

Architektura IS/IT
verze zkrácená
ES ESF MU 2006

Možné problémy neexistuje-li architektura IS/IT



Paralela se stavbou domu

- nepoužitelný materiál
- jiný materiál schází - nepokryté požadavky uživatelů na funkce IS,
- schází potřebné stroje a řemeslníci - tvorba IS v málo výkonném vývojovém prostředí, bez CASE a bez specialistů na tvorbu informační strategie, na počítačové sítě, na databáze atd.,
- často se musí bourat - opouštění nevhodně nakoupeného software nebo neustále se množící požadavky uživatelů, které jsou často v rozporu s logikou aplikace,
- nepoužitelné prostředky - draze nakoupený, ale z důvodů nekompatibility nevyužívaný hardware a software,

Možné problémy neexistuje-li architektura IS/IT:

- stavební projekty dle různých standardů - nepoužívají se žádné standardy, rozdílné uživatelské rozhraní aplikací, různé databázové systémy atd.,
- byla-li stavba strastiplným obdobím - bydlení je zcela nesnesitelné - časté poruchy, náročné opravy - viz údržba SW bez dokumentace

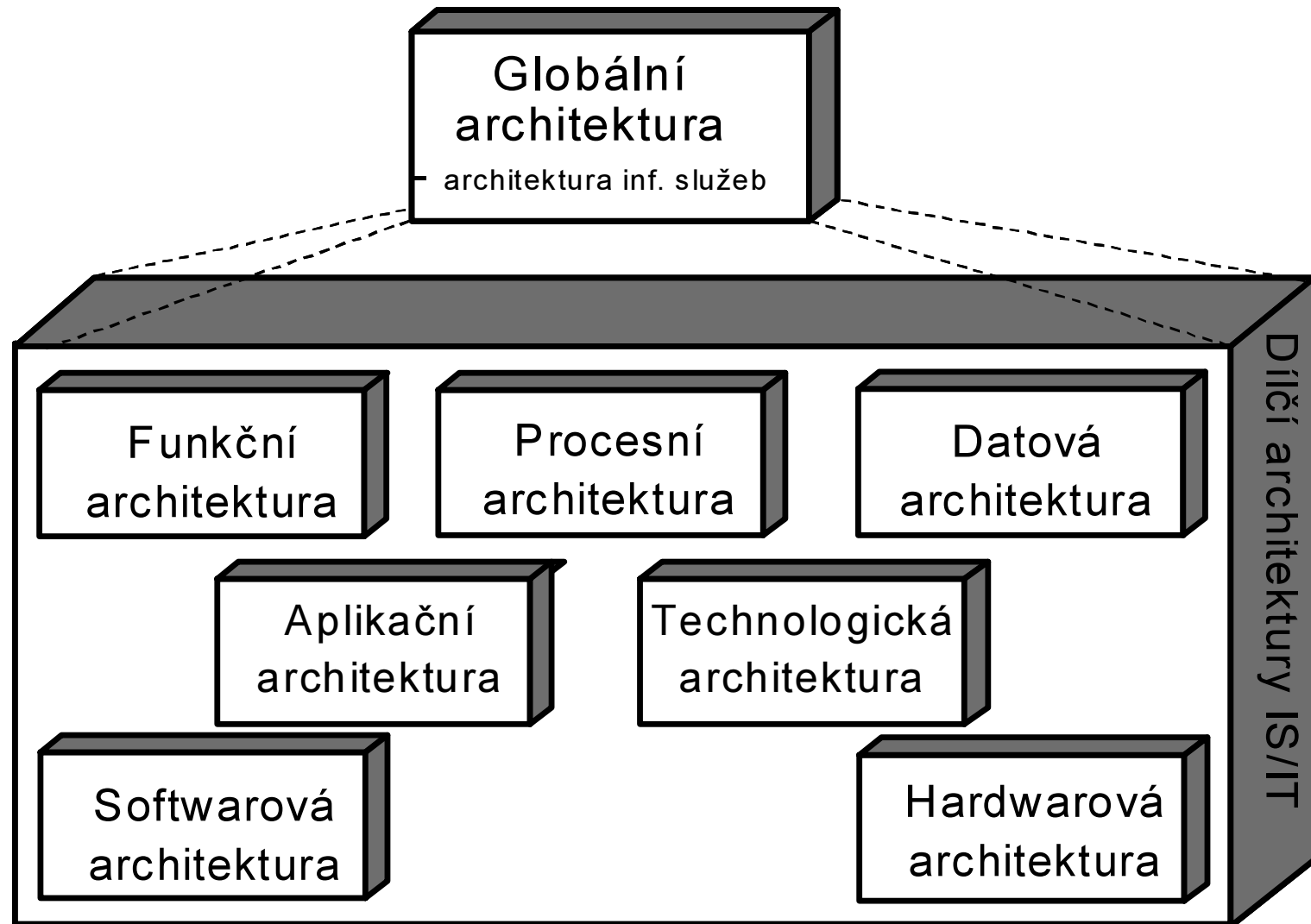
1. Globální a dílčí architektury

✱ Dvě úrovně architektury

- globální architektura = hrubý návrh celého IS/IT (viz generelní plán domu)
- dílčí architektura = detailní návrh IS/IT z hlediska různých dimenzí IS/IT (viz plán hrubé stavby, plán el. rozvodu, plán rozvodu vody a kanalizace)

