®Bluetooth LE untenrum

Tim Becker, Matthias Krauß
Press Any Key UG

Cocoaheads - 2014-04-24 - ble@press-any-key.de - @presseverykey

Zoé

Bluetooth (BR/EDR/HS)

Bluetooth LE

Classic

Smart

Smart Ready

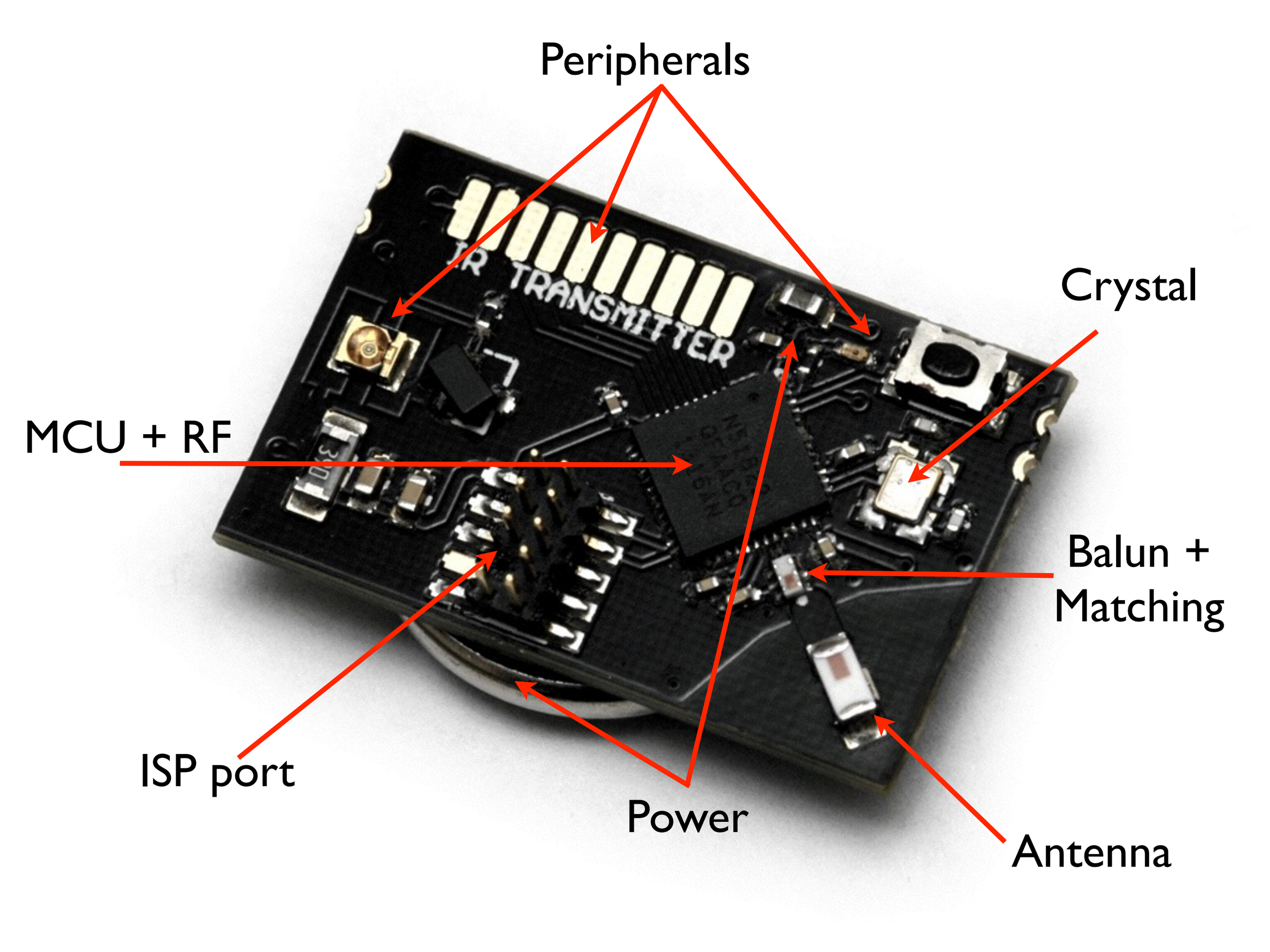
Cocoaheads - 2014-04-24 - ble@press-any-key.de - @presseverykey

What's so cool about it?

