Szegedi Tudományegyetem  
Programtervező informatika Bsc.

Szakdolgozat

EEG szenzor kezelése mobiltelefonon és felhőben

(EEG sensor management on smartphone and cloud)

Készítette:  
Hegedűs Valentin

Témavezető:  
Dr. Bilicki Vilmos

Szeged  
2020

Tartalomjegyzék

[Bevezető 4](#_Toc39763307)

[1.1. Motiváció 4](#_Toc39763308)

[Betekintés az elektroenkefalográfiába 4](#_Toc39763309)

[1.1. Mi az EEG? 4](#_Toc39763310)

[1.2. Hogyan működik az EEG? 4](#_Toc39763311)

[1.3. Agyhullámok típusai 5](#_Toc39763312)

[1.4.1. Bétahullám 5](#_Toc39763313)

[1.4.2. Alfahullám 6](#_Toc39763314)

[1.4.3. Thétahullám 6](#_Toc39763315)

[1.4.4. Deltahullám 6](#_Toc39763316)

[1.3. Az Emotiv EPOC+ készülék 6](#_Toc39763317)

[Meglévő EEG adatot feldolgozó alkalmazások 8](#_Toc39763318)

[2.1. EEGLab 8](#_Toc39763319)

[2.2. OpenBCI 9](#_Toc39763320)

[2.3. FieldTrip 9](#_Toc39763321)

[2.4. BrainVision Analyzer 10](#_Toc39763322)

[Felhasznált technológiák 11](#_Toc39763323)

[Ionic 11](#_Toc39763324)

[Firebase 11](#_Toc39763325)

[MNE library 12](#_Toc39763326)

[A Jellyfish alkalmazás 13](#_Toc39763327)

[3.1. Felhasználó autentikáció 14](#_Toc39763328)

[3.1. EEG készülékek keresése és a csatlakozás folyamata 15](#_Toc39763329)

[3.2. Felvétel beállítások 16](#_Toc39763330)

[3.3. Felvétel rögzítése 17](#_Toc39763331)

[3.4. Mentett felvételek 18](#_Toc39763332)

[3.5. Felvétel megtekintése 19](#_Toc39763333)

[Adatok kinyerése és feldolgozása 21](#_Toc39763334)

[4.1. Mi a szenzor adat? 21](#_Toc39763335)

[4.2. Adatfolyam, adatcsomagok 21](#_Toc39763336)

[4.3. Bluetooth LE – Az EEG, mint GATT szerver 22](#_Toc39763337)

[4.4. Keresés és csatlakozás a készülékre 23](#_Toc39763338)

[4.4.1. A scannelés folyamata 23](#_Toc39763339)

[4.4.1. A csatlakozás folyamata 24](#_Toc39763340)

[4.5. Adatfolyam indítása és kezelése 25](#_Toc39763341)

[4.6. Az adatok dekódolása 26](#_Toc39763342)

[4.6.1. Advanced Encryption Standard (AES) 27](#_Toc39763343)

[4.6.2. AES kulcsgenerálás 27](#_Toc39763344)

[4.6.3. Dekódolás 28](#_Toc39763345)

[4.6.4. Konvertálás 28](#_Toc39763346)

[Felhőalapú számítástechnika 30](#_Toc39763347)

[5.1. Mi a felhő? 30](#_Toc39763348)

[5.2. Felhőszolgáltatások típusai 31](#_Toc39763349)

[5.2.1. Infrastructure as a Service (IaaS) 31](#_Toc39763350)

[5.2.2. Software as a Service (SaaS) 31](#_Toc39763351)

[5.2.3. Kiszolgáló nélküli számítástechnika 31](#_Toc39763352)

[5.2.4. Platform as a Service (PaaS) 32](#_Toc39763353)

[5.3. Firebase Cloud Firestore 32](#_Toc39763354)

[5.3.1. NoSQL vagy relációs adatbázis 32](#_Toc39763355)

[5.3.2. Firestore adattárolás 33](#_Toc39763356)

[Firebase triggerek 33](#_Toc39763357)

[Google Cloud feldolgozás és MNE 33](#_Toc39763358)

[Felhasznált irodalom, források 35](#_Toc39763359)

# Bevezető

## Motiváció

Az emberi agy mindig is foglalkoztatta a kutatókat, ámbár összetettsége miatt igen bonyolult annak működését megérteni. Napjainkban számos precíz eszköz és módszer áll rendelkezésre a neurológiai területek kutatására, melyek pontosabb összképet adnak az agy felépítéséről. Az interneten rengeteg tanulmány található erről a kutatási területről, azonban jelenleg is folynak e téren a vizsgálatok, hogy mélyremenően megismerjük az agyunk működését.

A szakdolgozat céljául egy mobilalkalmazást fejlesztettem, mellyel EEG méréseket lehet felvenni, vizualizálni és feldolgozni az adatokat. A továbbiakban egy betekintőt adok az agyi folyamatok vizsgálatára szolgáló alkalmazásokba és eszközökbe, majd az általam készített programot és annak működését mutatom be, kezdve az alkalmazás használatától, egészen a rendszer működéséért felelős folyamatokig.

# Betekintés az elektroenkefalográfiába

## Mi az EEG?

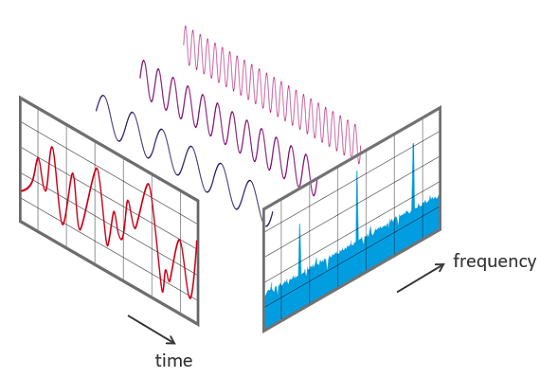
Az elektroenkefalográfia (EEG) tágabb értelemben véve egy pszichofiziológiai mérőeljárás, melynek segítségével a pszichés működés élettani hátterét vizsgálhatjuk meg. Szűkebb értelemben egy elektrofiziológiai mérőeszköz, mely az idegsejtek elektromos aktivitásának mérésére szolgál. Az EEG regisztrálása során az analóg görbéket digitális jellé alakítják át, így matematikai elemzéssel az EEG-görbét különböző frekvenciájú komponensekre bonthatják.

## Hogyan működik az EEG?

Az agy milliárdnyi sejtjei apró elektromos jeleket generálnak, úgynevezett agyhullámokat. Az agykéregi neuronok aktivitása elektromos feszültség ingadozást indukál. EEG mérés során a készülék szenzorjai nem a különálló neuronok által generált feszültséget mérik, hanem az adott agyi területen lévő neuroncsoportok összességét. A szenzorok egyetlen másodperc alatt, akár ezernyi mintavételt is végezhetnek. Az EEG eszköz, a mért jeleket felerősíti és digitalizálja, majd tovább küldi egy feldolgozó egységhez, mint például egy számítógép.

## Agyhullámok típusai

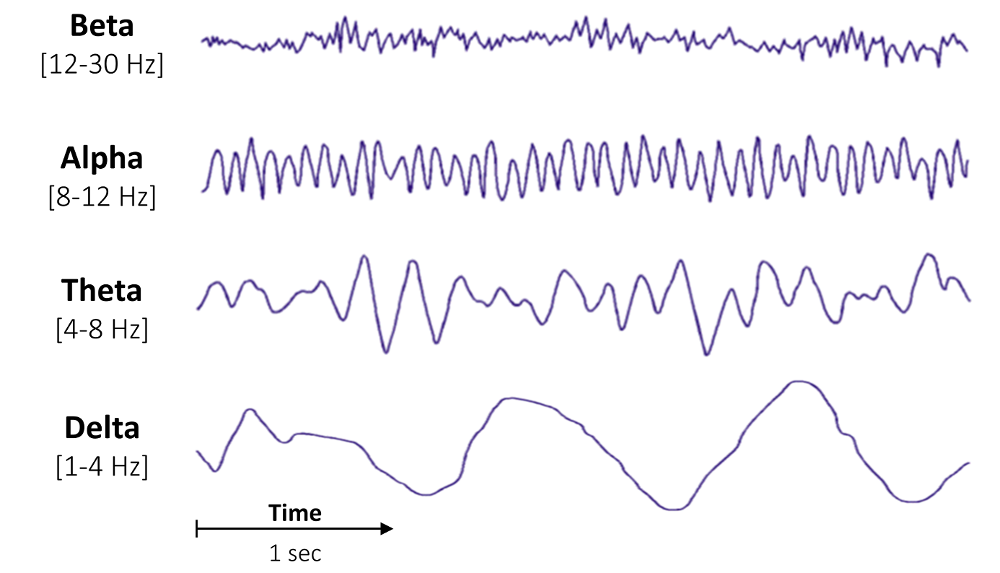
Az EEG készülék elektródái, a mért elektromos aktivitást különböző frekvenciákban fejezik ki. Az úgynevezett Fast Fourier transzformáció (FFT) segítségével lehetőség nyílik az időtartományon (time-domain) értelmezett nyers jeleket, frekvenciatartományba (frequency-domain) átalakítani. Ezt a folyamatot úgy lehet egyszerűen elképzelni, hogy az FFT algoritmus, az inputjában kapott agyhullámot különböző frekvenciatartományokra bontja szét, vagyis megmutatja, hogy a jel milyen komponensekből épül fel. Ezáltal az agyhullámokat egy másik formában tudjuk megvizsgálni.



A Fast Fourier transzformáción átesett agyhullámokat frekvenciatartományuk és előfordulásuk szerint 4 fő típusba sorolják: alfahullám, bétahullám, thétahullám és deltahullám.

