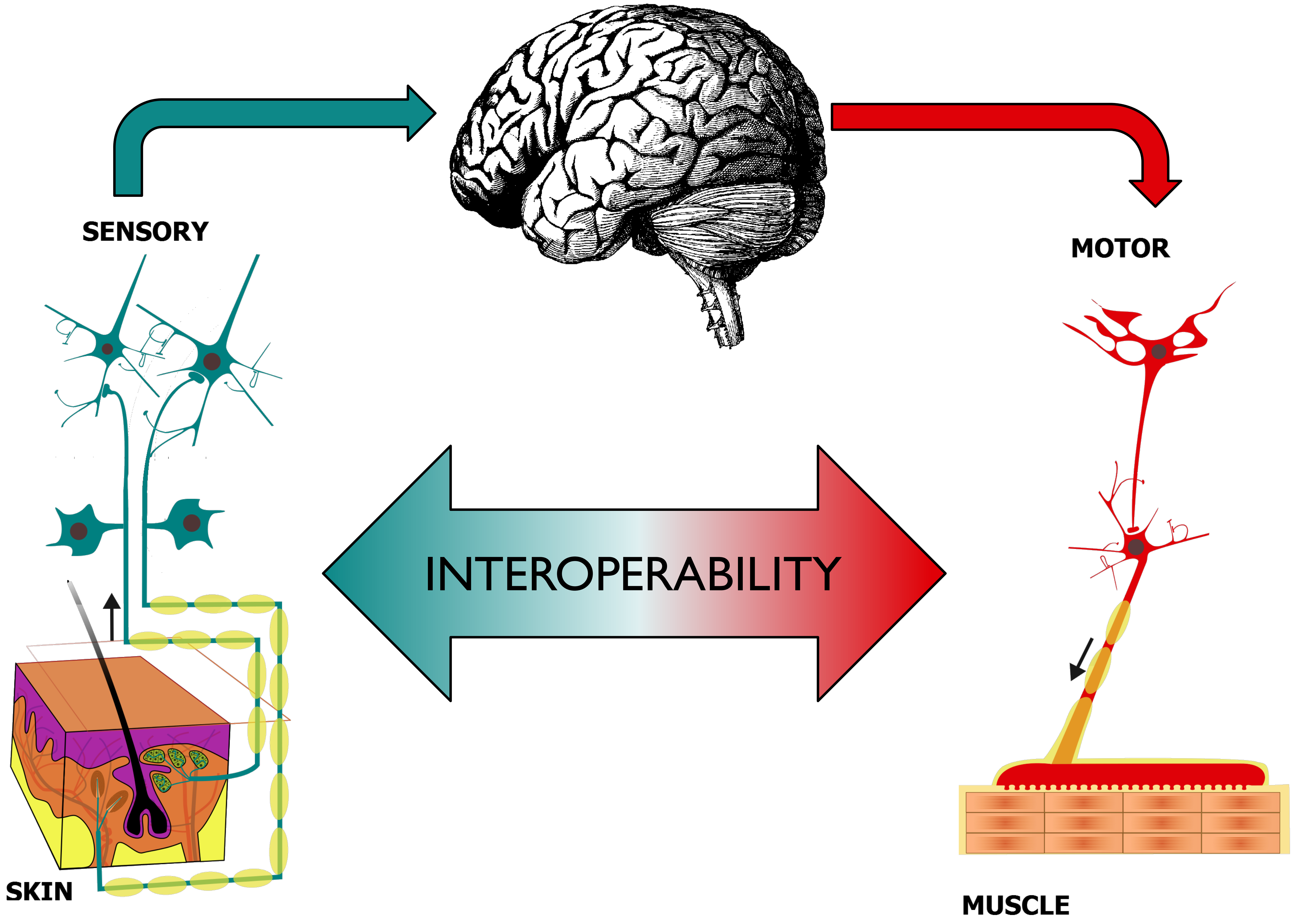
Interoperability Frameworks for RIOT-OS

@Mattia_Antonini

m.antonini@create-net.org


SENSORY
MOTOR
INTEROPERABILITY
SKIN
MUSCLE

IoTivity

AllJoyn®

THREAD

open source

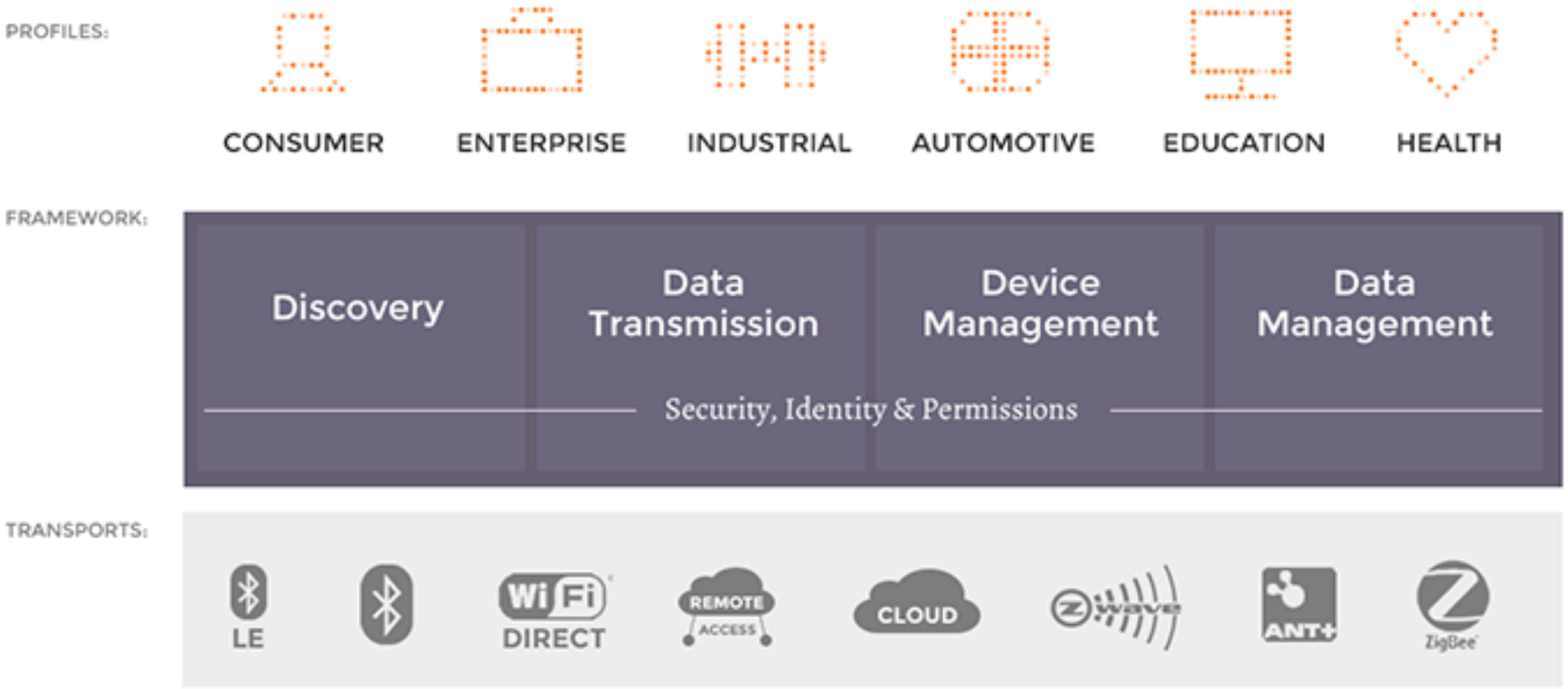
IoTivity

FRAMEWORK APIs

Common Object Model

OPEN INTERCONNECT CONSORTIUM

THE LINUX FOUNDATION



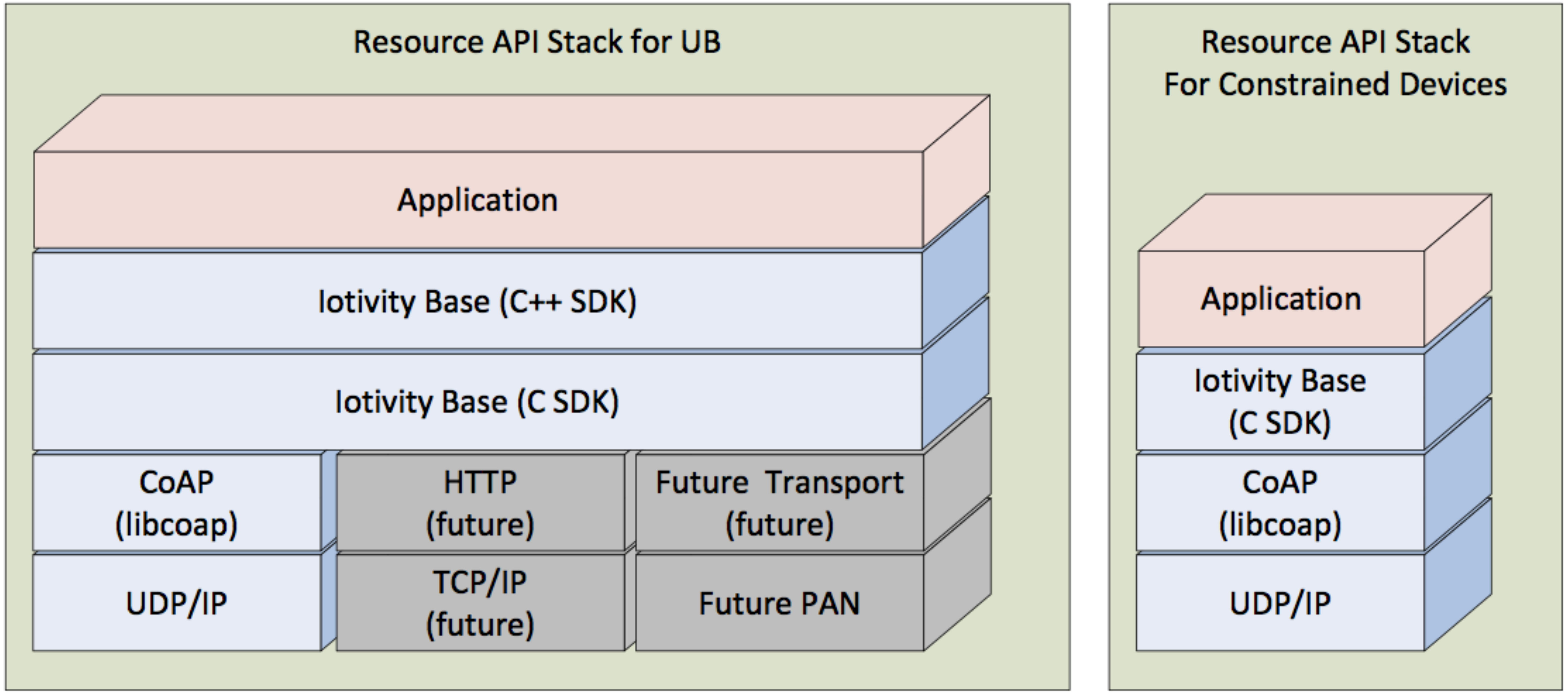
PROFILES: CONSUMER, ENTERPRISE, INDUSTRIAL, AUTOMOTIVE, EDUCATION, HEALTH

FRAMEWORK: Discovery, Data Transmission, Device Management, Data Management

Security, Identity & Permissions

TRANSPORTS: LE, WiFi DIRECT, REMOTE ACCESS, CLOUD, Z-wave, ANT+, ZigBee

IoTivity Stack



Resource API Stack for UB

Application

Iotivity Base (C++ SDK)

Iotivity Base (C SDK)

CoAP (libcoap)

HTTP (future)

Future Transport (future)

UDP/IP

TCP/IP (future)

