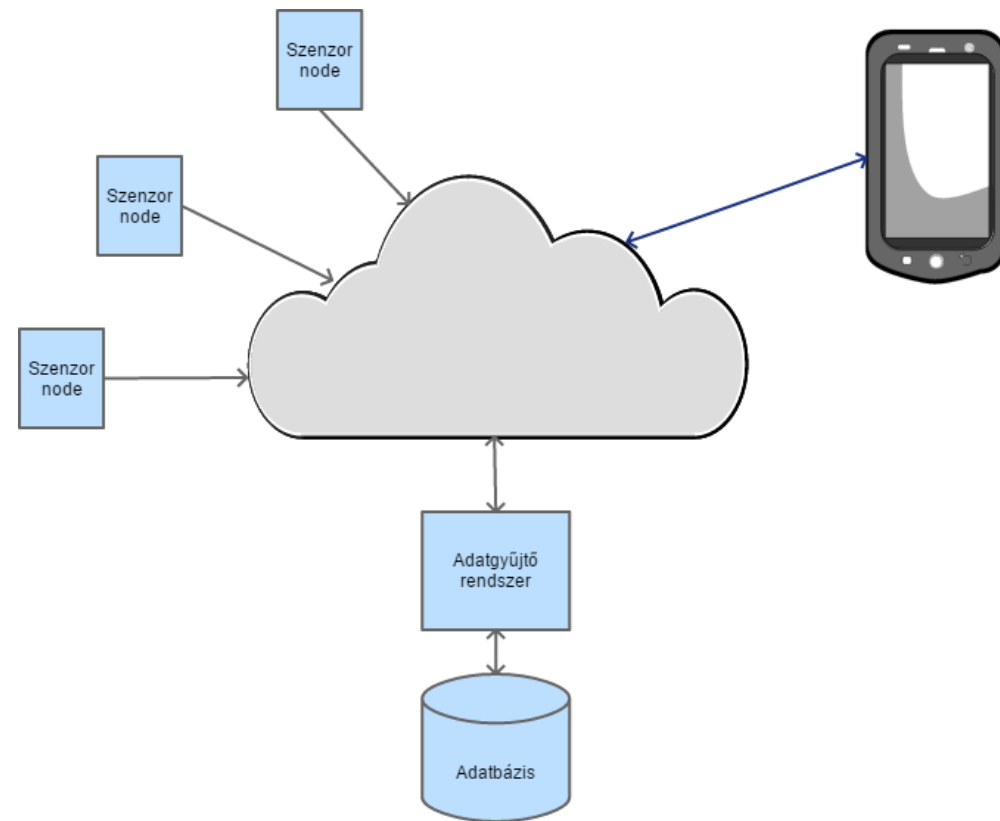


# API tervezése mobil környezetbe

gyakorlat

# Feladat

- Szenzoradatokat gyűjtő rendszer
- Mobil klienssel
- Webes adminisztrációs felület



# API „felhasználói”

- Szenzor node
  - Egyirányú adatküldés
  - Kis számítási kapacitás
  - Feltételezzük, hogy HTTPS-képes
  - Nincs humán interfész
- Telefonos kliens
  - Bizonytalan kapcsolat, esetleg drága is
  - Nagy számítási kapcsolat
- Webes felület
  - Web böngésző!
  - Lehet webes vékony-kliens
- Akkumulátoros táplálás (+napelem)
- Nehezen megközelíthető, távoli helyszínen
- Vezeték nélküli (bizonytalan) kapcsolat

# Megvalósítandó rész-funkciók

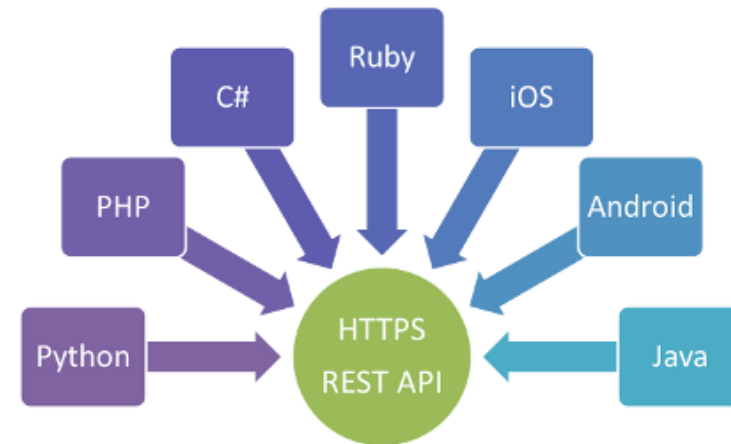
- A „felhasználók” azonosítása:
  - Nem akarjuk, hogy illetéktelenek adatokat olvassanak, illetve küldjenek be
- Adatok fogadása
- Adatok megjelenítése, törlése
- Az adatokat szolgáltató szenzorok nyilvántartása
- Felhasználók kezelése