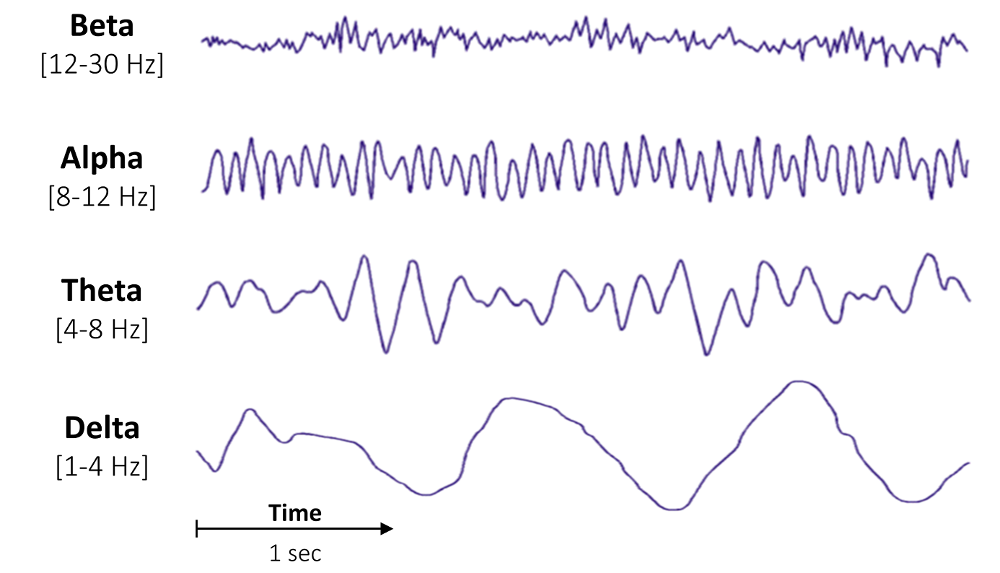
### 1.4.1. Bétahullám

A bétahullám leginkább a tudatos, éber és figyelmi állapotra jellemző, mely kis amplitúdójú, gyors potenciálként figyelhető meg. Az aktív koncentráció vagy a szorongó lelkiállapot bétahullámot generál. Ezt a hullámtípust gyakran a motoros működéshez is társítják. Frekvenciatartománya 12, illetve 30Hz közé tehető.



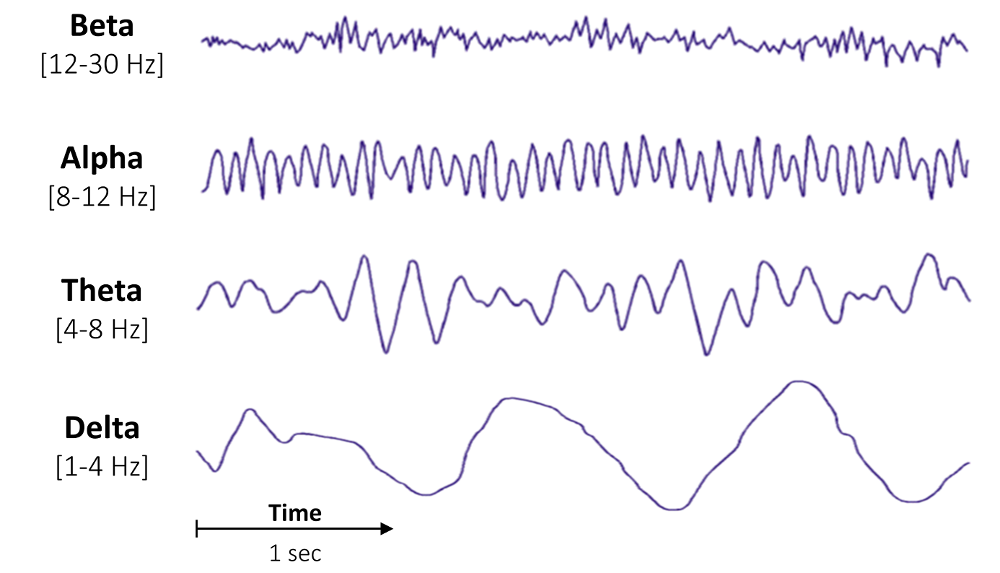
### 1.4.2. Alfahullám

Az alfahullámok gyakran összekapcsolódnak egy nyugodt tudatállapottal. Ezeket a hullámokat a szem bezárása és a pihenés indukálja, azonban előfordulhat intenzív állapotban is. A legtöbb felnőttnél az alfahullámok frekvenciája 8 és 12Hz közötti intervallumban található.



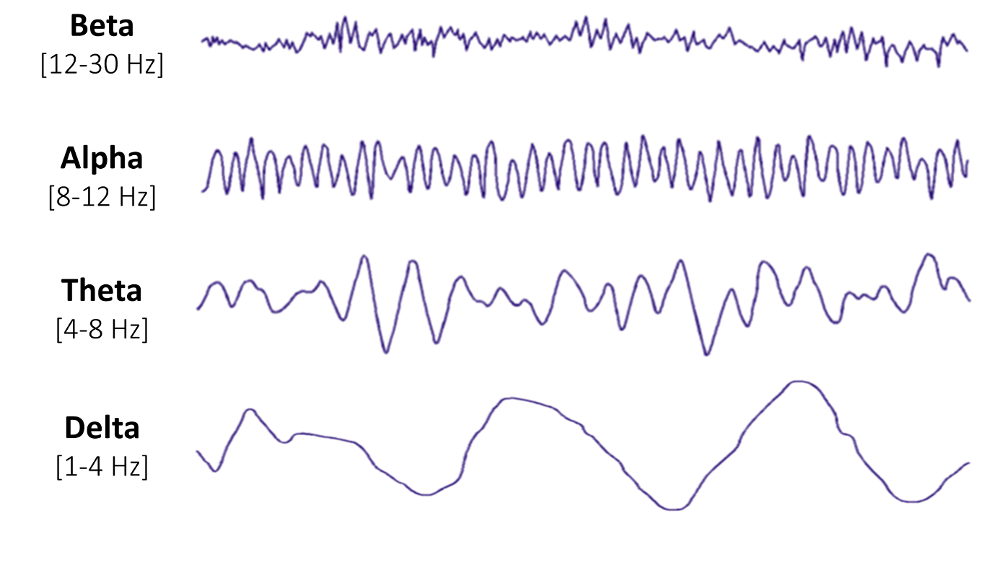
### 1.4.3. Thétahullám

Az agyiaktivitást 4 és 8Hz frekvenciatartományban thétahullámnak nevezzük. Frekvenciája alapján az alfa és delta hullámok közötti intervallumba esik. Felnőttek esetén fáradt állapotban figyelhető meg, gyerekeknél a serdülőkor idejére jellemző a mintázata.



### Deltahullám

A delta-aktivitás elsősorban a gyermekek éber agyi tevékenysége vizsgálata során figyelhető meg. Idősebb személyeknél mély alvás során jelentkezik, mérsékelt agyitevékenység mellett. A deltahullámokat alacsony frekvenciájú és nagy amplitúdójú hullámok jellemzik, melyek frekvenciatartománya körülbelül 1 és 4Hz között mozog.



## 1.3. Az Emotiv EPOC+ készülék

Az Emotiv EPOC+ egy vezetéknélküli EGG készülék, mely az agyi aktivitások mérésére szolgál. Az EEG által mért adatok Bluetooth kommunikációval, illetve USB kábellel érhetők el, így okostelefonon és asztali számítógéppel is feldolgozhatók. Az eszközön 14 elektróda található, melyek úgy lettek kialakítva, hogy a fejet leginkább lefedjék. Ezek a szenzorok a nemzetközi 10-20-as szabvány alapján lettek elhelyezve, ami szerint négy anatómiai referenciapont[[1]](#footnote-1) van kijelölve a fejen és e referenciapontok között az elektródák 10, illetve 20%-os távolsággal helyezkednek el. A készülékben gyorsulásmérő szenzor is helyett kapott, így a mért agyi aktivitás adatok mellett, a fej mozgását is képes feldolgozni az eszköz. Alapértelmezetten a mintavétel 128Hz-en történik, azonban lehetőség van ennek átállítására.



Az általam fejlesztett alkalmazás EPOC+ készülékkel lett tesztelve és ehhez lett optimalizálva, azonban működése szempontjából más Emotiv eszközzel is komplatibilis lehet.

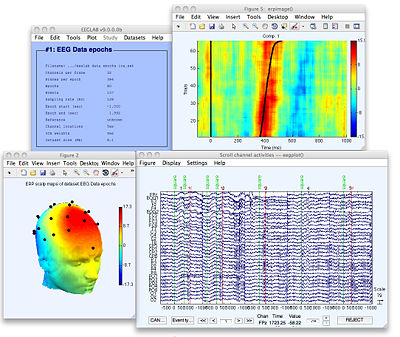
# Meglévő EEG adatot feldolgozó alkalmazások

Jelenleg az interneten számos EEG adatot feldolgozó szoftver található, ezek közül rengeteg open-source, amit a fejlesztők mindenki számára elérhetővé tettek, hogy a közösséggel együtt hatékonyabb alkalmazásokat készítsenek. Olyan alkalmazások is fellelhetők, amik zárt forráskódúak. Ezek programkódjait a fejlesztők nem hozták nyilvánosságra, ezzel védve a saját termékük értékét. Ezen alkalmazások mind hasonló módon működnek és hasonló algoritmusokat implementáltak, azonban használhatóságuk szempontjából és másként kialakított grafikus felületük eltérő lehet.

## EEGLab

Az EEGLab egy Matlab[[2]](#footnote-2) eszközkészlet, amely folyamatos és esemény-alapú EEG, MEG és egyéb elektrofiziológiai adatok feldolgozását teszi lehetővé. Egyaránt Windows, Mac Os és Linux operációs rendszeren is működik. Az EEGLab egy interaktív grafikus (2. ábra) felhasználói felületet biztosít, hogy a nagy mennyiségű adatot hatékonyan fel tudják dolgozni a felhasználók.

Minden alapszintű adatfeldolgozó algoritmus mellett, független komponens-elemzés (ICA - Independent component analysis), idő-frekvencia elemzés (TFA – time-frequency analysis), és más egyéb adat vizualizációs ezközt is implementáltak. A kutatócsoportok és fejlesztők számára az EEGLab kiterjeszthető nyílt forráskódú platformmá, amelyen keresztül új módszereket oszthatnak meg egymással, azáltal, hogy közzéteszik a fejlesztett plugin funkciókat, amik automatikusan megjelennek a másik felhasználó számára.



## OpenBCI

Az OpenBCI (Open-source Brain-Computer Interface) egy nyílt forráskódú platform, mely segítségével a biológia elektromos jelek aktivitásának feldolgozására van lehetőségünk.

