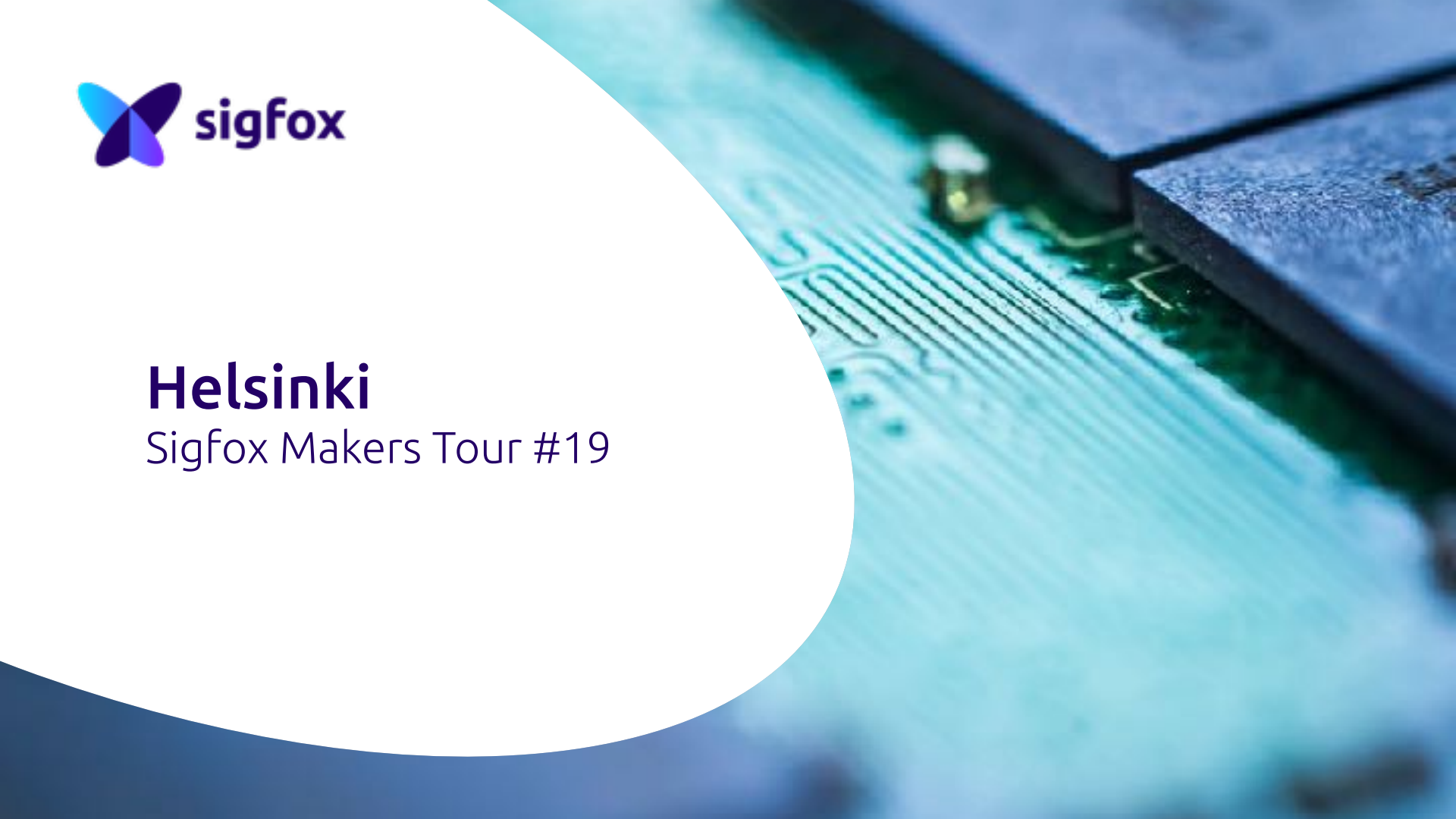


Helsinki

Sigfox Makers Tour #19

sigfox

Introduction

What’s Sigfox all about?

