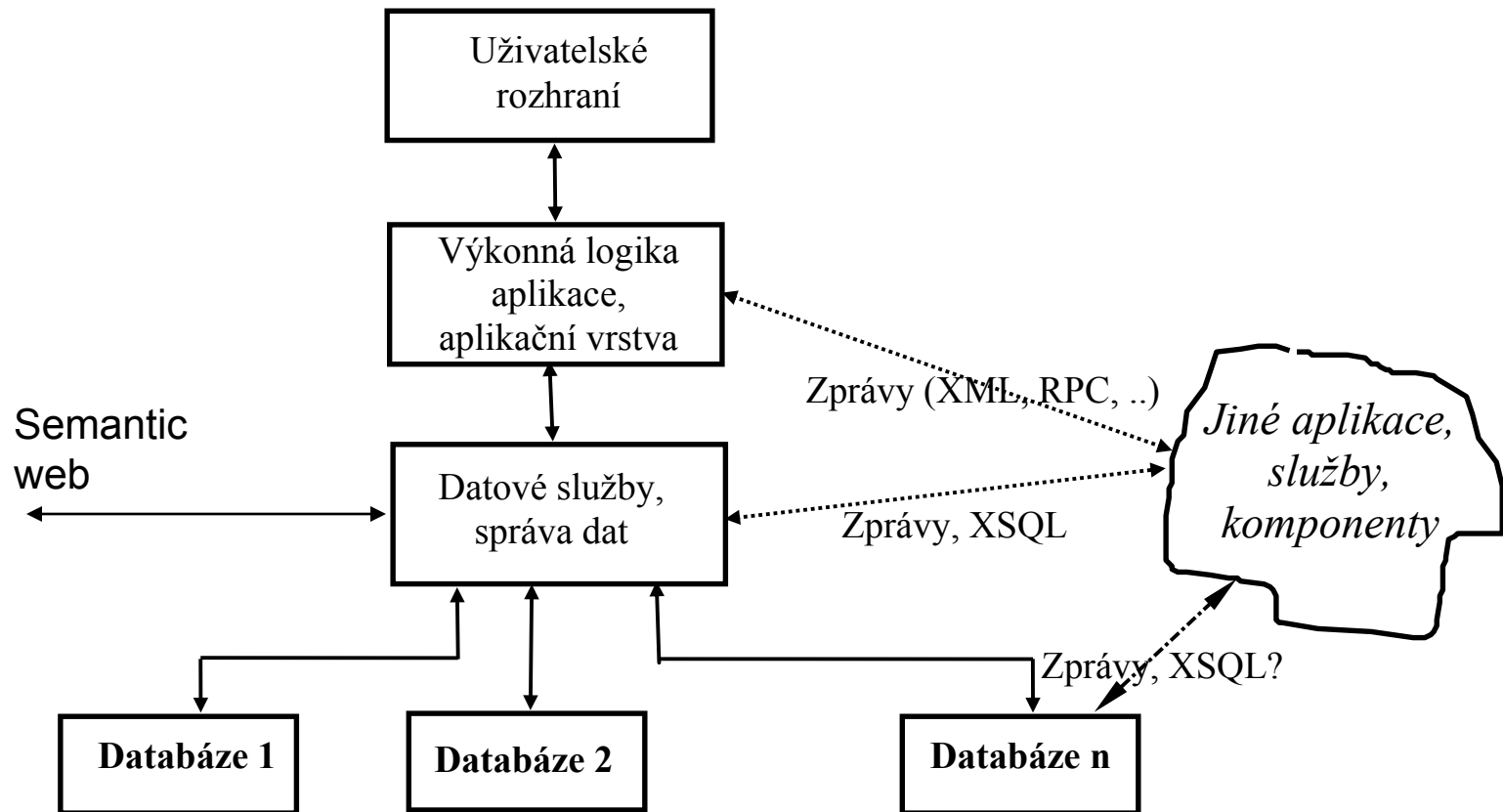


# Struktura týmu a SW architektura

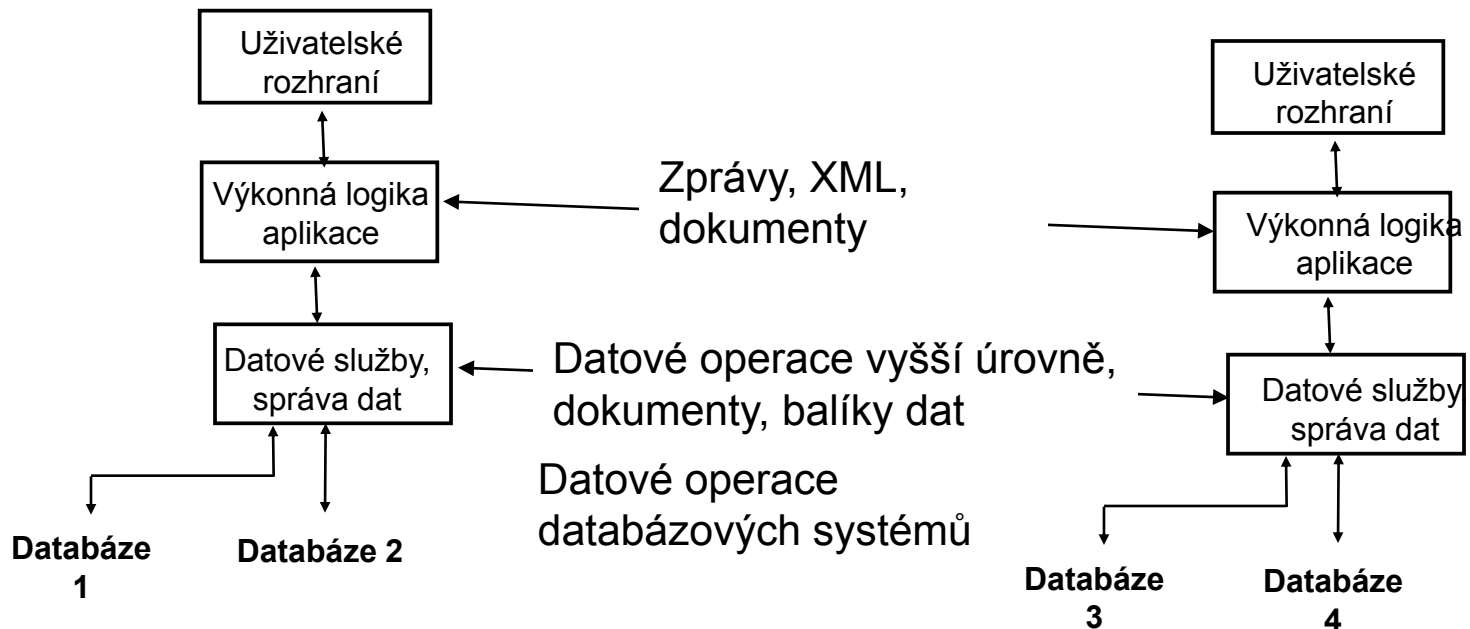
Mnohé role v týmu závisí na aktivitách ve zvolené architektuře SW

# Spolupráce třívrstvých komponent



# Spolupráce třívrstevných komponent

- Spolupráce podle vrstev,
  - Logika je na aplikačním serveru, datová na datovém