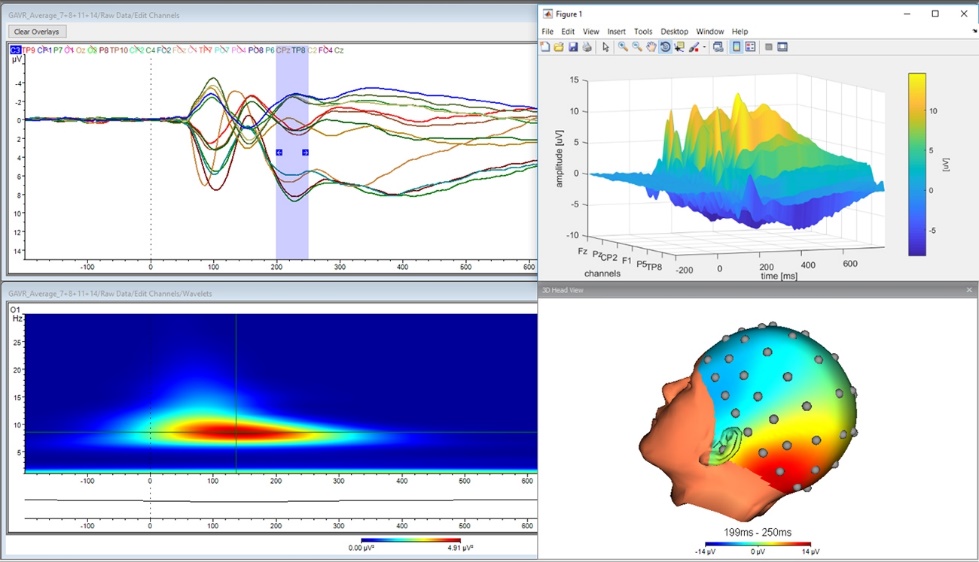
Ez az alkalmazás Joel Murphy és Conor Russomanno mérnökök által létrehozott Kickstarter sikeres kampánya végett készült el, mely lehetővé teszi mindenki számára az agyi kutatásokat. A sokoldalú platform az elektromos agyi aktivitás (EEG), izomaktivitás (EMG), pulzusszám (EKG) és sok más egyéb testi jel mérésére szolgál. Az alkalmazásban rengeteg algoritmust implementáltak, ilyen például az FFT (Fast Fourier Transform) vagy a másik gyakran használt eljárást a Filtering-et (low-pass és high-pass). A programon felül a fejlesztők olcsó, de jó minőségű hadware-eket is készítenek. Ilyenek az Arduino[[3]](#footnote-3) kompatibilis alaplapok, amelyek nagy felbontású EEG és EMG jelek felvételét teszik lehetővé.

## FieldTrip

Ez a szoftver Matlab eszközkészletként lett megvalósítva, amely magas szintű funkciókat tartalmaz az idegtudósok számára. Magában foglal közel 100 magasszintű és 900 egyszerű algoritmust, mint például a parametrikus és nem paraméteres spektrális elemzés, többfokozatú időfrekvencia-elemzés vagy az összeköttetési elemzés. Ezen kívül a felhasználók és a fejlesztők könnyedén kibővíthetik a funkciókat és új algoritmusokat vezethetnek be. Más EEG feldolgozó szoftverrel ellentétben a FieldTrip nem rendelkezik grafikus felhasználói felülettel, ezért a felhasználóknak tisztában kell lenniük a Matlab-ban lévő parancssor és scriptek használatával.

## BrainVision Analyzer

A BrainVision Analyzer a tudósok számára lett kifejlesztve, különféle neurofiziológiai adatok elemzésére. A szoftver alapvető és fejlett elemzési módszereket kínál, beleértve az összes szükséges előfeldolgozási funkciót, továbbfejlesztett idő-frekvencia elemzési lehetőségeket, ICA, LORETA és MRI korrekciókat. A program automatikusan felismeri a különféle EEG gyártok fájljait, és lehetővé teszik a felhasználók számára, hogy a világ minden tájáról kommunikálhassanak a kutatólaboratóriumokkal. Különböző nézetek korlátlan lehetőségeket teremtenek az eredmények megjelenítésére.



# Felhasznált technológiák

A szakdolgozat céljául kitűzött szoftver fejlesztése során számos technológia és módszer lett alkalmazva. A következőkben e technológiákat mutatom be röviden.

## Ionic

Az Ionic egy nyílt forráskódú felhasználói felület eszközkészlet, mellyel mobil, illetve asztali alkalmazásokat lehet készíteni a napjainkban népszerű Angular, React vagy Vue keretrendszerek integrációjával. Egyetlen kódbázissal lehetőség nyílik különböző platformokra (Android, iOS) szánt alkalmazások fejlesztésére. Az eszközkészlet előre implementált komponensei, például a gombok, ablakok, animációk, listák, ikonok és emellett még rengeteg egyéb eszköz, lehetővé teszik az egyszerű és gyors fejlesztést. A komponensek mellett az Ionic egy folyamatosan bővülő plugin könyvtárat tesz elérhetővé, amiket a Cordova vagy Capacitor kiegészítő projektekkel lehet használni.



## Firebase

A Firebase egy modern felhőalapú szolgáltatás, mely segítségével robosztus webes alkalmazások készülhetnek. Szolgáltatásai közül a leggyakrabban használtak a valós idejű adatbázis és a cloud functions. A NoSQL adatbázisban hierarchiába rendezett adatokat tudunk tárolni különálló, úgynevezett kollekciókban, melyeken végzett módostásról azonnal tudunk értesülni. A cloud functions a felhőben futó folyamatok, amikor akkor lépnek működésbe, ha bizonyos események bekövetkeznek. Ilyen lehet például, ha egy rekord értéke megváltozik.



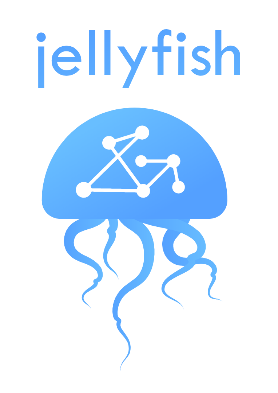
## MNE library

Az MNE egy nyílt forráskódú könyvtár, amellyel neurofiziológiai adatok (MEG, EEG, sEEG, ECoG) feldolgozására, elemzésére és vizualizációjára van lehetőség. A könyvtárban számtalan módszert és algoritmust implementáltak, kezdve az egyszerű adatfeldolgozó függvényektől egészen a bonyolult matematikai algoritmusokig, mint például a gyakran használt Fast Fourier transzformáció. Ezeken kívül 2D-s, illetve 3D-s adatvizualizációt is kínál, a hatékonyabb vizsgálatok érdekében. A könyvtár használatához nem feltétlen kell rendelkeznünk egy neurofiziológiai mérőeszközzel, hogy az adatokat fel tudjuk dolgozni, hiszen az MNE felhőalapú adatbázisából rengeteg mintaadat letölthető és szabadon felhasználható.



# A Jellyfish alkalmazás

A Jellyfish EEG jelek rögzítését, feldolgozását és vizualizációját teszi lehetővé okostelefonon. Az alkalmazás egy hibrid keretrendszerrel lett megvalósítva, vagyis egyaránt használható Android, illetve iOS operációs rendszer alatt. Elsősorban Emotiv készülékekkel használható (EPOC+ és Insight), azonban más termékcsaláddal való kompatibilitás megvalósítása is tervben van a jövőben. Az EEG készülék Bluetooth kommunikációval csatlakozik az adott telefonra, mely pontos és gyors adatátvitelt tesz lehetővé bizonyos hatótávolságon belül. Az előző fejezetben említett alkalmazásoktól főként a felhasználók kezelésével és az adatok tárolásával tér el. Az adatok tárolása a felhőben történik, így biztosítva, hogy más felhasználók is szabadon felhasználhassák az adatokat különböző projektekben a világ bármely részén. Természetesen lehetőség van privát felvételek készítésére és feldolgozására is, védve a páciensek személyiségi jogait.



Az EEG jelek feldolgozásáról az MNE könyvtár gondoskodik, amit számos más alkalmazás is használ. Az EEG felvételt követően az adatok a felhőbe kerülnek, ahol a könyvtár végrehajtja a szükséges módosításokat és előfeldolgozásokat a kívánt megjelenítés végett.

A fejlesztett alkalmazást számos területen lehet alkalmazni, kezdve az orvosi felhasználástól egészen a kutatómunkákig. Orvosi szakterületen az alkalmazás egy gyorsabb és hatékonyabb megoldást nyújt a páciensek agyi rendellenességeinek feltérképezésében. Az alkalmazást használva az orvosoknak nincs szükségük bonyolult számítógépekre és konfigurációkra, csupán egy okostelefonnal és egy hozzá kompatibilis EEG eszközzel kell rendelkezniük a munkához.

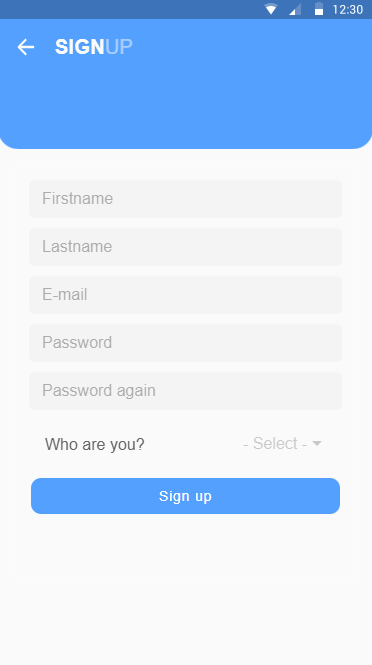
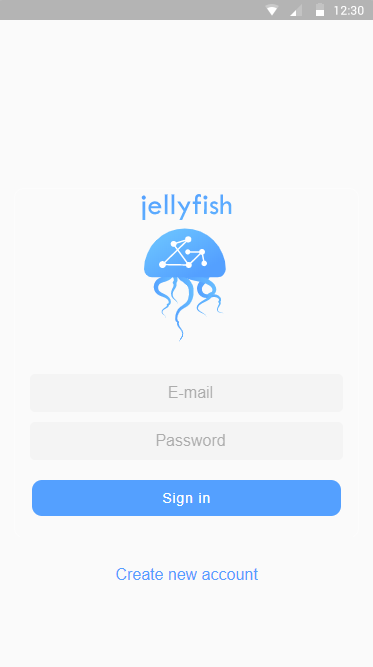
## 3.1. Felhasználó autentikáció

Az alkalmazás egy fontos egysége a felhasználó kezelés, hiszen minden EEG mérés egy adott felhasználóhoz kapcsolódik, ezért a program használatához rendelkeznünk kell egy felhasználói fiókkal.

