

Elosztott tartalomszolgáltatás

,avagy hogyan szolgáljunk ki területileg elkülönülő
weboldalakat több adatközpontból

Budapest, 2013 November 9.

Czimmermann Gábor

Ki vagyok?

- * Az ICR-System Kft. fejlesztési vezetője
- * Elosztott adattárolással és programozással foglalkozom
- * Főleg Java és Ruby nyelveken művelem a fentieket

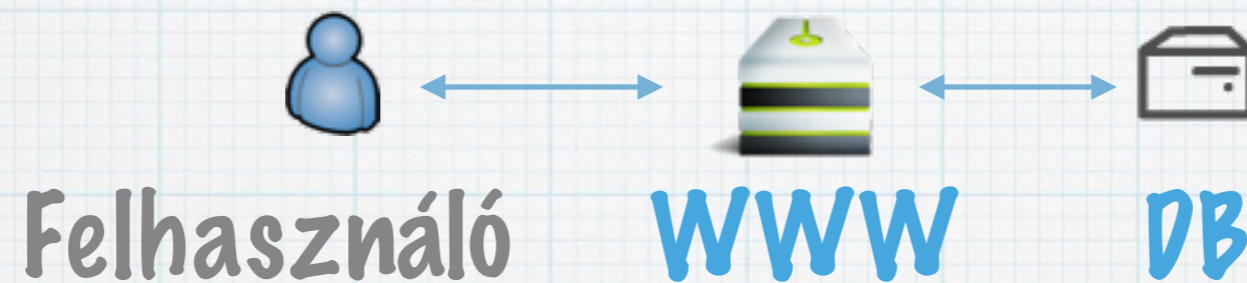
Mi lesz a mai témánk?

- * Miért jók az elosztott adattároló eszközök?
- * Hogyan használjuk őket?
- * Hogyan használjunk replikációt, particionálást és mire jók ezek?

