BRINGING BEACONS TO WINDOWS PLATFORM

MICHAŁ ŁUSIAK | @MLUSIAK | WWW.MLUSIAK.COM


tretton37

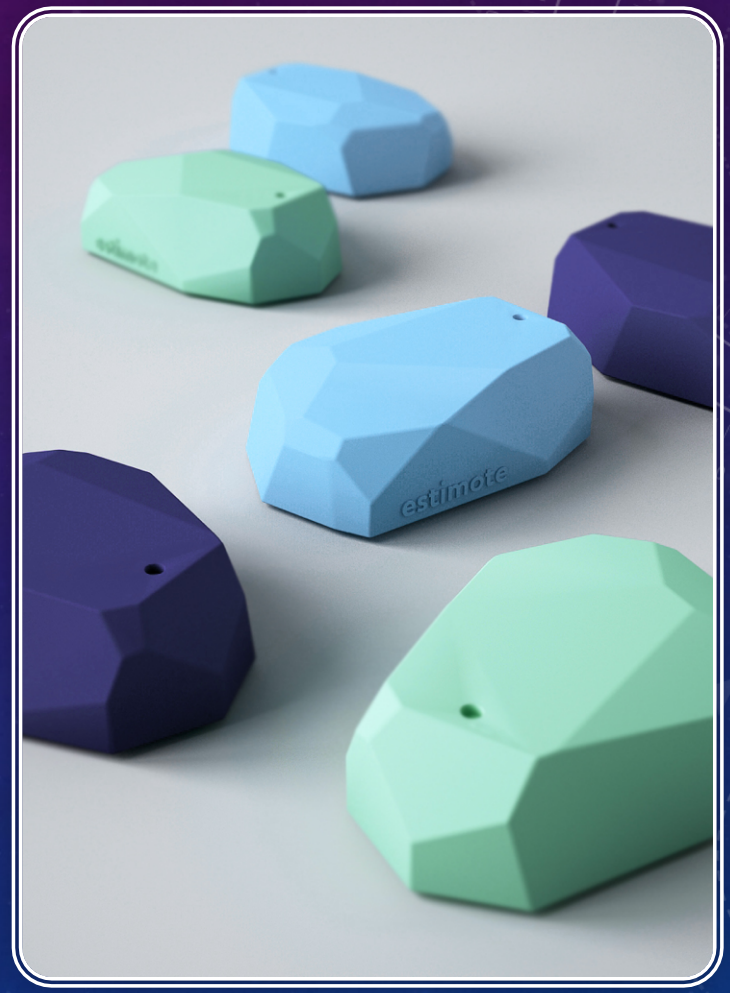
Bacon

WE’RE NOT GONNA TALK ABOUT IT HERE, SORRY :(

Beacons on Windows

BEACONS

Small BLE devices that add physical context to mobile apps



LET'S TALK BLUETOOTH LOW ENERGY (BLE)

- Also called Bluetooth Smart
- Part of BL 4.0 specification
- Not your mother's BL
- Up to 100m range (in reality <60m)
- Uses the same frequency as Bluetooth (2.4 MHZ)
- ... and Microwaves
- Water resonates at 2.4MHZ
- That's why Beacons don't like bottles of water
- And Humans are basically huge walking bottles of water

BEACONS

- Popularized by Apple as iBeacons
- You don’t have to pair with beacons (although some vendors support this model)
- Primarly they broadcast few bytes of Identifiers
- They can not track people...
- ... but app on your phone, that detects them can
- They can broadcast additional data (like temperature from built in sensor)
- They can receive data (i.e. getting updates over-the-air)
- Battery (usually lasting few years) or USB powered

The Hitchhikers Guide to iBeacon Hardware.

A Comprehensive Report by Aislelabs

Aislelabs


Accent Systems

April Brother


Bluecats


Blue Sense


Bkon


Estimote


EMBC01


Gelo


Gimbal Series 10


Gimbal Series 21


Gliworm


HM-10 Dev Kit


Kontakt.io


KS Technologies


Lightcurb


Motorola Mpact


Minew MS63/i3


Minew i5


Minew MS54V3

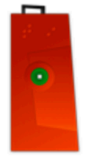

Roximity


Radius Networks


RECO Beacon


RedBear

Sensorberg


SensorTag


Tod

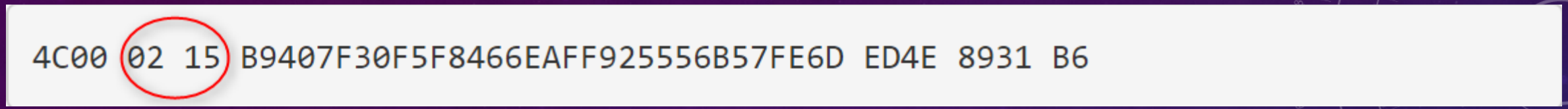
http://www.aislelabs.com/reports/beacon-guide/