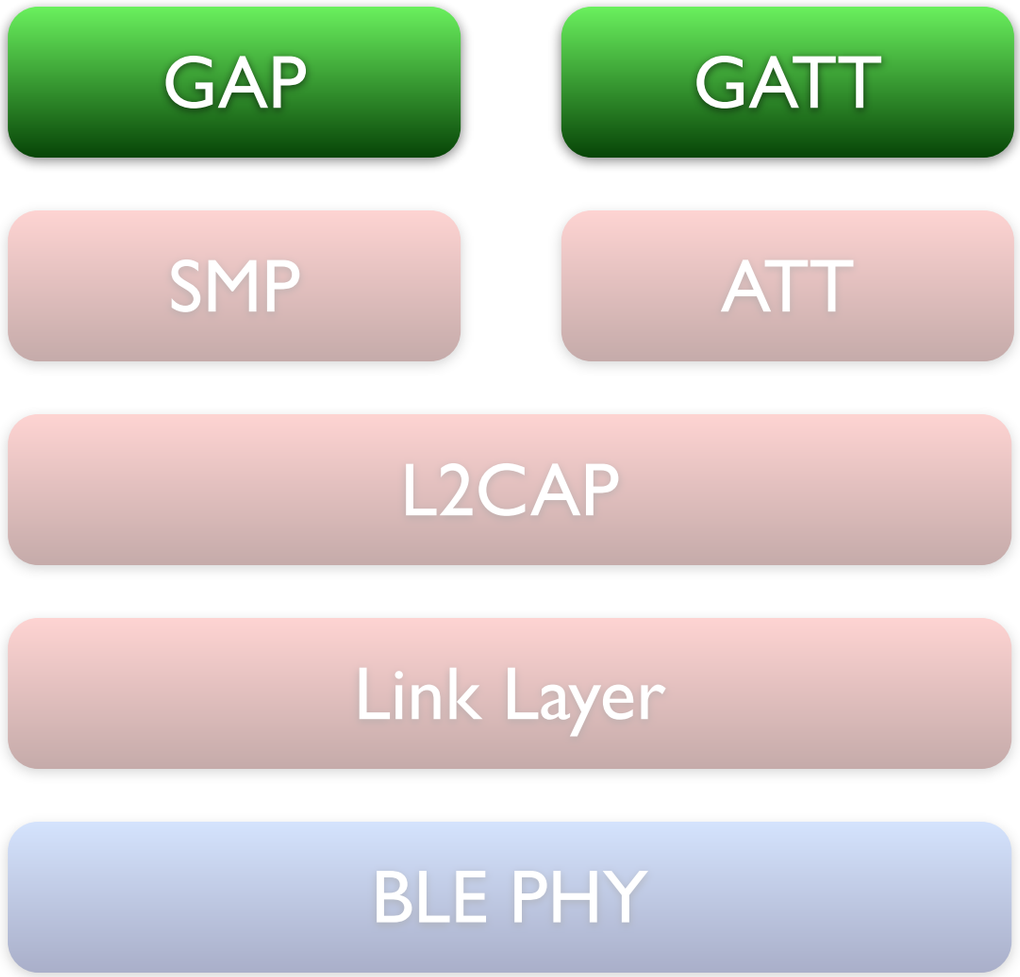
- Really low power consumption (~0.153uJ/bit)
- Cheap to build
- Relatively simple (for a wireless protocol)
- Wide support
- No MFI

1 Becher Joghurt = 727GB!

Der große Bauer Heidelbeere-Cassis


Peripherals
Crystal
MCU + RF
Balun +
Matching
ISP port
Power
Antenna

Stack



GAP Roles

Broadcaster ⟶ Listener

Peripheral ⟷ Central

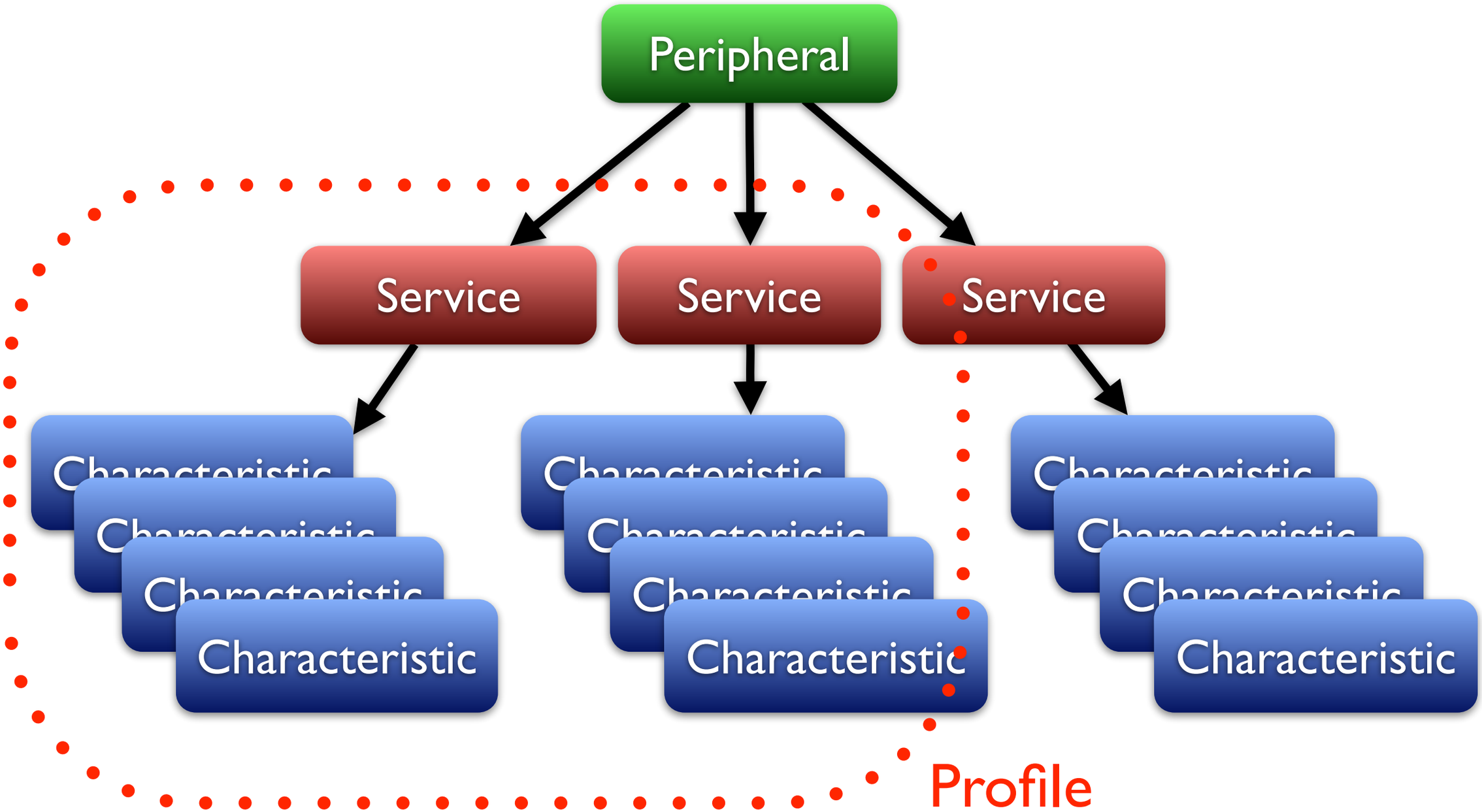
iBeacon advertisement

```
02 01 04 1a ff 4c 00 02
15 16 a2 23 0e 0b 8b 4d
3d bf 83 36 da 12 00 19
90 00 00 00 04 ba
```

