

*Materialien im Krankenhaus wiederfinden*




*Adam Mechouate*

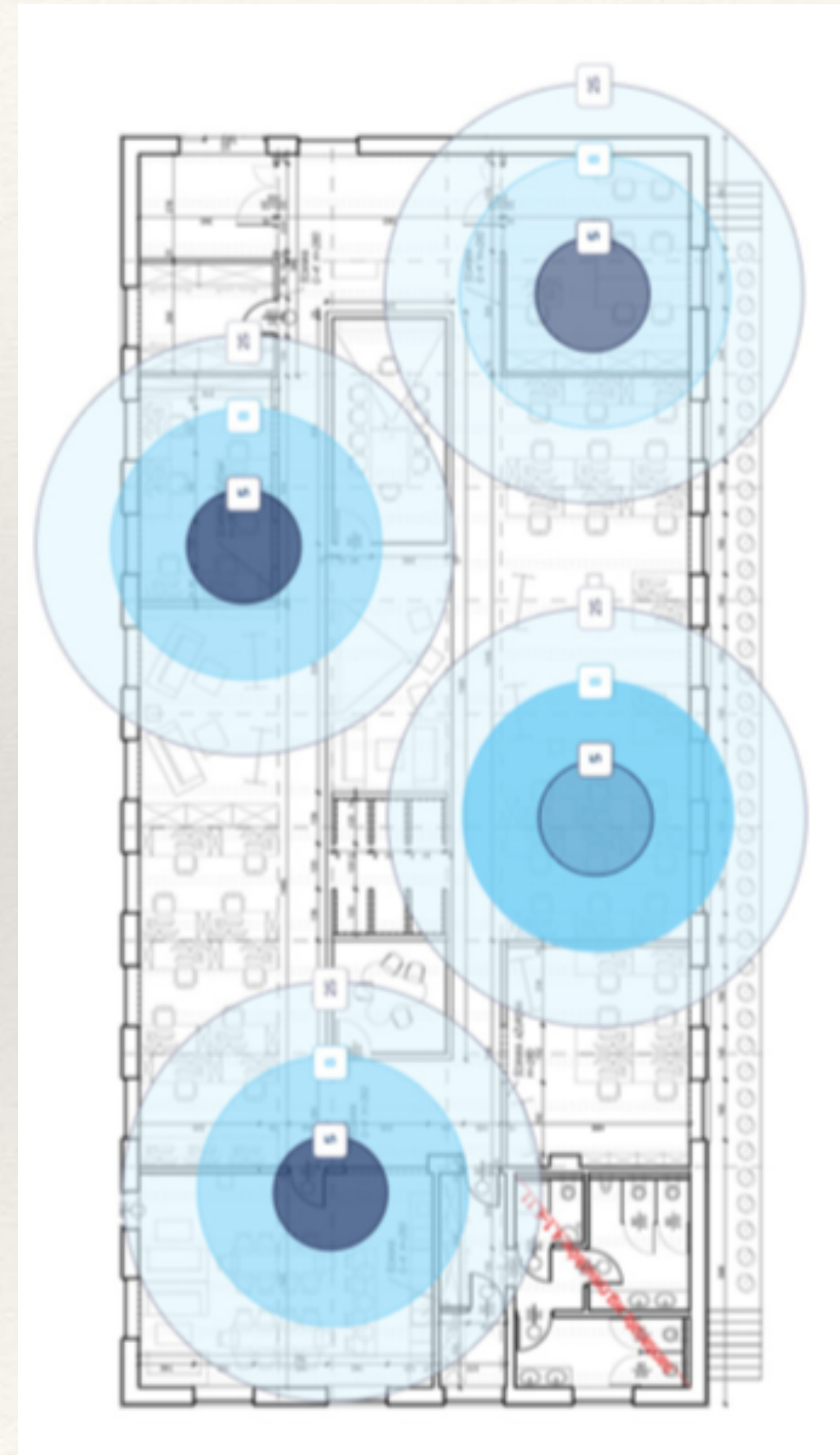
# Hospitracking





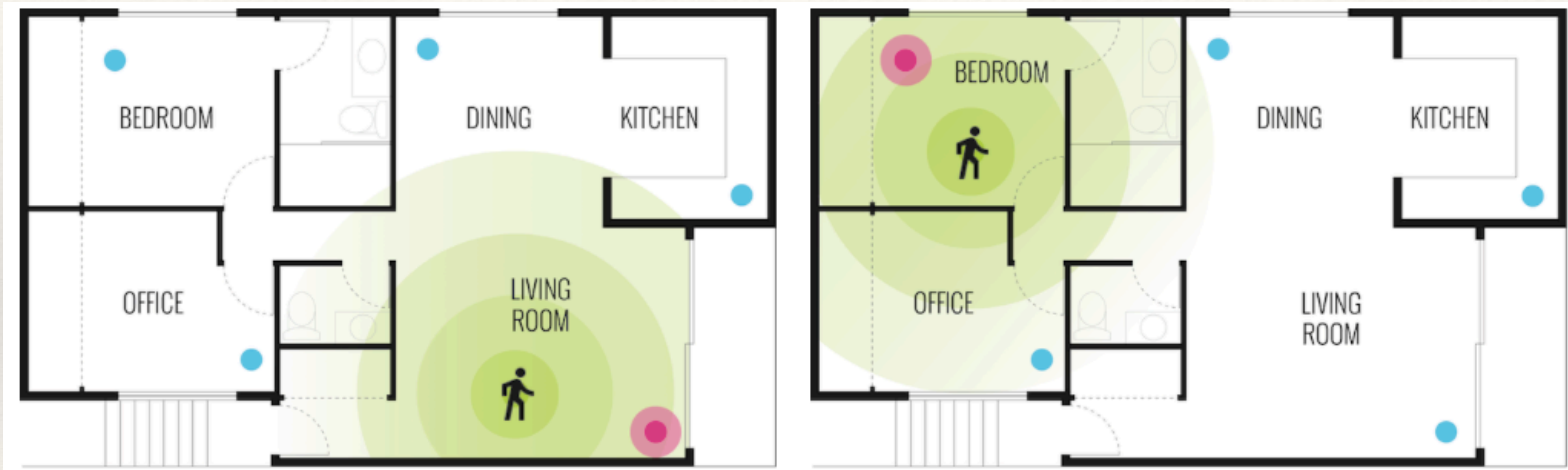
# Real Time Location Systems

- ❖ Comparing RTLS infrastructure: (Hardware)
  - ❖ Alle Technologie (RTLS)
    - ❖ Wlan RTLS (Power-hungry) 
    - ❖ Ultra wide-Band RTLS (less than a meter)
    - ❖ Passive RFID (ca. 20 cm, veraltete technologie) 
    - ❖ Bluetooth Low Energy (BLE) bis 30 meter 