Future PAN

Resource API Stack For Constrained Devices

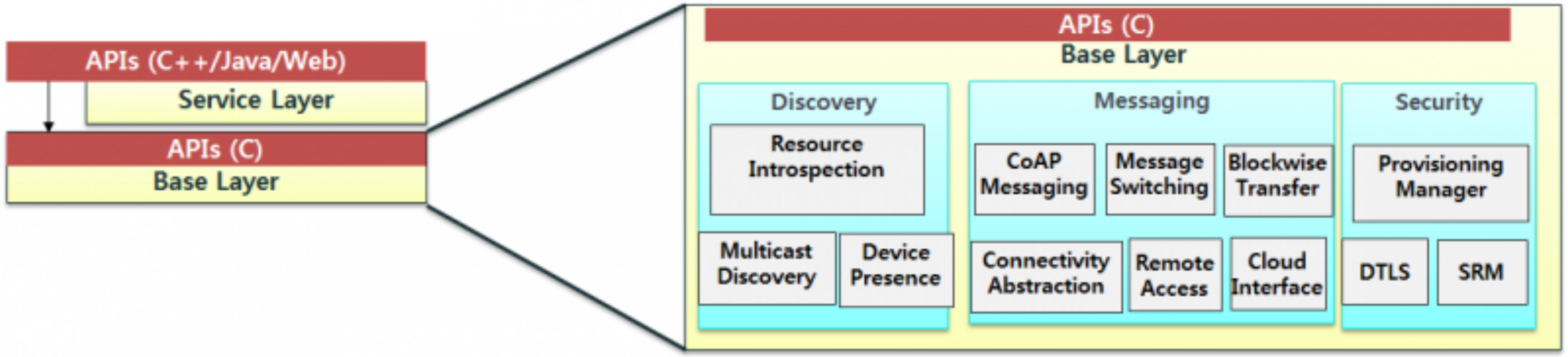
Application

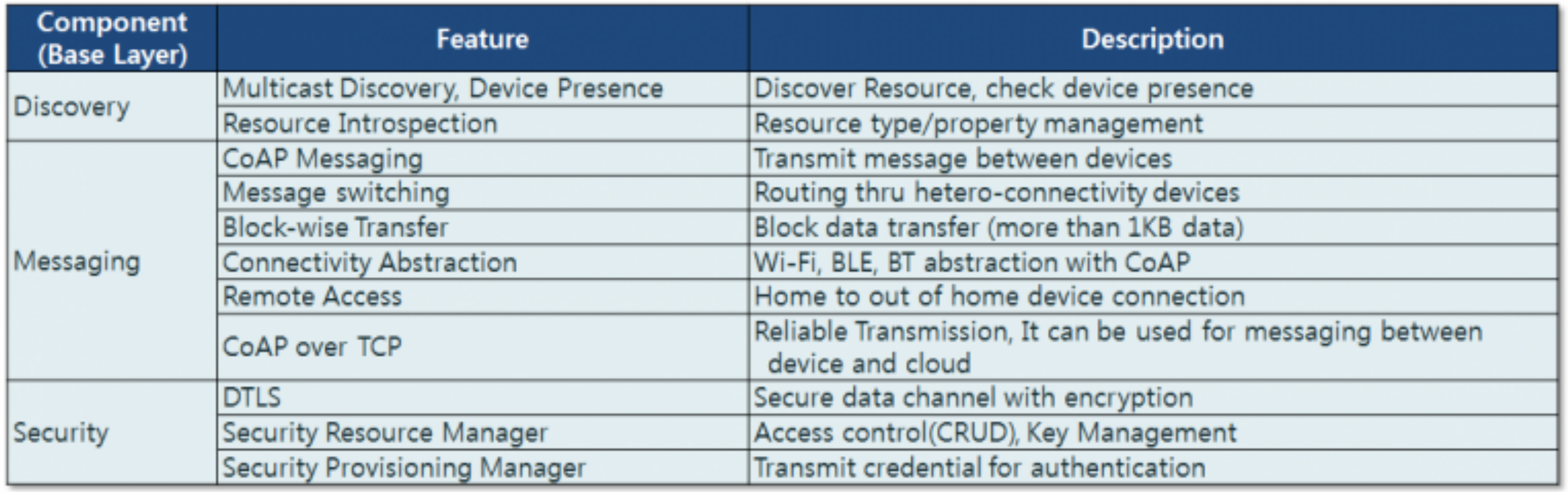
Iotivity Base (C SDK)

CoAP (libcoap)

