

# IoT 02

Felde Imre

# Vázlat

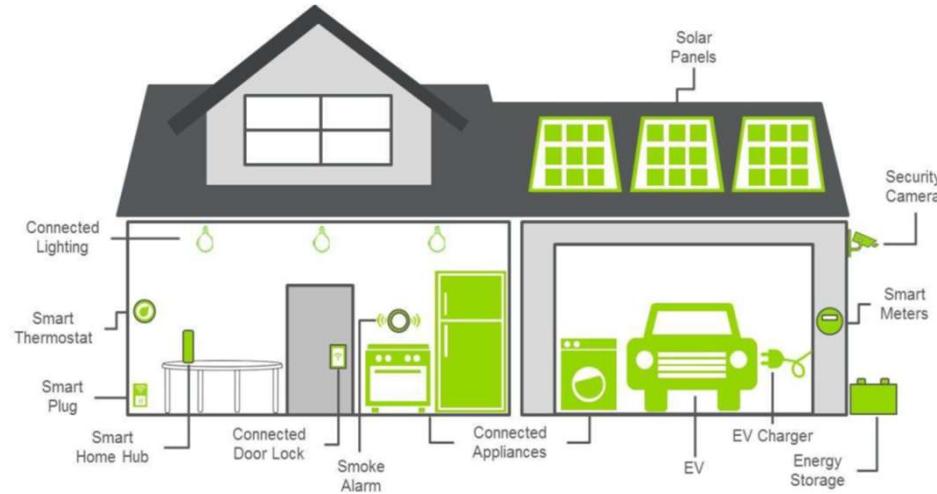
- Architektúra Terv
- 3-rétegű architektúra
- 4-rétegű architektúra
- oneM2M Architektúra
- IoTWF Architektúra

# Mi az az Architektúra Terv?

- Napjainkban a hálózatok működtetik a modern üzletet.
- Ezért soha nem szabad gondos tervezés nélkül felépíteni őket.
- Az architektúra az, ahogyan az alkalmazást vagy megoldást meghaladva megtervezzük.

## Az IoT architektúra lényege:

- hogyan történik az adatok továbbítása,
- gyűjtése,
- elemzése, és
- végül a rájuk adott válasz.