A Jellyfish alkalmazást elindítva okostelefonunkon, egy bejelentkezés képernyő fogad minket. Amennyiben rendelkezünk felhasználói fiókkal, a felületen meg kell adni az e-mail címünk, illetve a hozzá tartozó jelszavunk a bejelentkezéshez. Azonban, ha még nincs, a „Create new account” feliratra kattintva létrehozhatunk egy új fiókot. Ekkor a program átnavigál a regisztráció oldalra, ahol meg kell adnunk a szükséges adatokat.

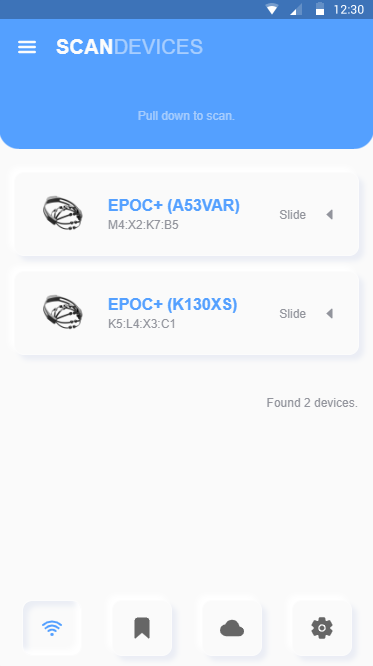
A vezetéknév, keresztnév, e-mail mezők kitöltése után ki kell még választanunk, hogy mi a szerepkörünk. A program három lehetőséget kínál fel: User, Doctor, Researcher. Erre azért van szükség, hogy a felhasználók könnyebben megtalálhassák a számukra fontos EEG méréseket felhasználói szerepkör szerint.

Minden mezőt kitöltve a „Sign up” gombra kattintva, validálódik a regisztrációnk a szerver felöl. Amennyiben a bejelentkezés során nem adódott hiba, átléphetünk a program fő oldalára, ahol a lényeges műveletek történnek.

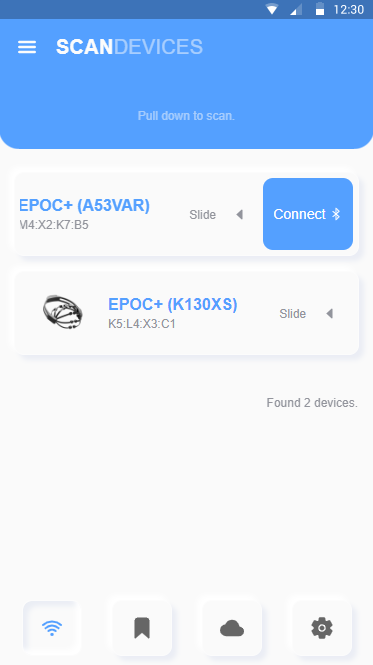


## EEG készülékek keresése és a csatlakozás folyamata

A megfelelő felhasználóval való bejelentkezés után a „Scan Devices” oldalra kerülünk. Itt lesznek láthatók a keresést követően a talált EEG készülékek.



Ahhoz, hogy EEG felvételeket tudjunk készíteni, szükségünk van egy EEG készülékre, amivel dolgozni fogunk. A jelenlegi rendszer Emotiv EPOC eszközökkel kompatibilis, ezért a keresés eredményei kizárólag ilyen headset-ek lesznek. Első lépésben az készülékeken és az okostelefonon is be kell kapcsolni a Bluetooth elérést, hogy csatlakozni tudjanak egymáshoz. Amennyiben ez sikerült, a képernyő tetejéről az aljára egy húzó mozgást kell végezni, amely elindítja az készülékek keresését. A keresés 5 másodpercig aktív, ezt követően jelennek meg a találatok a felületen. Minden találati sorban látható az készülékek neve és MAC címe. Ezen információk alapján fog a telefon az adott EEG eszközre csatlakozni. Amennyiben megtaláltuk az kívánt készüléket az aktuális sort balra húzva megjelenik egy „Connect” gomb, amire rákattintva csatlakozhatunk az eszközre. Ezzel a csatlakozás sikeresen megtörtént és következhet a felvétel beállításainak megadása.

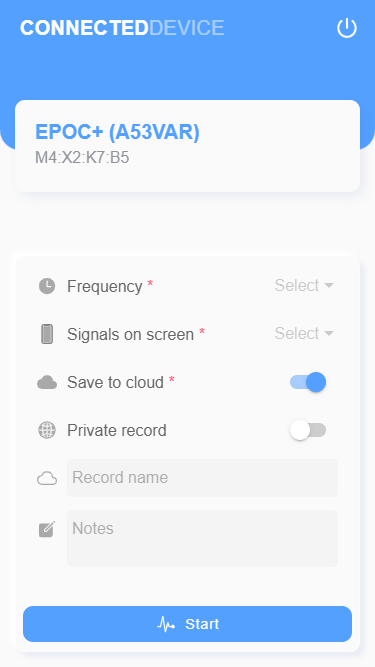
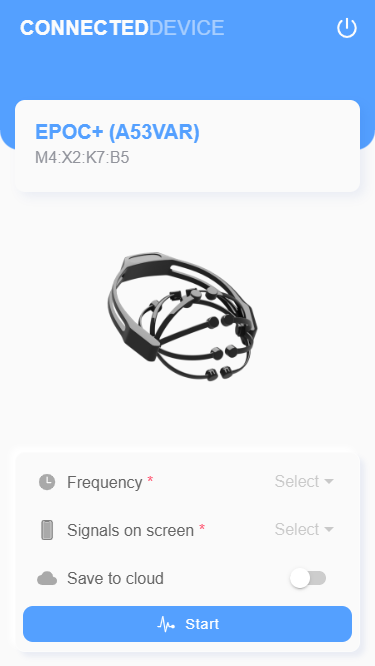


## Felvétel beállítások

Sikeres csatlakozás után lehetőség van megadni különféle specifikációkat, hogy számunka a felvétel a kívánt adatokat tartalmazza.

Meg kell adni a felvétel frekvenciáját, vagyis, hogy 1 másodperc alatt hány adatcsomagot kapjunk az egyes csatornáktól. Ezt a számot érdemes nagy értékre állítani, mivel ekkor a szenzorok által küldött jelek részletesebben jelennek meg. Továbbá ki kell választani, hogy a képernyőn mekkora sűrűséggel legyenek láthatók a jelek görbéi. Nagy sűrűség esetén a sok adat megjelenítése lassíthatja a számítást, ezért ezt az értéket az okostelefonunk teljesítményéhez viszonyítva érdemes megadni.

Amennyiben szükséges, a felvételt el tudjuk menteni későbbi feldolgozásra az alkalmazás által használt Firebase adatbázisba, ami egy felhő alapú adattárolást tesz lehetővé. Ekkor a „Save to Cloud” opciót be kell jelölni. Lehetőség van privát, illetve publikus tárolás megadására is. Privát tárolás estén csak számunkra lesz látható az adott felvétel. Publikus opciót választva az alkalmazás birtokában lévő felhasználók mind látni és használni tudják az adatokat. Ekkor meg kell adni egy nevet a felvétel beazonosítására és opcionálisan lehetőség van megjegyzés hozzáadására is. Amennyiben minden, a számunka fontos beállítást megadtuk, a felvételt elindíthatjuk a „Start” gombbal.



## Felvétel rögzítése

A csatlakoztatott eszköz oldalon a szükséges beállításokat megadva, elindíthatjuk az EEG mérést. Ekkor egy másik oldalra kerülünk, ahol a mérés közben tudjuk vizsgálni a felvételt. Amennyiben kiválasztottuk a felhőben tárolás lehetőséget az előző oldalon, úgy az adatok automatikusan mentésre kerülnek, és később vissza lehet nézni a mérést.

***[KÉP]***

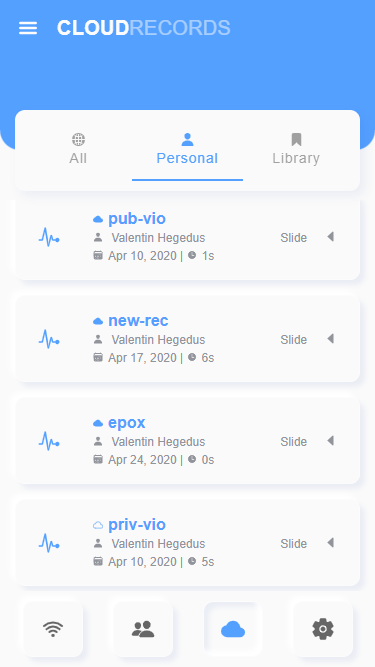
Az első oldalon az EEG jelek görbéit láthatjuk a vízszintes idő és horizontális amplitúdó tengelyen megjelenítve. Itt kettő megjelenítési mód közül lehet választani. Az egyik a „Line chart”, amit választva a jelek görbéi egymás alatt helyezkednek el. Ez lehetőség akkor ideális, ha a jeleket elszeparálva szeretnénk vizsgálni. A másik megjelenítési mód a „Butterfly mode”, ami egymásra halmozva jeleníti meg a 14 görbét, miközben egy úgynevezett „Baseline correction” algoritmus optimalizálja az azokat. Az algoritmus a futása során, minden pillanatban kiszámítja a szenzorok által kapott értékek átlagát, majd ezt az átlagértéket csatornánként kivonja minden értékből. Ezzel a módszerrel a megjelenített görbék egy alapvonalon fognak elhelyezkedni.

A második oldalon az EEG szenzorok által mért értékek találhatók lebegőpontos számábrázolásban kiírva egy listán. A lista elemeinek sorrendjét tetszés szerint meg lehet változtatni.

A harmadik oldalon található ábra az EEG készülék elektródáinak helyzetét és azok aktivitását reprezentálja. Amennyiben az egyik elektróda, az átlagértéktől eltérő aktivitást mér, úgy az őt helyettesítő indikátor kék színben felvillan. Ez a megjelenítési mód béta verzióban van, ezért működése hibákat eredményezhet. A végleges implementálására a közeljövőben kerül sor.