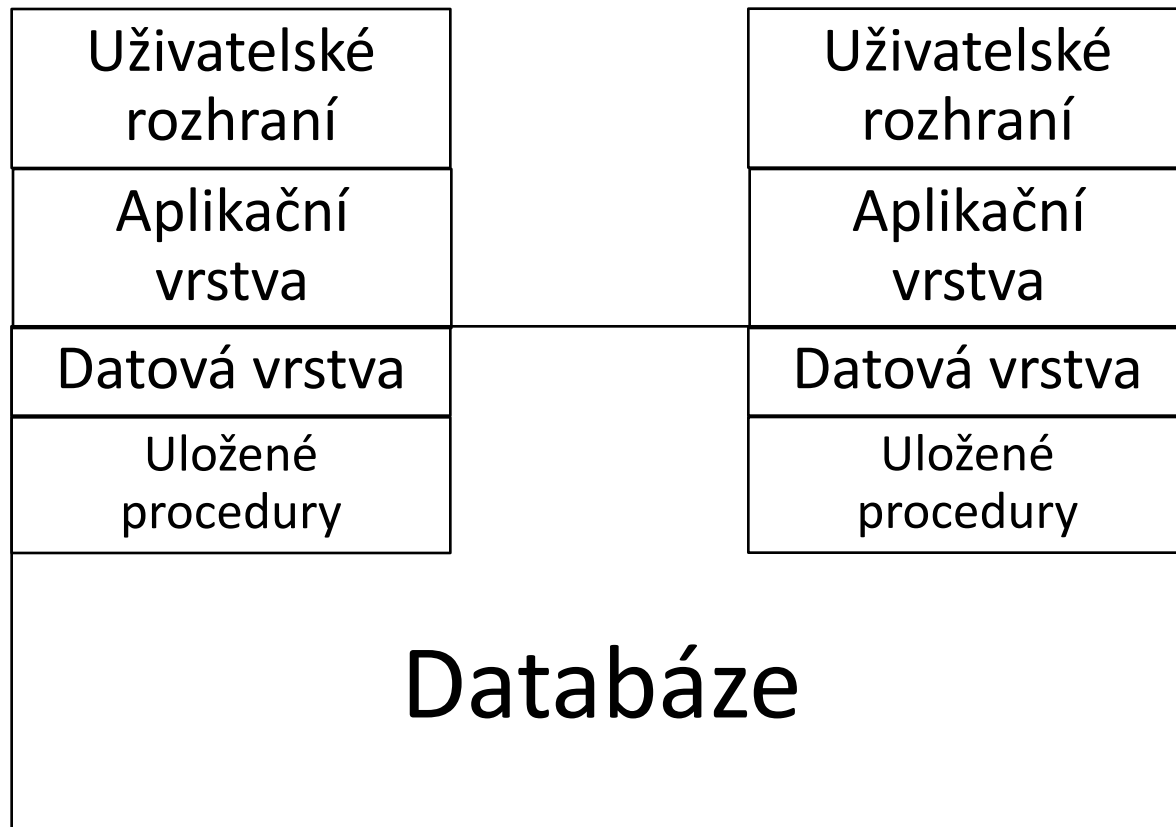


V SOA mohou být vrstvy tvořeny podsítěmi (opět SOA)

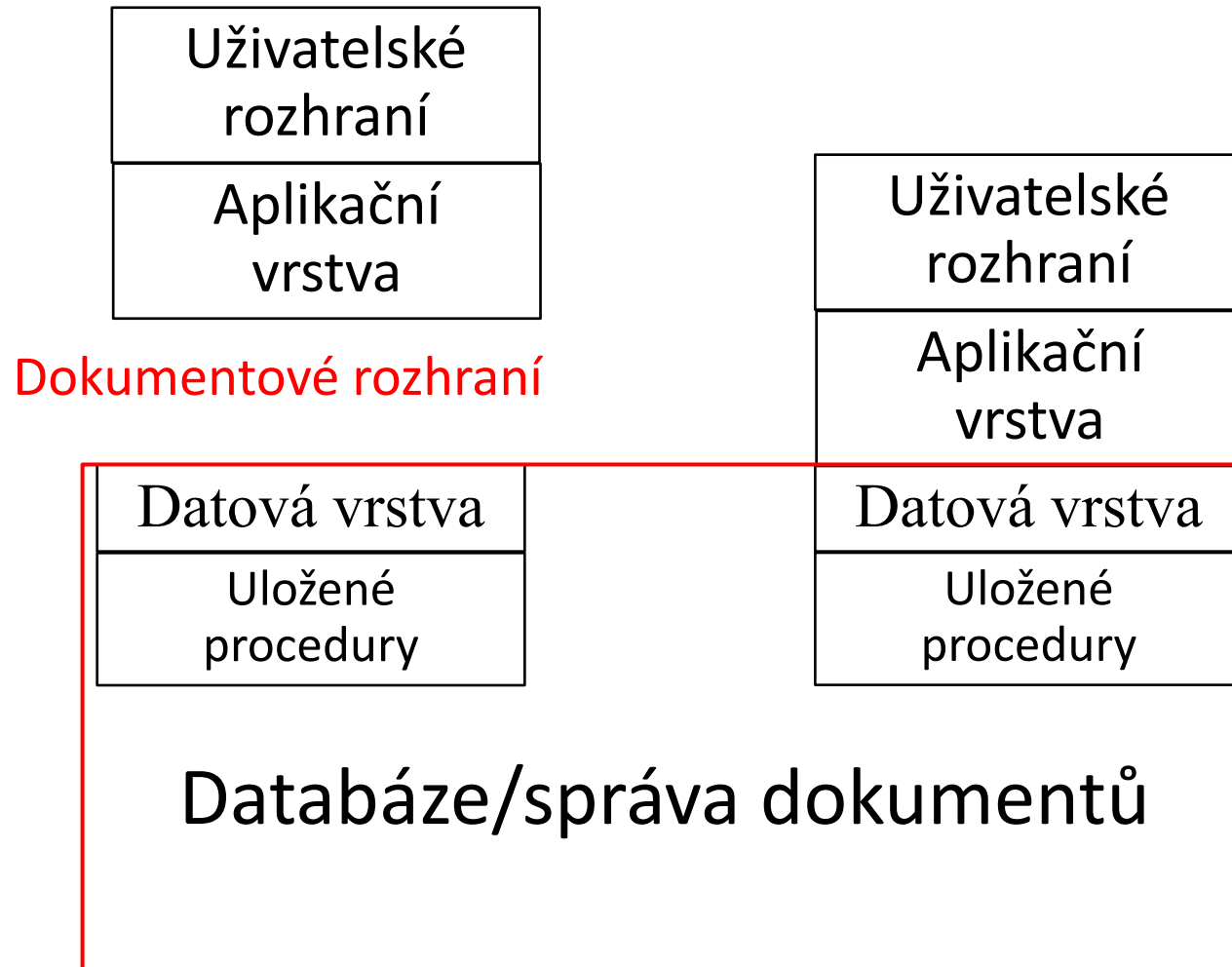
# Databázově orientované systémy

- Aplikace pracující nad stejnou (distribuovanou) DB
- Aplikační vrstva zčásti pomocí uložených procedur nebo wrapper dat
- Nutná disciplína při vývoji, lze pak vytvořit systém, který se do značné míry obejde bez explicitního používání middlewaru (ten je zakryt branami na datovou vrstvu)