UDP/IP

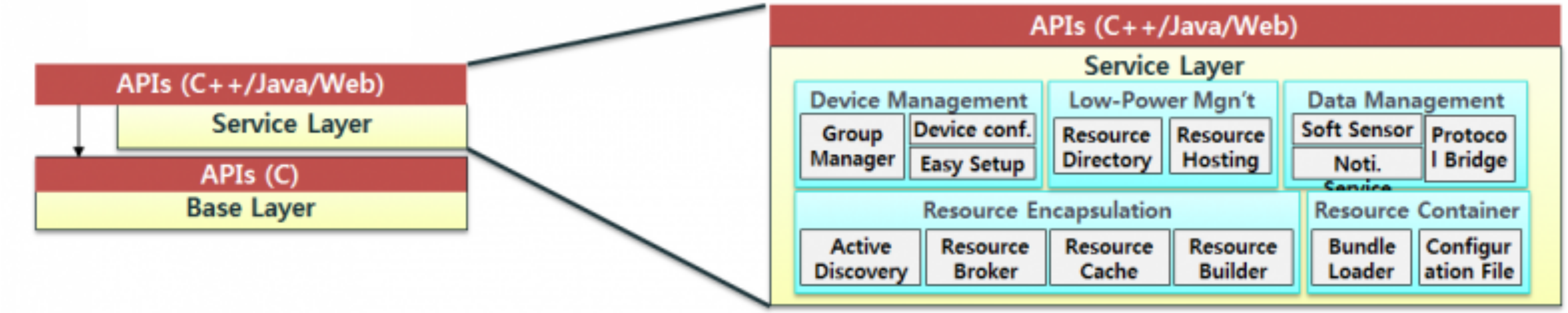
IoTivity Base Layer

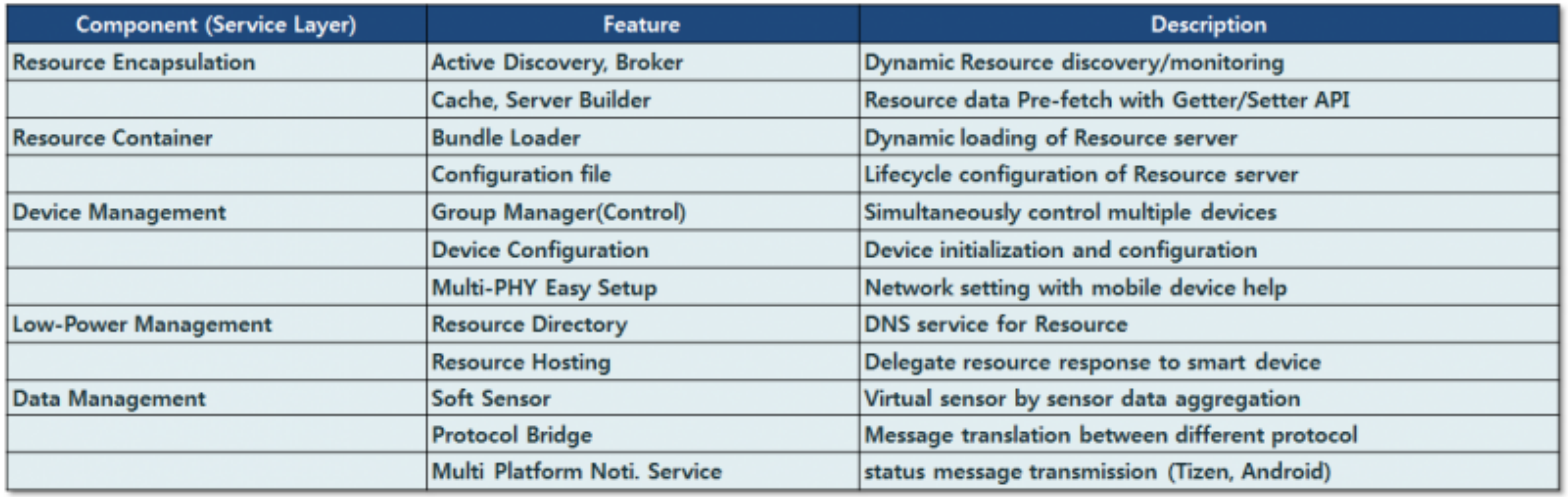




Component (Base Layer)	Feature	Description
Discovery	Multicast Discovery, Device Presence	Discover Resource, check device presence
	Resource Introspection	Resource type/property management
Messaging	CoAP Messaging	Transmit message between devices
	Message switching	Routing thru hetero-connectivity devices
	Block-wise Transfer	Block data transfer (more than 1KB data)
	Connectivity Abstraction	Wi-Fi, BLE, BT abstraction with CoAP
	Remote Access	Home to out of home device connection
	CoAP over TCP	Reliable Transmission, It can be used for messaging between device and cloud
Security	DTLS	Secure data channel with encryption
	Security Resource Manager	Access control(CRUD), Key Management
	Security Provisioning Manager	Transmit credential for authentication

IoTivity Service Layer





Component (Service Layer)	Feature	Description
Resource Encapsulation	Active Discovery, Broker	Dynamic Resource discovery/monitoring
	Cache, Server Builder	Resource data Pre-fetch with Getter/Setter API
Resource Container	Bundle Loader	Dynamic loading of Resource server
	Configuration file	Lifecycle configuration of Resource server
Device Management	Group Manager(Control)	Simultaneously control multiple devices
	Device Configuration	Device initialization and configuration
	Multi-PHY Easy Setup	Network setting with mobile device help
Low-Power Management	Resource Directory	DNS service for Resource
	Resource Hosting	Delegate resource response to smart device
Data Management	Soft Sensor	Virtual sensor by sensor data aggregation
	Protocol Bridge	Message translation between different protocol
	Multi Platform Noti. Service	status message transmission (Tizen, Android)

IoTivity Implementations

Official Implementation

- Latest Release: V1.1.0 on 19/04/2016
- Available for Android, Arduino, iOS, Linux, Tizen, Ubuntu, Windows 8
- APIs in C/C++ and Java
- https://www.iotivity.org/downloads