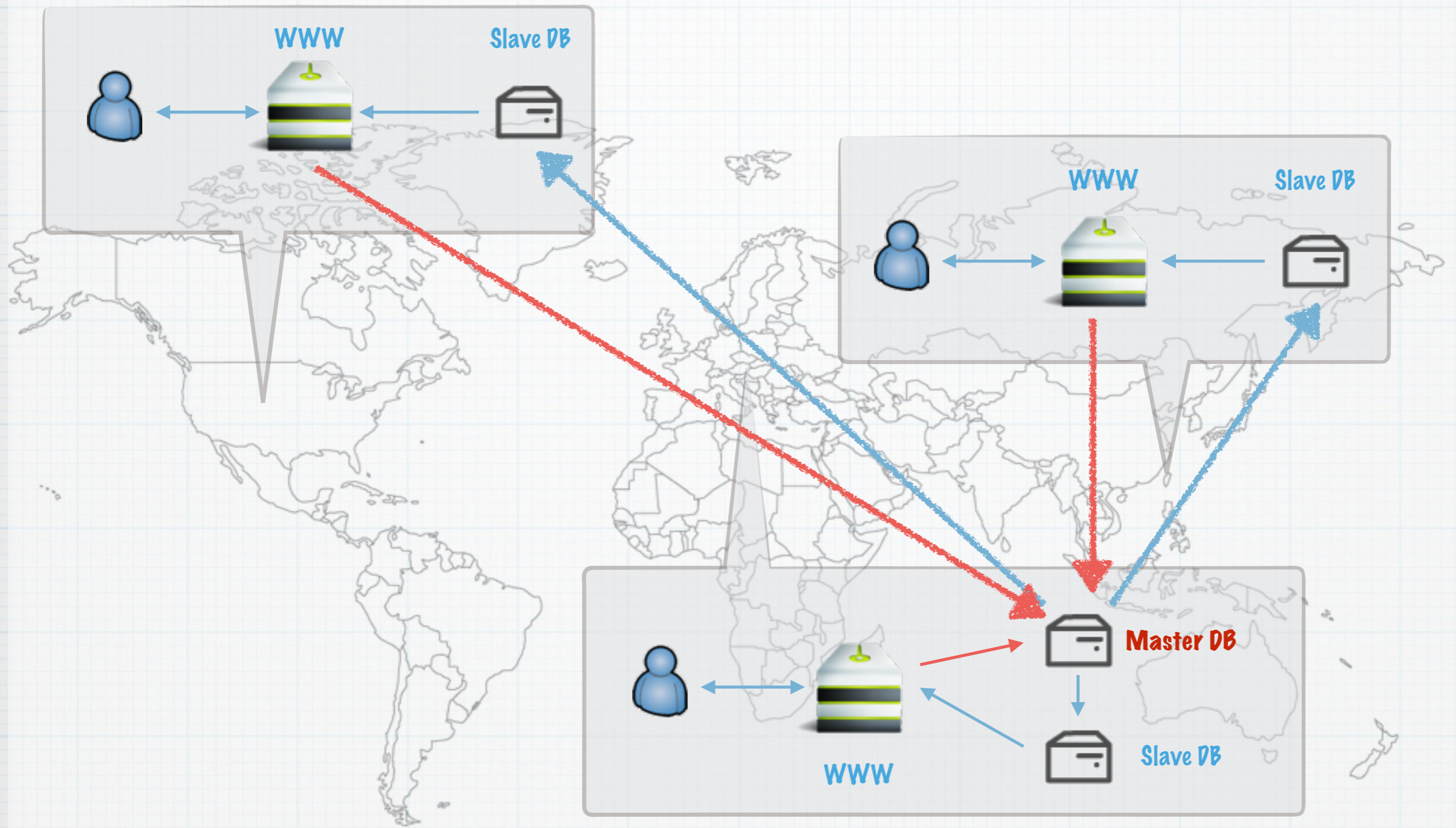
Gyakorlati problémák

- * Felhasználói adatok kezelése nagy késleltetésű hálózaton keresztül
- * Az adatok konzisztens tárolása
- * Gyors adatműveletek
- * Több helyről egyidejű írás problémája (pl. adatütközés)

Egy szerveres megoldás



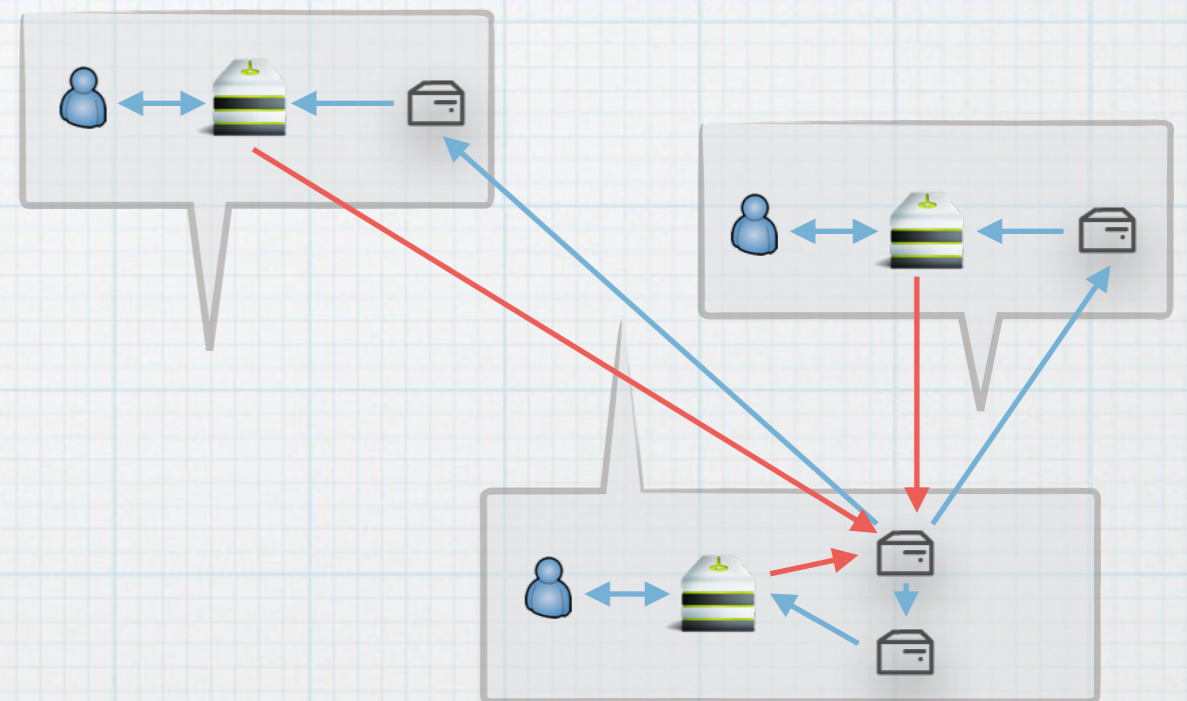
Ha bámi elromlik, akkor minden megáll!