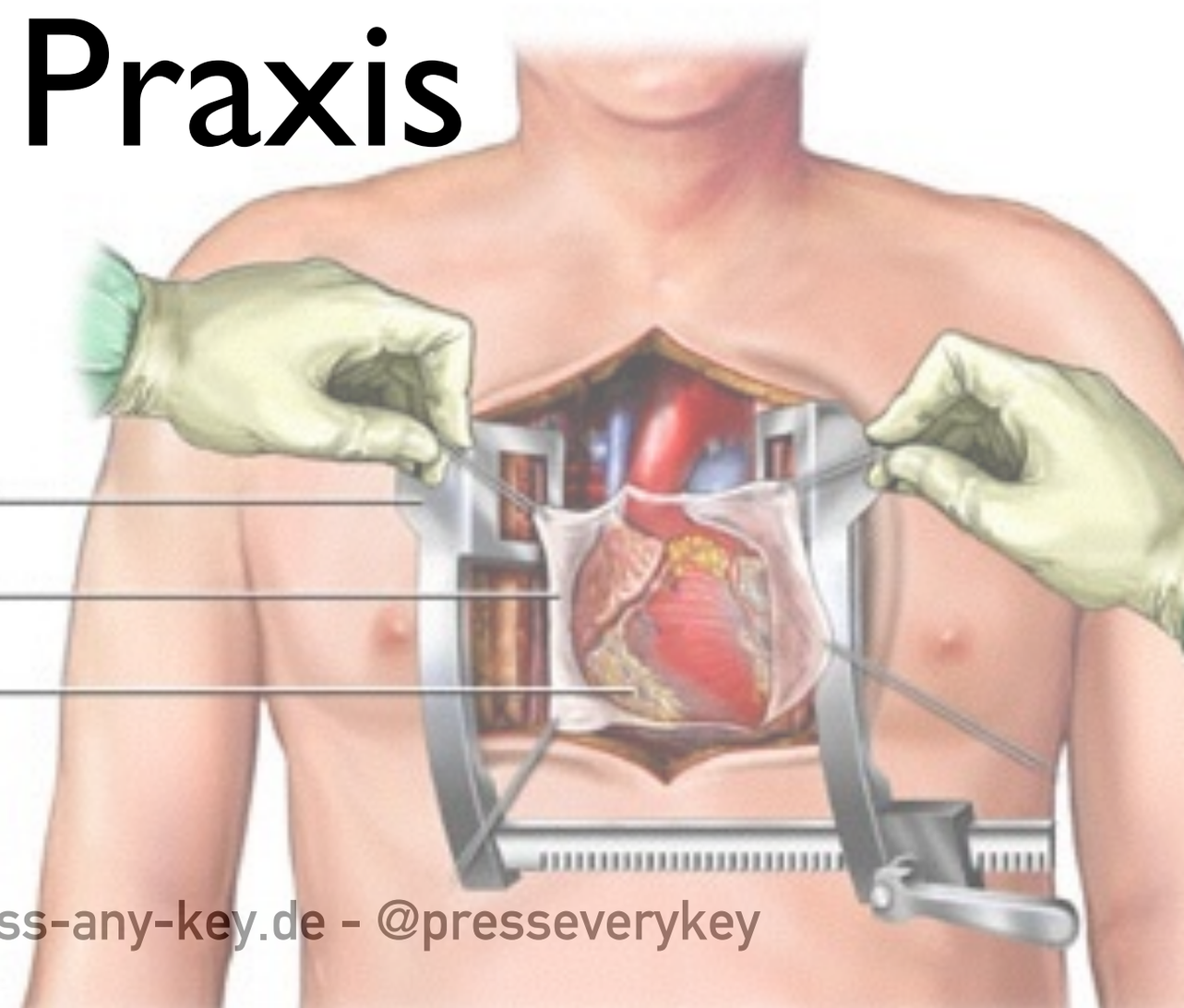
iBeacon advertisement

`02`	Field length: 2 (1 byte type + 1 byte payload)
`01`	Advertising data type: Flags
`04`	Flag: BR/EDR not supported (BLE only)
`1a`	Field length: 26 bytes
`ff`	Advertising data type: Manufacturer-specific data
`4c 00`	Manufacturer ID: 0x004c = Apple
`02`	Apple payload ID: 2 = iBeacon
`15`	Apple payload length: 21 bytes
`16 a2 23 0e 0b 8b 4d 3d bf 83 36 da 12 00 19 90`	16 byte iBeacon UUID
`00 00`	Major 0
`00 04`	Minor 4
`ba`	TX level: RSSI -70dB at 1m

GATT


Peripheral
Service
Service
Service
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Profile

®Bluetooth LE in der Praxis



iBeacon

0. include

```
//iOS 7+, OSX 10.9+ :  CoreLocation.framework

#import <CoreLocation/CoreLocation.h>
```

1. Create location manager

```
@property (strong) CLLocationManager* locationManager;


self.locationManager = [[CLLocationManager alloc] init];
if (!self.locationManager) {
    if (![CLLocationManager authorizationStatus]) {
        NSLog(@"Location Services not allowed");
    } else {
        NSLog(@"Location Services init failed");
    }
    return;
}
self.locationManager.delegate = self;
```

2. range beacon region

```
@property (strong) NSUUID* beaconUUID;
@property (strong) CLBeaconRegion* region;

NSString* uuidString = @"16A2230E-0B8B-4D3D-BF83-36DA12001990";
self.beaconUUID = [[NSUUID alloc] initWithUUIDString:uuidString];
self.region = [[CLBeaconRegion alloc]
                  initWithProximityUUID:self.beaconUUID
                                 identifier:uuidString];

[self.locationManager startRangingBeaconsInRegion:self.region];
```

3. enjoy beacon ranging

```
- (void)locationManager:(CLLocationManager*)manager
        didRangeBeacons:(NSArray*)beacons
               inRegion:(CLBeaconRegion*)region {

    if ([beacons count] > 0) {
        CLBeacon* strongestBeacon = [beacons objectAtIndex:0];
        NSLog(@"Found major: %i minor: %i proximity: %i rssi: %i",
                                strongestBeacon.major.intValue,
                                strongestBeacon.minor.intValue,
                                strongestBeacon.proximity,
                                (int)(strongestBeacon.rssi));
    } else {
        NSLog(@"No beacon in range");
    }
}
```

Bluetooth LE Device

Cocoaheads - 2014-04-24 - ble@press-any-key.de - @presseverykey

0. include

```
//iOS: CoreBluetooth.framework

#import <CoreBluetooth/CoreBluetooth.h>

//OSX: IOBluetooth.framework/CoreBluetooth.framework

#import <IOBluetooth/IOBluetooth.h>
```

0. XML Schmerz

```
<uses-permission
  android:name="android.permission.BLUETOOTH"/>
<uses-permission
  android:name=“android.permission.BLUETOOTH_ADMIN”/>


<uses-feature android:name="android.hardware.bluetooth_le"
  android:required=“true”/>

… oder …

getPackageManager().hasSystemFeature(
  PackageManager.FEATURE_BLUETOOTH_LE
)
```

