Feasibility Study:

Minimum Viable Device to support Internet of Things Realization

Imprint

Mobile Heights
Mobilvägen 10
Lund

Mats Ekstrand
CEO
mats.ekstrand@mobileheights.org

Mobile Heights is a member-financed organisation based in Lund, Sweden. With Mobile Heights as the foundation, member organizations act in unison to establish Southern Sweden as an internationally leading region in research, innovation and entrepreneurship in mobile communications and its entire value chain in hardware, software and services.

Mobile Heights executes multi-disciplinary projects that creates possibilities for members to cooperate in creating innovative and competitive solutions to the grand challenges of our future society - solutions leading to international competitiveness in business and research. In these projects, the public sector is often an important customer by demanding solutions to complex societal problems.

Minimum Viable Device is one of these projects and in this book the feasability study for the project is presented.

Author

Bogdan Tudosoiu
CTO, Mobile Heights
Bogdan@mobileheights.org

Lund, January 2014

Feasibility Study

Introduction

We see devices, working prototypes, end to end solutions and ideas around the world to connect simple objects of our day-to-day life to Internet, or rather to let them build ad-hoc networks, so that they can collaborate, talk each other, self generating new networks in their environment and act with a certain degree of intelligence. On top of this initial stage of hardware architecture a lot of new, modern business models and opportunities of the Internet of Things (IoT) are raising with high speed. The world was never so connected and this is just beginning… However, the concept of the Internet of Things does not only provide opportunities for numerous industries and domains but also contains challenges. Several complex and scalable technologies are needed for a successful implementation of the IoT and therefore development and research have to improve dramatically to fulfill these needs.

This paper presents a possible solution that needs to be further developed, for enhancing opportunities of the IoT on the one hand, showing how IoT will solve challenges in some industry verticals such as: Smart Home, Mobile Health and Logistics and for a successful, easy to use, open ecosystem implementation on the other hand. Furthermore it presents technical challenges with current technologies and a clear way forward.

Keywords: #IoT, #BLE, #Gateway, #WiFi

Background

The concept of the Internet of Things is not new, first time the term popped up was in 1999, but it is now starting to settle in the process of becoming a reality, though still some key factors must concur to establish.

The Internet of Things, as well as Big Data, has continued to emerge as a trend the consumer electronics sector. Everyone's trying to get into the game, with connected devices all getting connected to “the cloud.” A big problem starts to occur: each industry has its own standard, its own cloud solution its own vertical, completely separated of other industries. If this trend starts to develop, Ericsson, CISCO, Qualcomm vision of Internet of Everything and 50 billion connected devices, road to networked society will crush.

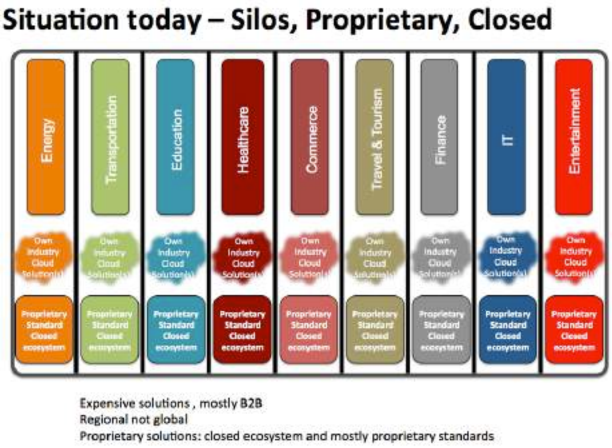

Situation today – Silos, Proprietary, Closed
Energy
Transportation
Education
Healthcare
Commerce
Travel & Tourism
Finance
IT
Entertainment
Own Industry Cloud Solution(s)
Proprietary Standard Closed ecosystem
Expensive solutions , mostly B2B
Regional not global
Proprietary solutions: closed ecosystem and mostly proprietary standards

Fig1. Industry Silos

Anyway, Internet of Things is an exciting trend for consumer electronics in general, but we as an industry need to take a step back and make connectivity as an defacto standard for giving connectivity possibility to extend beyond industry silos, beyond business models and why not beyond just the cloud. Cisco and Ericsson estimate that it will be 50 billion connected devices. Are they going to be connected to Internet? All of them? Same technology?

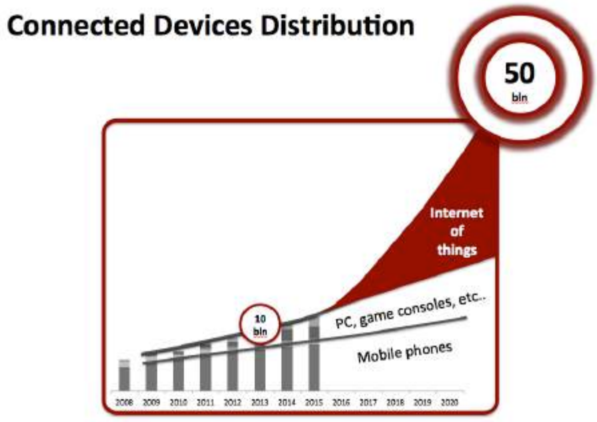


Fig 2. 50 billion connected devices

Just because something is connected to the Internet, doesn't mean it's truly part of an Internet of Things. The vision is to give Things a chance to use the uniqueness of Internet - the openness – the ability for one Thing to link to any other and leverage information in novel ways. You could have one Thing leverage data and APIs from another Thing and mash that up to deliver a completely new, cool service…

Challenges and opportunities in our time

First - It is easier and cheaper than ever to prototype hardware - some components are open sourced (e.g. Arduino microcontrollers); 3D printing helps with fast prototyping and emerging marketplaces can help with distribution. Crowd funding sites like Kickstarter, FundedbyMe or Indiegogo considerably de-risk the early phase of creating hardware by establishing market demand and providing financing.

Second - the world of wireless connectivity boomed over the last few years at a terrible pace. The mobile phone (or tablet), now a supercomputer in everyone's hand, is becoming the universal remote control of the Internet of Things. Pervasive connectivity is becoming a reality (Wi-Fi, Bluetooth, 3G, 4G) and standards are starting to emerge (MQTT, GATT). The slight irony of the “Internet of Things” is that things are often connected via M2M (machine to machine) protocols rather than the Internet itself.

Third - the Internet of Things is able to leverage an entire infrastructure that has emerged in related areas. Cloud computing enables the creation of “dumb” (simpler, cheaper) devices, with all the intelligence processed in the cloud. Big data tools, often open sourced (Hadoop), enable the processing of massive amounts of data captured by the devices and will play a crucial role in the space.

Choosing the projects enabling technology

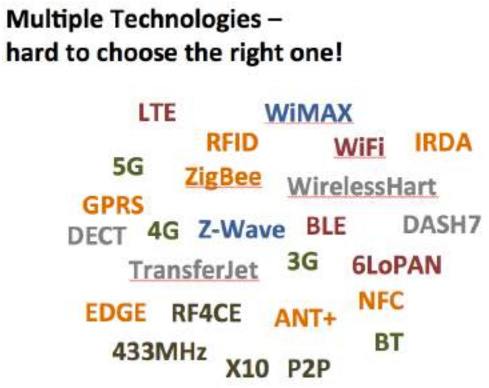


Fig 3 - Technologies

The available technologies spectrum for connecting Things with Internet is endless, but putting them in a context (can be distance, price, ecosystem, etc…) makes the list quite small, but still complex.

Verticals

Unlike the Big Data space, where the action is gradually moving from core infrastructure to vertical applications, the Internet of Things space is seeing a lot of early action directly at the vertical application level as shown in Fig 1. Some notable players like Nest Labs seem to have adopted a deeply integrated vertical strategy where they control key pieces of the product, including both hardware and software, in order to have complete control over the end-user experience (a lot like Apple, which is not surprising considering the founders’ background).

Beyond the Nest, home automation in general has become the central battlefield of the Internet of Things, with some of the most exciting startups in the space jockeying for position. Unfortunately the solutions presented are not scalable and not easy to use, most of the closed (Securitas, Schneider Electric, ABB) making interoperability between Things a nightmare

Another hot consumer-facing area is obviously - quantified self, which is playing a huge role in developing consumers’ awareness of the potential of the Internet of

Things. It has facing exactly same problems as home automation, fast wins, uncorrelated, uncoordinated.

Beyond consumer, B2B/enterprise vertical applications of the Internet of Things, fueled in part by robotics, hold considerable promise in a number of areas such as manufacturing, transportation, healthcare, retail and energy.

Horizontals

While a lot of the action is happening at the vertical application level, the ultimate prize for many ambitious players in the space is to become the software platform upon which all vertical applications in the Internet of Things will be built. For example, several of the home automation providers (SmartThings, Ninja Blocks, etc.) also provide a software platform, and seem to be leveraging their vertical focus as a way to kickstart activity on the platform.

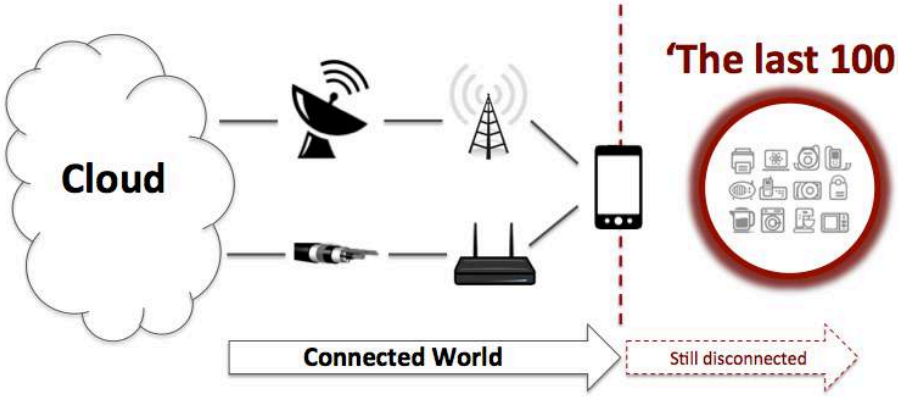
Large corporations (GE, IBM, etc.) are very active in the space and are developing their own platforms. Carriers (AT&T, Verizon) have a large opportunity in the area, as well.

One open question is whether a platform developed for a vertical will easily translate to another vertical. In addition, whether the winning platforms are open or closed will play a huge role in the future of the space.
The space is extraordinarily exciting, but still very much in its infancy – expect this chart to change dramatically over the next few months and years.

Technology

How we came up with right technology for Internet of Things? Out in the market, though myriads of technologies it is hard to find THE Technology for enabling the 50 billion connected devices. The scope is to find out the ONE with best potential to connect Internet of Things, with right enablers in terms of: cost, ease of use, lifetime, reliability, security, ecosystem.

The last 100m

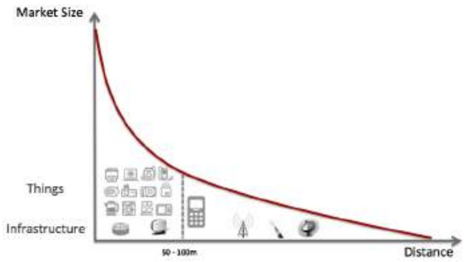


Internet is a great invention of humanity, http protocol created the Internet of people, with easily access both wired and wireless. The coverage is amazing, now we are talking about 2 billion people connected to Internet and possibility to increase to 5 billion in couple of years due to huge technology acceleration especially in wireless technologies.

If we are talking now about 40 exabytes data generated, this will be a drop in the ocean when we are going to add another 3 billion people and up to 50 billion things.

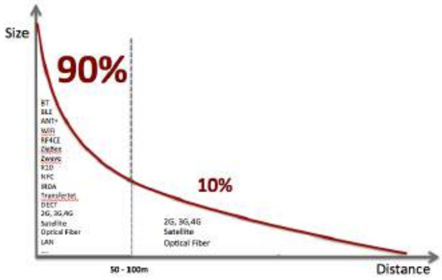
Most of data will be generated by things around you, things that are close to you in the house, at work, in the car. All these things are still not connected to Internet. So the question to answer is:

How to connect things in 'the last mile'



Things that are still not connected are close to us, can be a chair, a piano, a light bulb, a remote, a toaster, etc… most of them a 'silent' data generator with no display and sending quite low amount of data.

We estimate that the market share of these things will be definitely more than 90% of all the other already connected devices, but the amount of existing technologies to couple them is extremely diversified.