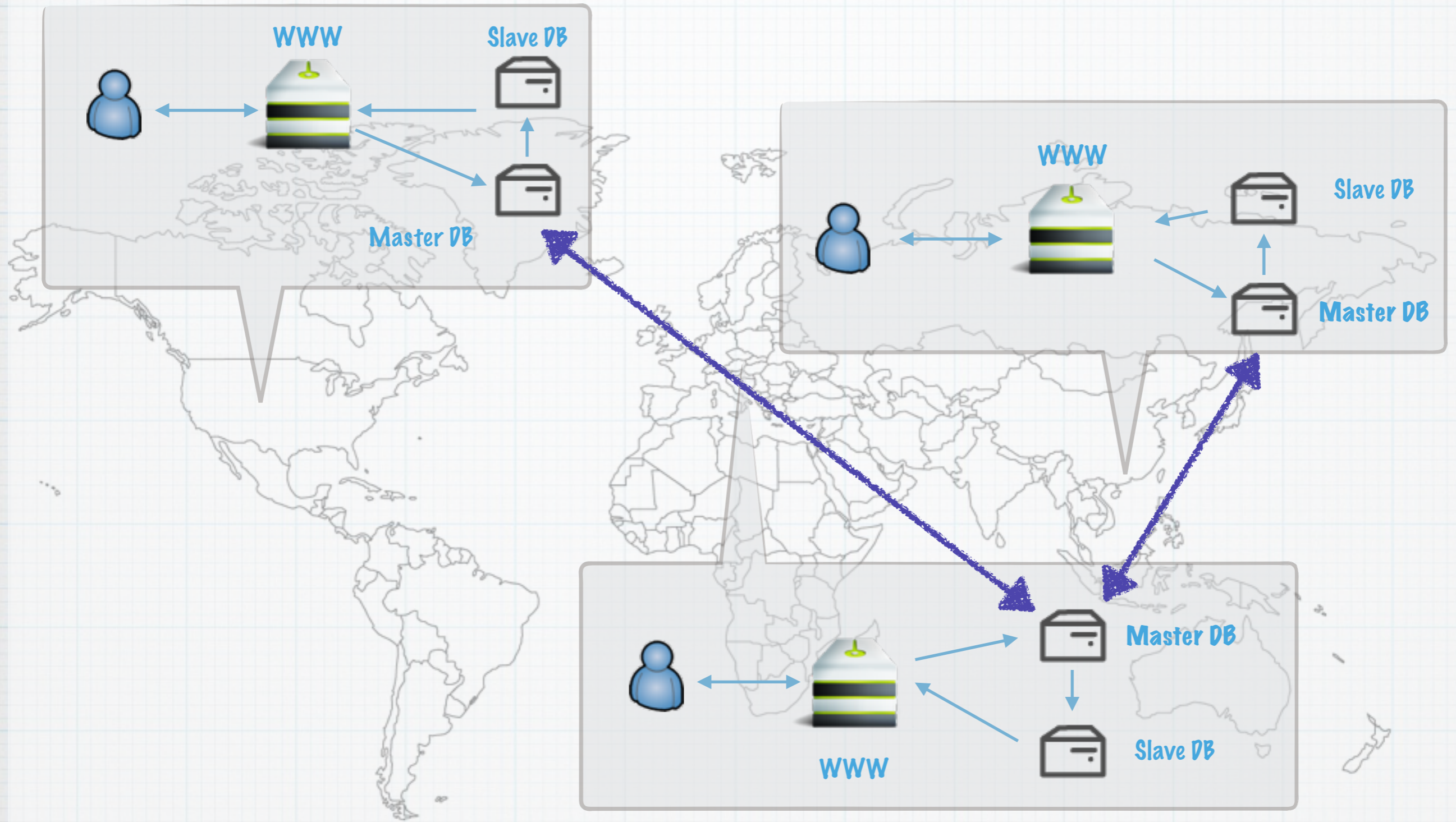


Master-slave replika

Master-slave replika

- * Nagyon nagy a késleltetés, sokszor másodpercek
- * A master túlterhelődhet
- * Ha nem éri el a mestert, akkor megáll az egész?