- https://www.youtube.com/watch?v=SrsHBjzt2E8

```
4C00 02 15 B9407F30F5F8466EAFF925556B57FE6D ED4E 8931 B6
```

```
4C00 02 15 B9407F30F5F8466EAFF925556B57FE6D ED4E 8931 B6
```

- Apple company identifier



- Apple company identifier
- Data type and data length

```
4C00 02 15 B9407F30F5F8466EAFF925556B57FE6D ED4E 8931 B6
```

- Apple company identifier
- Data type and data length
- UUID (in this example it’s default Estimote UUID)

```
4C00 02 15 B9407F30F5F8466EAFF925556B57FE6D ED4E 8931 B6
```

- Apple company identifier
- Data type and data length
- UUID (in this example it’s default Estimote UUID)
- iBeacon Major and iBeacon Minor

```
4C00 02 15 B9407F30F5F8466EAFF925556B57FE6D ED4E 8931 B6
```

- Apple company identifier
- Data type and data length
- UUID (in this example it’s default Estimote UUID)
- iBeacon Major and iBeacon Minor
- Tx (Measured Power) (for humans – strenght of signal expected 1m from device)

DISTANCE

- Your app knows you're near beacon when you are able to receive it's signal
- You can approximate "how near" thanks to Tx power and RSSI (Received signal power)
- RSSI = TxPower - 10*n*log(d)
- n ~= 2 in empty spaces
- d - distance
- d = 10 ^ ((TxPower - RSSI) / (10 * n))
- With more beacons (min 4 for good precision) you can do indoor location by triangulating distances

EDDYSTONE™

- Google answer to iBeacon
- More frame types
- Eddystone-URL for Physical Web
- Eddystone-UID
- Eddystone-TLM

EXAMPLE APPLICATIONS

- Museums
- Retail spaces
- Airports
- Public transport
- Healthcare
- Home automation