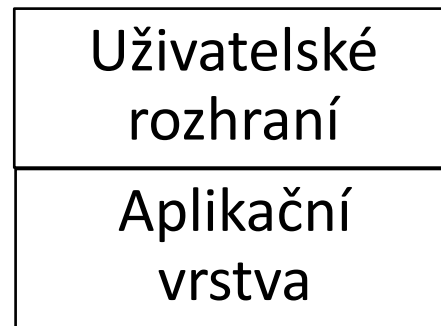
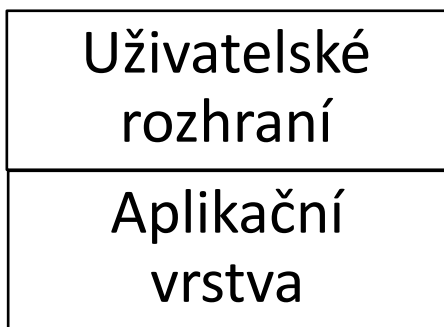
# Systemy propojené přes databázi



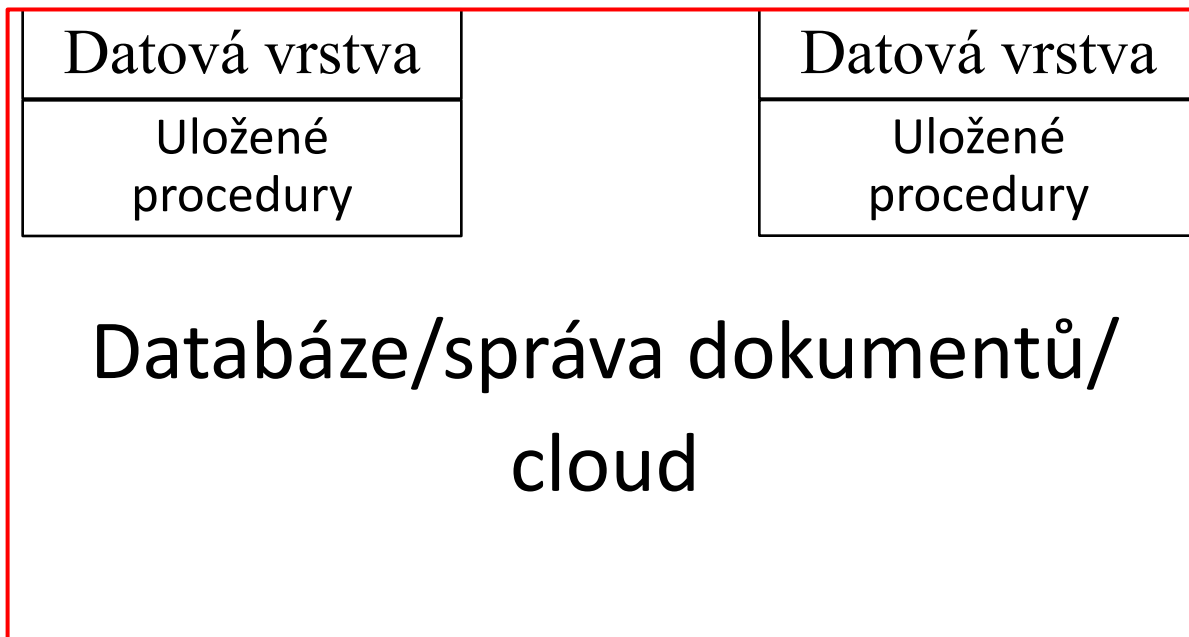
# Systemy propojené přes databázi



# Systemy propojené přes databázi



## Dokumentové rozhraní



# Rozhraní pomocí byznys dokumentů

- Uživatelsky a technologicky výhodné
- Umožňuje hladké připojení systémů různých výrobců SW,
  - i přímo malých s velkými pomocí zpráv
  - I malých s velkými pomocí dokumentů zapouzdřující cloud nebo datové baze



## **Výhody byznys dokumentů z pohledu (koncového) uživatele.**

1. Využití dekompozice a uživatelských jazyků a prověřených řešení
2. Transparentnost chování systémů, lze snadno pochopit, co se od jednotlivých služeb (subsystémů) v daném okamžiku požaduje.
3. Snazší specifikace požadavků a snazší specifikace změn
4. Možnost zapojení uživatelů do vývoje a údržby
5. Možnost on-line provádění manažerských činností, především analytické práce. Snazší spolupráce managementu a obchodníků
6. Zkvalitnění byznys dat (včasnost, snadná analýza, spolehlivost a důvěryhodnost, logování a využívání informací o průběhu byznys procesů). Lze to využít pro zdokonalování znalostí o byznysu, řešení nečekaných problémů i během soudní procesů týkajících se byznysu
7. Flexibilní byznys procesy, které mohou být i globální
8. Možnost zapouzdření datové vrstvy jako specifické služby. Snadný outsourcing a insourcing

# Výhody byznys dokumentů z hlediska technického

1. Lokalizace změn v důsledku možností skrývání implementačních informací
2. Snadnost kombinace hotových i vyvíjených systémů, vlastních i od různých výrobců. Zvláště výhodné to je pro malé a střední SW firmy.
3. Použitelnost systémů správy dokumentů
4. Lze zapouzdřit i datovou (DMS a dokumentové rozhraní) i uživatelskou (klientskou) vrstvu (excel atp.)
5. Snazší využití cloudů a obecně datových vrstev autonomních SW entit
6. Možnost integrace pomocí webu, cloudů, datových vrstev a systémů řízení dokumentů
7. Integrace vývoje a údržby
8. Redukce nákladů a řešení problému permanentní zastaralosti.
9. Snadný sourcing, snadné připojování věcí od malých firem k systémům od velkých výrobců (to je výhoda i uživatelská)