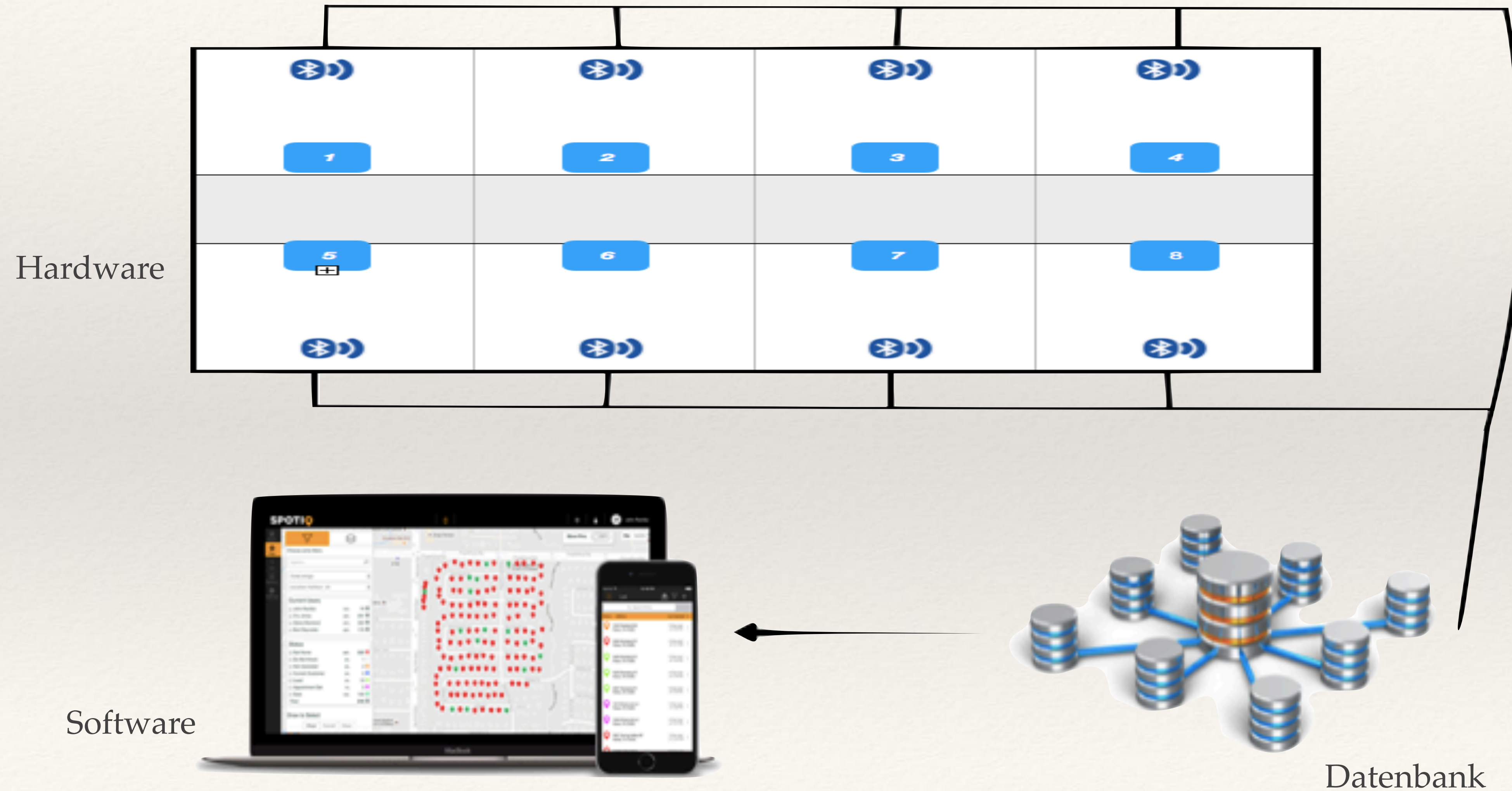




# Wie funktioniert das ?



# Wie funktioniert das ?





# Materialien im Krankenhaus wiederfinden !

- ❖ Was brauchen wir dafür ?



Small, configurable, long-lasting, Bluetooth LE beacon

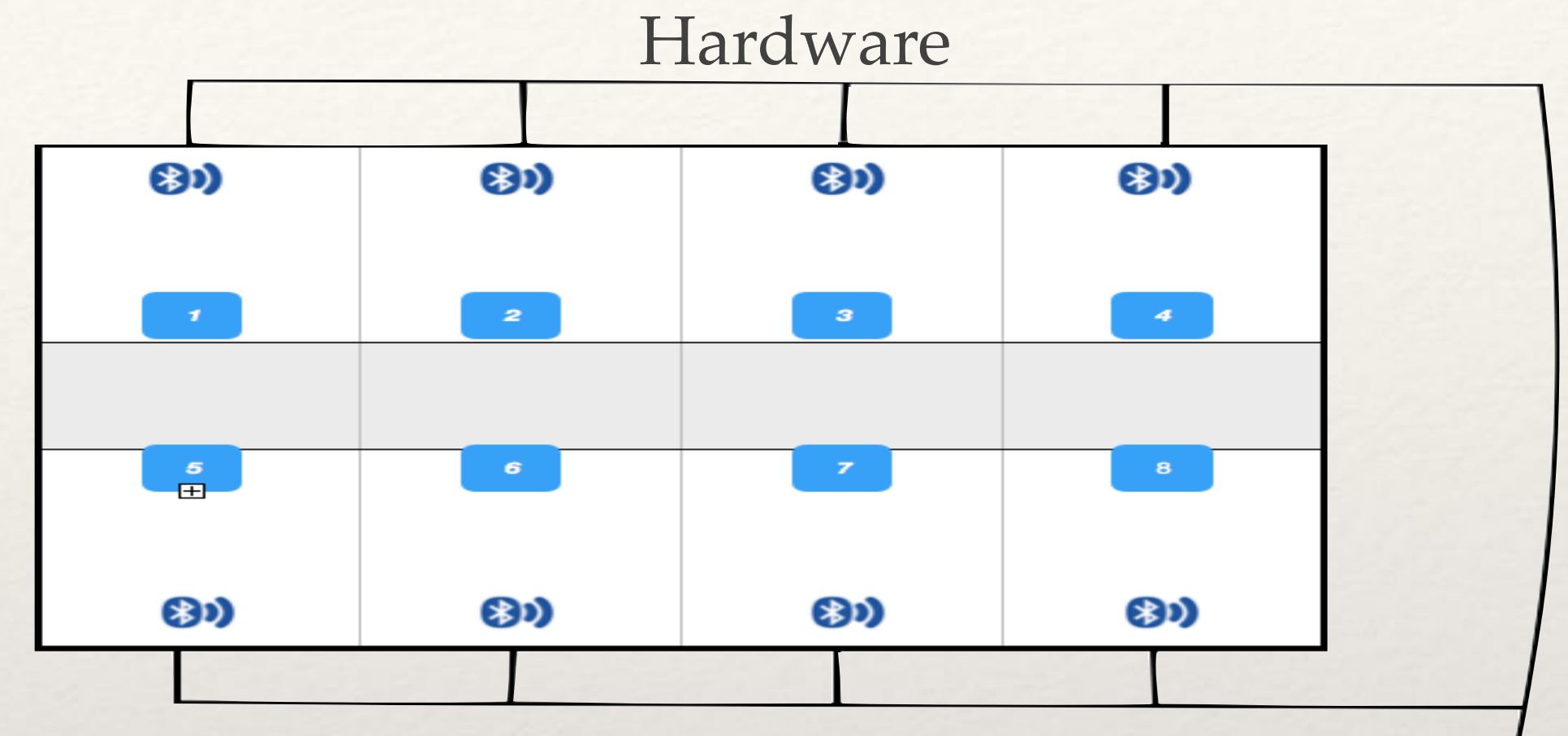


Bluetooth-detecting, wifi-connecting presence detector



# Softwareentwicklung

- ❖ Datenbank My SQL ( open-source )
- ❖ MQTT server (Publisher / Subscriber)
  - ❖ Mosquitto
- ❖ Webbasierte Anwendung HTML & Javascript
- ❖ Facultatif
- ❖ (Mobile App Android Studio)