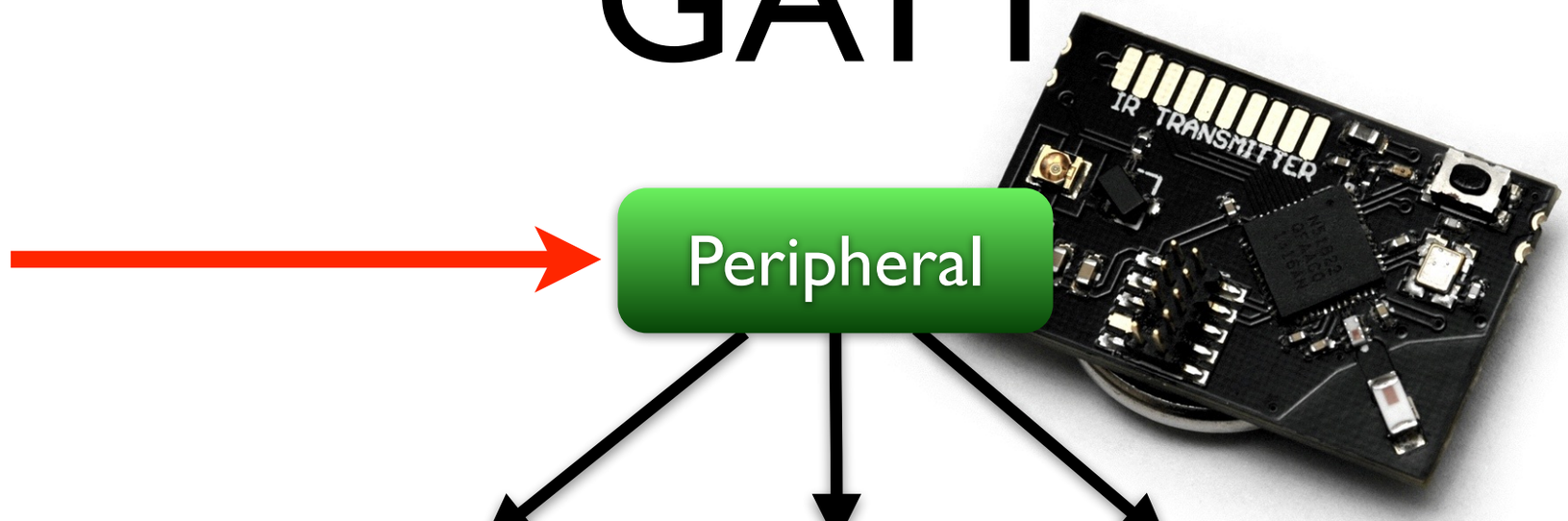
1. Boot, scan

```
- (void)applicationDidFinishLaunching:(NSNotification*)n
{
    self.mgr = [[CBCentralManager alloc] initWithDelegate:self
                                                    queue:nil];
    ...
}

- (void)centralManagerDidUpdateState:(CBCentralManager*)central
{
    if (state == CBCentralManagerStatePoweredOn)
    [self.mgr scanForPeripheralsWithServices:nil
                                     options:nil];
}
```

I. Init, anschalten

```
BluetoothManager bluetoothManager =
  (BluetoothManager) getSystemService(Context.BLUETOOTH_SERVICE);

BluetoothAdapter bleAdapter = bluetoothManager.getAdapter();

if (!bleAdapter.isEnabled()) {
  // Android Intent Geraffel …
}

BluetoothAdapter.LeScanCallback callback = new … {
  public void onLeScan(
