

Integrácia aplikácií

Pavol Mederly

Centrum informačných technológií UK
Fakulta informatiky a informačných technológií STU

mederly@rec.uniba.sk

Obsah

1. Úvod
2. Prístupy k integrácii aplikácií
3. Príklad: integrácia na UK

Úvod

- v typickej organizácii je množstvo aplikácií
- je potrebné, aby spolupracovali
 - napr. keď treba automatizovať **biznis procesy**, ktoré sú podporované viacerými aplikáciami
 - alebo keď chceme získať **ucelený pohľad na údaje** spravované viacerými aplikáciami (napr. údaje o zákazníkovi)

Príklad

Obchodná spoločnosť

Proces: **spracovanie objednávky**

- prijatie objednávky (zákaznícky portál)
- overenie stavu zákazníka (CRM)
- overenie kreditu zákazníka (ERP)
- zistenie stavu skladu (skladový systém)
- príprava tovaru (ľudská úloha)
- doručenie (externý partner)
- fakturácia (ERP)
- zobrazenie stavu procesu (zákaznícky portál, mail)

CRM = Customer Relationship Management System
ERP = Enterprise Resource Planning System

Otázka 1

Prečo je tam toľko systémov?

- je ťažké (nemožné) vytvoriť systém, ktorý by pokryl všetky potreby organizácie
 - relatívne najbližšie k tomuto cieľu sú systémy typu ERP (napr. SAP R/3), ale ani tie nepokrývajú všetko
- „best of breed“ má svoje výhody
- často chýba koordinácia medzi oddeleniami

Otázka 2

Prečo tie systémy nespolupracujú?

„Od chvíle, keď bol spustený druhý počítačový program, používatelia nevedia pochopiť,
prečo sa automaticky plne neprepojil s prvým.“

-- P. Cadarette, K. M. Durward

- **lebo sú vyvíjané nezávisle na sebe**
- technologické otázky
- konceptuálne otázky: typicky rôzne dátové modely
 - od triviálnych rozdielov ako „Meno“ – „KrstneMeno“
 - cez problémy typu „kto udržiava databázu klientov?“
 - až po ťažko riešiteľné problémy so sémantikou, napr.:
čo označuje údaj „Adresa“?

Prečo ... ? (2)

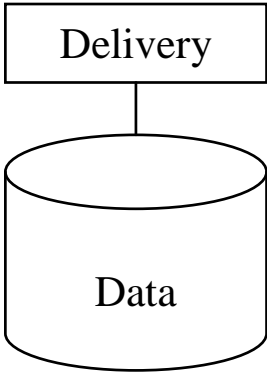
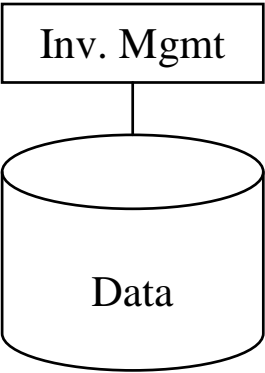
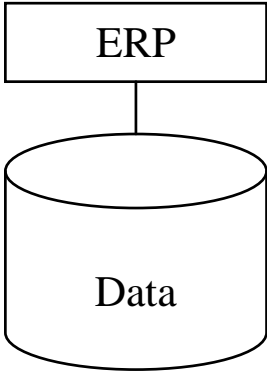
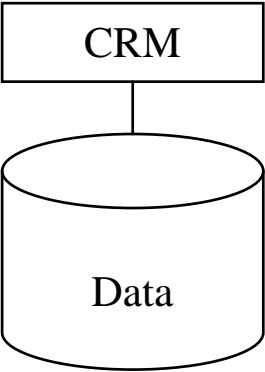
- pri COTS sú rozdiely pochopiteľné
- pri vývoji na zákazku už trochu menej

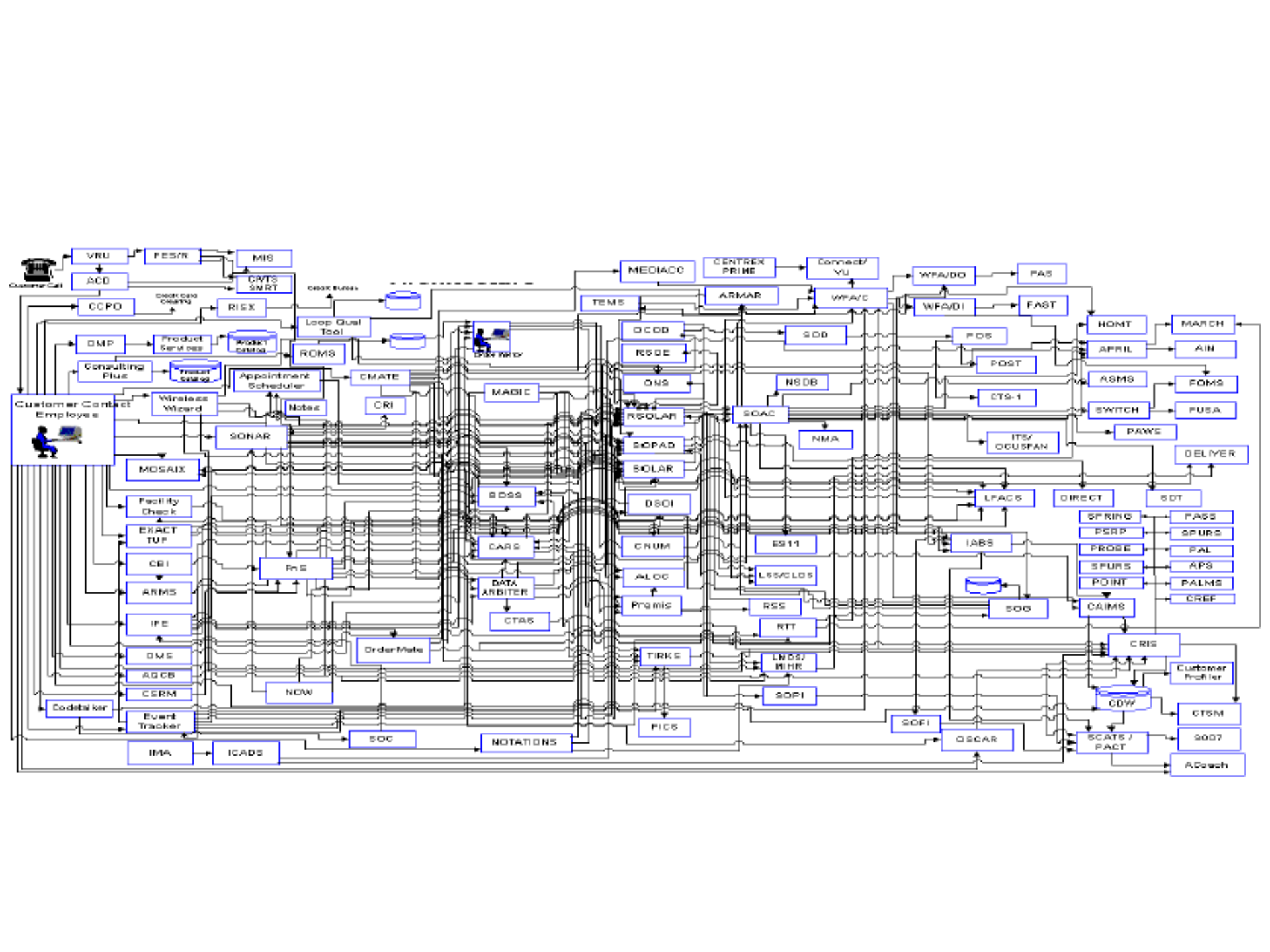
- konkrétne:
 - času/zdrojov je málo → sústreďujeme sa na dosiahnutie primárneho cieľa (požadovaná funkčnosť)
 - Conwayov zákon: „Organizácie produkujú systémy, ktorých dizajn kopíruje komunikačnú štruktúru týchto organizácií.“
 - t.j. ak oddelenia vo firme nekomunikujú (nespolupracujú), nebudú spolupracovať ani systémy, ktoré vytvorili / nasadili
 - spájanie firiem, akvizície

Otázka 3

Čo s tým?

