Getting started with Proprietary Wireless

WHAT IS RAIL AND WHAT IS CONNECT

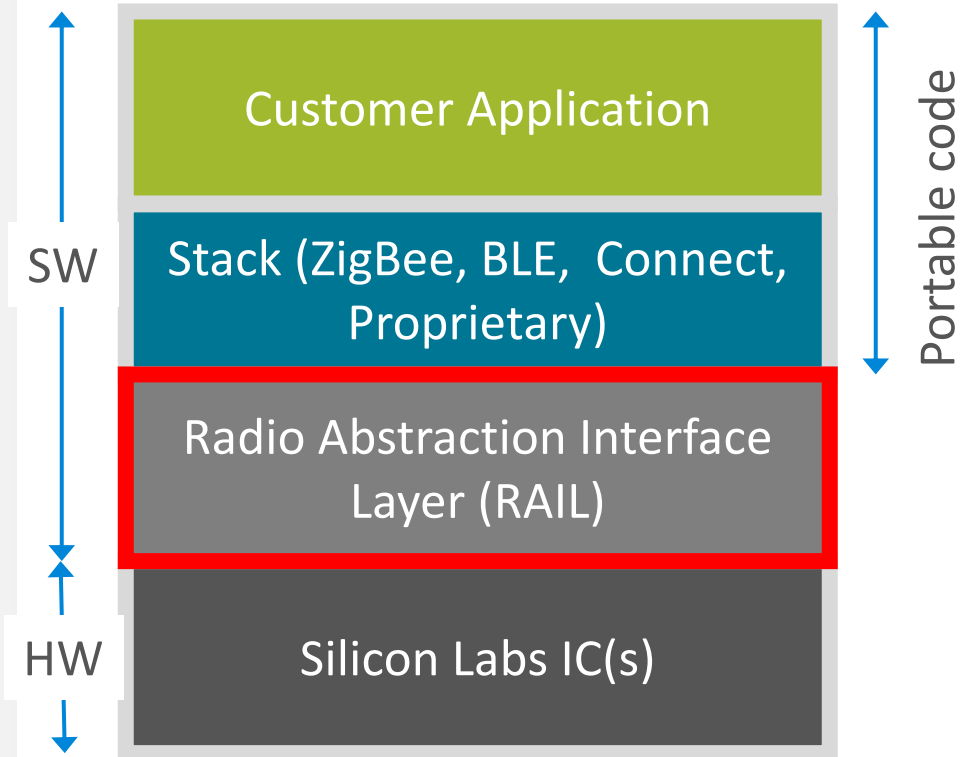
When to use Proprietary?

+ Compatibility required with existing proprietary protocol
+ Highly optimized solution needed
  + For energy consumption
  + For wireless range
+ Full control of the protocol is necessary

In exchange:

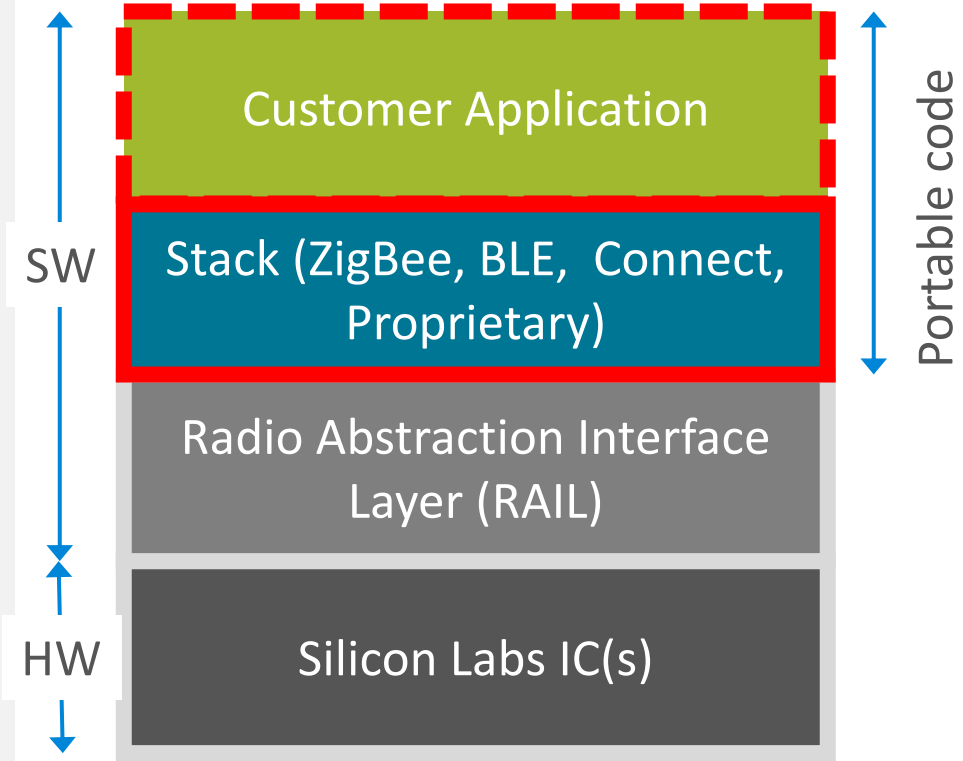
– More difficult development, longer time to the market
– Incompatibility with existing infrastructures
– Security holes can remain hidden for a long time due to the difficulty of the analysis
  - But once they discovered, it’s usually easy to exploit them