Schedule

Intro

Presentation

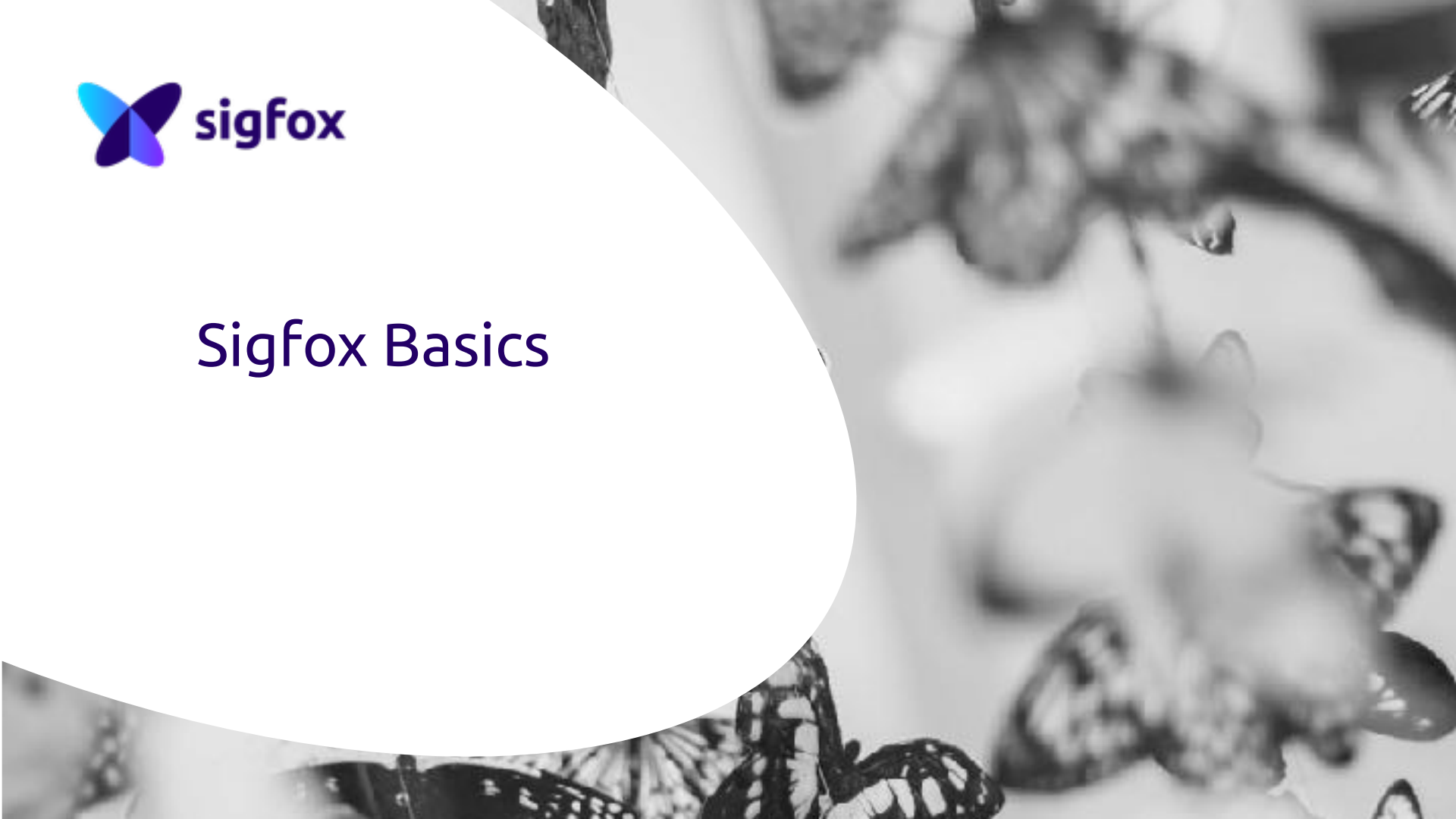
Demo

Workshop

Enjoy!

sigfox

Sigfox Basics



About Sigfox

Sigfox does **not selling chips**

Sigfox does **not building connected solutions**

Sigfox invented a **radio protocol**

Sigfox operates a **global network**

sigfox

Why Sigfox ?

Need for a solution dedicated to the IoT and not an existing one tweaked for it.

We only serve the IoT, that’s the reason why we’re doing it efficiently

Low **P**ower **W**ide **A**rea **N**etwork

From far away…

With very little energy!

sigfox

sigfox

FIRE STARTING

MY BATTERY IS LOW

I'M FULL

I'M LOST

I AM AT
THE WAREHOUSE

I NEED
TO BE REPLACED

I NEED
REPAIR

PRESS
TO ORDER

sigfox

New Possibilities

Existing solutions: Cheaper connectivity & extended battery life

Enables new IoT applications

Backup connectivity for higher bandwidth devices

How to Communicate

Decide on something to send

Power on the communication module

Send

Message is picked up by the network

Data is received on your server

sigfox

How Hard ?

Send an AT command

You receive an HTTP Request on your application server

Core Concepts

Out of the Box

No connection

No configuration

No pairing

No signalling

sigfox

Energy Efficiency

The Sigfox protocol has been designed to maximise energy efficiency

Tx: <50 mA during a few seconds (25mW, 14dB)

Key factor: idle consumption (unconnected 99.x% of the time)

Idle consumption: a few µA

sigfox

Very Long Range

Best case scenario

+100km between transmitter & receiver (base station)

Real life

A few kms (city) to tens of kms (countryside),
depending on the topography

sigfox

Outdoor & Indoor

Good indoor propagation properties

Of course, you need to consider signal attenuation (~20dB)

Two-way Communication

Devices can receive updates sent from your application server

Communication is initiated by the device

sigfox

Small Messages

Useful payload: up to **12 bytes**

Up to **140 times each day**

100 bit/s

sigfox

Payload Examples

GPS coordinates (lat x lng) : 6 bytes

Temperature: 2 bytes

State reporting : 1 byte

Heartbeat, update request : 0 byte

And… who needs full bytes when 5 bits are enough ?

sigfox

Payload Examples

A (int): 17568 —> **0100010010100000**

B (0-32): 17 —> **010001**

C (state): 3 —> **10**

Frame: **01000100 10100000 01000110**

Frame: 0x44 0xA0 0x46

AT$SF=44A046

sigfox

More !