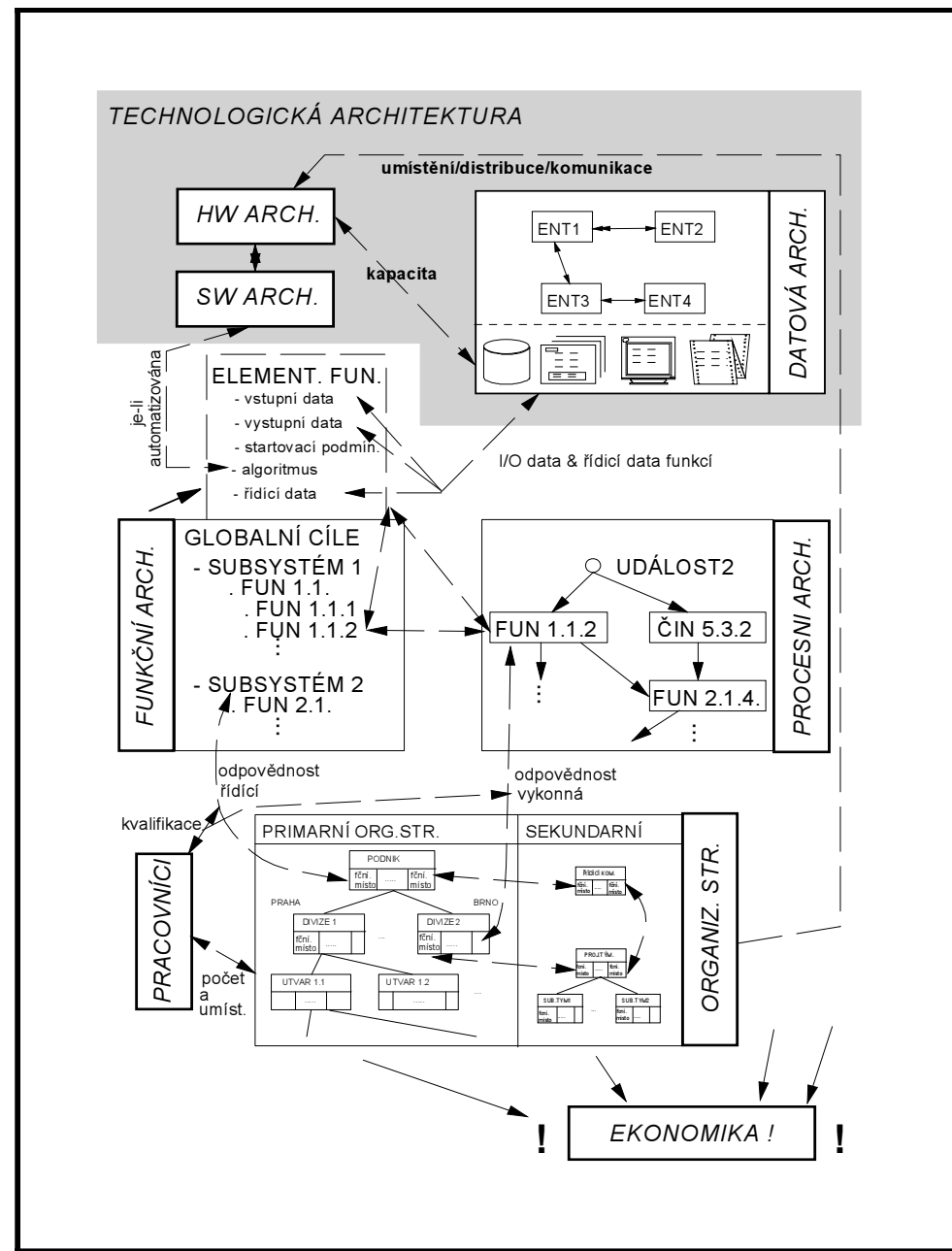


Globální a dílčí architektury



Globální a dílčí architektury

- Návrh dílčích architektur je zakončen vymezením vazeb mezi těmito architekturami.



Globální architektura IS/IT

- ✱ = vize budoucího stavu IS/IT, která zachycuje jednotlivé komponenty IS/IT a jejich vzájemné vazby

Základní stavební bloky architektury IS/IT

- ✱ = blok představuje množinu informačních služeb (funkcí), které slouží na podporu jednoho nebo více podnikových procesů
- ✱ alternativní pohled : množina informačních služeb pro různé skupiny uživatelů - veřejnost, partneři, zákazníci, zaměstnanci

Globální architektura dle skupin uživatelů



Vlastníci,
management

IS/IT obchodní firmy

Business Intelligence, manažerské aplikace

ERP

Prodej,
nákup,
výroba,
sklady

Finance,
Control-
ling, ...

Zdroje
(personál,
majetek),

CRM – B2C

Zákazníci

SCM – B2B

B2A

Obchodní
partneři –
dodavatelé,
dopravci,
banky,...

Státní
administrativa –
finanční úřad,
celní úřad,...