Integrácia aplikácií





Vieme to spraviť lepšie?

Integrácia – čo to teda je?

- Integrácia aplikácií je proces **spájania aplikácií** tak, aby boli schopné **spolupracovať** na poskytovaní **definovanej funkčnosti**.
- To, čo za týmto účelom vytvárame, nazývame **integračným riešením**.
- Sú to
 - úpravy aplikácií (ak sú vôbec možné a vhodné!)
 - komponenty slúžiace na komunikáciu aplikácií / komunikáciu s aplikáciami

Študent

Rozvrh

Preukazy

Prístupový systém

eLearning

Dochádzka

Kurzy

Stravovanie

Ubytovanie

Integračné riešenie

Univ.
terminály

FIS

EMcard

PaM

CKM

Granty

Helpdesk

Knižnica

VoIP

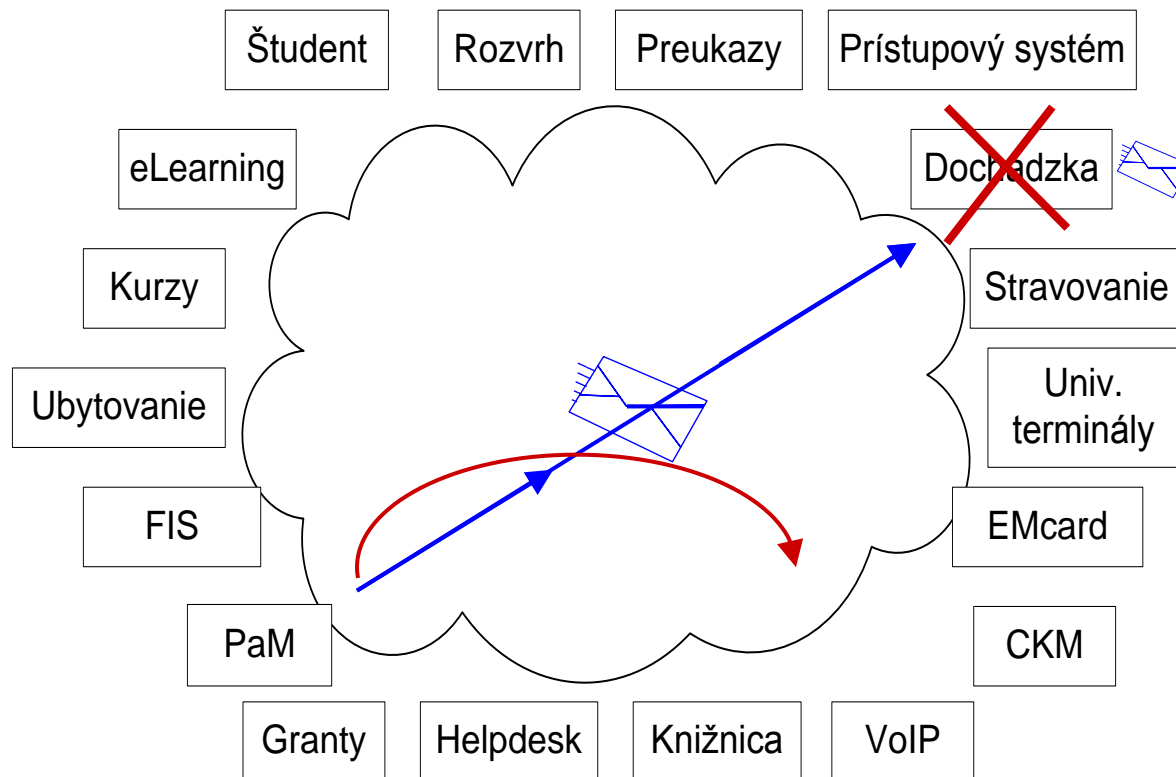
Čo je najdôležitejšie na integračnom riešení?

VOL'NÁ VÄZBA **(loose coupling)**

- v čase **behu** (run-time)
- v čase **vývoja** (design-time)

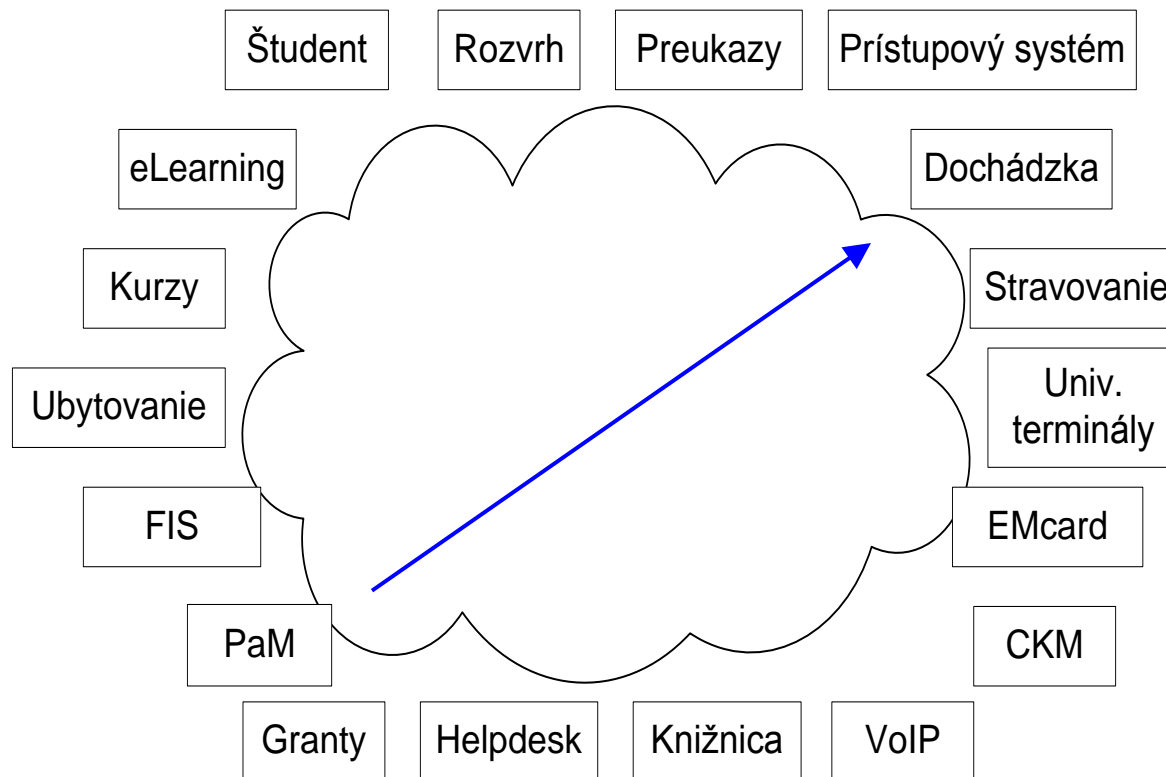
Voľná väzba v čase behu

- Integračné riešenie nevyžaduje, aby prepojené systémy boli funkčné **v tom istom čase**