What is RAIL?



- Radio Abstraction Interface Layer
- Library, used to access radio transceiver hardware
- Has some MAC features that can be accelerated by HW
  - CRC, whitening
  - Auto ACK
  - Address filtering
  - CSMA/CA or LBT
  - Scheduling and timestamping
- RAIL should provide a common API across all supported chips
- All Silicon Labs stacks are implemented on top of RAIL

What is Connect?



- Stack, up to the Network layer
  - Configurable PHY (pre-set PHYs available for all ISM regions)
  - 15.4 based MAC
  - Extended star based network topology
  - Implemented on top of RAIL
- Special „direct MAC mode”
  - Pure IEEE 802.15.4 MAC implementation
- Also includes some application layer features
  - Task and sleep scheduler
  - Over-The-Air (OTA) bootloader image distribution

What is Gecko SDK suite? What is Flex?

- Gecko SDK suite is what we usually call “common SDK”
  - All stacks should use common drivers (usually MCU) from a common SDK
  - You can install multiple SDKs under Gecko SDK suite, eg:
    - EmberZNet for Zigbee
    - Bluetooth Smart SDK for BLE
    - Flex SDK for Connect / RAIL
  - It includes common tools for all stacks
    - Gecko Bootloader
    - Non-volatile memory library
- Flex SDK for proprietary development
  - It includes the Connect stack
  - The RAIL library
  - RAIL and Connect plugins and examples
  - Basically it’s RAIL and Connect sharing the same GUI
- Connect and RAIL represent two different development workflows in Flex SDK

Connect or RAIL?

+ Full featured stack, including network layer
+ Task and sleep scheduler
+ OTA bootloader
+ MAC provides 15.4 security
+ Supports MicroC OS

– Fixed proprietary frame format, can't connect to other networks
– Not very flexible, e.g. difficult to set up deeper than EM2 sleep state
– Not compatible with DMP (currently)
– Only MicroC OS is supported
– RTC oscillator is required for scheduler

+ Basic PHY and MAC layer functions
+ Very flexible, a lot of services
  - Supports legacy proprietary systems
+ Accelerated MAC is usually enough for single hop networks
+ Supports Dynamic Multi Protocol (DMP)
+ Can be used with any RTOS
  - But requires careful IRQ setup

– No network layer, so no multi-hop support
– No application features like OTA
– Security must be done in application

Connect or RAIL?

+ Full featured stack, including network layer
+ Task and sleep scheduler
+ OTA bootloader
+ MAC provides 15.4 security
+ Supports MicroC OS

– Fixed proprietary frame format, can't connect to other networks
– Not very flexible, e.g. difficult to set up deeper than EM2 sleep state
– Not compatible with DMP (currently)
– Only MicroC OS is supported
– RTC oscillator is required for scheduler

+ Basic PHY and MAC layer functions
+ Very flexible, a lot of services
  - Supports legacy proprietary systems
+ Accelerated MAC is usually enough for single hop networks
+ Supports Dynamic Multi Protocol (DMP)
+ Can be used with any RTOS
  - But requires careful IRQ setup

– No network layer, so no multi-hop support
– No application features like OTA
– Security must be done in application

Connect direct MAC or RAIL 802.15.4?