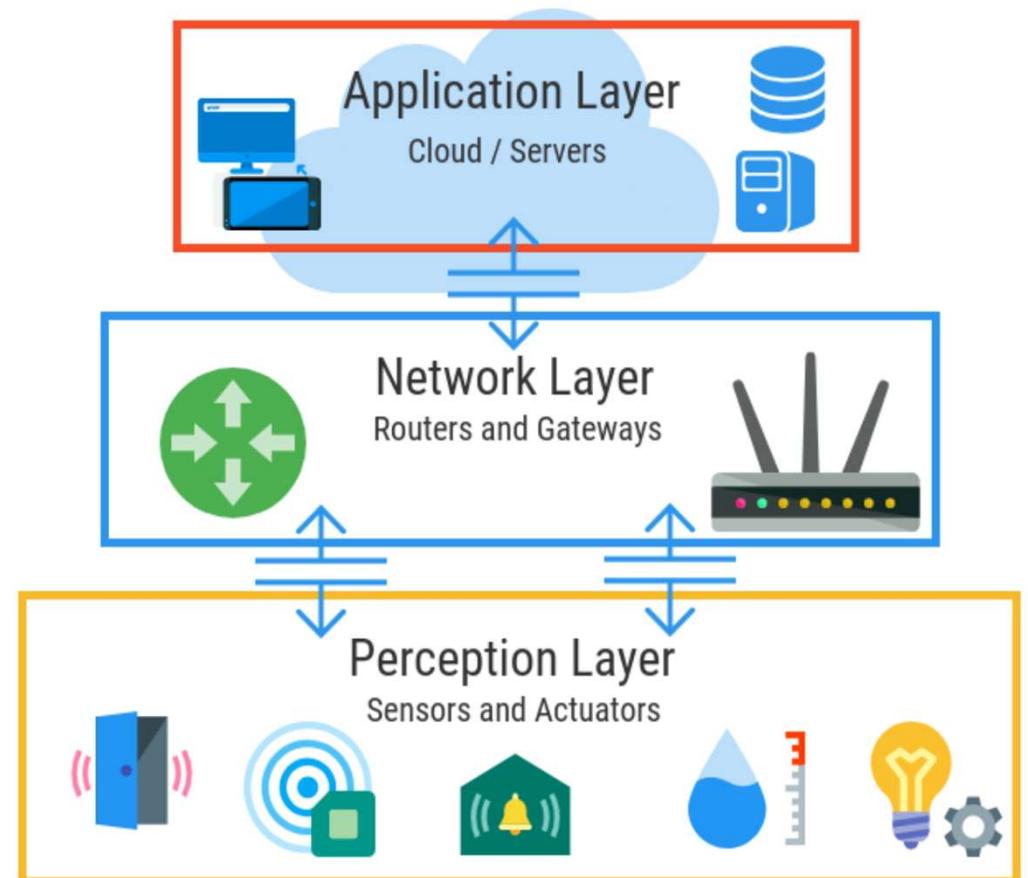


## Hajtóerők:

- Méretezhetőség
- Biztonság
- Korlátolt eszközök
- Nagy mennyiségű adat
- Adat-elemzés
- Régi eszközök támogatása

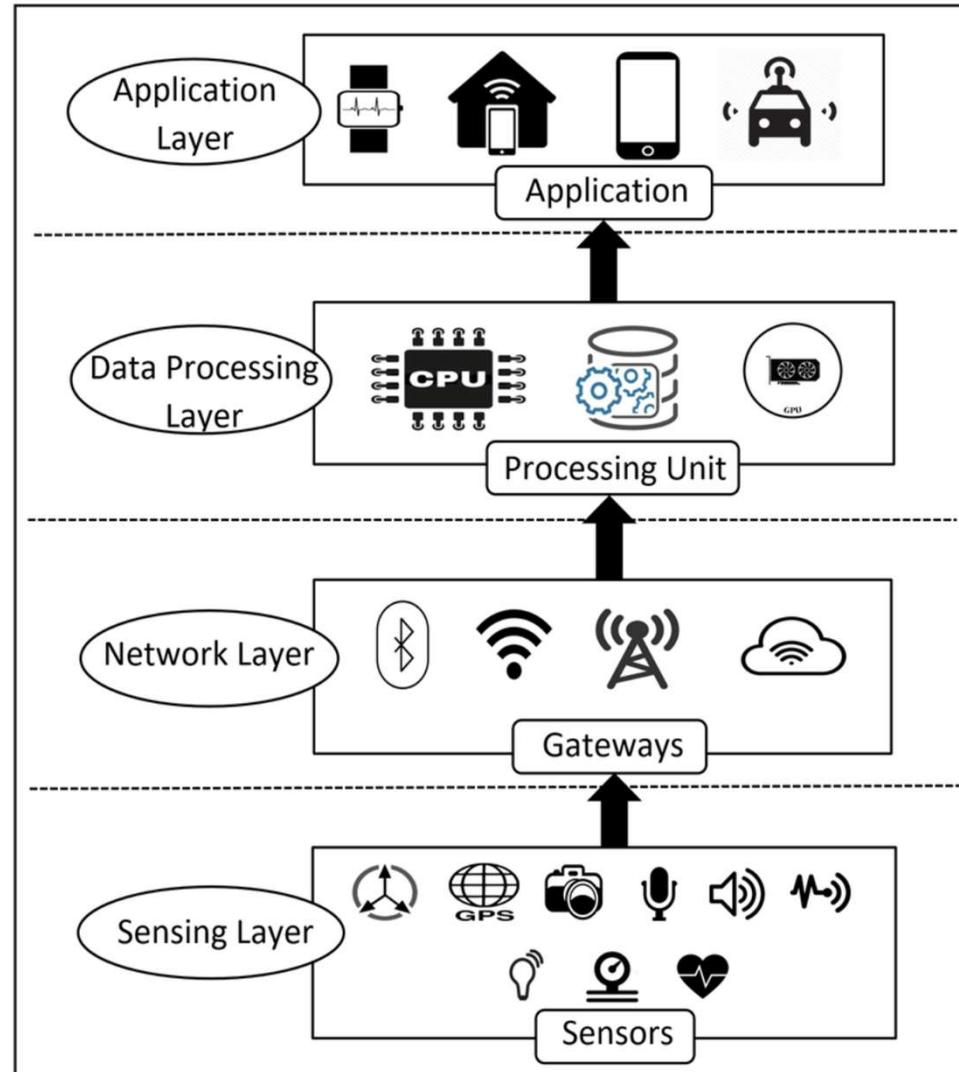
# 3-rétegű architektúra

- Az érzékelési réteg (**Perception layer**) a fizikai réteg, amely érzékelőket tartalmaz az információk észlelésére és gyűjtésére.
- A hálózati réteg (**Network layer**) felelős más intelligens eszközökhöz, hálózati berendezésekhez és szerverekhez való kapcsolódásáért, valamint az érzékelő adatok továbbításáért és feldolgozásáért.
- Az alkalmazási réteg (**Application layer**) az alkalmazásspecifikus szolgáltatásokat nyújtja a felhasználónak.
  - Például: okos otthonok, okos városok, okos egészségügy stb.



