

# **Rendszerkapcsolatok nyilvántartása és vizualizációja Neo4j segítségével a CERN-nél**

**Csikós Donát – Gradle Inc.**

# A CERN madártávlatból



# A kísérletek nagyságrendje



# CERN Control Centre



# Monitorozás – ha probléma van

- cs-ccr-builds
- cs-ccr-ciet12
- cs-ccr-ciet13
- cs-ccr-ciet34
- cs-ccr-ciet45
- cs-ccr-ciet56
- cs-ccr-ciet67
- cs-ccr-ciet78
- cs-ccr-ciet81
- cs-ccr-circuit1
- cs-ccr-cis
- cs-ccr-cmw1**
- cs-ccr-cmw2
- cs-ccr-cmw3
- cs-ccr-cmw4
- cs-ccr-cmw5
- cs-ccr-coca
- cs-ccr-colsa
- cs-ccr-colsa3
- cs-ccr-cpssis
- cs-ccr-ctb06
- cs-ccr-ctb07
- cs-ccr-ctf1

bmu20s	bmu30s	bmu40s	bmu50s
bmu60s	cerntsbe05	cerntsbe06	cerntsbe07
cerntsbe08	cerntsbe09	cerntsbe10	cerntsbe11
cerntsbe100-1	cerntsbe101-1	cerntsbe102-1	cerntsbe103-1
cerntsbe100-2	cerntsbe101-2	cerntsbe102-2	cerntsbe103-2
cerntsbe1b01-1	cerntsbe1b01-2	cerntsbe1b02-1	cerntsbe1b02-2
cerntsbe1b02-3	cerntsbe1b03-1	cerntsbe1b04-1	cfc-112-r1
cfc-150-rpow	cfc-152-cgsrcl	cfc-152-cln4aos1	cfc-152-cln4aos4
cfc-152-r1	cfc-152-r1n4pow1	cfc-163-r1	cfc-163-r2
cfc-170-arexplcgw	cfc-170-rplc	cfc-181-r1	cfc-193-allrf2
cfc-193-ascl	cfc-193-rpow1	cfc-193-rpow2	cfc-193-tadscrap
cfc-195-mkadhorn	cfc-197-bisobeam	cfc-197-rplcgps	cfc-197-rplchrs
cfc-197-rplclin	cfc-197-rplcrex	cfc-197-tislplcgw	cfc-2001-agw
cfc-2001-bmtv	cfc-2001-rpow1	cfc-2010-bpm	cfc-2010-ccaos4
cfc-2010-rpow1	cfc-2013-a12gts2	cfc-233-rpow	cfc-250-rpow
cfc-251-rpseasthal1	cfc-263-rpseastarea	cfc-269-cpaos13	cfc-269-cstep
cfc-269-rpsft16	cfc-281-r1	cfc-351-cgen	cfc-351-rpow
cfc-353-bpmops	cfc-354-rpsft16	cfc-355-cpaos12	cfc-355-rpseccloud
cfc-355-rpspfw	cfc-355-rpsring	cfc-358-rpsa	cfc-359-cstep
cfc-359-rsepta	cfc-361-allcav	cfc-361-cbaos1	cfc-361-cbaos11

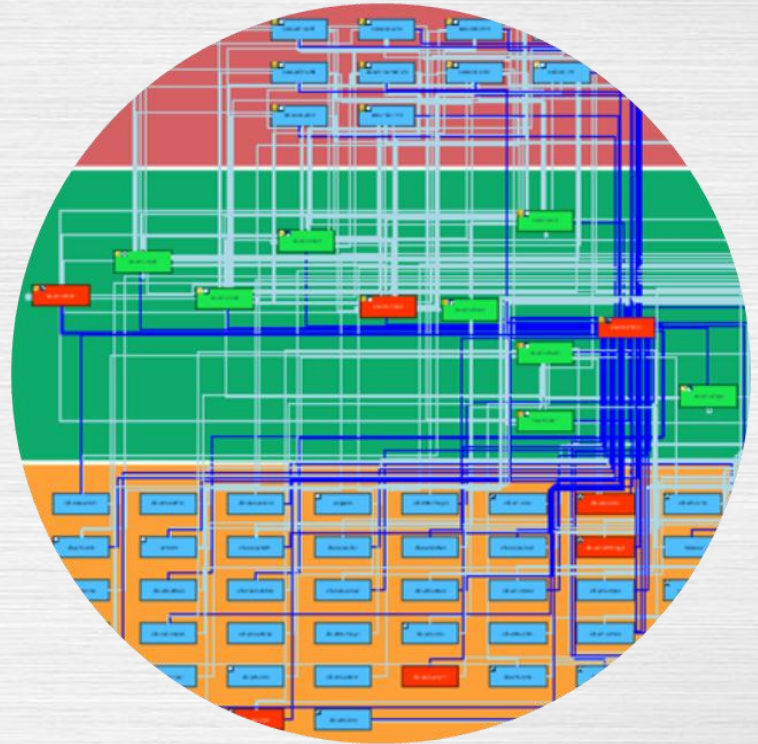
## cs-ccr-cmw1 (CMW main server)