816b1954 | 10000001 01101011 00011001 01010100

10000001 01101011 00011001 01010100 Active mode (Temp)

10000001 01101011 00011001 01010100 Temp. MSB & LSB

10000001 01101011 00011001 01010100 Humidity x2

Temp : 0110011001 = 409 . (409-200) / 8 = 26.125°C

Humidity: 01010100 = 84. 84/2 = 42%

sigfox

Low Cost of Communication

Small subscription fees

Short software development cycle

Low cost hardware components

sigfox

Security

Security

Each message is **signed with a key unique** to the device

Messages can be **encrypted or scrambled**

No keys exchanged over the network, no handshake

Security is an **ever ongoing effort**

Message Signature

With each message, a hash is calculated & sent, using:

Device ID

Secret key, unique to the device. Never transmitted OTA

Payload

Internal increment

sigfox

Radio Properties

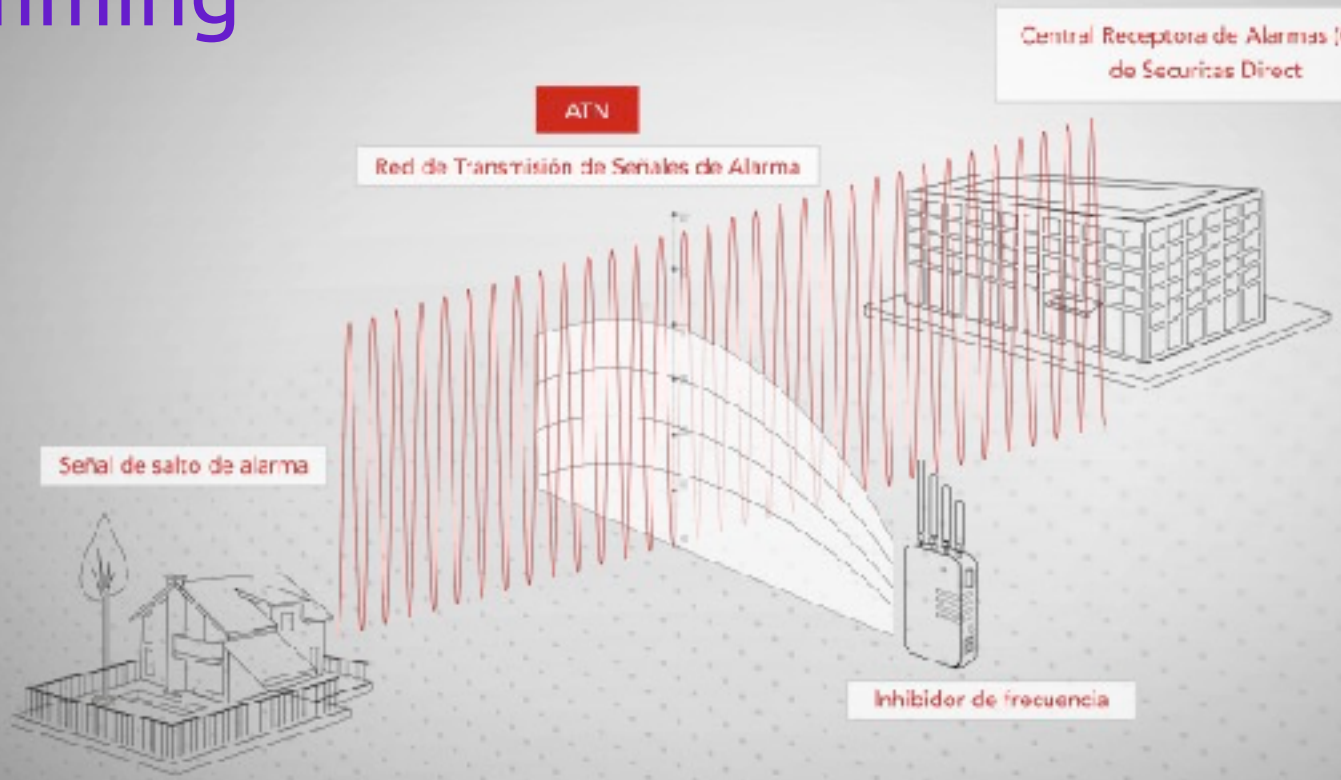
Great tolerance to interference

Jamming resistant

Interception is hard: UNB & frequency diversity

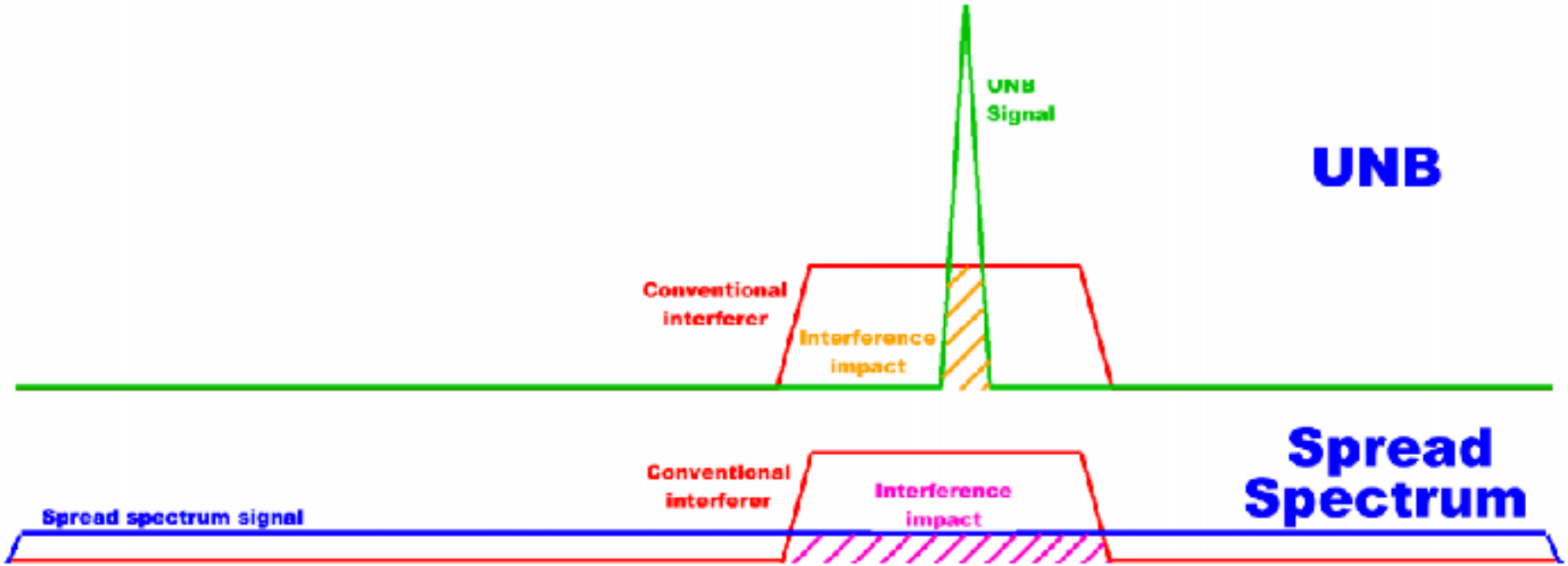
sigfox

Jamming



sigfox

Interference



sigfox

sigfox

Radio Spectrum

Ultra Narrow Band

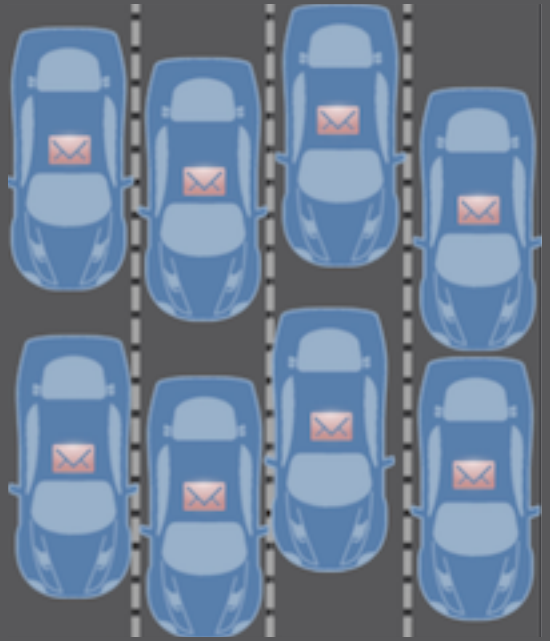
Sigfox use

The network currently monitors a 192KHz part of the spectrum

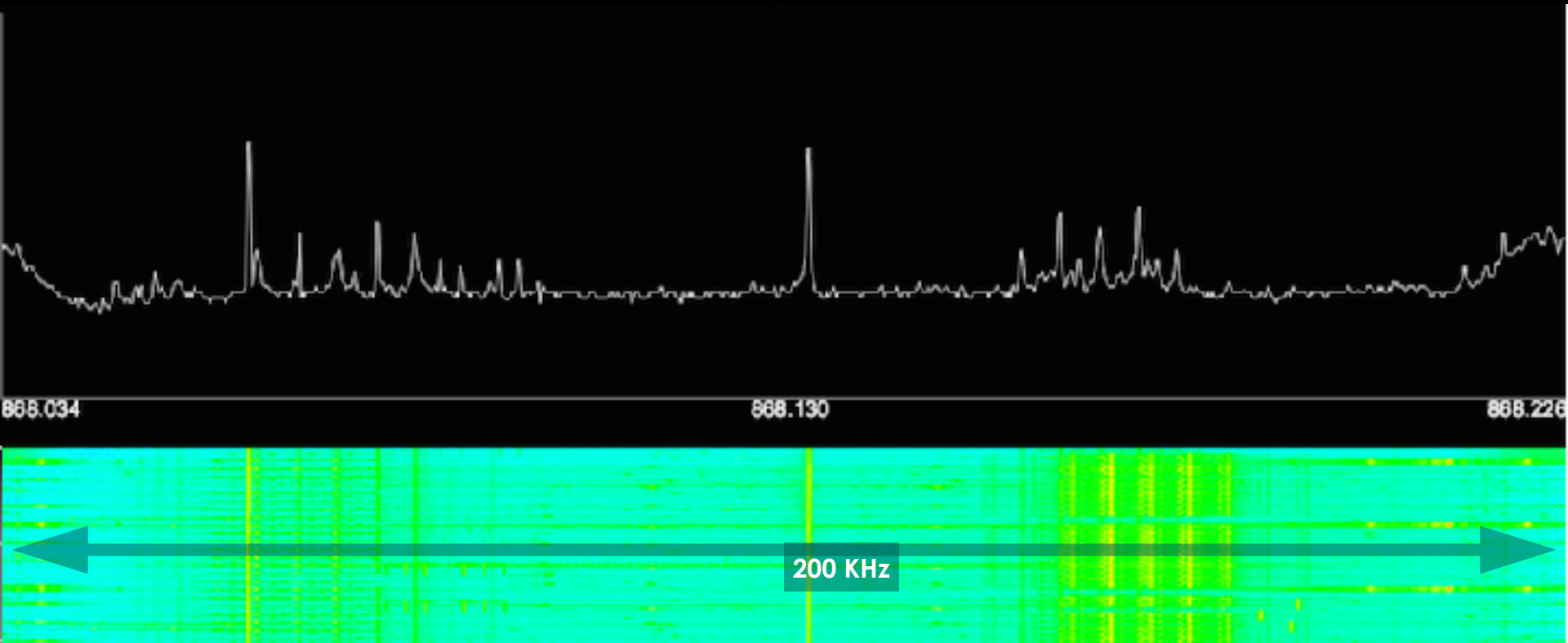
Each message is ~100Hz wide

Why Ultra Narrow Band?