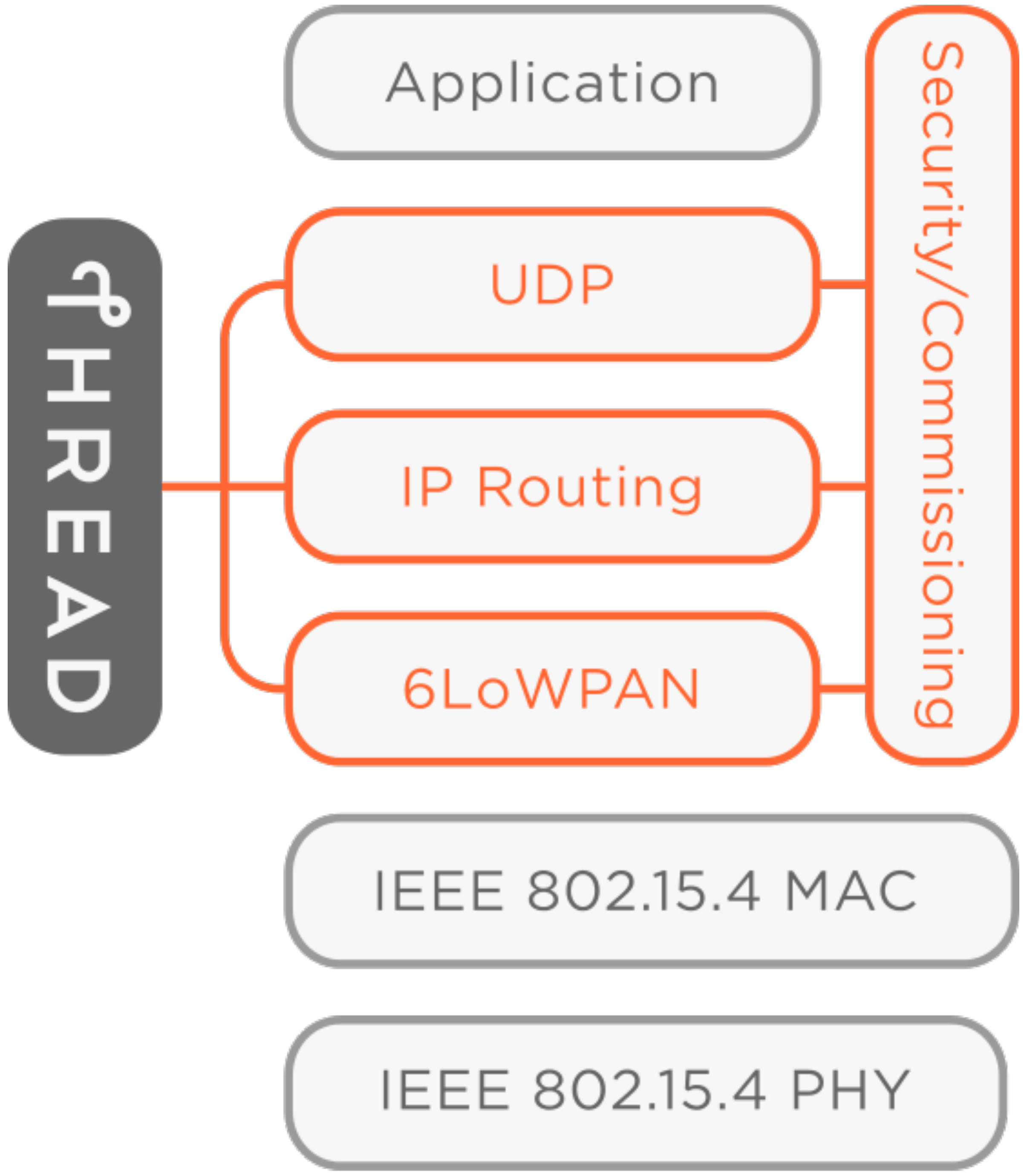
Soletta Project

- Cross-Platform framework, Implements OIC protocol (Compatible with IoTivity)
- Support for Zephyr, Linux and RIOT-OS
- Latest Release: V1_beta19 on 22/04/2016
- Written in C
- Footprint ROM:192 KB RAM:10KB
- https://solettaproject.org/#downloads

THREAD

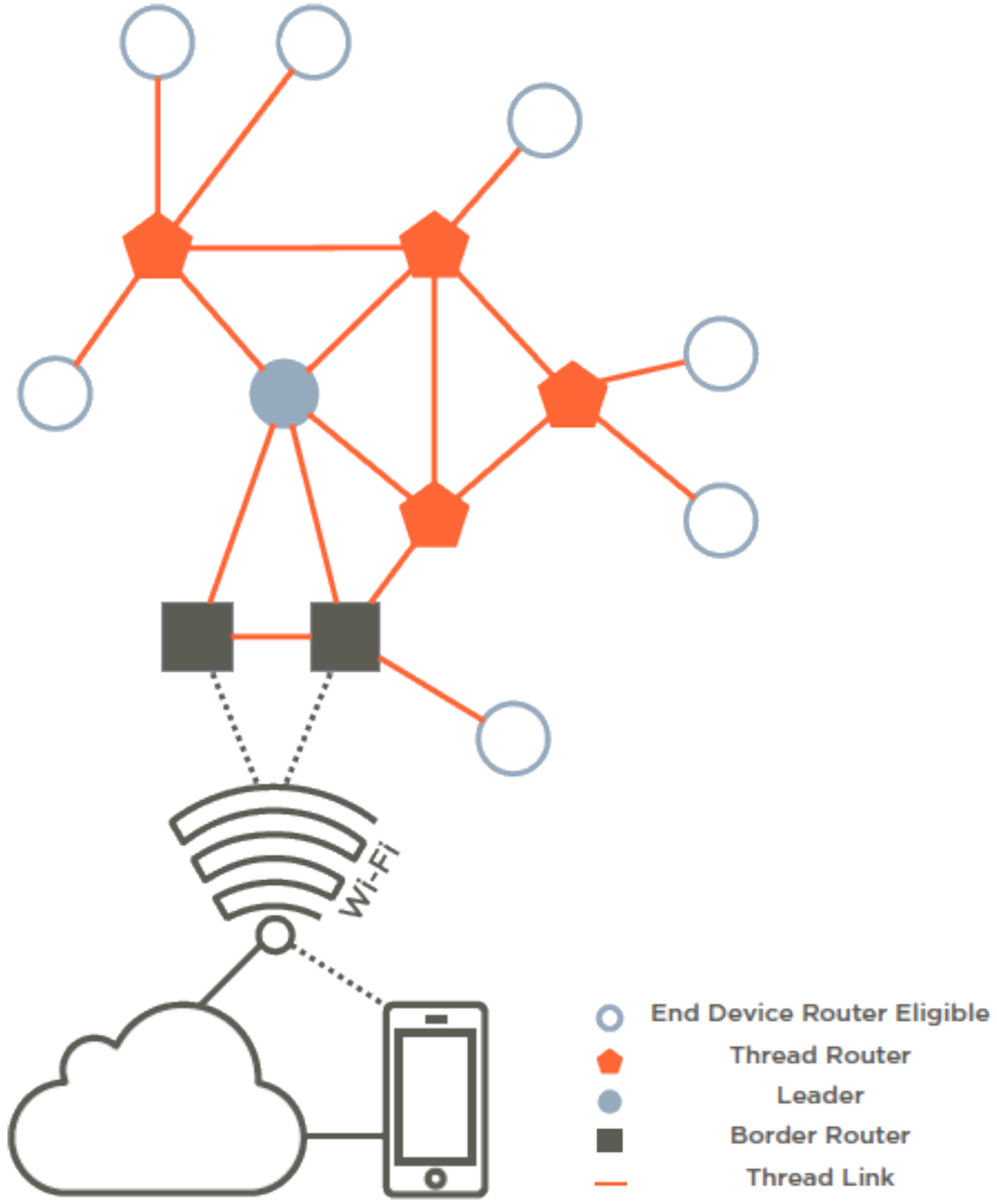
- Open LowPower Networking Stack for IoT Smart Home Automation
- Close-Documentation partnership
- Based on Well-Known Technologies like IEEE802.15.4, 6LoWPAN, IPv6, UDP, …
- Supports Up 250 devices
- Thread can be added in already-deployed devices by Software Update
- Application-Layer agnostic (CoAP, MQTT, XMPP, …)
- Founded by