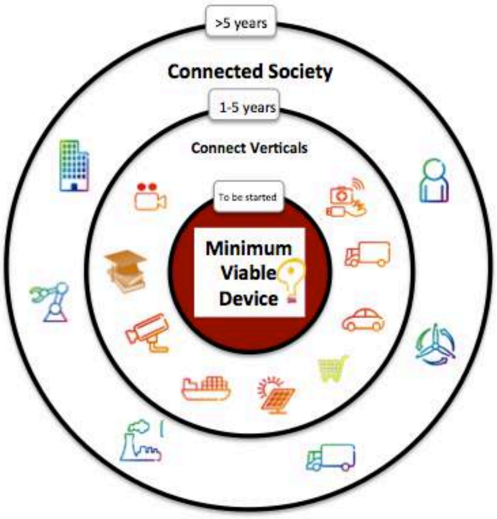


To be able to propose the right technology we need to know, what kind of devices are going to be connected and we estimate that the main driver for connected the things will be sensors. The right technology must be able to create wireless sensor networks.

The concept of wireless sensor networks is not new, started in 1990 but the realization was not possible until these days.

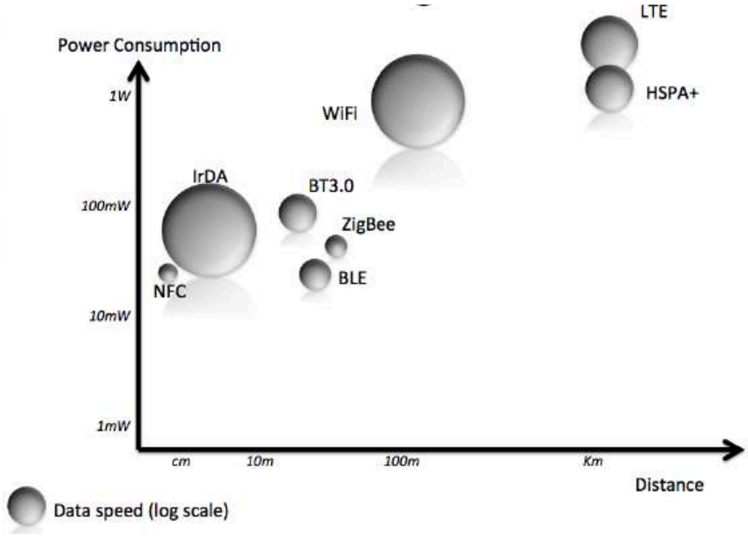
The main drivers for wireless Sensor network realization are:

Cost
Ease of use
Reliability
Lifetime
Ecosystem
Security



Technology ecosystem for Wireless Sensor Networks

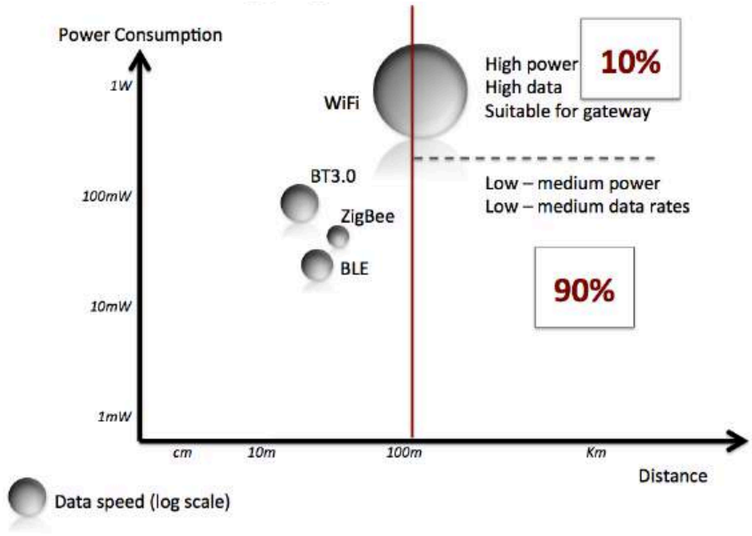
Let's take a look at the available technologies that fulfills the drivers presented above.



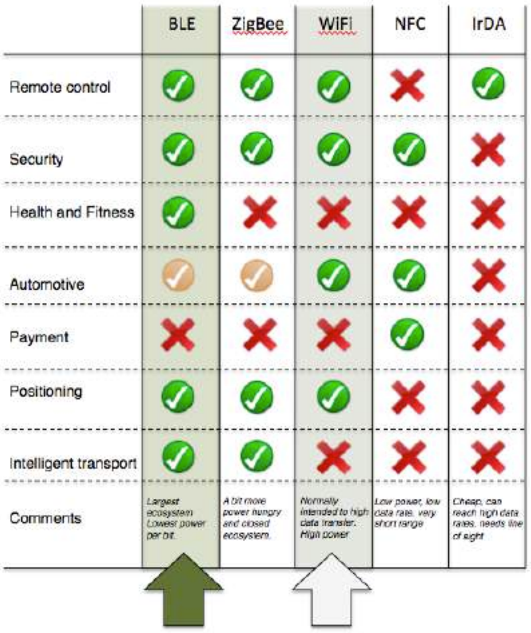
The figure above is showing technologies with big ecosystems, already established. Looking into parameters characterizing wireless sensor networks we can sort out even more technologies:

As we have presented our focus will be on 'the last mile' and especially things in 100 m range. So excluding the cellular technologies like, GSM, HSPA+, LTE Definitely, NFC (even RFID) and IrDA are presenting big potential but the main use case for them will be line of sight or proximity (up to 2m) making these 2 standards now fulfilling the required range.

So, the conclusion - we need to focus on following technologies:



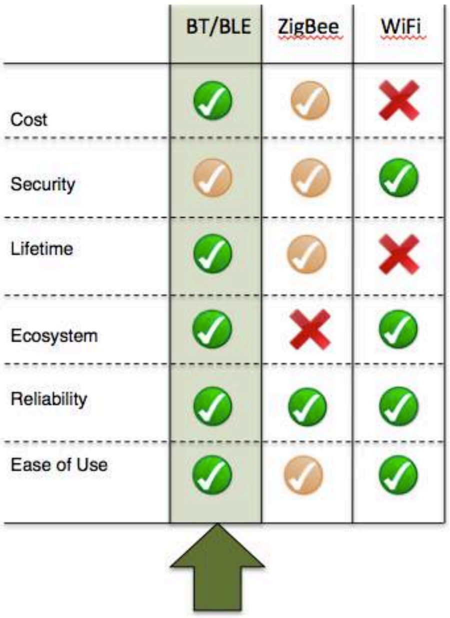
Going further, the need for a huge ecosystem for Internet of Things realization is coming from technology of choice possibility to fulfill a big feature set. Here is a mapping of main features (a.k.a. industry verticals) versus technologies presented above: (Next page)



	BLE	ZigBee	WiFi	NFC	IrDA
Remote control	✓	✓	✓	✗	✓
Security	✓	✓	✓	✓	✗
Health and Fitness	✓	✗	✗	✗	✗
Automotive	✓	✓	✓	✓	✗
Payment	✗	✗	✗	✓	✗
Positioning	✓	✓	✓	✗	✗
Intelligent transport	✓	✓	✗	✗	✗
Comments	*Largest ecosystem Lowest power per bit.*	*A bit more power hungry and closed ecosystem.*	*Normally intended to high data transfer. High power*	*Low power, low data rate, very short range*	*Cheap, can reach high data rates, needs line of sight*

The clear choice is Bluetooth low energy – known also as Bluetooth Smart for wireless sensor network realization. In the picture above WiFi as a technology is also highlighted as a natural option as infrastructure for wireless network realization. In the next coming chapters this option will be clarified as a technology of choice for increasing Bluetooth range.

Going back to key factors for wireless sensor networks realization per technology:



	BT/BLE	ZigBee	WiFi
Cost	✅	✓	❌
Security	✓	✓	✅
Lifetime	✅	✓	❌
Ecosystem	✅	❌	✅
Reliability	✅	✅	✅
Ease of Use	✅	✓	✅

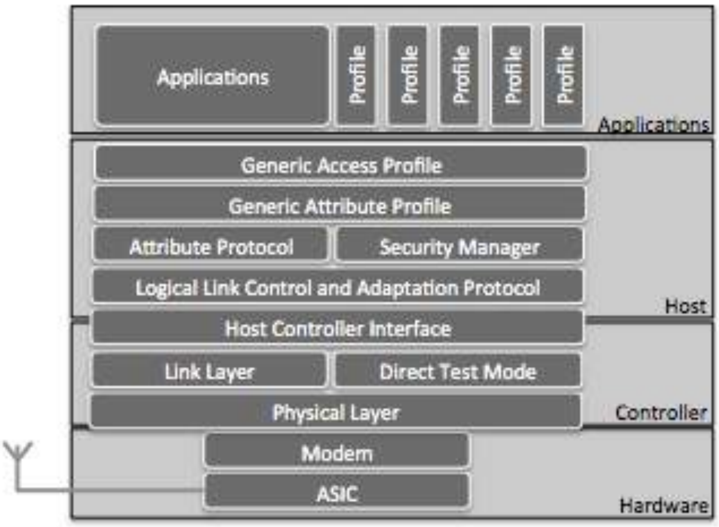
From comparisons above it is obvious that the technology of choice will be Bluetooth Smart or Bluetooth low energy.

Bluetooth Low Energy

Bluetooth low energy technology is a key feature of the Bluetooth Core Specification 4.0 (Bluetooth v4.0) and has inherited several technical features from Classic Bluetooth technology and provides robust, reliable connections in tough environments. Examples of inherited features include the Bluetooth radio with Adaptive Frequency Hopping (AFH), Logical Link Control and Adaptation Protocol (L2CAP) interface.
It is designed to work last long with energy from for example a simple coin cell to last years. Bluetooth LE is targeting healthcare, sports and fitness, security and home entertainment applications. It dropped support for headsets and for streaming audio data, but never than less dual circuits are existing to support both Bluetooth low energy and classic Bluetooth so the feature set as presented in the chapters above is huge, almost unlimited.

Bluetooth low energy facts:

- Very low power consumption.
- High numbers of communication nodes with limited latency requirements.
- Robustness close to classic Bluetooth technology.
- Short wake-up / connection time



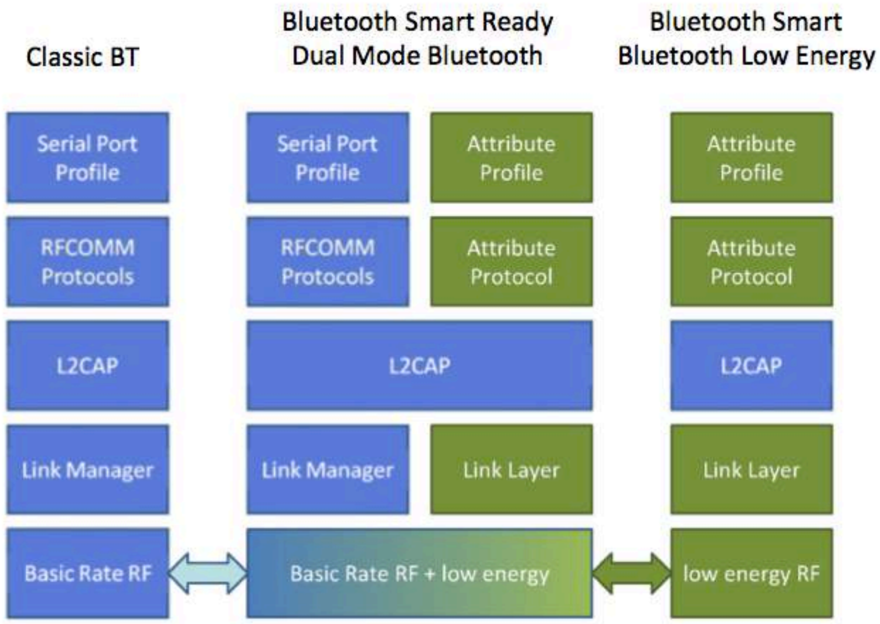
Bluetooth Low Energy (BLE) power consumption

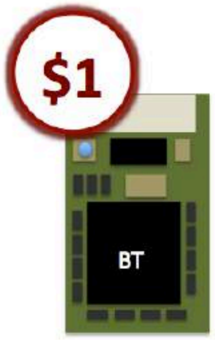
Classic Bluetooth is connection oriented. When a device is connected, a link is maintained, even if there is no data flowing. Sniff mode allow devices to sleep, reducing power consumption to give some months of battery life. In classical Bluetooth the peak TX current is typically around 25 mA. Even though it has been independently shown to be lower power than other standards, it is still not low enough power for coin cells and energy harvesting applications
Bluetooth low energy solution solved these concerns. Independent of technology (LTE and LTE Advanced are some exceptions) it is the RF block consuming most of energy. Bluetooth low energy solved this by keeping the device in sleep mode for most of the time. When an event occurs, RF will be up; it sends a short message to the gateway and down to sleep again. Compared to Classic BT, BLE max peak current is about 15 mA and the average power consumption is about 1µA. In low duty cycle applications a coin cell battery can last 5-10 years!