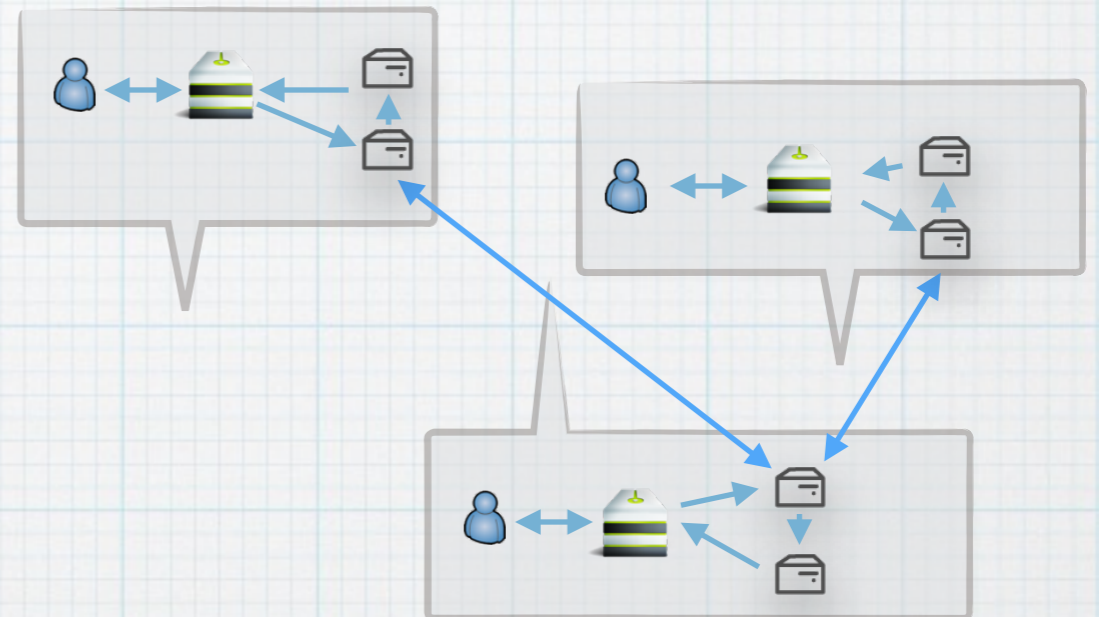


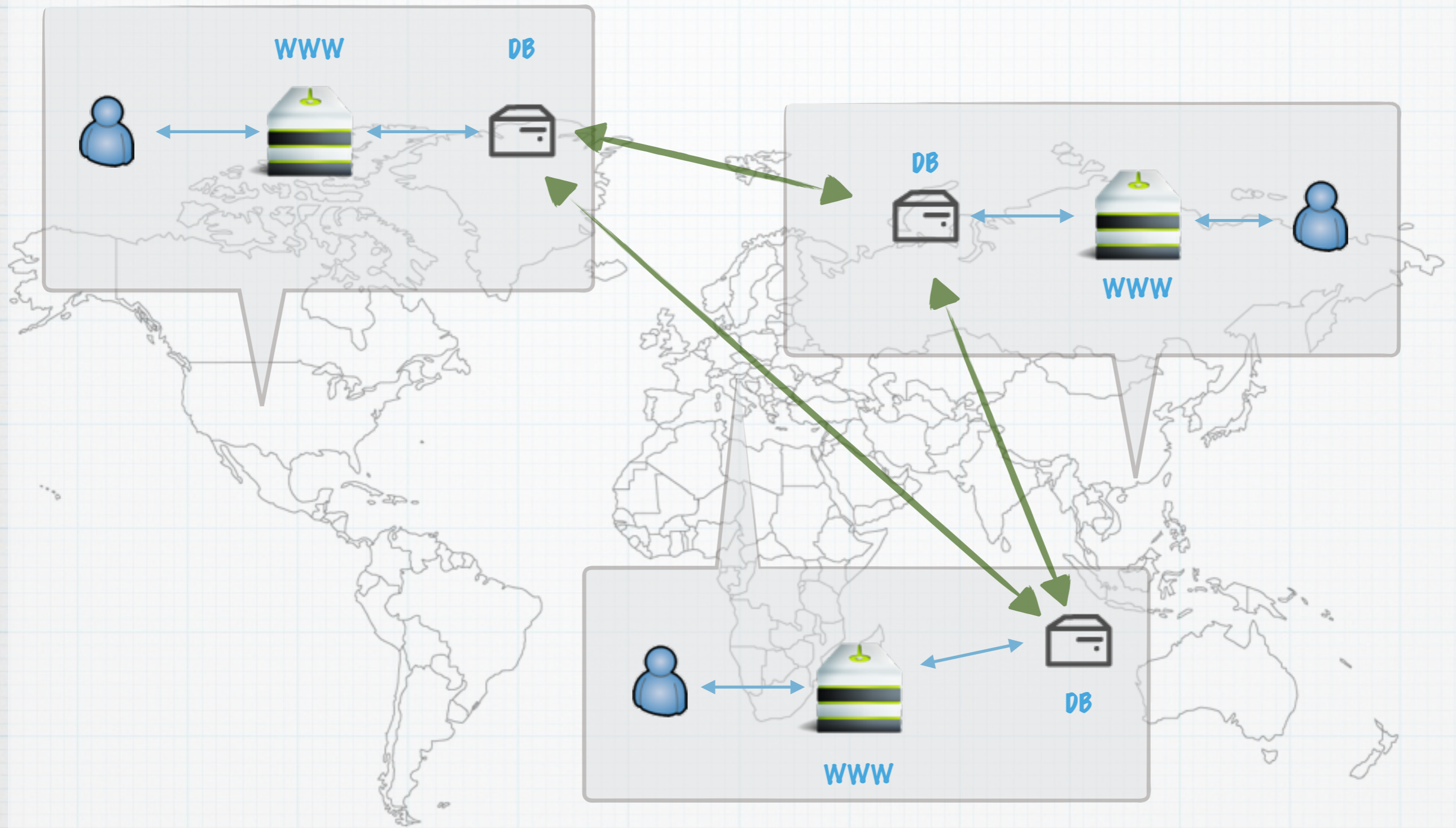


Multi master replika

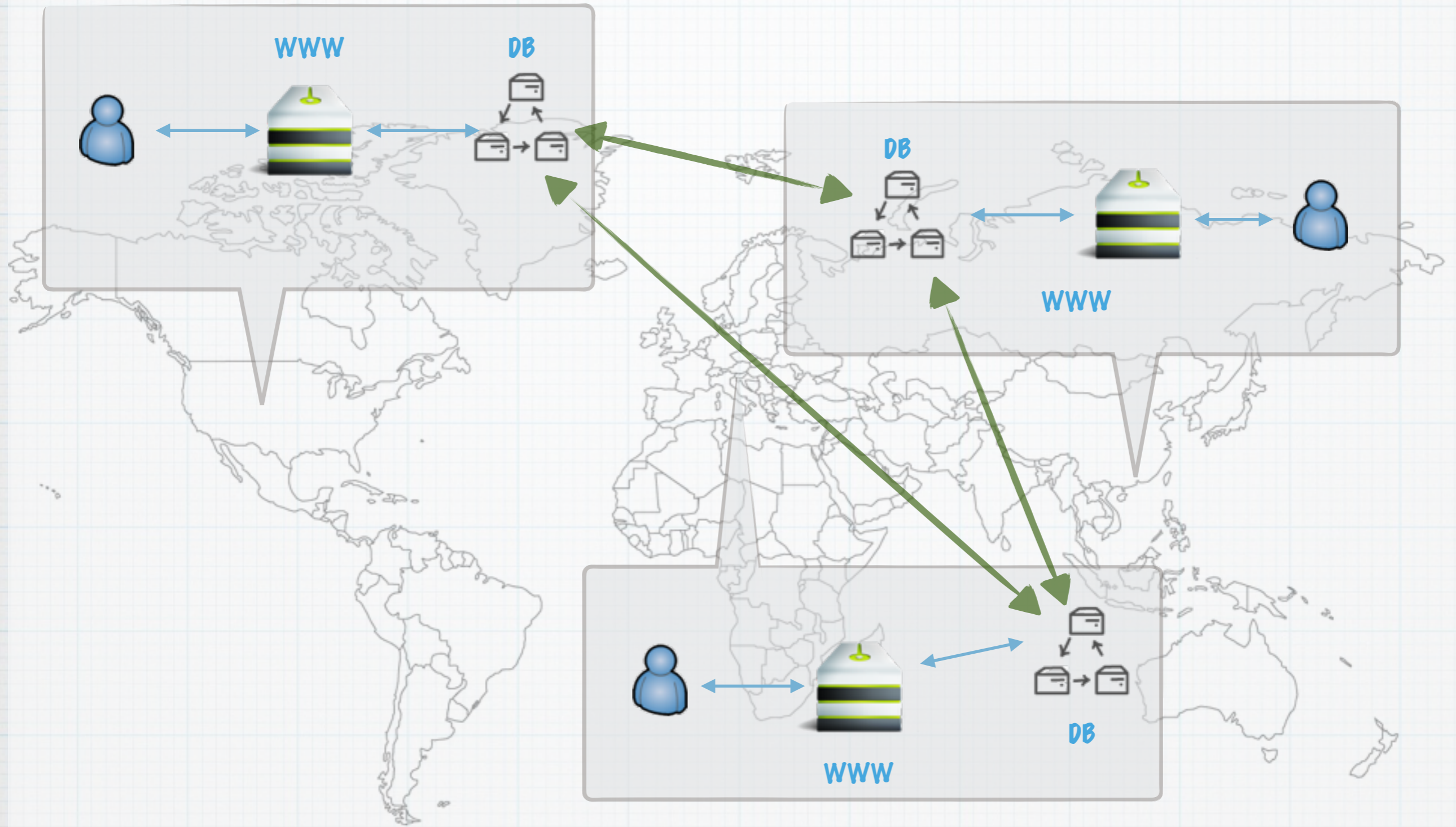
Multi master replika