ARM BIGASS FANS No Equal freescale semiconductor nest SAMSUNG SILICON LABS Yale

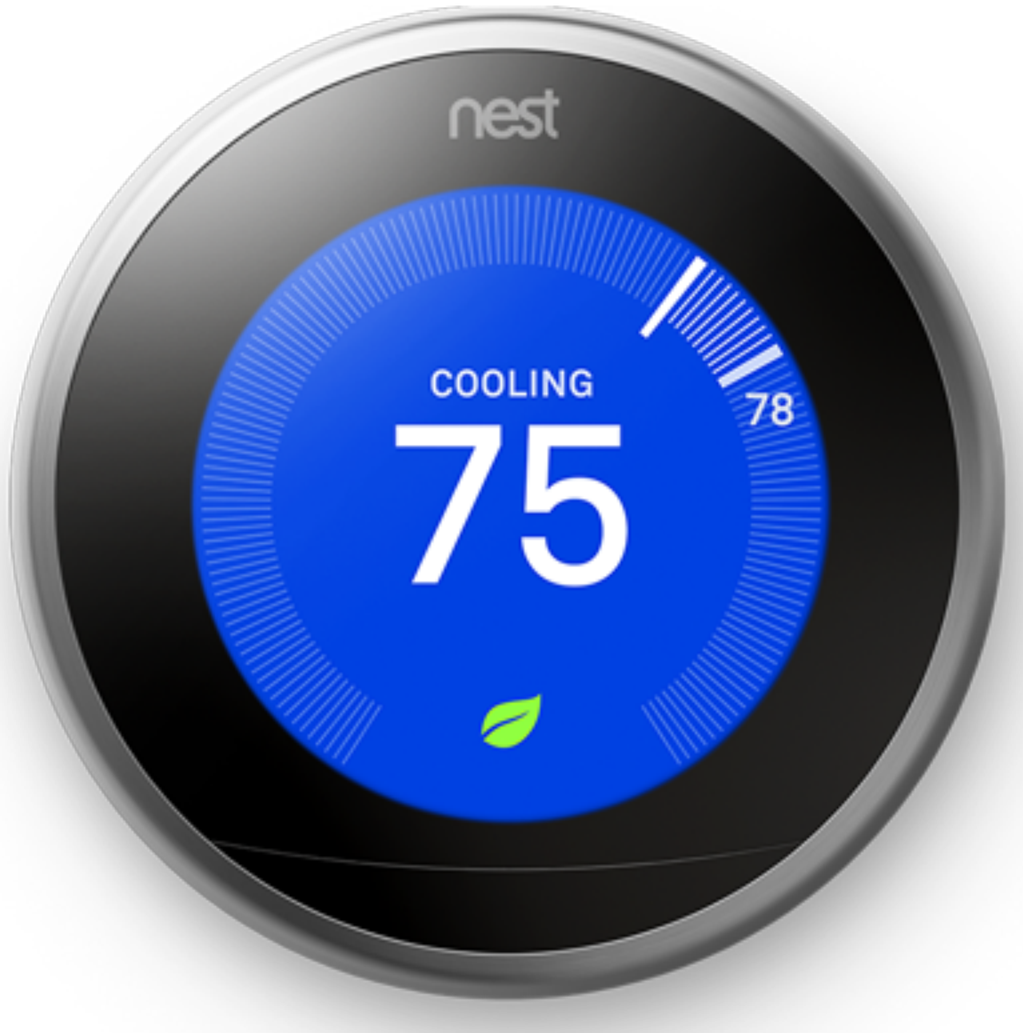


THREAD Characteristics

- IP-Based Mesh Network: IPv6 & 6LoWPAN over IEEE802.15.4 with no single point of failure
- Secure: Smartphone-era Authentication & AES encryption
- Reliable & Scalable: easy to setup & secure to use. Up to 250 devices
- Battery Friendly: extremely low power consumption. Devices can run for years on the smallest batteries
- Designed to support different products for: appliances, access control, climate control, energy management, lighting, safety, and security



THREAD Devices



openthread

released by Nest

- Open-Source Implementation of the Thread Networking protocol
- Released by Nest
- OS and Platform agnostic with Radio abstraction layer
- Implements the End Device, Router, Leader and Border Router roles
- Small memory footprint
- https://github.com/openthread/openthread

→ @jia200x is working on porting for RIOT-OS ←

AllJoyn®

- Collaborative Open-Source Framework for devices and apps to discover and communicate with each other
- Transports, Brands and OSes agnostic
- Runs on popular platform like Linux, Android, iOS, Windows, OS X, OpenWRT