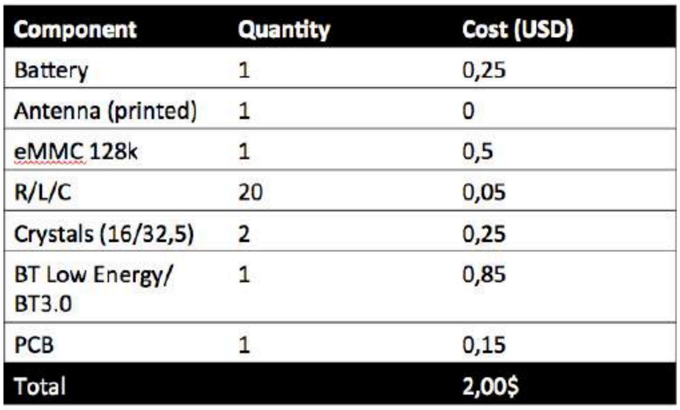
Bluetooth Low Energy cost

In order to offer compatibility with Classic BT two kinds of devices are available:

1. Stand alone BLE
2. Dual mode devices including both Classic BT and BLE







Component	Quantity	Cost (USD)
Battery	1	0,25
Antenna (printed)	1	0
eMMC 128k	1	0,5
R/L/C	20	0,05
Crystals (16/32,5)	2	0,25
BT Low Energy/ BT3.0	1	0,85
PCB	1	0,15
Total		2,00$

Cost target for a dual mode Classic BT and BLE device

Due to low cost and easy to implement plus already existing circuits from known manufacturers such as: Nordic, TI, CSR, Qualcomm, Broadcom etc. Bluetooth Smart ready (dual mode) and Bluetooth Smart (Bluetooth Low Energy) are already in the market creating a huge ecosystem.
Know operating system such as iOS (Apple), Windows, Android and BBRY 10 and up have implemented API for supporting Bluetooth v4.0 and already deployed into market.

Some devices supporting Bluetooth v4.0:

Examples of Bluetooth Smart Ready products:

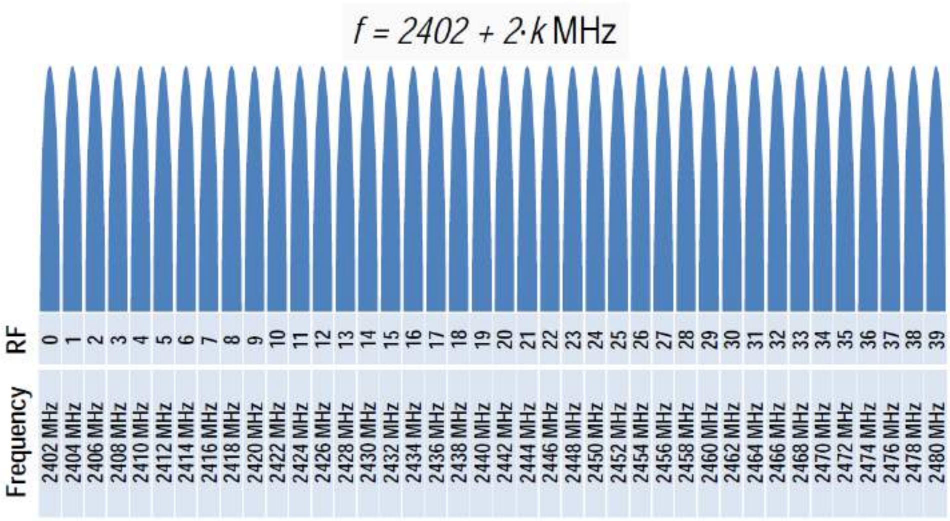
Examples of Bluetooth Smart products:

Bluetooth Low Energy – PHY Layer

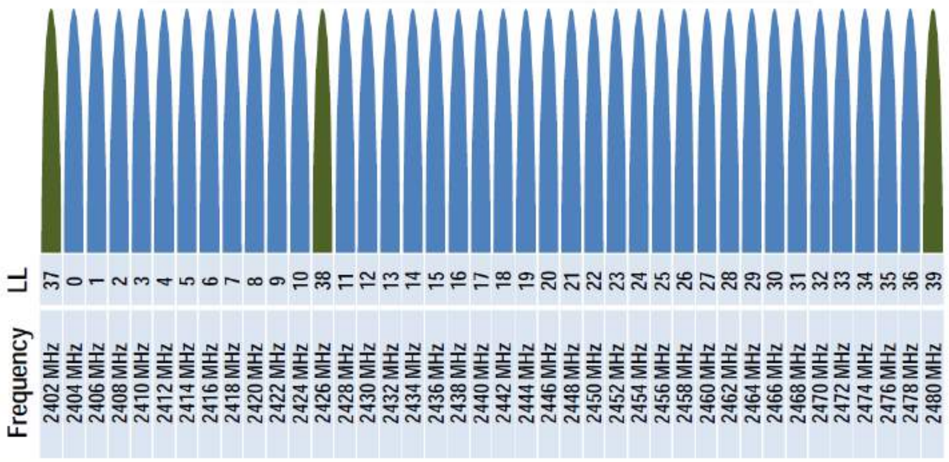
Bluetooth Low Energy uses Adaptive Frequency Hopping (AFH), same as Classic Bluetooth. This is one of clear advantages compared to Zigbee as it gives better robustness for transmission and reception in noisy environments.

Characteristics:

- 2.4 GHz ISM band
- 1Mbps GFSK – Larger modulation index than Bluetooth BR (which means better range)
- 40 Channels on 2 MHz spacing compared to 79 channels – 1 MHz wide in Classic BT, improving power consumption

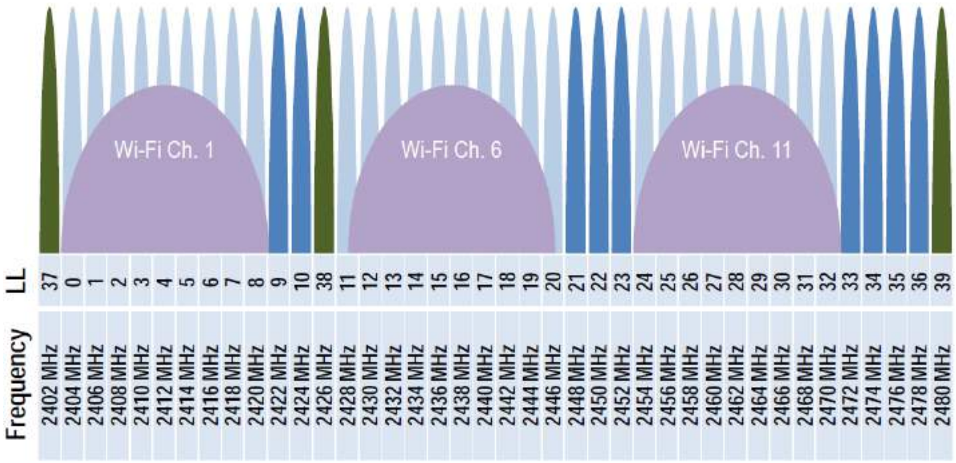

f = 2402 + 2·k MHz
RF
Frequency
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
2402 MHz 2404 MHz 2406 MHz 2408 MHz 2410 MHz 2412 MHz 2414 MHz 2416 MHz 2418 MHz 2420 MHz 2422 MHz 2424 MHz 2426 MHz 2428 MHz 2430 MHz 2432 MHz 2434 MHz 2436 MHz 2438 MHz 2440 MHz 2442 MHz 2444 MHz 2446 MHz 2448 MHz 2450 MHz 2452 MHz 2454 MHz 2456 MHz 2458 MHz 2460 MHz 2462 MHz 2464 MHz 2466 MHz 2468 MHz 2470 MHz 2472 MHz 2474 MHz 2476 MHz 2478 MHz 2480 MHz

- 2 type of channels: 3 advertising channels and 37 data channels


LL
Frequency
37 0 1 2 3 4 5 6 7 8 9 10 38 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 39
2402 MHz 2404 MHz 2406 MHz 2408 MHz 2410 MHz 2412 MHz 2414 MHz 2416 MHz 2418 MHz 2420 MHz 2422 MHz 2424 MHz 2426 MHz 2428 MHz 2430 MHz 2432 MHz 2434 MHz 2436 MHz 2438 MHz 2440 MHz 2442 MHz 2444 MHz 2446 MHz 2448 MHz 2450 MHz 2452 MHz 2454 MHz 2456 MHz 2458 MHz 2460 MHz 2462 MHz 2464 MHz 2466 MHz 2468 MHz 2470 MHz 2472 MHz 2474 MHz 2476 MHz 2478 MHz 2480 MHz

The smart thinking with Bluetooth low energy advertising channels is the placement in order to avoid WiFi (802.11 b/g) channels. It is known that ISM band (2.4 GHz specifically), du to its license free characteristics has a lot of interference that can decease data rates and latency due to more retransmissions and error corrections. Bluetooth low energy is created to avoid as much as possible such problems by:

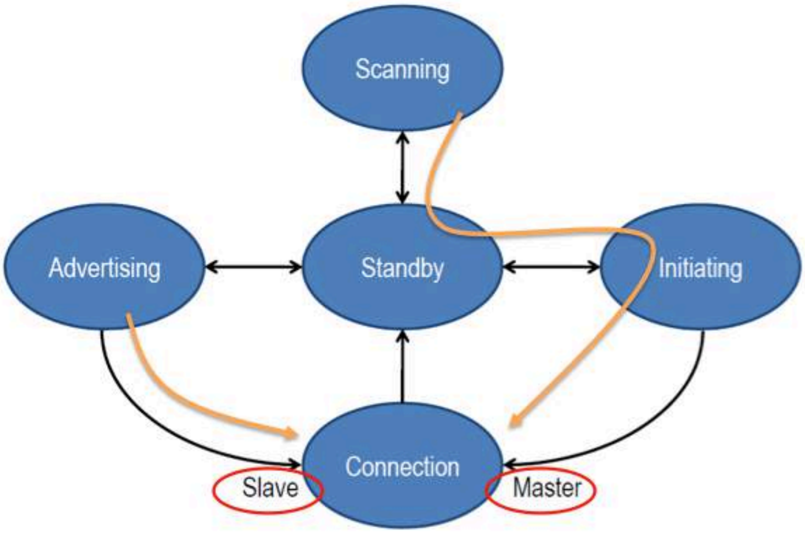
1. Placement of advertising channels
2. Frequency hopping



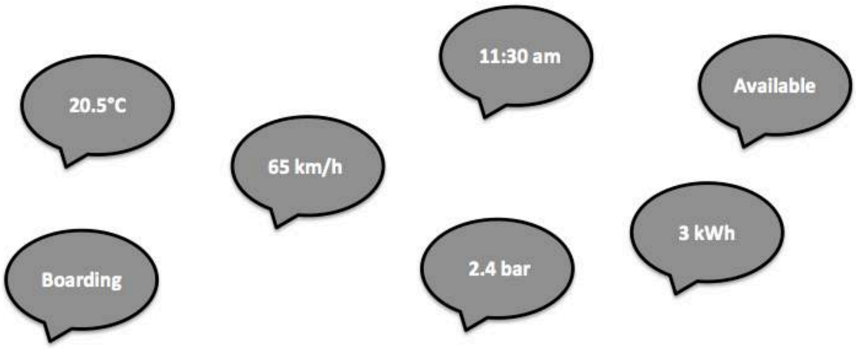
Bluetooth Low Energy – Link Layer

The new Link layer introduced by Bluetooth low energy has low complexity and plenty of useful features. The connection is as in classical Bluetooth, a star topology where a master is connected to a number a slaves. A device can be either a master or slave, never both. There is no scatternet topology for Bluetooth low energy.

