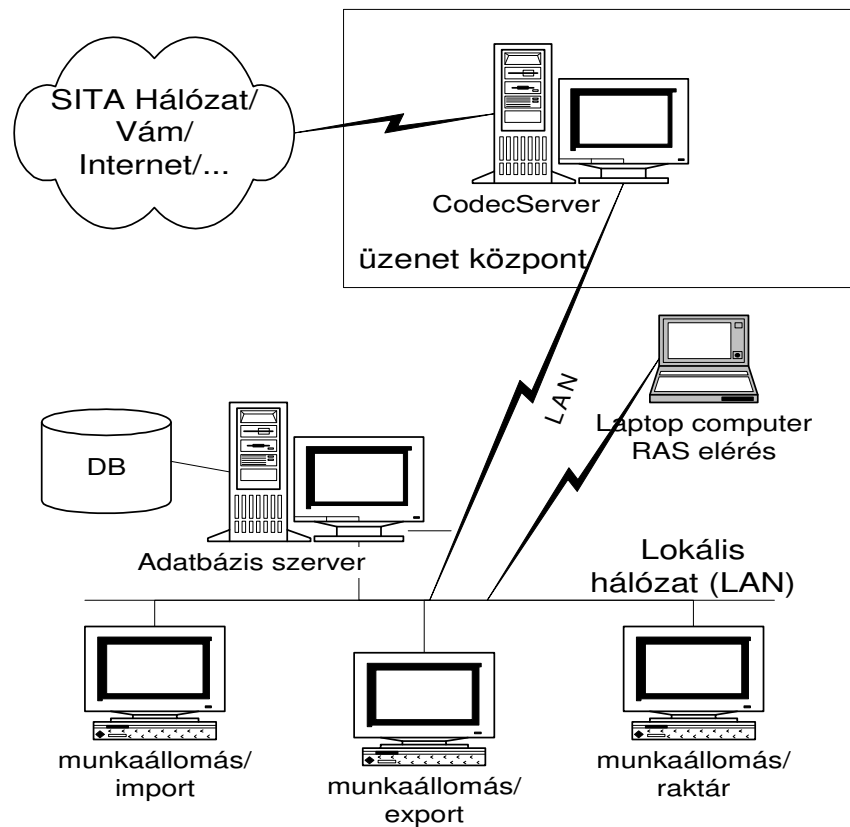


# Handling rendszer technikai specifikáció



CCS Hungary,  
2000 szeptember

# □ Hálózati architektúra



# Fejlesztési vezérelvek

- nemzetközi szabványok támogatása az adatcserében (IATA, Cargolmp, ...)

- TCP/IP hálózati kommunikáció

- SQL relációs adatbázis kezelés

- többretegű alkalmazás megközelítés, moduláris, paraméterezhető rendszer

- iteratív és inkrementális fejlesztés és bevezetés, követelmény követés

- objektum orientált programtervezés (Unified Modelling Language)

## - SQL relációs adatbázis kezelés javasolt lehetőségek

### I. (általunk javasolt): Microsoft SQL Server 7

**Előnyök:** SQL92 szabvány+kiterjesztések, gazdag paraméterezési lehetőségek, vizuális adminisztráció, ütemezhető feladatok (pl. archiválás), replikáció támogatás, rendszergazdák automatikus riasztása hiba esetén, beépített Internet támogatás, stb., viszonylag kedvező ár/teljesítmény arány