# Gyakran használt 4-rétegű architektúra

- **Alkalmazási réteg:** Olyan szoftvereket és alkalmazásokat tartalmaz, mint mobilappok vagy webes felületek.
- **Adatfeldolgozási réteg:** Adatkezelő rendszereket, analitikai platformokat és gépi tanulási algoritmusokat alkalmaz.
- **Hálózati réteg:** A kommunikációt és kapcsolatot biztosító protokollokat és technológiákat tartalmazza.
- **Érzékelési réteg:** Érzékelőket és beavatkozókat tartalmaz az adatok gyűjtésére.



# A szabványos IoT-architektúra kialakulása

- A 3- vagy 4-rétegű alap architektúrák számos problémát nem kezeltek.
- Az IoT-nek sokáig nem volt univerzálisan elfogadott, egységes architektúrája



## Problémák:

- Széttagolt, gyártóspecifikus alkalmazások
- Interoperabilitás hiánya
- Inkompatibilitás a heterogén rendszerek között
- A biztonság nem volt központi elem

Az utóbbi években néhány szabványos keretrendszer megjelent:

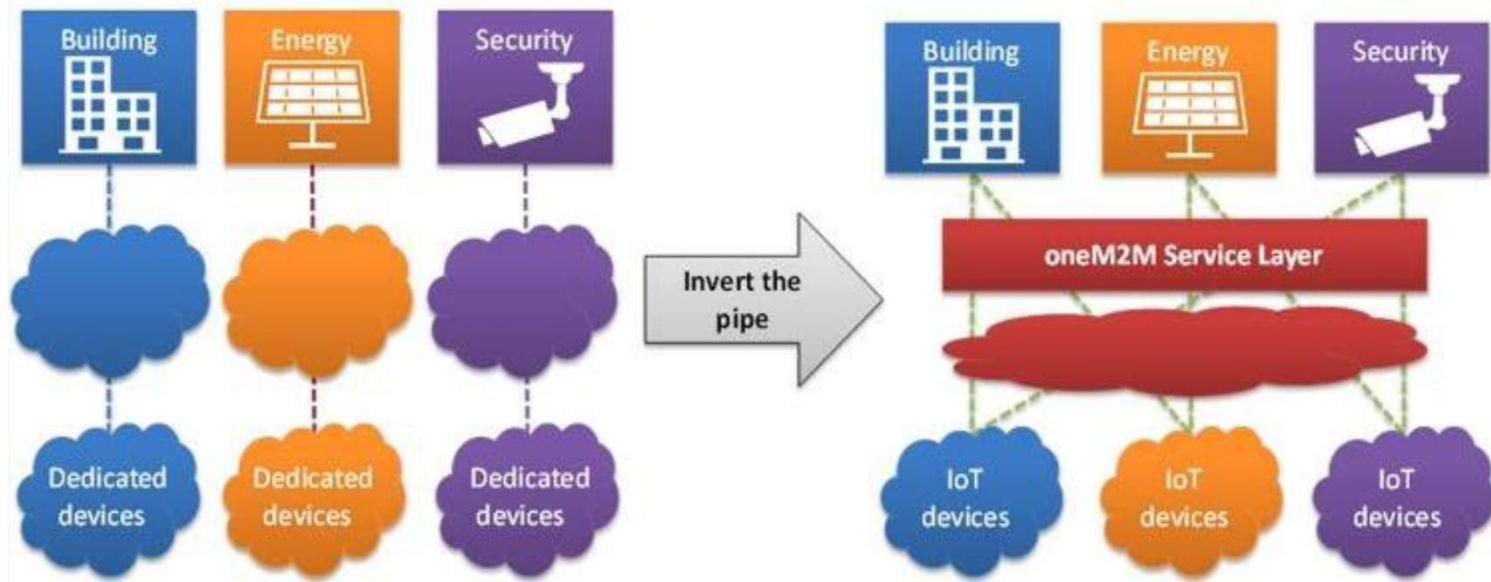
- **oneM2M architektúra**
- **IoT World Forum architektúra**