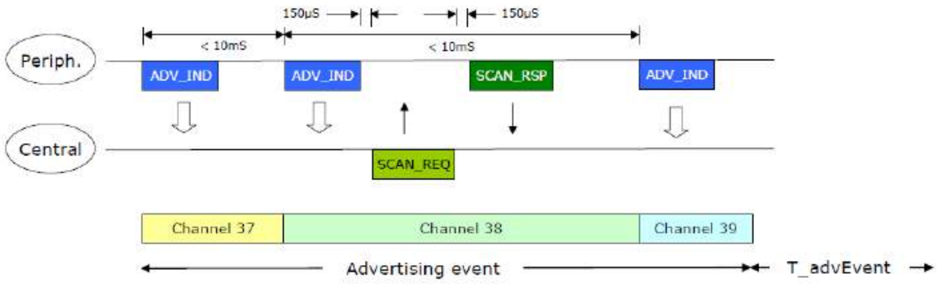
Anyway, Bluetooth low energy introduces a new ‘state’ called advertising compared to classical Bluetooth (Scanning, Initiating, Connection).



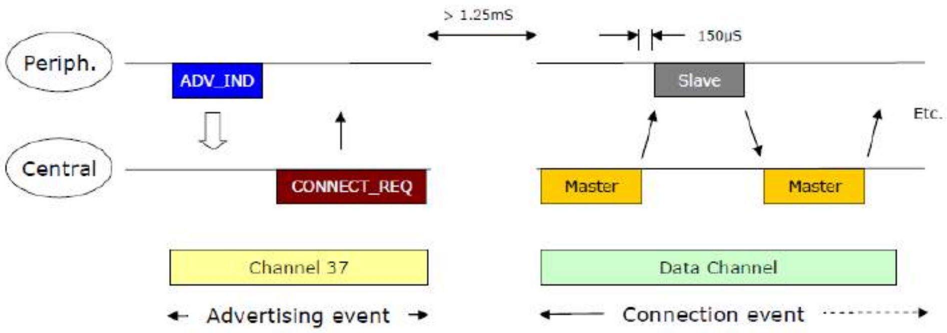
A ‘thing’ acting as slave can ‘announce’ it has something to transmit to the master. This announcement can be an event, presence, a measurement value or reconnect asynchronously. Here are some examples:



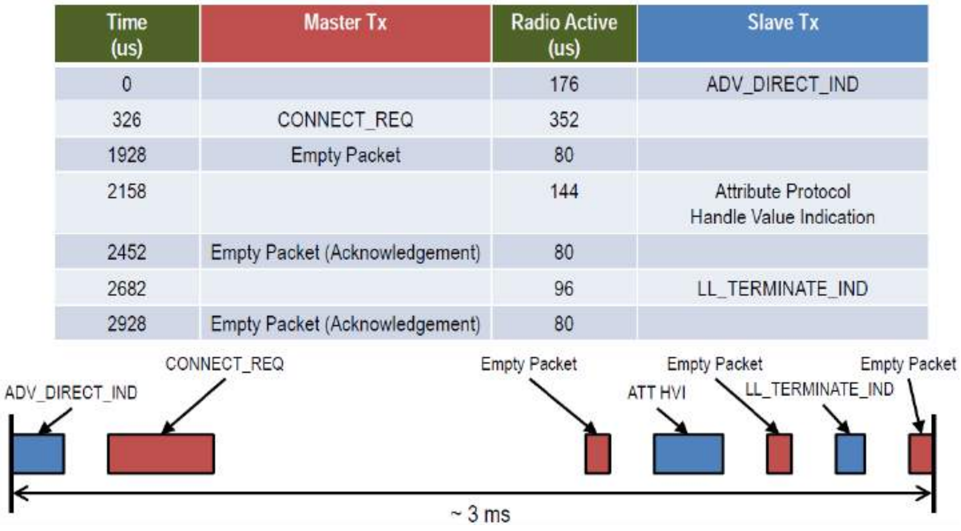
How does it work?



Once the connection was made, master informs slave on hopping sequence and when to wake. All subsequent transactions are performed in a data channel of 37 available and indicated by the hopping sequence. Transactions can be encrypted and after the transaction realization both can go in deep sleep mode



The beauty of these advertisements, beside richness of possible events, messages and other information to be transmitted is the low latency. The minimum transaction is taking… 3 ms.



Time (us)	Master Tx	Radio Active (us)	Slave Tx
0		176	ADV_DIRECT_IND
326	CONNECT_REQ	352	
1928	Empty Packet	80	
2158		144	Attribute Protocol Handle Value Indication
2452	Empty Packet (Acknowledgement)	80	
2682		96	LL_TERMINATE_IND
2928	Empty Packet (Acknowledgement)	80	

Bluetooth Low Energy - ATT/GATT

Bluetooth Low Energy specification, brings two new core protocols: ATT (Attribute Protocol) and GATT (Generic Attribute Profile). They are mainly targeted for Low Energy, and every LE profile is expected to use them. But they can also be used over classic Bluetooth (BR/EDR).

ATT is a wire application protocol, while GATT dictates how ATT is employed in service composition. Every Low Energy profile must be based on GATT.
So, ultimately, every LE service uses ATT as the application protocol.

Locking profiles into these protocols brings several advantages:

- Development and implementation of new LE profiles is much easier, since there is no wire protocol to do from scratch;
- ATT is optimized to run on Low Energy devices: it uses as few bytes as possible, and implementation may use fixed-size structures in memory to make data packets (PDUs).
- ATT/GATT simplicity means that firmware may offer some degree of ATT/GATT assistance, saving the microcontroller software from the trouble.
- For software-based stacks, ATT/GATT may be implemented only once in stack itself, saving applications from the trouble.
- There may be profiles for which ATT/GATT is not ideal as the application protocol. But there can always be a second L2CAP connection in parallel with ATT channel, which in turn implements a profile-specific protocol.

Now, let's take a deeper look into each protocol.

ATT: Attribute Protocol

Most of the ATT protocol is pure client-server: client takes the initiative, server answers. But ATT has notification and indication capabilities, in which the server takes the initiative of notifying a client that an attribute value has changed, saving the client from having to poll the attribute.

The sole building block of ATT is the attribute. An attribute is composed by three elements:

- a 16-bit handle;
- an UUID which defines the attribute type;
- a value of a certain length.

From the point of view of ATT, value is amorphous; it is an array of bytes of any size. The actual meaning of the value depends entirely on UUID, and ATT does not check if the value length is consistent with a given UUID etc.

The handle is just a number that uniquely identifies an attribute (since there may be many attributes with the same UUID within a device).

ATT itself does not define any UUID. This is left to GATT and higher-level profiles.

An ATT server stores attributes. An ATT client stores nothing; it uses the ATT wire protocol to read and write values on server attributes.

There may be security permissions associated with each attribute. They are stored somewhere inside the value, and are defined by higher-level profiles. ATT itself does not "know" them, and does not try to interpret attribute values to test permissions. This is GATT's (and higher profile's) problem.

ATT wire protocol has some nice features:

- Search attributes by UUID
- Get all attributes given a handle range

and so on, so the client does not need to know handle numbers beforehand, nor the higher-level profiles have to hardcode them.

But handle numbers are expected to be stable for each given device. This allows clients to cache information, to use fewer packets (and less energy) to retrieve attribute values after a first discovery.

Higher-level profiles specify how to "hint" a client that a server has changed attribute layout (e.g. after a firmware upgrade).

The wire protocol never sends value length; it is always implied from PDU size, and client is expected to "know" the exact value layout for the UUID types it understands. Not sending value length explicitly saves bytes, which is particularly important in Low Energy, since MTU (maximum transmission unit) in LE is just 23 bytes.

ATT is very generic, and would leave too much for higher-level profiles to define. Apart from the excess of freedom, there are some open issues, like: what if a device offers multiple services? There is just one ATT handle space for each device, and multiple services must share the space in a cooperative way.

Fortunately, we have GATT, which shapes and delimits usage of attributes.

GATT: Generic Attribute Profile

GATT is a base profile for all top-level LE profiles. It defines how a bunch of ATT attributes are grouped together into meaningful services.

GATT services

The cornerstone of a GATT service is the attribute with UUID equal to 0x2800. All attributes following this belong to that service, until another attribute 0x2800 is found.

Each attribute does not "know" by itself to which service it belongs. GATT needs to figure it out based on handle ranges, and ranges are discovered solely on basis of UUID 0x2800 "cornerstones".

Ok, how do I know if a given service is a thermometer, of keyfob, or GPS? By reading its value. The service attribute value contains an UUID, the service UUID.

The UUID 0x2800, which is well known by GATT, is used to search for service boundaries. Once they are found, the attributes are read and the second UUID (stored as value) specifies the service. So a client may find all GATT services without knowing the specifics of e.g. a thermometer service.

Each GATT service has a number of characteristics. The characteristics store useful values for the services, as well as their permissions.

For example, a thermometer would likely have a "temperature" characteristic, which is read-only, and possibly a date/time for time stamping, which is read/write.

Handle	UUID	Description	Value
0x0100	0x2800	Thermometer service definition	UUID 0x1809
0x0101	0x2803	Characteristic: temperature	UUID 0x2A2B

			Value handle: 0x0102
0x0102	0x2A2B	Temperature value	20 degrees

First off, there may be several characteristics per service, and each handle ranges for each characteristic are discovered by GATT the same way it does for services: by finding the "cornerstone" attributes.

The main characteristic attribute has UUID = 0x2803. As happens with services, this attribute has "double UUIDs": the generic one (0x2803) which allows for easy discovering, and the specific one (in example: 0x2A2B for temperature) which tells exactly which information the characteristic contains.

Each characteristic has at least two attributes: the main attribute (0x2803) and a value attribute that actually contains the value. The main attribute "knows" the value attribute's handle and UUID. This allows for a certain degree of cross-checking.

The actual value format is entirely defined by its UUID. So, if the client knows how to interpret the value UUID 0x2A08, it is capable of reading date and time from any service that contains such a characteristic. In the other hand, if the client does not know how to interpret a certain value UUID, it may safely ignore it.

Characteristic descriptors

Apart from value, we can hang more attributes in every characteristic, if we need them. In GATT lingo, those extra attributes are called descriptors.

For example, we may need to identify the temperature unit of our thermometer, and this may be carried out by a descriptor:

Handle	UUID	Description	Value
0x0100	0x2800	Thermometer service definition	UUID 0x1809
0x0101	0x2803	Characteristic: temperature	UUID 0x2A2B Value handle: 0x0102
0x0102	0x2A2B	Temperature value	20 degrees
0x0104	0x2A1F	Descriptor: unit	Celsius

a) It is not the value attribute, since the value attribute is known to be 0x0102; and
b) It falls into the range 0x0103..0x010F, which falls between one characteristic and the next.

Each service may define its own descriptors, but GATT defines a set of standard descriptors that cover most cases, for example:

- Numeric format and presentation;
- Human-readable description;
- Valid range;
- Extended properties;

and so on.

One particularly important descriptor is the *client characteristic configuration*.

Client Characteristic Configuration descriptors

This descriptor, who's UUID is 0x2902, has a read/write 16-bit value, which is meant to be a bitmap.

It is not some kind of client-side descriptor. It is server-side as any other attribute. But the server is required to store and present a separate instance of the value for each bonded client, and each client can only see its own copy.

First two bits of CCC are already taken by GATT specification. They configure characteristic notification and indication. The other bits might be used for other functions, but they are currently reserved.

Handle	UUID	Description	Value
0x0100	0x2800	Thermometer service definition	UUID 0x1809
0x0101	0x2803	Characteristic: temperature	UUID 0x2A2B Value handle: 0x0102
0x0102	0x2A2B	Temperature value	20 degrees
0x0104	0x2A1F	Descriptor: unit	Celsius
0x0105	0x2902	CCC descriptor	0x0000

As usual, GATT knows that CCC belongs to temperature characteristic because the handle falls into the range (0x0102..0x010F) and it knows it is CCC because of the distinctive UUID (0x2902).

Service discovery in Low Energy

Since GATT puts all service details on ATT, there is no need for a separate service discovery protocol (SDP), like we have in BR/EDR. The ATT protocol is used for everything: discovering services, finding services' characteristics, reading/writing values, and so on.