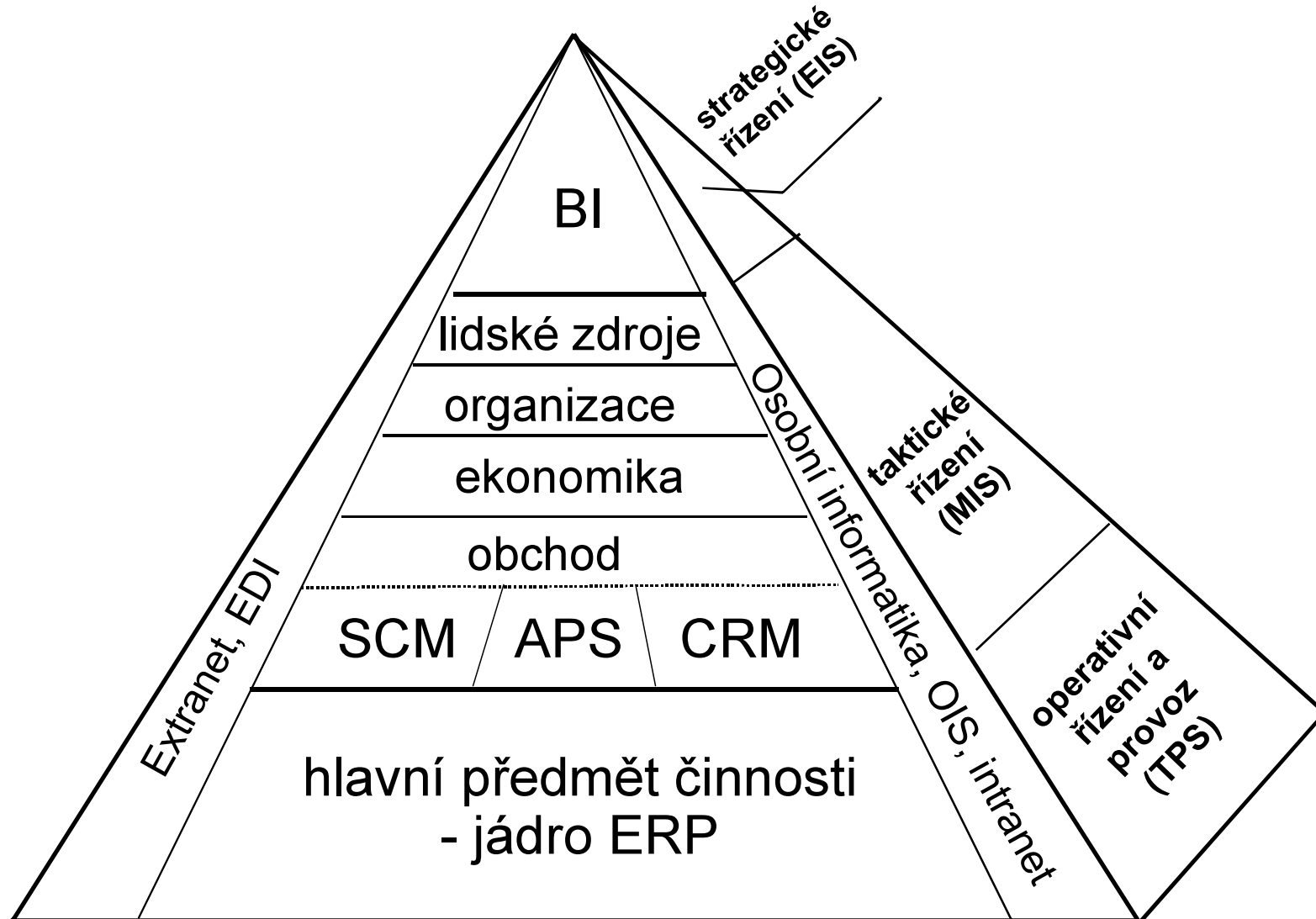
Interní infrastruktura

Zaměstnanci

Veřejnost



Globální architektura dle služeb podnikovým procesům



◆ TPS (Transaction Processing System)

- ✱ = blok zaměřený na podporu hlavní činnosti podniku na operativní úrovni - závislý na charakteru podniku,
- ✱ Výrobní podniky - CIM (Computer Integrated Manufacturing)
- ✱ Princip = integrace výrobních procesů na linii výrobní a zakázkové
- ✱ Dílčí komponenty CIM
 - CAD - (design) konstrukce,
 - CAP - (planning) automatizovaná technologická příprava,
 - CAT - (testing) automatizované testování výrobku,
 - PPC - (Production Planning and Control) - kapacitní plánování a operativní řízení výroby, pozor –DBR a TOC a Advanced Scheduling and planning
 - CAM - (manufacturing) automatizované dílenské řízení,
 - CAQ - (quality) - kontrola celého procesu výroby a kvality produkce