- * Nincs jó megoldás
- * Nagy késleltetés
- * Kulcs ütközés
- * Felemás megoldások





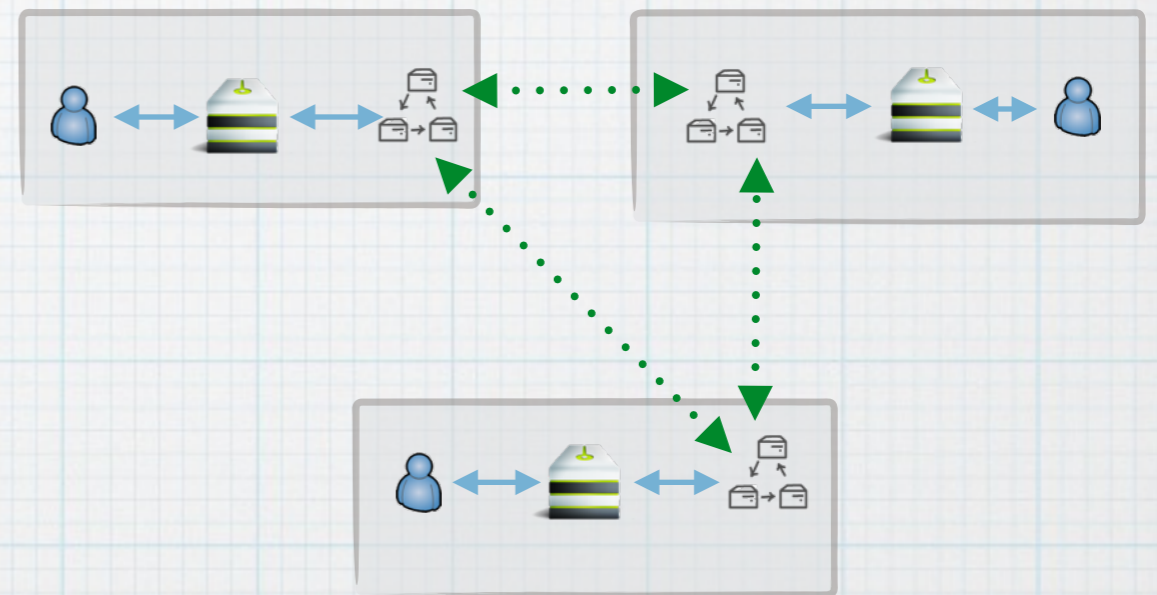
Masterless megoldás (P2P)



Masterless megoldás (teljes redundancia)

Teljes redundancia

- * Datacenteren belüli redundancia
- * DC-k közötti redundancia
- * Token-ring szerű működés
- * A konzisztencia szabályozás