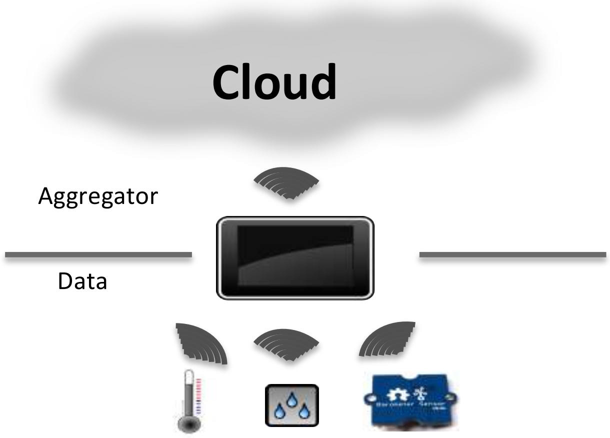
In Generic Attribute Profile (GATT) service groups, features and declarations are brought together to specify a se of features available in the devices. Through these attributes, it is possible to build numerous services and profiles as_

- Proximity
- Time
- Automation
- Lighting
- Remote
- Find me
- Fitness
- Medical devices
- Battery

And so on

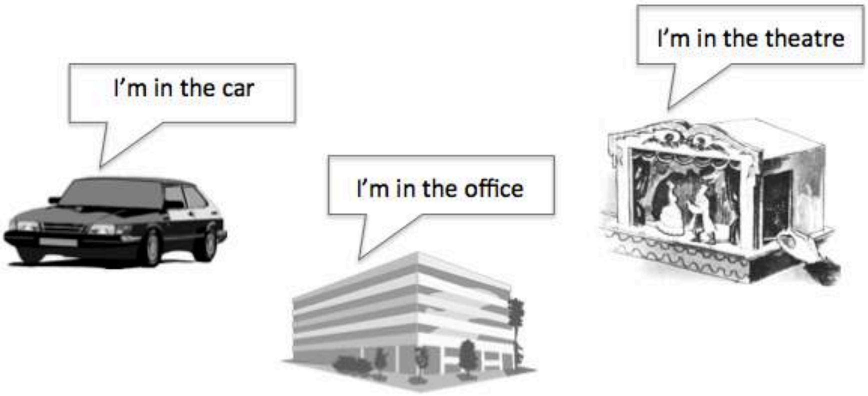
Example

- Home data aggregator



Enabling sensors in the house (each room) is possible due to reduced power and cost of Bluetooth low energy devices. Sensors around the house can be used for optimizing current consumption as they provide real time feedback to home automation devices.

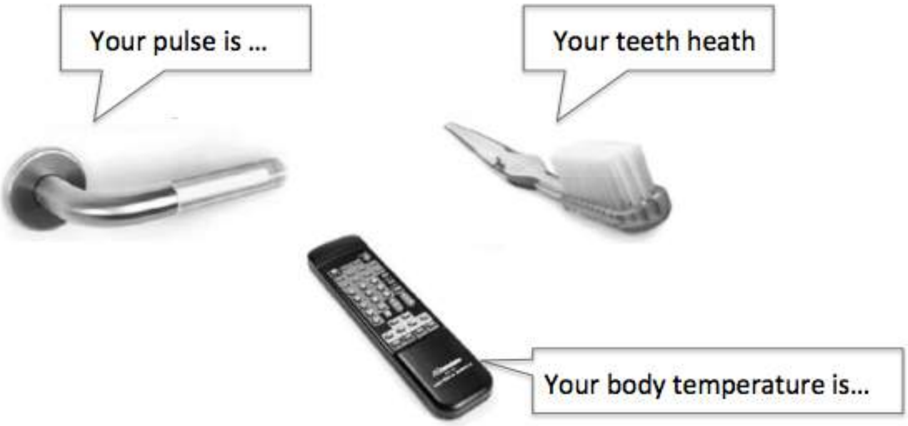
- Proximity and detection



It can detect also presence, for example:

- Turn on/off lights when I'm entering/leaving a room
- Locks the door when I'm leaving home or office
- Turn off alarm if already awake
- Find my phone or keys

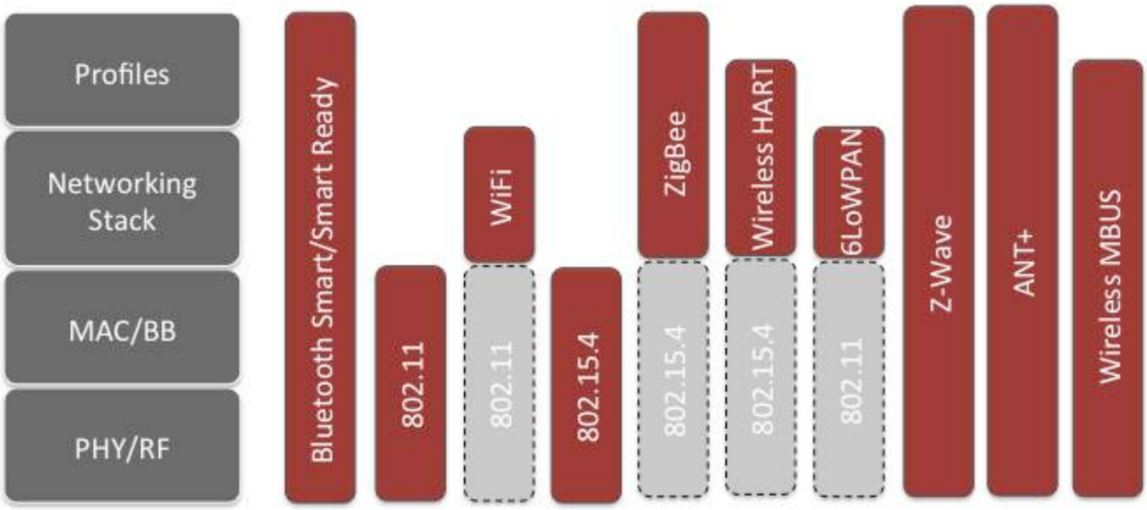
- Devices and objects can become monitoring sensors

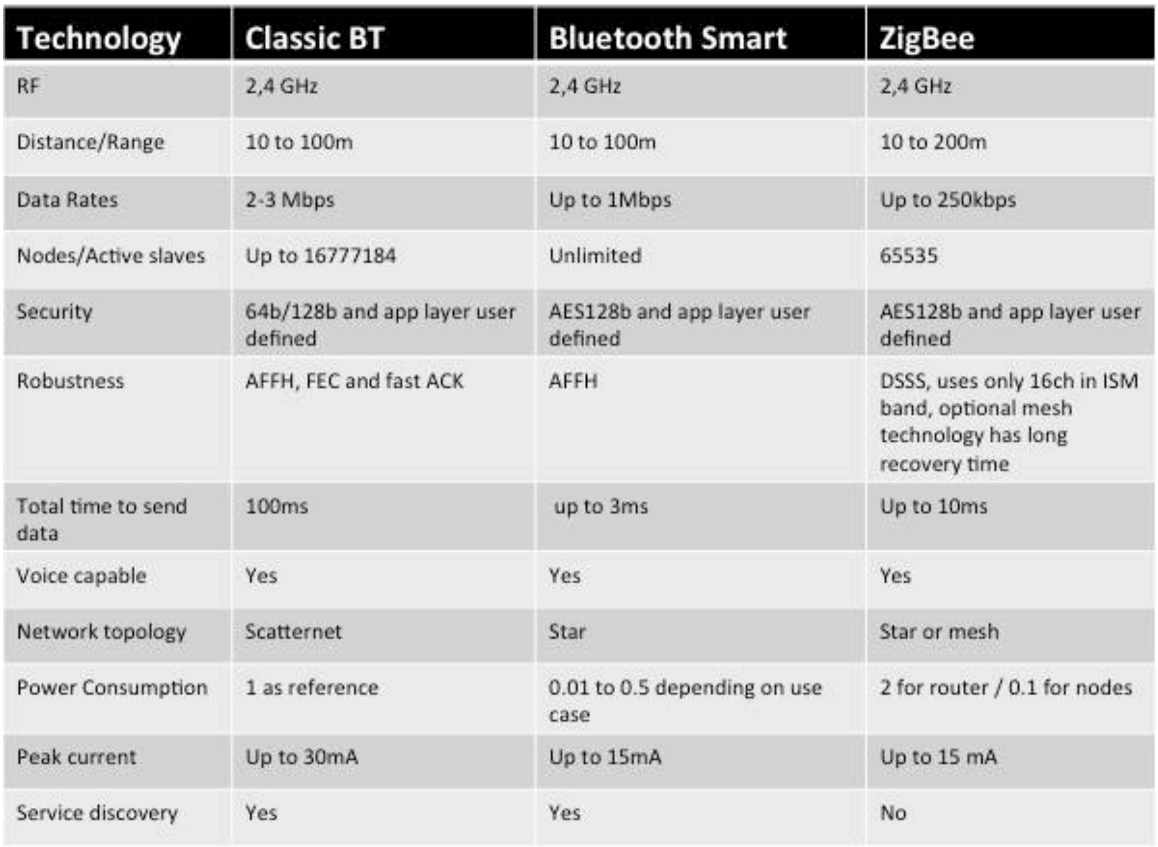


The simplicity of GATT servers makes easy device representation on the Internet. There are still some problems to be solved in order to increase the range of discoverability of these devices, solved to an extent by other 'standards'. With a longer range of BLE, a single device can be the home control gateway.

Nevertheless the scope of this study is to present future work and possible development of BLE standard in order to cope with Internet of Everything.

Competitive landscape - ZigBee Alliance





Technology	Classic BT	Bluetooth Smart	ZigBee
RF	2,4 GHz	2,4 GHz	2,4 GHz
Distance/Range	10 to 100m	10 to 100m	10 to 200m
Data Rates	2-3 Mbps	Up to 1Mbps	Up to 250kbps
Nodes/Active slaves	Up to 16777184	Unlimited	65535
Security	64b/128b and app layer user defined	AES128b and app layer user defined	AES128b and app layer user defined
Robustness	AFFH, FEC and fast ACK	AFFH	DSSS, uses only 16ch in ISM band, optional mesh technology has long recovery time
Total time to send data	100ms	up to 3ms	Up to 10ms
Voice capable	Yes	Yes	Yes
Network topology	Scatternet	Star	Star or mesh
Power Consumption	1 as reference	0.01 to 0.5 depending on use case	2 for router / 0.1 for nodes
Peak current	Up to 30mA	Up to 15mA	Up to 15 mA
Service discovery	Yes	Yes	No

The most significant competitor is ZigBee alliance. Other to be named: ANT, Z-Wave, Wireless HART, Wireless MBUS etc… Anyway, ZigBee is older and better established.

Some words on Z-wave

Z-Wave is a low-power wireless technology designed specifically for remote control applications.
The Z-Wave wireless protocol is optimized for low-latency communication of small data packets with data rates up to 100Kbps and operates in the sub-gigahertz frequency range, around 900 MHz.

- Bandwidth: 9.6 or 40 kbit/s, speeds are fully interoperable
- Modulation: GFSK
- Topology: mesh network
- Range: Approximately 30m assuming LoS, with reduced range indoors depending on building materials
- Frequency band: The Z-Wave Radio uses the 868.42 MHz SRD Band (Europe); the 900 MHz ISM band: 908.42 MHz (United States); 919.82 MHz (Hong Kong); 921.42 MHz (Australian/New Zealand).

In Europe, the 868 MHz band has a 1% duty cycle limitation, thus a Z-Wave unit is only allowed to transmit 1% of the time. Z-Wave units can operate in power-save mode and only be active 0.1% of the time, thus reducing power consumption substantially.

Some words on ANT+

ANT is a low power proprietary wireless technology, which operates in the 2.4GHz spectrum. A sensor company Dynastream established it in 2004.
It requires a special transceiver. Its primary goal is to allow sports and fitness sensors to communicate with a display unit, for example a watch or cycle computer. It also typically operates from a coin cell.
ANT+ has taken the ANT protocol and made the devices interoperable in a managed network, thereby guaranteeing all ANT+ branded devices work seamlessly. Similar to BLE, ANT devices may operate for years on a coin cell.
ANT devices are not subject to the extensive conformance and interoperability testing applied to other standardized technologies.

Never than less, the two above technologies are not big competitors for BLE as is ZigBee

How does ZigBee work?

