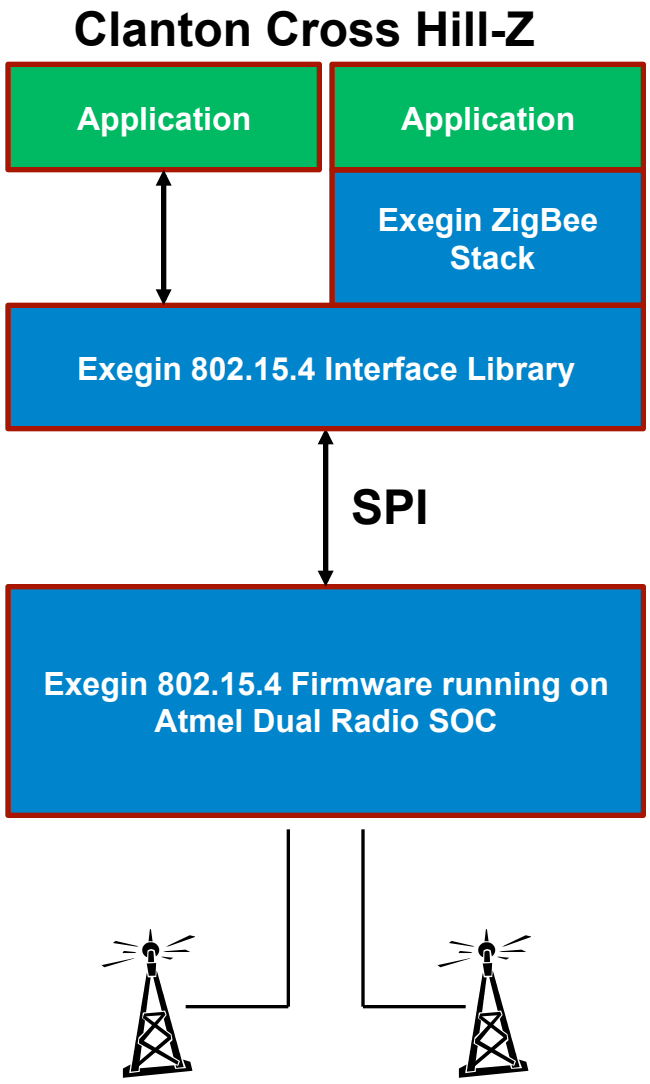
- intended to be simpler and less expensive than other WPANs, such as Bluetooth or Wi-Fi.

Exegin ZigBee Software Stack

- Provided by Exegin (based in Vancouver, BC)
- 32-bit ZigBee stack for ZigBee 2006, ZigBee PRO, and proprietary stack profiles
  - Fully reconfigurable at run-time
  - Coordinator, router, or end device
  - Support of both standard and high-security modes
  - Targeted at embedded devices, uses <256kB Flash, <24kB RAM
- certified in January 2010 by NTS
- Deployed with several million smart meters
- ZigBee Smart Energy Profile 1.0
- Allows multiple instances to share code on one processor

IDP ZigBee Implementation

- Uses Atmel ATmega WPAN SOC on Cross Hill SPI Module
- Provides 2 completely independent WPAN networks
- ZigBee Stack Profiles
  - ZigBee 2007
  - ZigBee PRO
- Cluster Libraries
  - ZigBee Cluster Library 2008
  - ZigBee Smart Energy 1.0
- Programming Language
  - C
- Customers may interface at either ZigBee or 802.15.4 layers



Agenda

Architecture and Connectivity

- Architecture
- SRM Components
- Hardware Connectivity Options
- **Software Connectivity Options**
- Device Management

Software Connectivity Options

- Connectivity provided with IDP that is beyond Wind River Linux 5.0.1.
  - VPN
    - **ipsec_vpn**
      - strongSwan implementation --with-template=feature/ipsec_vpn
    - **pptp_vpn**
      - pptpvpn.org implementation –with-template=feature/l2tp
    - **l2tp_vpn**
      - openl2tp.org implementation --with-template=feature/l2tp_vpn
  - MQTT
    - Message Queue Telemetry Transport
    - mqtt.org –with-template=feature/mqtt
    - Mosquitto server provided, lua client

Agenda

Architecture and Connectivity

- Architecture
- SRM Components
- Hardware Connectivity Options
- Software Connectivity Options
- **Device Management**

Device Management

- OneAgent TR-069
  - Auto-configuration and dynamic service provisioning
  - Software/firmware image management
  - Status and performance monitoring
  - Diagnostics
  - **--with-layer=wr-wks-oneagent-tr069**
- Webif
  - Web browser based management of network interfaces
    - Ethernet, Wi-Fi, 3G
  - Review/alter configuration of many other target system features.

Device Management (cont'd)

- OMA-DM
  - DevInfo provides device information that identifies the device.
  - DMAcc provides the authentication.
  - ConnMO provides management for connectivity settings.
  - SCOMO manages package installation and activation.

Platform Config example(1)

- $WIND_LINUX_CONFIGURE --enable-board=intel-quark --enable-addons=wr-idp --enable-kernel=standard –enable-rootfs=glibc-idp --enable-parallel-pkgbuilds=4 --enable-jobs=4 --enable-ccache=yes --with-ccache-dir=<build_ccache_Dir> --with-sstate-dir=<build_sstate_Dir> --with-template=feature/intel-wilkinpeak2 --with-layer=wr-digi-idigiconnector,wr-exegin-zigbee-ia, wr-prosyst-mbs-smarthome-sdk-ia,wr-wks-oneagent-tr069, wr-wks-oneagent-oma-dm-ia,wr-intel-support --enable-reconfig

Platform Config example(2)

- $WIND_LINUX_CONFIGURE --enable-board=intel-quark --enable-addons=wr-idp --enable-kernel=standard –enable-rootfs=glibc-idp --enable-parallel-pkgbuilds=4 --enable-jobs=4 --enable-ccache=yes --with-ccache-dir=<build_ccache_Dir> --with-sstate-dir=<build_sstate_Dir> --with-template=feature/intel-wilkinpeak2 --with-layer=wr-digi-idigiconnector,wr-exegin-zigbee-ia, wr-prosyst-mbs-smarthome-sdk-ia,wr-wks-oneagent-tr069, wr-wks-oneagent-oma-dm-ia,wr-intel-support --without-layer=wr-srm --enable-reconfig

Platform Config example(3)

- $WIND_LINUX_CONFIGURE --enable-board=intel-quark --enable-addons=wr-idp --enable-kernel=standard –enable-rootfs=glibc-idp --enable-parallel-pkgbuilds=4 --enable-jobs=4 --enable-ccache=yes --with-ccache-dir=<build_ccache_Dir> --with-sstate-dir=<build_sstate_Dir> --with-template=feature/wr-idp-devkit-full,feature/intel-wilkinpeak2 --with-layer=wr-digi-idigiconnector,wr-exegin-zigbee-ia, wr-prosyst-mbs-smarthome-sdk-ia,wr-wks-oneagent-tr069, wr-wks-oneagent-oma-dm-ia,wr-intel-support --enable-reconfig

Questions

1. How do you enable the IDP software for use in your project?
2. What are some hardware connectivity options for IDP?
3. Name some software connectivity options for IDP.
4. What are the device management options available in IDP?

Review

In this chapter you learned:

- The location of the IDP components within your system
- Hardware connectivity options supported by IDP
- Software connectivity options provided by IDP
- Device management options provided by IDP

WIND RIVER