



# Nagyvállalati rendszerek

Vállalati informatikai rendszerek  
architektúrájának áttekintése

# Informatika szerepe egy vállalat életében

- ▶ Értéktéremtés
  - Saját fejlesztések
  - Gyártás vezérlés
- ▶ Támogatás
  - Levelező rendszer
  - Címtárak
  - Automatikus riportok

# Kinek vannak igényei?

- ▶ Felhasználók
- ▶ Üzlet
- ▶ Biztonsági szakértők
- ▶ Jogi osztály
- ▶ Üzemeltetők

# Teljesítmény

- ▶ “Legyen gyors”
  - Mikortól számít jónak/rossznak?
- ▶ Service Level Agreement (SLA)
  - 99.999% (“öt kilences”)
- ▶ Mérés és riasztás
  - Automatikus beavatkozás, ha lehetséges

# Robosztusság/hibatűrés

- ▶ Single point of Failure (SPF)
- ▶ Redundancia
  - Áramellátás (UPS)
  - Hálózati eszközök (bonding)
  - Komponensek (cluster)
- ▶ Hibakezelés
- ▶ Mentés és visszaállítás



# Magas rendelkezésre állás

- ▶ Nem csak hiba miatt lehet leállítás
  - Karbantartás (külső, szoftveres, hardveres)
  - Verzió frissítés
- ▶ Megoldás: redundancia
- ▶ SLA itt is használandó
  - 99.999% rendelkezésre állás → 5 perc/év kiesés
- ▶ Példa: RAC, WebLogic Cluster



# Változásra képes

- ▶ Minden projektvezető rémálma: scope bővülés
- ▶ Megoldás: absztrakció
  - “X szolgáltatás-t kell meghívni”
  - “A `https://192.168.60.13:8443/myService` címen található szolgáltatást kell meghívni”
- ▶ Érdemes haladni a korral
  - Technical debt