# oneM2M Architektúra

**Cél:** Egységes szolgáltatási réteg létrehozása, amely kommunikálni tud az alkalmazásszerverekkel

Kihívások:

- Eszközök heterogenitása
- Szoftverek sokfélesége
- Hozzáférési módszerek különbözősége

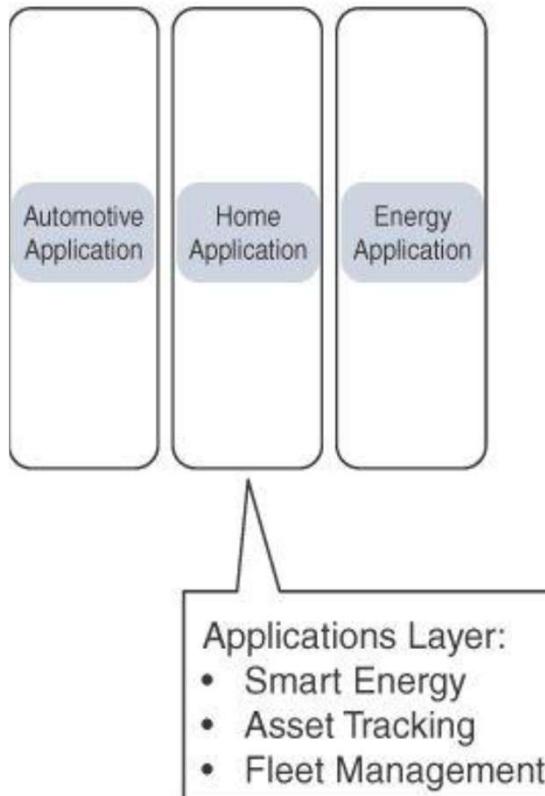


Példa: Okos épület – ha senki sincs bent, a rendszer lekapcsolja a világítást és visszaveszi a HVAC működését

# oneM2M Architektúra – 2

- Az ETSI (European Telecommunications Standards Institute) javaslata.
- Három fő doménre osztja az IoT funkciót.