**Hátrányok:** nem ingyenes, beruházást igényel

### II. : Interbase 6 SQL Server

**Előnyök:** nyílt forráskódú, ingyenes, SQL92 szabvány, viszonylag egyszerű kezelhetőség

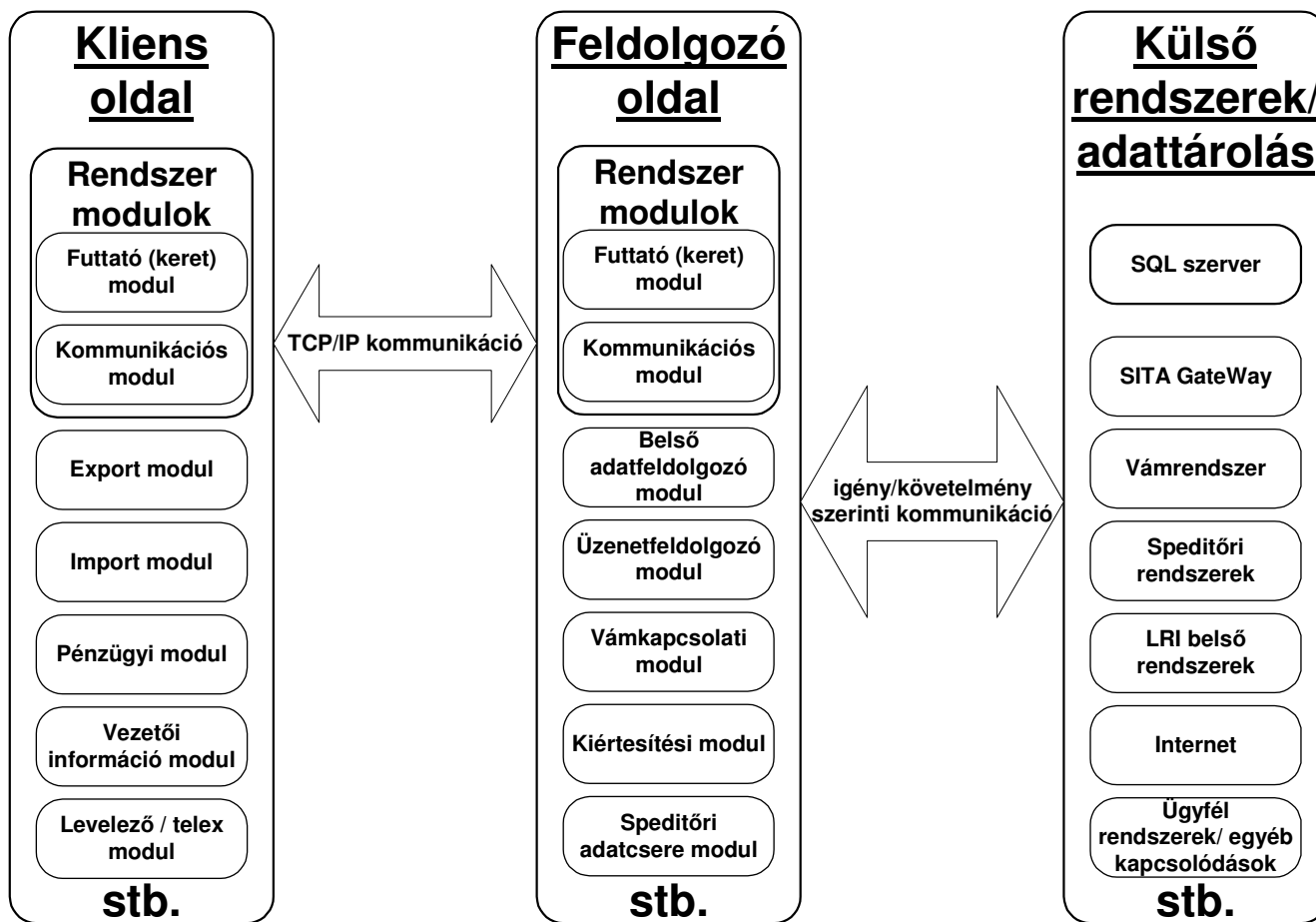
**Hátrányok:** kevésbé paraméterezhető, viszonylag kevés eszköz a rendszeradminisztráció támogatására

### III. : A belső informatikai politika által meghatározott adatbázis

(pl. ORACLE, INFORMIX,...)