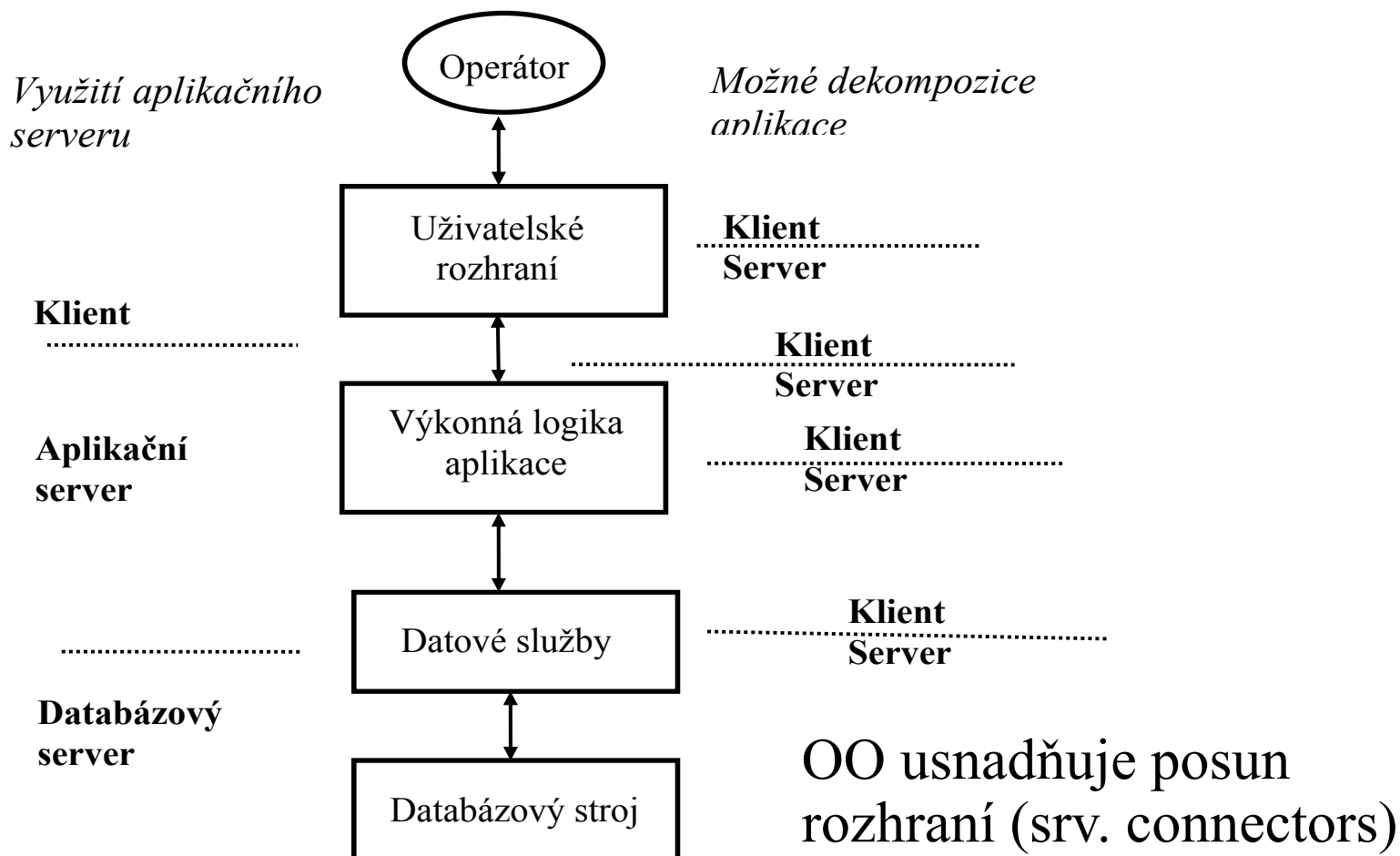
# Cloud a jiné triky

- Databáze nahrazena správou dokumentů
- Správa dokumentů skryta za konektory (v SOA implementované jako služby)
- Roli konektoru, který přebírá i část UI a část logiky, může sehrát tabulkový kalkulátor. Dokumenty jsou pak spreadsheetsy ve formátu XML.

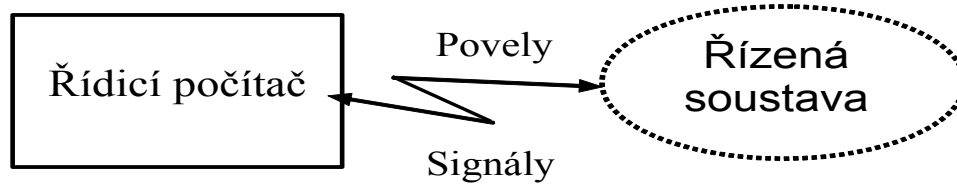
# Otázky dokumentové orientace

- Ne vždy je možná byznys orientace a hrubozrnnost
  - Regulační SW
  - Samotné hrubozrnné komponenty
  - Architekturní služby