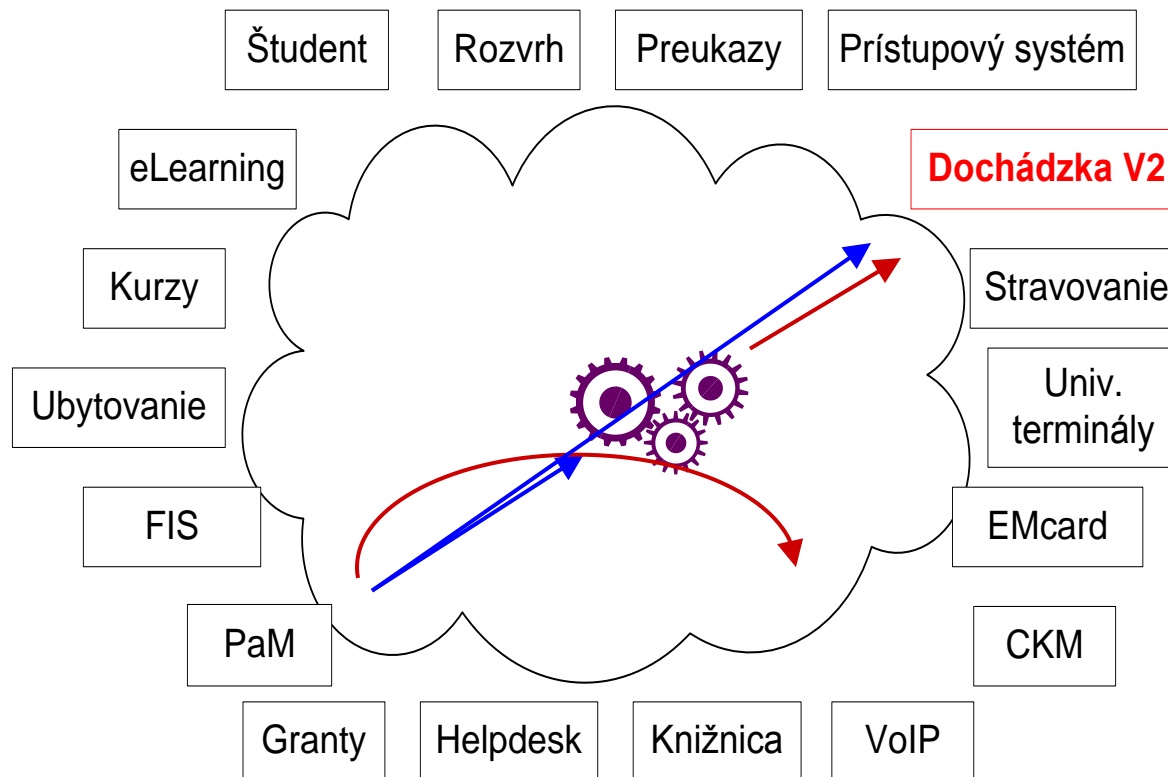
Voľná väzba v čase návrhu

- Integrované riešenie umožňuje, aby sa prepojené systémy mohli **vyvíjať** nezávisle od seba



Voľná väzba v čase návrhu

- Integračné riešenie umožňuje, aby sa prepojené systémy mohli **vyvíjať** nezávisle od seba



Tri dimenzie klasifikácie

- Poskytovaná funkčnosť
- Spôsob komunikácie s aplikáciami
- Vnútoraná komunikácia v rámci int. riešenia

Klasifikácia 1

Typ poskytovanej funkčnosti

- Integrácia prostredníctvom portálu (Portal Integration)
- Agregácia dátových entít (Entity Aggregation)
- Integrácia procesov (Process Integration)

Integrácia prostredníctvom portálu

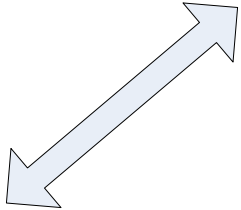
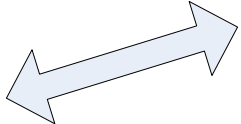
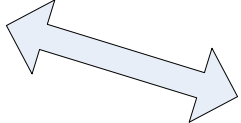
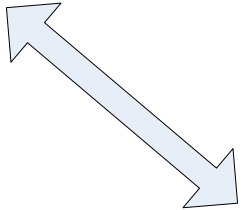
- poskytuje **používateľom** vizuálne konzistentný pohľad na viacero systémov naraz
- **príklad:**
 - v jednej časti obrazovky vidím objednávku
 - v druhej údaje o zákazníkovi (z CRM)
 - v tretej údaje o jeho finančných transakciách (ERP)
 - vo štvrtej údaje o skladových zásobách (inventory)
- v niektorých prípadoch viem údaje aj aktualizovať
- integračné riešenie „samo od seba“ v aplikáciách nevykonáva žiadne aktivity!

Databáza
objednávek

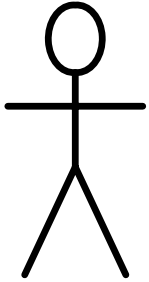
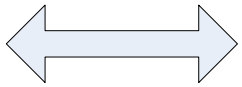
ERP

CRM

Sklad

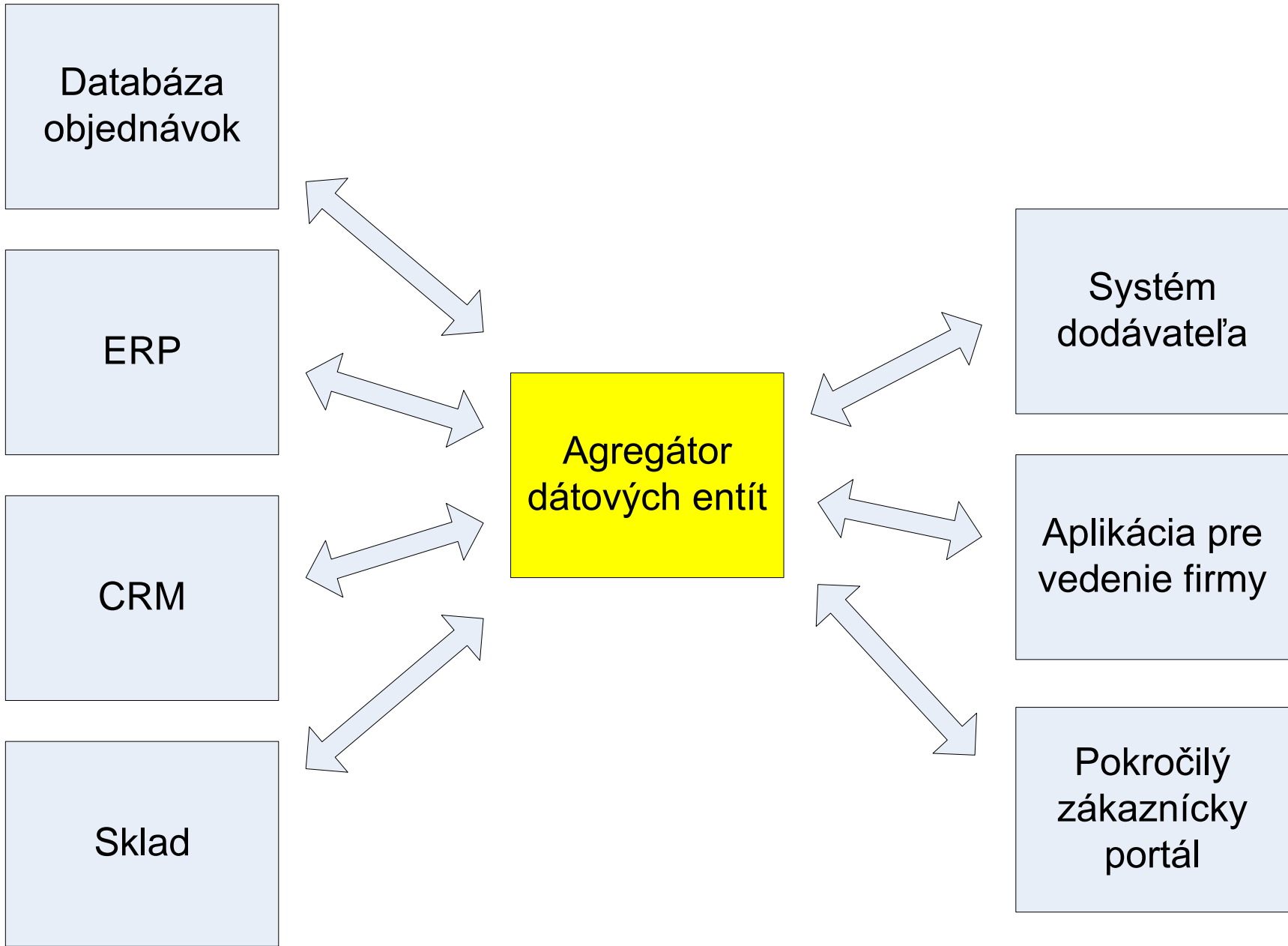


Portál



Agregácia dátových entít

- poskytuje (ďalším, novým) aplikáciám konsolidovaný pohľad na údaje spracovávané jednotlivými systémami
- príklad: virtuálna entita „Objednávka“
 - atribúty týkajúce sa objednávky ako takej
 - atribúty týkajúce sa zákazníka
 - atribúty týkajúce sa produktov a skladu
 - (čo ak mám napr. viac systémov s údajmi o zákazníkovi?)
- int. riešenie „samo“ nevykonáva žiadne aktivity
- tiež „Enterprise Information Integration“ (EII)