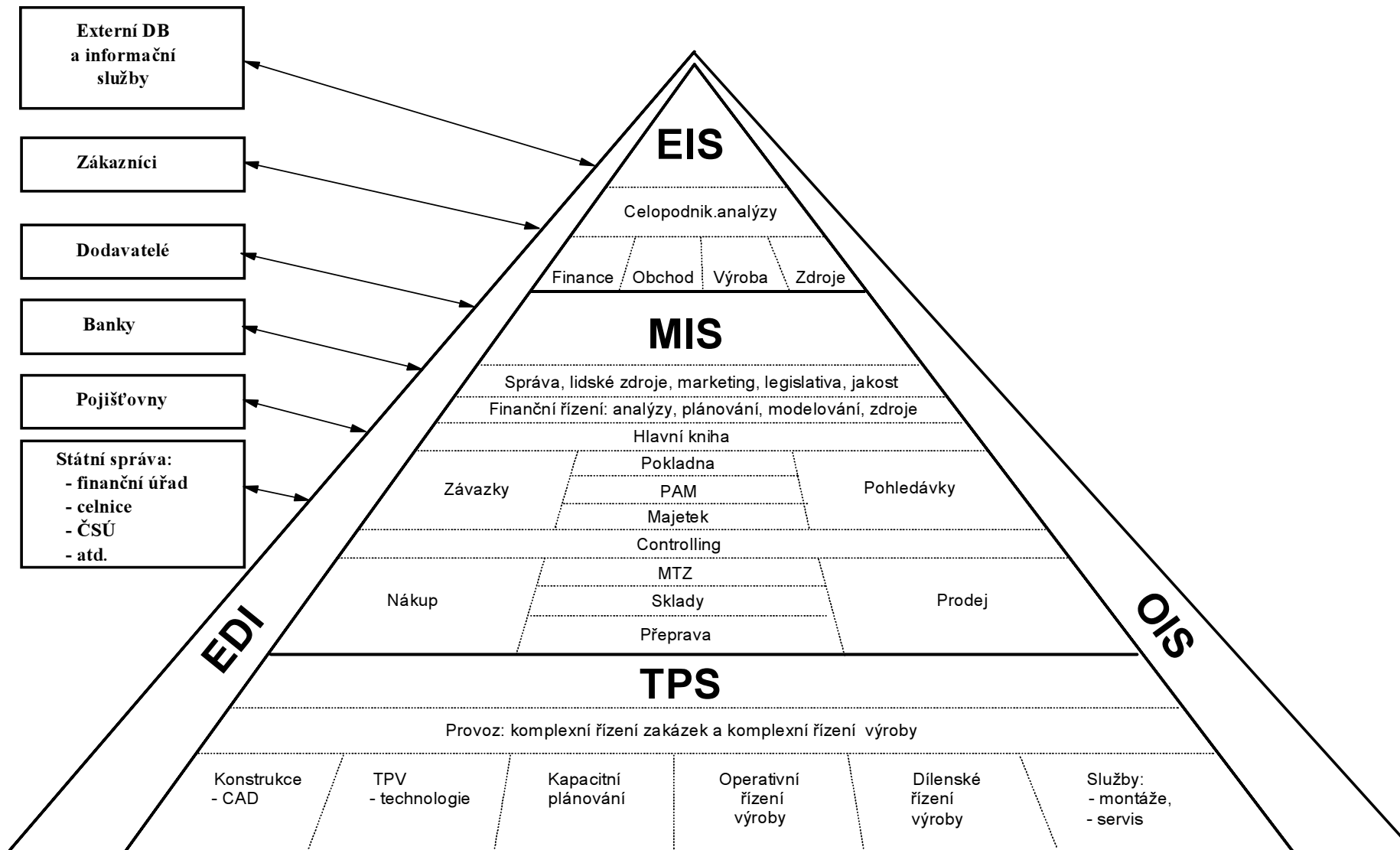
◆ MIS (Management Information System)

- ✱ = blok orientovaný na řízení podniku na taktické úrovni (střednědobé plánování) zahrnující hlediska:
 - ✱ obchodní (SCM, APS, CRM)
 - ✱ ekonomická (účetnictví, finance, daně, majetek)
 - ✱ organizační (org. struktura, lokality)
 - ✱ lidské zdroje

◆ EIS (Executive Information System)

- ✱ = blok orientovaný na strategické řízení podniku
 - data z TPS, MIS a externích IS agregují, vytvářejí časové řady a vzájemné vazby výstupy z EIS ⇒ podklady pro strategická rozhodnutí členů vrcholového managementu





Odlišnost EIS x TPS a MIS 1

- EIS zaměřeny na delší časový úsek (do minulosti a budoucnosti)
- pracují s historickými daty \Rightarrow možnost vysledovat vývojové tendence – data mining
- uchovávají údaje o stejném objektu vzniklé v různých časech \Rightarrow možnost hodnocení kvality
- často založeny na technologii data warehouse
- SW nástroje pracující na OLAP technologii, která pracuje s daty v n-dimenzionální tabulce, kde se sledují ekonomické ukazatele, časové a jiné pohledy managementu na sledovanou realitu – bude probíráno

Odlišnost EIS x TPS a MIS 2

..

- TPS a MIS udržují data vypovídající o právě aktuálním stavu interních a externích podnikových procesů
- historická data jsou udržována pouze mají-li vztah k současnosti (např. data zakázky uloženy v MIS dokud zakázka je v garanční době)

◆ OIS (Office Information System)

- ✱ = blok orientovaný na podporu kancelářských prací a na podporu týmové práce
- ✱ Zahrnuje:
 - textový editor, elektronické publikování,
 - tabulkový procesor,
 - prezentační program pro tvorbu obrázků, schémat a prezentací,
 - snímání papírových dokumentů a rozpoznávání jejich textu (document imaging),
 - plánovací kalendář,
 - sledování úkolů,

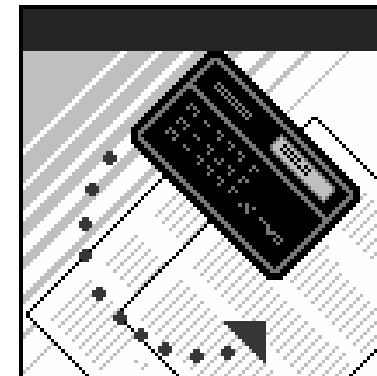


◆ OIS (Office Information System)