      BluetoothDevice device,
      int rssi,
      byte[] scanRecord) { (…) }
}

bleAdapter.startScan(callback);
```

GATT


IR TRANSMITTER
Peripheral
Service
Service
Service
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic

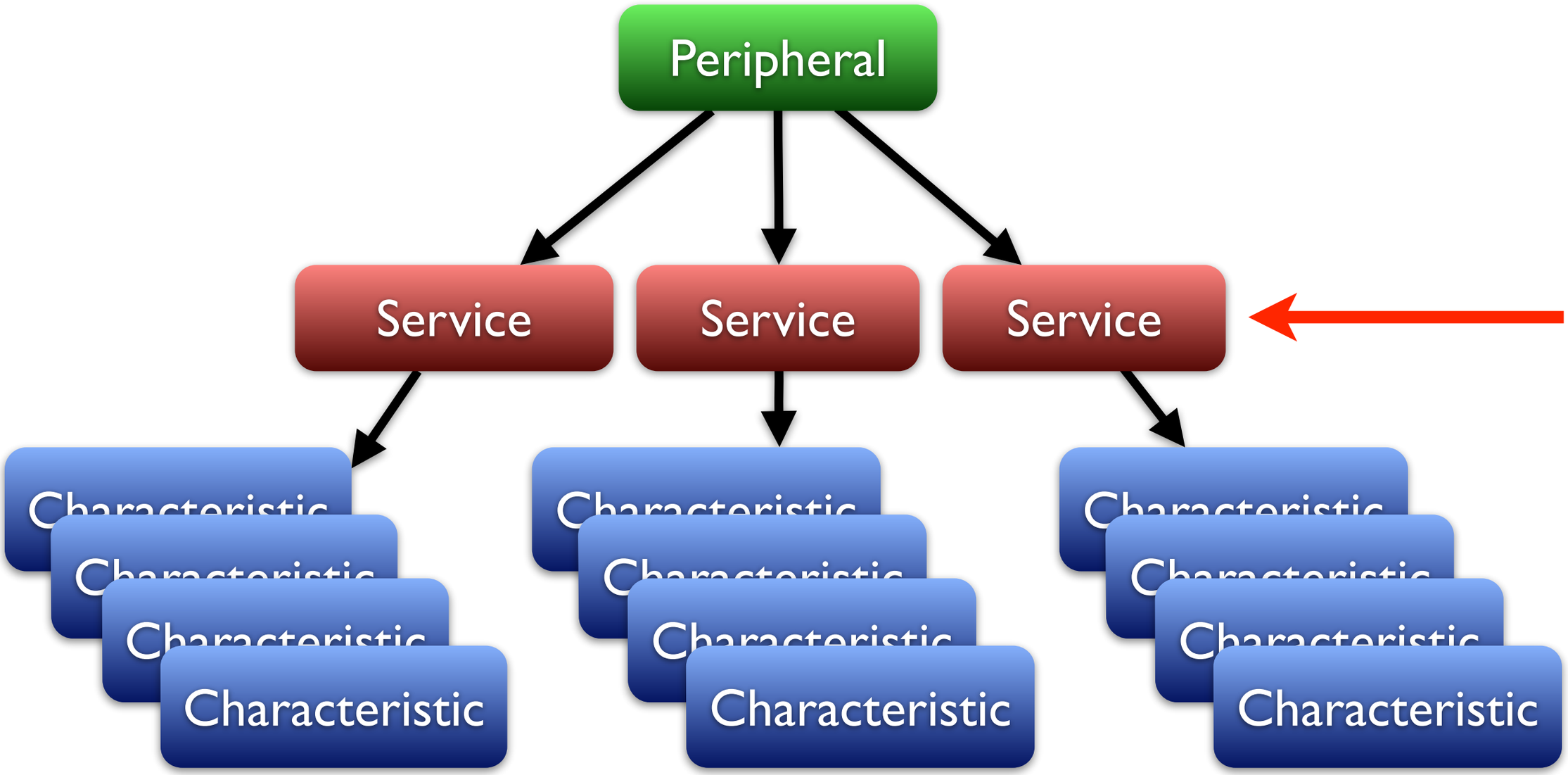
2. Connect, discover services

```
-  (void)centralManager:(CBCentralManager*)central
  didDiscoverPeripheral:(CBPeripheral*)peripheral
      advertisementData:(NSDictionary*)advertisementData
                   RSSI:(NSNumber*)RSSI
{
    ...
      [self.mgr stopScan];
      peripheral.delegate = self;
      self.peripheral = peripheral;
      [self.mgr connectPeripheral:peripheral options:nil];
}

- (void)centralManager:(CBCentralManager*)central
  didConnectPeripheral:(CBPeripheral*)peripheral
{
    [peripheral discoverServices:nil];
}
```