+ Includes MAC security
+ Includes association
+ Simpler MAC header creation

– Only a single security mode is supported
– Only fully 802.15.4 compatible link layer is allowed
– No DMP, limited multiPhy
– emberHAL

+ Deviations from the standards are allowed
+ emlib/emdrv
+ Full DMP/multiPhy capabilities

– MAC header must be created manually
– No standard MAC association
– Security must be done in application

Workflow: Connect

- “Create a free network layer which has the most used Zigbee features!”
- Development flow is similar to the zigbee stack
- Base (a.k.a. emberHAL) as HAL
- AppBuilder
  - Heavy use of plugins
- Application is written as callback implementations

RAIL

- “Create an easy to use radio driver, which hides the complex hardware, and the differences between generations!”
- Development flow is similar to EFM32
- Uses the radio on the register level
- Very small HAL dependency
  - em_cmu and em_core
- RadioConfigurator
  - AppBuilder is mostly used as a container
- Examples are written as an infinite loop in main.c

Silicon Labs Confidential

RAIL framework components

- RAIL framework
  - Part of the Flex SDK
  - Installed in Simplicity Studio with Gecko SDK Suite
  - RAIL 2.x
    - Major API change with DMP in mind.
    - It fixed many API level problems in RAIL
    - It's expected to be a very stable API
- Main software components:
  - RAIL API Library

    Implements core features and runtime API-s to configure & control the radio
  - Radio Configurator

    Calculates all modem parameters – complete radio configuration
  - Sample Applications

    Provides code examples for application development
  - Documentation

    Includes QSG138 Quick Start Guide, API reference and application notes

RAIL Features 1/4

- General
  - Initialize the RAIL API layer.
  - Collect entropy from the radio (if available).
- Radio Configuration
  - Configure the radio frequency, packet format, channel configuration and other PHY parameters.
  - Query current PHY data rates and parameters like current channel.
- State Transitions
  - Configure automatic radio state transitions and transition timings.
- Auto ACK
  - Configure the radio for automatic acknowledgments.
  - Load the auto ack payload.
- System Timing
  - Get the current time in the RAIL timebase.
  - Configure a timer to trigger an event callback after an absolute or relative delay.
  - Specify where within a packet its timestamp is desired.

RAIL Features 2/4

- Events
  - Configure which radio or RAIL events the application wants to be told about via callback.
- Data Management
  - Allows the application to choose the type of data and the method of data interaction through RAIL.
- Receive
  - Configure receive options like CRC checking.
  - Start or schedule when to receive.
  - Enable and configure Address Filtering for each packet.
- Transmit
  - Configure the power amplifier (PA) and set transmit power output.
  - Load and send packet data, either immediately, scheduled, or using CSMA or LBT.
  - Control per-transmit options like CRC generation, ACK waiting, etc.
- Multiprotocol
  - Manage time-sharing of the radio among different protocols.

RAIL Features 3/4

- Calibration
  - APIs for handling various radio calibrations for optimal performance.
- RF Sense
  - Enable RF energy sensing of specified duration across the 2.4 GHz and/or Sub-GHz bands (EFR32 only).
- Packet Trace (PTI)
  - Configure Packet Trace pins and serial protocol.
  - Specify the stack protocol to aid network analyzer packet decoding.
- Diagnostic
  - Output debug signals like an unmodulated tone and a continuously modulated stream of data.
  - Configure crystal tuning for your radio.
  - Fine-tune the radio tuner frequency.