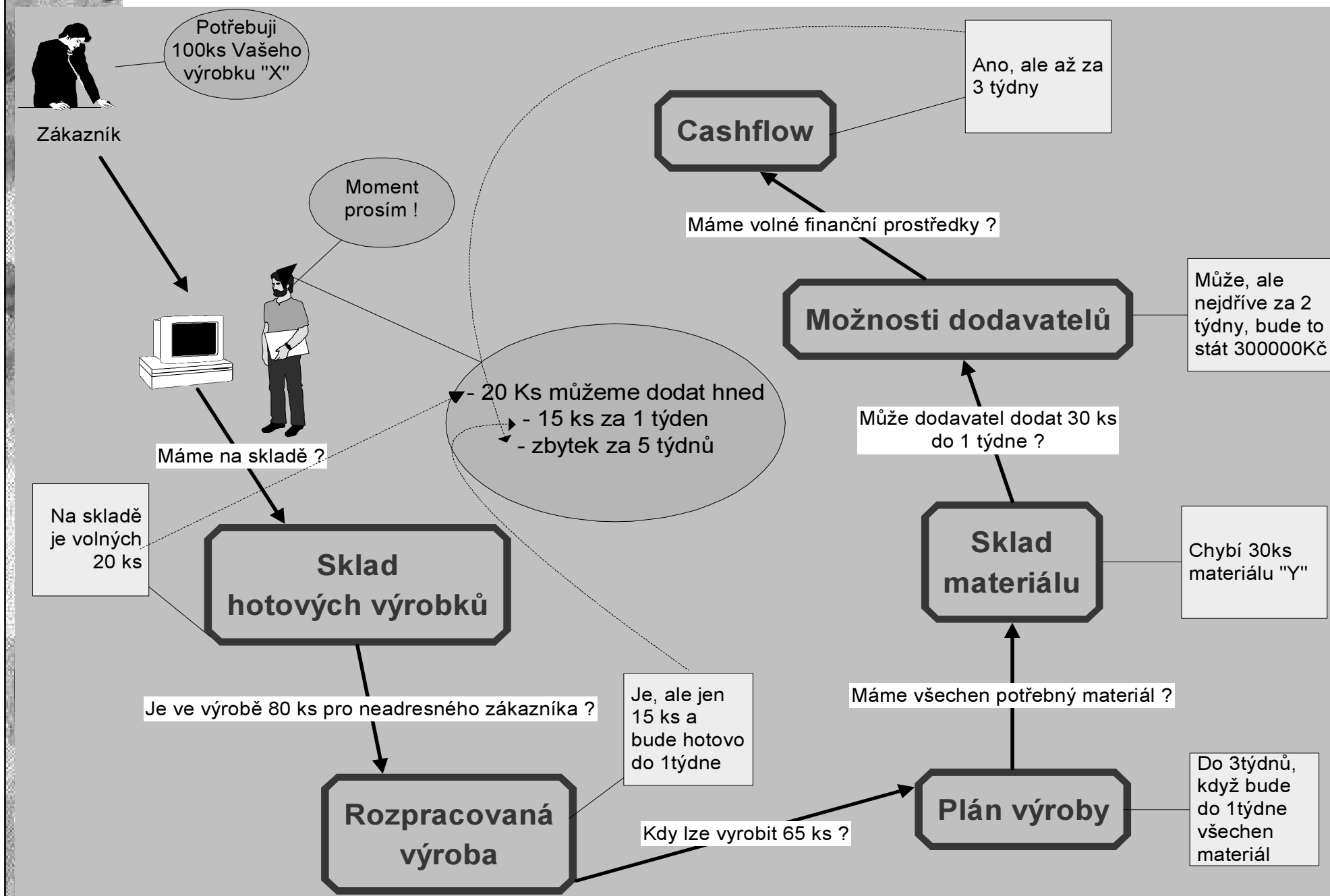
- elektronická pošta
 - elektronické diskusní skupiny,
 - videokonference,
 - tvorba a prohlížení WWW stránek,
 - řízení týmové práce na dokumentu,
 - archiv dokumentů.
- ✱ Podstatný prvek – workflow – bude probíráno
- ✱ = průřez přes všechny úrovně řízení podniku; vazba na výsledky BPR (zkratku do slovníčku)

◆ EDI (Electronic Data Interchange)

- ✱ = blok zajišťující komunikace podniku s jeho významným okolím
 - standardizovaná výměna dat