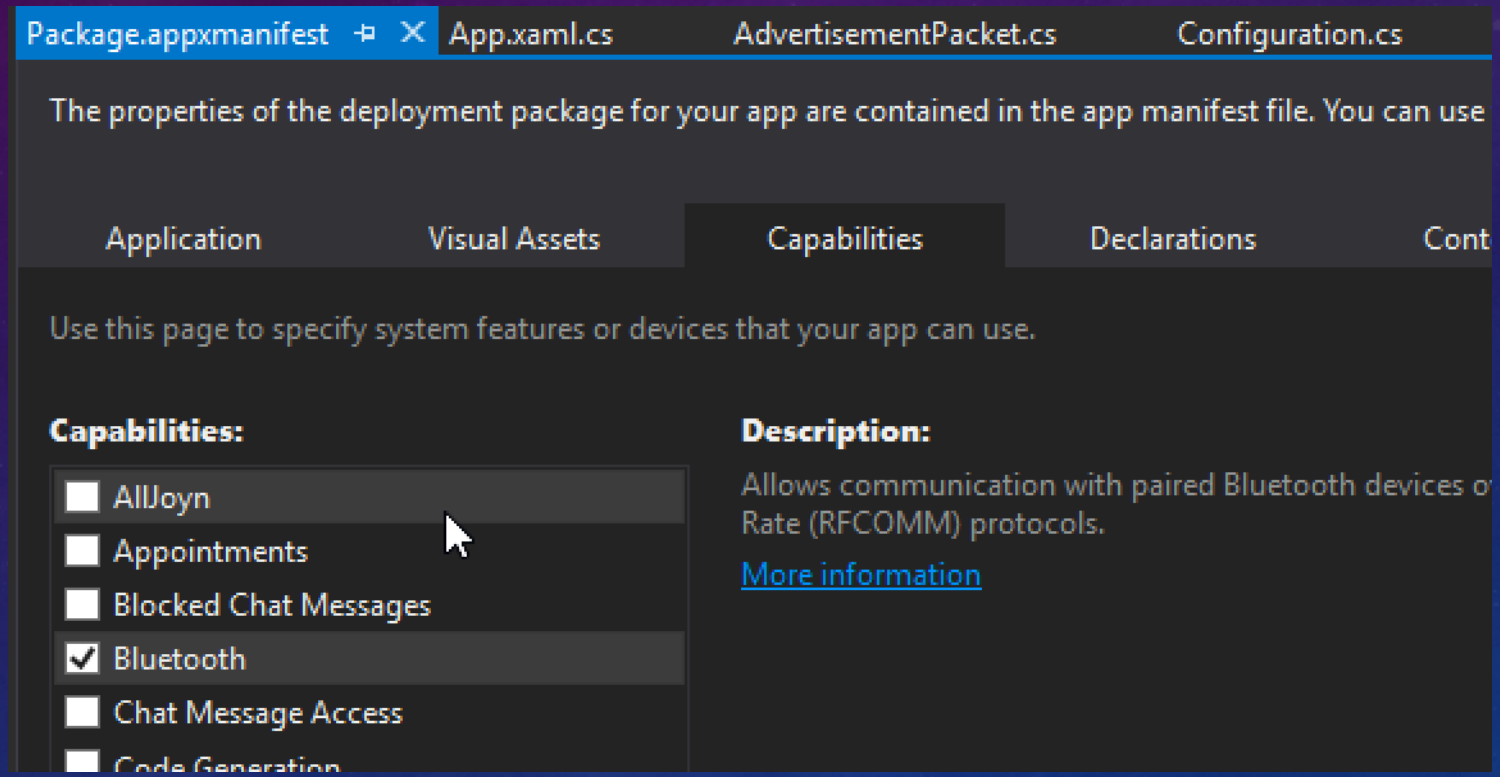
CUSTOM DATA

- Some vendors offer additional features through GATT services
- Or Custom SDKs
- Or GPIO input / outputs
- Fleet management through cloud services

BEACONS AND WINDOWS PHONE: CURRENT STATUS

- None of the major vendors officially support Windows Phone
- All higher end Lumias had BLE enabled radios
- Windows 8 had SDKs to deal with BLE, but quite limited and only on tablets/laptops
- Windows 10 supports BLE advertisment on all devices

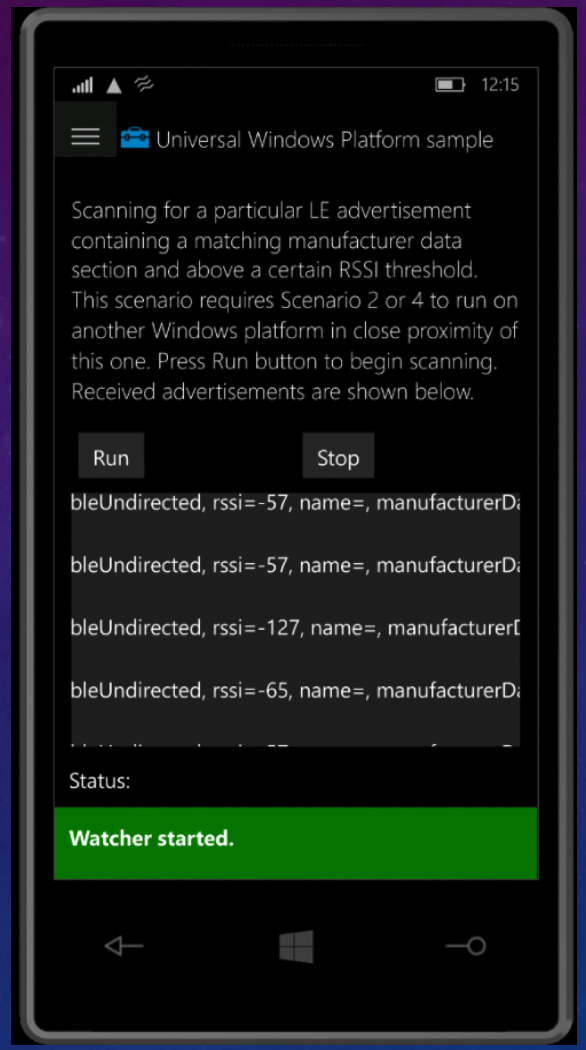
FILE -> NEW PROJECT



CODE

```
public Scenario1_Watcher()
{
    this.InitializeComponent();

    // Create and initialize a new watcher instance.
    watcher = new BluetoothLEAdvertisementWatcher();

    // Filter apple packets (iBeacon packets)
    var manufacturerData = new BluetoothLEManufacturerData();
    manufacturerData.CompanyId = 0x004c;

    // Configuring the signal strength and timeout filter for proximity scenarios
    watcher.SignalStrengthFilter.InRangeThresholdInDBm = -70;
    watcher.SignalStrengthFilter.OutOfRangeThresholdInDBm = -100;
    watcher.SignalStrengthFilter.OutOfRangeTimeout = TimeSpan.FromMilliseconds(2000);

}
```