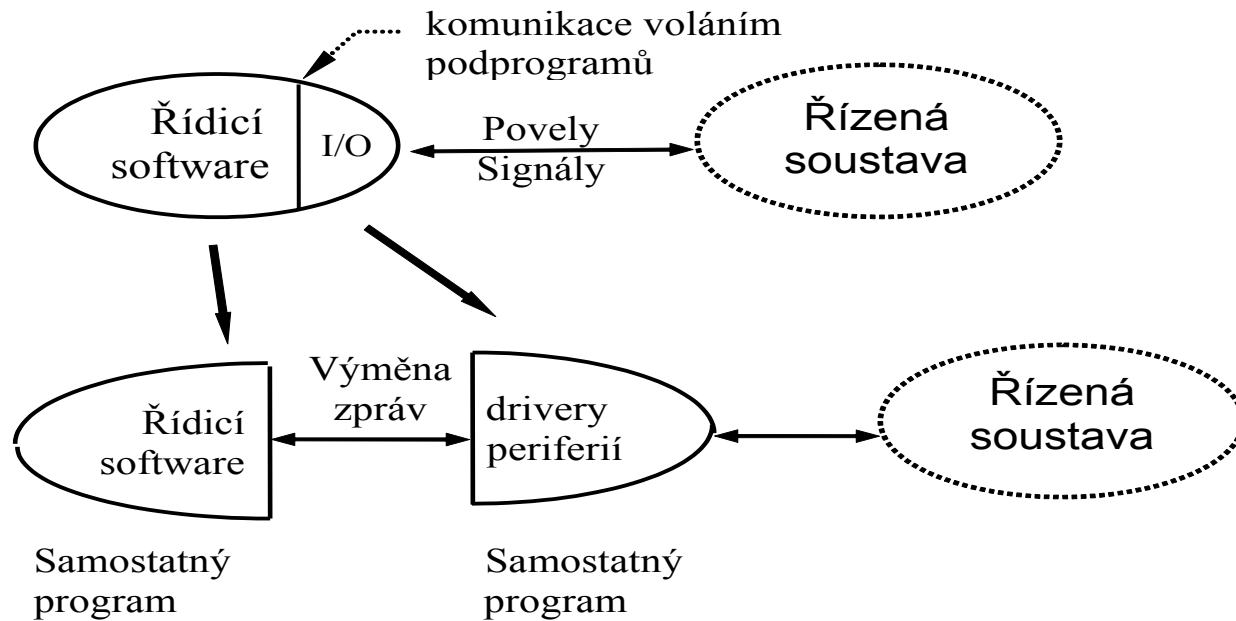
# Tři vrstvy a server



# K SOA

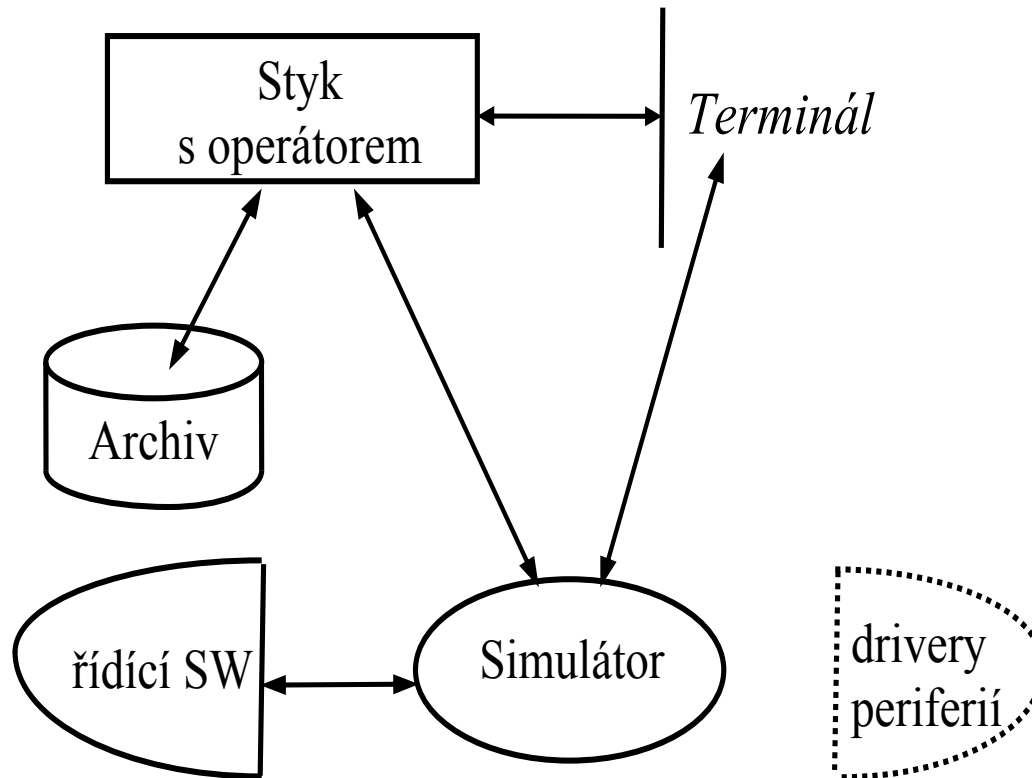


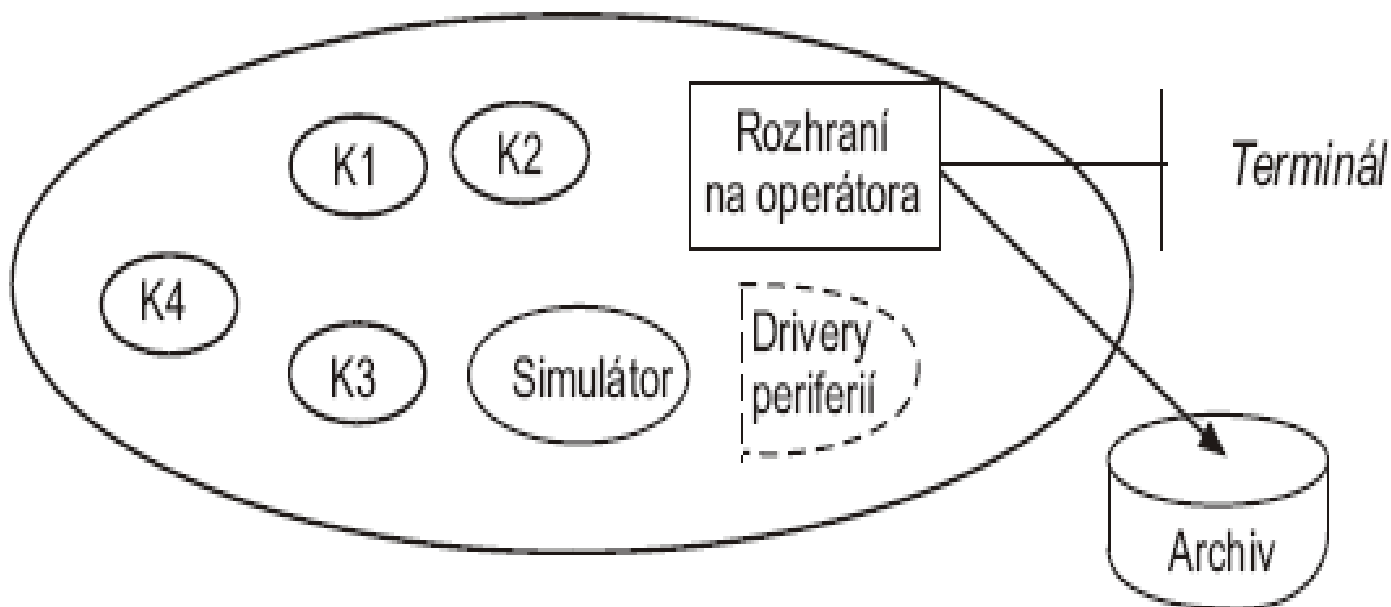
a) Soustava řízená počítačem



b) Oddělení ovladačů periférií realizujících styk s řízenou soustavou.

# K SOA, základní sestava

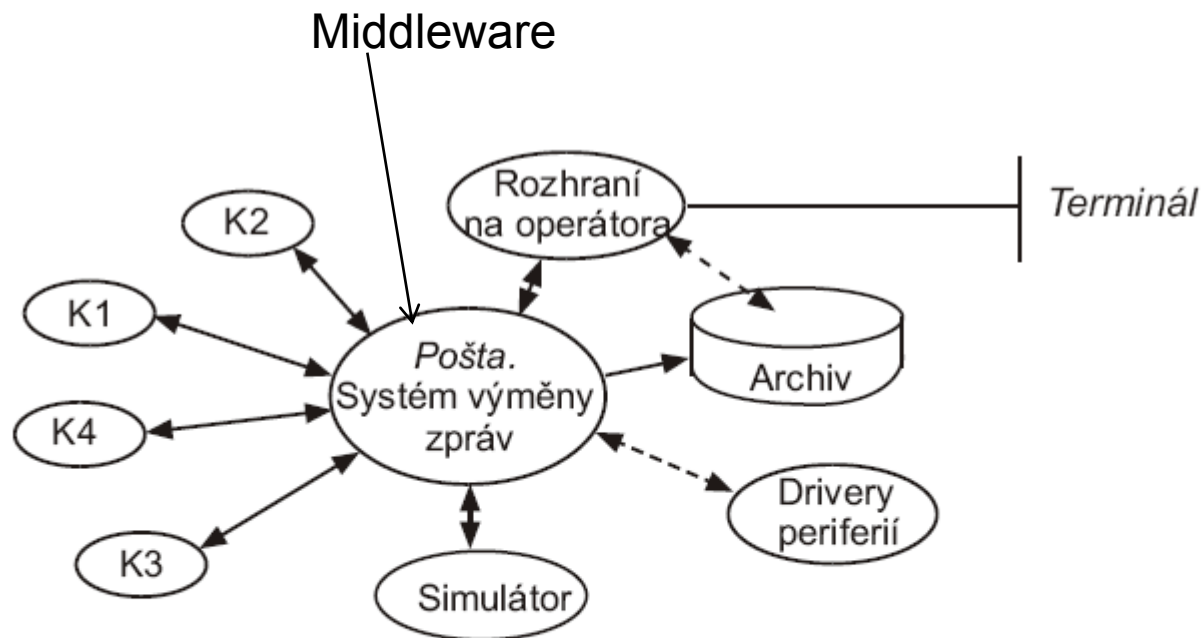




## **System po dekompozici..**

*Každá komponenta může komunikovat s libovolnou komponentou*





### **Dekomponovaný systém se subsystémem výměny zpráv.**

*Jednotlivé zprávy jsou posílány prostřednictvím specializovaného subsystému zajišujícího monitorování a přesměrovávání zpráv.*