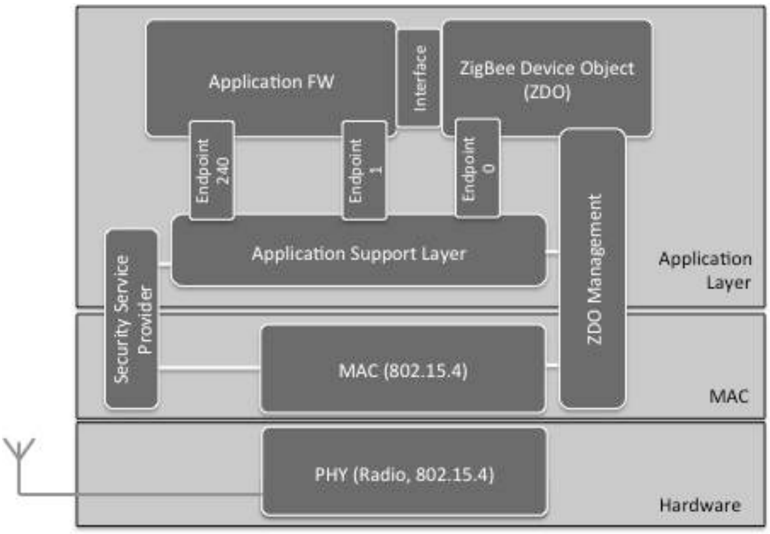
ZigBee is a specification for a suite of high-level communication protocols using small, low-power digital radios, which have low data rates, consume very and low

power. With ZigBee technology, interoperability will be enabled in multi-purpose, self-organizing mesh networks.
ZigBee is standard for embedded application software. The bandwidth of Bluetooth Low Energy is 1 Mbps and ZigBee has one- fourth of this value. ZigBee was meant to cope with sensors and remote controls and other battery driven devices.

For such wireless applications, a new standard called IEEE 802.15.4 has been developed . The new standard is also called ZigBee. The name ZigBee is said to come from the domestic honeybee that uses a zig-zag type of dance to communicate important information to other hive members.

ZigBee has a defined rate of 250 Kbit/s best suited for periodic or irregular data or a single signal transmission from a sensor or input device .Due to its low cost they are used wireless control and monitoring applications.

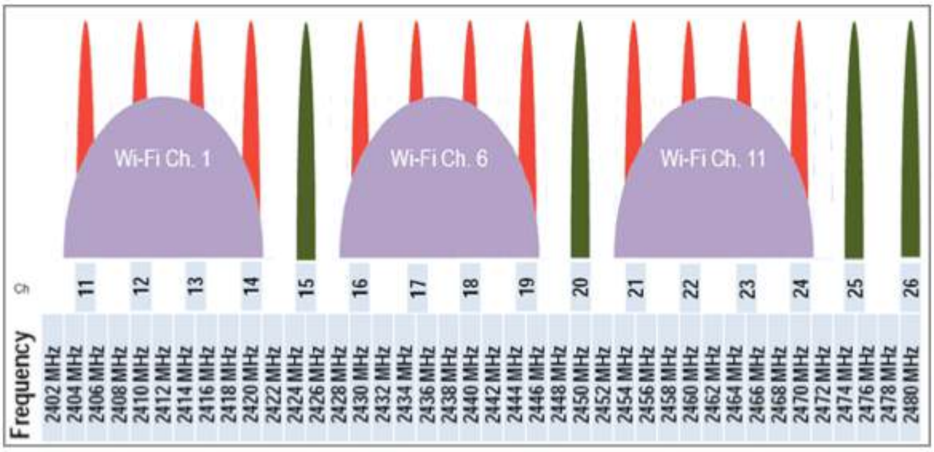
ZigBee Stack



The ZigBee stack architecture is based on the IEEE 802.15.4-2003 standard, which defines the two lower layers: the physical (PHY) layer and the medium access control (MAC) sub-layer.
Using this radio doesn't even give ZigBee much benefit. ZigBee doesn't control the radio design.
The only way to really build a low power, low energy system is to build it form the bottom all the way up to the top.
The ZigBee Alliance builds on this foundation by providing the network layer and the framework for the application layer. The application layer framework consists of the application support sub-layer (APS) and the ZigBee device objects (ZDO). Manufacturer- defined application objects use the framework and share APS and security services with the ZDO.
IEEE 802.15.4-2003 has two PHY layers that operate in two separate frequency ranges: 868/915 MHz and 2.4 GHz. The lower frequency PHY layer covers both the

868 MHz European band and the 915 MHz band, used in countries such as the United States and Australia. The higher frequency PHY layer is used virtually worldwide. The IEEE 802.15.4-2003 MAC sub-layer controls access to the radio channel using a CSMA-CA mechanism. Its responsibilities may also include transmitting beacon frames, synchronization, and providing a reliable transmission mechanism

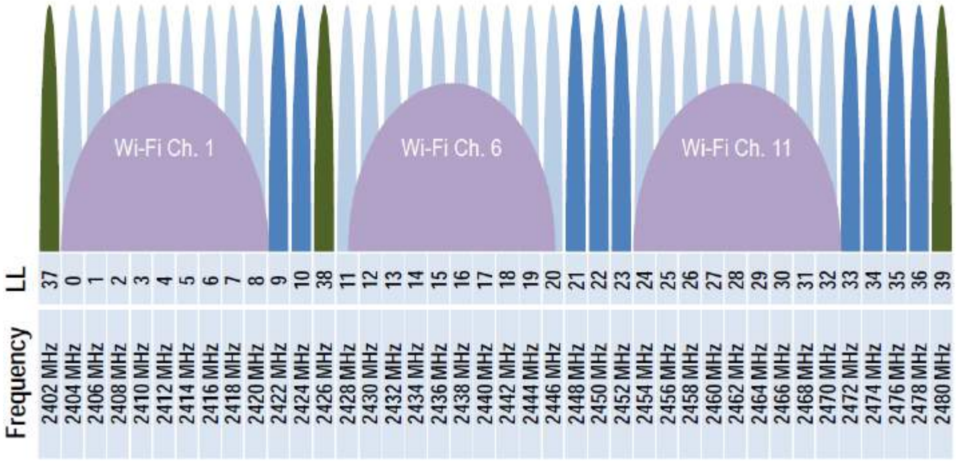
ZigBee Channel Allocation and Problems Occuring



ZigBee Channel Allocations

Each channel is 2MHz wide, but the spacing and placement of ZigBee channels implies that only 4 are likely to be free in the presence of average Wi-Fi network settings. Typically, channels 1, 6 and 11 are defaults. With an on-air signaling data rate of only 250kbps and the inability to implement hopping, ZigBee is at high risk of non-delivery of its packets. BLE makes much more efficient use of the spectrum and employs adaptive frequency hopping as proven by Bluetooth.

…compared to BLE

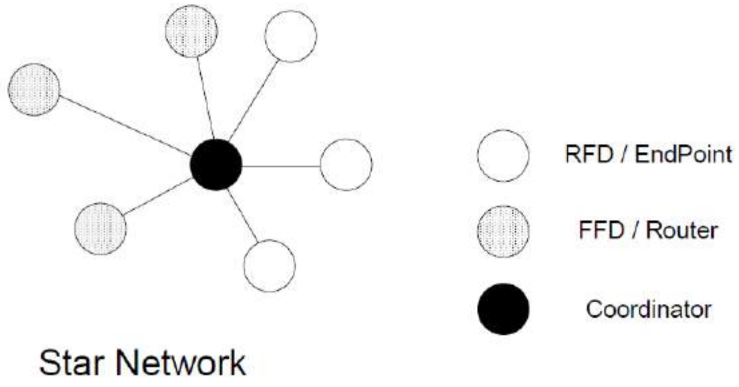


ZigBee doesn't implement a coexistence scheme, but does have the ability to continuously listen for clear time on its channel.
If the channel is heavily used then ZigBee throughput and latency are affected, eventually halting .
ZigBee PRO has a feature known as frequency agility (not the same as hopping) where it may be possible to search for a clear channel (of the 16 channels defined) and then re-establish the network. The frequency agility function makes using these extra channels easier. When a network is first formed the node seeks a channel with the least noise or traffic. If over time extra traffic appears or noise becomes present the host application can scan for a better channel and move the whole network to the new channel allowing the network to adapt over time to changing RF environments.
Placing a ZigBee node in close proximity to a wide band (Wi-Fi) device causes severe problems to the ZigBee network…
For avoiding this a different standard has been pushed on the market: RF4CE.
RF4CE only uses channels 15, 20 and 25 for avoiding WiFi disturbances…
In real world testing ZigBee(RF4CE) is actually slightly more robust than BLE because BLE is connection based and need to reconnect when it loose the connection. RF4CE retransmits if it does not receive an ACK. Disturbance normally gives longer latency, but the data comes through.
But at the end of these assumptions ZigBee is not ONE standard, it is a multitude of corrections and add-ons (PRO, RF4CE, HART, etc…) to catch up with Bluetooth Low Energy

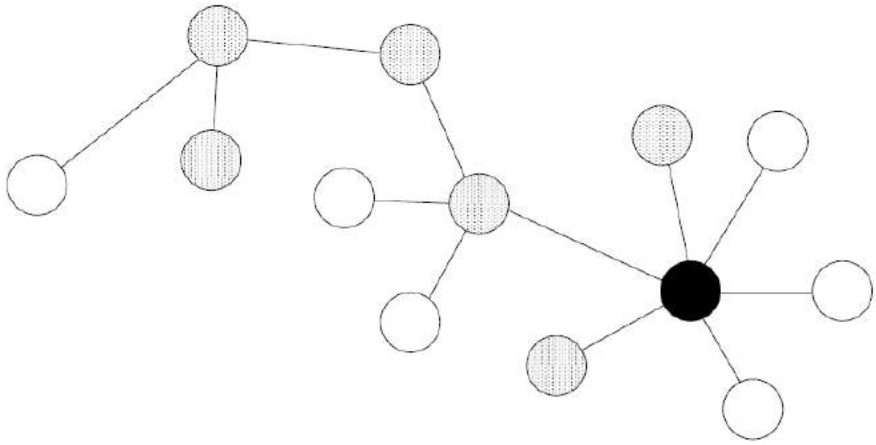
ZigBee Topologies

…or where ZigBee advantage is for the moment

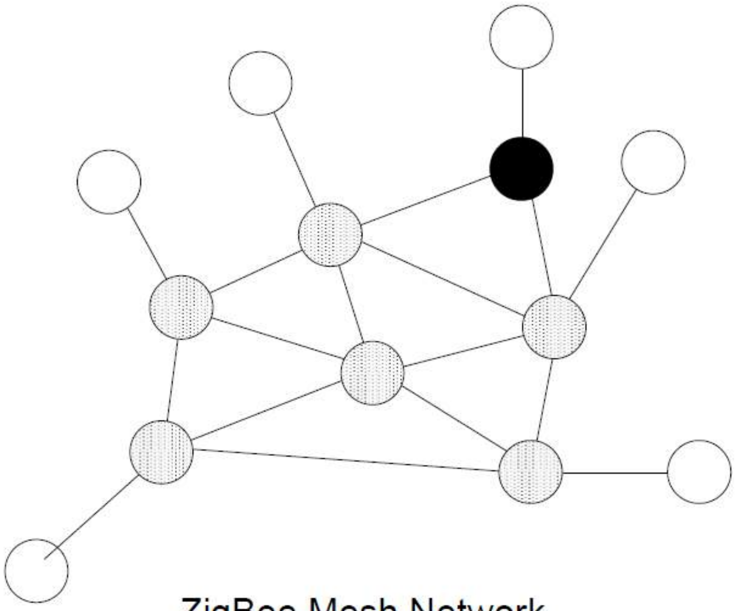
ZigBee basic topology is same as Bluetooth a Star Topology



… also it adds a tree network. Basically this one can be also done with BLE



… and where the main advantage is – Mesh Network



ZigBee Mesh Network

Bluetooth Low Energy uses a star topology; so one master can only talk to slaves that are in range at a given moment.
Is the star topology a weakness compared to ZigBee's mesh? Actually it needs more knowledge and cannot be easy to implement as mesh topology.
In fact, using a mesh is one of ZigBee's mistakes - low power and mesh topology at the same time are not a good match. Any mesh relay node can't go to sleep to save power, as it has to stay ready to hand on data it receives. The only way for nodes to relay info, is to have them on all the time.
The best way to mesh devices is hierarchy, so masters are linked over appropriate technology.
One technology to do it all is actually the wrong design!