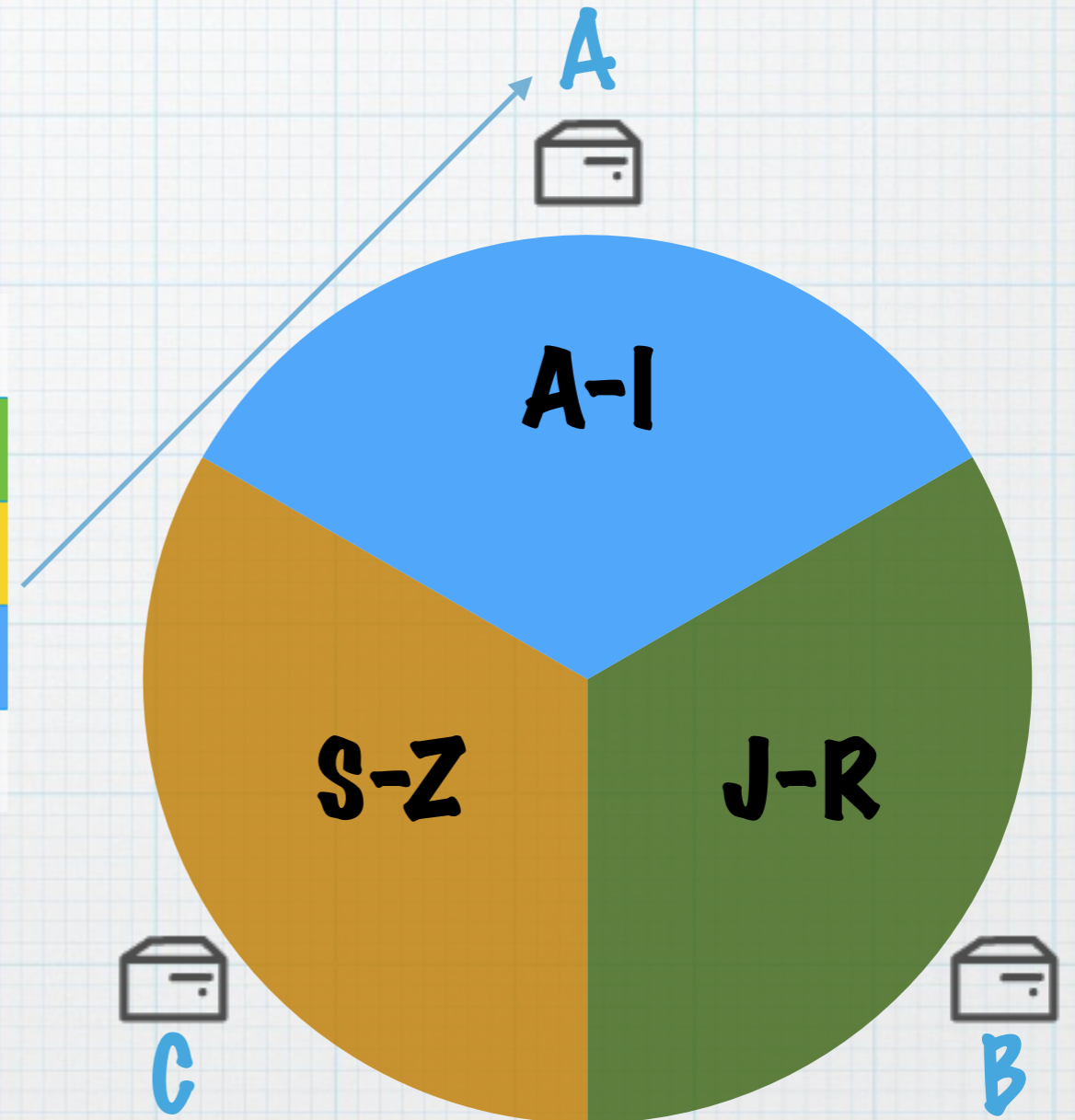


Replikáció, particionálás, konzisztencia

- * Hogyan használjunk és mire jók ezek?
- * Hogyan válasszunk adatbázist?
- * Mire kell figyelni a konfigurálásnál?

