**Úvod do datového inženýrství**

**Úvod:**

**Data use-cases** – pochopení zákazníku, předpovídat a navrhovat, co si budou chtít koupit, informace o výrobcích, řízení zásob, banky, zdravotnictví – data je potřeba dostatečně chránit, státní správa

**Business**

* BPMN = Business Process Model Notation - proces, který poskytuje podniku určitou hodnotu
* Během tohoto podnik generuje data (preference uživatelů, doba pečení, použité zdroje…). Data se mohou ztratit, mohou být uložena na papíře nebo mohou být spravována softwarem. Pro mnohé jsou data klíčovou složkou podniku. Naším cílem je pomoci podnikům se správou dat.

**Data** **management**

* Sběr dat: sensory, skenery, chytrá zařízení, sociální media, strukturovaná data, text, logy, …
* Organizace dat: kde a jak ukládat data, Ochrana dat: osobní údaje, průmyslová špionáž
* Ověřování dat: kontrola očekávání a kvality, Zpracování dat
* Destrukce dat: archivace, Certified data erasure of SSD, HDD and mobile (certifikované mazání dat)
* Data warehouse (datový sklad): nástroj pro shromažďování různorodých zdrojů dat na jednom centrálním místě za účelem podpory business analýzy a reportů
* Business inteligence (BI) management: schopnost učit se/pochopit/řešit nové/zkušební situace

**Komplexní datové typy:**

* Konceptuální datový model (doménové modely): celkový pohled na obsah systému, jak bude organizován a na zahrnutá obchodní pravidla. Entity bez podrobností a jejich vztahy.
* Logický datový model: přidáme atributy, cizí klíče, klíčové atributy... databázově agnostický
* Fyzický datový model: přidáváme datové typy nebo uložené procedury, potřebné pro konkrétní databázi. Názvy jsou kompatibilní s databázemi.

**Úvod k data warehouse, Data cube**

* Máme velké množství dat, každý v modelu musí mít přístup k datům, které potřebuje

**Operační databázový systém**

* Ukládají se sem všechna potřebná data
* Podpora každodenního provozu, Normalizované schéma
* Transakčně orientovaná – Konstantně vytížená malými zápisy
* Vysoká dostupnost – nemůže padat, vysoký výkon
* Systémy pro zpracování transakcí on-line (OLTP)

**DIKW (pyramida)**

* Data (nuly a jedničky) -> informace (data + kontext, interpretace, klasifikace, např. trend) -> znalosti (důvody a příčiny) -> moudrost (co s tím budu dělat, změním nabídku, dodavatele)

**Klasifikace dat**

* **Běh business dat** – pro efektivní provozování businessu – platby, markování, …
* **Integrace business dat** – integrace/synchronizace businessu mezi aplikacemi, např. jakým způsobem propagovat změnu přes jednotlivá oddělení (změna adresy, časová pásma)
  + Master data = data vložíme jen najedno místo v organizaci
  + Cílem je zlepšit efektivitu podnikání integrací/synchronizací aplikací
* Sledování business dat - např. aby doručovatel pizzy zvolil nejkratší cestu, Nutná vyšší kvalita dat – data pro rozhodování

**Data asset (datové aktivum)**

* = jakákoli data, která mají hodnotu nebo jsou užitečná pro organizaci nebo jednotlivce
* mnoho podob, např. údaje o zákaznících, o zaměstnancích, finanční údaje, průzkum trhu, údaje o zásobách
* Hodnota datových aktiv je dána kvalitou dat, spočívá ve schopnosti využít je k podpoře rozhodování, poskytování informací a pomoci dosáhnout obchodních cílů.

**Business inteligence (BI)**

* Typ analýzy dat zaměřený na pochopení organizačních činností a příležitostí
* Sada nástrojů, které tento druh analýzy dat podporují, např. pro vytvoření grafů, statistik, … tak aby data byla srozumitelná i pro lidi mimo IT

**Kategorie BI**

* Reporty a dotazování – co se stalo
  + Zásadou reportu je zodpovědět předem podanou otázku
  + Forma prezentace statická (třeba pdf, prezentace)
  + Vrací informaci, kterou neumíme dál zkoumat
  + Dotazování – několik klíčových identifikátorů na které se ptám ("několik hlavních kontrolek, které mi řeknou, jak na tom jsem")
* Business analýza – co se stalo a proč
  + Větší interakce, uživatel s nimi musí umět pracovat
  + vizualizace dat ve více dimenzích
  + Obvykle se prezentuje jako tabulka s dimenzemi namapovanými na sloupce a řádky
  + "sql tabulka"
* Data mining - Dobývání znalostí
  + Co by se stalo kdyby, nebo čím to můžeme zlepšit
  + Potřebujeme více času a vyškolený personál
  + "používáme speciální jazyk"