Proximal Network

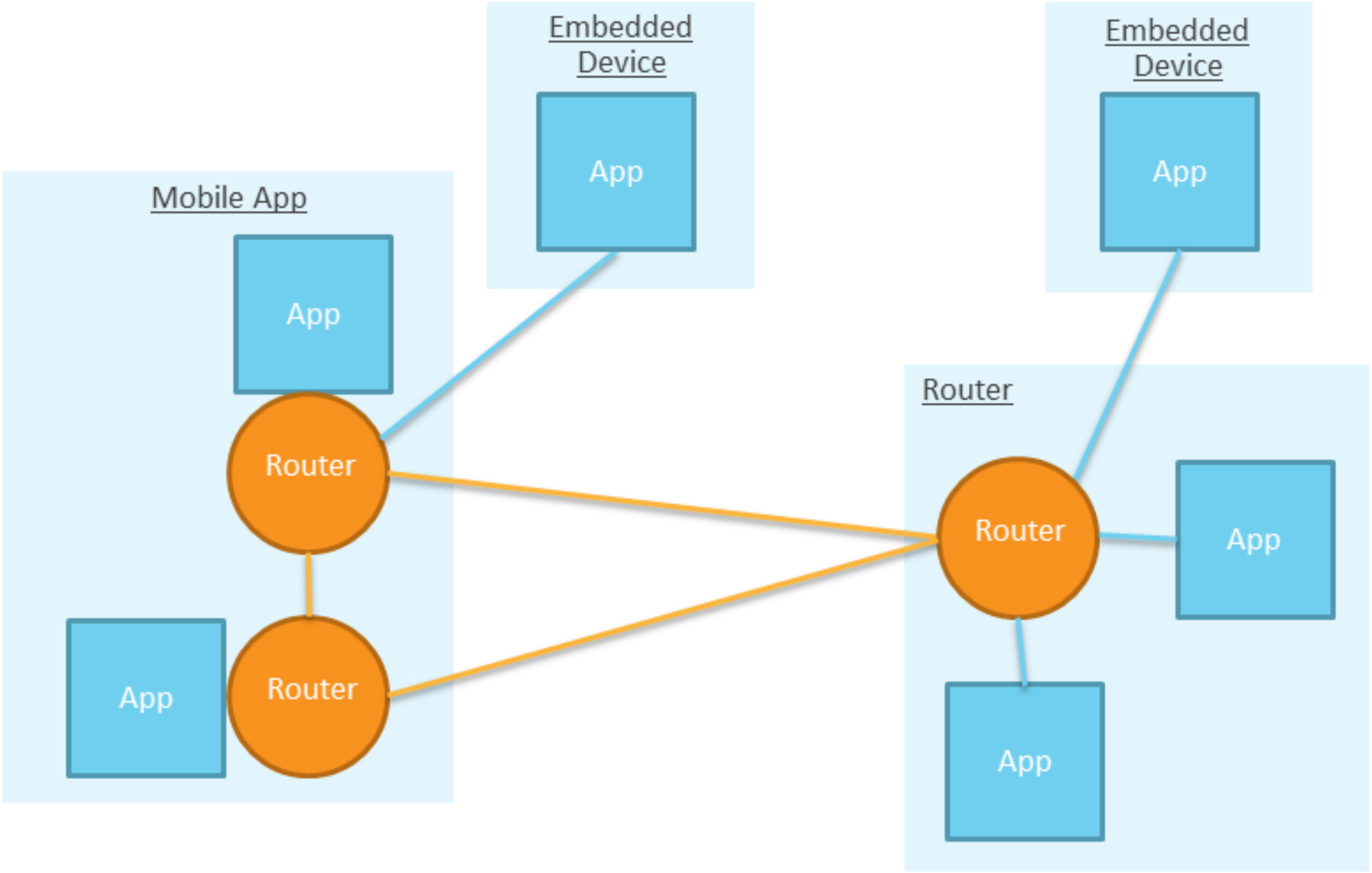
- Handles complexity of discovery
- Creates sessions
- Security framework supports many mechanism

Flexible

- Different transports (Wi-Fi, Eth, serial, PLC)
- Different Languages (C/C++, Java, Obj-C)
- P2P encryption (AES128) and Authentication (PSK, ECDSA)

AllJoyn works on local networks, if the Cloud is required it supports a Gateway Agent.