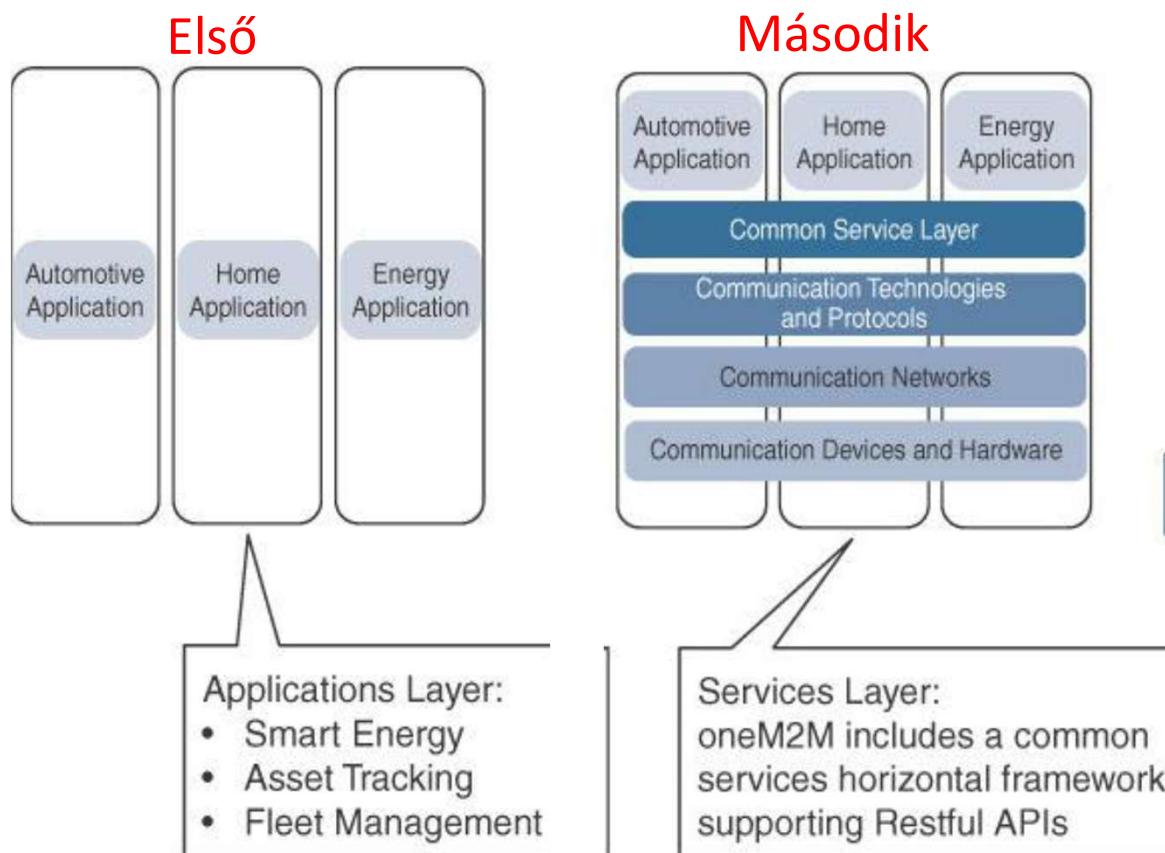
Első



- Meghatározza az alkalmazási réteg protokolljait, egységesíti az északi API-kat.
- API = szolgáltatási szerződés két alkalmazás között.
  - Az interfész úgy fogható fel, mint egy **szolgáltatási szerződés két alkalmazás vagy fél között**. Ez a szerződés határozza meg, **hogyan kommunikálnak egymással**, vagyis **hogyan küldenek és fogadnak kéréseket és válaszokat**.
- Az északi interfész a hálózati komponens és egy magasabb szintű komponens közötti kapcsolat.
- Az alkalmazások saját adatmodellekkel rendelkeznek.

# oneM2M Architektúra – 3

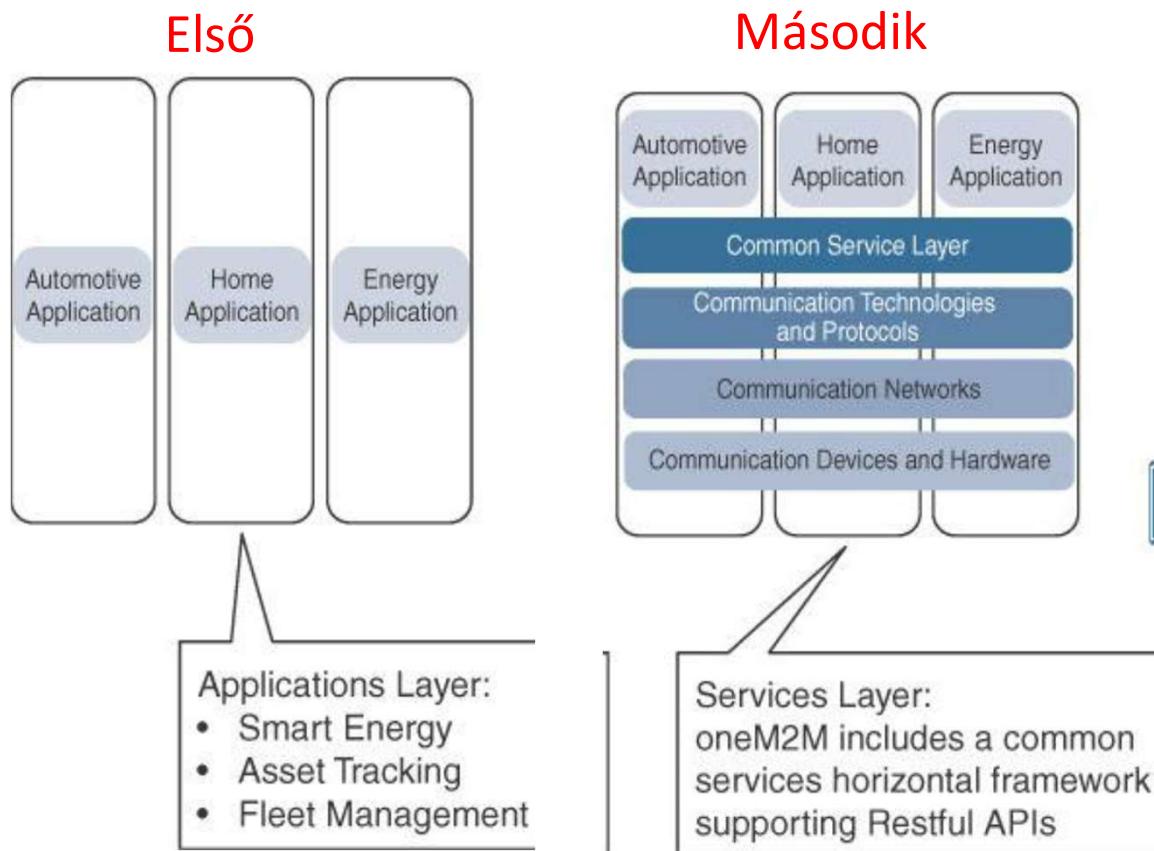
- Az ETSI (European Telecommunications Standards Institute) javaslata.
- Három fő doménre osztja az IoT funkciót.