◆ Vazby mezi stavebními bloky IS/IT

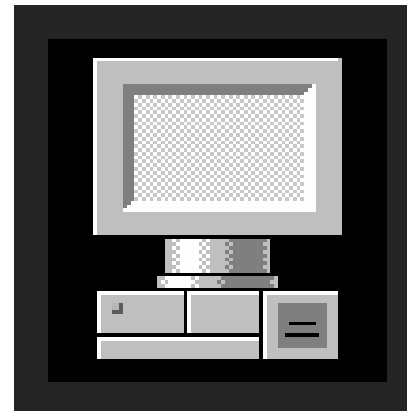


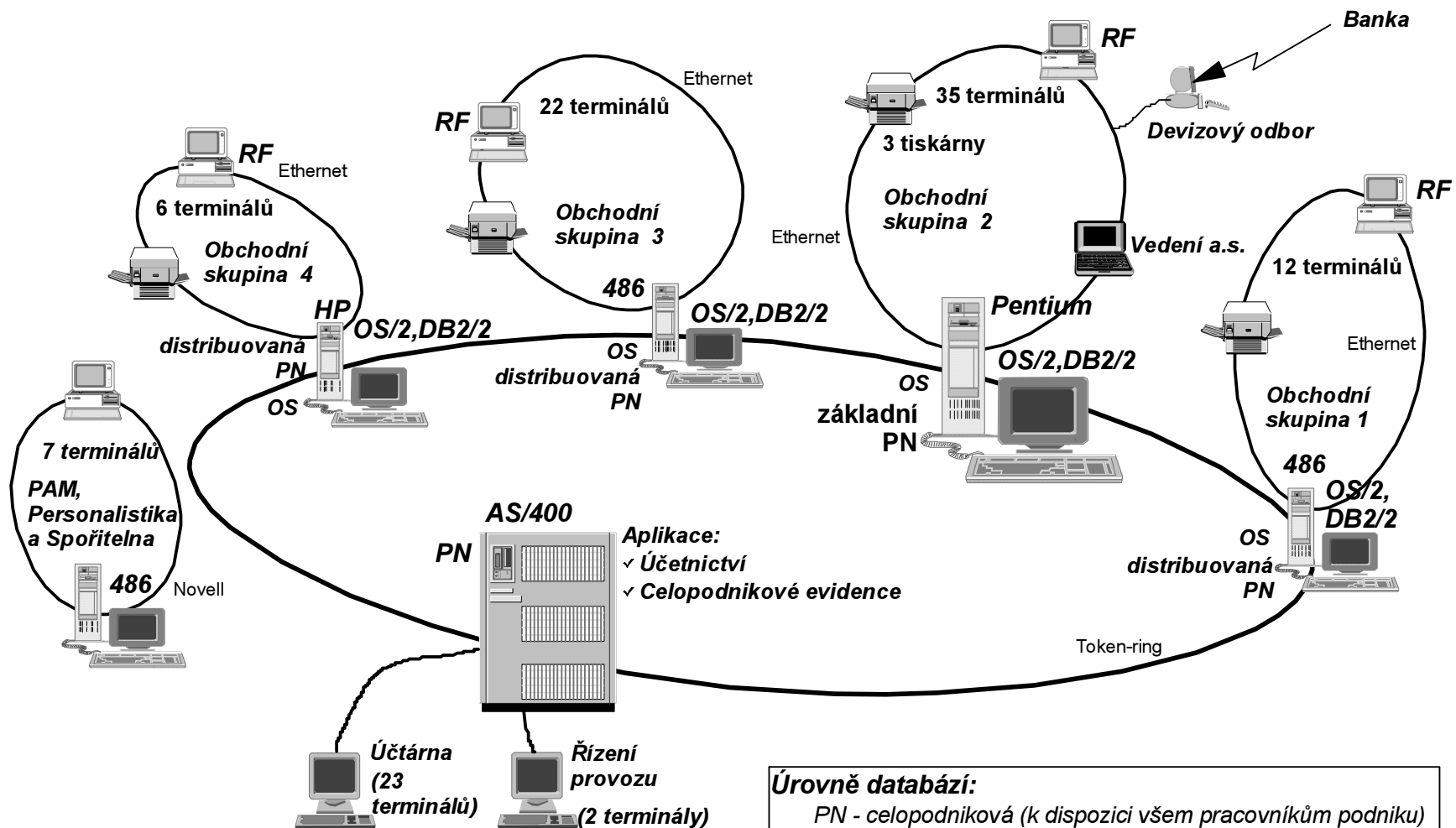
2. Dílčí architektury

- řeší samostatně jednotlivé obsahové dimenze IS
- datová, funkční, procesní

◆ *Hardwarová architektura*

- ✱ určuje typy, počty a vzájemné vazby hardwarových komponent - personálních počítačů (koncových stanic), serverů, tiskáren a dalších přídatných zařízení, použitých přenosových cest



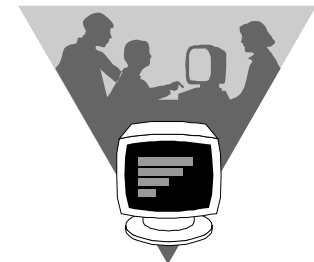


Úrovně databází:
 PN - celopodniková (k dispozici všem pracovníkům podniku)
 OS - celoskupinová (k dispozici prac. obchodní skupiny)
 RF - referentská (osobní databáze referentů)

Obr. 1: Hardwarová architektura IS/IT obchodní organizace (velkoobchodní prodej, výrokově-teritoriální organizace)

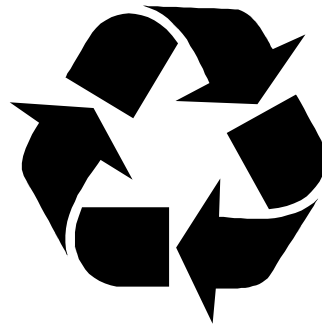
◆ *Technologická architektura*

- ✱ rozhoduje o technologickém řešení aplikace, resp. jednotlivých funkcí aplikace
- ✱ propojuje softwarovou, hardwarovou a datovou dimenzi a definuje způsob zpracování jednotlivých aplikací IS (dávkové, interaktivní, řízené událostmi, v reálném čase), vnitřní stavbu aplikací (klient/server architektura, tříúrovňová architektura apod.) a uživatelské rozhraní aplikací.
- ✱ může být pro jednotlivé aplikace odlišná



zpracování - dávkové / interaktivní / řízené událostmi / v reálném čase

- ✱ Metoda zpracování určuje, jakými podněty jsou jednotlivé funkce aplikace startovány, jaká je doba odezvy funkcí a jaký vztah má zpracování funkcí aplikace k procesům reálného světa



Dávkové zpracování

- ✿ jednotlivé požadavky na zpracování a související vstupní data jsou shromážděna v dávce před odstartováním aplikace, která po svém spuštění zpracuje najednou všechny shromážděné požadavky
- ✿ Příklad: sběr a doručování poštovních zásilek
- ✿ Výhody:
 - snadná programová realizace
 - malé nároky na počítačové zdroje
- ✿ Nevýhody:
 - dlouhá a nezaručená doba odezvy
 - bez komunikace s uživatelem
- ✿ Příklad: účetní uzávěrka

Interaktivní zpracování

- ✿ uživatel je v přímém kontaktu s počítačem a jeho požadavky na zpracování jsou vyřizovány okamžitě a s garantovanou dobou odezvy a jsou realizovány jednou transakcí
- ✿ Příklad: vyřizování telegramů na poště
- ✿ nejrozšířenější metoda zpracování sociálně-ekonomických aplikací

- ✿ Výhody:
 - uživatelsky příjemnější
- ✿ Nevýhody:
 - náročné na tvorbu
 - náročné na potřebu počítačových zdrojů

Aplikace řízené událostmi

- startovány událostmi (datové, časové, mimořádné), které nastávají v reálném světě
- *Př. automatické vystavení objednávky (datová), pravidelné automatické odesílání údajů (časová)*
- Výhody:
 - zvyšují automatizaci a tím obvykle i efektivnost podnikových procesů.