## Mentett felvételek

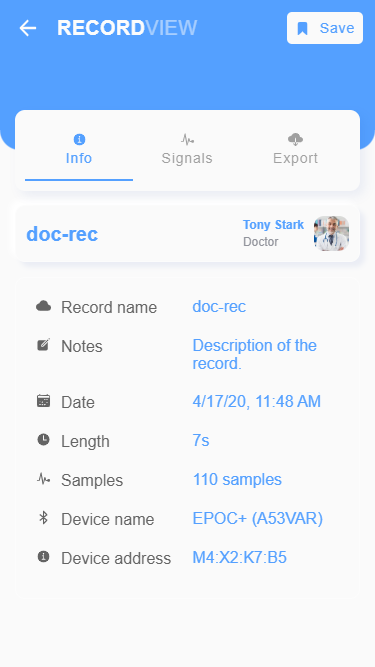
Az alkalmazás fő képernyőjén, a felhő felvételek oldalra navigálva (alsó navigációs sávban a felhő ikonra kattintva) érhetők el azok az EEG mérések felvételei, amiknél a felvétel beállításai során a felsőbe mentés opciót kiválasztottuk. Az oldal tetején egy szegmensnavigáció található, ami segítségével jeleníthetjük meg a felvételeket csoportokba bontva. A baloldali szegmensben az összes felvétel megtalálható, amiket a készítője nem tett privát láthatóságba. Középen a személyes privát és publikus felvételek egyaránt fellelhetők. Jobb oldalon azok a mérések szerepelnek, amelyeket elmentettünk későbbi felhasználásra. Ez amolyan tárolóként viselkedik, így a számunka fontos felvételeket egy helyen lehet gyűjteni.



Minden találati sor egy EEG mérést reprezentál, mely rövid információt tartalmaz a felvételről. Ezek az információk a felvétel neve, a tulajdonos teljes neve, a készítés dátuma és a felvétel hossza. Egy adott találati sort jobbra húzva megjelenik egy „More” gomb, amire kattintva bővebb információt kaphatunk a felvételről és megvizsgálhatjuk a szenzor adatok görbéit.

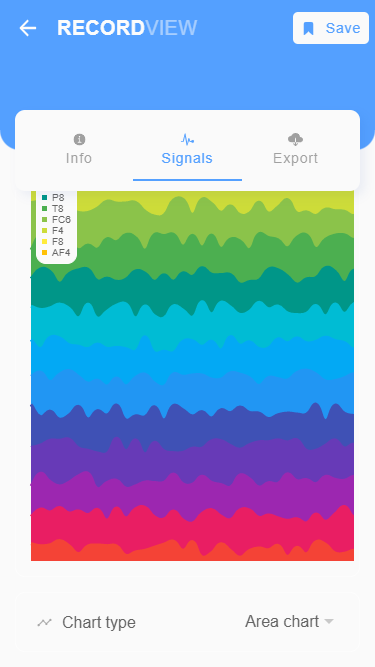
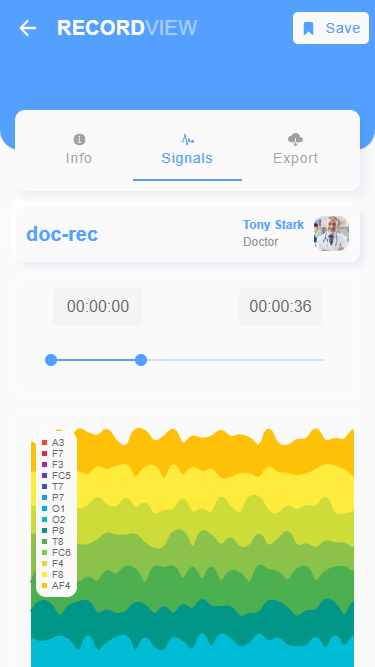
## Felvétel megtekintése

Amennyiben kiválasztottuk a felvételt, amit meg szeretnénk vizsgálni, a következő oldalra navigálja a felhasználót az alkalmazás:

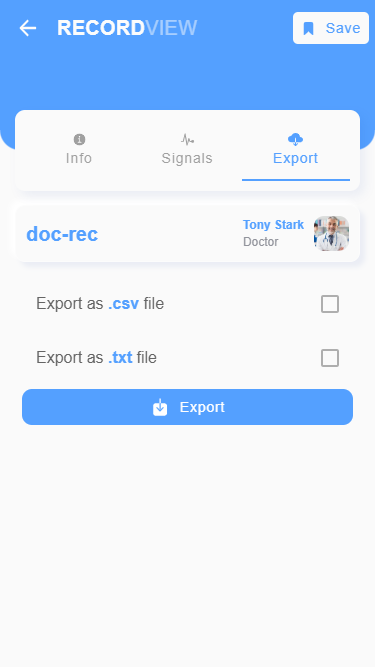


Ezen az oldalon bővebb információt kapunk a felvételről, mint például egy ismertető leírás, a felvétel során vett mintavételek száma, a felvételt készítő EEG készülék neve és MAC címe. A felvétel nevének sorában látható a készítője neve, amire kattintva átkerülünk a felhasználó profiljára, ahol a felhasználóról tudhatunk meg bővebb információkat. Ezeken kívül az oldal jobb felső sarkában egy „Save” felirattal rendelkező gomb található, mellyel el tudjuk menteni a felvételt későbbi felhasználásra egy tárolóba (library).

A szegmensnavigáción a „Signals” menüpontot választva, a következő oldalon a felhőbe mentett EEG szenzorok adatait láthatjuk. Az oldalon található időtartomány csúszkával vagy az input mezőkkel beállítva a kezdő és végpontot, lehetőség van a felvétel egy bizonyos szakaszát megtekinteni a kijelölt időintervallum között. A kirajzolt görbék területén ujjmozdulatokkal is be lehet állítani a kívánt időtartamot. Itt színnel elkülönítve jelennek meg a szenzorok görbéi, melyek beazonosításához a bal felső sarokban címkék találhatók a szenzorok neveivel és a hozzá rendelt színekkel. Az oldal aljára lapozva hasonlóan, mint a mérési oldalon, itt is meg lehet adni, hogy az adatok milyen formában legyenek megjelenítve.



A szegmensmenü utolsó opciója az „Export” lap, ahol a felvett méréseket és a hozzájuk tartozó kiegészítő információkat különböző formátumú fájlba tudjuk exportálni. Ekkor nem a szenzor görbék kerülnek mentésre, hanem az átalakított lebegőpontos számok. Az adatok lementéséhez először a megfelelő intervallumot kell kiválasztani az előző oldalon, majd rányomni a kívánt exportálás gombra. Szükség szerint TXT vagy esetleg CSV formátumba tudjuk kimenteni az adatokat a programból a telefon memóriájába. Mindkét esetben a fájl elején lesznek találhatók a kiegészítő információk, amiket a szenzor adatok követnek csatornánként elkülönítve.



# Adatok kinyerése és feldolgozása

Az alkalmazás fejlesztése során az EEG készülékekkel való kommunikáció, dekódolás és az adatok visszafejtése Android Native Plugin-ként lett kivitelezve Java programozási nyelven. Az efféle megvalósítás azért hatékony, mert maga az alkalmazás és az EEG-vel kapcsolatos metódusok el vannak különítve egymástól, így más platform-ra szánt alkalmazásnál is fel lehet használni ugyanezt a kódot. A következőkben e funkciók működését ismertetem részletesebben.

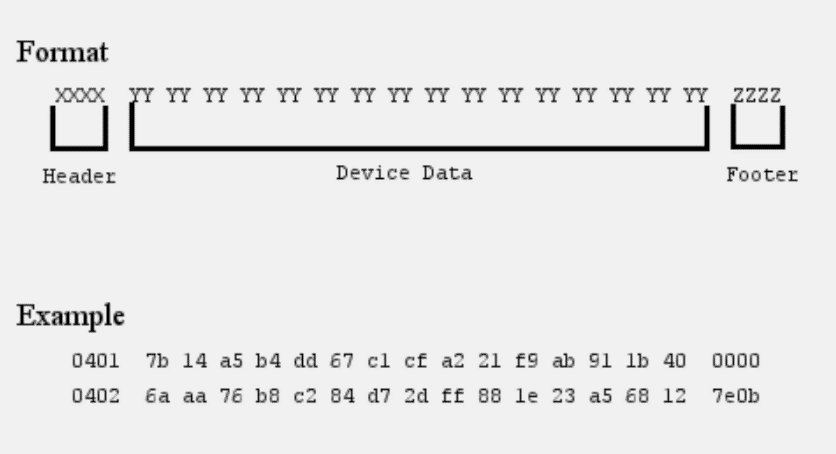
Az EEG-vel kapcsolatos algoritmusok és metódusok, a CyKIT Python nyelven íródott program alapján lettek implementálva.

## 4.1. Mi a szenzor adat?

## 4.2. Adatfolyam, adatcsomagok

Az EEG-re csatlakozást követően, az adatokat nem olyan formában kapjuk meg, hogy azokat használni tudjuk az alkalmazásban. A kinyert adatokon számos algoritmus kell futtatni, mire használható formátumú eredményt kapunk belőlük.

Az adatok úgynevezett adatfolyamon érkeznek az EEG készüléktől. Nem csupán egy számot kapunk meg ilyenkor, hanem egy adatcsomagot, ami több részből tevődik össze. A Bluetooth adatfolyamon kapott csomagok formátuma különbözik a WiFi-nél használt formátumnál (X. ábra).



X. ábra. Az adatcsomag

Minden csomag 20 bájtból áll össze. A csomag elején lévő kettő bájt a fejléc (Header), ami egy számlálóból és egy csomagmegkülönböztető jelzőszámból áll. A számláló azt jelzi, hogy az adott csomag hányadik sorszámú az adatfolyamban. Ez a szám 0-tól 127-ig terjed, mivel az EPOC+ készülék a 128Hz-et használja alapértelmezetten. A 128Hz megadja, hogy egy másodperc alatt 128 darab adatcsomag fog érkezni az adatfolyamon. Lehetőség van bizonyos EMOTIV készülékeket 256Hz-re is állítani, ekkor a számláló 0 és 255 között fog mozogni.

Egy adatcsomag két részben érkezik meg az EEG eszköztől, ezért szükséges ezek érkezési sorrendjükkel tisztában lenni. Ezt a cél szolgálja a fejléc második fele a csomagmegkülönböztető jelzőszám, ami azt jelzi, hogy az adatcsomag melyik részét kaptuk meg a kettő közül. 128Hz-et használva ez a szám 01 vagy 02 lehet. A 01 szám az EEG jelek csomagját jelzi, a 02 pedig az készülék giroszkóp adatait. A jelzőszámok alapján a két csomag részt összeillesztve kapjuk meg az adatfolyam egy adatcsomagját bájtonként ábrázolva.

