

# TRACKSY

MESTERSÉGES INTELLIGENCIÁVAL  
VEZÉRELT MUNKAIDŐ-KÖVETÉS

GINOP-2.1.2-8-1-4-16-2017-00300

**SZÉCHENYI**  2020



MAGYARORSZÁG  
KORMÁNYA

Európai Unió  
Európai Regionális  
Fejlesztési Alap



**BEFEKTETÉS A JÖVŐBE**

# MI A TRACKSY?

**Egy munkaidő követő rendszer,  
ahol rögzítésre kerülnek az  
elfeledett munkaórák is  
Mesterséges Intelligencia  
segítségével.**

A Tracksy egy olyan egyedi munkaidő követésre használható rendszer, amely új alapokra helyezi a projekt alapú munkavégzés nyilvántartását.

Több online rendszerből álló szoftverkörnyezet és a hozzá tartozó asztali alkalmazás, amely először használ mesterséges intelligenciát, hogy az emberi tevékenység hibáit minimalizáljuk az egyes feladatokra fordított idő követésnél.



# HOGYAN MŰKÖDIK?

**A Tracksy ott hívja segítségül a Mesterséges Intelligenciát, ahol a legnagyobb szükség van rá.**

A projekt alapú munkavégzésnél nagyon fontos az idő logolása, azaz, hogy melyik munkafázisra mennyi időt használunk, mert ez az alapja a profitábilis működésnek és a hatékonyságnak.

Erre vannak megoldások (papír alapú adminisztráció, munkaidő követő szoftverek), azonban nagyon sok az olyan munkaóra, amely nem kerül rögzítésre. Ez jórészt feledékenységből fakad, a munkavállaló elfelejt átváltani egy másik feladatra, vagy egyáltalán nem logol.

Itt jön segítségül a **Tracksy**, amely egyrésztől automatikusan nézi a felhasználó tevékenységét, melyik projekten dolgozik, mikor hagyja abba a munkát, és kezdi újra, mennyi időt tölt el az adott munkafolyamattal anélkül, hogy ezt valahová feljegyezné.

Másik oldalról a Mesterséges Intelligencia megvizsgálja rendelkezésre álló adatokat, és azok hiányát és az alapján javaslatot tesz, hogy a hiányzó idősorban milyen feladattal foglalkozhatott a munkavállaló.

1. Begyűjti a projekt időnyilvántartáshoz szükséges adatokat.
2. Mesterséges Intelligencia segítségével meghatározza és javaslatot tesz, hogy melyik munkavállaló, melyik feladaton dolgozott, anélkül, hogy ezt valamilyen felületen adminisztrálni kellene.

# HOGYAN ÉPÜL FEL A RENDSZER?

## A rendszer az alábbi fő komponensekből épül fel:

- » Monitoring, adatgyűjtő komponensek (asztali alkalmazás és aggregátor)
- » Gépi tanuló komponens (Mesterséges Intelligencia): a gyűjtött adatok alapján megállapítja, hogy a munkavállaló melyik feladaton dolgozhat.
- » Feladatkövető back-end rendszer: tárolja a feladatokat és a hozzájuk párosított időbélyegzős bejegyzéseket, az MI segítségével keletkezett logokat.
- » Feladatkövető rendszer frontend: webes interface, ami alapvető monitorozási funkciókat valósít meg. Megjeleníti az ügyfeleket, *projekteket, feladatokat, felhasználókat, idő logokat*

# ADATFORRÁSOK

**A Tracksy jelen verziója az informatikai cégeket célozza meg**, hiszen itt minden munka projekt alapú, az elszámolásnak és a hatékonyság mérésének is a munkára fordított idő képezi az alapját.

Könnyen targetálható célcsoport, így az első adatforrások olyan bemeneti egységek, amelyek a szoftverfejlesztők munkáját segítik.

**1.** Az elsődleges adatforrás **a Visual Code programból** jön, (rövidítve: VSCode vagy VS Code) egy ingyenes, nyílt forráskódú kódszerkesztő, melyet a Microsoft fejleszt Windows, Linux és OS X operációs rendszerekhez. Fejlesztők milliói használják a napi munkájukhoz, így ideális választás volt a rendszerhez illesztéshez. Egy egyedi bővítményt írtunk a programhoz, amely adatokat szolgáltat a Tracksy rendszer aggregátor felületéhez.