- Vízszintes keretrendszert hoz létre a vertikális ipari alkalmazások között.
- Tartalmazza a fizikai hálózatot (pl. backhaul), a menedzsment protokollokat és a hardvert is.

# oneM2M Architektúra – 4

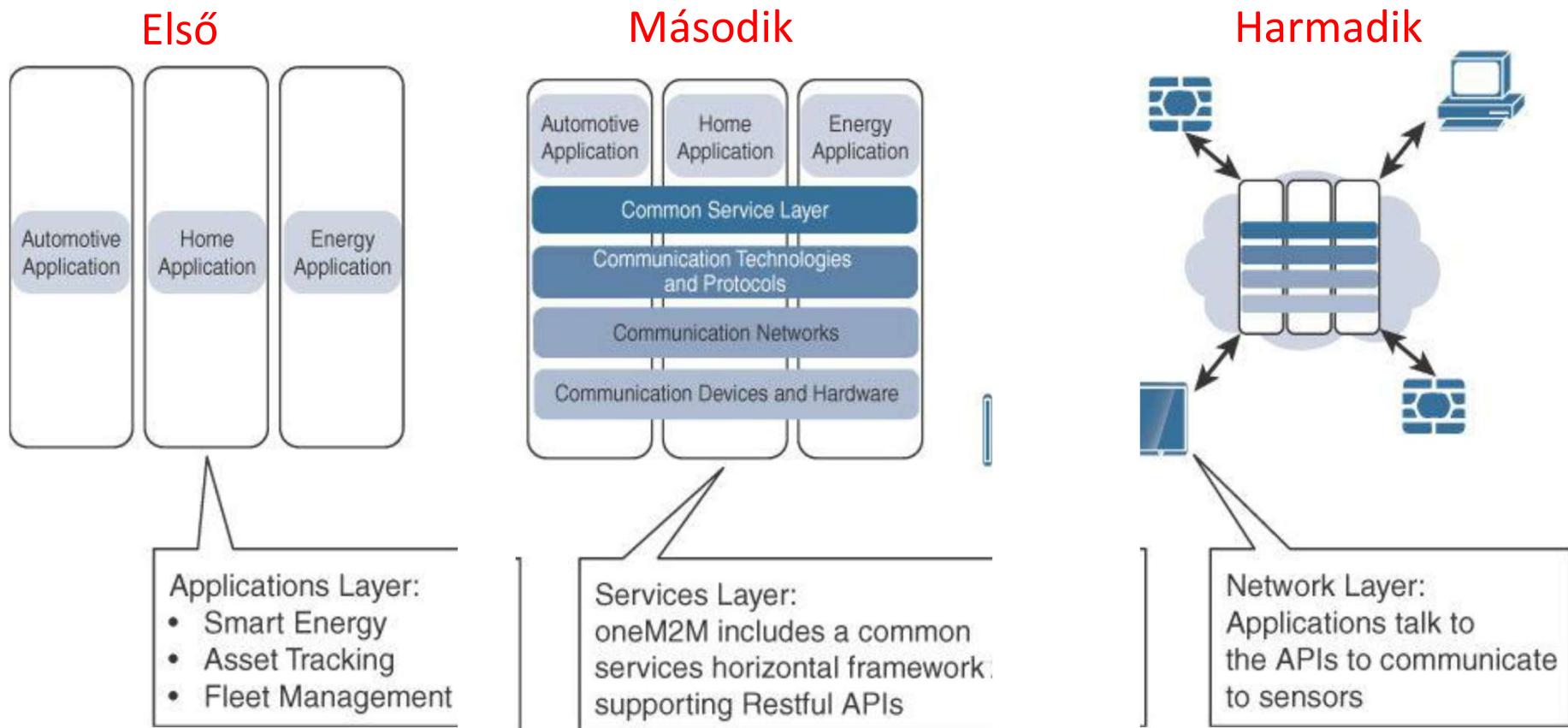
- Az ETSI (European Telecommunications Standards Institute) javaslata.
- Három fő doménre osztja az IoT funkciót.



