



Développement mobile avancé

Protocoles de proximité

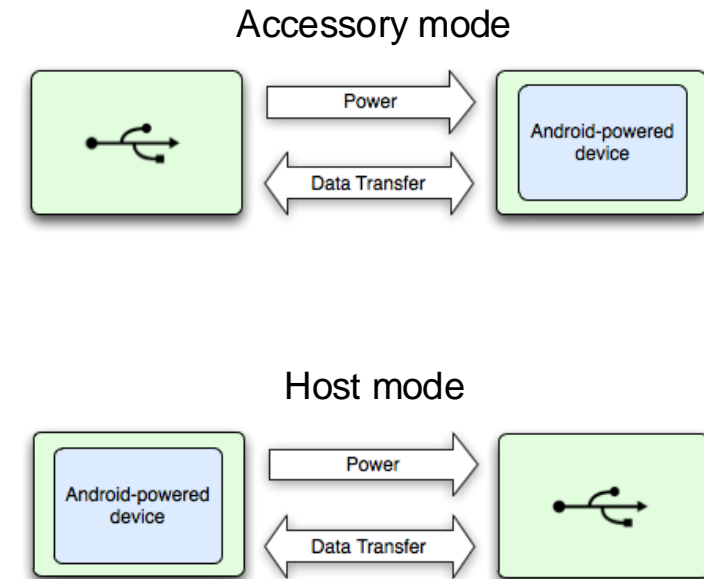
Fabien Dutoit

USB



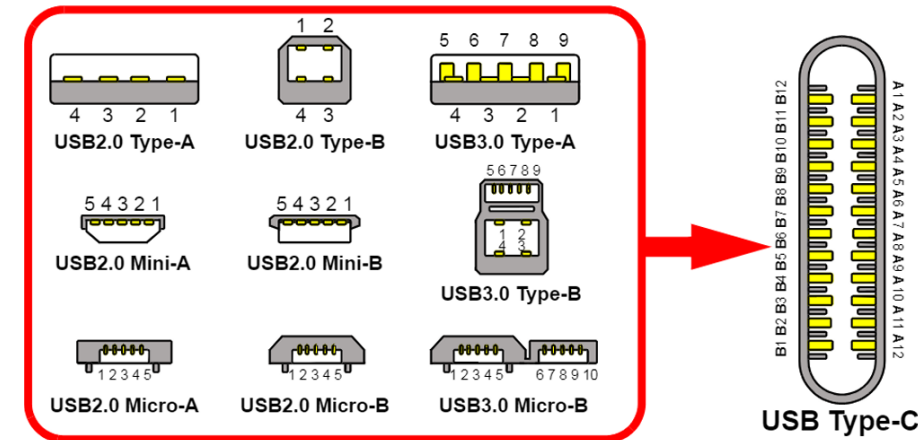
USB

- *USB* est présent sur tous les mobiles récents; il constitue un moyen fiable et efficace de connexion pour des quantités de données volumineuses
- Un bus *USB* possède une direction, on va généralement l'utiliser pour connecter un périphérique à un ordinateur (l'hôte)
 - Initialement les smartphones pouvaient uniquement prendre le rôle de périphérique. Par exemple, lorsque l'on lance une app depuis *Android Studio* sur le smartphone branché par *USB*
 - Depuis quelques années, les smartphones peuvent également prendre le rôle d'hôte auquel on peut brancher des périphériques
 - Sur *Apple*, il existe-ait un adaptateur *Lightning / USB*



USB-C

- L'*USB-C* est un connecteur réversible à 24 pins
- Sur un câble *USB-C*, on peut
 - établir un échange de données (différents standards) et/ou
 - alimenter un périphérique en énergie (également différents standards)
- Rien ne ressemble plus à un câble *USB-C* qu'un autre câble *USB-C*...



5A 60W Fast Charging USB-C to Type-C Cable - Compatible with Samsung, MacBook, iPad & Xiaomi Devices

9.9K+ sold | Provided by (100K+ sold) >

4.8 ★★★★★

CHF3.10 (-61%) 7.99

Spring SUMMER

✓ Free shipping for you

Limited-time offer

68.90

Club 3D Club3D câble USB 4 type C PD 240W / 8K / 40Gbps 3m p/p retail

3 m, USB 4.0

Evaluations

★★★★★ 1

17.90

4smarts USB C – USB C

3 m, USB 2.0

Évaluations

★★★★★ 105

Marque

Plus de produits 4smarts

Câble Thunderbolt 4 (USB-C) Pro (3 m)

CHF 169.60

Longueur

3 mètres

USB-C – Quelques exemples de standards

- Différentes normes *USB*:
 - USB 2.0 - 480 mbps - 2000
 - USB 3.2 Gen 1 (USB 3) - 4 gbps - 2008
 - USB 3.2 Gen 2 (USB 3.1) - 10 gbps - 2013
 - USB 3.2 Gen 2x2 - 20 gbps - 2017
 - USB 4 Gen 3x2 - 40 gbps - 2019
 - USB 4 Gen 4 - 80 gbps - 2022
- *USB Power Delivery* :
 - USB « de base » : $5V \times 0,5A = 2,5\text{ W}$
 - Si le chargeur, l'appareil et le câble le permettent, ils peuvent augmenter l'ampérage et le voltage pour fournir une puissance de charge plus importante, quelques exemples :

• $5V \times 3A = 15\text{ W}$	• $20V \times 5A = 100\text{ W}$
• $15V \times 2A = 30\text{ W}$	• $28V \times 5A = 140\text{ W}$
• $20V \times 3A = 60\text{ W}$	• $48V \times 5A = 240\text{ W}$
- *Thunderbolt* est une autre technologie utilisant le connecteur *USB-C*

USB – *Android*

- Certains périphériques sont supportés par défaut par les OS mobiles :
 - Stockage (clé USB), clavier, souris, carte Ethernet, manette, etc.
 - La compatibilité va varier entre les différents modèles et versions des OS
- Sur *Android*, le *SDK* offre la possibilité d'intégrer à une app un pilote *USB* permettant de prendre en charge un périphérique spécifique, par exemple une caméra :





- Sur *iOS*, la prise en charge d'un accessoire qui n'est pas reconnu nativement nécessite de passer par le *MFi program* :
 - Il s'agit d'un programme *Apple* de certification des accessoires pour *iPhone* et *iPad*
 - L'accès à certaines ressources et spécifications techniques n'est ouvert qu'aux participants à ce programme. Il peut s'agir, par exemple, de l'accès aux *API Bluetooth*, *CarPlay*, *Find My*, etc.
 - Certains périphériques devront intégrer une puce *MFi*
 - *Apple* validera alors l'accessoire: concept, design, certification, etc.
 - NDA: il est difficile de savoir les coûts réels... 4\$ par accessoire vendu ?
- Le connecteur *Lightning* offrait un débit USB 2.0 (480 Mbps, standard d'avril 2000)
- Avec l'*iPhone 15*, Apple débute sa transition vers l'USB 3.2 Gen 2 (10 Gbps)