**2.** A következő adatforrásunk a **GitLab** egy webalapú Git verziókövető, amely wiki, kiadáskövető és CI / CD pipeline funkciókat biztosít, nyílt forráskódú licenc felhasználásával, azaz segítségével egy szoftverfejlesztési folyamat minden egyes lépése lekövethető, így ideális választás volt a munkaidő automatikus követésére.

**3. Billentyűzet és egér figyelés** adatforrása Szükség van az úgynevezett lightwork figyelésére, ahol azt lehet figyelni a számítógép használatban van-e egyáltalán, a programozói tevékenységtől függetlenül. Ennek a segítségével tudjuk megállapítani ha a felhasználó abbahagyta az adott feladattal való tényleges foglalkozást.

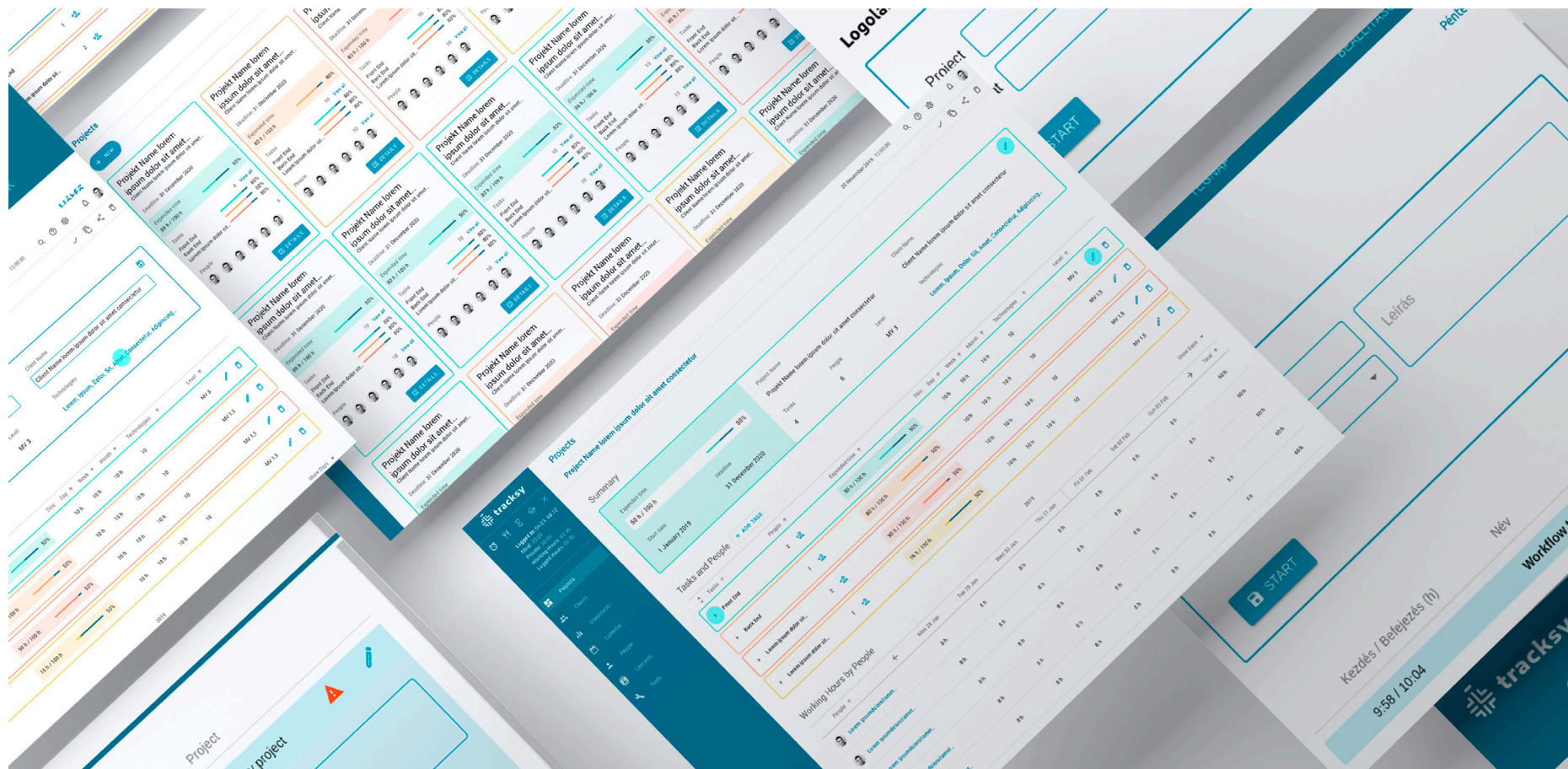
#### **4. Aktív ablak figyelése**

Az aktív ablak figyelésével tudjuk megállapítani, hogy melyik az a tevékenység amellyel ténylegesen foglalkozik a fejlesztő egy adott pillanatban.

# FELHASZNÁLÓ FELÜLET

## A webes felületen tudjuk nyomon követni a projekteket és a rájuk fordított időt

A felhasználók a jogosulási szintnek megfelelően a böngészőn elérhető webes interfészt használva nyomon követheti a munkaidőt, kezelheti a felhasználókat, a projekteket, feladatokat hozhat létre, módosíthat, feladatot delegálhatnak.



# TELEPÍTHETŐ ALKALMAZÁS ÉS TRACKSY BRAIN

## TELEPÍTHETŐ ALKALMAZÁS

Mivel a fejlesztők (és más számítógépen dolgozó felhasználók) sokszor egyszerre több böngésző ablakot hagynak megnyitva munkájuk végzéséhez, ezért úgy döntöttünk, hogy szükség van egy telepíthető alkalmazásra is, ahol fel monitorozni tudjuk és fel tudjuk rögzíteni a projekt követéshez szükséges információkat anélkül, hogy belépnénk a webes felületre.

## TRACKSY BRAIN

A mesterséges intelligencia az aggregátorról lekérdezhető, illetve felhasználóra és idő intervallumra szűrhető adatokat feldolgozza, ebből összeállít egy munkaidő naplót. Az összeállított naplót az aggregátornak küldi vissza, majd az aggregátor továbbítja ezt a szerver alkalmazás felé.