Software



Datenbank



# Technical



Happy Bubbles Bluetooth Presence Detector



raspberry pi 3



Bluetooth

Wlan

(pub/sub)

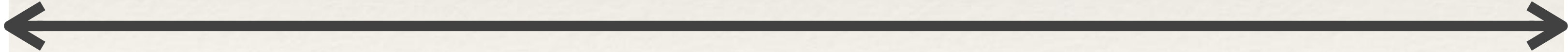
MQTT Server  
Mosquitto

Publish

Subscribe


HTML  
Javascript + Datenbank (My SQL)

IP-Adresse  
Browser



Informationsstrom

Added Beacons

Beacon Name	Current Location	Last seen	Edit	Delete
				





# Technical



Message Filtering -1-  
Happy Bubbles Bluetooth Presence Detector



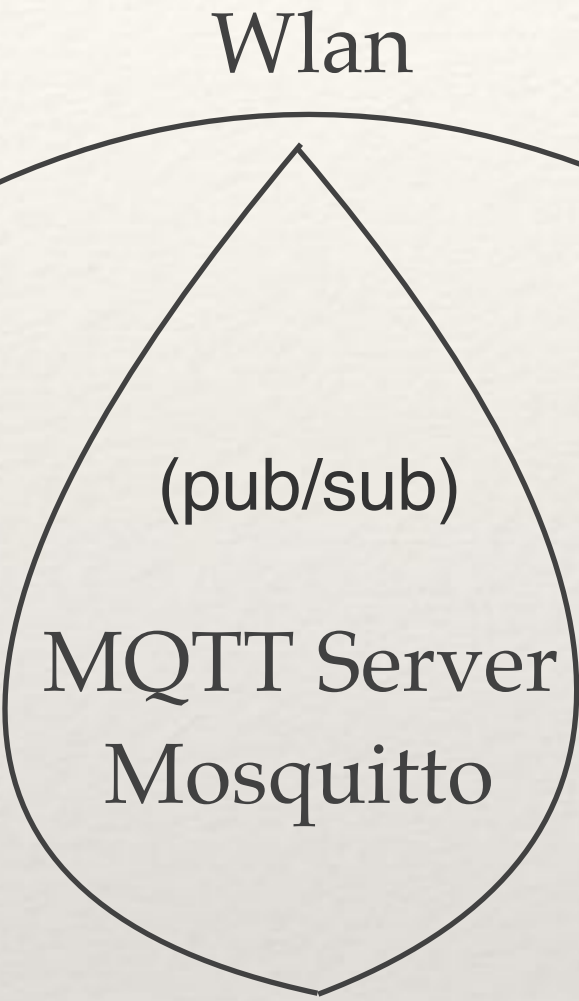
Bluetooth

Message Filtering 2  
raspberry pi 3



HTML Javascript + Datenbank (My SQL)