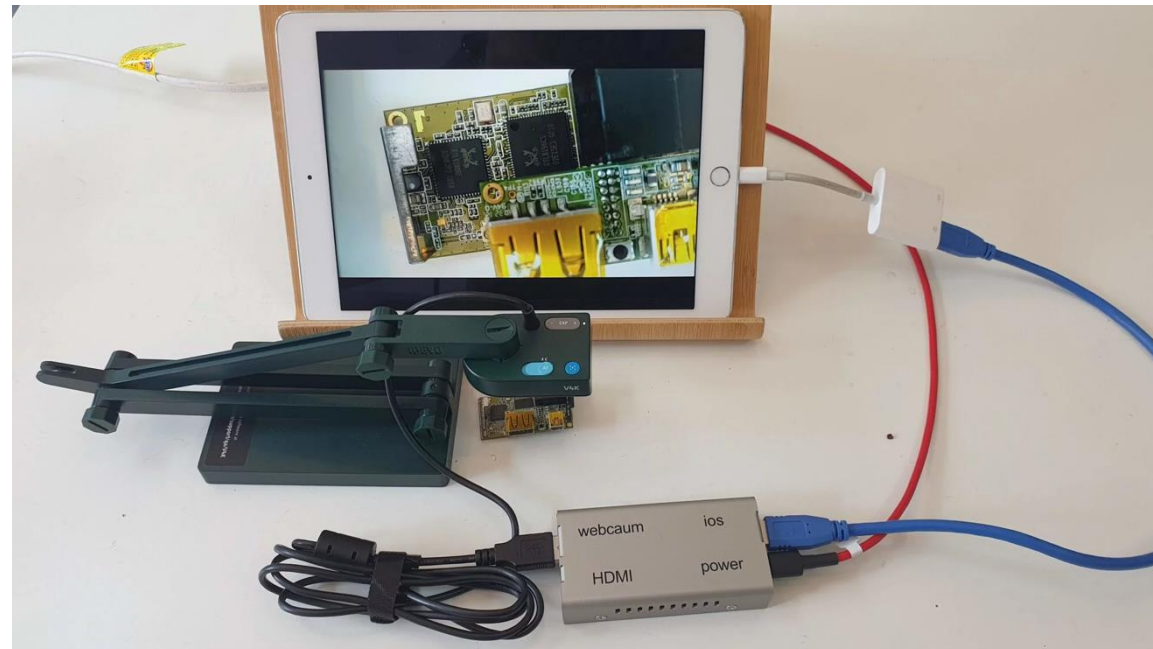
USB – *iOS* – Alternatives

- Les fabricants d'accessoires avaient trouvé une « parade » au programme *MFi*, en utilisant le port *jack* des *iPhones*
- *Apple* a supprimé les ports *jack* en 2016, aujourd'hui les adaptateurs *lightning/jack* nécessitent également une certification *MFi*...



USB – *iOS* – Alternatives

- La reconnaissance des cartes Ethernet sur *iOS* est nativement supportée
- Possibilité de développer ou d'utiliser un protocole réseau
- Exemple pour brancher une webcam, boîtier externe encapsulant le flux de la caméra dans un protocole réseau RTP



Les codes-barres



Qu'est-ce qu'un code-barres ? – Origines

- Représentation de données numériques ou alphanumériques sous la forme d'un symbole constitué de barres claires et foncées. Les données sont encodées dans l'épaisseur et l'espacement de ces barres. Destinés à pouvoir être lus par des machines.



- 1961 Utilisation pour étiqueter des wagons. Identification du matériel roulant lors de son passage
- 1974 Le premier paquet de chewing-gum doté d'un code-barres passe en caisse



Différents types de codes-barres

- Les codes-barres unidimensionnels (1D)



123456

Code-11

[0..9] + "-"



AB12

Code-39

[A-Z] [0-9] [- . \$ / + %]



AB-12/+

Code-93

[A-Z] [0-9] [_ - . \$ / + %]



Eeée

Code-128

ISO-8859-1



7 601234 568903

EAN-13

[0..9] – longueur 12



7 60123 45678 7

UPC-A

[0..9] – longueur 12

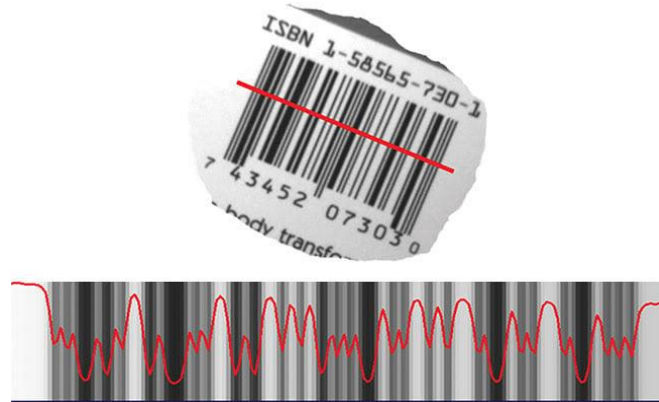


DAFT

[DAFT]

Lecture d'un code-barres 1D

- Historiquement, les lecteurs utilisent un faisceau laser projeté sur le code-barres. L'intensité de la lumière réfléchie va être analysée et permettre son décodage



- Il s'agit de la méthode la plus efficace pour lire des codes-barres 1D
- Les codes-barres 1D possèdent une densité d'information faible :
 - Ils sont bien adaptés pour stocker un identifiant
 - Les applications récentes nécessitent de pouvoir stocker plus d'information

Différents types de codes-barres

- Les codes-barres bidimensionnels (2D)



QR Code



Data Matrix
ASCII



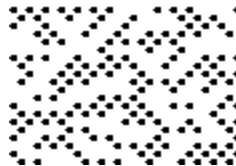
Aztec
ASCII



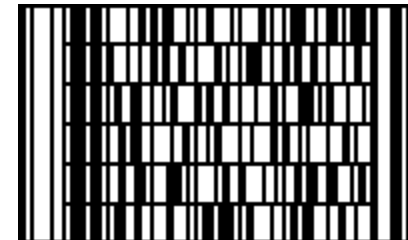
MaxiCode
ASCII



PDF417
ASCII



DotCode
ASCII



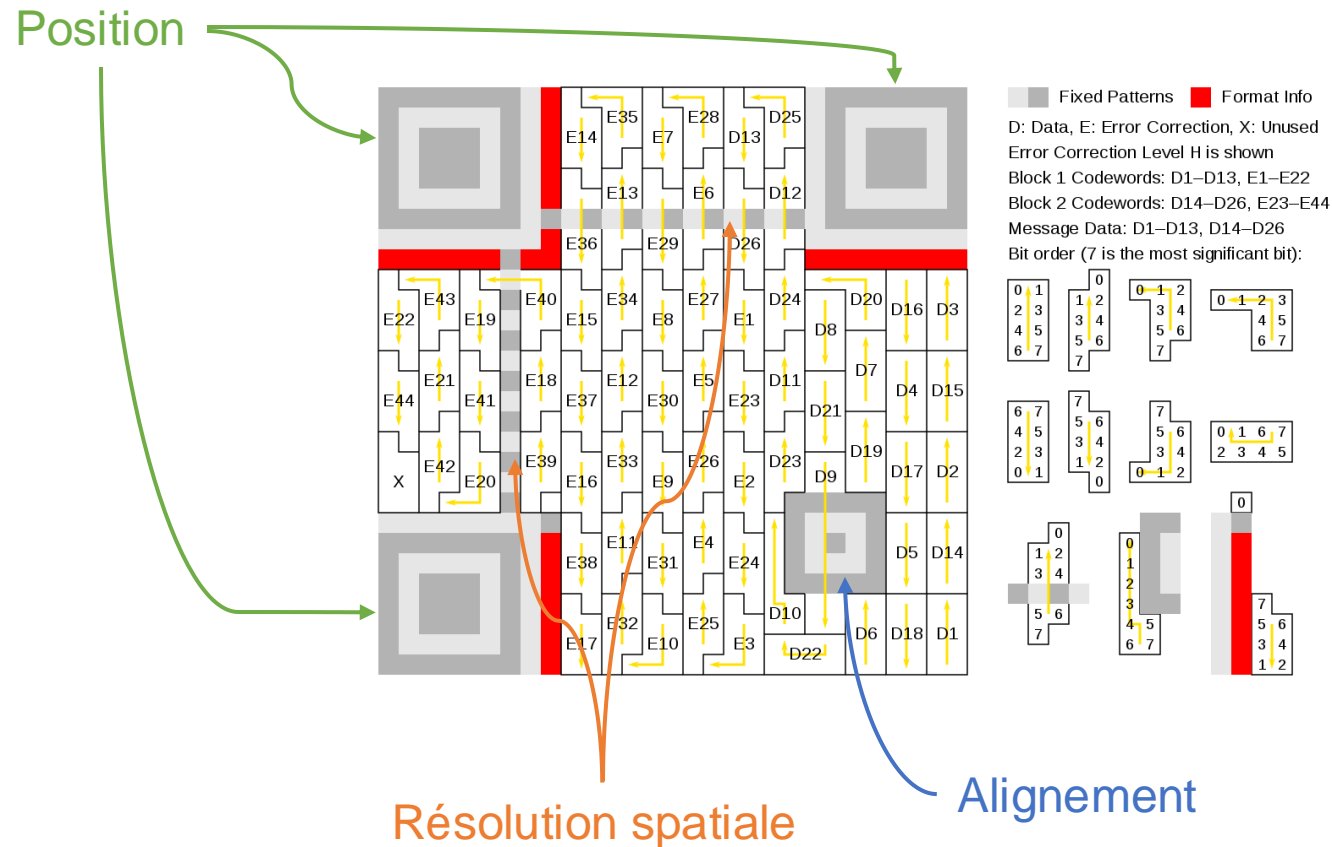
Codablock-F
ASCII



Swiss QR Code

Lecture d'un code-barres 2D

- La lecture des codes-barres 2D reposent principalement sur l'analyse d'une photo du code-barres, par exemple pour un *Code QR* :