Az összeállított adatok pontosságát az adatok mennyisége eredményezi, minél több adat áll rendelkezésre, annál pontosabb becslést ad. Az asztali alkalmazásban módosított bejegyzéseket a rendszer konstansnak tekinti így az biztosan egy fix bejegyzés marad, ebben az esetben is az alkalmazás az aggregátornak küldi el a beállítást, az asztali alkalmazásban mindig a mesterséges intelligencia által szolgáltatott adatok jelennek meg.

# MIVEL TUD TÖBBET MINT A TÖBBI PROJEKTKEZELŐ SZOFTVER?

**A Tracksy segíthet egy IT vállalat adminisztrációra fordított ideje csökkentésében, a hatékonyság növelésben.**

Funkcionalitásában egyik oldalról megegyezik más hasonló szoftverekkel, feladatokat lehet felvinni, manuálisan lehet azokra logolni.

**Azonban több olyan elem került kifejlesztésre a kutatási folyamat végére, amelyek teljesen egyedülállóvá teszik a Tracksy rendszert:**

- 1.** Automatán monitorozza egy felhasználó (mondjuk egy IT fejlesztő) tevékenységét, mikor, mivel foglalkozik, ezeket az adatokat más programokból nyerjük ki a bővítmények segítségével.
- 2.** Több megnyitott ablak esetén figyel, hogy melyik ablak az aktív, amely a munkavégzés meghatározása szempontjából nagyon fontos.
- 3.** Figyeli ha a felhasználó nem végez semmilyen tevékenységet a számítógépen, és ezt elfelejtette jelöli a rendszerben (megbeszélésen van, ebédelni ment stb).
- 4.** A Mesterséges Intelligencia pedig segít meghatározni azokat az időszávokat, amelyekhez nincsen megfelelő adatunk, és ez alapján javaslatot tesz az adott nap munkavégzésére és a feladatokra fordított időre.



# A TRACKSY FELHASZNÁLÁSA EGYÉB TERÜLETEN

Elsődleges célpiacunk az informatikai cégek, mert óriási lehetőséget látunk ebben a piacban, és kézenfekvő volt az adatforrások tekintetében is. Azonban nem szeretnénk megállni csak az informatikai területen.