**Jiné rozdělení**

* Provozní reporting – Krátkodobé a dlouhodobé (meziroční) trendy a vzorce
  + Podpora krátkodobých obchodních rozhodnutí
* Business performance management (BPM) - řízení výkonosti podniku
  + Metriky sladěné s cíli organizace
  + Podpora dlouhodobých obchodních rozhodnutí, výkonná úroveň
  + Optimalizovat provádění obchodní strategie
* Provozní analytika (Analytic Applications)
  + Využití analytiky téměř v reálném čase k řízení provozních rozhodnutí
  + Business Intelligence pro přední linie podnikání

**Business perspektiva**

* Informační technologie
  + Nerealistická očekávání pro nové aplikace nebo vylepšení stávajících aplikací
  + Malý zájem o design a testování – z hlediska businessu není vždy testování možné
  + Nedostatek školení
* Business - IT nemá žádné tušení o businessu, pomalá doba odezvy, náhodné funkce, nefunguje to

**OLTP (On-line transaction processing systems) vs. OLAP (Online Analytical Processing)**

* OLTP musí fungovat furt, tisíce zápisů denně
  + Online relační databáze
* OLAP = Online analytical processing
  + Podpora rozhodování
  + Stovky uživatelů, read only
  + Multidimenzionální analytické dotazy
  + Vysoká flexibilita, pravidelné aktualizace
  + Nemusí běžet nonstop (např. v noci, pokud jedem jen v jedno časovém pásmu)
  + Výsledek dotazu je nějaká matice hodnot, data cube (multidimenzionální)

**Multidimensionální data**

* Dimenze jsou hlediska nebo entity, o kterých chce organizace vést záznamy
* Fakta jsou číselné hodnoty nebo pozorování pro dané dimenze.
* Dimenze mohou být hierarchické – široký sloupec pizzy postupně dělím na jednotlivé typy pizz
* Míry, vypočtené hodnoty podél vybraných dimenzí: agregace, jako je součet, min, max

**Data cube**

* Hierarchická dimenze, dodatečné dimenze
* Operace: pivot (změní uspořádání vizualizace, např. přesune dimenzi ze sloupce do řádků), vnoření (více dimenzí na sloupce nebo řádky)
* Rdf vocabulary – nejdřív musím definovat dané objekty (dimension and measure property)

**Datové sklady**

**Přístup k datům**

* Máme mnoho různých zdrojů dat a pro Business inteligenci potřebujeme, aby dat byla konzistentní
* Řízení dotazů
  + Wrappers / Integrátory / Mediátory / …, Slovník metadat, Distribuované dotazy
* Řízení aktualizací
  + Integrace do jediného úložiště dat (starší způsob, zbavujeme se problému s distribuovanými daty), Cloud / Streaming / ... (proto se používají data warehouse)

**Data engineering**

* budování systému pro sběr, zpracování a poskytování dat
* máme různé zdroje dat: data lake, data warehouse, use-case specific
  + data warehouse – business inteligence vyžaduje report a OLAP
  + data lake, vlastní pipelines – datový analytik potřebuje přístup k sql databázi, data scientist může potřebovat surová data nebo vlastní datové toky
  + big data – potřebujeme specializované řešení, ať už pro zpracování nebo pro feeding modelu machine learningu. Pravděpodobně streamované a distribuované zpracování

**Proces vytváření projektu**

* Requirements – sběr požadavků, "co opravu chtějí"
* Design – implementace
* Development (vývoj) a testování – jaké použijeme technologie, jaká data, …
* Deployment – nasazení do provozu

**Data Source 
Data Source 
Extract 
Staging Area 
Transform 
Data 
Warehouse 
Load Datový sklad**

* Místo pro ukládání data assets, dobře organizované, přístupné, cíleně vedené, „není to datová skládka“
* začínáme s datovými zdroji, které chceme dostat na jedno místo
  + Nejprve je dostaneme do "staging area", kde se provádí transformace dat (z csv json, …)
  + Poté je převedeme do datového skladu
* Proces získávání dat ze zdroje do Datového skladu se nazývá Extract-Transform-Load.
* Hlavní účel: podpora rozhodování, abychom toho mohli dosáhnout, musíme se spojit s businessem
* Komponenty dělíme do dvou kategorií: na databázi (zde Data Warehouse) a veškerý podpůrný software