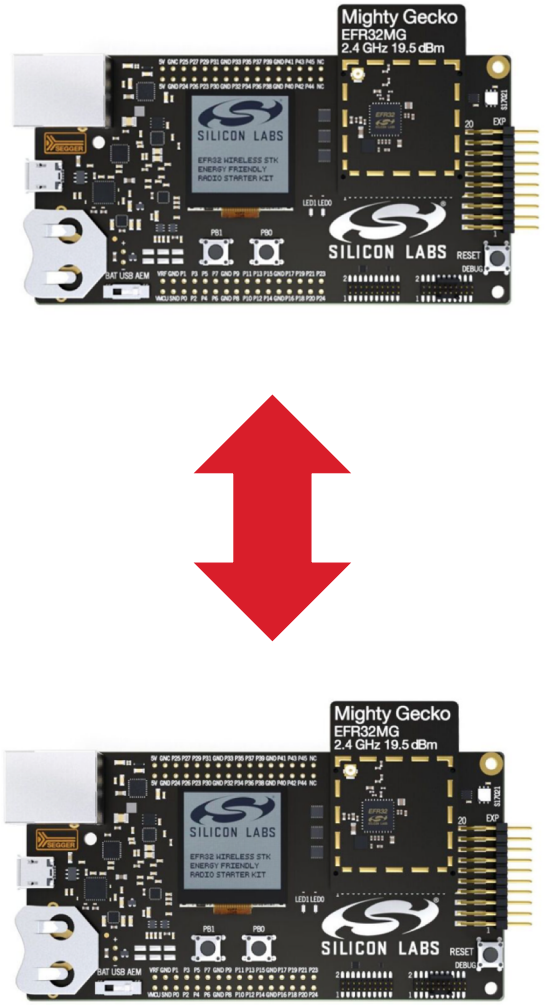
RAIL Features 4/4

- Protocol-specific hardware acceleration:
  - IEEE 802.15.4
    - Configure the IEEE802.15.4 2.4GHz PHY.
    - Configure node address and address filtering for IEEE 802.15.4.
    - Configure auto ack for IEEE 802.15.4.
  - BLE
    - Configure the Bluetooth Low Energy 1Mbit PHY.
    - Preamble, sync word and whitening adjustment function for connections.

Silicon Labs Confidential

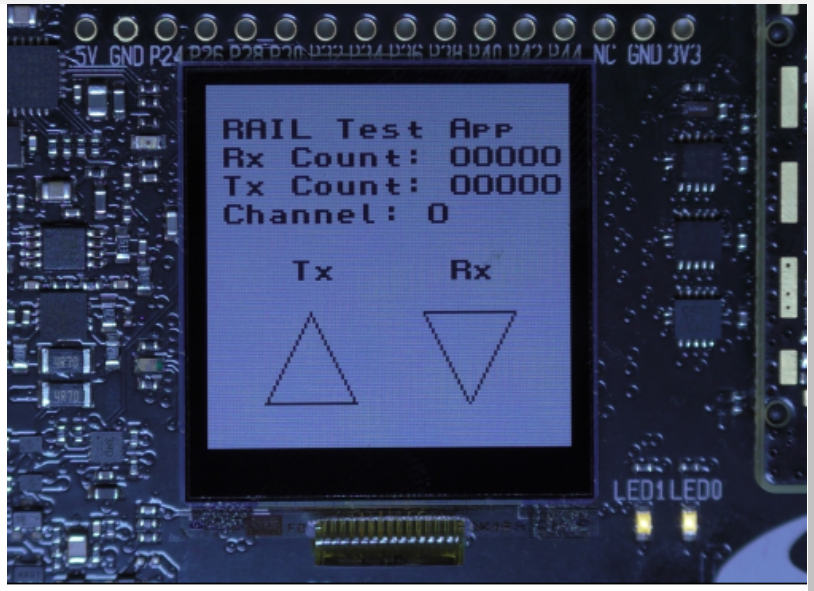
RAIL Software Examples

- RAIL: Simple RAIL with HAL
- RAIL: Simple RAIL without HAL
- RAIL: Duty Cycle
- RAIL: Energy Mode
- RAIL: Range Test
- RAIL: RAILTEST
- RAIL: Simple TRX
- RAIL: Simple TRX with ACK (Software)
- RAIL: Simple TRX with FIFO (Long Packet)
- RAIL: WMBus Meter and WMBus Collector



RAIL Software Examples – RAILTEST

- Lab measurement software (CW, PN9, BER, PER, Direct mode, Ctune)
- Demonstrates the features of RAIL through a command line interface
- For the full list of commands, type “help” on the CLI, or check the user guide
- Consider Current Consumption
  - Uses LCD, pushbuttons and LEDs on the WSTK -> turn off peripherals by setPeripheralEnable ()
- Dual Sync support added to configTxOptions and setRxOptions
- Added the following commands: rxConfig, setRxOptions, setFixedLength, fifoStatus, dataConfig, setTxFifoThreshold, setRxFifoThreshold
- See AN972 for more details