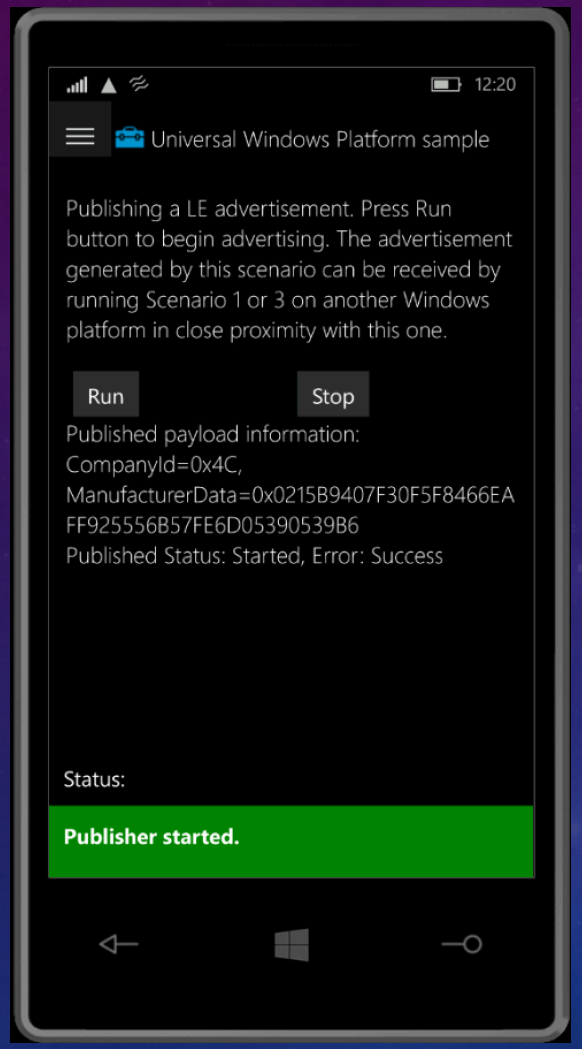
DEMO

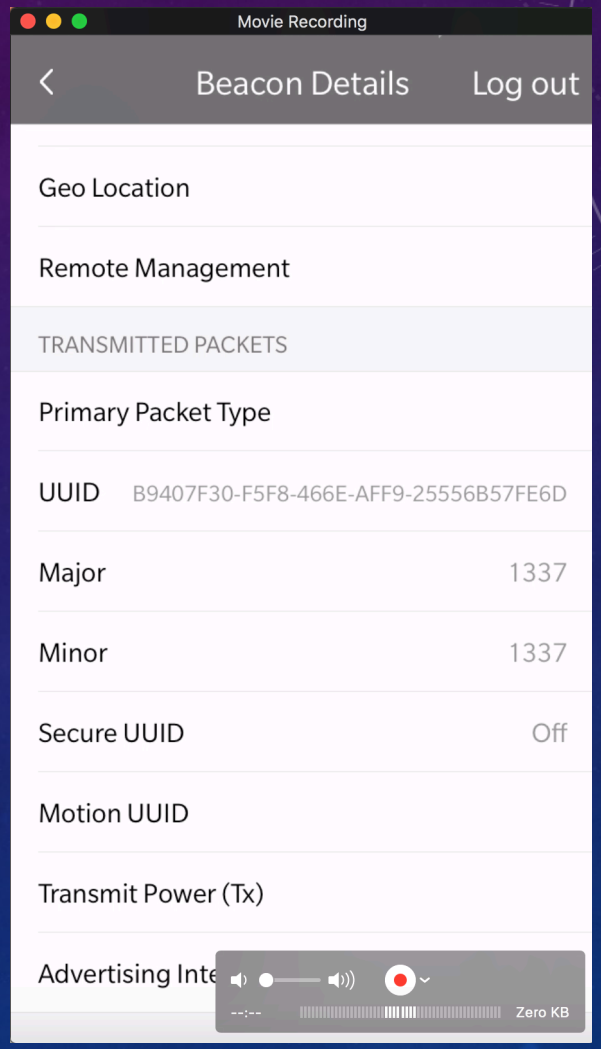

12:15
Universal Windows Platform sample
Scanning for a particular LE advertisement containing a matching manufacturer data section and above a certain RSSI threshold. This scenario requires Scenario 2 or 4 to run on another Windows platform in close proximity of this one. Press Run button to begin scanning. Received advertisements are shown below.
Run
Stop
bleUndirected, rssi=-57, name=, manufacturerDa
bleUndirected, rssi=-57, name=, manufacturerDa
bleUndirected, rssi=-127, name=, manufacturerD
bleUndirected, rssi=-65, name=, manufacturerDa
Status:
Watcher started.

CODE

```
public Scenario2_Publisher()
{
    this.InitializeComponent();

    publisher = new BluetoothLEAdvertisementPublisher();

    var manufacturerData = new BluetoothLEManufacturerData();
    manufacturerData.CompanyId = 0x004C;
    publisher.Advertisement.ManufacturerData.Add(manufacturerData);

    var hex = "0215B9407F30F5F8466EAFF925556B57FE6D0539053986";

    var bytes = Enumerable.Range(0, hex.Length)
              .Where(x => x % 2 == 0)
              .Select(x => Convert.ToByte(hex.Substring(x, 2), 16))
              .ToArray();


    var writer = new DataWriter();
    writer.WriteBytes(bytes);

    // Display the information about the published payload
    PublisherPayloadBlock.Text =
        string.Format("Published payload information: CompanyId=0x{0}, ManufacturerData=0x{1}",
            manufacturerData.CompanyId.ToString("X"),