A Tracksyt úgy készítettük el, hogy újabb és újabb adatforrások illeszthetők a rendszerhez, az aggregátor és a Mesterséges Intelligencia segítségével azok feldolgozhatóak lesznek.

Emellett nagy lehetőséget látunk a pozíció követésben, olyan területen, ahol nem mindig tartózkodik a munkavállaló a számítógépe előtt. Itt nehéz meghatározni automatikusan, hogy mikor, mivel foglalkozik.

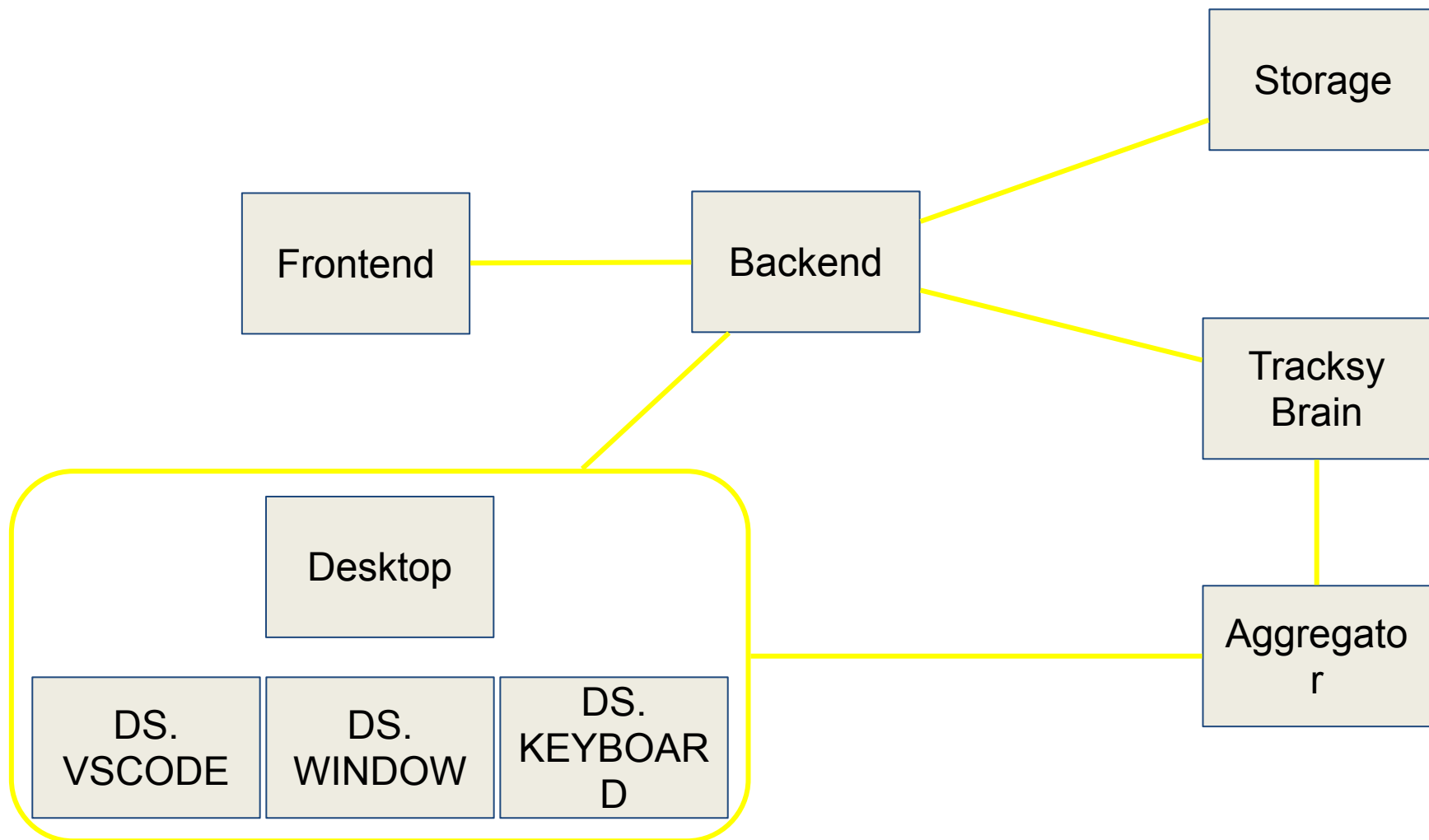
Azonban ha követjük a mozgást, és letároljuk, hogy mikor hol tartózkodott akkor ahhoz hozzá tudjuk rendelni, hogy milyen