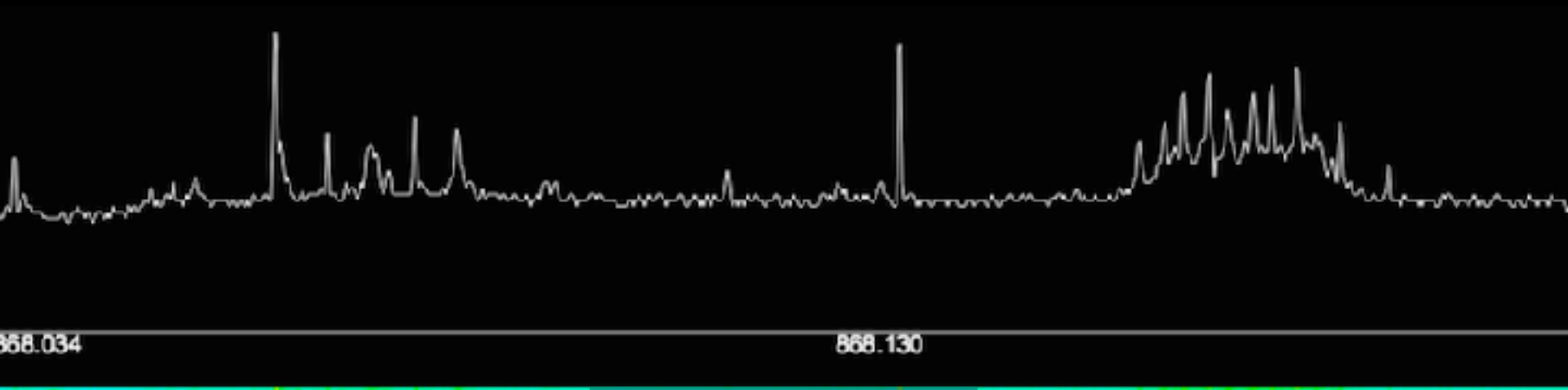
sigfox



Radio Spectrum



Messages



Each repetition is 100 hz wide

sigfox

Regulations

Regulations

Sigfox is operating on unlicensed Sub-GHz frequency bands all over the world

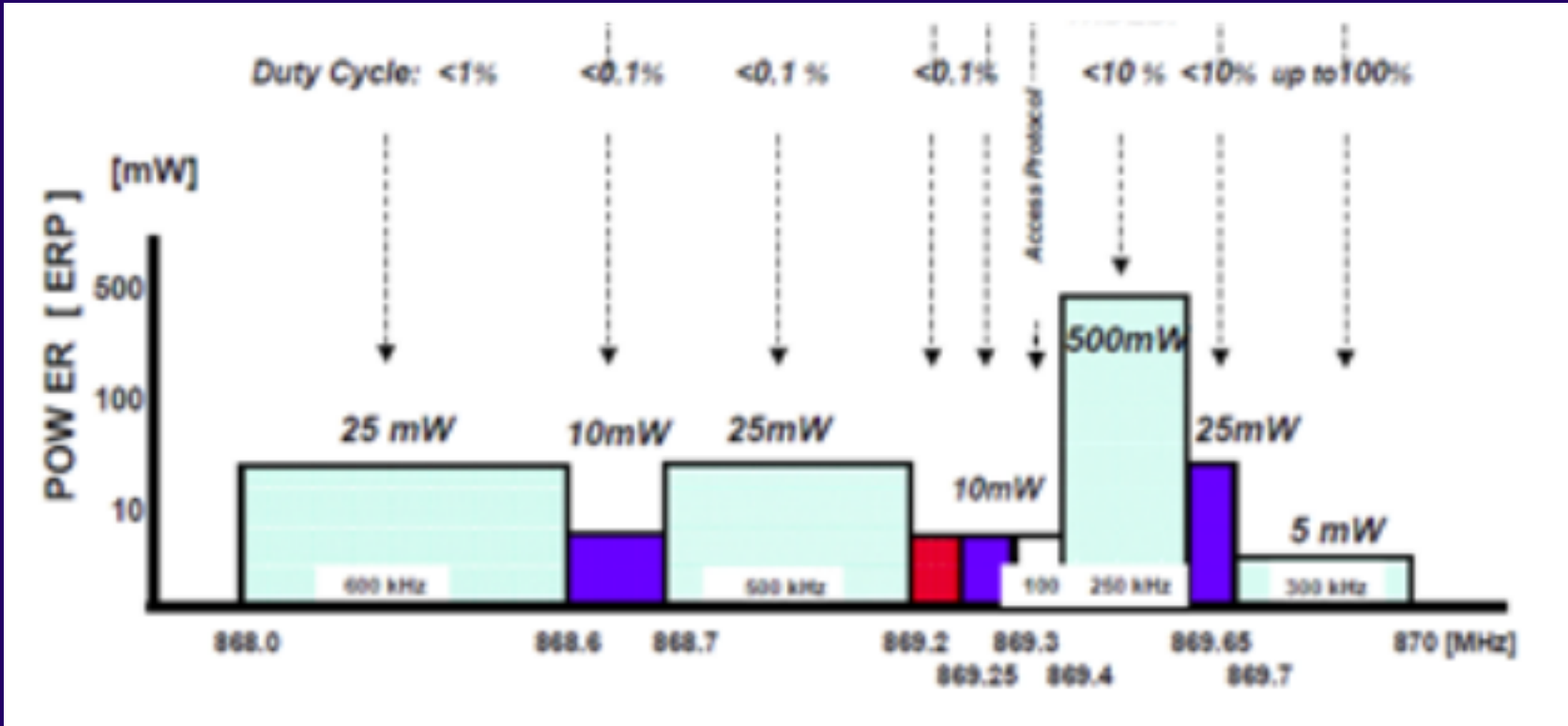
We just have to pick the right central frequency

Easy, right ?