Lecture d'un code-barres 2D

- Il est difficile d'avoir une image parfaite du *Code QR* à décoder :
 - Reflets
 - Mise au point
 - Contraste, ...
- Les *Codes QR* intègrent un code correcteur d'erreurs leur permettant d'être décodés, même si une partie est masquée ou illisible, il existe 4 niveaux :
 - L (défaut), permet 7% d'erreurs
 - M, 15%
 - Q, 25%
 - H, 30%



L



M



Q



H

Différentes tailles de Codes QR

- Le Codes QR peuvent contenir une chaîne de caractères de longueur variable
- 4 modes: max. 7'089 chiffres, 4'296 alphanum, 2'953 ISO-8859-1 et 1'817 *Kanji*



Version 1 (21×21)
152 bits (L)



Version 2 (25×25)
272 bits (L)



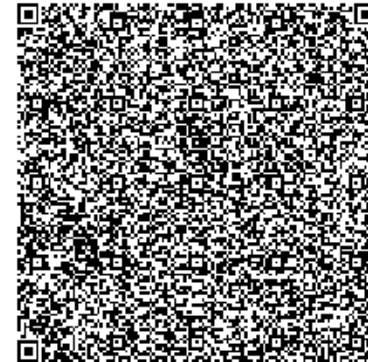
Version 3 (29×29)
440 bits (L)

...



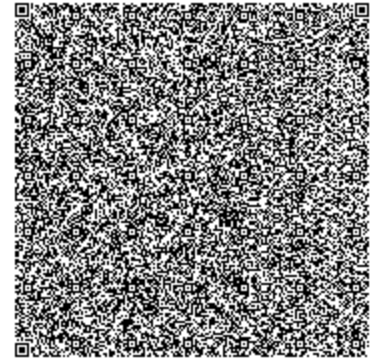
Version 10 (57×57)
2'192 bits (L)

...



Version 25 (117×117)
10'208 bits (L)

...



Version 40 (177×177)
23'648 bits (L)

Différents types de *Codes QR*



tel:+41245577588



<https://heig-vd.ch>



mailto:fabien.dutoit@heig-vd.ch



geo:46.779188,6.6591319?q=HEIG-VD



WIFI:S:MyWifi;T:WPA;P:MyPassword;;



BEGIN:**VEVENT**
SUMMARY:Festival
DTSTART:20230728T160000Z
DTEND:20230728T213000Z
LOCATION:Bord de plage
END:VEVENT



BEGIN:**VCARD**
VERSION:3.0
N:Dutoit Fabien
ORG:HEIG-VD
TEL:+41245577588
EMAIL:fabien.dutoit@heig-vd.ch
ADR:Rte de Cheseaux 1
END:VCARD

Différents types de Codes QR – Texte libre

- L'utilisation du mode texte libre permet de créer des Codes QR répondant à des besoins spécifiques :



```
{
  "n": "Dutoit",
  "f": "Fabien",
  "e": "fabien.dutoit@heig-vd.ch"
}
```



HC1:NCFQ607G0/3WUWGSCLKH47GO0:S4K
QDITFAUO9CK2600XKY-CE59-
G80:84F3TJJM:K2F3*ENK\$4Y50.FK6ZK7:ED
OLOPCO8F6%E3.DA%EOPC1G72A6YM8MG
7GR6E%6.96+F62R62X6SL65:6CB8D*83S8
KB7UPC0JCZ69FVCPD0LV6JD846Y96B463
W5.A6+EDG8F3I80/D6\$CBECUER:C2\$NS3
46\$C2%E9VC- CSUE145GB8JA5B\$D% [...]

Base45

```
{
  "1" : "CH BAG",
  "4" : 1654243238,
  "6" : 1622707238,
  "-260" : {
    "1" : {
      "v" : [...],
      "dob" : "1964-03-14",
      "nam" : {
        "fn" : "Rochat",
        "gn" : "Céline",
        "fnt" : "ROCHAT",
        "gnt" : "CELINE"
      },
      "ver" : "1.2.1"
    }
  }
}
```

Différents types de *Codes QR* – *Exemple GS1*

- D'ici 2027, le *Code QR* devrait avoir remplacé sur tous les produits, les codes-barres *EAN* actuellement utilisés datent des années 70
- Il s'agit d'un format spécifique défini par le *GS1*, l'organisme international responsable de la standardisation de l'utilisation des codes-barres sur les produits du commerce de détail. Les *Codes QR* contiendront :
 - Un *URI* vers une page produit sur le site du producteur (digital link)
 - Le lien pourra être générique ou alors contenir des paramètres additionnel (n° de lot, n° de série)
 - La page pourra informer le consommateur avec, par exemple, des valeurs nutritionnelles
 - C'est aussi un nouvel outil marketing...
 - L'identifiant *GS1* (déjà contenu dans les codes-barres actuels)
 - Des métadonnées directement intégrées dans le *Code QR* :
 - Numéro de série, de lot
 - Date d'expiration
 - Etc...
- Ce format pourra également être utilisé sur des tags *NFC*

Différents types de *Codes QR* – Dynamiques

- Certains services proposent des *Codes QR* dynamiques, ceux-ci permettent :
 - De mettre à jour le contenu ultérieurement. Le *Code QR* contient un *url* permettant d'obtenir le contenu réel au moment de sa lecture
 - De rediriger le client en fonction de son matériel. Par exemple pour le téléchargement d'une app, les clients *Android* sont redirigés vers le *Play Store* et les clients *iOS* vers l'*App Store*
 - D'être plus compacts. Le *Code QR* contient un *shorten url*
 - D'obtenir des statistiques d'utilisation. L'endpoint à l'*url* présent dans le *Code QR* pourra logger toutes les requêtes
- Il faut faire confiance au fournisseur de ce service...
 - Disponibilité dans le temps ?
 - Protection des données ?