tevékenységet végzett, így például egy feldolgozó üzemben is használható lenne a Tracksy automatikus munkaidő követésre

mind az irodai dolgozók, mind a fizikai munkások tekintetében.



# TECHNOLÓGIAI FELÉPÍTÉS



# TECHNOLÓGIA

## Back end

A NestJS egy keretrendszer hatékony, méretezhető Node.js webalkalmazások létrehozására. Modern JavaScriptet használ, TypeScript-szel épül fel, és az OOP (Object Oriented Programming), az FP (Functional Programming) és az FRP (Functional Reactive Programming) elemeit egyesíti.



## Front end

Frontend Framework a böngészőben futó adminisztrátori alkalmazások létrehozásához a REST/GraphQL API-kon felül ES6, React és Material Design használatával. Korábbi nevén admin-on-rest. Nyílt forráskódú.



## Asztali alkalmazás

Az Electron egy keretrendszer natív alkalmazások létrehozására olyan webes technológiákkal, mint a JavaScript, a HTML és a CSS.



## AGGREGÁTOR

Fastify egy gyors és hatékony webes alkalmazásfejlesztési keretrendszer JavaScript-ben, amelyet kifejezetten a Node.js számára terveztek. Kiemelkedően gyors teljesítményt, alacsony memórialhasználatot és könnyű skálázhatóságot kínál, valamint támogatja az aszinkron programozást és a middleware-ek használatát.

### Fastify

76835 req/sec ([Check the code](#))



### Koa

50933 req/sec ([Check the code](#))



### Express

38510 req/sec ([Check the code](#))



# STORAGE

```
const handler = require('serve-handler');
const http = require('http');

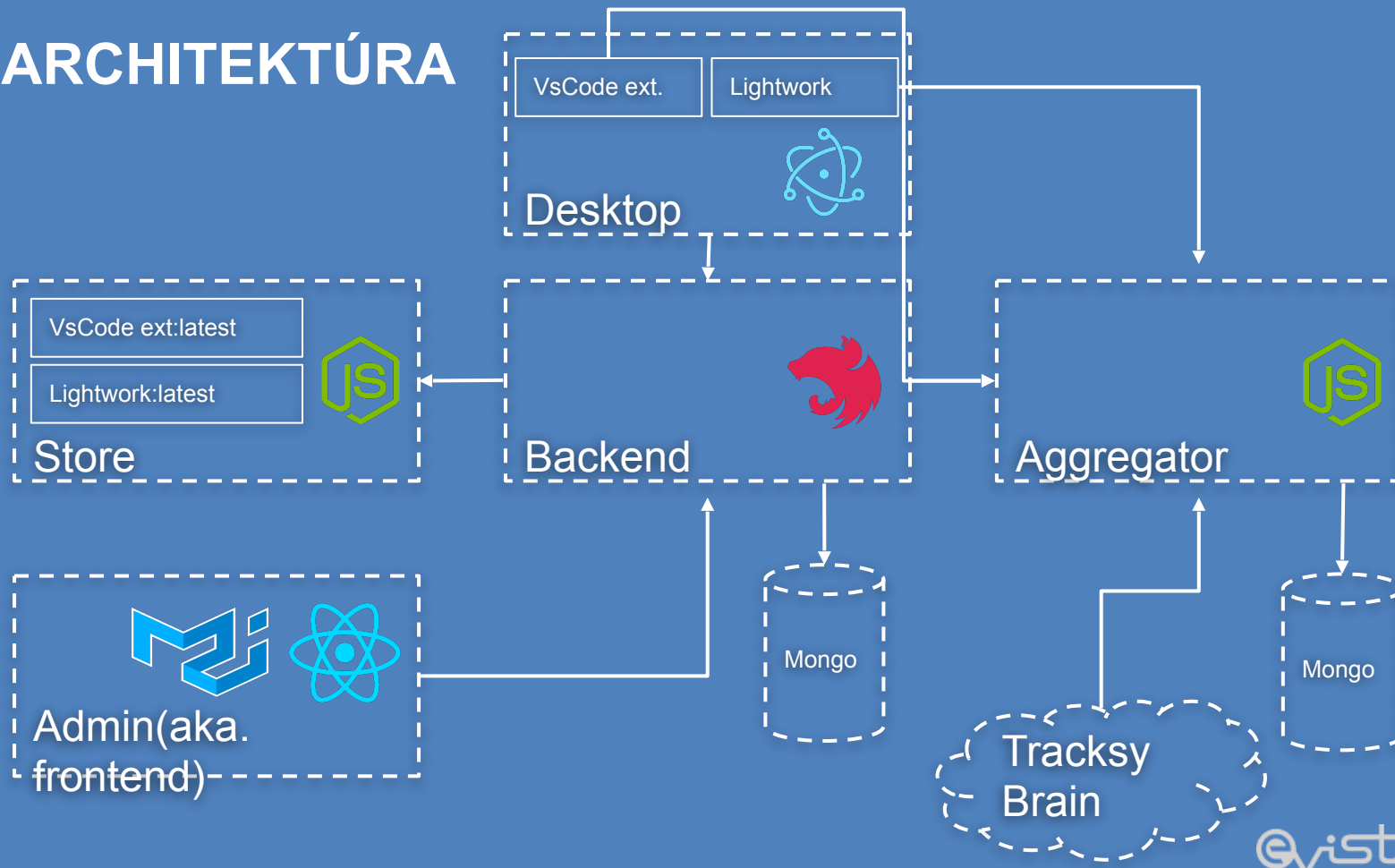
const server = http.createServer((request, response) => {
  // You pass two more arguments for config and middleware
  // More details here: https://github.com/zeit/serve-handler#options
  return handler(request, response, {
    unlisted: [
      "node_modules",
      "index.js",
      "docker-compose.yml",
      "package*",
      ".htpasswd"
    ]
  });
});

server.listen(3000, () => {
  console.log('FileServer is running');
});