IP-Adresse  
Browser




Publish

Subscribe

Informationsstrom

## Added Beacons

Beacon Name	Current Location	Last seen	Edit	Delete
				



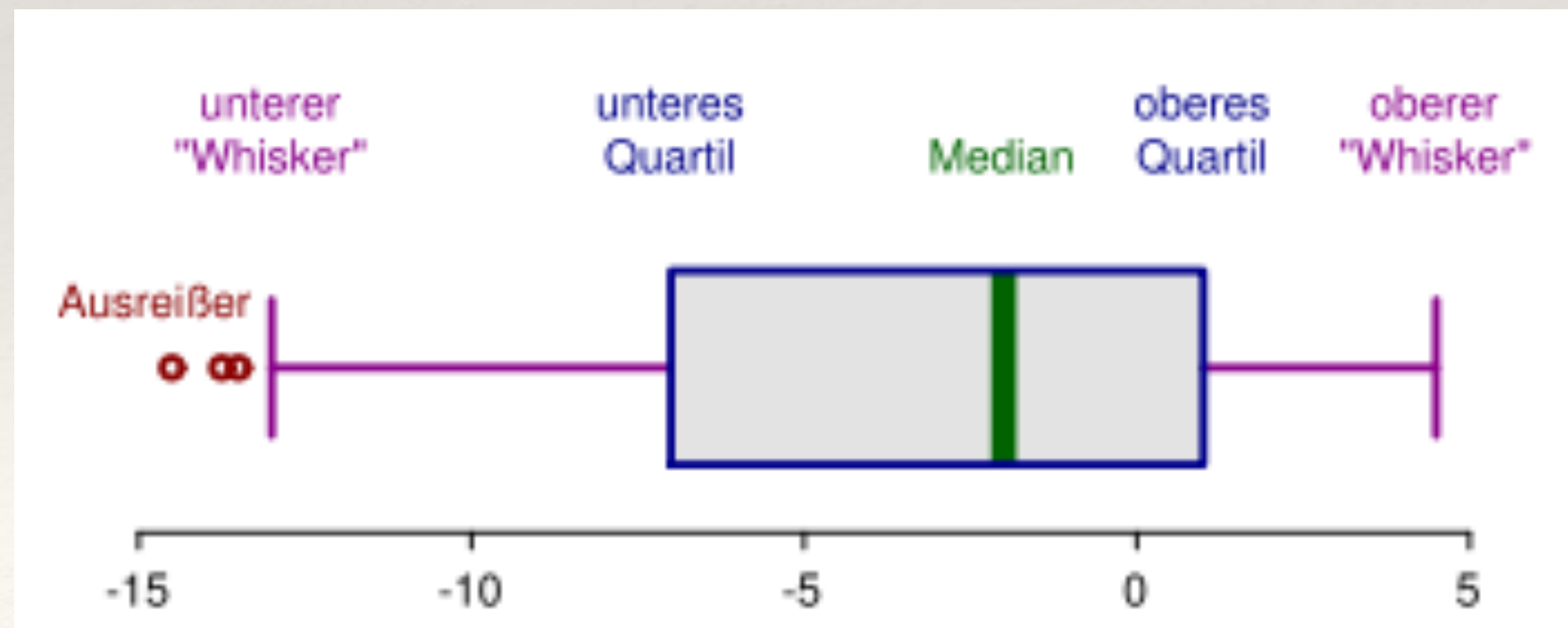


# Algorithmus -1- : Message Filtering

Beispiel

=> The filtering is based on a subject, which is part of each message.

- Der Median, das unteres und oberes Quartil werden bei der Installation gerechnet dadurch wird ein Boxplott gebildet und der Annahmebereich von jedem Zimmer (Bluetooth Detector) bei der Installation genannt und Gespeichert.



erste Schätzung	zweite Schätzung
1	-7
-11	-6
7	-2
2	1
2	11
-4	-5
-4	-6
1	5
9	6
1	-14
11	-6
-1	-5
-3	-12
8	5
13	8
24	3
5	2
30	6
-4	3
-22	10
26	11
30	5
13	2
5	-4
2	6
8	1
6	-10
-9	2

```
9 #include <stdio.h>
10 int main(int argc, const char * argv[]) {
11     // Der Mittelwert
12     int i;
13     int T[10]={5,7,11,9,12,6,9,8,6,13};
14     float mw=T[0];
15     for (i=1;i<10;i++) {
16         mw+=T[i];
17     } // trie du table! /* Déclarations */
18     int l;
19     int j;
20     int AIDE; /* pour la permutation */
21     int FIN; /* position où la dernière permutation a eu lieu. */
22     /* dimension */ /* rang à partir duquel A est trié */ /* indice courant */
23     /* permet de ne pas trier un sous-ensemble déjà trié. */ /* Affichage du tableau */
24     printf("Tableau donné :\n");
25     for (i=0; i<10; i++)
26         printf("%d ", T[i]);
27     printf("\n");
28     printf(" Der Mittelwert beträgt = %.2f \n",mw/10);
29     /* Tri du tableau par propagation de l'élément maximal. */
30     for (I=9; I>0; I=FIN)
31     {
32         FIN=0;
33         for (J=0; J<I; J++)
34             if (T[J]>T[J+1])
35             {
36                 FIN=J;
37                 AIDE=T[J];
38                 T[J]=T[J+1];
39                 T[J+1]=AIDE;
40             }
41     } // Edition du résultat
42     printf("Tableau trié :\n");
43     for (i=0; i<10; i++)
44         printf("%d ", T[i]);
45     printf("\n");
46     // calcule de la Mediane
47     float md=(T[4]+T[5]);
48     printf("la mediane est : %.2f \n",md/2);
49     //ersten Quartil Q1
50     printf("Tableau donné :\n");
51     for (i=0; i<5; i++)
52         printf("%d ", T[i]);
53     printf("\n");
54     printf(" Der ersten Quartil beträgt = %d \n",T[2]);
55     // dritten Quartil Q3
56     printf("Tableau donné :\n");
57     for (i=5; i<10; i++)
58         printf("%d ", T[i]);
59     printf("\n");
60     printf(" Der dritten Quartil beträgt = %d \n",T[7]);
61     // Annahme bereich !
62     printf(" der Annahmebereich liegt zwischen [%d;%d] \n",T[2],T[7]);
```



# Algorithmus -2- : Message Filtering



## Der Erwartungswert einer Zufallsgröße

Bei Wahrscheinlichkeitsverteilungen interessiert man sich oft für den Wert, den die Zufallsgröße im Durchschnitt annehmen kann. Man erhält daher den Mittelwert, indem man die einzelnen Werte der Zufallsgröße mit den Wahrscheinlichkeiten gewichtet und addiert.