Codes QR – Lecture sur mobile

- La lecture de *codes-barres* est généralement possible aujourd'hui directement avec l'application appareil photo, en tous cas pour les *Codes QR*
- Il existe une multitude d'applications sur les stores offrant la lecture de codes-barres, toutes ne se valent pas... La plupart servent uniquement de vecteur de publicité...
- Sur *Android*, il existe principalement 2 librairies permettant d'intégrer la lecture de codes-barres dans son application :
 - *zxing* – librairie *Java* avec une intégration native sur *Android* (licence Apache)
Le projet est actuellement en mode « maintenance »
 - *ML Kit* – librairie de *Google* pour le machine learning, utilisation de modèles entraînés par *Google* pour la reconnaissance des codes-barres (*Google APIs Terms of Service*)

Codes QR – Utilisations sur smartphone



Le smartphone comme lecteur



Génération et affichage d'un *Code-QR*

La librairie *zxing* peut générer des codes-barres que l'on peut afficher sur l'écran du smartphone

Le NFC



NFC – Near Field Communication

- Le *NFC* constitue une classe particulière de *RFID* :
 - Un tag peut être passif, le lecteur va créer un champ EM qui alimentera le tag
 - Lecture à proximité immédiate (max. 3-4 cm), pour limiter la consommation du smartphone
 - *NFC* permet une communication bidirectionnelle
- Le *NFC* est aujourd'hui disponible sur la plupart des smartphones :
 - Sur *Android* quasiment tous les modèles en sont équipés, le *SDK* est très complet
 - Sur *iOS*, les *iPhones* et *Apple Watch* sont équipés depuis plusieurs années, pas les *iPads*...
 - HW disponible depuis 2014, pas d'API pour les apps tierces
 - En 2017, mise à disposition d'une API pour la lecture (*NDEF* uniquement)
 - En 2018, l'OS peut lire automatiquement certains types de tags *NDEF*
 - En 2019, *Apple* étend son API pour permettre l'écriture de tags *NDEF*
 - En 2022, la Commission Européenne accuse *Apple* d'abus de position dominante en limitant l'accès à *NFC* sur ses appareils, en particulier pour les paiements sans contact

NFC – Near Field Communication

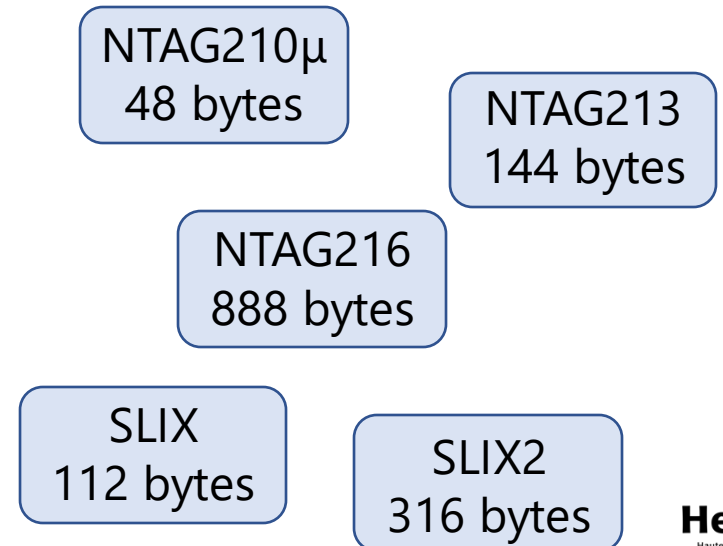
- *NFC* offre trois modes de fonctionnement :
 - **Mode émulation de carte (HCE)**
Le terminal mobile fonctionne comme une carte sans contact. Un lecteur *NFC* externe, tel qu'un terminal de point de vente *NFC*, peut alors accéder à la carte *NFC* émulée
 - **Lecture / Ecriture**
Le mobile équipé du *NFC* est capable de lire et écrire des «tags» passifs. Permet de stocker et de récupérer des données, également pour lancer une action de manière automatique
 - **Mode peer-to-peer**
Ce mode de fonctionnement permet l'échange direct d'informations entre deux appareils équipés de *NFC*



NFC – Lecture / Ecriture de tags

- Il existe plusieurs technologies différentes de *NFC* :
 - NFC-A (ISO 14443A)
 - NFC-B (ISO 14443B)
 - NFC-F (JIS 6319-4)
 - NFC-V (ISO 15693)
 - MifareClassic et MifareUltralight (formats propriétaires)
- Et une multitude de tags, de plusieurs types :

Type	Technologies	Modèles
1	NFC-A	Peu commun – Topaz
2	NFC-A	Ultralight, NTAGX, ST25TN
3	NFC-A	Peu commun – Sony FeliCa
4	NFC-A et NFC-B	Peu commun – DESFire
5	NFC-V	SLI, SLIX, ST25TV



NDEF – *NFC* Data Exchange Format

- *NFC* est une technologie très complète et donc très complexe...
- Heureusement plusieurs niveaux d'abstraction sont proposés, en particulier l'utilisation des messages *NDEF*:
 - Il s'agit d'un format standardisé sous la forme de messages que l'on peut lire et écrire sur la plupart des tags *NFC*, ou les échanger entre 2 smartphones
 - Plusieurs types de messages sont proposés, dont les *Well-Knowns*:
 - URI, action d'ouvrir une communication (http:, tel:, mailto:, etc.)
 - TEXT, stocker du texte, le premier byte indique l'encodage
 - SMARTPOSTER, un *uri* et des métadonnées (titre, logo, etc.)
- En général sur mobile on va essayer de rester sur le plus haut niveau d'abstraction. L'*API* d'*Android* permet de travailler à tous les niveaux, mais cela n'est pas (encore?) possible sur *iOS*
 - *iOS* 18.1 ajoute des possibilités à son *API* pour des cas d'utilisation très spécifiques (paiements *NFC*, billetterie, clés numériques, etc.), uniquement dans quelques pays et pour des développeurs sélectionnés

Utilisation de *NFC* sur *Android*

- Il est nécessaire d'avoir la permission NFC :

```
<uses-permission android:name="android.permission.NFC" />
```

- L'application va ensuite devoir s'inscrire pour être notifiée lors de la lecture d'un *TAG NFC* via un *intent-filter*, il existe deux possibilités :

- Dans le *Manifest* :

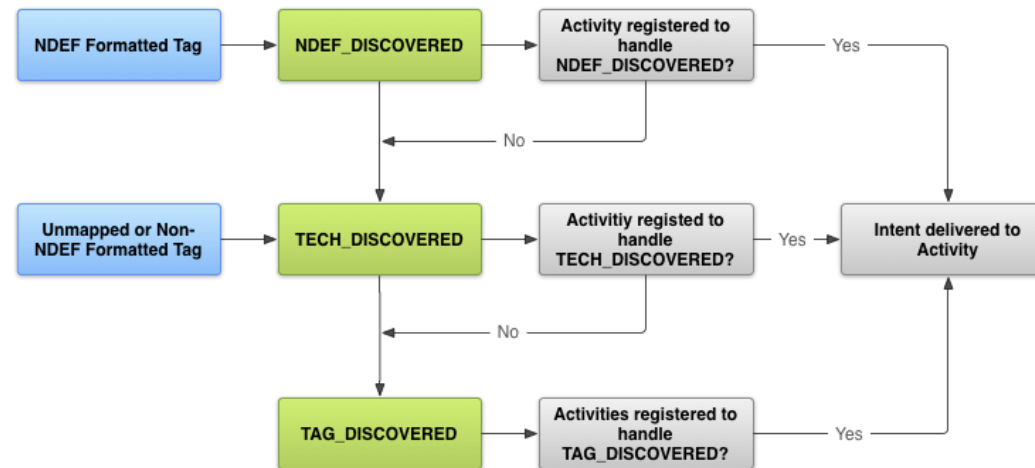
```
<activity  
    android:name=".MainActivity"  
    android:exported="true">  
    [...]  
    <intent-filter>  
        <action android:name="android.nfc.action.NDEF_DISCOVERED" />  
        <category android:name="android.intent.category.DEFAULT" />  
        [...]  
    </intent-filter>  
</activity>
```