UNITED STATES FREQUENCY ALLOCATIONS

THE RADIO SPECTRUM

ETSI Duty cycle



sigfox

Unlicensed Bands

Compliant with regulations

ETSI 300-220

FCC Part 15

ANATEL 506

AS/NZS 4268

sigfox

Different bands

Regional regulations affect

Central frequency

Power Output / Data Rate

Spectrum access

Handled by the Sigfox stack

Same hardware can be used, with software switches

sigfox

Coverage

Global network

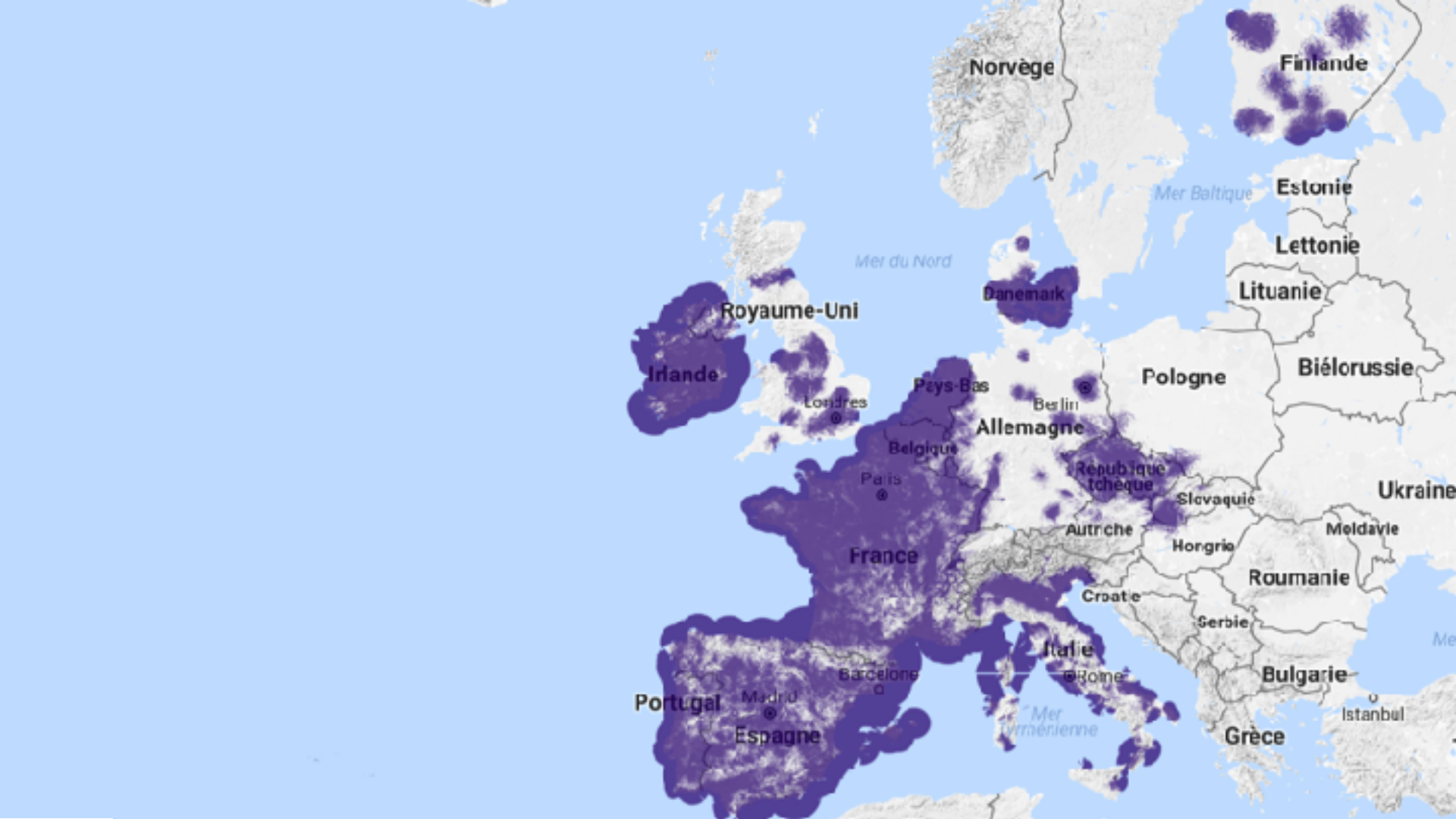
Sigfox is offering a global network, not a solution to build private networks