## Erwartungswert:

Die Zufallsgröße  $X$  habe die Wertemenge  $\{x_1, x_2, \dots, x_n\}$ . Dann heißt die Zahl  $E(X)$  Erwartungswert.

$$E(X) := \mu := \sum_{i=1}^n x_i \cdot P(X = x_i) = x_1 \cdot P(X = x_1) + x_2 \cdot P(X = x_2) + \dots + x_n \cdot P(X = x_n)$$

```
9 #include <stdio.h>
10 int main(int argc, const char * argv[]) {
11     // Der Mittelwert
12     int i;
13     int T[10]={5,7,11,9,12,6,9,8,6,13};
14     float mw=T[0];
15     for (i=1;i<10;i++) {
16         mw+=T[i];
17     } // trie du table! /* Déclarations */
18     int l;
19     int j;
20     int AIDE; /* pour la permutation */
21     int FIN; /* position où la dernière permutation a eu lieu. */
22     /* dimension */ /* rang à partir duquel A est trié */ /* indice courant */
23     /* permet de ne pas trier un sous-ensemble déjà trié. */ /* Affichage du tableau */
24     printf("Tableau donné :\n");
25     for (i=0; i<10; i++)
26         printf("%d ", T[i]);
27     printf("\n");
28     printf(" Der Mittelwert beträgt = %.2f \n",mw/10);
29     /* Tri du tableau par propagation de l'élément maximal. */
30     for (I=9 ; I>0 ; I=FIN)
31     {
32         FIN=0;
33         for (J=0; J<I; J++)
34             if (T[J]>T[J+1])
35             {
36                 FIN=J;
37                 AIDE=T[J];
38                 T[J]=T[J+1];
39                 T[J+1]=AIDE;
40             }
41     // Edition du résultat
42     printf("Tableau trié :\n");
43     for (i=0; i<10; i++)
44         printf("%d ", T[i]);
45     printf("\n");
46     // calcule de la Mediane
47     float md=(T[4]+T[5]);
48     printf("la mediane est : %.2f \n",md/2);
49     //ersten Quartil Q1
50     printf("Tableau donné :\n");
51     for (i=0; i<5; i++)
52         printf("%d ", T[i]);
53     printf("\n");
54     printf(" Der ersten Quartil beträgt = %d \n",T[2]);
55     // dritten Quartil Q3
56     printf("Tableau donné :\n");
57     for (i=5; i<10; i++)
58         printf("%d ", T[i]);
59     printf("\n");
60     printf(" Der dritten Quartil beträgt = %d \n",T[7]);
61     // Annahme bereich !
62     printf(" der Annahmebereich liegt zwischen [%d;%d] \n",T[2],T[7]);
```



# Algorithmus -2- : Message Filtering



## Varianz und Standardabweichung einer Zufallsgröße

Unter der Varianz versteht man eine Zahl, die angibt, wie stark die einzelnen Werte der Zufallsgröße  $X$  von ihrem Erwartungswert  $E(X)$  abweichen, wie weit die Werte also von  $X$  streuen. Eine andere wörtliche Definition ist die Beschreibung der Varianz als „mittleres Abweichungs –oder Streuungsquadrat.“

## Varianz und Standardabweichung:

Eine Zufallsgröße  $X$  mit  $E(X) = \mu$  und der Wertemenge  $\{x_1, x_2, \dots, x_n\}$  heißt Varianz(„mittleres Abweichungs -oder Streuungsquadrat“) der Zufallsgröße  $X$ .

$$Var(X) := E[(X - \mu)^2] = \sum_{i=1}^n (x_i - \mu)^2 \cdot P(X = x_i)$$

$\sigma(X) := \sqrt{Var(X)}$  heißt Standardabweichung oder Streuung von  $X$ .

```
9 #include <stdio.h>
10 int main(int argc, const char * argv[]) {
11     // Der Mittelwert
12     int i;
13     int T[10]={5,7,11,9,12,6,9,8,6,13};
14     float mw=T[0];
15     for (i=1;i<10;i++) {
16         mw+=T[i];
17     } // trie du table! /* Déclarations */
18     int l;
19     int j;
20     int AIDE; /* pour la permutation */
21     int FIN; /* position où la dernière permutation a eu lieu. */
22     /* dimension */ /* rang à partir duquel A est trié */ /* indice courant */
23     /* permet de ne pas trier un sous-ensemble déjà trié. */ /* Affichage du tableau */
24     printf("Tableau donné :\n");
25     for (i=0; i<10; i++)
26         printf("%d ", T[i]);
27     printf("\n");
28     printf(" Der Mittelwert beträgt = %.2f \n",mw/10);
29     /* Tri du tableau par propagation de l'élément maximal. */
30     for (I=9 ; I>0 ; I=FIN)
31     {
32         FIN=0;
33         for (J=0; J<I; J++)
34             if (T[J]>T[J+1])
35             {
36                 FIN=J;
37                 AIDE=T[J];
38                 T[J]=T[J+1];
39                 T[J+1]=AIDE;
40             }
41     } // Edition du résultat
42     printf("Tableau trié :\n");
43     for (i=0; i<10; i++)
44         printf("%d ", T[i]);
45     printf("\n");
46     // calcule de la Mediane
47     float md=(T[4]+T[5]);
48     printf("la mediane est : %.2f \n",md/2);
49     //ersten Quartil Q1
50     printf("Tableau donné :\n");
51     for (i=0; i<5; i++)
52         printf("%d ", T[i]);
53     printf("\n");
54     printf(" Der ersten Quartil beträgt = %d \n",T[2]);
55     // dritten Quartil Q3
56     printf("Tableau donné :\n");
57     for (i=5; i<10; i++)
58         printf("%d ", T[i]);
59     printf("\n");
60     printf(" Der dritten Quartil beträgt = %d \n",T[7]);
61     // Annahme bereich !
62     printf(" der Annahmebereich liegt zwischen [%d;%d] \n",T[2],T[7]);
```