Databáza
objednávok

ERP

CRM

Sklad

Agregátor
dátových entít

Systém
dodávateľa

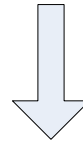
Aplikácia pre
vedenie firmy

Pokročilý
zákaznícky
portál

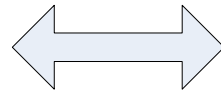
Integrácia procesov

- vykonáva interakcie medzi systémami s cieľom realizácie určených požiadaviek
 - typicky ide o realizáciu definovaných biznis procesov
- príklad:
 - už spomenutá automatizácia spracovania objednávky
- toto je význam snád' najbližší pojmu „Enterprise Application Integration“ (EAI)

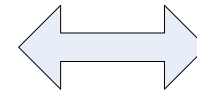
Databáza
objednávok



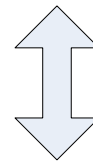
CRM



Manažér
procesov



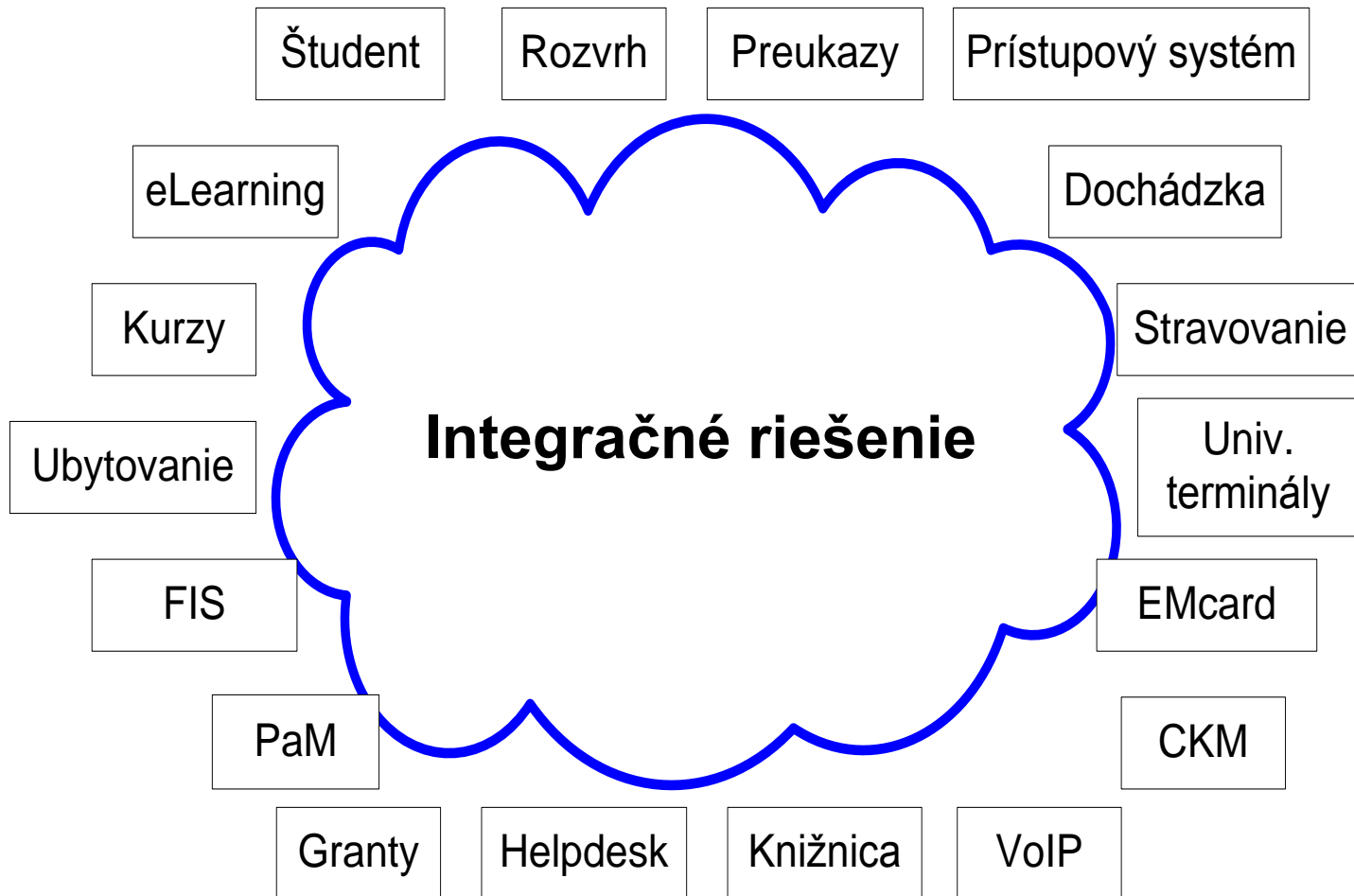
Sklad



ERP

Klasifikácia 2

Ako komunikovať s aplikáciami?



Klasifikácia 2

Ako komunikovať s aplikáciami?

- Ideálne je pracovať na úrovni **aplikačnej logiky**
 - napr. „vráť zoznam zamestnancov“
 - sem môžeme zaradiť aj „import/export“
- Výhody:
 - **jednoduchosť**: pracujeme s pojmami AL
 - **zapuzdrenie**: nie sme závislí napr. na štruktúre databázy (dôležité pre vývoj!)
 - **robustnosť**: aplikuje sa validácia vstupu, kontrola prístupových práv, ...
 - **známy programátorský model** t.j. volanie funkcie

Používateľské
rozhranie

**Aplikačná
logika**

Perzistentné
údaje
(databáza)

Klasifikácia 2

Ako komunikovať s aplikáciami?

- Nevýhody / možné problémy:
 - nie každá aplikácia poskytuje API
 - nie vždy je API dostatočné a správne fungujúce
 - ak voláme API priamo z inej aplikácie, vzniká tesná väzba
 - API spravidla nie sú stavané na hromadný prenos údajov
 - API sú niekedy (zbytočne?) zložité – napr. dátové štruktúry alebo interakčné protokoly:
 - Call BAPI_EMPLOYEE_ENQUEUE
 - Call BAPI_EMPLCOMM_CREATE
 - Call BAPI_EMPLOYEE_ENQUEUE

Používateľské
rozhranie

**Aplikačná
logika**

Perzistentné
údaje
(databáza)

Klasifikácia 2

Ako komunikovať s aplikáciami?

- Niekedy treba pristupovať priamo k databáze
 - napr. čítaj tabuľky LFA1, LFBK, KNA1, KNBK
- Výhody:
 - prístup k všetkým údajom aplikácie
 - efektívnosť: vieme preniesť veľa dát naraz
 - dostupnosť nástrojov: drivery, ETL, ...

Používateľské
rozhranie

Aplikačná
logika

**Perzistentné
údaje
(databáza)**

Klasifikácia 2

Ako komunikovať s aplikáciami?