            hex);

    // Display the current status of the publisher
    PublisherStatusBlock.Text = string.Format("Published Status: {0}, Error: {1}",
        publisher.Status,
        BluetoothError.Success);
}
```

DEMO


12:20
Universal Windows Platform sample
Publishing a LE advertisement. Press Run button to begin advertising. The advertisement generated by this scenario can be received by running Scenario 1 or 3 on another Windows platform in close proximity with this one.
Run
Stop
Published payload information:
CompanyId=0x4C,
ManufacturerData=0x0215B9407F30F5F8466EAFF925556B57FE6D05390539B6
Published Status: Started, Error: Success
Status:
Publisher started.


Movie Recording
Beacon Details
Log out
Geo Location
Remote Management
TRANSMITTED PACKETS
Primary Packet Type
UUID
B9407F30-F5F8-466E-AFF9-25556B57FE6D
Major
1337
Minor
1337
Secure UUID
Off
Motion UUID
Transmit Power (Tx)
Advertising Inte
--:--
Zero KB

SMOKEY BEACON

```
public static BluetoothLEAdvertisementWatcher WatcherConfiguration()
{
    var watcher = new BluetoothLEAdvertisementWatcher();
    watcher.SignalStrengthFilter.InRangeThresholdInDBm = -90;
    watcher.SignalStrengthFilter.OutOfRangeThresholdInDBm = -120;
    watcher.SignalStrengthFilter.OutOfRangeTimeout = TimeSpan.FromMilliseconds(5000);

    var manufacturerData = new BluetoothLEManufacturerData();
    manufacturerData.CompanyId = 0x004c;
    watcher.AdvertisementFilter.Advertisement.ManufacturerData.Add(manufacturerData);

    return watcher;
}
```