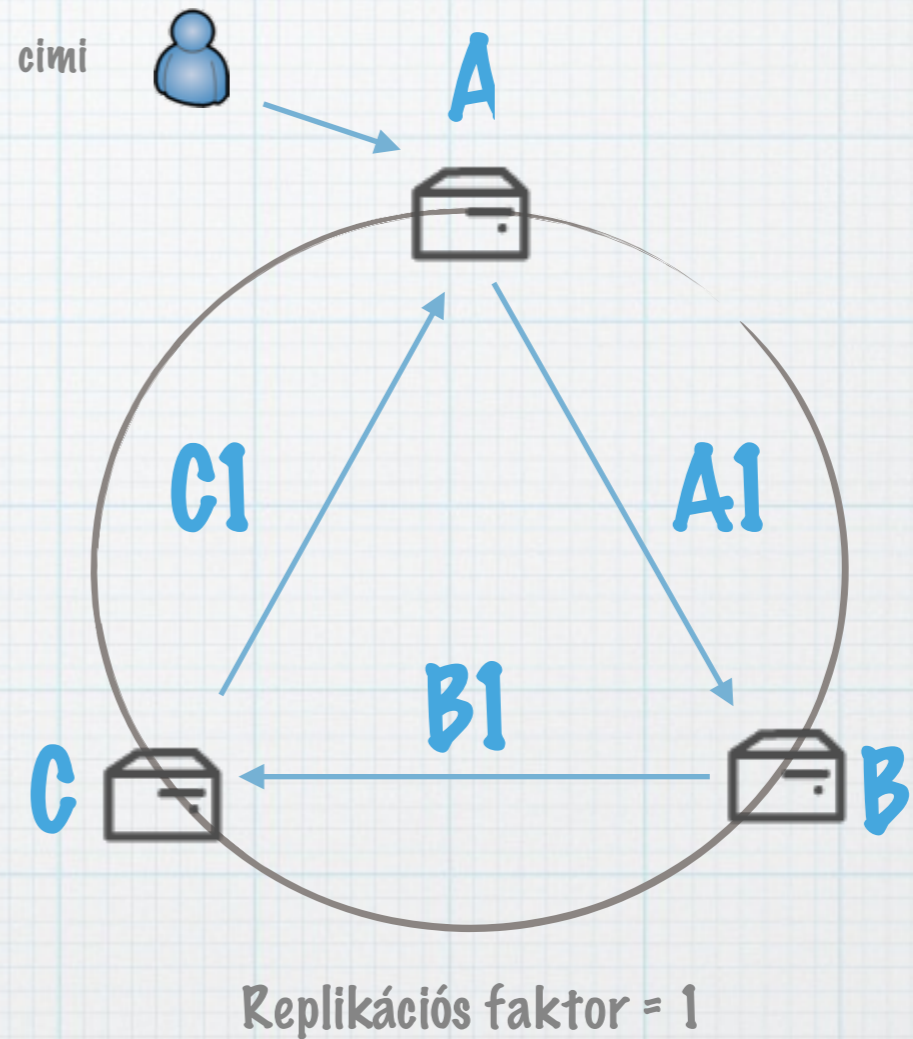
Particionálás

Név	E-mail
Kovács Pál	pal@freemail.hu
Sárkány Anna	anna@gmail.com
Czimmermann	cg@javaforum.hu



Replikáció

- * Adatközponton belül
- * Adatközpontok között



Konzisztencia

- * Az adat tárolási helyességének meghatározása (adat egyezés)
- * $Q = (Rf/2) + 1$ replikáció, ahol Rf a replikáció faktor és minden szám lefelé kerekítve számolandó
Pl. 3 node esetén $Q = (3/2)+1 = 2$

Hogyan válasszunk adatbázist

- * Feladat
- * Adat
- * Művelet
- * Sebesség
- * Redundancia
- * Feladat + adat + művelet + sebesség + redundancia!

Mire kell figyelni a konfigurálásnál?

- * Rendelkezésre álló eszközök, korlátok
- * Üzemeltetési szempontok
- * Automatizmus

Eszközök

- * SQL szerverek: MySQL, PostgreSQL, OracleDB, VoltDB, stb.
- * NoSQL: Riak, Redis, MongoDB, Cassandra, stb.
- * Egyéb: Memcached, Infinispan, Hazelcast, stb.

Cassandra

- * Miért ezt választottuk
- * Hogyan működik
- * Datacenter
- * Replika stratégiák (Simple, Network)
- * Sharding
- * Keyspaces
- * Column families
- * CQL 3 + natív protokoll

Köszönöm figyelmüket!

Czimmermann Gábor

* czimmermann.gabor@icr-system.hu

* <http://www.icr-system.hu>

* @czimmermannng