# Ismétlés – REST API

- Egy API akkor lesz REST, ha teljesíti a következő megszorításokat:
  - Kliens-szerver architektúra
  - Állapotmentesség
  - Gyorsítótárazhatóság
  - Réteges felépítés (proxy szerverek támogatása)
  - Igényelt kód
  - Egységes interfész

# Átvitel: HTTPS

- HTTPS:
  - Használjuk a HTTP hibakódokat
  - Használjuk a titkosítást
  - Használjuk a hitelesítést



# Bejelentkezés

- A szenzor node-ok esetén nincs lehetőség felhasználónév, jelszó bekérésére minden adatküldéskor:
  - API key
  - Szenzor azonosító
  - Az API key és a szenzor azonosító védelme elvileg megoldható:
    - Flash titkosítás, kiolvasás megakadályozása stb.
- A mobil, illetve webes interfész esetén felhasználónév-jelszó alapú bejelentkezés

# Adatok küldése

- Lehet valósidejű, és nem valósidejű adatküldés
  - Utóbbi esetben időbélyeg is tartozik az adathoz.
- Hálózati kimaradások esetén köteget beküldés lehetséges
  - Overhead csökkentése érdekében
  - Gyorsabb beküldés – használjuk ki, ha van hálózat
- Mit tartalmaz egy mérés?
  - Időbélyeg – explicit, vagy implicit (aktuális adat)
  - Mért érték
  - Mérő eszköz azonosítója
  - Esetleg metaadatok (például mértékegység, a mérési körülményekre vonatkozó adatok stb.) – nem foglalkozunk vele



# Szenzorok, mérések, felhasználók nyilvántartása

Művelet	RESTful Web Services HTTP művelet (method)
Create	POST*
Read	GET
Update	PUT*
Delete	DELETE

\* Nem csak így jó, az [URL, HTTP művelet] – [resource, művelet] összerendelés nem triviális.

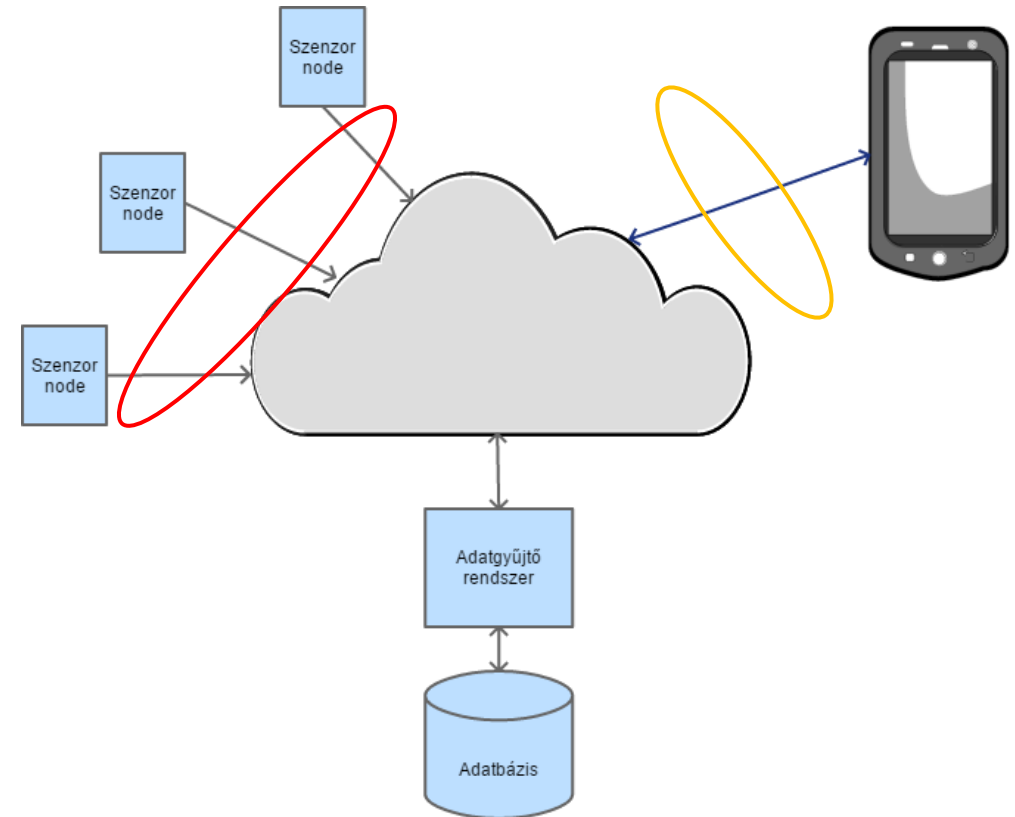
# Adatok átvitele általánosan

- Strukturált adat
  - Könnyebb feldolgozás
  - Bővíthetőség
    - Újabb metaadatok
    - Visszafele kompatibilitás biztosítása újabb verziók esetén

(adatbázis kialakítása: szeretjük az új attribútumok megjelenését)