# Algorithmus -2- : Message Filtering



## Zusammenfassung:

Mit Hilfe von dem Mittelwert / Erwartungswert  $E(X)$  und der empirische Standardabweichung  $\text{Var}(X)$  bekommen wir ein Standardabweichungs-Intervall (s-Intervall)  $[E(X) - \text{Var}(X); E(X) + \text{Var}(X)]$  in der laut Gauß'sche Faustregel ca. 70% der Messwerten liegen.

Gegeben ist eine Urliste  $x_1, x_2, x_3, \dots, x_n$ . Die zugehörigen Kenngrößen sind:

der **Mittelwert**  $\bar{x} = \frac{1}{n}(x_1 + x_2 + x_3 + \dots + x_n)$  und

die empirische **Standardabweichung**  $s = \sqrt{\frac{1}{n}((x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \dots + (x_n - \bar{x})^2)}$ .

Wenn eine relative Häufigkeitsverteilung mit den Werten  $m_1, m_2, m_3, \dots, m_k$  und den relativen Häufigkeiten  $h_1, h_2, h_3, \dots, h_k$  vorliegt, so gilt auch

$\bar{x} \approx m_1 \cdot h_1 + m_2 \cdot h_2 + m_3 \cdot h_3 + \dots + m_k \cdot h_k$  und

$s \approx \sqrt{(m_1 - \bar{x})^2 \cdot h_1 + (m_2 - \bar{x})^2 \cdot h_2 + \dots + (m_k - \bar{x})^2 \cdot h_k}$ .

Bei Messungen charakterisiert die Standardabweichung die Messungenauigkeit.

## Gauß'sche Faustregel (zum Sinn der Standardabweichung)

Nach einer groben **Faustregel**, die C. F. Gauß entdeckte, weichen **bei zufälligen Messfehlern** ca. 68,3% der Messwerte höchstens um die Standardabweichung  $s$  vom Mittelwert  $\bar{x}$  ab, sie liegen also im **Standardabweichungs-Intervall** (s-Intervall)  $[\bar{x} - s; \bar{x} + s]$  mit dem Mittelpunkt  $\bar{x}$ .

```
9 #include <stdio.h>
10 int main(int argc, const char * argv[]) {
11     // Der Mittelwert
12     int i;
13     int T[10]={5,7,11,9,12,6,9,8,6,13};
14     float mw=T[0];
15     for (i=1;i<10;i++) {
16         mw+=T[i];
17     } // trie du tableau! /* Déclarations */
18     int l;
19     int j;
20     int AIDE; /* pour la permutation */
21     int FIN; /* position où la dernière permutation a eu lieu. */
22     /* dimension */ /* rang à partir duquel A est trié */ /* indice courant */
23     /* permet de ne pas trier un sous-ensemble déjà trié. */ /* Affichage du tableau */
24     printf("Tableau donné :\n");
25     for (i=0; i<10; i++)
26         printf("%d ", T[i]);
27     printf("\n");
28     printf(" Der Mittelwert beträgt = %.2f \n",mw/10);
29     /* Tri du tableau par propagation de l'élément maximal. */
30     for (I=9; I>0; I=FIN)
31     {
32         FIN=0;
33         for (J=0; J<I; J++)
34             if (T[J]>T[J+1])
35             {
36                 FIN=J;
37                 AIDE=T[J];
38                 T[J]=T[J+1];
39                 T[J+1]=AIDE;
40             }
41         // Edition du résultat
42         printf("Tableau trié :\n");
43         for (i=0; i<10; i++)
44             printf("%d ", T[i]);
45         printf("\n");
46         // calcule de la Mediane
47         float md=(T[4]+T[5]);
48         printf("la mediane est : %.2f \n",md/2);
49         //ersten Quartil Q1
50         printf("Tableau donné :\n");
51         for (i=0; i<5; i++)
52             printf("%d ", T[i]);
53         printf("\n");
54         printf(" Der ersten Quartil beträgt = %d \n",T[2]);
55         // dritten Quartil Q3
56         printf("Tableau donné :\n");
57         for (i=5; i<10; i++)
58             printf("%d ", T[i]);
59         printf("\n");
60         printf(" Der dritten Quartil beträgt = %d \n",T[7]);
61         // Annahme bereich !
62         printf(" der Annahmebereich liegt zwischen [%d;%d] \n",T[2],T[7]);
```