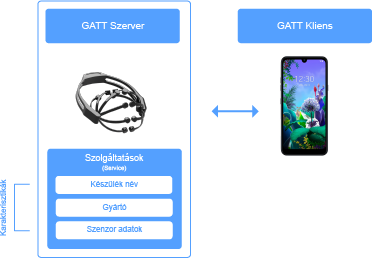
A fejlécet követő tizenhat bájt a csomag törzse (Device Data), ahol az EEG szenzorok adatai helyezkednek el. Ezek szintént bájtonként érkeznek, ezért későbbi feldolgozás során át kell alakítani használható formátumba, mint például lebegőpontos számokra. A tizenhat bájt megfeleltethető az EEG eszközön a tizenhat szenzornak, amelyek a következők: AF3, F7, F3, FC5, T7, P3, P7, O1, O2, P8, P4, T8, FC6, F4, F8, AF4.

A csomag végén található kettő bájt a lábléc (Footer), ahol olyan kiegészítő információkat találhatunk, mint az eszköz akkumulátorának feltöltöttsége vagy a szenzorok illeszkedésének minősége. Ez a minőségi mutató megadja, hogy a szenzorok milyen jól érintkeznek a fejbőrrel, ami alapján nagy biztonsággal meg tudjuk állapítani a kapott adataink minőségét.

## 4.3. Bluetooth LE – Az EEG, mint GATT szerver

Az Emotiv EEG készülékek a Bluetooth Low Energy (BLE) vezetéknélküli technológiát használják. A klasszikus Bluetooth technologiához képest a BLE, amit az angol neve is sugall, kevesebb energiát fogyaszt. Ez a protokoll arra lett optimalizálva, hogy az eszközök, mint például a szenzorok, kis mennyiségű adatot küldjenek.

A BLE az úgynevezett általános specifikáció profilon (General Attribute Profile - GATT) alapszik, ami lehetővé teszi rövid adatok küldését és fogadását. Megkülönböztetünk GATT klienst és GATT szervert a működésük szempontjából. Az EEG esetén, GATT kliens az okostelefon és GATT szerver az EEG készülék (X. ábra). A GATT szerverek szolgáltatásokat nyújtanak, amik tulajdonképpen az adott eszköz funkciói.



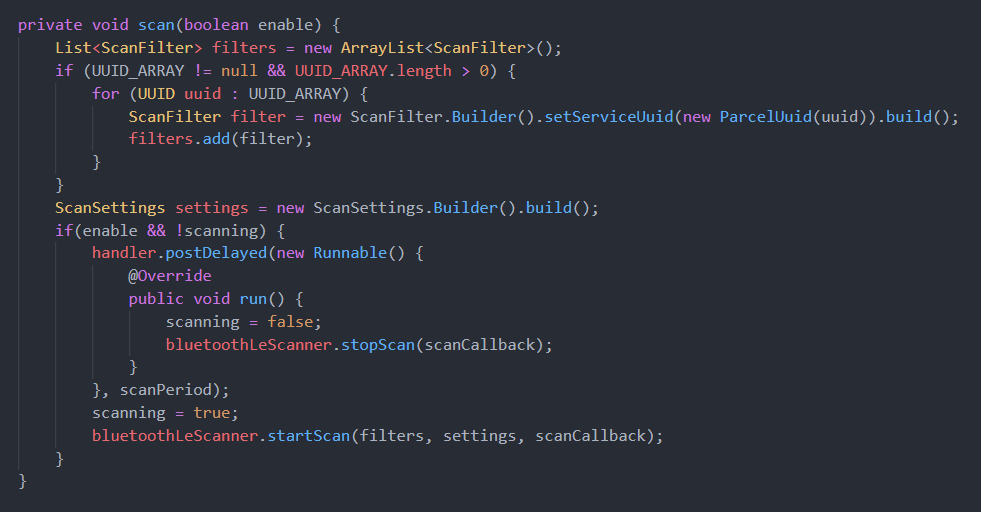
A szolgáltatások beazonosításához egy-egy egyedi UUID[[4]](#footnote-4) azonosítót használnak. Ezek a szolgáltatások magukban foglalnak karakterisztikákat, amiken írás, illetve olvasás műveletet lehet végrehajtani. Későbbiekben a szenzor adatok kinyerésére a karakterisztikákat fogjuk felhasználni.

## 4.4. Keresés és csatlakozás a készülékre

### 4.4.1. A scannelés folyamata

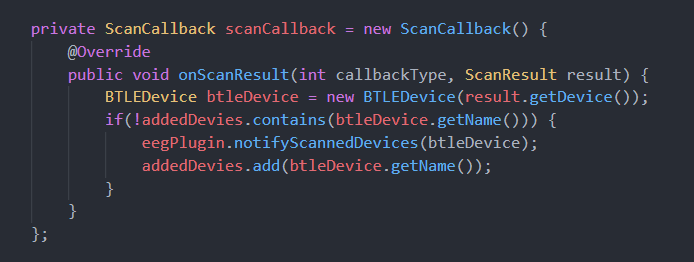
Az Jellyfish alkalmazásban a készülékek keresését elindítva az Android Plugin megkezdi a scennelést, amely háttérfolyamatként fut, így az alkalmazással való interakció továbbra is lehetséges eközben. Alapértelmezetten 5 másodpercig aktív a keresés, azonban az alkalmazásban a beállítások oldalon szükség szerint ezt megváltoztathatjuk.

A keresés során nem szeretnénk, hogy az EEG készülékek kivételével, más oda nem illő eszközök is a találatok között legyenek, ezért az EEG egyedi azonosító száma[[5]](#footnote-5) szerint szűrést végzünk (X. ábra).



Amikor egy készüléket találtunk (X. ábra), az eredményobjektumból létrejön egy BTLE Device objektum. Ez tartalmazza az eszköz nevét, gyártójának nevét, MAC címét és RSSI (Received Signal Strength Indication) értékét. Az RSSI érték a telefonban lévő antenna által mért rádióhullám erőssége. Ez a szám alapján lehet további szűréseket végezni, mint például, hogy csak bizonyos jel intervallumon belül találjon a telefon EEG készülékeket, ezzel kihagyva a távoli, gyenge jelerősséggel rendelkező eszközöket.

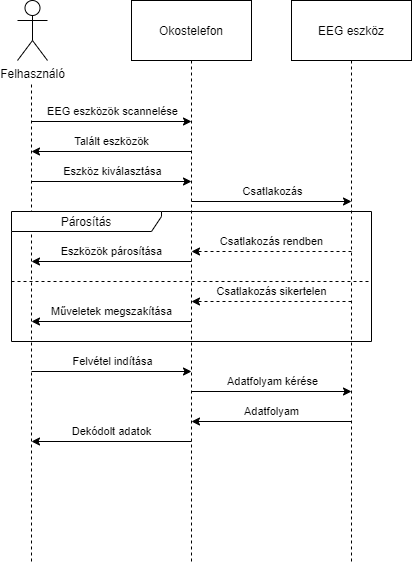
Miután a BTLE Device objektum létrejött és beállította saját paramétereit, a Plugin kódban egy listában tárolásra kerül a készülék objektum, amely a csatlakozás folyamata közben lesz felhasználva. Ezt követően a notifyScannedDevices() metódus értesíti a képernyőt a talált eszközről, ahol megjelennek az adatai.



### 4.4.1. A csatlakozás folyamata

[REFACTOR]

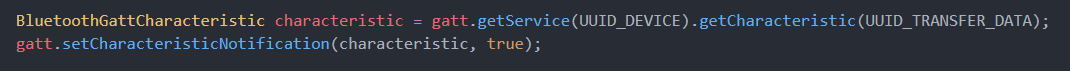
Amennyiben az eszközök scannelése megtörtént és sikeresen találtunk EEG készüléket, meg kell tenni a szükséges lépéseket a csatlakozáshoz, ahogy azt a korábbi fejezetben (3.2. EEG készülékek keresése és csatlakozása) részletesen kifejtettem. A csatlakozás és keresés menetét a következő szekvencia diagrammal lehet szemléltetni:



Először meg kell keresni azt a készüléket, amire csatlakozni szeretnénk. Ez a folyamat visszaad készülékeket, amiket talált, majd ki kell választani a csatlakoztatni kívánt eszközt. A forráskódban ennek a folyamata az, hogy a kiválasztott EEG készülék neve alapján a scannelt eszközök listájából megkeresi azt a BTLE Device objektumot, amelynek neve megegyezik. Ezután ezzel az objektummal létrehoz egy GATT objektumot. Ahogy a BluetoothLE fejezetben kifejtettem, a mi esetünkben az EEG készülék GATT szerverként fog viselkedni. Amennyiben a folyamat során nem adódott hiba, létrejött az eszköz profilja és ez által el tudjuk érni az összes funkcióját és karakterisztikáját.

## 4.5. Adatfolyam indítása és kezelése

Adatfolyam indítása a csatlakoztatott eszköz oldalon lévő „Start” gombbal lehetséges. Amikor a gombbal megtörtént az interakció, a programnak jeleznie kell az EEG készüléknek, hogy az alkalmazás készen áll az adatok fogadására és megkezdődhet az adatok küldése. Ekkor a Java plugin felderíti az eszköz által nyújtott szolgáltatásokat és megkeresi a megfelelő karakterisztikát az UUID (”81072F41-9F3D-11E3-A9DC-0002A5D5C51B”) alapján, amitől a nyers adatokat fogadjuk. Először ennek a karakterisztikának a „notification” adattagját be kell állítani megfelelően, hogy értesítést kapjunk minden pillanatban, amikor egy szenzor érték megváltozott:



Ezt követően egy bájtot kell küldeni az előbbi karakterisztikára, hogy ezzel elindítsuk az adatfolyamot:



Amennyiben minden sikeresen lezajlott az „onCharacteristicChanged()” függvényben megkapjuk a szenzor adatokat minden pillanatban, amikor változás történik. Alapértelmezetten az EPOC+ készüléknél 128Hz-en, másodpercenként 128 adatcsomagot kapunk. Mivel ezek a csomagok két részben érkeznek, ezért az összetartozókat egymás után kell illeszteni. Erre egy DataPacket struktúrát alakítottam ki, ami elvégzi a szükséges műveleteket. Négy adattagja van, az egyik a csomag első fele, a másik a csomag második fele és a hozzájuk tartozó logikai adattagok, amik jelzik, ha a csomag egyik része megérkezett. Amikor egy adat érkezik, automatikusan a struktúra megfelelő adattagjába kerül és a hozzátartozó logikai változó igaz értékre módosul, miszerint a csomag megérkezett. Előfordulhat olyan eset is, hogy valami hiba folytán (gyakran gyenge akkumulátor töltöttségnél) a csomag egyik fele már megérkezett, de a másik nem. Ekkor az adattagok értéke törlésre kerül és a logikai változók hamis értékre állítódnak. A kiürítést követően a struktúra készen áll az adatfolyamon lévő következő csomag fogadására.