2. Connect, discover services

```
BluetoothDevice device = …
 device.connectGatt(ctx, false, gattCallback);

BluetoothGattCallback  gattCallback = new BluetoothGattCallback()
{
 public void onConnectionStateChange(
  BluetoothGatt gatt,
 int  status,
 int newState) {
 // error handling, check state
 …
 gatt.discoverServices();
}
```

GATT

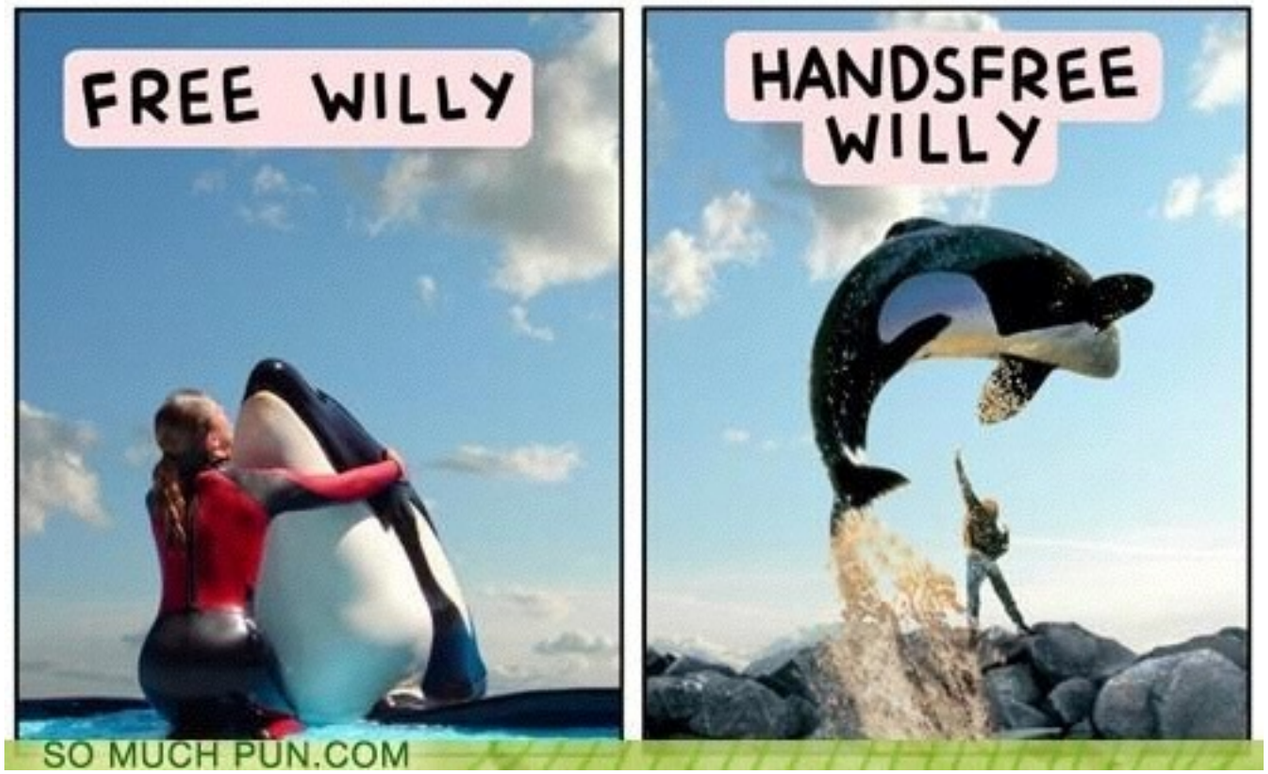

Peripheral
Service
Service
Service
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic

3. Discover characteristics

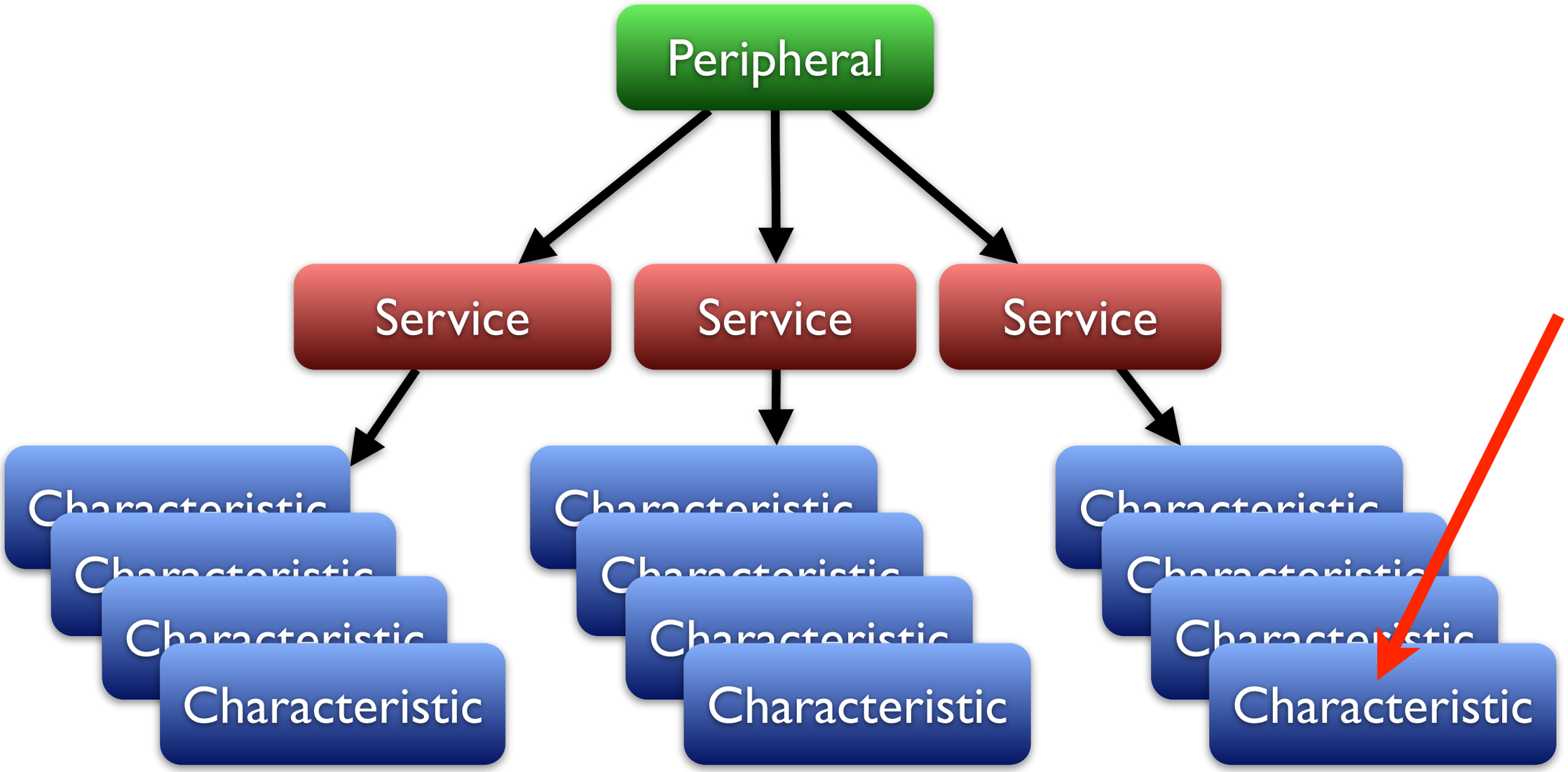
```
-(void)    peripheral:(CBPeripheral*)peripheral
  didDiscoverServices:(NSError*)error
{
  for (CBService* service in peripheral.services) {
    if ([service.UUID isEqual:
          [CBUUID UUIDWithString:IRTEMP_UUID]]) {
      self.tempService = service;
     [self.peripheral discoverCharacteristics:nil
                                   forService:self.tempService];
    }
  }
}
```