- Nevýhody / možné problémy:
 - nepublikované a zložité DB schémy (problém s pochopením a s prechodom na vyššiu verziu aplikácie!)
 - práca na nízkej úrovni abstrakcie
 - chýba validácia vstupov a kontrola oprávnení (možný spor s pravidlami pre prístup k údajom)
 - možná kolízia s činnosťou aplikácie (kešovanie, zamykanie, ...)
 - niekedy nie je možné (ASP/SaaS)
 - ak pristupujeme priamo z inej aplikácie, vzniká tesná väzba

Používateľské
rozhranie

Aplikačná
logika

**Perzistentné
údaje
(databáza)**

Klasifikácia 2

Ako komunikovať s aplikáciami?

- Poslednou možnosťou je pristupovať k používateľskému rozhraniu aplikácie („screen scraping“)
 - simuláciou terminálu / stláčaní klávesov a klikaní myšou / cez HTTP session / ...
- Výhody:
 - nízke riziko poškodenia údajov (zapojená je validácia i bezpečnosť)
 - použiteľné pre väčšinu aplikácií

Používateľské
rozhranie

Aplikačná
logika

Perzistentné
údaje
(databáza)

Klasifikácia 2

Ako komunikovať s aplikáciami?

- Nevýhody / možné problémy:
 - extrémna krehkosť: aj malá zmena UI (alebo neočakávaná situácia) spôsobí problémy s fungovaním integračného rozhrania
 - komplikovaný prístup k údajom – napr. nutnosť transformácie z prezentačnej formy na „čisté údaje“
 - obmedzený prístup k údajom: dostaneme len to, čo vidí používateľ
 - nízka efektívnosť

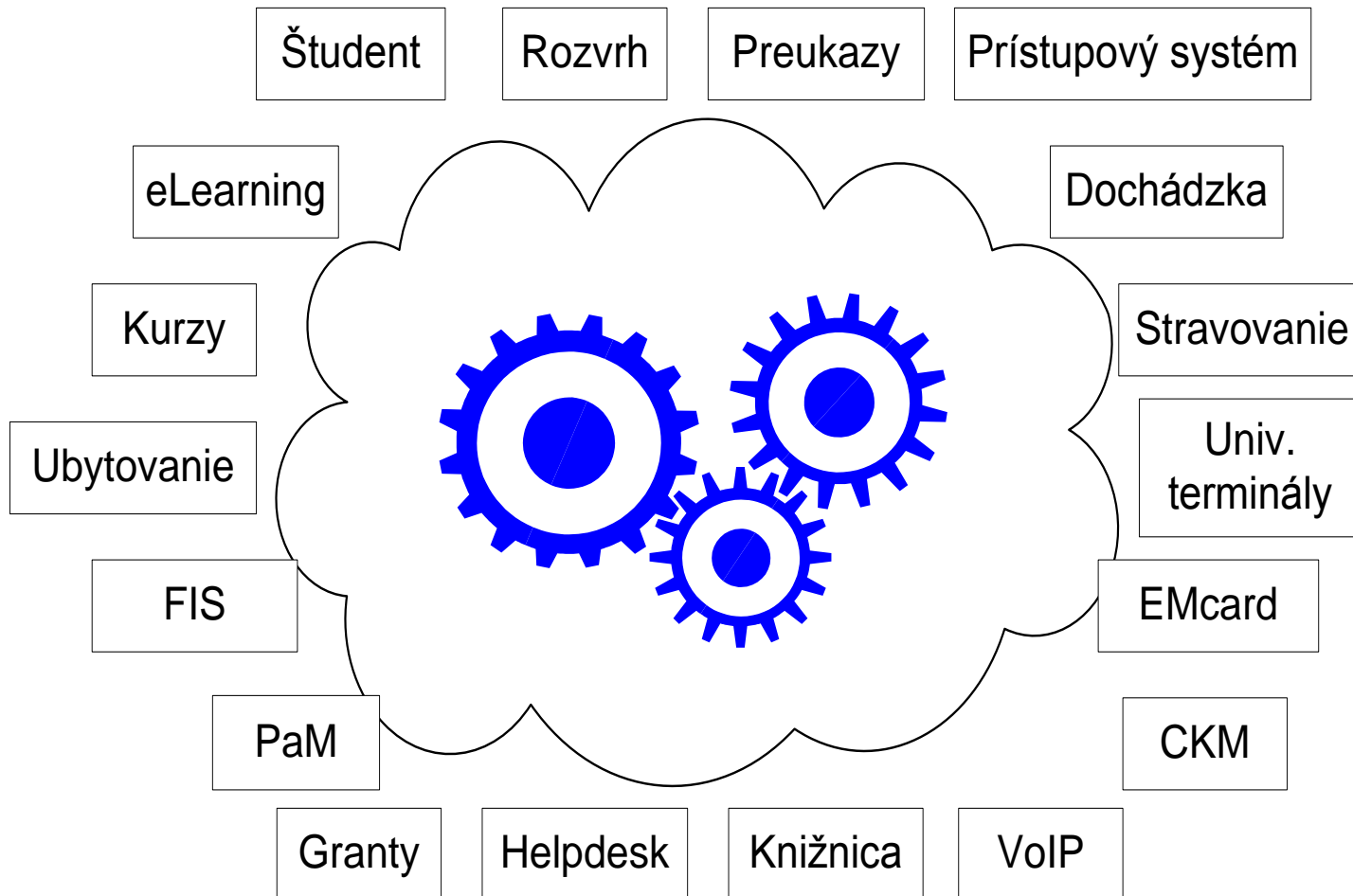
**Používateľské
rozhranie**

Aplikačná
logika

Perzistentné
údaje
(databáza)

Klasifikácia 3

Ako komunikovať vnútri int. riešenia?