**Előnyök:** feltehetően van hozzáértő belső személyzet a rendszertámogatásra

**- többretegű alkalmazás megközelítés,  
moduláris, paraméterezhető rendszer**



# □ Adatbiztonság

## Követelmények:

- ◆ felhasználói jogosultságok kezelése
- ◆ védekezés adatvesztés ellen
- ◆ illetéktelen hozzáférések megakadályozása
- ◆ hibás adatok kiszűrése
- ◆ a rendszer folyamatos működésének biztosítása

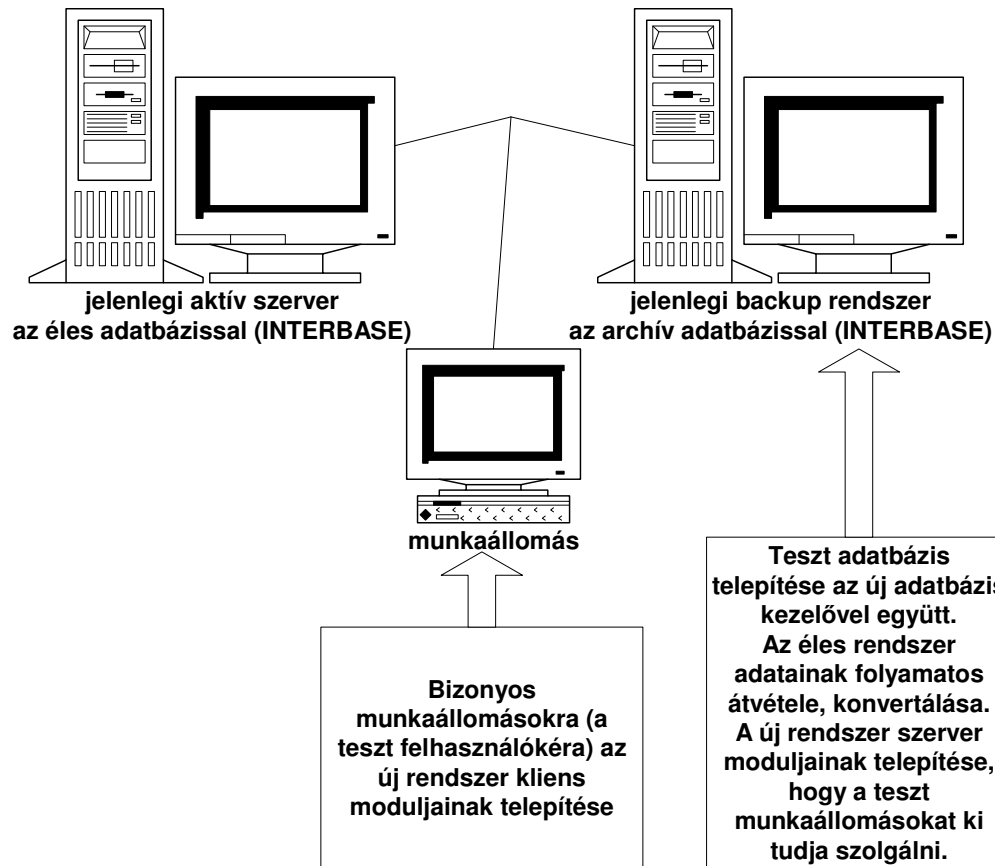
## Megoldások:

- ❖ kétszintű jogosultság kezelés, az applikációs szerver futtatórendszere a felhasználónak csak az engedélyezett modulok, menüpontok, műveletek elérését teszi lehetővé + az SQL szerveren csak a szükséges objektumokat érhetik el az egyes szoftvermodulok (applikációs szerepkör); a felhasználók csoportokba szervezhetőek, ezzel könnyebbé válik a jogosultságok karbantartása
- ❖ a rendszer minden egyes tranzakciót naplóz, az egyszer létrehozott rekordot a felhasználók csak logikailag tudják törölni, fizikailag nem (ehhez csak az SQL szerverhez megfelelő jogosultsággal rendelkező adatbázis adminisztrátornak van lehetősége); az MS SQL Server 7 alkalmazása esetén beépített és ütemezhető adatmentési technológiák; saját adatarchiválási modul a rendszerhez
- ❖ az adatbázist csak a jogosult adminisztrátor konfigurálhatja, érheti el közvetlenül; a nejelentkezések naplózása; lehetőség titkosított hálózati kommunikáció (akár 256 bites kulcs) használatára (pl. RAS elérés esetén)
- ❖ adatfelvitelkor logikai ellenőrzés + az SQL adatbázisba beépített másodlagos ellenőrzés
- ❖ az adatbázis replikációja másodlagos szerverre + feldolgozó modulok előre telepítése