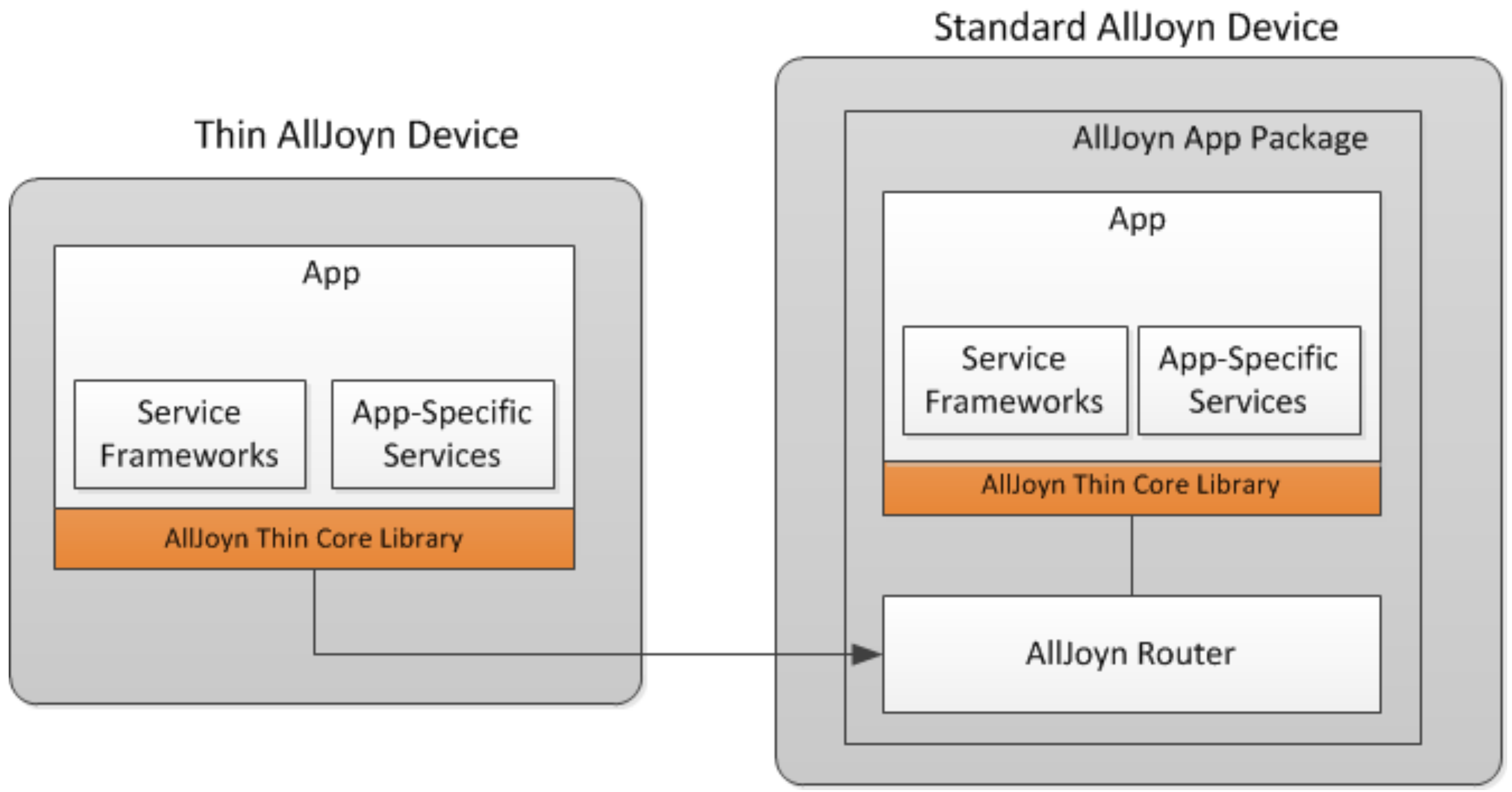
AllJoyn® Network

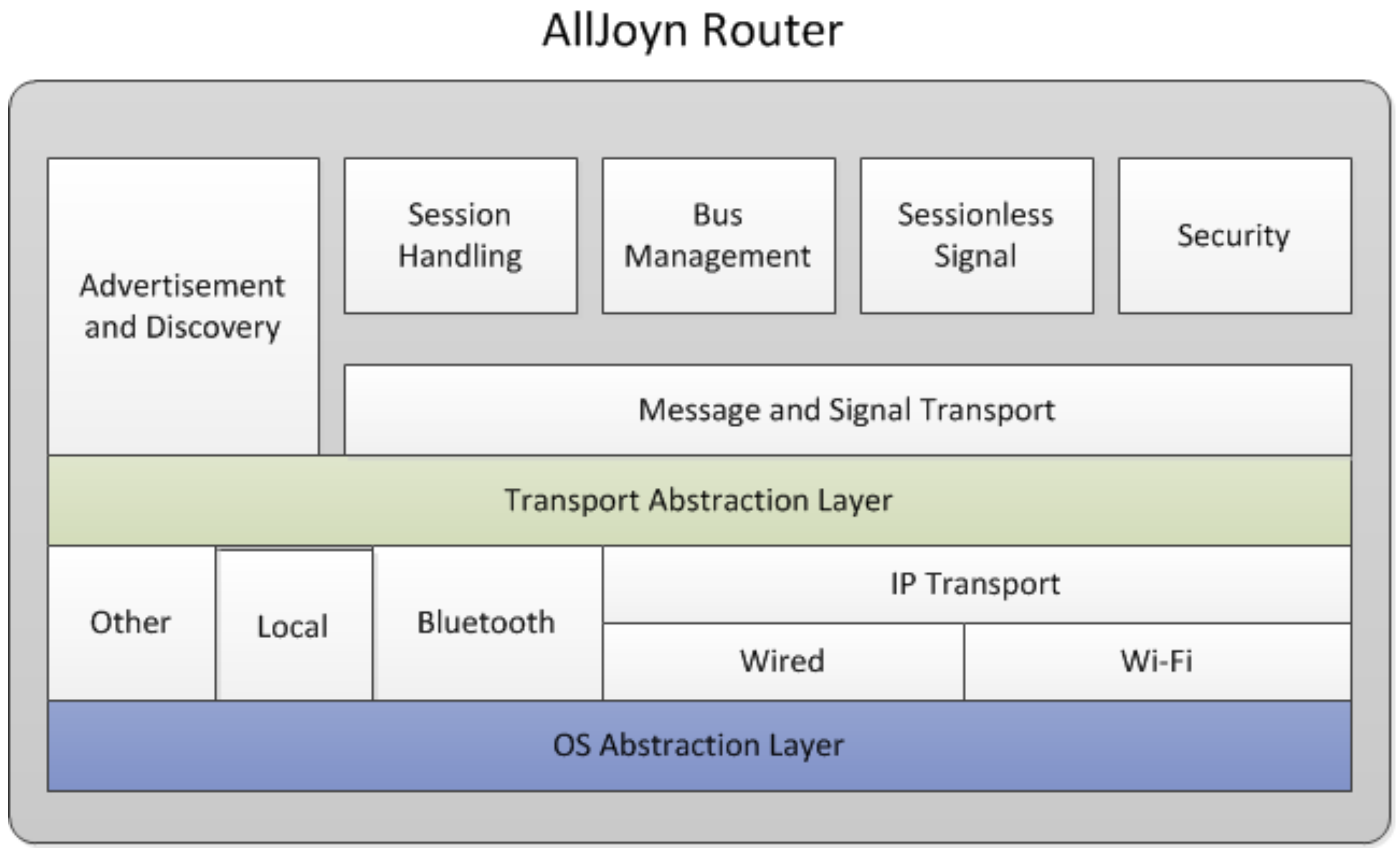


AllJoyn® Architecture

AllJoyn provides 4 components:

- Router: routes messages among apps and routers
- Core Library: low level APIs to interact with AllJoyn Network
- Service Framework Libs: implements services like notification, control panel, configuration
- AppCode: is the logic of the app





Two different kind of Devices:

- Thin: tipically is an embedded device, uses a Standard device as Router
- Standard: is a more powerful device (i.e. Raspberry Pi)

AllJoyn® Official Implementation

- Open-Source Implementation
- Latest Release V16.04 on 3/5/2016
- Implements ThinCore, StandardCore and BaseServices
- Supports for Android, Windows, iOS, OS X, Ubuntu, OpenWRT and JavaScript (coming soon)
- https://allseenalliance.org/framework/download

→ AllJoyn is too heavy for RIOT-OS ←

AllJoyn® Devices

That's All Folks!

Thanks You!

🐦 @Mattia_Antonini

✉ m.antonini@create-net.org