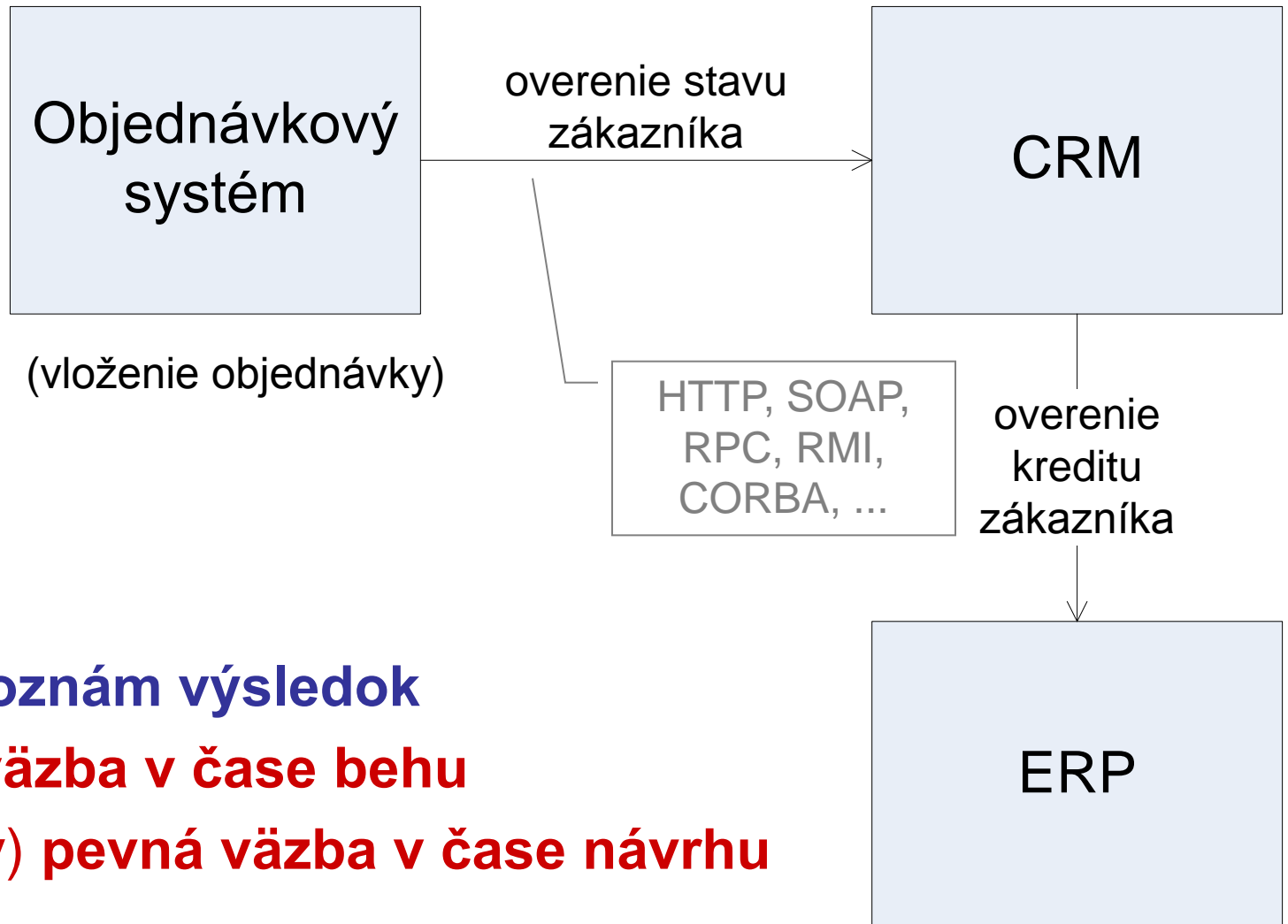


Klasifikácia 3

Ako komunikovať vnútri int. riešenia?

- Synchronne
- Asynchrónne, s využitím sprostredkovateľa
 - dátovody a filtre
 - manažér procesov

Synchrónna komunikácia



☺ **hneď poznám výsledok**

☹ **pevná väzba v čase behu**

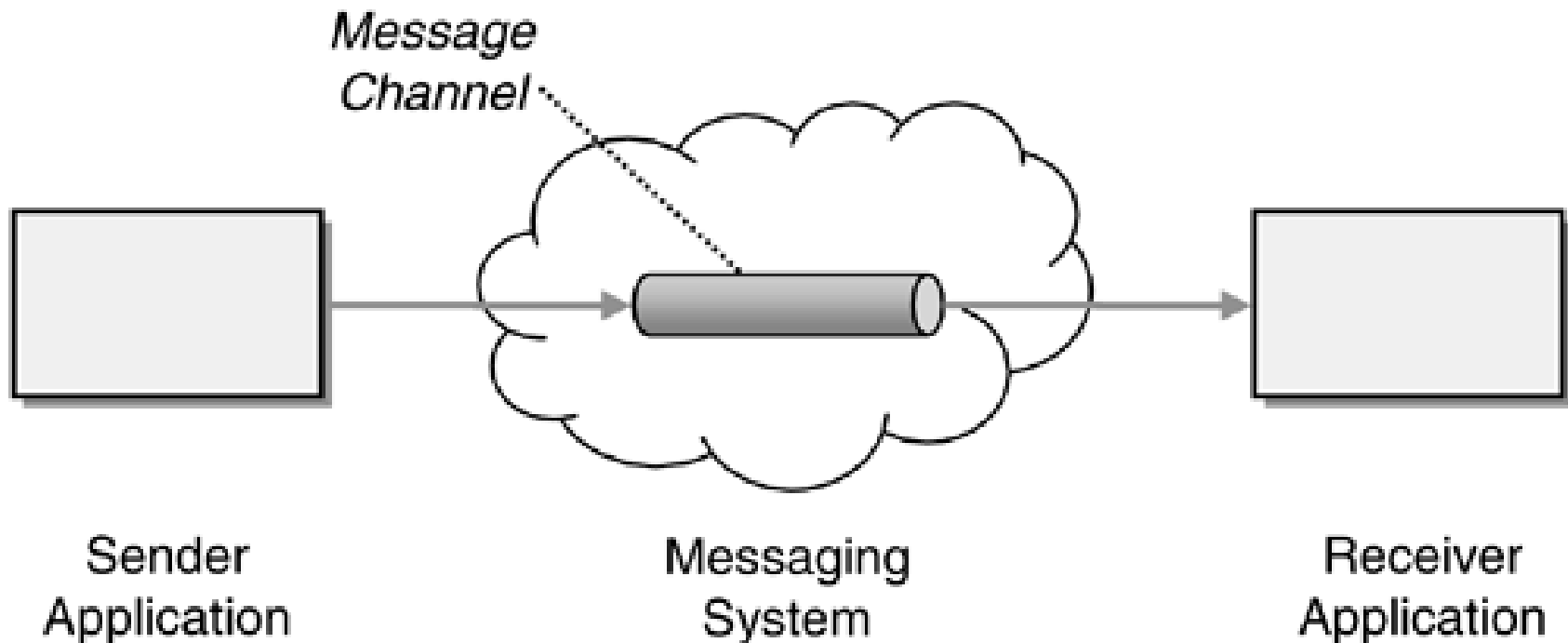
☹ (niekedy) **pevná väzba v čase návrhu**

Synchrónna komunikácia (2)

- synchrónny prístup sa niekedy používa pri budovaní **SOA** (architektúry orientovanej na služby)
- isté funkcie sú on-line prístupné pre aplikácie
 - napr. informácie o zákazníkoch
 - na UK napr. overenie „Level 0“ hesla, zistenie informácií o držiteľovi preukazu, ...
- tu je však extrémne dôležité definovať **SLA** pre tieto služby a dôsledne ho (ju) dodržiavať

Asynchrónna komunikácia

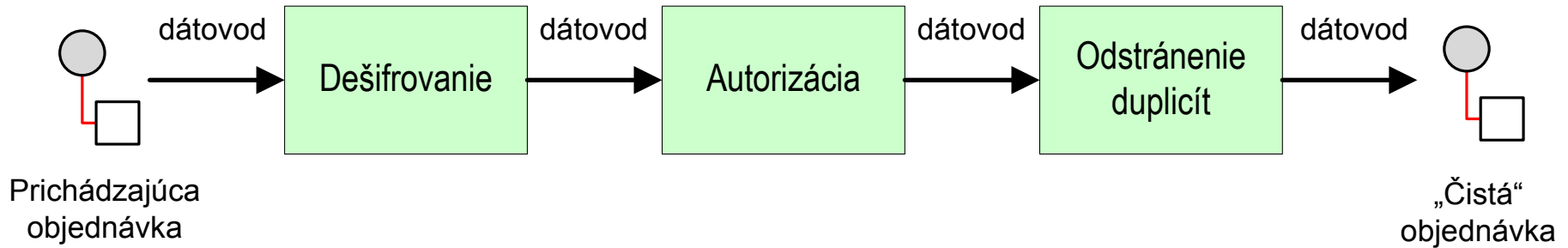
- aplikácie komunikujú **posielaním správ**
- pri komunikácii sa využíva **sprostredkovateľ** – broker
- broker obsahuje **komunikačné kanály**, prostredníctvom ktorých aplikácie komunikujú



Asynchrónna komunikácia (2)

- Výhody:
 - **voľná väzba v čase behu:** komunikujúce systémy nemusia byť on-line v tom istom čase: jediný komponent, ktorý musí neustále bežať, je broker
 - (niekedy) **voľná väzba v čase návrhu:** do int. riešenia je možné zapojiť rôzne mediačné komponenty, ktoré zaistia **transformáciu a smerovanie** správ, a tak uľahčia nezávislý vývoj komunikujúcich aplikácií
- Nevýhody:
 - zložitejšie programovanie
 - nevhodné, ak potrebujem poznať výsledok operácie hneď (i keď broker umožňuje aj synchrónnu kom.)