Architekturní služby fungují jako rozšíření middleware, architekturu jako službu, agilní vývoj a agilní byznys procesy

# Hlavní výhody SOA

- Znovupoužití existujících a cizích aplikací
- Autonomní vývoj částí
- **Inkrementální vývoj**
- Modifikovatelnost a udržitelnost
- Umožnění principů agilního vývoje ve velkých systémech
- *Cesta k softwaru jako high tech*

# Některé nevýhody SOA

- Sekvenční zpracování je nutné zajišťovat
  - Řešení: Odpovím určené službě
  - Mohu čekat na odpověď
  - Identifikátor zprávy na kterou se odpovídá a nebo vratný parametr (identifikuje odkud pokračovat)
- Nejasné jak efektivně spolupracovat se SOAP a obecně jak optimálně aplikovat to nejlepší z objektové orientace
- Obtížné přijetí filosofie SOA

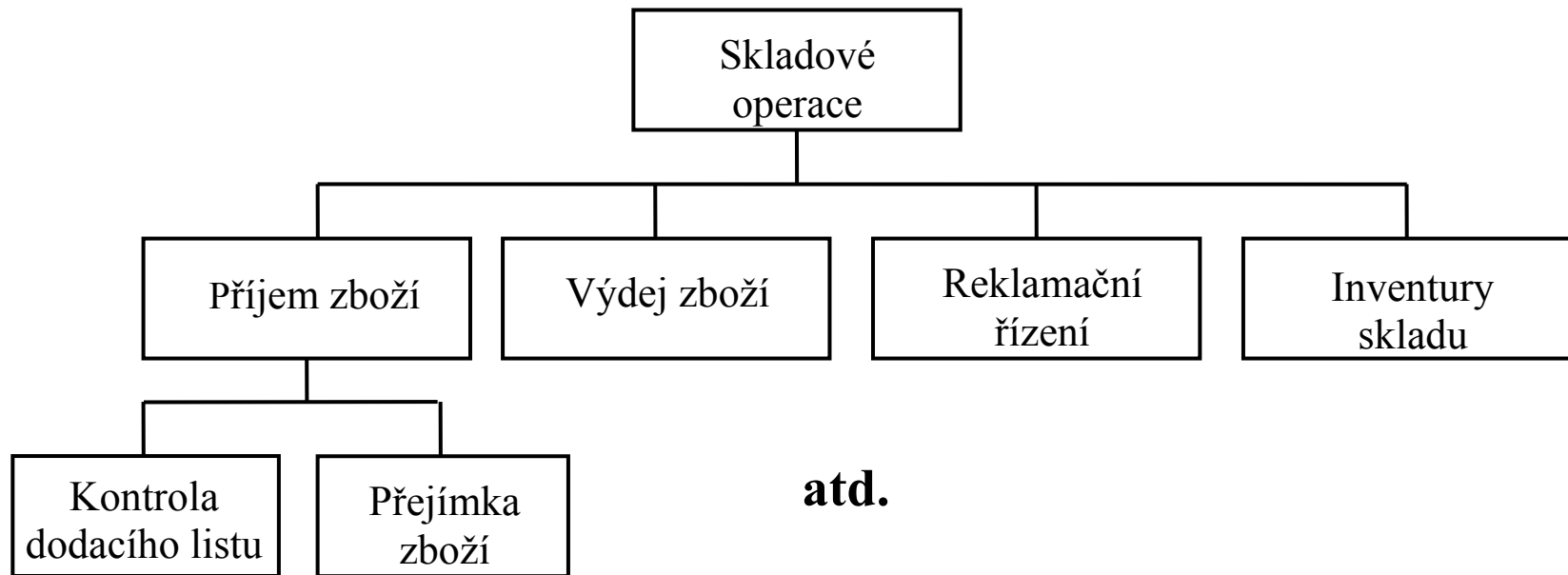
# Jaksonova metoda

- Je vhodná pro sekvenční dávkové zpracování uspořádaných souborů
- Základní princip:
  - Logiku mnohých programů lze odvodit z toho, jak se zpracovává soubor
    - Akce na začátku nebo ří změně klíče (pohyby na daném účtu)
    - Varianty zpracování vět
    - Akce na konci seznamu pro daný klíč
  - Lze opakovat při změně hodnoty klíče