- General
- Metric
- Services**
- Processes
- MX
- Config
- MOTD
- CLIC State

name	seq	description	prio
CMW-DIRECTORY-PRO1			
CMW-DBA-ACC.jvm			
CMW-LEGACY-CONV.jvm			
CMW-RBAC-PRO1.jvm			6
CMW-DIP-SERVICES.jvm			

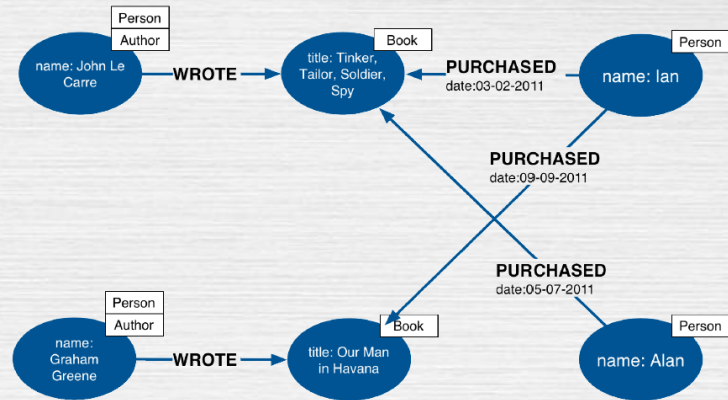
# Az ACET-DEPS projekt

- Vizualizáljuk a rendszerek globális állapotát
  - Adatgyűjtés + rendszeres frissítés
- Nem pontosan specifikált domain



# Miért Neo4j? (1/3)

- Gráfadatbázisok jól illeszkedtek a problémára
  - Szemi-struktúrált adatkezelés
- Első tesztek alapján a legjobb választás
  - Képességek, teljesítmény, API, dokumentáció, licenz



# Miért Neo4j? (2/3)

- Java interfész

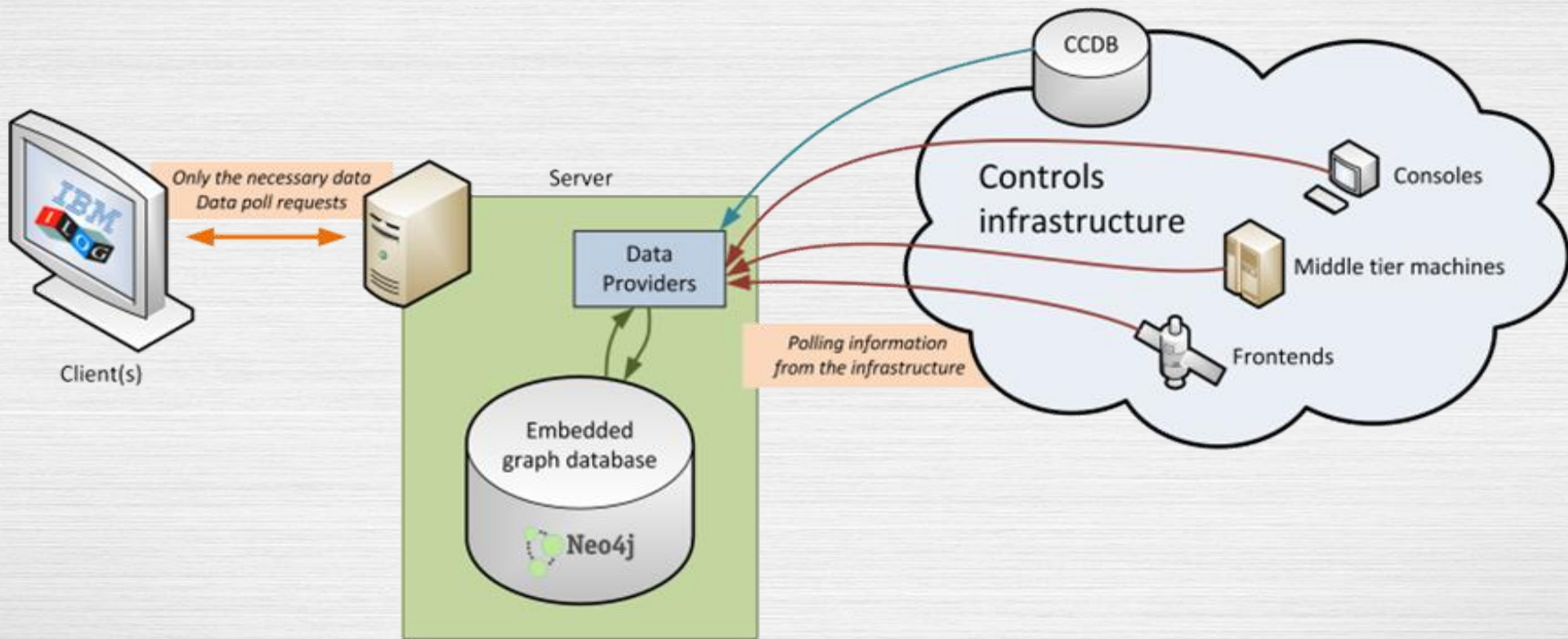
```
GraphDatabaseService graphDb = new
GraphDatabaseFactory().newEmbeddedDatabase(new File("/dev/neo4jdb"));
Transaction tx = graphDb.beginTx();
try {
    Node author = graphDb.createNode();
    author.setProperty("name", "Alan");
    Node steve = graphDb.createNode();
    steve.setProperty("name", "Steeve");
    Relationship relationship = author
        .createRelationshipTo(steve, RelTypes.KNOWS);
    relationship.setProperty("since", "2014-12-17");
    tx.success();
} finally {
    tx.close();
}
```