# Bibliographie & Webographie



- ❖ <https://www.happybubbles.tech/presence/detector>
- ❖ <https://www.happybubbles.tech/presence/beacon>
- ❖ [https://minew.en.alibaba.com/product/60377700306-220649015/Bluetooth\\_nRF51822\\_iBeacon\\_Portable\\_UFO\\_Beacon\\_ibeacon\\_tag.html](https://minew.en.alibaba.com/product/60377700306-220649015/Bluetooth_nRF51822_iBeacon_Portable_UFO_Beacon_ibeacon_tag.html)
- ❖ [https://wholesaler.alibaba.com/product-detail/BLE-Finder-Key-Locator-APP\\_60703721374.html](https://wholesaler.alibaba.com/product-detail/BLE-Finder-Key-Locator-APP_60703721374.html)
- ❖ [https://minew.en.alibaba.com/product/60705123893-220649015/Nordic\\_nRF52832\\_Wide\\_Range\\_Beacon.html?spm=a2700.8304367.rect38f22d.3.76c543e4qUS8Z0](https://minew.en.alibaba.com/product/60705123893-220649015/Nordic_nRF52832_Wide_Range_Beacon.html?spm=a2700.8304367.rect38f22d.3.76c543e4qUS8Z0)
- ❖ [https://minew.en.alibaba.com/product/60704730807-220649015/multi\\_protocol\\_programmable\\_beacon\\_with\\_iBeacon\\_Eddystone.html?spm=a2700.8304367.rect38f22d.9.57d7d268iFy3Qf](https://minew.en.alibaba.com/product/60704730807-220649015/multi_protocol_programmable_beacon_with_iBeacon_Eddystone.html?spm=a2700.8304367.rect38f22d.9.57d7d268iFy3Qf)





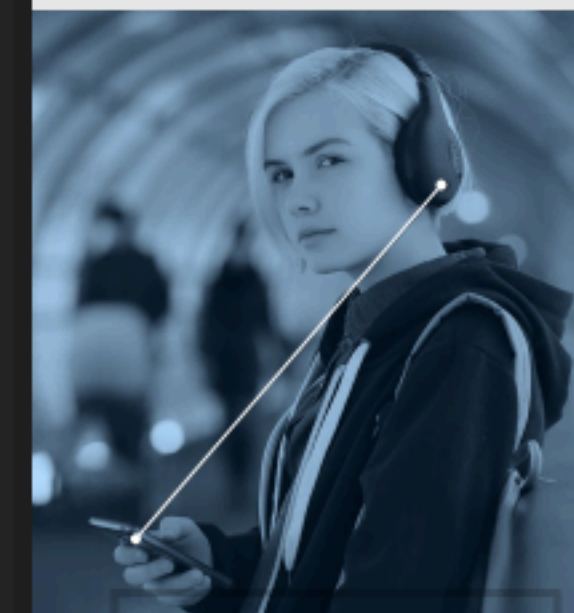
# Danke für Ihre Aufmerksamkeit



the global wireless standard for simple, secure connectivity.

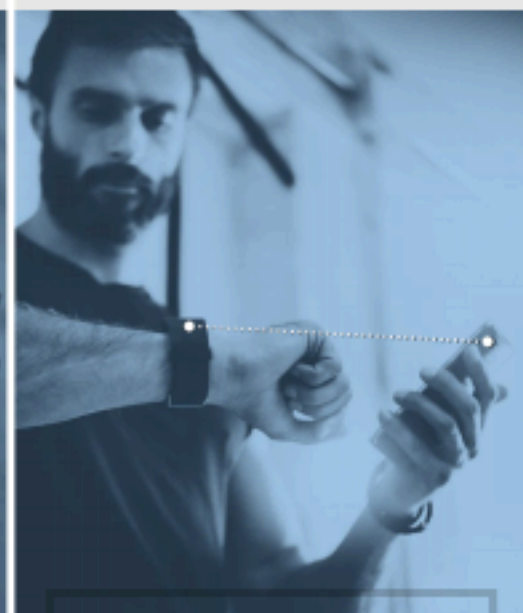
**Bluetooth® Basic Rate/Enhanced Data Rate (BR/EDR)**  
for continuous connections

**Bluetooth® Low Energy (LE)**  
for short burst connections



**point-to-point (1:1)**

**audio streaming**  
· wireless headsets  
· wireless speakers  
· in-car audio



**point-to-point (1:1)**

**data transfer**  
· sports & fitness devices  
· health & wellness devices  
· peripherals & accessories



**broadcast (1:m)**

**localized information**  
· point of interest beacons  
· item finding beacons  
· way finding beacons



**mesh (m:m)**

**large device networks**  
· building automation  
· wireless sensor networks  
· asset tracking