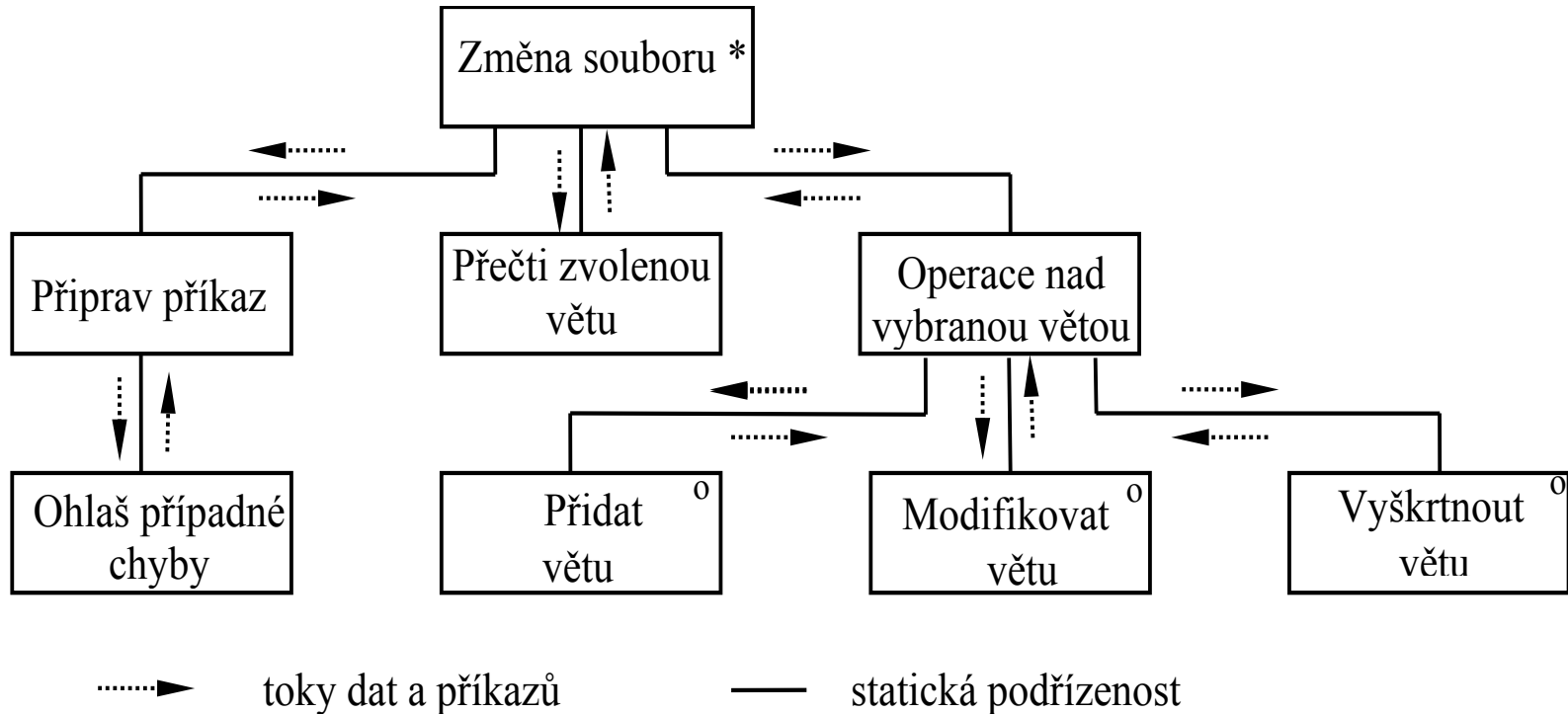
# Jaksonova metoda

- Je vhodná pro vývoj jednotlivých procesů v diagramu toků dat
- V jistém smyslu obdoba objektové orientace pro dávkové systémy

# Jaksonova metoda

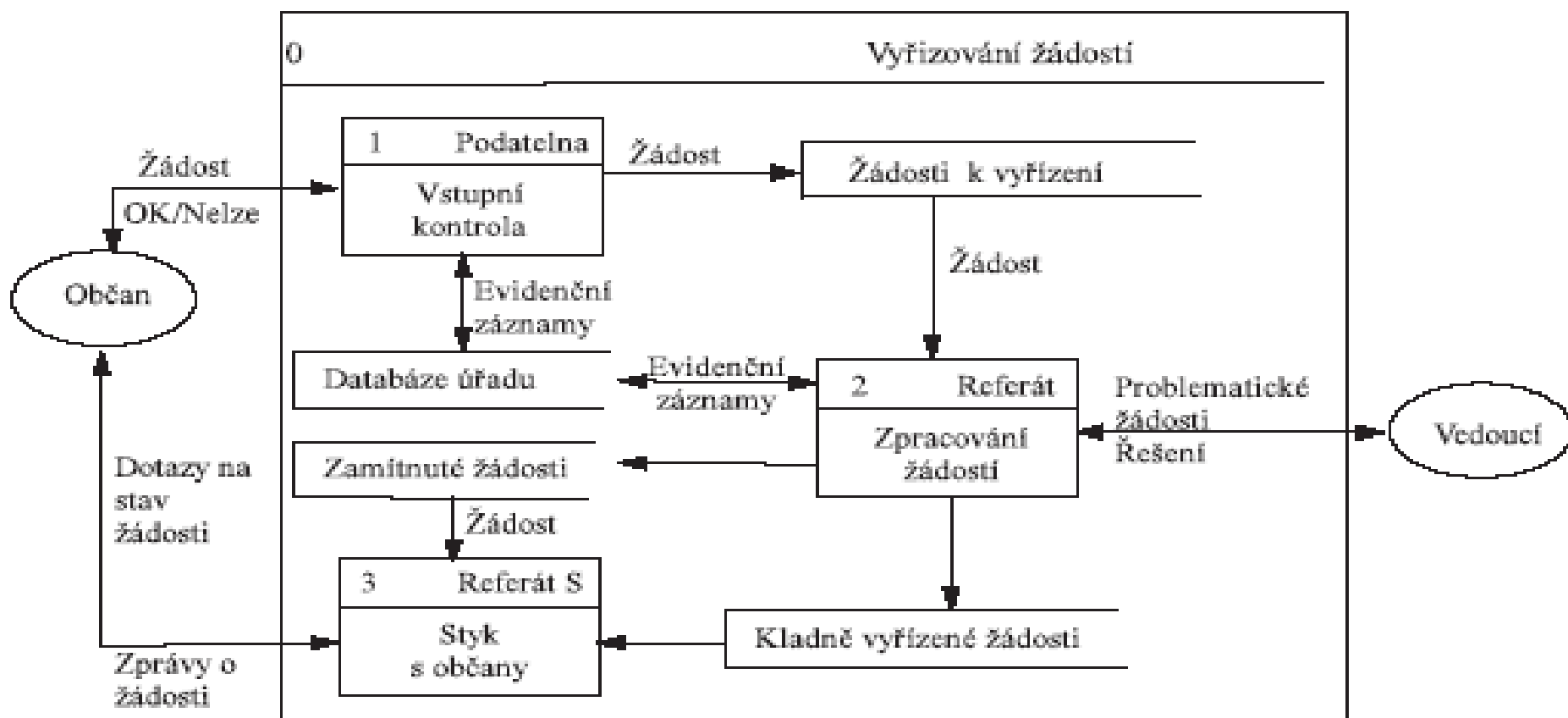


# Jaksonova metoda



Často lze strukturu programu odvodit z dekompozice činností, vhodné pro dávkové zpracování