Le système ouvrira notre
Activité lors du scan d'un
TAG NFC

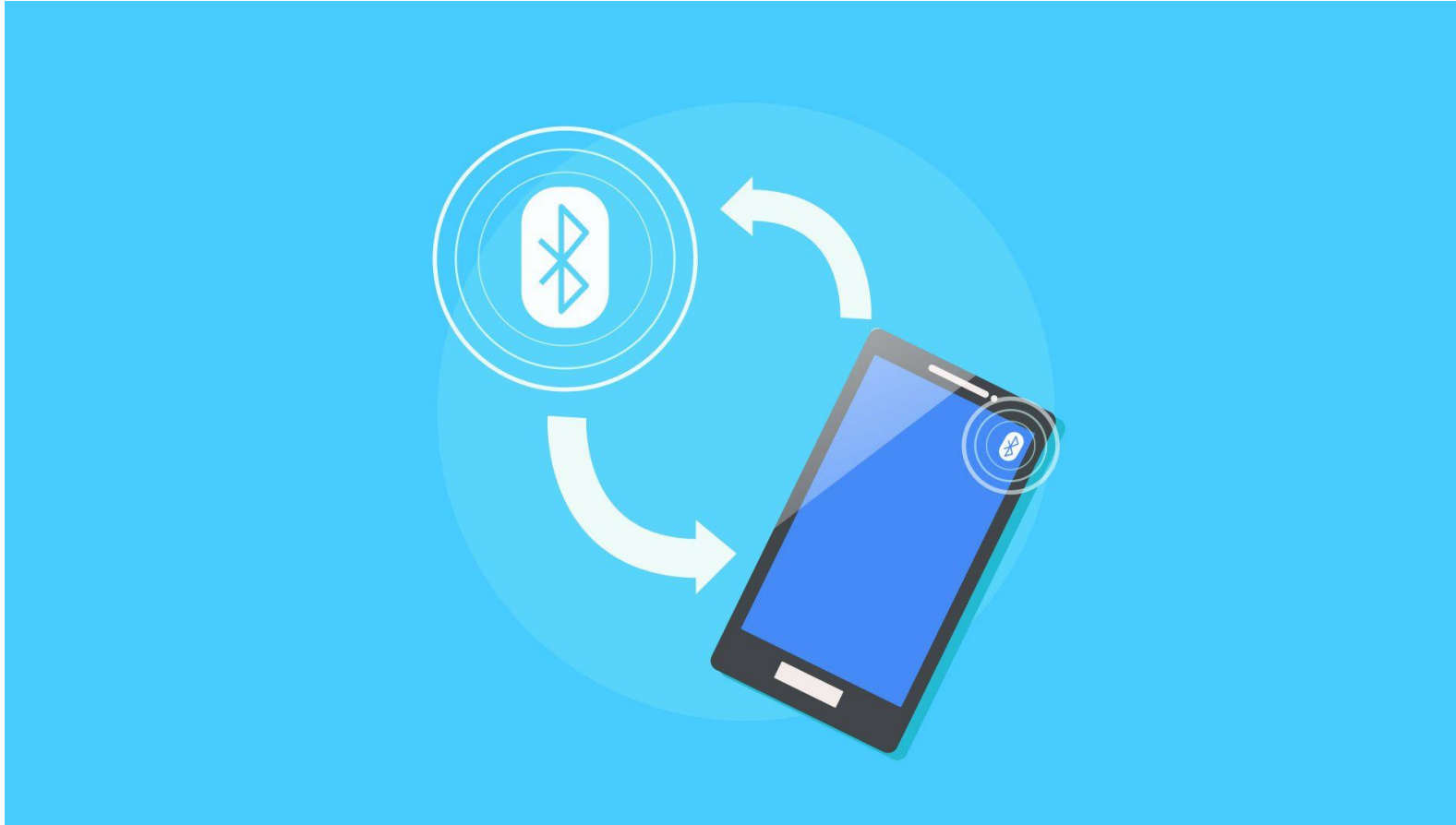
- Au runtime, notre *Activité* va s'enregistrer dans *onResume* et se désenregistrer dans *onPause* pour être notifiée uniquement lorsqu'elle est active (*ForegroundDispatch*)

Utilisation de *NFC* sur *Android*

- Un *Intent* sera transmis à l'activité pour l'informer de la présence d'un *TAG* correspondant à l'*intent-filter* enregistré
- 3 niveaux d'abstraction possibles :
 - ACTION_NDEF_DISCOVERED Lecture d'un TAG avec un payload au format *NDEF*
 - ACTION_TECH_DISCOVERED Lecture d'un TAG de certaines technologies (A, B, F, V, etc.)
 - ACTION_TAG_DISCOVERED Lecture de n'importe quel TAG NFC



Le Bluetooth



Le *Bluetooth* « Classique »

- Il s'agit d'un protocole de proximité développé à la fin des années 1990, initialement destiné à offrir une alternative sans-fil à la connectique *USB*
 - Connexion d'un périphérique (imprimante, clavier, ...) à un hôte (ordinateur). Les deux appareils, une fois appairés, disposeront d'une connexion bidirectionnelle chiffrée
 - Comme il était destiné à une utilisation « fixe », sa portée (env. 10 m.) et sa consommation en font un protocole peu adapté à la mobilité
 - Evolution de la norme :
 - 1994-98 Création de *Bluetooth* et mise en place du consortium d'entreprises
 - 1999 *Bluetooth* 1.0, jusqu'à ~721 kbits
 - 2006 *Bluetooth* 2.0, *EDR* (Enhanced Data Rate), jusqu'à ~2,1 Mbits
 - 2009 *Bluetooth* 3.0, *HS* (High Speed), peu d'utilisation
 - 2010 *Bluetooth* 4.0, définition du *Bluetooth Low Energy BLE*
 - 2016 (5.0), 2019 (5.1 et 5.2), 2023 (5.3), principalement des nouveautés pour *BLE*
 - 2024 *Bluetooth* 6.0, *Channel Sounding* (localisation), améliorations diverses

Le *Bluetooth* « Classique »

- Le *Bluetooth* «Classique» fonctionne sur la base de profils (spécification fonctionnelle pour un cas d'utilisation), quelques exemples fréquents :
 - *Advanced Audio Distribution Profile (A2DP)*
Streaming de musique vers un casque, un haut-parleur, l'autoradio, etc.
 - *Headset Profile (HSP) - Hands free Profile (HFP)*
Utilisés par les kits mains-libre, par exemple dans les voitures
 - *Phone Book Access Profile (PBAP, PBA)*
Généralement combiné à l'HSP pour accéder aux contacts
 - *Audio/Video Remote Control Profile (AVRCP)*
Permet le contrôle des télévisions et appareils multimédia compatibles
 - *Personal Area Network (PAN)*
Partage de connexion Internet
 - *Human Interface Device (HID) - HID over GATT Profile (HOGP)*
Claviers, souris, manettes

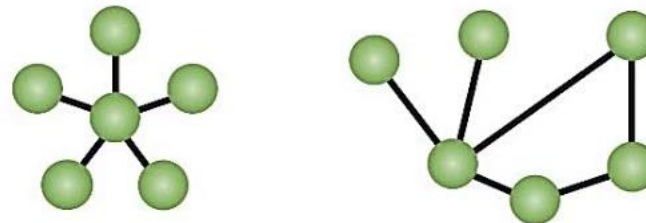


Le *Bluetooth* « Classique »

- La liste des profils supportés nativement peut varier d'un appareil à un autre, d'une version de l'OS à l'autre, etc.
 - Par exemple *iOS* gère nativement :
 - *Hands-Free Profile (HFP 1.7)*
 - *Phone Book Access Profile (PBAP 1.2)*
 - *Advanced Audio Distribution Profile (A2DP 1.3)*
 - *Audio/Video Remote Control Profile (AVRCP 1.6)*
 - *Personal Area Network (PAN) Profile*
 - *Human Interface Device (HID) Profile*
 - *Message Access Profile (MAP 1.1)*
 - *Braille*
- La prise en charge d'un périphérique «custom» nécessite bien souvent l'ajout de nouveaux profils sur le mobile. Bien que complexe, c'est possible sur *Android*. Sur *iOS* en revanche, cela nécessite de passer par le *MFi Program*...