Aplikace pracující v reálném čase

- aplikace řízené událostmi s tím, že jejich doba odezvy přesně odpovídá podnikovým procesům, které aplikace řídí
- *Př. přímé řízení strojů a celé výrobní linky počítačem*



centralizované / decentralizované / distribuované / kooperativní zpracování

Centralizované zpracování

- ✱ využití hlavního počítače, na který jsou napojeny všechny koncové stanice
- ✱ veškerá data i programy aplikace jsou umístěny na hlavním počítači
- ✱ Výhody:
 - jednoduchá tvorba a řízení provozu aplikace
- ✱ Nevýhody:
 - přetížení hlavního počítače, který nemůže být specializován, ale musí realizovat všechny části algoritmů všech aplikací včetně zpracování grafického rozhraní pro všechny komunikující uživatele

centralizované / decentralizované / distribuované / kooperativní zpracování

Decentralizované zpracování

- ✱ založeno na samostatných počítačích, mezi kterými neexistují žádné komunikační linky.
 - ☞ komunikace mezi počítači nemůže být řízena ⇒ propojení přes off-line přenášená data na páskách, disketách apod.
- ✱ Výhoda:
 - umožňuje práci v oddělených lokalitách
- ✱ Nevýhody:
 - nekonzistence v datové základně podniku,
 - narušení plynulosti podnikových procesů
- ✱ ⇒ použitelné pouze jako přechodová varianta v případě, kdy mezi vzdálenou lokalitou podniku a centrem neexistuje vhodná přímá komunikační cesta

centralizované / decentralizované / distribuované / kooperativní zpracování

Distribuované zpracování

- využití několika navzájem propojených počítačů (serverů) – obvykle specializovaných, na které jsou napojeny koncové stanice
- algoritmus zpracování a data jsou rozdělena na několik částí, které jsou umístěné na různých počítačích a vzájemně spolu komunikují - viz klient/server architektura a vícevrstvová architektura
- Výhody:
 - distribuovaný charakter odpovídá charakteru podnikových procesů
 - výpadky nemají takový dopad jako při centralizovaném zprac.
 - kratší doba odezvy z lokální DZ
- Nevýhody:
 - nutnost zajistit relativně složitou koordinaci zpracování aplikace na různých počítačích
 - komplikovaná ochrana a zabezpečení aplikace

centralizované / decentralizované / distribuované / kooperativní zpracování

Kooperativní zpracování

- ✿ kooperace mnoha počítačů z celosvětové počítačové sítě
 - ☞ vyšší forma distribuovaného zpracování
 - ☞ využití u WWW aplikací



◆ *klient/server architektura*

- ✱ aplikace je rozdělena na dva nebo více kooperujících programů
- ✱ *Klient* = program, který vyžaduje provedení určité služby
- ✱ *Server* = danou službu na požádání poskytuje
- ✱ Jeden a tentýž program přitom jednou může vystupovat jako klient, jindy jako server

◆ *klient/server architektura*

✿ Výhody :

- nižší nároky na server,
- lze volit vhodnou platformu (hardware a základní software) pro jednotlivé části aplikace,
- snadnější zabezpečení ochrany dat pomocí specializovaného datového serveru.

✿ Nevýhody:

- vyšší počáteční investice nutné pro zprovoznění,
- problémy s integrací různých platforem,
- vyšší nároky na kvalifikaci řešitelů.

monolitická, dvouvrstvá a třívrstvá architektura

- ✿ Obvyklá aplikace obsahuje tři základní skupiny funkcí:
 - datové funkce,
 - věcně orientované funkce, které zajišťují vlastní logiku aplikace,
 - komunikační funkce.

Monolitická architektura

- ✱ všechny skupiny funkcí jsou realizovány jedním programem (modulem)
- ✱ Výhody:
 - snadné zajištění ochrany funkcí a dat aplikace před neautorizovaným použitím
 - snadné zajištění aplikace proti výpadkům
- ✱ Nevýhody/problémy:
 - obtížná údržba,
 - obtížná přenositelnost aplikace mezi různými platformami,
 - obtížné inkrementální zvyšování výpočetních kapacit dle růstu počtu uživatelů a jejich nároků.

Dvouvrstvá architektura

- ✱ jedna ze skupin funkcí je oddělena od ostatních
- ✱ typická pro jednoúvarové aplikace do několika desítek uživatelů
- ✱ Dělení
 - lehký klient
 - těžký klient
- ✱ Cíl: oddělení komunikační a datové funkce aplikace



(a) Lehký klient



(b) Těžký klient

Dvouvrstvá architektura

☀ Výhody:

- v aplikaci se snadněji zajišťují různé formy komunikace s různými koncovými stanicemi a s různými uživateli,
- zvyšuje se přenositelnost aplikace,
- lze snadněji plánovat a doplňovat výpočetní kapacity.