Roaming is included is the standard service

Devices will work the same all over the network

sigfox


Islande
Finlande
Suède
Norvège
Russie
Canada
Royaume-Uni
Pologne
Ukraine
Allemagne
France
Espagne
Italie
Kazakhstan
Mongolie
États-Unis
Turquie
Chine
Corée du Sud
Japon
Océan Atlantique Nord
Irak
Iran
Afghanistan
Pakistan
Algérie
Libye
Égypte
Arabie saoudite
Inde
Thaïlande
Mexique
Mali
Niger
Soudan
Tchad
Nigeria
Éthiopie
Venezuela
Colombie
Kenya
Indonésie
Papouasie-Nouvelle-Guinée
Tanzanie
Brésil
Pérou
Angola
Bolivie
Namibie
Madagascar
Océan Indien
Botswana
Australie
Chili
Océan Atlantique Sud
Afrique du Sud
Argentine
Nouvelle-Zélande

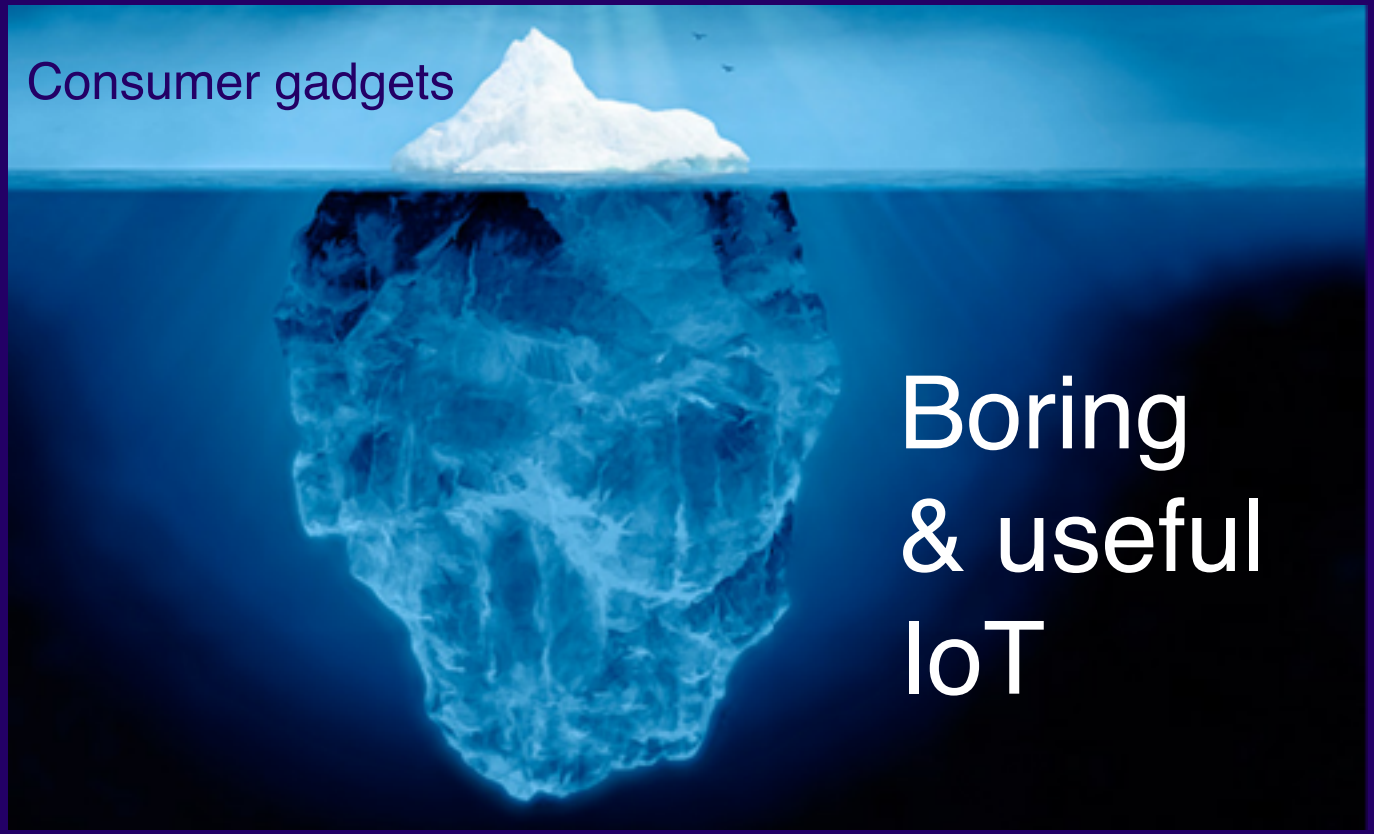

Norvège
Finlande
Mer Baltique
Estonie
Lettonie
Lituanie
Mer du Nord
Danemark
Royaume-Uni
Irlande
Pologne
Biélorussie
Pays-Bas
Londres
Berlin
Allemagne
Belgique
Paris
République tchèque
Slovaquie
Ukraine
Autriche
Hongrie
Moldavie
France
Roumanie
Croatie
Serbie
Italie
Barcelone
Rome
Bulgarie
Portugal
Madrid
Espagne
Mer Méditerranée
Istanbul
Grèce

Demo #1

Sending ‘hello world’

Current Use Cases

Internet of **Things** ?



sigfox

Ideal Sigfox use cases

Independent solutions

No user, no power socket, no local network

Shy devices

Doesn’t speak much, but only useful data

Sigfox foundation

Antarctica

Tracking scientists & assets

Long reach

Ease of use

Robust trackers

sigfox

Wildlife

Rhinos tracking & monitoring

Anti-poaching operation

Implant & play

Long reach

sigfox

sigfox

Industry

sigfox

Health &
Assisted living

sigfox

Public sector

sigfox

Utilities

sigfox

Home & lifestyle

sigfox

Agriculture

sigfox

Retail & other services

sigfox

Hardware solutions

Hardware

SIGFOX is not a hardware vendor

Ecosystem of established partners

Solutions

Modules

Easy to get started

Atim, Telit, TD

Quite **expensive**

for industrialisation

Chipsets

Cheaper

Atmel, OnSemi

Skills needed

Ref.Designs

Transceivers

SiLabs, Texas Instrument

Cheapest solution

Skills++ needed

Certification program

sigfox

Antenna

Not optional :)