```

# RÉSZLETES ARCHITEKTÚRA

## ARCHITEKTÚRA



# TECHNOLÓGIAI DÖNTÉSEK

## Általános:

- mikroszervíz architektúra
- konténerizáció
- CD/CI folyamatok használata
- REST API dokumentálás (Swagger)

## Komponensek:

- Backend: NestJS, typescript
- Admin: Webről elérhető felület, ReactJS + Redux + MaterialUI (React Admin)
- Aggregator: node, fastify
- Store: npm serve (node-os fileserver csomag)
- Desktop: ElectronJS, React

# ADATGYŰJTÉS VSCODE-AL

- Telepítés konfigurálással
- Események kezelése
- Git olvasása
- Config szinkronizálás
- Adatok küldése az aggregátornak
- Milyen problémákkal találoztunk?



# ADATGYŰJTÉS LITEWORK SEGÍTSÉGÉVEL

- Mi az a LiteWork
- Adatgyűjtés menete
- Milyen adatok, és milyen módon kerülnek elküldésre
- Milyen problémákkal találoztunk?

# AGGREGÁTOR BEMUTATÁSA

- Mi ez?
  - Mire használjuk?
  - Hogyan használjuk?
  - Hogyan használja a mesterséges intelligencia?
  - Milyen problémákkal találoztunk?
- 
- **SEGÉD SZERVIZEK BEMUTATÁSA**
  - Milyen szervizek vannak?
  - Mire használjuk?
  - Milyen jövőbeli tervek vannak?

# FORRÁSKÓD

Teljes forráskód és dokumentáció nyilvános elérhető a nyilvános gitlabunkon:

<https://gitlab.dev.e-vista.hu/tracksy>

**KÖSZÖNÖM  
A FIGYELMET!**

**TÓTH ANDRÁS  
EVISTA KFT.**

**SZÉCHENYI**  2020



MAGYARORSZÁG  
KORMÁNYA

**Európai Unió**  
Európai Regionális  
Fejlesztési Alap



**BEFEKTETÉS A JÖVŐBE**