# Egy API?

- Két API-t tervezünk
  - Egy egyszerű adatfogadó API a szenzor node-okhoz
  - Egy CRUD műveleteket tartalmazó API a humán felhasználású felületekhez



# API URL-ek

- Szenzor node API
  - /sensor\_api/
  - Adatfogadó végpont:
    - /sensor\_api/measurement/new [POST]
    - /sensor\_api/bulk/measurements [POST]

Művelet	RESTful Web Services HTTP művelet (method)
Create	POST*
Read	GET
Update	PUT*
Delete	DELETE

- Web és mobil API
  - /api
  - Autentikáció, token-generálás
    - /api/login
  - Adatműveletek:
    - /api/<adat típusa>/new [POST]
    - /api/<adat típusa>/<id> [GET]
    - /api/<adat típusa>/<id> [DELETE]
    - /api/<adat típusa>/<id> [PUT]
    - /api/<adat típusa többszám> [GET]

# Szenzor felület

## /sensor\_api/measurement/new

- A szenzor node-ok küldenek pontosan egy darab adatot

```

{
  "sensor_node_id": "00:11:22:33:44:55",
  "api_key": "Y2lybW9zY2ljYWwhhaixob3ZhbGV0dGF2YWwo",
  "measurement": {
    "sensor_node_id": "00:11:22:33:44:55",
    "value": 15,
    "meta": {
      "unit": "°C"
    }
  }
}

```

Ez jobb helyen lehet egy request headerben

# Kitérő: API key küldése

- URL:

GET /new ?api\_key=Y2lybW9zY2ljYWwhhaixob3ZhbGV0dGF2YWwo

- HTTPS használata esetén a TCPpayload-ban SSL titkosított URL: ...Ø.a].b©..
- Szerver logban: /new ?api\_key=Y2lybW9zY2ljYWwhhaixob3ZhbGV0dGF2YWwo
- (browser history, referer request header: kép, js, analytics)

- Request header:

GET /new HTTP/1.1

X-API-Key: Y2lybW9zY2ljYWwhhaixob3ZhbGV0dGF2YWwo

- Cookie:

GET /new HTTP/1.1

Cookie: X-API-KEY=Y2lybW9zY2ljYWwhhaixob3ZhbGV0dGF2YWwo

# Szenzor felület

## /sensor\_api/measurement/new

- Mi legyen a válasz?
  - Ki kell, hogy derüljön a művelet sikeressége.
  - Kis sávzélesség, nem küldünk fölöslegesen adatot.

```
{ "result": "ok" }
```

# Szenzor felület

/sensor\_api/bulk/measurements/

```
{
  "sensor_node_id": "00:11:22:33:44:55",
  "api_key": "Y2lybW9zY2ljYWWhhaixob3ZhbGV0dGF2YWo ",
  "measurement": [
    {
      "sensor_node_id": "00:11:22:33:44:55",
      "timestamp": "2018-09-12T12:14:00Z",
      "value": 15,
      "meta": {
        "unit": "°C"
      }
    }
  ]
}
```



# Web (és mobil) felület – bejelentkezés

- /api/login



```
{ "username": "sandor", "password": "piros7es" }
```



```
{ "token": "xZByw7xsdCBtaW5kZW4gZsWxc3rDoWwu" }
```

# Webes felület – CRUD műveletek – Create

- /api/<adat típusa>/new [POST]
- A szenzor felület beküldő interfészéhez teljesen hasonlóan, az autentikációs tokennek szerepelnie kell: betesszük egy HTTP kérés fejlécbe. A hozzáférési jogosultság ellenőrzése az alapján történik
- Mivel itt a sávszélesség megengedi, és haszna lehet, új elem esetén visszaküldhető az új létrehozott objektum, vagy annak egyedi azonosítója (később ezzel hivatkozhatunk rá)

# Webes felület – CRUD műveletek – Read, Update

- /api/<adat típusa>/<id> [GET, PUT]
- Kellhet a token, az adatra az egyedi azonosítójával hivatkozunk
- GET esetén nincs beküldött adat, ezért is praktikus a fejlécbe tenni a tokent.

# Webes felület – tételek listázása

- Semmi különös, tömbbel térünk vissza
- Mi történik, ha sok az adat?
- Lapozás – paging, adatok szűrése
  - Ha nem érvényes a szűrő kifejezés, visszatérés hibával
- GET paraméterekkel
- `/api/measurements?from=2017-12-12T10:00:00Z&until=2018-01-12T10:00:00Z`

# Webes felület – tételek listázása, másként

- Gyakori lekérdezések entitásokhoz kapcsolódóan
- `/api/sensor/5/measurements`
- `/api/measurements?sensor=5`
- Nem egy jó megoldás létezik

# Az API hívások eredménye – hibák

- HTTP hibakódokkal kezelhetők
  - 200 OK
  - 201 Created
  - 401 Unauthorized
  - 403 Forbidden
  - 404 Not Found
  - 418 I'm a teapot
  - 500 Internal Server Error