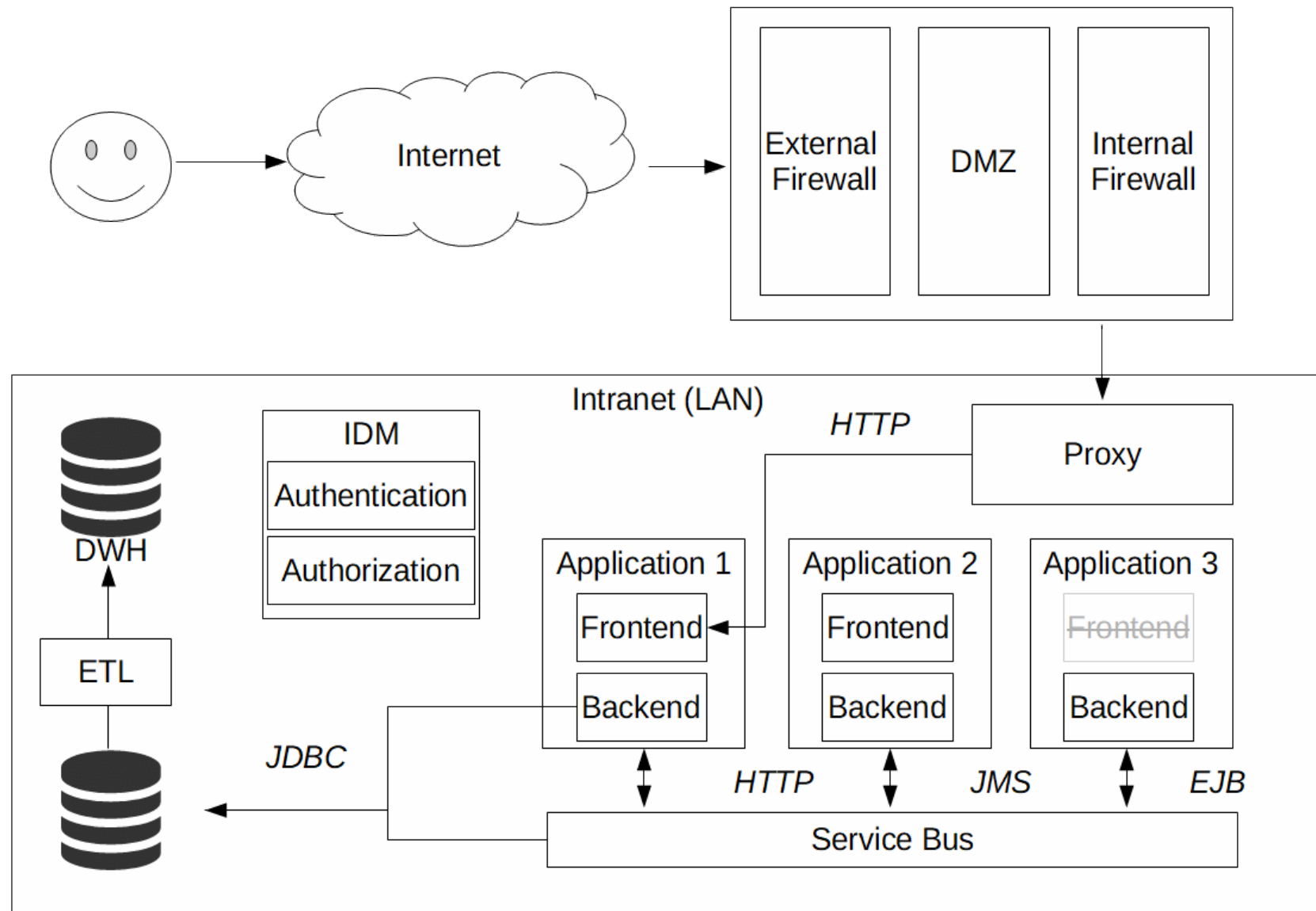
# Biztonság

- ▶ Senki sem férhet hozzá mindenhez
- ▶ A hozzáférések központi rendszer(ek)en keresztül kaphatók
  - Szerepkörök
- ▶ “Közös”, illetve “service account”-ok kerülése
- ▶ Események auditálása
- ▶ Gyakori, ütemezett biztonsági frissítések



# Megismételhetőség

- ▶ Biztos működni fog? Biztos “úgy” fog működni?
- ▶ Változtatások, fejlesztések előzetes tesztelése
- ▶ Több célhoz több környezet szükséges
  - Funkcionális tesztelés
  - Teljesítmény tesztelés
  - Fejlesztés támogatás
- ▶ Automatizálással érdemes támogatni



# Szerverek, virtuális gépek

- ▶ Erőforrások ezeken keresztül biztosíthatók
  - CPU, RAM stb.
- ▶ Jellemzően ez az üzemeltetési egység
- ▶ Főleg virtuális gépek (VM)

# Storage

- ▶ Jellemzően Network Attached Storage (NAS)
- ▶ Redundancia támogatás (RAID)
- ▶ Adatbázisok esetén komoly teljesítmény tényező

# Tűzfal

- ▶ Hálózati komponens
- ▶ ISO/OSI Layer 3/Layer 4 (Network + Transport)
- ▶ Célok:
  - Forgalom szűrés (filtering)
  - Network Address Translation (NAT)
- ▶ Hardveres és szoftveres (iptables) is

# (Reverse) Proxy

- ▶ Hálózati vagy Middleware komponens
- ▶ Jellemzően HTTP szolgáltatásokhoz
- ▶ Apache, nginx, TinyProxy
- ▶ Gyarkan SSL végződési pont
- ▶ Terheléselosztóként (load balancer) is használatos
  - WebLogic Cluster-hez szükséges
- ▶ VirtualHost
  - app1.example.com vs. app2.example.com



# Alkalmazás szerver

- ▶ Jellemzően JavaEE (de lehet .NET stb.)
- ▶ Absztarkciós réteg az alkalmazás számára
  - pl. adatbázis kapcsolatok
- ▶ Clusterezési képesség
- ▶ Több komponens futtatását is lehetővé teszi
  - Jobban menedzselhető
  - Támogatja a modularitást

# Frontend vs Backend

- ▶ Érdemes az alkalmazásokat szétbontani
  - Komponens szinten (külön EAR/WAR/stb.)
  - Infrastruktúra szinten (külön JVM/szerver/stb.)
- ▶ Szabványos protokollok a komponensek között
  - HTTP, SOAP, JMS, EJB ...



# Integrációs réteg (Service Bus)

- ▶ Szolgáltatások hívása egy közös rétegen keresztül
- ▶ Nem csupán proxy
  - Kompozit szolgáltatások
  - Protokol váltás
  - Throttling (ráta korlátozás)
- ▶ Logikai komponens
  - Alkalmazás szervern fut ez is



# Perzisztencia

- ▶ Nem minden szolgáltatás igényli
- ▶ Jellemzően adatbázis alapú (de: PDF generálás)
- ▶ Alkalmazás szerver oldali támogatás
  - Adatbázis kapcsolatok
  - Persistent store-ok

# Identity Management

- ▶ Címtár szolgáltatások
  - Felhasználók (authentication)
  - Csoportok (authorization)
- ▶ Jellemzően LDAP, SAML technológiákkal
- ▶ Jogosultságok automatikus kiosztása
  - Folyamatok alapján (igény, elfogadás, kiosztás)

# Monitoring

- ▶ Két megközelítés
  - Push/ágens alapú architektúra (pl. OEM)
  - Pull architektúra (Nagios)
- ▶ Előre meghatározott metrikák gyűjtése
- ▶ Metrika határok alapján történő riasztás
  - Akár trendek alapján (historikus adatokkal)
- ▶ Nehézség: alkalmazás specifikus metrikák

# Logolás és Audit

- ▶ Érdemes központi log gyűjtést alkalmazni
- ▶ Nem jó ha:
  - Nagyon kevés bejegyzés van
  - Túl sok bejegyzés van
- ▶ Auditálás: komponens függő
  - Oracle adatbázisban beépített
- ▶ Monitoring eszközzel jól integrálható