- A felső réteg a közös szolgáltatási réteg (Common Services Layer).
- Ez API-kat és middleware-t biztosít harmadik féltől származó szolgáltatásokhoz.
- A szolgáltatási réteg beágazható különféle hardver- és szoftverkomponensekbe.
- RESTful API: HTTP GET, PUT, POST, DELETE műveletek.

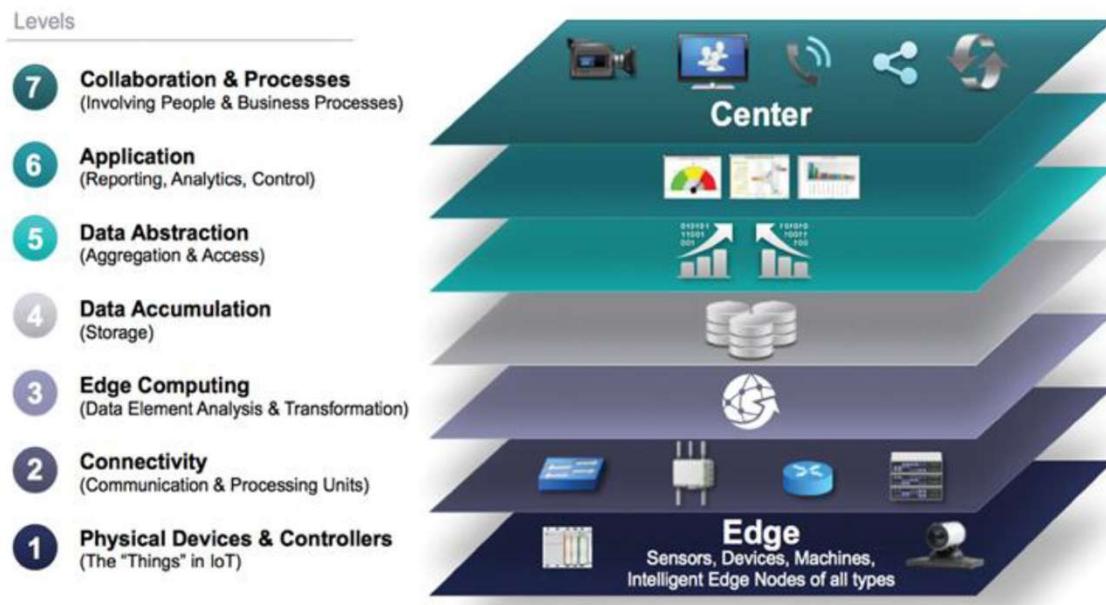
# oneM2M Architektúra – 5

- Az ETSI (European Telecommunications Standards Institute) javaslata.
- Három fő doménre osztja az IoT funkcióit.



# IoTWF Architektúra – 7 réteg

- Cél: Egyszerű, egységes IoT-modell létrehozása (Cisco, IBM stb.).
  - Tartalmazza az edge computingot, adattárolást, hozzáférést
  - Többszintű biztonsági modellt alkalmaz
  - Rétegenként más technológiákat különít el
  - Interoperabilitás interfészek definiálásával



## Az IoT-probléma lebontása **kisebb részekre**

- Azonosítsd a **különböző technológiákat** minden egyes rétegben
- A rendszer **különböző részeit** különböző **gyártók** biztosíthatják
- **Többszintű biztonsági modell** kerül alkalmazásra a szintek közötti átmeneti pontokon
- **Interoperabilitást biztosító interfészeket** kell meghatározni

# 1–2. réteg

## 1. réteg: Fizikai eszközök és vezérlők

- Az IoT „dolgai” – szenzoruktól a nagy gépekig.
- Adatokat generálnak, és hálózaton keresztül vezérelhetők.

## 2. réteg: Kapcsolódási réteg (Connectivity)

- A hálózati kapcsolat biztosítására szolgál

### Layer 2 Functions:

- Communications Between Layer 1 Devices
- Reliable Delivery of Information Across the Network
- Switching and Routing
- Translation Between Protocols
- Network Level Security