Best way to ruin a great device is to mess the antenna integration

Balance between design & performance

We’re here to help you get in touch with specialists if needed

Prototyping

Arduino & Raspberry Pi kits available from various websites

Check out http://partners.sigfox.com for the full details

sigfox

sigfox

Cloud

Get your data

View messages : Sigfox web platform

Get messages : REST API (pull)

Receive new messages : HTTP Callbacks (push)

Callbacks

Each message received from your devices will be forwarded to your application server

Customisable headers & body

You can set more than one callback

sigfox

3rd party platforms

You can easily push your data to a 3rd party platform :

AWS, Azure, Telefonica, thethings.iO, IBM, Samsung…

Downlink messages

A downlink message can be

Semi automatic : sent directly by the network

Customised : sent by your own application server

Semi automatic callback

Simply set up the message to send, it can be:

an hardcoded frame

pre defined variable (timestamp, rssi)

Downlink callbacks

Same mechanism as for the uplink callback, set an URL

Reply with the 8-byte downlink frame

Respect this JSON format :

```
{
  '{{deviceId}}': {
    'downlinkData':{{data}}
  }
}
```

sigfox

sigfox

Workshop session

Using SmartEverything boards

Contribute back

Contribute

Don’t forget to publish your experiments

Code Samples, HW design, fails … will be useful to other people

We all start by copy/pasting ;)

Your own website, github, hackster.io, instructables, etc.

About the Board

The SmartEverything is a multi-purpose development board distributed by Arrow.

Using a Cortex M0+ MCU with an Arduino bootloader, it features a Telit LE-868S Sigfox module, BLE, a GPS and various environnement sensors

Full documentation available on smarteverything.it

The Sigfox module documentation is available on telit.com

sigfox

Useful Resources

Workshop slides http://bit.ly/SMTSmartEverything

Questions ? http://ask.sigfox.com

Github http://github.com/sigfox/makers-tour-resources

Github http://github.com/nicolsc

sigfox

Register

http://backend.sigfox.com/activate

Provider: Arrow // SmartEverything

Country : Slovakia

Device ID & PAC: Sticker + Paper

sigfox

sigfox

Hello World

SmartEverything test

Open the Arduino IDE

Select the board

Board type : `Smart Everything Fox (USB)`

Try one of the `File > Examples > SmartEverything` samples

Sigfox : `File>Examples>SmartEverything>Sigfox>DataModeEu`

sigfox

Hello World Sketch

```
#include <Arduino.h>
#include <Wire.h>

void setup() {
  SerialUSB.begin(115200);
 SigFox.begin(19200);
  delay(500);
  Sigfox.print("+++");
}
```

```
void loop() {
 char output;

SigFox.print("AT$SF=CAFECAFE")
;
 SigFox.print(« \r");
delay(600000);
}
```

sigfox

Message received ?

http://backend.sigfox.com

Navigate to the *devices* menu in the top bar

Click on the ID of your device

Enter the *messages menu* from the left navigation column

sigfox

First callback

Callback setup

Device Type menu

Click on your *device type* name

Enter the *Callbacks* menu

Select *new default callback*

sigfox

Callback setup

TYPE : DATA UPLINK

Choose a *CHANNEL* : URL (EMAIL for a quick test)

Url pattern: URL of your own server

Use HTTP method: GET/POST/PUT

sigfox

Callback status

In the *Devices > Messages* panel, you have a indicator of the callback status (an arrow)

Black : in progress

Green : Callback OK

Red : Callback KO (at least one of the callbacks failed)

Click the arrow to display details.

Downlink

How does it work ?

Send a message, with a *downlink* flag

Once message is sent, the module gets back to sleep

After 20s, it will wake up automatically, in Rx mode

It will wait 20s for a *downlink* message

Afterwards it will get back to sleep

Downlink setup

To setup an automatic callback :

Device Type > Info > Edit

In the *Downlink data* settings, set the following :

Downlink Mode : DIRECT

Set the following value : 123400000BADCAFE

sigfox

How to request a downlink

Same AT command, with additional parameters

```
AT$SF=[hex byte]*, 1
```

sigfox

Handle the response

When entering Rx mode, the module will display

```
+RX BEGIN
```

Received frame (if any) will be displayed as:

```
  +RX= [byte] [byte] [byte] [byte] [byte] [byte] [byte] [byte]
```

End of Rx mode

```
+RX END
```

sigfox

Downlink callback

In *Device Type > Info > Edit*

change *Downlink mode* to CALLBACK

Create a new default callback, with TYPE : DATA | BIDIR

Then set up your URL

sigfox

Sample input output

```
AT$SF=55 50 4C 49 4E 4B,1

OK

+RX BEGIN

+RX=44 4F 57 4E 4C 49 4E 4B

+RX END
```

sigfox

Sample code

Arduino

- sigfox.github.io/makers-tour

Server side

- https://github.com/nicolsc/sigfox-downlink
- PR welcome in different languages

Keep in touch

devrelations@sigfox.com

twitter: @AlexRBucknall

sigfox