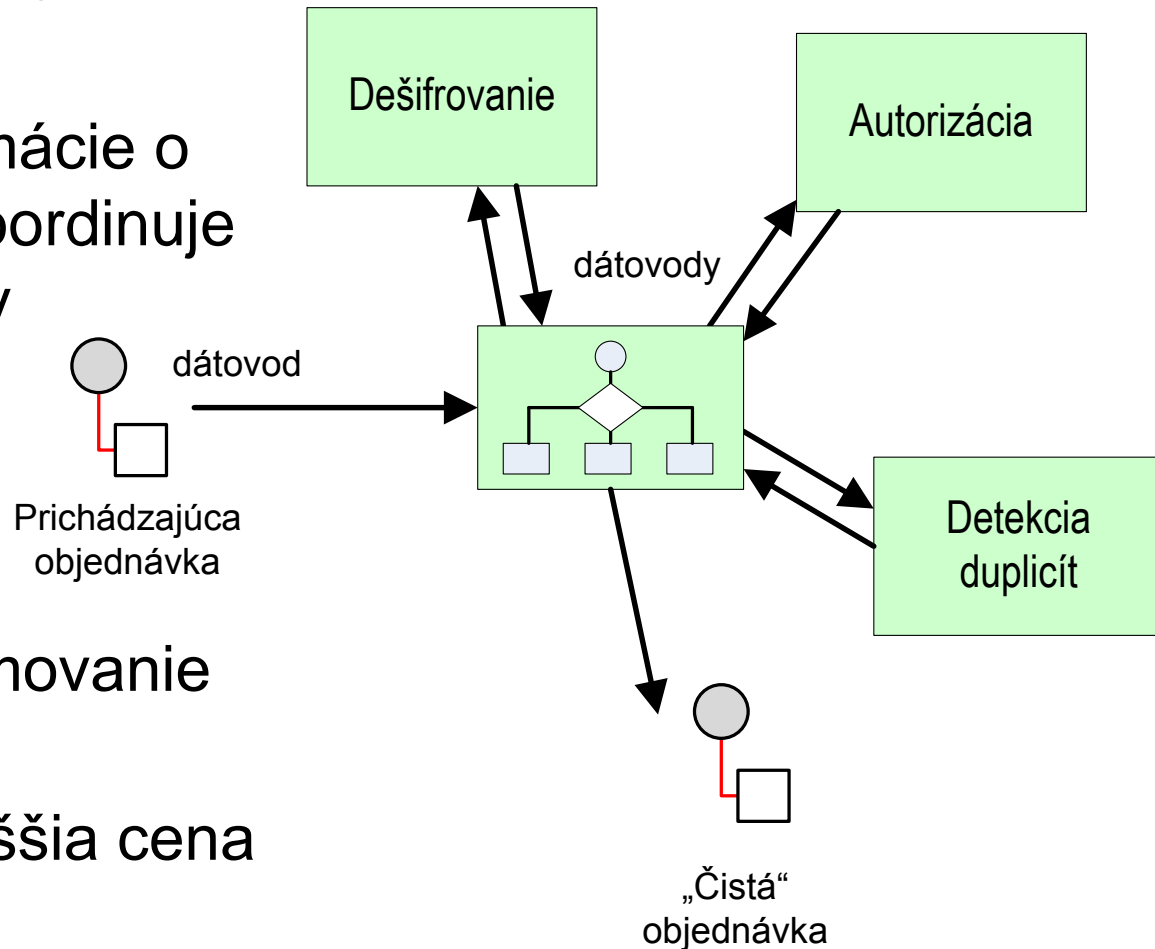
Dátovody a filtre (decentralizované riadenie)



- dátovody sú realizované kanálmi (v brokeri alebo v pamäti)
 - prenášajú správy, nemenia ich
- filtre so správami pracujú – správy transformujú, smerujú, ...
- vlastnosti dátovodov a filtrov sú často nastaviteľné správcom (typ kanálu, perzistentnosť, počet vlákien/procesov, transakčnosť spracovania, ...)

Správca procesov (centralizované riadenie)

- jeden z filtrov plní úlohu správcu procesov
- tento uchováva informácie o stave spracovania, koordinuje prácu ostatných filtrov



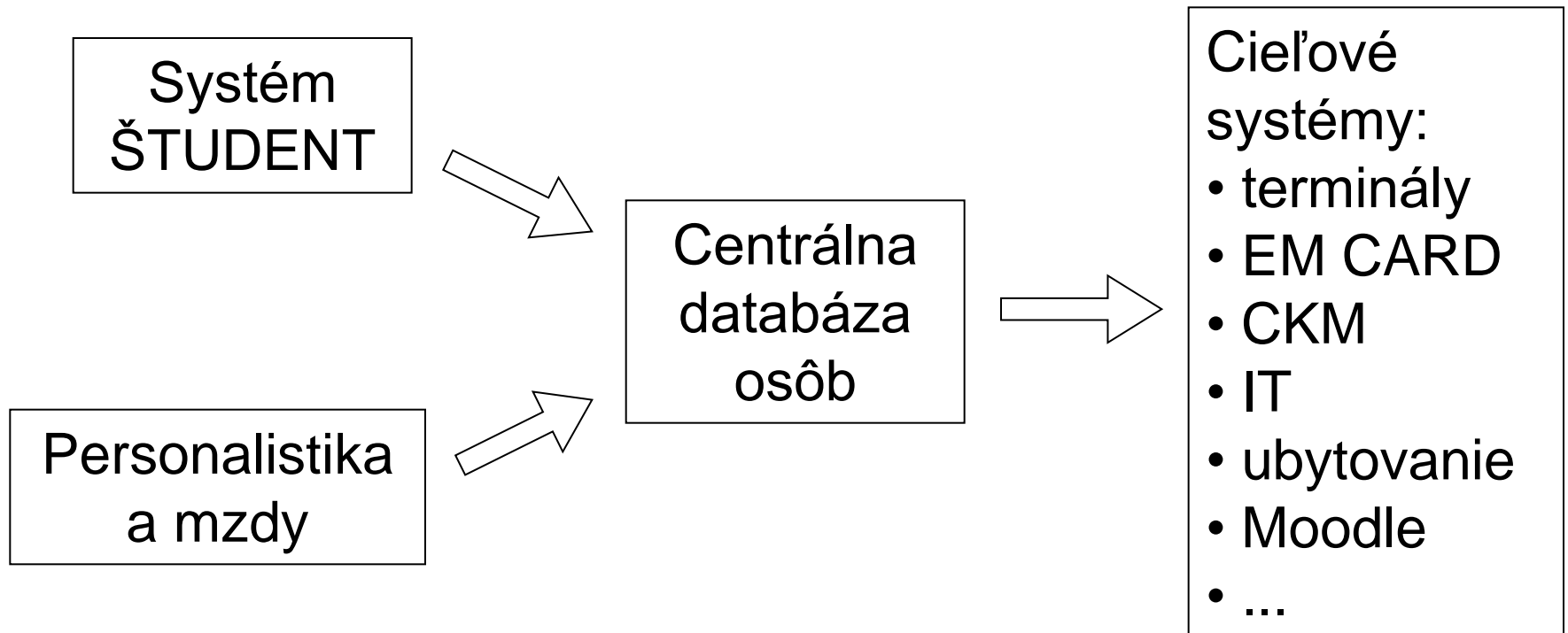
- jednoduchšie programovanie a administrácia
- nižšia efektívnosť, vyššia cena (licencie, podpora)

Integrácia na UK

- 2003: vydané preukazy študenta a zamestnanca
- leto 2004: čo s dopravou?
- aj ďalšie aplikácie:
 - tlač preukazov
 - prístupový systém
- zdrojové systémy:
 - Študent (13 inštancií)
 - personalistika (19 inštancií)
 - technológia DBF, nie všetky inštancie on-line