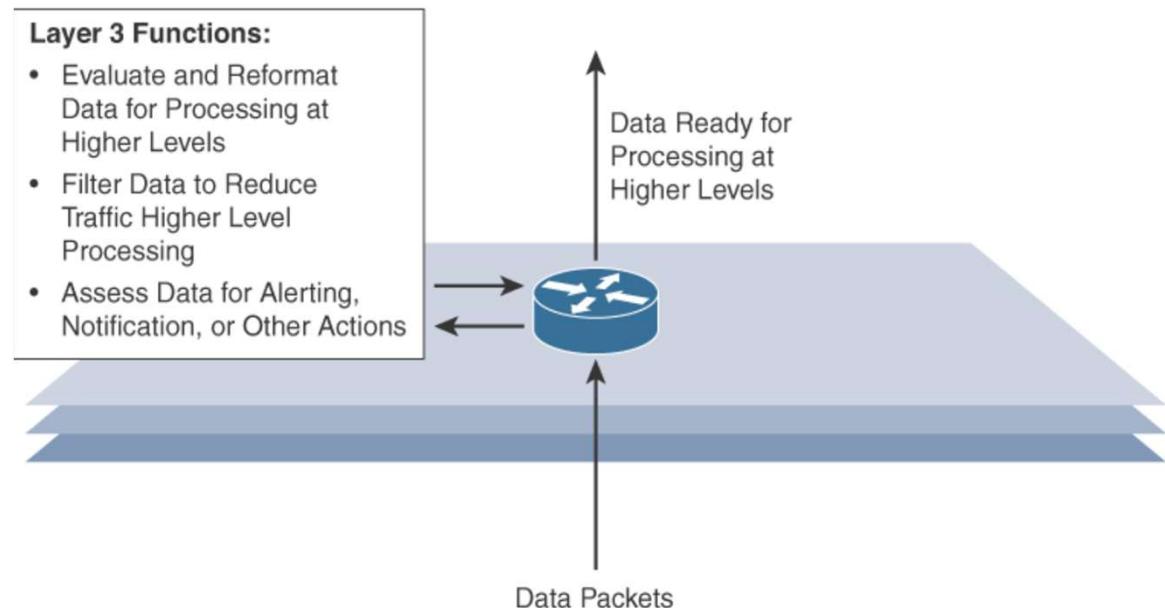
# 3. réteg – Peremhálózat (Fog Layer)

Kód” rétegnak is nevezik.

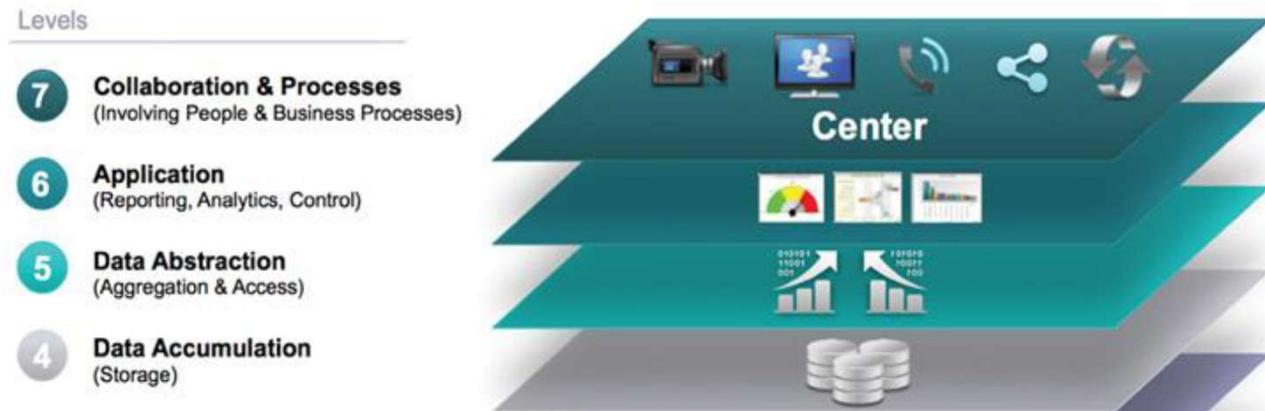
Feladatok:

- Adatok szűrése, tisztítása
- Újraformázás, tömörítés
- Elsődleges feldolgozás (pl. riasztás generálása)

Elv: Az információ feldolgozása minél közelebb történjen a hálózat széléhez.



## 4–7. rétegek



- **4. Adatgyűjtés:** az adatok rögzítése és tárolása.
- **5. Adatabsztrakció:** az adatok egységesítése és ellenőrzése.
- **6. Alkalmazás:** az adatok értelmezése, elemzése, jelentések.
- **7. Együttműködés és folyamatok:** az információ megosztása.

# Egységesített IoT Architektúra

- Kiemeli azokat az **alapvető építőelemeket**, amelyek a legtöbb IoT-rendszerben **közösek**, és amelyek célja, hogy **segítsenek egy IoT-hálózat megtervezésében**.
- Két fő rétegre bontható:
  - Alap IoT funkcionális réteg
  - Adatkezelési és számítási réteg