☀ Nevýhody:

- přetrvává problém přílišného propojení věcně orientovaných funkcí s jednou z dalších skupin funkcí

Třívrstvá architektura

- ✱ všechny tři skupiny funkcí jsou odděleny do samostatných programů a ty mezi sebou komunikují jako klient/server
- ✱ ideální pro tvorbu otevřených, distribuovaných a flexibilních informačních systémů, které lze pružně přizpůsobovat změnám
- ✱ typická pro celopodnikové rozlehlé aplikace dynamického charakteru

- ✱ Výhody:
 - každou ze skupin funkcí lze udržovat a rozvíjet zcela samostatně,
 - pro každou ze skupin funkcí je možné zvolit nejvýhodnější vývojové a provozní prostředí

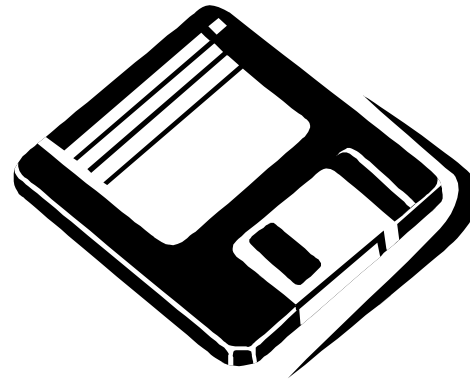
◆ *Softwarová architektura*

- ✱ určení SW komponent IS a jejich vazeb
- ✱ určení vnitřní struktury (modulů a jejich vazeb) SW komponent
- ✱ Charakteristiky modulů:
 - funkce, které modul zajišťuje,
 - vstupní, výstupní a řídicí data modulu,
 - algoritmus, který předepisuje způsob transformace vstupních dat na výstupní a způsob ošetření mimořádných stavů,
 - vývojové prostředí modulu (programovací jazyk, CASE atd.),
 - provozní prostředí modulu (operační systém, databázový systém, prezentační systém),

◆ *Softwarová architektura*

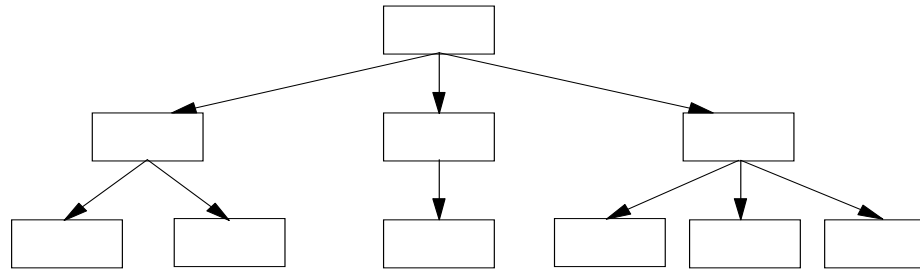
✱ Typy SW architektur:

- lineární,
- hierarchická,
- vrstvená,
- síťová.

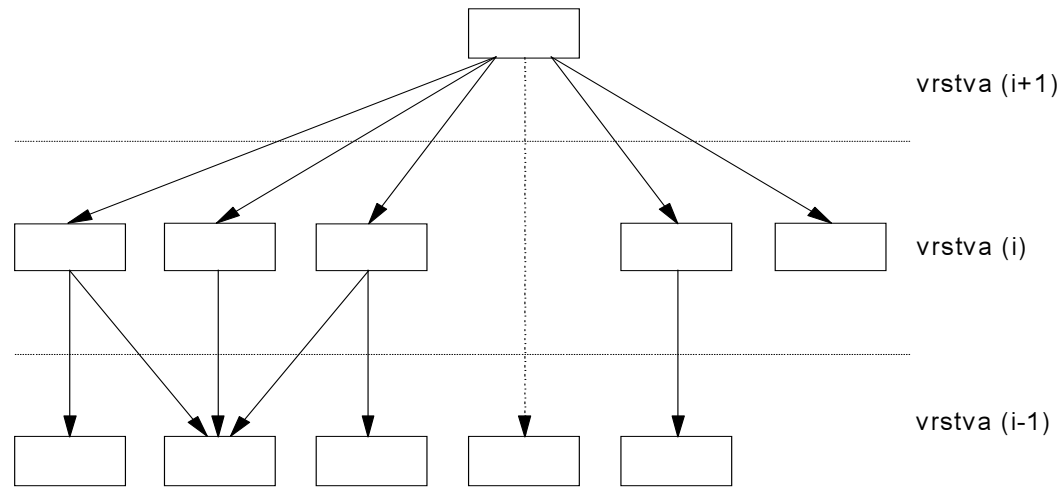




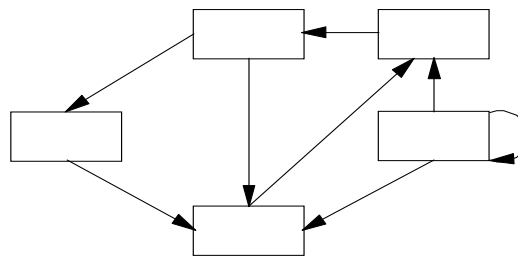
(a) lineární architektura



(b) hierarchická architektura



(c) vrstvená architektura



(d) síťová architektura

◆ *SW architektura*

- univerzálně jsou použitelné pouze vrstvená a síťová architektura (lineární a hierarchická pouze pro specifické aplikace)
- síťová je preferována v případech, kdy musíme preferovat nízké náklady provozu před nízkými náklady tvorby, údržby a užití (v ostatních případech je vhodnější vrstvená architektura)



3. *Vrstvená architektura*

◆ *Základní cíl VA*

- ✱ minimalizace nákladů tvorby a údržby a nákladů užití softwarového systému

Vrstvená architektura

uspořádání funkcí
do několika vrstev
s tím, že funkce
vyšší vrstvy
mohou využívat
jen funkce vrstev
podřízených

Typické vrstvy současných počítačů