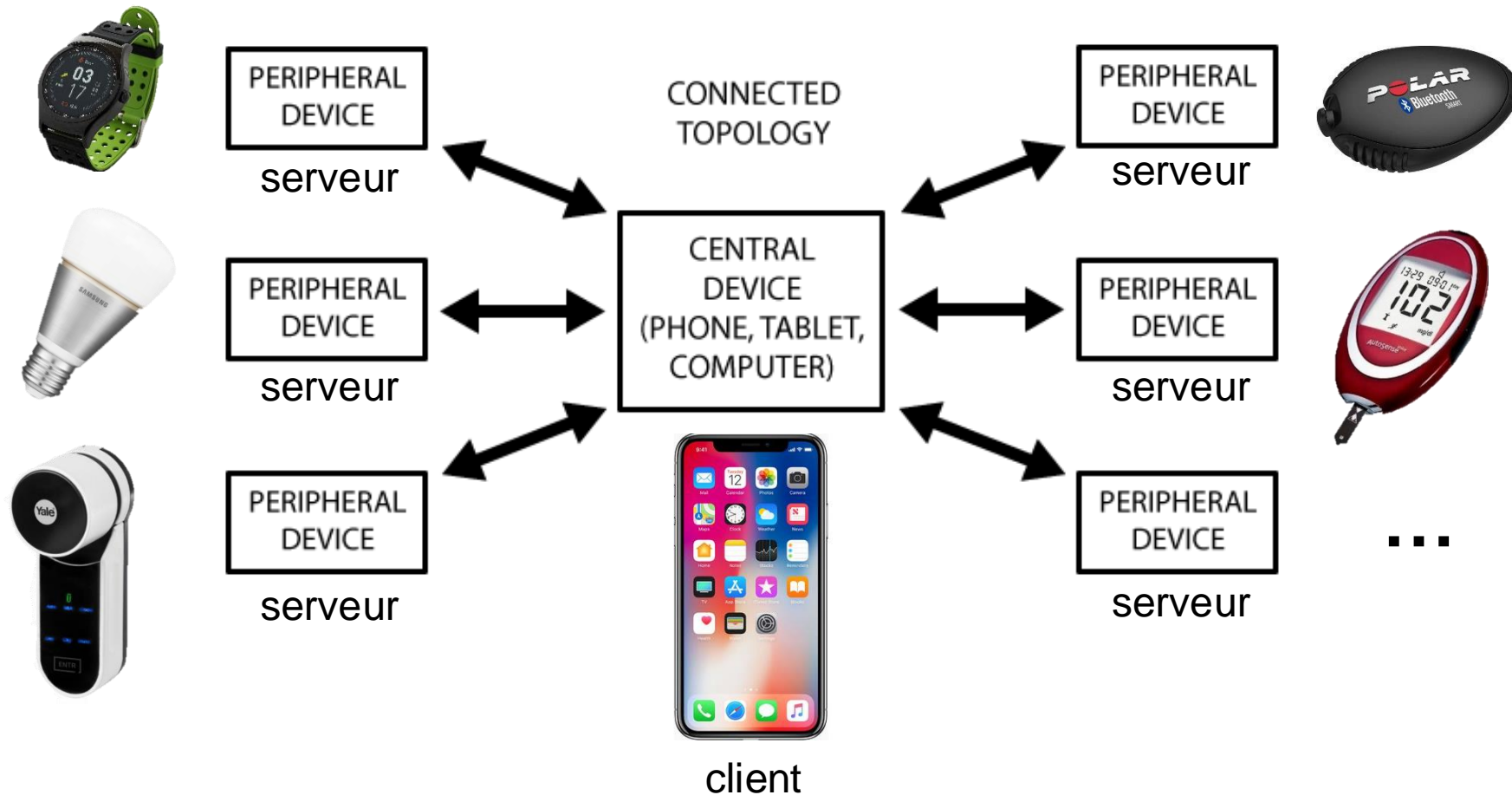
Le *Bluetooth Low Energy* (BLE)

- Le *Bluetooth Low Energy* est une technologie cousine au *Bluetooth* «*Classique*», mais totalement indépendante :
 - En dépit du nom en commun, elles ne sont pas compatibles
 - Aujourd'hui, la plupart des smartphones permettent de travailler avec les deux technologies
 - Les puces actuelles sont généralement *dual-mode* et peuvent partager la même antenne
- Comme son nom l'indique, le *BLE* est destiné à permettre en communication de proximité en utilisant peu d'énergie :
 - La portée varie entre 5 et 100 m.
 - Le débit théorique s'approche de 1 Mbits, mais au maximum **env. 100 kbits utile**
 - Destiné à de petits périphériques pouvant fonctionner quelques mois sur une pile bouton
 - La version 5.0 (2016) permet de \pm doubler la portée ou le débit. Elle introduit également les réseaux maillés (mesh networks)



Le *Bluetooth Low Energy* (BLE)

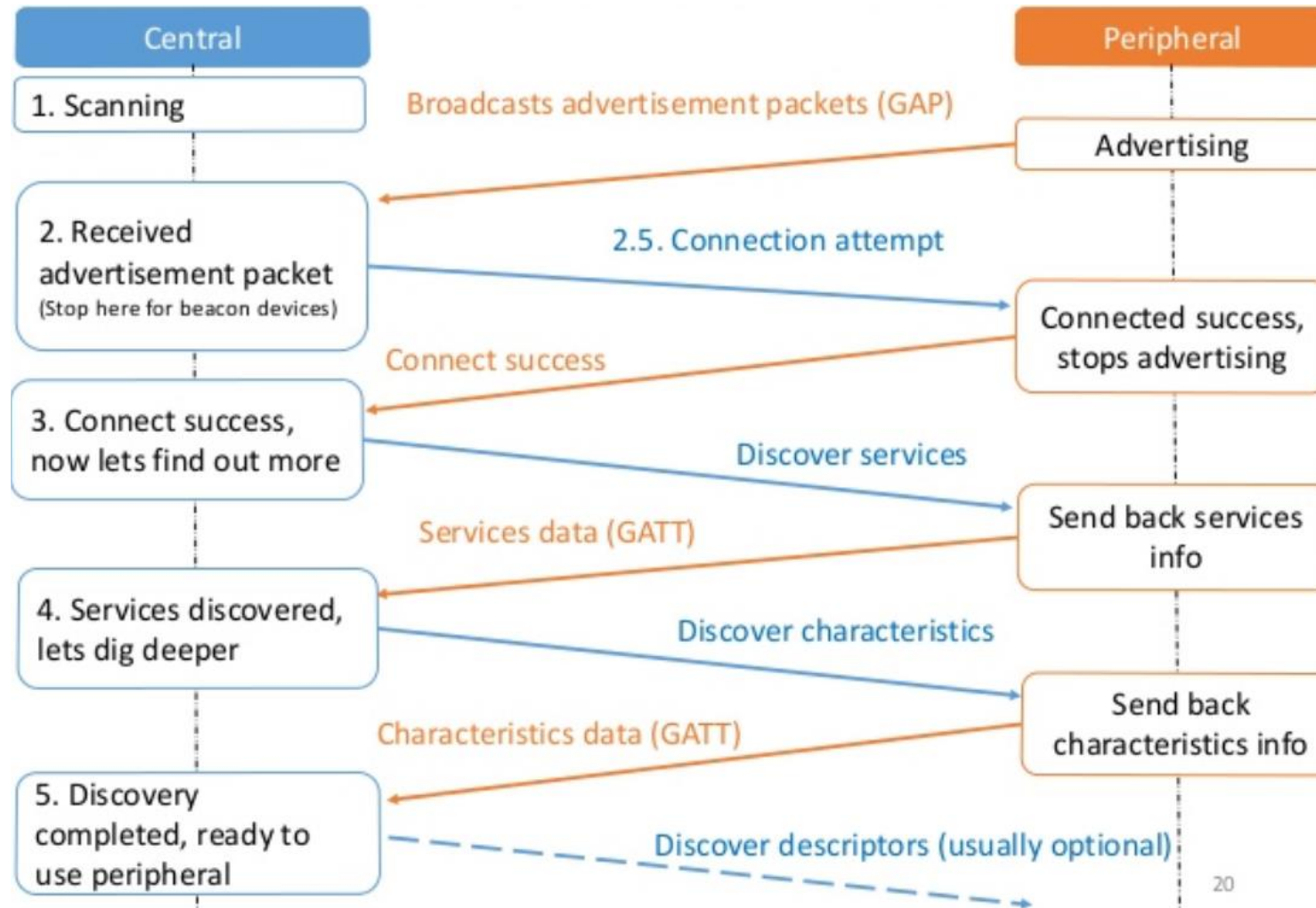
- Le *BLE* est organisé sur une topologie *client / serveur*