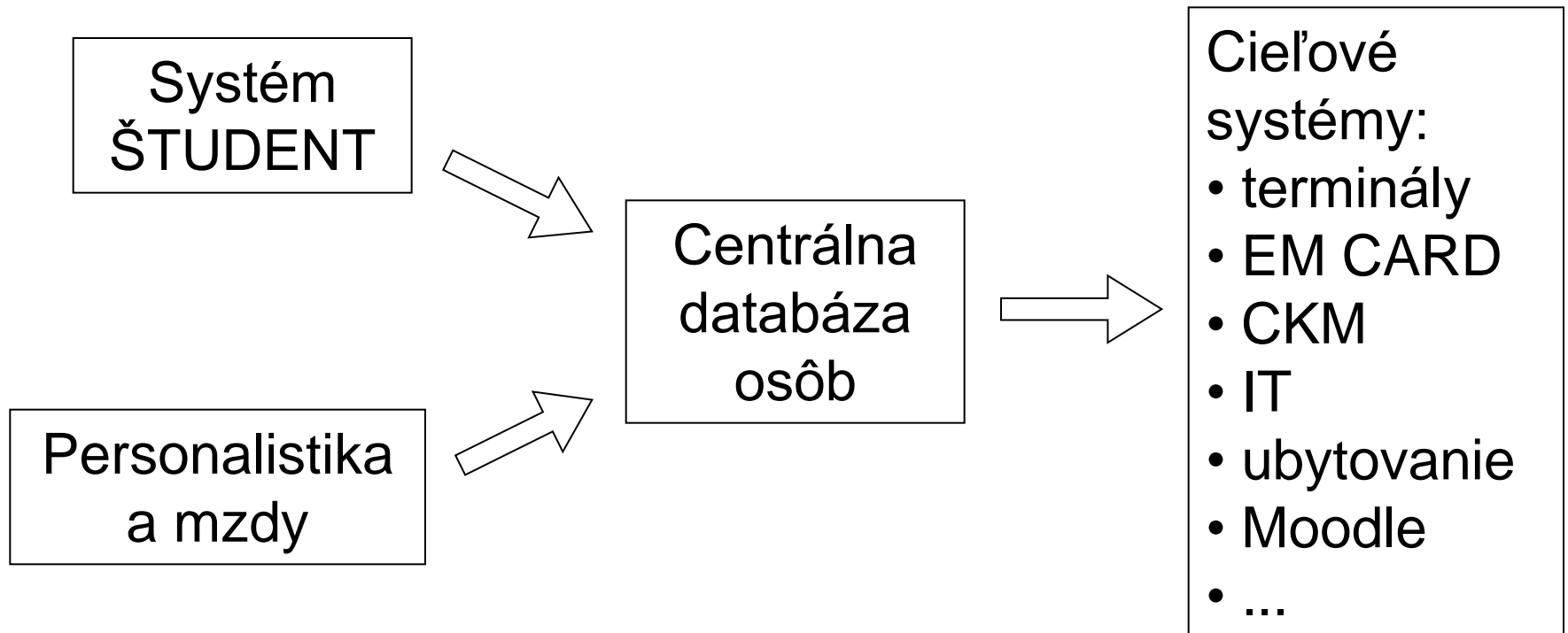


Centrálne databáza osôb



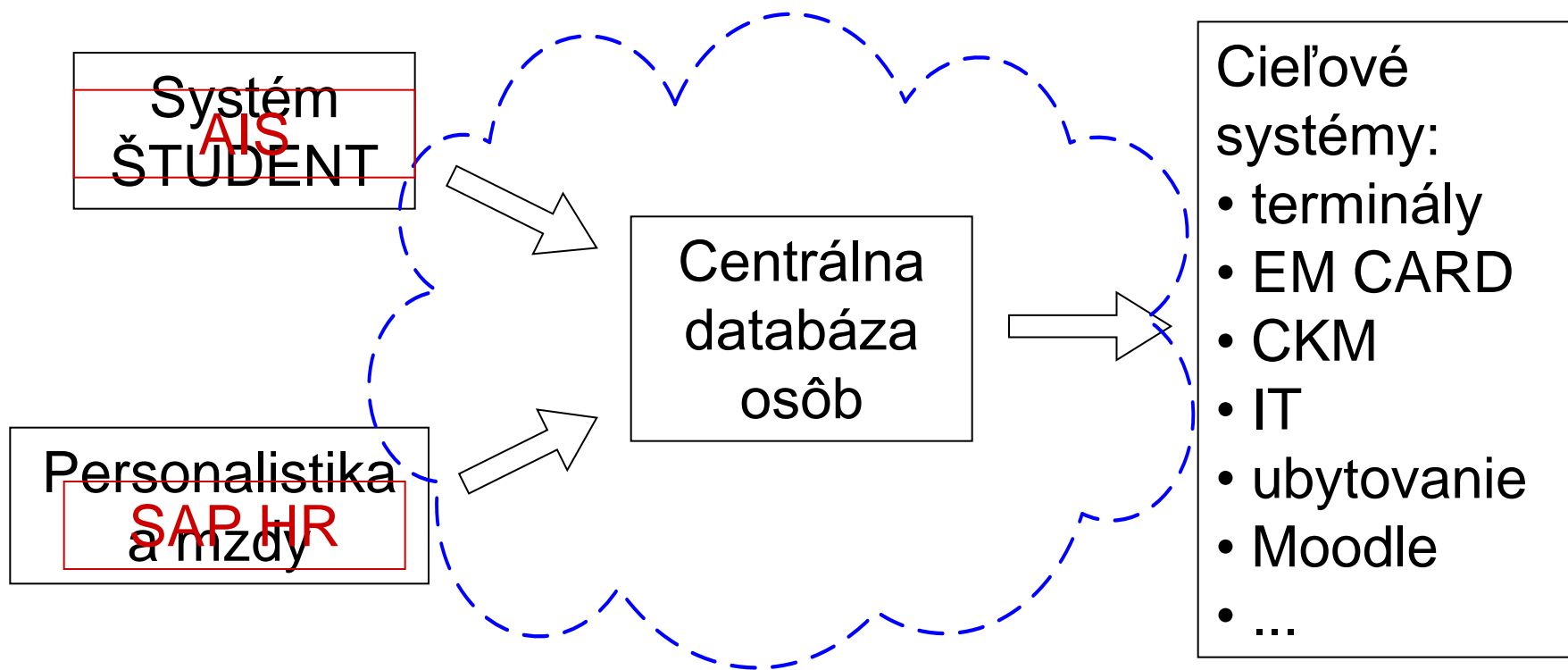
2004: maily a diskety

Centrálne databáza osôb



2005: ŠTUDENT – automatický prenos
(Sun Java System MQ)

Centrálna databáza osôb



2007: ESB (Progress Sonic ESB)

- asynchrónny prenos údajov (message queuing)
- transformácie údajov podľa potreby

Prenos údajov dnes

- prenos všetkých údajov (asynchrónny)
 - typicky každú noc (niekde raz za týždeň)
 - trvá pomerne dlho (desiatky sekúnd až desiatky minút)
 - zaručenie prenesenia všetkých údajov
- prenos zmien (asynchrónny)
 - okamžite po vykonaní zmeny
 - trvá veľmi krátko
 - v prípade neočakávanej chyby sa niektoré údaje môžu stratiť
- prístup k údajom na požiadanie (synchronný)
 - používa sa zriedkavo

(Niektoré) ťažkosti

- Konceptuálne hľadisko:
 - rôzny pohľad na fungovanie sveta
 - nutná (?) byrokracia v personálnych otázkach
 - zložité procesy týkajúce sa preukazov
 - kvalita údajov
- Technické hľadisko:
 - zložité dátové transformácie
 - asynchrónnosť spracovania, dátovody a filtre
 - spracovanie výnimočných situácií
 - zmeny v aplikáciách, dostupnosť aplikácií

Záver

- integrácia aplikácií: náročná a veľmi zaujímavá
- kľúčové slovo: **voľná väzba**
- existujú kvalitné nástroje, avšak stále je tu množstvo práce vývojárov
- literatúra na ďalšie štúdium
 - Hohpe & Woolf: **Enterprise Integration Patterns**, Pearson Education, 2004
 - Trowbridge et al.: **Integration Patterns**, Microsoft Corporation, 2004.
 - resp. **<http://www.eaipatterns.com/books1.html>**