3. Discover characteristics

```
BluetoothGattCallback  gattCallback = new BluetoothGattCallback {
  (…)
  public void onServicesDiscovered(
    BluetoothGatt gatt,
    int status
  ) {
    List<BluetoothGattService> services = gatt.getServices()
  for (BluetoothGattService service : services) {
     BluetoothGattCharacteristic characteristic =
        service.getCharacteristic(uuid);
     gatt.readCharacteristic(characteristic)
  }
}
```


FREE WILLY
HANDSFREE WILLY
SO MUCH PUN.COM

GATT


Peripheral
Service
Service
Service
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic
Characteristic

4. Configure, subscribe

```
- (void) peripheral:(CBPeripheral*)peripheral
  didDiscoverCharacteristicsForService:(CBService*)service
                                 error:(NSError*)error
{
  self.tempDataChr =
   [self findCharacteristicWithUUIDString:IRTEMP_DATA_UUID
                                inService:service];

  self.tempConfChr =
     [self findCharacteristicWithUUIDString:IRTEMP_CONF_UUID
                                  inService:service];
  uint8_t on = 1;
  NSData* onOffData = [NSData dataWithBytes:&on length:1];

  [self.peripheral writeValue:onOffData
            forCharacteristic:self.tempConfChr
                         type:CBCharacteristicWriteWithResponse];

  [self.peripheral setNotifyValue:YES
                forCharacteristic:self.tempDataChr];
};
```

4. Configure, subscribe

```
BluetoothGatt gatt = … ;
gatt.readCharacteristic(characteristic);
gatt.writeCharacteristic(characteristic);
gatt.setCharacteristicNotification(characteristic);
```

5. Enjoy your data

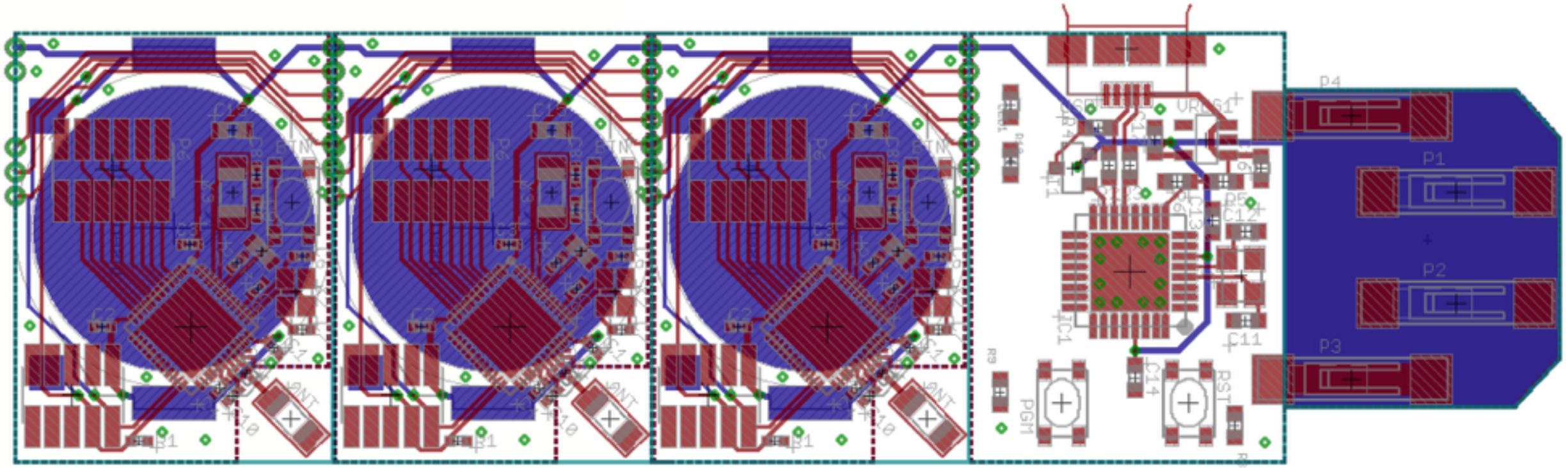
```
- (void)peripheral:(CBPeripheral*)peripheral
     didUpdateValueForCharacteristic:(CBCharacteristic*)c
                               error:(NSError*)error
{
  const uint8_t* data = [self.tempDataChr.value bytes];
  ...
}
```

5. Enjoy your data

```
BluetoothGattCallback  gattCallback = new BluetoothGattCallback {
 public void onCharacteristicRead(
  BluetoothGatt gatt,
  BluetoothGattCharacteristic characteristic,
  int status
 ){(…)}

 public void onCharacteristicWrite(
  BluetoothGatt gatt,
  BluetoothGattCharacteristic characteristic,
  int status
 ){(…)}

 public onCharacteristicChanged(BluetoothGatt gatt,
  BluetoothGattCharacteristic characteristic){}
};
```

https://github.com/presseverykey/everykey-beacon-samples

Shameless self-plug



Thanks!

Questions?

Cocoaheads - 2014-04-24 - ble@press-any-key.de - @presseverykey

Thanks!

Questions?

Cocoaheads - 2014-04-24 - ble@press-any-key.de - @presseverykey