RAIL: Getting Started

- Radio Configurator: AN971
- API documentation: Installed under <studio_folder>\developer\sdks\gecko_sdk_suite\v2.3\protocol\flex\documentation\API_RAIL_HTML\index.html
- Software tutorial: RAIL tutorials table of contents

Silicon Labs Confidential

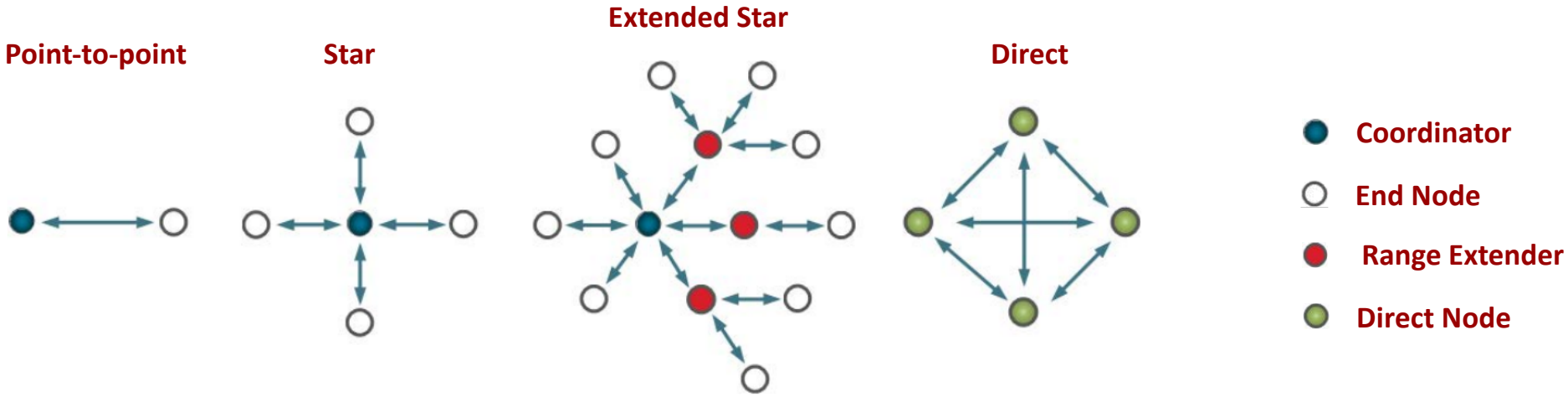
Silicon Labs Connect Networking Stack

- ***Connect is a production-quality wireless networking stack and development environment for broad based proprietary applications***
- Full-featured, easily customizable wireless connectivity solution for the Sub-GHz and 2.4GHz proprietary market
- Based on 802.15.4 like MAC
  - Low level details of network formation and radio configuration
  - Customer can focus on application development
- Provides software portability across platforms and reduces our customers' time-to-market
- Optimized for devices that require low power consumption
- Addresses broad range of applications and supports proprietary wireless protocols across worldwide geographic regions
- (EZR32 support was dropped with Flex SDK 2.0)

Key Features

- Network and Application
  - Reliable point-to-point, star, extended star and direct mode network topologies
  - Network formation: Improved and more secure 15.4-like association mechanism; Centralized address allocation at the coordinator
  - Full routing support: Any node in the network can communicate with any other node in the network; Routing is totally transparent to the application
  - NCP (Network Co-Processor) mode operation
  - OTA bootloader support
  - MicroC OS support
- PHY and MAC
  - based on IEEE 802.15.4 standard
  - Support for 2.4 GHz and sub-GHz PHY
  - Fully tested pre-set PHY
  - Almost complete PHY configuration
  - Encryption and authentication of data packets
    - Key distribution is still the task of the application
  - Nodes are provided with short and long IDs; Network is identified by its PAN ID (Personal Area Network)
  - Channel access is regulated via CSMA/CA
  - Frequency hopping provided for regulatory compliance