Bluetooth - ZigBee Comparison

Comparison	Bluetooth Low Energy	ZigBee
Technical	Newer Even lower power Simple GATT stack	Do not own the PHY Low power Stack is quite light WiFi coexistence an issue Lower bandwidth
Business	Focusing mainly on mobile devices: phones, laptops, tablets, etc now getting a way in M2M	Older, gone already through some iterations Too many standards and implementation
Future	Must implement 6LoWPAN to catch ZigBee Replace classic BT with dual mode devices will boost presence All major OS:s implemented BLE APIs **GATEWAY profile as new way of hierarchical governance for avoiding mesh problems**	Already implemented 6LoWPAN Already present in some markets

Bluetooth Low Energy and Internet of Everything

As in the future work row in the table presented above, two major upgrades are needed to Bluetooth Low Energy standard for being able to cope with Internet of Everything:

- 6LoWPAN for being able giving devices an identity
- Increase range to catch up with mesh networking or other technologies based on 802.15.4

The following chapter will focus on possible implementation idea of a new open profile (standard) and open software stack to comply with main improvements presented above

As presented in chapters above Bluetooth Smart and Smart Ready concepts are already introduced in the market and are implemented as Open APIs in major operating systems, such as: Android, iOS, Blackberry, Microsoft, Bluetooth as standard getting even more traction from developers.

6LoWPAN over BLE

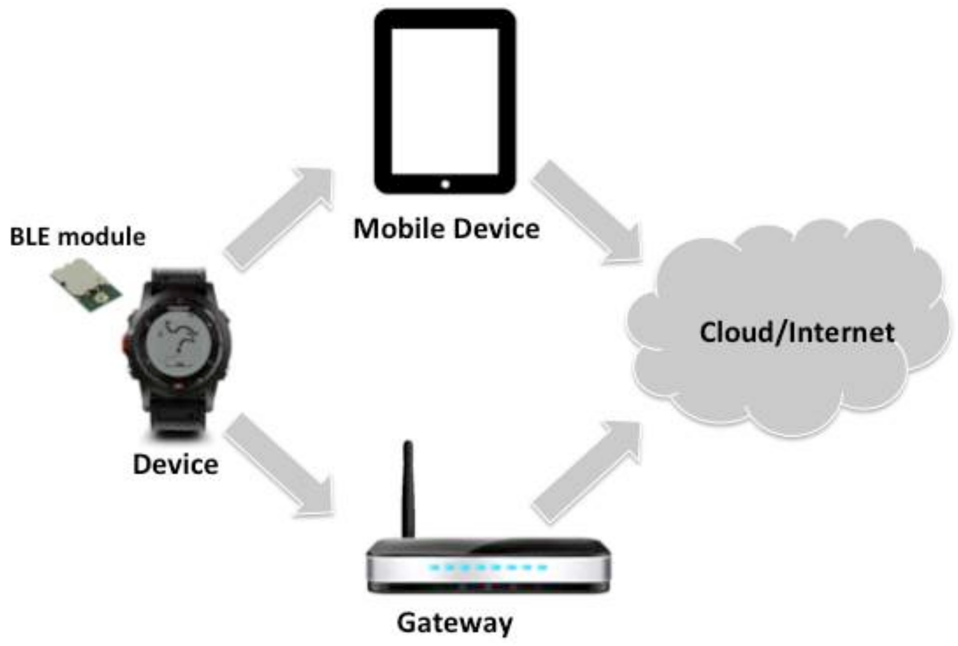
6LoWPAN is a standard today for sending TCP/IP data over constrained wireless networks to resource constrained devices. It was originally developed over 802.15.4 network but there are proposal for a variant that is used on top of Bluetooth Low Energy as a Special Working Group under Bluetooth SIG, Nokia being one of the main contributors.

It is currently possible to connect BLE sensors with the Internet using protocol translation in the mobile device acting as a gateway; however, solutions are application and operating system specific. The current solutions do not scale and do not enable open web services creation environment for developers

The most flexible approach would be to use IP for end-to- end communication between the sensors and a server. IPv6 would be the ideal protocol due to the large address space it provides.

Key components of the solution include adapting 6LoWPAN for BLE:

- Differences in the header compression and fragmentation functionality
- BLE operates in a star topology, thus source and/or destination IPv6 addresses can be elided in many cases based on known context
- Fragmentation will be performed in the link layer, not in the network layer
- Configuration, application protocol efficiency and security, context awareness as well as gateway operation



IPv6 over BLE IETF draft Working Group Last Call completed, moving the draft to IESG approval queue. Sensor Internet protocol FRD approved in BT-SIG BARB – Goal is to have BT-SIG stamp on the solution, and a fixed channel ID reserved for IP traffic

See http://en.wikipedia.org/wiki/6LoWPAN for an overview.
See http://datatracker.ietf.org/wg/6lowpan/charter/ for more detailed information about 6LoWPAN. Shows the IETF working group status all RFC and other documents.

View on IoT protocols

Internet of Things, the latest wave of the Internet, is about connecting physical objects in ways that help us analyze and control our environment to provide better safety, comfort, and efficiency.

Amid this move toward IP, IoT requires a messaging stack. Here is a flavor of existing possibilities

- CoAP (Constrained Application Protocol) over UDP is used for resource constrained, low-power sensors and devices connected via small BW networks, especially supporting a high number of sensors and devices within the network. CoAP has already found success as a key enabling technology for electric utility
- XMPP (Extensible Messaging and Presence Protocol) has its roots in instant messaging and is a contender for mass scale management of consumer white goods, such as washers, dryers, refrigerators, and so on. But because it assumes a persistent TCP connection and lacks an efficient binary

encoding, it's typically not been practical over LLNs (Low-power and Lossy Networks). But the recent work of XEP-0322, XEP-323, and XEP-324 aim to make XMPP suited for IoT.

- RESTful HTTP over TCP is particularly attractive for connecting consumer premise devices, given the near universal availability of HTTP stacks for various platforms. The RESTful HTTP approach has found success in smaller scale LLNs requiring message latencies of several seconds (home energy management, etc.).
- MQTT is a publish/ subscribe messaging transport protocol optimized to connect physical world devices and events with enterprise servers and other consumers. It is designed to overcome the challenges of connecting the rapidly expanding physical world of sensors, actuators, phones, and tablets with established software processing technologies. MQTT has been used in sensors communicating to a broker via satellite links, over occasional dial-up connections with healthcare providers (medical devices), and in a range of home automation and small device scenarios. MQTT is well suited for mobile applications because of its small size, minimized data packets, and efficient distribution of information to one or many receivers.

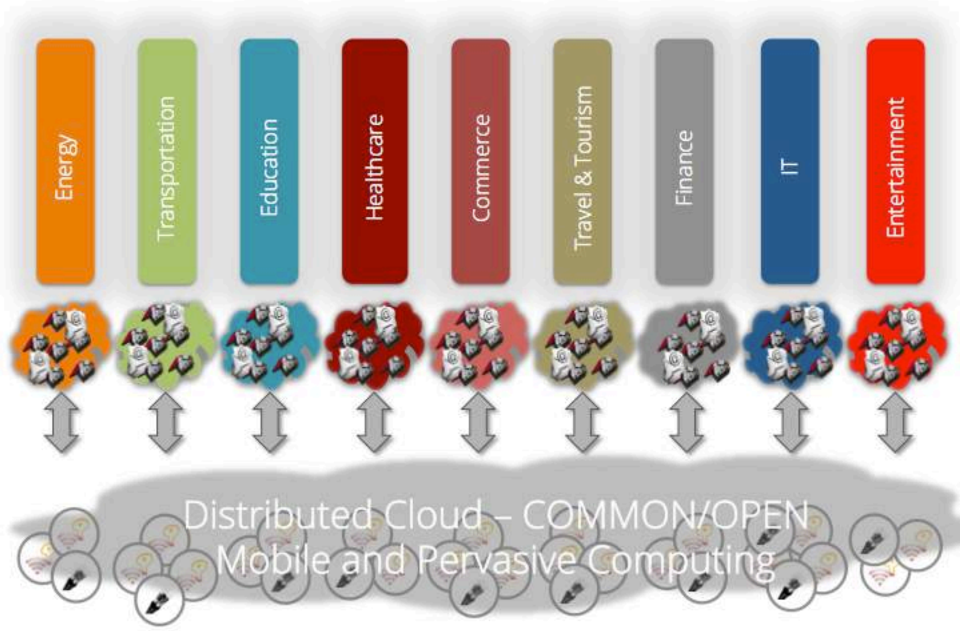
The table below explains how we view the IoT protocol landscape.

Protocol	CoAP	XMPP	RESTful	MQTT
Transport	UDP	TCP	TCP	TCP
Messaging	Request/Response	Publish/Subscribe Request/Response	Request/Response	Publish/Subscribe Request/Response
Cellular Network suitability (1000s nodes)	Excellent	Excellent	Excellent	Excellent
LLN Suitability (1000s nodes)	Excellent	Fair	Fair	Fair
Compute resources	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash
Success Stories	Utility Field Area Networks	Remote management of consumer white goods	Smart Energy Profile 2 - premise energy management/home services	Extending enterprise messaging into IoT applications

Which one will succeed? Hard to say! Each of them has great adopters and supporting groups. It will be interesting for the continuation of Minimum Viable Device project to test all of them and identify pro:s and con:s from a 3rd party perspective without implication in any of the above supporting groups and propose the best solution.

Gateway Profile - the Need

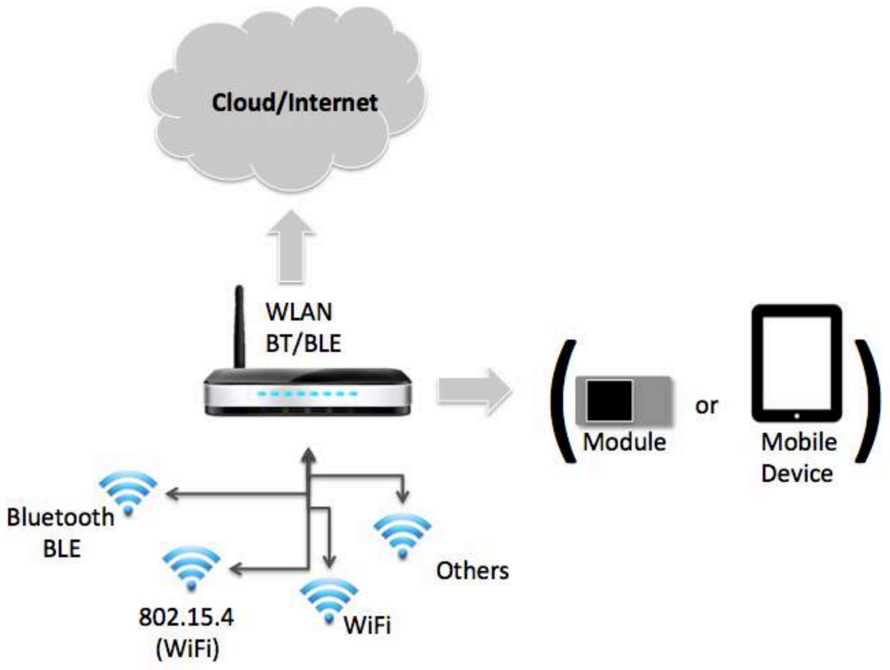
Pervasive computing vs. embedded computing? Which one will be the key of Internet of Things? Do we have now Internet of Things? If yes, where?
What we have today is Internet of silos. We have an Internet of Bluetooth, Internet of ZigBee, Internet of ABB solution and so on…
If this will continue, will be almost impossible to get 50 billion connected devices. We need something to unify myriads of technologies, operating systems, Internet protocols and free tools in order to get it there.