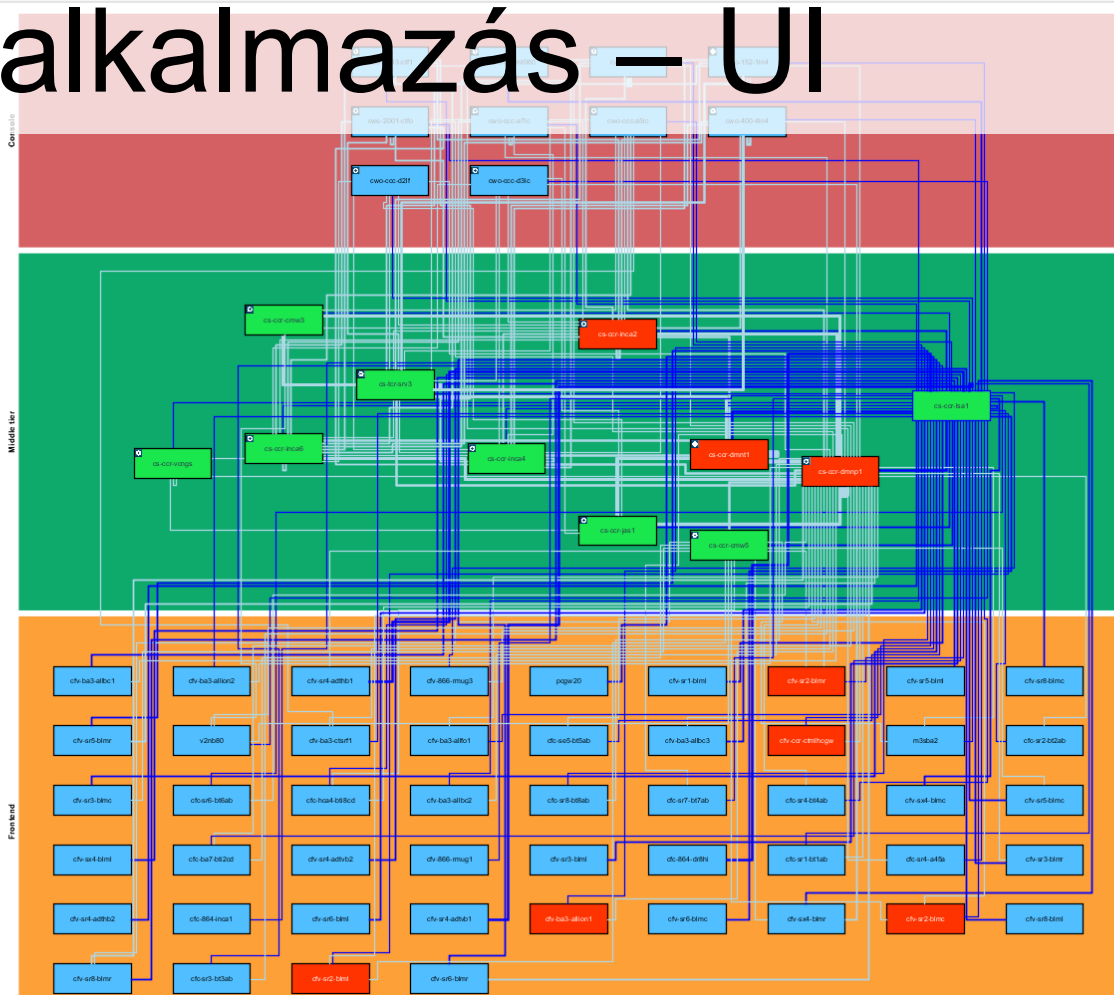
# Miért Neo4j? (3/3)

- Deployment
  - EmbeddedGraphDatabase,  
ImpermanentGraphDatabase
  - A fejlesztők adatbázis adminon is

# Kész alkalmazás - Architektúra



# Kész alkalmazás – UI



# Tapasztalatok

- Neo4j
  - A mi projektünkhöz bevált
  - Agilis, intuitív
  - „Elég gyors”

# Köszönöm a figyelmet!

- Kapcsolat:
  - GitHub: <http://github.com/donat>
  - LinkedIn: <https://hu.linkedin.com/in/donatcsikos>
  - Twitter: @donatcsikos