Network Topology



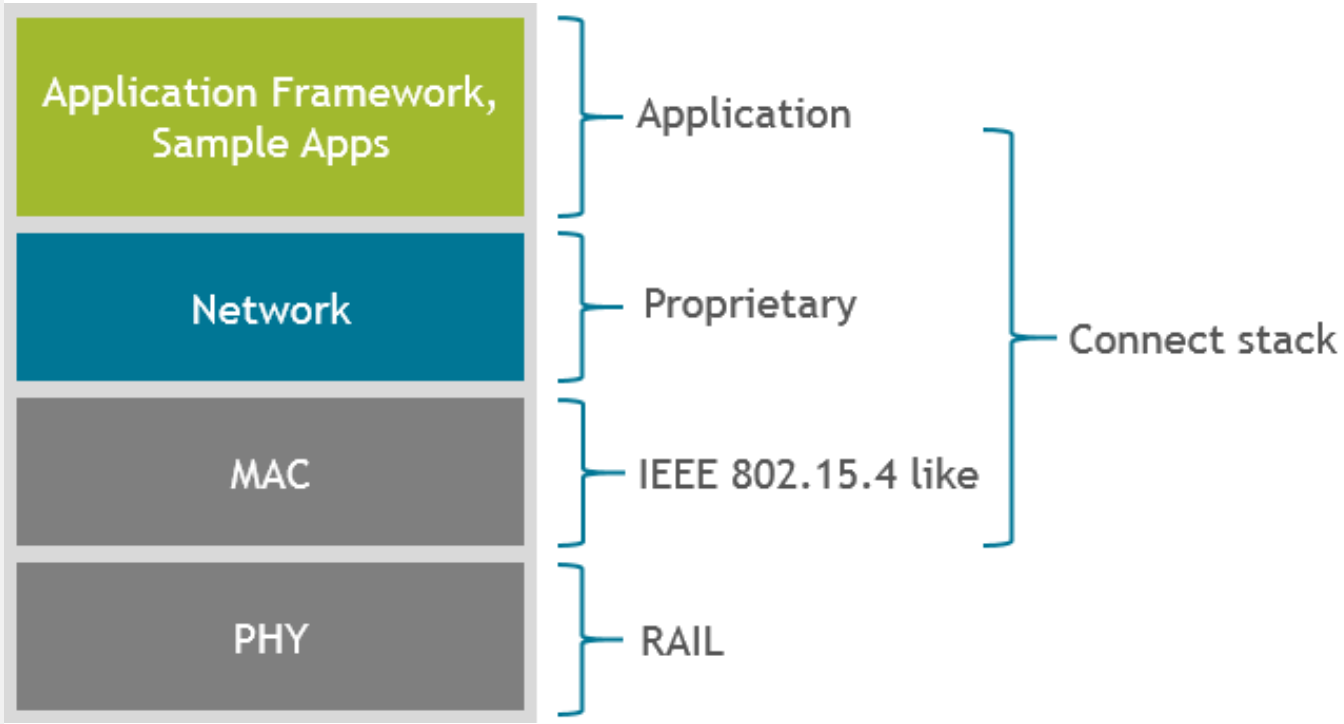
- Point-to-point
  - Simplest communication between two devices
- Star
  - Single Coordinator hub communication with multiple Star End Nodes
- Extended Star
  - Includes a Star Range Extender between Star Coordinators and Star End Nodes
  - Communication between Coordinator and Far Star End nodes flows through Star Range Extenders
- Direct mode
  - Two or more Direct nodes everyone communicates with everyone in a single hop fashion

Direct MAC mode

- Special mode, implementing a 100% IEEE 802.15.4 compatible MAC layer
- The phy config (e.g. carrier frequency) can deviate from the standard
- It’s not a full implementation
- Supports 802.15.4 beacons and association
- Supports 802.15.4 level-5 MAC security (authentication and encryption)

Silicon Labs Confidential

Connect Stack Architecture



- RAIL (Radio Abstraction Interface Layer) based radio configuration for EFR32
- MAC is based on IEEE 802.15.4-2006 standard
- Network layer is based on a proprietary protocol
- All network tasks are encapsulated in stack (Libraries)
- Full routing support that is transparent to the Application Layer
- Application Framework can be configured by the user through a GUI
- Application code becomes completely portable – recompile for different regions, different MCUs and different radios

Connect: Getting started

- Radio Configurator: AN971
- API documentation: Installed under <studio_folder>\developer\sdks\gecko_sdk_suite\v2.3\protocol\flex\documentation\API_CONNECT_HTML\index.html
- Connect user guider: UG235

Silicon Labs Confidential

Thank You!

WWW.SILABS.COM