# □ Tesztrendszer megvalósítása és az új rendszer fokozatos bevezetése

## A vázolt megoldás előnyei:

- a jelenlegi rendszer működése zavartalanul folyhat
- a tesztrendszerben a felhasználók valós adatokat láthatnak, és az új kezelő felületekkel is megismerkedhetnek ezzel párhuzamosan
- könnyebben kontrolálhatóak az új rendszerrel szemben támasztott követelmények (pl. a sebessége), hiszen a tényleges adatokkal, adatforgalommal dolgozik
- a végső átállítás előtt már folyamatosan tesztelve van az adatbázisok kompatibilitása
- nem igényel extra hardver beruházást



## - iteratív és inkrementális fejlesztés és bevezetés, követelmény követés

### □ a folyamat: Objectory (Rational Unified Process)

Az Objectory folyamatot jelenlegi formájában a Rational fejlesztette ki korábban kidolgozott objektumorientált módszereinek tapasztalatai alapján. Az Objectory iteratív és inkrementális folyamat, azaz a probléma megoldását fokozatos finomítások sorozatán keresztül éri el.

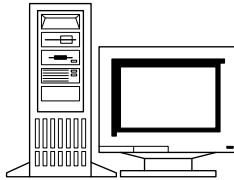
### □ a megközelítés fő előnyei:

- az Objectory ellenőrzött folyamat, melynek eredményeként a csapat minden tagja a fejlesztési folyamat minden pillanatában tisztában lehet a kitűzött célokkal, és az adott pillanatig elért eredményekkel
- lehetővé teszi a folyamatos minőségellenőrzést
- projektben felbukkanó legkomolyabb kockázati elemek felmérése és kiküszöbölése
- architektúra ellenőrzésére demo rendszert implementálunk, mely igazolja az architekturális döntések helyességét, és képes a leglényegesebb használati esetek végrehajtására, a rendszer leendő használói megismerkednek a kezelői felületekkel és a folyamatokkal, visszajelzéseik alapján finomíthatjuk a rendszert
- az építkezés során iteratív és inkrementális módon kifejlesztjük a teljes rendszert, amely kielégíti a felhasználók által támasztott követelményeket és változtatási igényeket (ezek lehetnek jogszabályi változtatások is pl. VÁM, APEH rendelet, stb.)
- miután a termék a felhasználó kezébe került, gyakran előfordulnak olyan esetek, melyek további fejlesztéseket követelnek a rendszer hangolására, ezért az átadás előre tervezetten ún. béta-verzióval kezdődik



# □ Folyamatos működés támogatása

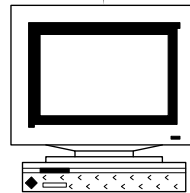
**Aktív (online) szerverek az üzenetek és tranzakciók feldolgozására (fizikailag ezek egy vagy több gépen is lehetnek)**



- SQL szerver (adatbázis tárolás)
- CCS feldolgozó server modulok

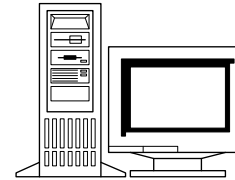
LAN kapcsolat, az adatbázis adatok folyamatos áttöltése, illetve a feldolgozó servermodulok konfigurációs változásainak áttöltése

LAN



**kliens gép**  
folyamatosan tud működni

**Tartalék (backup) szerverek az üzenetek és tranzakciók másolatának tárolására (fizikailag ezek egy vagy több gépen is lehetnek)**



- SQL szerver (adatbázis tárolás), folyamatosan futó alkalmazás, az adatok szinkronizálása miatt, hiba esetén ez lesz az aktív adatbázis
- CCS feldolgozó server modulok, előre telepítve vannak, de nem futnak folyamatosan, hiba esetén aktivizálódnak + riasztás a rendszer-gazdák felé