## 4.6. Az adatok dekódolása

Ahogyan a korábbi fejezetekben kifejtettem, az adatfolyamon az adatcsomagok bájt formátumban érkeznek két részben. Ezek az értékek még nem a végleges szenzor adatok, ezért át kell esniük különböző algoritmusokon, mire a visszafejtett lebegőpontos számokat megkapjuk. Az EEG szenzor karakterisztikájából kapott bájtcsomagok titkosítva vannak biztonsági okokból, ezért első lépésben ezt a kódolást kell feloldani. Az adatok titkosítását AES/ECB/NoPadding módszerrel oldották meg az Emotiv fejlesztői, így a dekódolást ennek megfelelően kell végrehajtani.

## 4.6.1. Advanced Encryption Standard (AES)

Az Advanced Standard Encryption adatok titkosításához használt módszer. Egy hálózaton való biztonságos információ küldés és fogadás érdekében, egy AES kulcs van használva a titkosításhoz. Az adatokat csak olyan személyek tudják elérni, akik rendelkeznek a titkosításkor használt kulccsal. Más titkosító módszerrel ellentétben, az AES szimmetrikus kulcsú titkosítást használ, vagyis a kódolás és dekódolás folyamata egyetlen kulccsal történik.

## 4.6.2. AES kulcsgenerálás

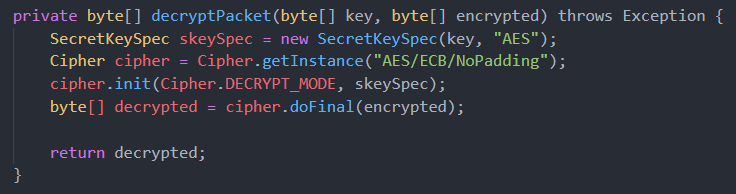
Az AES dekódoláshoz szükség van egy kulcsra, ami megmondja, hogy az adat hogyan legyen feloldva. Ezt a kulcsot automatikusan az EEG készülék nevéből és szériaszámából lehet legenerálni, amit a következő ábra (X. ábra) demonstrál.



A kulcs generálásának folyamatát egy példán keresztül mutatom be. Vegyük például az EPOC+ (4A2C06F6) névvel és szériaszámmal rendelkező eszközt. Ahogy a kódrészleten is látható, először a szériaszám karaktereit bizonyos sorrendbe[[6]](#footnote-6) kell egymás után helyezni. Az igy kapott eredmény a *„F6062C4A”* szöveg lesz, amit bájt tömbbé (*[246, 6, 44, 74]*) kell konvertálni. Ezután ezt a bájt tömböt egy 12 hosszú 0-ás bájtokból (0x00) álló tömbbel kell konkatenálni. A kapott lista elemeinek megfelelő permutációját[[7]](#footnote-7) véve kapjuk meg az AES kulcsot (*[74, 44, 44, 6, 6, 6, 44, 246, 74, 246, 44, 44, 246, 246, 44, 74]*) bájttömbként, melyet a dekódolás során tudunk használni. Ez a kulcsgenerálás folyamat minden alkalommal, amikor egy EEG készülék csatlakozik az okostelefonhoz, automatikusan lezajlik.

## 4.6.3. Dekódolás

Az adatfolyamon érkező bájtcsomagok többletinformációt (Header és Footer) tartalmaznak, amikre nincs szükség a szenzor adatok dekódolásánál, ezért ezeket egyszerűen levágjuk. Így a 20 bájtból álló listából az első kettő és utolsó kettő bájtot elhagyjuk. Az ekképpen előálló lista, tehát 16 elemből fog állni, ami a szenzor adatokat tartalmazza. Ezt követően a Java plugin-ban implementált decryptPacket() nevű függvénynek át kell adni a dekódolni kívánt csomagot és a korábbiakban legenerált AES kulcsot.



[TODO]

## 4.6.4. Konvertálás

Az Jellyfish alkalmazásban egy diagramon szeretnénk megjeleníteni az EEG szenzor adatait, ami a bájtokat nem tudja feldolgozni. Ezért szükséges egyfajta konvertálást végrehajtani az adatokon, hogy a megfelelő formában legyenek megjelenítve. Ehhez a bájtfolyam értékeit lebegőpontos számmá kell átalakítani. Ezt az műveletet, a kódban szereplő convertEPOCPlus() függvény hajtja végre, melynek bemenete az előző lépésben méretre vágott és dekódolt adat. A függvény működése során a dekódolt lista bájtjain kettesével végig haladva végzi el a számításokat. Az alábbi ábrán (X. ábra) látható convertData() függvény alakítja a bájtokat lebegőpontos számokká. Mivel az itt implementált műveletek a CyKIT program mintájára készültek, ezért a kiszámított értékek kisebb eltéréseket tartalmazhatnak.



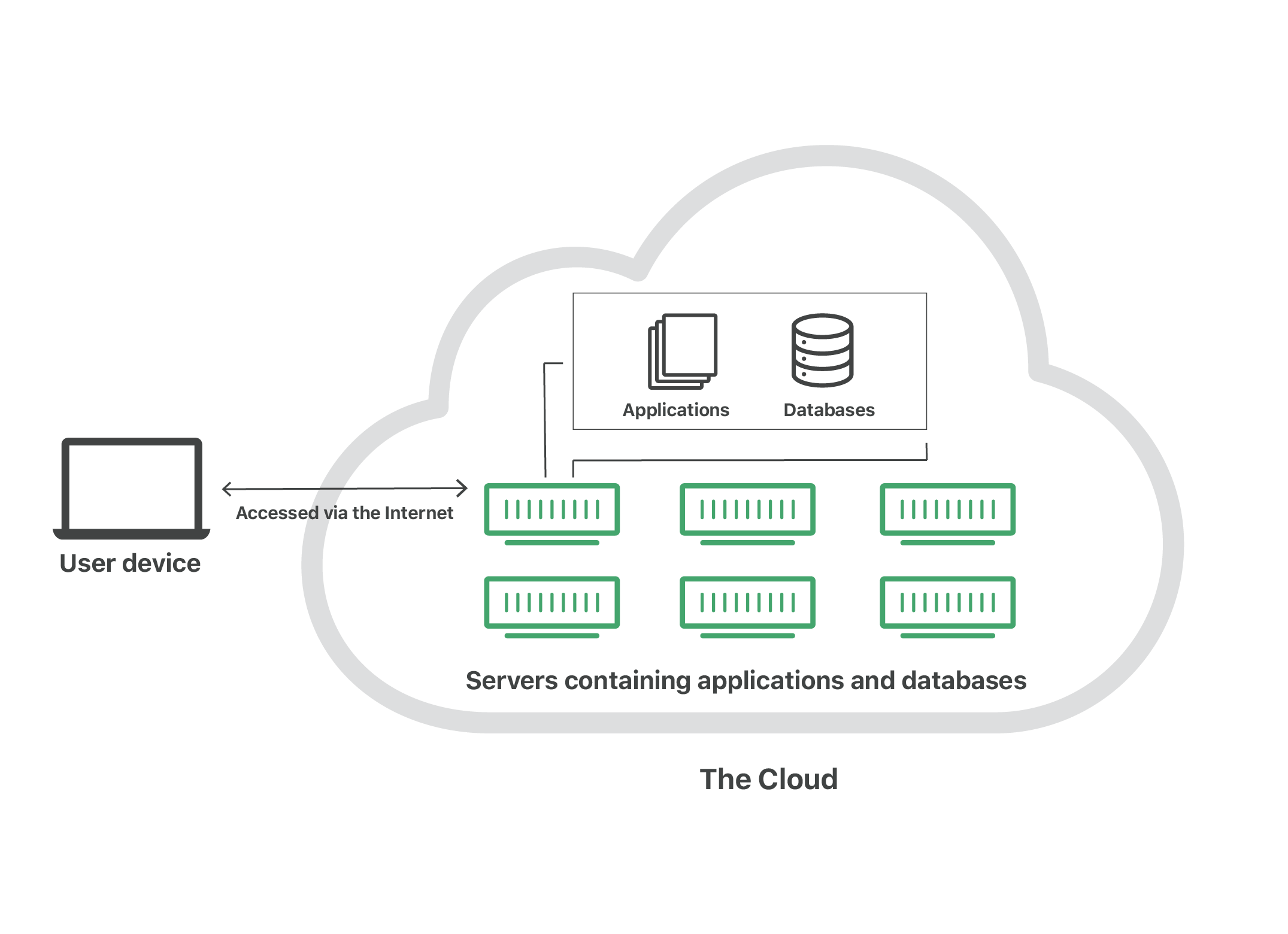
[TODO]

# Felhőalapú számítástechnika

Manapság a felhőben való adattárolás és feldolgozás egy hatékony és megbízható megoldás minden ember számára, melynek számos előnye van. A Jellyfish alkalmazásban fontos szerepet játszik ez a technológia, mivel az adatok tárolása és feldolgozása a felhőben történik. A következőkben ezt a modern szolgáltatást mutatom be, majd példákkal szemléltetem a működését a fejlesztett alkalmazásban.

## 5.1. Mi a felhő?

A felhő az interneten elérhető kiszolgálókra, valamint ezeken a kiszolgálókon futó szoftverekre és adatbázisokra utal. Ezek a felhő szervek az adatközpontokban találhatók a világ minden táján. Egy felhőszolgáltatást igénybe véve a felhasználóknak és a vállalatoknak nem kell a fizikai szervereket kezelniük vagy szoftvereket futtatni a saját lokális gépükön. A felhő lehetővé teszi a felhasználók számára, hogy a saját fájliakat és alkalmazásaikat elérjék szinte minden eszközről a világ minden táján, mivel az adatok tárolása az adatközpont szerverin kerül sor, nem pedig a felhasználó eszközén.