Le *Bluetooth Low Energy* (BLE)

- Le *BLE* repose sur deux *phases* :
 - **Generic Access Profile (GAP)** - avant la connexion
Diffusion d'informations sur le périphérique
 - Nom du périphérique
 - Peut-on s'y connecter ?
 - Liste des services disponibles
 - **Generic Attribute Profile (GATT)** - une fois connecté
Informations sur l'échange de données
 - Définition des services, des caractéristiques et des descripteurs

Le *Bluetooth Low Energy* (BLE)



Le *Bluetooth Low Energy* (BLE)

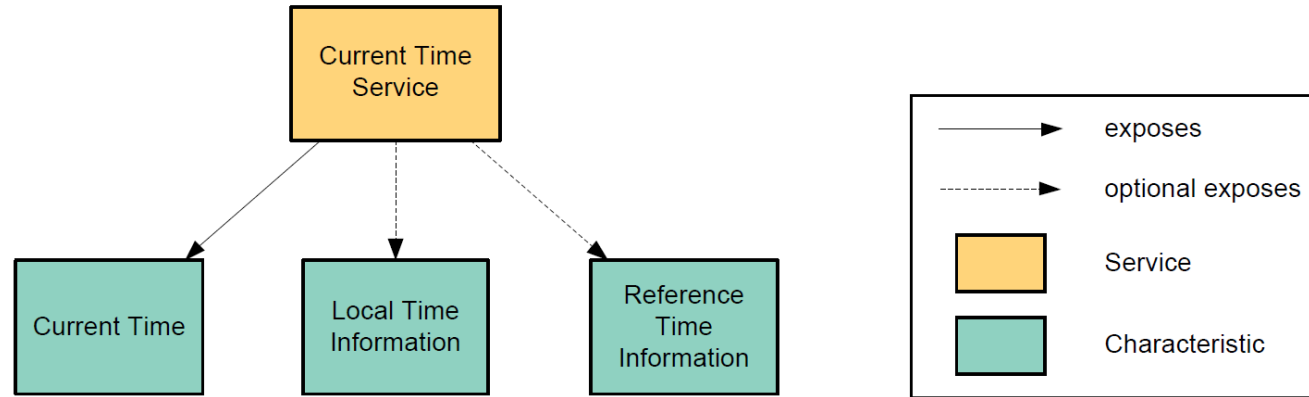
- Le *BLE* introduit la notion de *Service*, il s'agit d'un ensemble de *Characteristics* (des variables) qui fonctionnent ensembles pour offrir une certaine fonctionnalité
- Le consortium *Bluetooth* a défini un certain nombre de *Services* standards, quelques exemples :
 - *Battery Service (0x180F)*
 - *Current Time Service (0x1805)*
 - *Volume Control Service (0x1844)*
 - *Blood Pressure Service (0x1810)*
 - *Continuous Glucose Monitoring Service (0x181F)*
 - *Heart Rate Service (0x180D)*
- Les *Services standards* disposent d'un identifiant court (*uuid* sur 16 bits)
0x1805 → 0000**1805**-0000-1000-8000-00805f9b34fb l'*uuid* de 128 bits est implicite
- Les *Services propriétaires* utilisent des *uuid* de 128 bits

Le *Bluetooth Low Energy* (BLE)

- Les périphériques BLE peuvent communiquer selon 2 modes :
 - **Connecté** on peut se connecter et échanger **en clair** :
 - Les périphériques publient les services qu'ils proposent, tout appareil à proximité peut s'y connecter
 - **Appairé / Bondé** les deux périphériques peuvent communiquer sur un canal **chiffré** après échange des clés, plusieurs approches pour l'échange des clés :
 - Just WorksTM : «automatique», vulnérable MITM
 - Out of Band : second canal de communication: *NFC*, *Wi-Fi*, ...
 - Passkey : affichage et saisie d'un code *PIN* → clavier + écran
 - Numeric comparison : 2 écrans, uniquement depuis BLE 4.2+

Le *Bluetooth Low Energy* (BLE)

- Un *Service* est composé d'une ou plusieurs *Characteristics* (variables) :



- Chaque *Characteristic* met à disposition certaines opérations :

	Broadcast	Read	Write without Response	Write	Notify	Indicate	Signed Write	Reliable Write	Writable Auxiliaries
Current Time	X	M	X	O	M	X	X	X	X
Local Time Information	X	M	X	O	X	X	X	X	X
Reference Time Information	X	M	X	X	X	X	X	X	X

M mandatory
O optional
X not permitted

Le *Bluetooth Low Energy* (BLE)

- Une *Characteristic* :

- Représente une valeur (int, float, string, format binaire custom, etc.), généralement limitée à 20 bytes (valeur par défaut correspondant à l'échange d'un paquet unique), max: 512
- Propose plusieurs opérations (propriétés), les principales étant :

- Lecture

Le *Central* peut lire la valeur présente sur le périphérique

- Ecriture (avec ou sans confirmation)

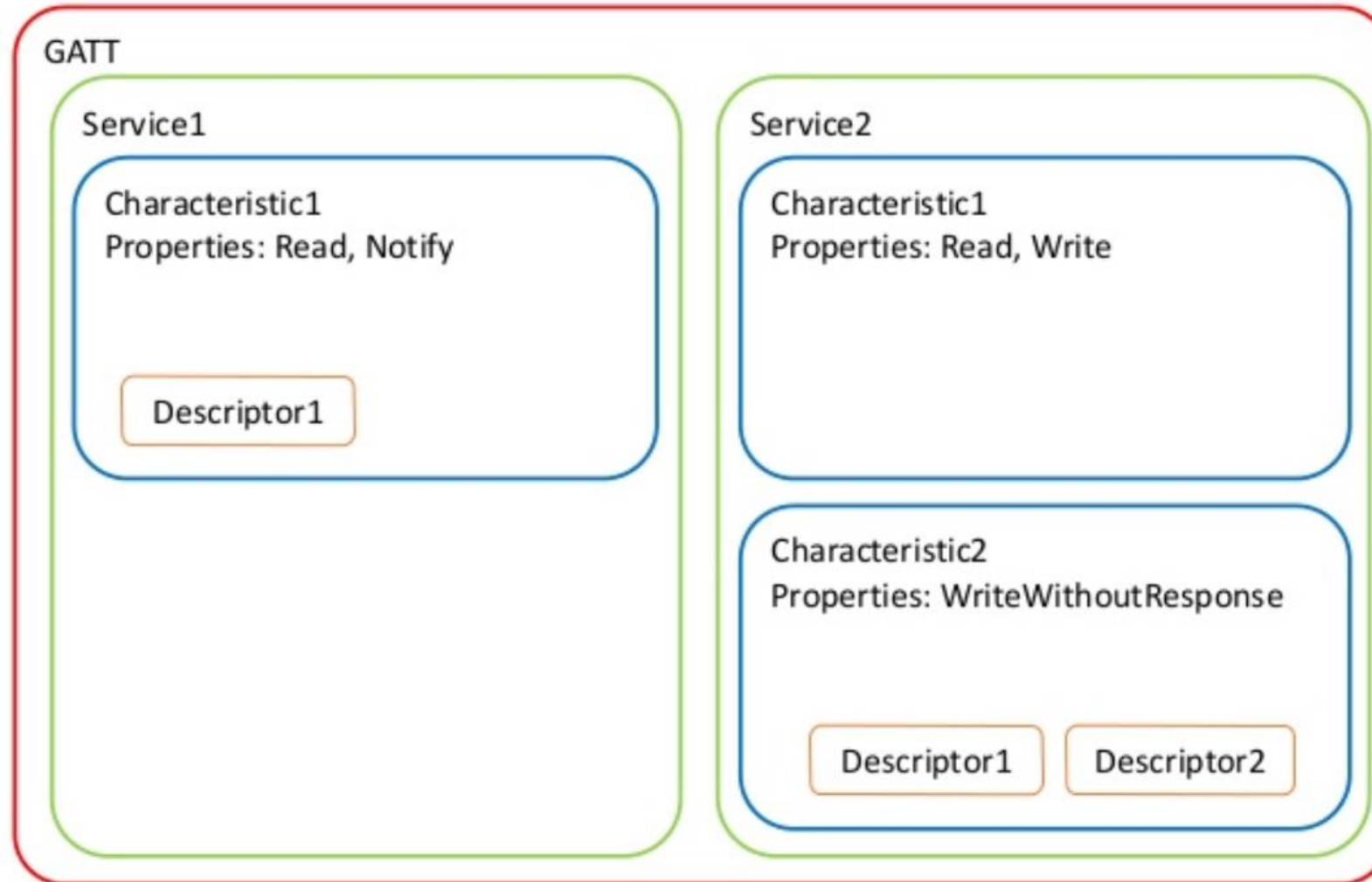
Le *Central* peut écrire une valeur sur le périphérique