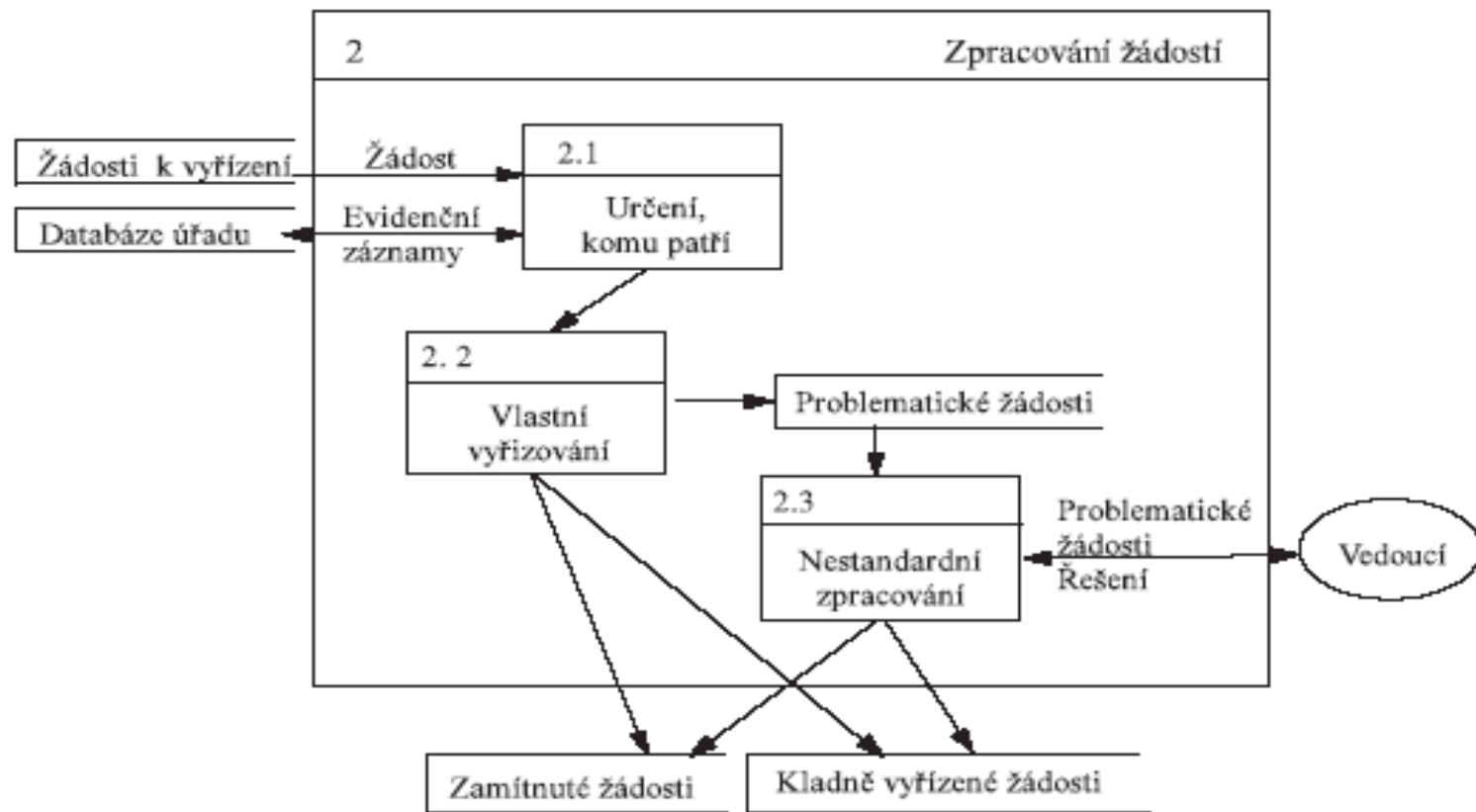
# Dataflow



Obr. 12.9: Diagram toků dat systému Vyřizování žádosti.



# Dataflow, funkcionální dekompozice



Obr. 12.10: Dekompozice procesu Zpracování žádosti. Kontext diagramu musí odpovídat kontextu procesu Zpracování žádosti v diagramu Vyřizování žádosti.

# Prvky DFD v SOA a cloudu

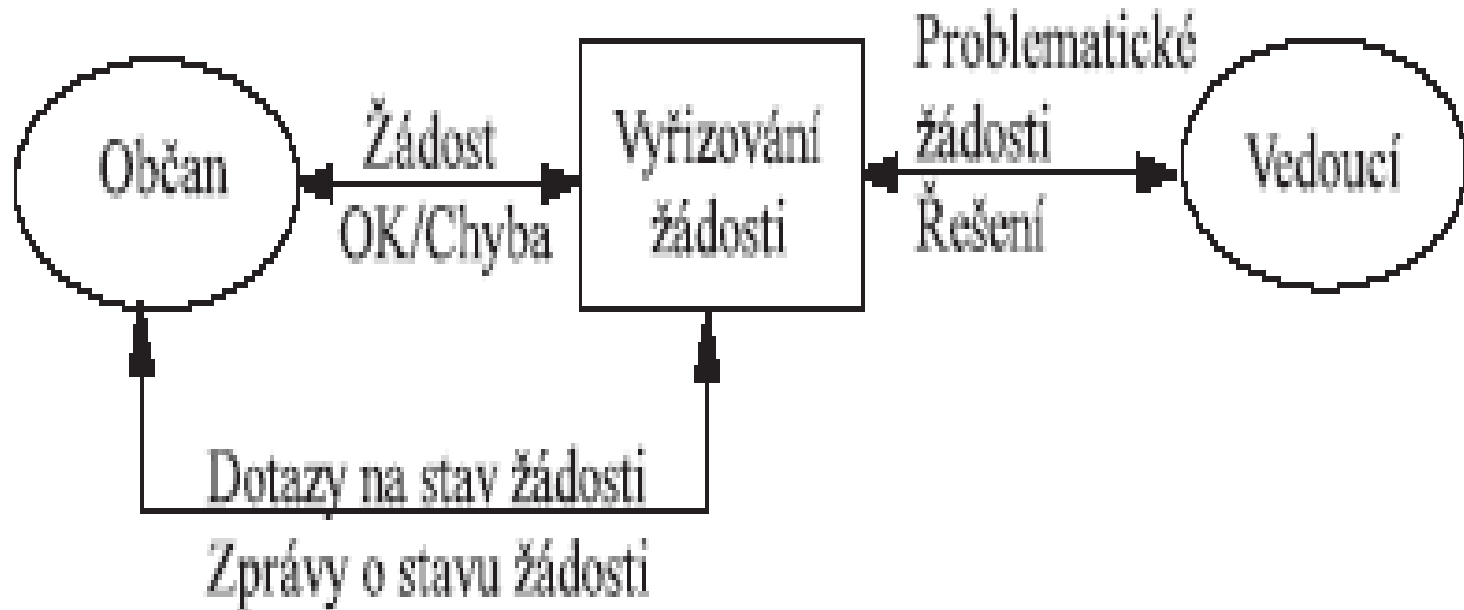
- Datové úložiště se zapouzdří jako služba
- Rozhraní dávkoé (bulk) i interaktivní)

# Odvozená hierarchická dekompozice



Obr. 12.11: Hierarchie vytvořená postupnou dekompozicí systému Zpracování žádostí.

# Diagram kontextu. Dají se použít Use Case



Obr. 12.12: Diagram kontextu systému Vyřizování žádosti.

# Výhody a nevýhody

- Vhodnější pro dávkové zpracování, tam ale významem obdoba SOA pro interaktivní spolupráci
- Pokud je možné použít, může podstatně usnadnit vývoj a modifikace využitím úložišť
- Nevýhoda je, že může omezit použití on-line operací
- Je možná kombinace úložišť a komunikace výměnou zpráv, to je zvláště vhodné pro manažery, viz Generalized Petri Places

# Rozhodovací tabulky

- Umožňuje přehledně zapsat, za jakých podmínek učinit příslušnou akci/akce
- Do horního pole se zapisují požadované pravdivostní hodnoty jednotlivých podmínek (ano A, ne N, na podmínce nezáleží X)
- Do dolního pole se zapisuje značkou x, zda se má příslušná akce pro kombinaci podmínek uvedenou v horní části sloupce provést

# Rozhodovací tabulky

Starý zákazník	A	A	A	A	A	...	N	N
Běžný leasing	A	A	A	A	N		x	x
Rušení nájmu	A	A	N	N	A		x	x
Nový nájem	A	N	A	N	A		A	N
Zařadit zákazníka							x	x
Test platby	x	x	x	x	x			
Zrušení smlouvy	x	x						
Nová smlouva	x		x		x		x	
Faktura	x	x	x	x	x			
Úprava smlouvy	x	x	x	x				

# Rozhodovací tabulky

- Vhodné spíše pro dávky a menší úlohy
- Osvědčuje se pro vyjasnění všech možností
- Podmínek nesmí být příliš mnoho
- Spíše jen okrajová metoda
- Použitelné při specifikaci požadavků i při návrhu systému
- Používá se v podnicích