A felhőalapú szolgáltatások a virtualizációnak nevezett technológia miatt lehetséges. A vitrualizáció lehetővé teszi egy digitális számítógép létrehozását, amely úgy viselkedik, mintha egy fizikai számítógépünk lenne saját hardware-rel. Ezeket a digitális számítógépeket virtuális gépeknek nevezik. Amennyiben a rendszer megfelelően ki van alakítva, úgy a gazdagépen lévő virtuális gépek elkülönítve vannak kezelve, így azok interakciója egymással nem történik meg és fájljaik nem láthatók más virtuális gépek által, még akkor sem, ha ezek egy fizikai eszközön helyezkednek el. Előfordulhat olyan eset, hogy az egyes szerverek átmenetileg nem elérhetők, azonban a felhőkiszolgálóknak ebben az esetben is online kell maradniuk. Az ilyen nem várt hibák kiküszöbölésére biztonsági mentést készítenek több különböző szerveren és helyszínen, így az adatokat minden esetben el lehet érni.

## 5.2. Felhőszolgáltatások típusai

Nem minden felhő egyforma, ezért az igényeknek megfelelő szolgáltatást nyújtó architektúrát kell választani. A felhőszolgáltatásokat aszerint lehet csoportosítani, hogy a felhőt alkotó informatikai rendszer összetevőimből mennyit veszünk igénybe. Ezeket a szolgáltatásokat négy fő kategóriába lehet csoportosítani: infrastruktúraszolgáltatás (IaaS), platformszolgáltatás (PaaS), kiszolgáló nélküli és szoftverszolgáltatás (Saas).

5.2.1. Infrastructure as a Service (IaaS)

A felhőszolgáltatás legalsó szintje, melyet használva a felhőszolgáltató rendelkezésre bocsátja informatikai infrastruktúrája egy részét, amin saját rendszereket és alkalmazásokat lehet kiépíteni. Ebben a modellben egy cég béreli a szükséges kiszolgálókat és tárolókat felhő-szolgáltatótól. Az IaaS szolgáltatók között szerepel a DigitalOcean, a Google Compute Engine és az OpenStack.

5.2.2. Software as a Service (SaaS)

A felhő alapú szolgáltatások közül a leggyakrabban használt a SaaS szolgáltatás. Hagyományos módszerekkel szemben, itt nincs szükség az alkalmazások saját eszközre való telepítésére. A SaaS alkalmazásokat a felhőkiszolgálón tárolják, így közvetlenül a szolgáltató szerveréről futtatható. A felhasználók egy kliensprogramban vagy egy böngészőprogramban férnek hozzá az alkalmazásokhoz, amelyek bárhonnan elérhetők interneten keresztül. A modell nagy előnye, hogy a felmerülő technikai problémákat a gyártók kezelik, ezért egy cég IT csapatának nem kell foglalkoznia az alkalmazást kiszolgáló szerverekkel, verziókövetéssel és tárhellyel.

5.2.3. Kiszolgáló nélküli számítástechnika

Használatával egy cég fejlesztői gyorsabban fejleszthetnek alkalmazásokat, mivel nincs szükség arra, hogy ők kezeljék az infrastruktúrát. A kiszolgáló nélküli alkalmazások esetén a felhőszolgáltató automatikusan kiépíti és kezeli a kód futtatásához szükséges rendszert. A kiszolgáló nélküli kifejezés arra utal, hogy az infrastruktúra kiépítésével és kezelésével kapcsolatos feladatok a fejlesztő számára nem láthatók. Ez a módszer lehetővé teszi a fejlesztők számára, hogy az üzleti logikára összpontosítsanak és ez által több értéket hozzanak létre az üzlet számára.

### 5.2.4. Platform as a Service (PaaS)

Ez a modell, fejlesztők számára nyújt egy olyan keretrendszert, amelyben alkalmazásokat hozhatnak létre. A PaaS gyártók mindent kínálnak az alkalmazás létrehozásához az interneten keresztül, beleértve a fejlesztési eszközöket, az infrastruktúrát és az operációs rendszereket. A modell előnye, hogy a fejlesztők felhőalapú komponenseket integrálhatnak a szoftverjeikbe, bizonyos funkcionalitást már készen kapnak és tudják használni, mint például adatbázist. Egy projekten egyszerre több fejlesztőcsapat dolgozhat és a tesztelést, telepítést a felhőszolgáltatások segítik elő. Közkedvelt PaaS szolgáltatások például a Heroku és a Microsoft Azure.

## 5.3. Firebase Cloud Firestore

A Cloud Firestore egy rugalmas felhőalapú NoSQL adatbázis, mobil illetve webes alkalmazásfejlesztéshez a Firebase és a Google Cloud Platform segítségével. Az adatbázisban lévő adatokat szinkron módon el lehet érni kliens oldali programban és valós időben értesülhetünk azok változásaikról. A Firestore gyorsítótárazza a gyakran használt adatokat, így az alkalmazások adatot írhatnak és olvashatnak még akkor is, ha offline állapotban vannak. Újbóli online állapotba kerülést követően, a Firestore szinkronizálja a helyi változtatásokat az adatbázisba.

### 5.3.1. SQL és NoSQL

Az SQL adatbázisok strukturált lekérdezési nyelvet használnak az adatok létrehozására és manipulálására. Ez a fajta adatbázis megköveteli, hogy egy előre definiált sémát használjon az adatok szerkezetének kialakítására, mielőtt azokon dolgozna. Ennek a következménye, hogy az adatok új attribútummal való ellátása egy nehézkes folyamatot von maga után. Ezen kívül minden adatnak ugyanazt a struktúrát kell megvalósítania, mint a hozzá hasonlóknak. Az SQL adatbázisban az adatok táblázatokban kerülnek tárolásra, fix számú oszlopszámmal. A sorok, másnéven rekordok, reprezentálnak egy példányt, az adott tábla struktúrájának megfelelően. Skálázás szempontjából az SQL függőlegesen skálázható. Ez azt jelenti, hogy amennyiben a kiszolgáló gép tárhelye elfogy, több új szerver egység bekötése helyett, a CPU, RAM és SSD száma kerül növelésre.

Ezzel szemben a NoSQL adatbázis viszont dinamikus sémával rendelkezik a strukturáltalan adatok tárolására. Ilyen séma például az oszlop-orientált, dokumentum-orientált, grafikon-orientált vagy a kulcs-érték párként való tárolás. Az efféle adatbázis nagyban hatékony, mert nincs szükség előre definiált struktúrára, használat közben dinamikusan meg lehet változtatni. A dokumentumoknak nem szükséges egységes sémát megvalósítaniuk, így struktúrájuk eltérő lehet. Ezen felül minden dokumentum rendelkezhet egy vagy több al-dokumentummal is, amely a hierarchia kialakításában nagy szerepet játszik. A NoSQL adatbázis horizontálisan skálázható, vagyis amennyiben a kiszolgáló tárhelye elfogy, úgy az adatok feldarabolásra kerülnek, és a rendszerbe új fizikai gépek kerülnek, ezzel növelve a tárhelyet.

Az SQL és NoSQL adatbázisoknak eltérő a struktúrájuk, az adatok másképp kerülnek tárolásra és skálázhatóságuk is más, azonban a megfelelő helyen használva mindkettő hatékonynak minősül.

### 5.3.2. Firestore adattárolás

A NoSQL adatbázisan az adatok úgynevezett dokumentumokban vannak tárolva,

## Firebase triggerek

## Google Cloud feldolgozás és MNE

* Firebase adatbázis és felhőszolgáltatás
  + firebase adattárolás
  + google cloud adatfeldolgozás
  + stb
* Felhasznált technológiák
  + ionic
  + native plugin
  + firebase
* Mi a szenzor adat
  + 2 fajta szenzor adat
  + stb
* Advanced Encryption Standard (AES)
  + mi az aes
  + programban mire van
* MAC (Message Authentication Code)
  + mi a mac
  + példa jellifysh
* EEG
  + mi az
  + hogyan működik
  + alpha, beta, gamma

# Felhasznált irodalom, források

* OpenBCI - https://openbci.com
* EEGLab - https://sccn.ucsd.edu - Swartz Center for Computational Neuroscience
* Emotiv, EPOC+ - https://www.emotiv.com
* FieldTrip - Open Source Software for Advanced Analysis of MEG, EEG, and Invasive Electrophysiological Data - Robert Oostenveld
* BrainVision Analyzer - https://www.mathworks.com/
* *Bluetooth LE - https://developer.android.com/guide/topics/connectivity/bluetooth-le*
* *CyKIT - https://github.com/CymatiCorp/CyKit*
* *AES - A Review on Advanced Encryption Standards (AES) - https://www.researchgate.net/publication/338853730\_A\_Review\_on\_Advanced\_Encryption\_Standards\_AES*
* *Atkinson, R.L., Atkinson, R.C., Smith, E.E., Fredrickson, B.L., Nolem-Hoeksema, S., Loftus, G.R. (2005)*
* *Élettani, Szervezettani és Idegtudományi Tanszék honlapja - https://phys.bio.u-szeged.hu/*
* *MNE library - https://mne.tools/*
* *Ionic - https://ionicframework.com/docs*
* *https://www.cloudflare.com/learning/cloud/what-is-the-cloud/*
* *https://azure.microsoft.com/hu-hu/overview/what-is-cloud-computing/*
* *https://www.invitech.hu/*
* *https://firebase.google.com/docs/firestore*
* *https://www.xplenty.com/blog/the-sql-vs-nosql-difference/*
* *https://hackernoon.com/database-scaling-horizontal-and-vertical-scaling-85edd2fd9944*

NCBI, PubMed, Wiley, ResearchGate, Elseveir oldalak es altalaban be lehet jelentkezni az intezmenyen keresztul es akkor tobb cikkhez hozzafersz

1. Nasion - orrnyereg, inion - occipitális területen lévő kiemelkedés, és két preauriculárisnpont. [↑](#footnote-ref-1)
2. A Matlab speciális programrendszer, amelyet numerikus számítások elvégzésére fejlesztettek ki és emellett egy programozási nyelv. [↑](#footnote-ref-2)
3. Az Arduino egy szabad szoftveres, nyílt forráskódú elektronikai fejlesztőplatform, arra tervezve, hogy a különböző projektekben az elektronikus eszközök könnyebben hozzáférhetőek, kezelhetőek legyenek. [↑](#footnote-ref-3)
4. UUID (Universally Unique Identifier) – egyedi azonosító szám [↑](#footnote-ref-4)
5. Az EEG egyedi azonosító száma (UUID) - 81072F40-9F3D-11E3-A9DC-0002A5D5C51B [↑](#footnote-ref-5)
6. A sorrend minden Emotiv eszközöknél más, azonban a példában az EPOC+ készülékkel foglalkozok. [↑](#footnote-ref-6)
7. A szériaszám sorrendjéhez hasonlóan az AES kulcs permutáció is különbözik az Emotiv temékeknél. [↑](#footnote-ref-7)