- Indication / Notification

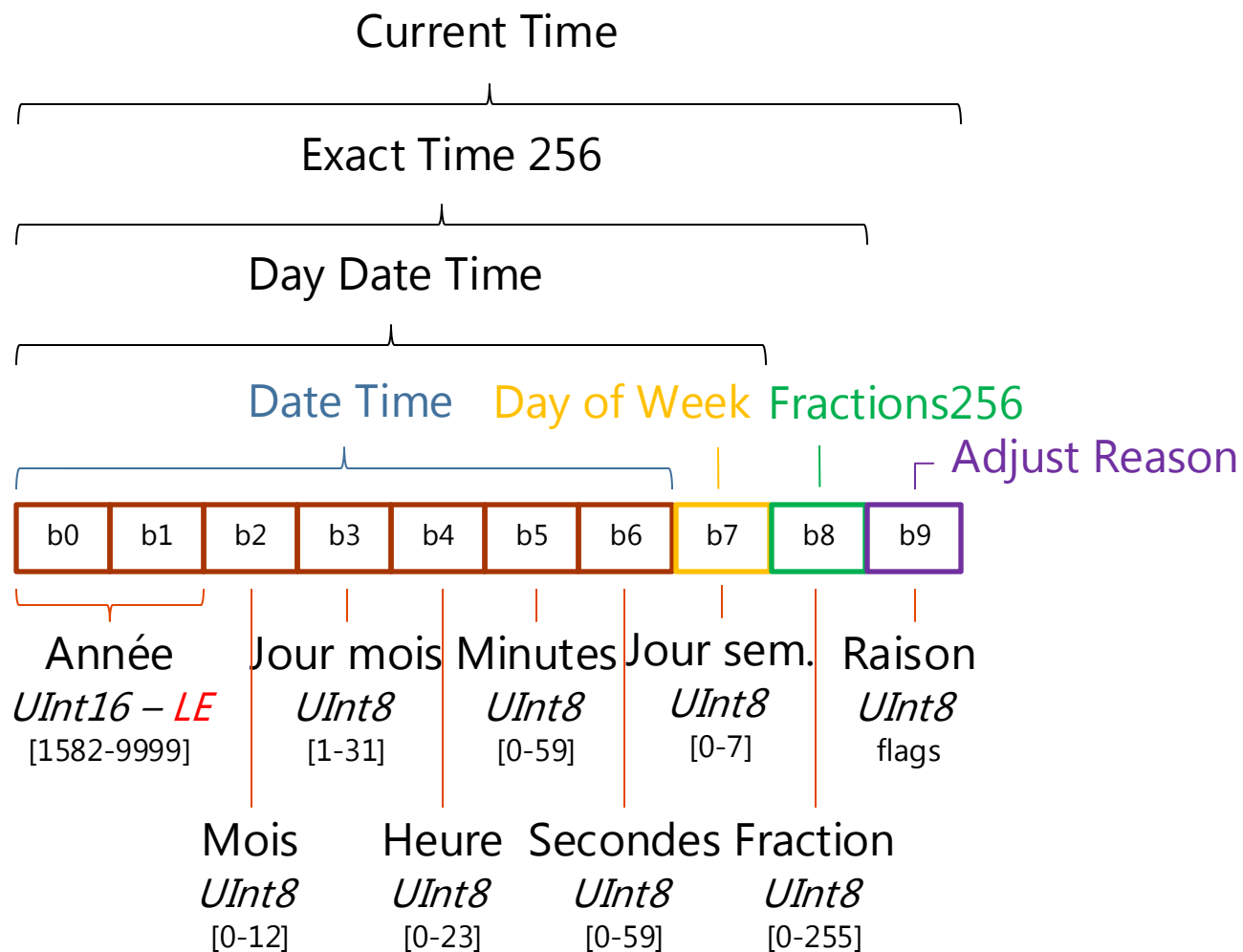
Le *Périphérique* notifie le *Central* lorsque la valeur change, avec (indication) ou sans (notification) confirmation par le *Central*. Le *Central* doit préalablement s'inscrire pour être notifié

- Peut contenir zéro ou plusieurs *Descriptors* :
 - Des attributs (métadonnées) décrivant la *Characteristic*

Le *Bluetooth Low Energy* (BLE)



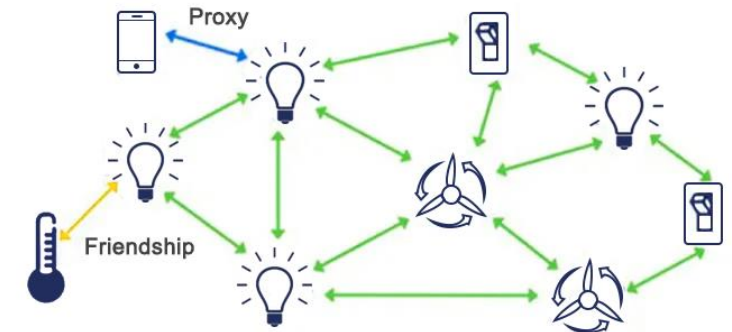
Le *Bluetooth Low Energy* (BLE) – Exemple



Format de la *Characteristic Current Time* du *Current Time Service*

Le *Bluetooth Low Energy* (BLE) – Evolutions

- Bluetooth 5.0 (2016)
 - augmentation du débit *2M PHY* (jusqu'à 1,4 Mbits) ou augmentation *Coded PHY* de la portée (jusqu'à 200 m.)
 - introduction des réseaux maillés
- Bluetooth 5.1 (2019)
 - Calcul de l'angle d'arrivée ou de départ (cf. slides *Localisation*)
- Bluetooth 5.2 (2020)
 - Création d'un profil audio BLE destiné à remplacer A2DP
- Bluetooth 5.3 (2021)
 - Optimisations
- Bluetooth 5.4 (2023)
 - Chiffrement possible des annonces



LE Audio is on Bluetooth Low Energy

As the names suggest, Classic Audio operates on the Bluetooth Classic radio while LE Audio operates on the Bluetooth Low Energy radio. LE Audio will not only support development of the same audio products and use cases as Classic Audio, it will introduce exciting new features that promise to improve their performance as well as enable the creation of new products and use cases.

Autres



Nearby

Google Nearby



Apple AirDrop

Autres

- **Wi-Fi direct** Possibilité de créer une connexion *p2p* par *Wi-Fi* entre deux smartphones. Intéressant pour des échanges importants de données
- **Google Nearby** Librairie proposée par *Google* pour *Android* pour offrir une API de communication *p2p* avancée. Utilisation du *Bluetooth*, du *Wi-Fi* et de l'*audio (ultrasons)*. Très intéressante pour échanger en direct entre deux ou plusieurs smartphones
- **AirDrop** Fonctionnalité offerte par *Apple* sur *MacOS*, *iOS* et *ipadOS* permettant le partage de fichiers entre appareils à proximité. Utilisation du *Bluetooth* et du *Wi-Fi*

**HE^{VD}
IG**

**HAUTE ÉCOLE
D'INGÉNIERIE
ET DE GESTION
DU CANTON
DE VAUD**



Développement mobile avancé

Les capteurs et les wearables

Fabien Dutoit

Les Capteurs