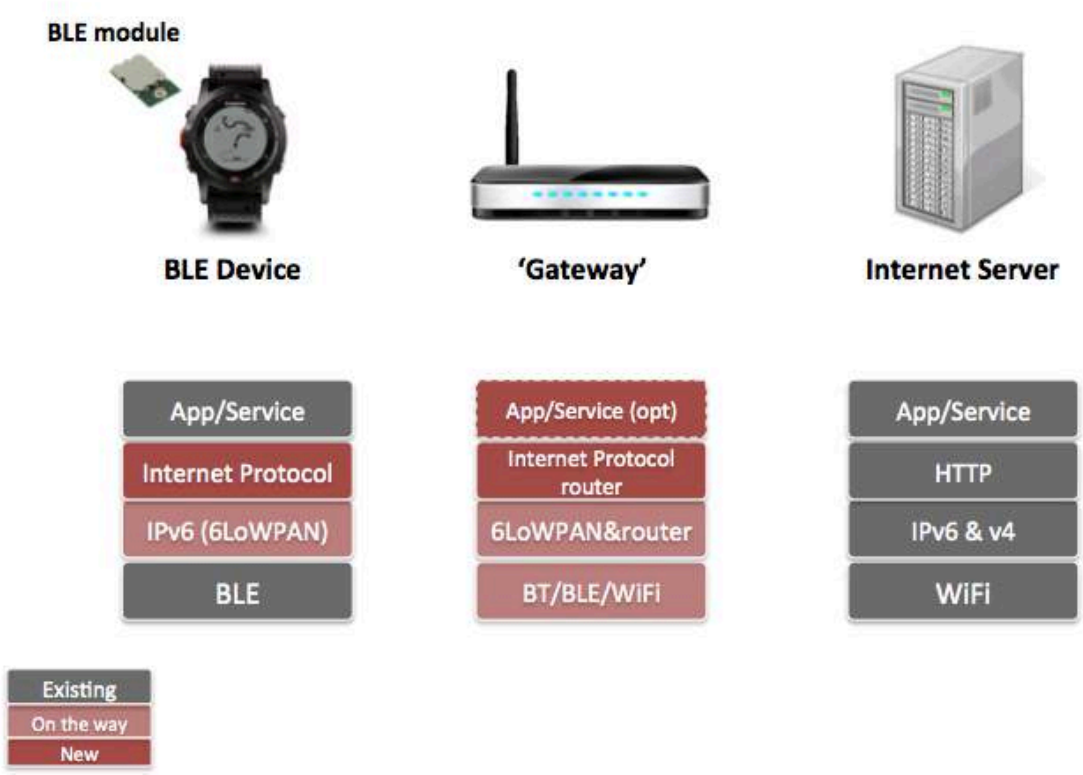
In many cases, for being able to 'talk' to different industries, it is a need for gateways to send data from sensors or smart devices to appropriate industry or services to make use of this data.
To do that, gateways must be open architecture both hardware and software!

Our proposal for such a gateway will be a scalable architecture supporting multiradio protocols: WiFi and BT/BLE and an MCU, scalable from low cost low power to more advanced smartphones.

Since the upstream connection is based on Internet protocols the IP protocol contains all the necessary mechanisms to support routing of the traffic to the services in the cloud and in some cases also between the local Bluetooth Low Energy devices, when they are presented in the area.



Gateway Profile - Implementation Proposal



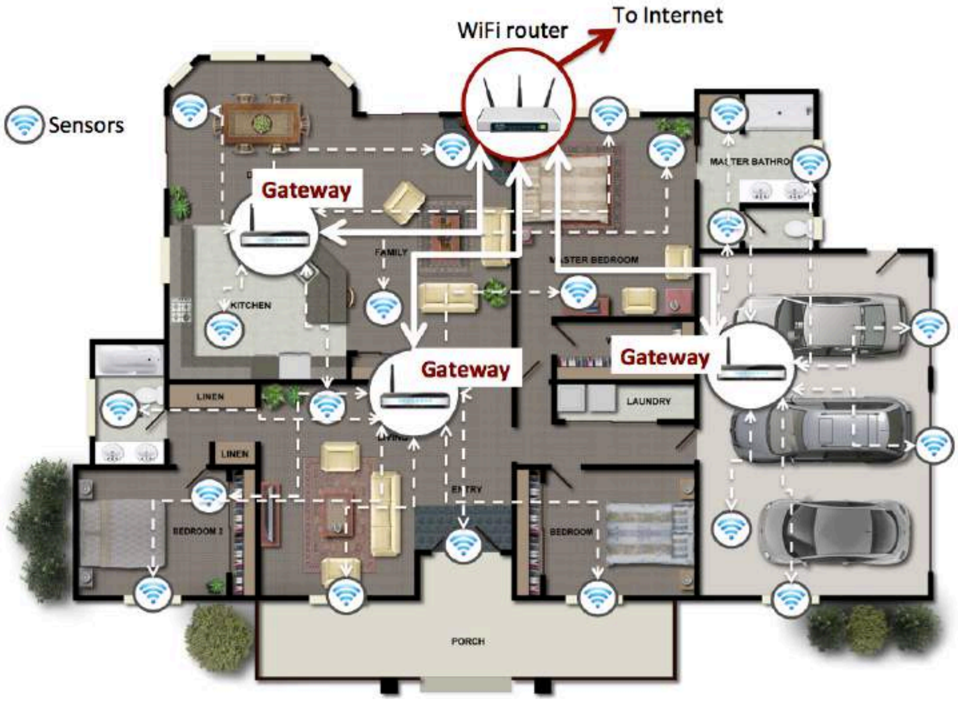
Our proposal comes into anticipation that Internet use cases for sensors are going to bloom, we want to prepare technical enablers for them using BLE and other radios. As described before there is no specification today on implementation of IP/IPv6 over BLE.
The 6LoWPAN standard provides useful generic functionality like header compression, link-local IPv6 addresses, Neighbour Discovery and stateless IP-address auto configuration but cannot be applied to BLE as it is today.
Considering application protocols, IPv6 can in principle support any protocol. BLE technology, however, sets limitations to protocol overhead such as header sizes. CoAP, RESTful, MQTT are Internet protocols specifically designed for resource constrained environments. They could be run on top of IPv6 supporting response/requests from the Internet server.

The figure above represents a possible representation of future architecture of IoT devices and gateways. The red parts are to be developed in an open source, open architecture environment.

Gateway Profile - Proposed use cases as proof of concept

Gateway as range extender - Smart House

A problem that sometimes will occur when using Bluetooth Low Energy for IoT applications is the limited range (as Bluetooth Low Energy is using a star topology). Competing technologies using the 2.4 GHz ISM band (e.g. 802.15.4 based technologies) does often support the concept of meshing and routers to extend the coverage and that is currently not possible in Bluetooth Low Energy.



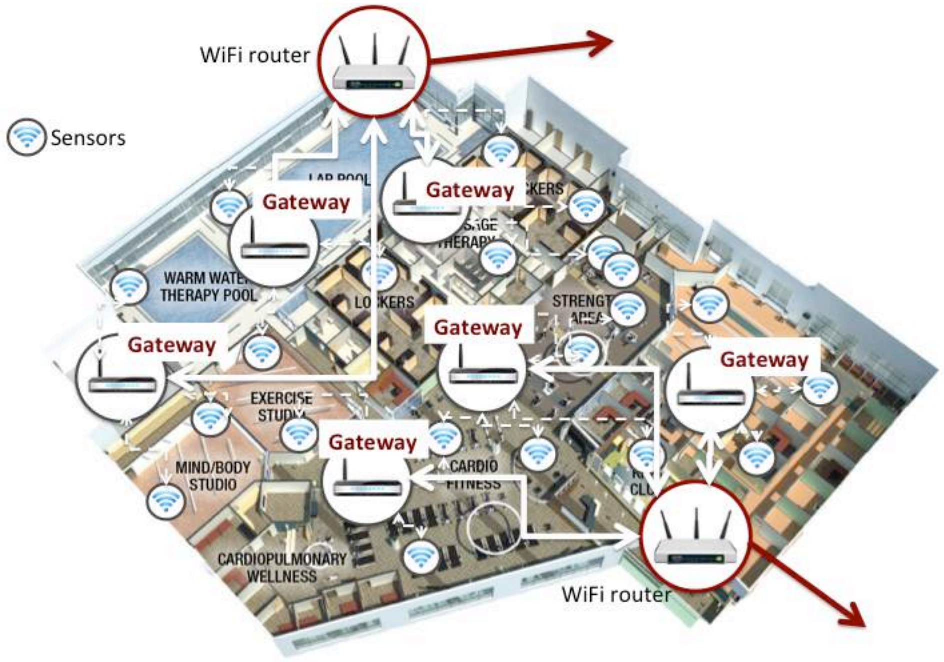
The figure above shows a possible solution to extend coverage using interconnected gateways. The upstream link is using WiFi towards the WiFi router to connect to Internet and the sensor to gateway link is over BLE.
Since the upstream connection in all the examples above is based on Internet protocols the IP protocol contains all the necessary mechanisms to support routing of the traffic to the services in the cloud and in some cases also between the local Bluetooth Low Energy devices.

In this scenario the ‘Gateway’ can be a low-cost, low-power processor (M3 or M4 core) in combination with a multi-radio circuit (a radio chip with built-in support for: Bluetooth, Bluetooth Low Energy and WiFi). The resulting gateway will be cost-efficient, low power, thou not battery driven and very small in physical size.

Gateway and BLE sensors for upgrading old Hospitals and elderly houses to - Smart Hospital or Smart Elderly Houses

Today is almost impossible to connect sensors within a big hospital, from multiple reasons:

- They were not built for coping with sensors, so the wired infrastructure is not present
- The costs for coupling via wires are huge
- Multiple sensors and measurement instruments are popping up at huge speed and it is a need for big hospitals to jump on the latest trend due to cost efficiency
- A present WLAN infrastructure already exist but partially used at its normal capacity



The solution above shows a possible solution to improve coverage and use of multiple sensors in medical world, identification sensors, indoor positioning, etc. using interconnected gateways.

The upstream link is using WiFi towards the WiFi router; already presented in the hospital infrastructures to connect to Internet and the sensors to gateway link over BLE. Some sensors might use RFID (for example for identification and BLE) for sending identification parameters. For example each bed will have an RFID tag and the patient an BLE+RFID reader. The reader reads the tag and sends its information over BLE to gateway and forward to via WLAN hospital IT software for localization.

Since the upstream connection in all the examples above is based on Internet protocols the IP protocol contains all the necessary mechanisms to support routing of the traffic to the services in the cloud and in some cases also between the local Bluetooth Low Energy devices.

Also, in this scenario the ‘Gateway’ can be a low-cost, low-power processor (M3 or M4 core) in combination with a multi-radio circuit (a radio chip with built-in support for: Bluetooth, Bluetooth Low Energy and WiFi). The resulting gateway will be cost-efficient, low power, thou not battery driven and very small in physical size. The costs for adapting old hospitals to the newest technology will be extremely low compared to build in new-wired infrastructure. Security is not a problem due to IPv6 security all over the network from sensor to IT infrastructure.

Conclusion

￼ The Internet of Things, despite the silo approach today, promises many benefits for different domains and industries. The businesses, end users and even the whole society will be positive impacted by the potential changes resulting of applications of this concept. A successful implementation of a network of intelligent, interactive and autonomous things would probably change the daily life of each individual.

Bluetooth low energy technology meets all the requirements of a wireless solution for sensors and actuators: low cost, reliable, huge lifetime, easy to use, secured. Also from comparison with other technologies it shows major advantages, mainly due to huge ecosystem already existing and continuously growing.

However, this paper also presented several challenges and problems that have to be solved before a working Internet of Things will be possible. Many of the key technologies, struggled to solve alone different aspects to be applicable and scalable in a network of the expected size and kind of the IoT.

The paper shows present a possible solution for the future development of IoT and creation of wireless sensor networks, using a proposed gateway profile, that will couple existing solutions, such as classic Bluetooth, Bluetooth Low Energy and WiFi and creating an open, de facto, standard and an open ecosystem. For its realization, some parts of architecture requires deeper study: a software firmware for being able to ‘convert’ proprietary solution (ABB, Schneider Electric, Honeywell, etc.) to a proposed open standard, chose an Internet protocol (CoAP, RESTful, XMPP, MQTT) as best solution for a defacto standard, creating a consortium for getting technology price even cheaper.

This solution will give possibility to enhance and adapt actual, houses, schools, hospitals, etc. to modern technology with low cost and easy deployment of new modern sensor, highly scalable and taking into consideration the existing ecosystem, both hardware wise (mobile phones, laptops, tablets, TV, routers, etc…), mixed open environments such as: Arduino, Raspberry Pi and software wise (Open software SDK over BLE and WiFi). This will create an easy to use ‘database of things’ helping new startups and SME to create new business models, jobs and create new services.

Mobile Heights is going to support and drive this initiative to next level, by a B level application with defined focus areas such as: m-health, logistics, transportation and test new areas still closed today, such as: energy, tourism, education, etc.

While ICT community looks sceptical to the fact that wireless ecosystems such as: Android, iOS, Microsoft, Blackberry, etc are NEVER going to be one , and we agree with this statement, we consider that the unification will happen via hardware with the two technologies identified by this study: Bluetooth Smart and WiFi and low level software, while applications and services will be built on top supporting each of the named ecosystems. Even today, all of the named ecosystems have native support!

We are looking forward, building actual hardware and software for the identified gateway, opening up sw layers as much as possible, at least with an Open SDK and Open API:s

Appendix 1

[1] Comparisons between low power technologies – White paper CSR

[2] internet of Everything - CISCO Whitepaper

[3] - Blouetooth Low Energy Technology Makes New Applications POssible - ConnectBlue – whitepaper

[4] How Low Energy is Bluetooth Low Energy? Comparative measurements with ZigBee/802.15.4