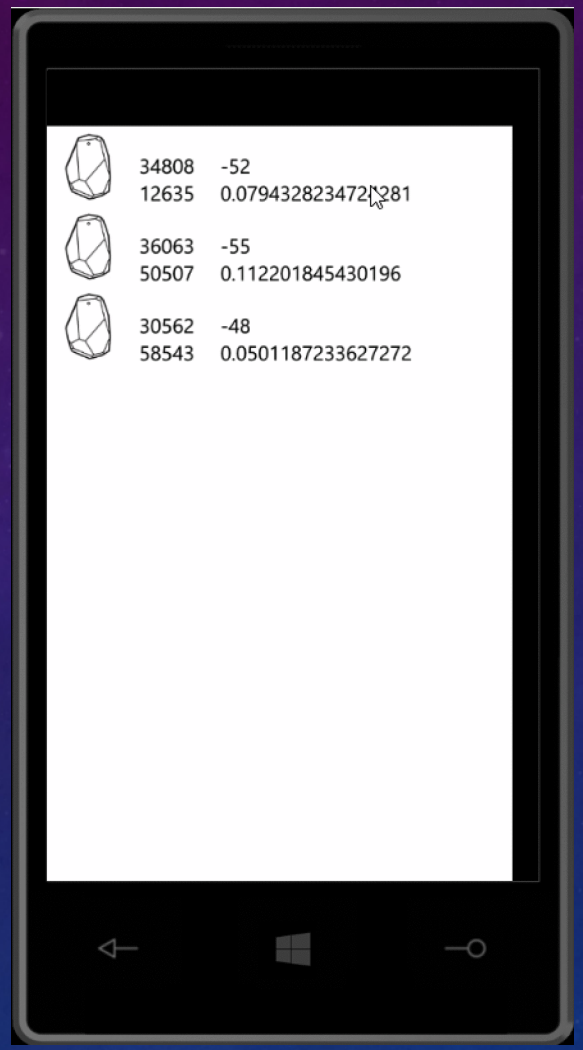
SMOKEY BEACON

```
public class EstimoteBeacon : IBeacon
{
    public string DeviceAddress { get; set; }
    public string UUID { get; set; }
    public ushort Major { get; set; }
    public ushort Minor { get; set; }
    public short RSSI { get; set; }
    public sbyte TxPower { get; set; }
    public DateTime Timestamp { get; set; }
    public double Distance => CalculateDistance(this.RSSI, this.TxPower);

    private static double CalculateDistance(short rssi, sbyte txPower)
    {
        /*
         * RSSI = TxPower - 10 * n * lg(d)
         * n = 2 (in free space)
         *
         * d = 10 ^ ((TxPower - RSSI) / (10 * n))
         */

        return Math.Pow(10d, ((double)txPower - rssi) / (10 * 2));
    }
}
```

SMOKEY BEACON

```
public MainPage()
{
    this.InitializeComponent();

    Watcher = Configuration.WatcherConfiguration();
    Discovery = new EstimoteDiscovery();
    ViewModel = new BeaconListViewModel();
}

protected override void OnNavigatedTo(NavigationEventArgs e)
{
    Watcher.Received += OnAdvertisementReceived;
    Watcher.Stopped += OnAdvertisementWatcherStopped;
    Watcher.Start();
}

private async void OnAdvertisementReceived(BluetoothLEAdvertisementWatcher watcher, BluetoothLEAdvertisementReceivedEventArgs eventArgs)
{
    Discovery.BeaconFromRawBluetoothPacket(eventArgs);
    var beacons = Discovery.BeaconsInRange;

    /* code handling passing data to view */
}
```

DEMO


34808 -52
12635 0.079432823472[illegible]281
36063 -55
50507 0.112201845430196
30562 -48
58543 0.0501187233627272

SECURITY IMPLICATIONS

- Spoofing beacons
- Countermeasures

CODE REPOSITORIES

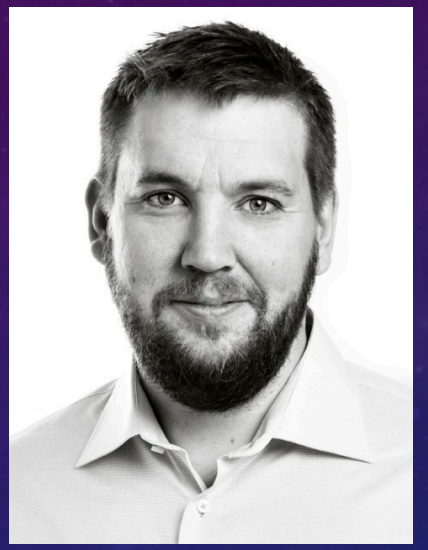
- Microsoft's UWP Samples
- Smoky Beacon
- danardealan's Beacons.Universal

THANK YOU

MICHAŁ ŁUSIAK | @MLUSIAK | WWW.MLUSIAK.COM

Enrico Campidoglio

- Friday 11:40 - The Things Git Can Do



Karl-Henrik Nilsson

- Wednesday 11:40 - Hacking your home
- Wednesday 17:40 - IoT at home
- Thursday 11:40 - Windows IoT Core