**Architektura datového skladu**

* 1. Zdrojová data - externí, řešíme data movement a kvalitu dat
* 2. cílová data (bottom) (staging struktury, data warehouse master data, …) – data, servery, správa a monitorování dat, někdy je nad ní middle tier s OLAP servery, ty poskytují OLAP funkcionalitu pro top tier
* 3. uživatelský přístup (tabulky, reporty, dotazy)

**Klasifikace datových skladů**

* Lite – chci realizovat požadavek s minimálními náklady
  + Často pouze restrukturalizujeme data z jednoho zdroje dat, často lze považovat za data mart
  + Rozsah omezen na 1–2 tematické oblasti, stačí použít relační databázi s vhodným schématem
  + Zaměření na: „řekni mi, co se stalo“
* Deluxe - větší množství dat, použití BI, založeno na user-case
  + Od jednoduchého reportingu, obchodní analýzy, dashboardů, data miningu, statistické analýzy
  + Různé zdroje dat, velikosti v terabajtech
  + V blízkosti zdrojů mohou být transformační místa, data lze publikovat do data martu
* Supreme - jedná se o distribuované prostředí
  + Datový sklad je řešen přes streamování
  + Neomezený, přístup ke zdrojům na internetu, konzumují nestrukturovaná data, video, audio
  + Externí zdroje, zaměřeno na data mining – intelligent assistant (AI)
  + Distribuované, s katalogem metadat uživatel nemusí vědět, kde přesně se data nacházejí.

**Data mart**

* Data warehouse lite, může být pouze zdroj pro data warehouse
* Limitované scope (rozsah) – geograficky, organizačně (např. na jedno oddělení), funkčně omezený
* Vrací automaticky odpovědi na otázky „jak se nám daří“ – pomocí předem definovaných kritérií
* Top-down implementace
  + Neexistuje žádný data warehouse
  + Rychlá implementace, nízká cena, nízký risk
* Bottom-up - Vize distribuovaného data warehousu

**Jak vytvořit data warehouse?**

* Top-down (shora dolů)
  + Obtížná realizace, model náročný na návrh a údržbu, když začneme stavět, už nemůžeme nic měnit
* Bottom-up
  + Integrace menších projektů
  + Business otázky motivované BI: kolik máme ziskových zákazníků? Které tržní scénáře pro nás představují nevětší úspěch? …
* Držet si guide-line
  + Jasné cílé, jasně definované scope a procesy
  + Datová integrace, Vytvářet metadata, starat se o zabezpečení dat
  + Spolupracovat s jinými institucemi – data quality, data goverment
* Změnit business proces!!! Vysvětlit lidem, jak to funguje, proč, aby to používali

**Governance**

* Řeší, kdo je zodpovědný za jednotlivé úkony, kdo má daný problém řešit
* Konceptuální data modely, zpětná vazba o kvalitě dat
* Koncová metadata – co tato zpráva znamená, jak byla vytvořena, …
* Linie dat (odkud data jsou, trasování dat)
* Service level agreements – jak často a kdy nebude data warehouse fungovat

**Implementační detaily data warehousu**

* Rozdělení dat:Časové období, Geografie, Organizační jednotka
* Typy dat: Údaje v reálném čase, Odvozená data
* Databáze**:** Relační databáze (Query optimalizace, Transakční -> přišel OLAP a bylo to k ničemu), Online analytical processing (Multi-dimenzionální analytické dotazy, XML pro analýzy (XMLA)), Relační online analytical processing (Start schema), Multidimenzionální online analytical processing (Multidimenzionální databáze), Hybridní přístup (Rozdělení dat)

**(Multi)dimezionální modelování**

* technika pro konceptualizaci a vizualizaci datových modelů jako souboru mír, které jsou popsány společnými aspekty businessu – star model, snowflake model, galaxy model

**Alternativy datových skladů**

**Data lake**

* 2.generace – vynecháme transformaci
* v případě datových skladů mám velké množství zdrojů a pipeline – když se něco pokazí, je problém najít chybu
* Zde dáme data rovnou na jedno místo, netransformujeme a nepředzpracováváme, jednoduše se dělají změny
* U datového skladu vím, co který sloupeček znamená, jaký má vztah k ostatním – scheme on write
* Ale u data lake vytváříme schéma až při čtení – uživatel musí mít větší technické dovednosti, vyznat se v datech
* Nemají žádnou strukturu, např. pouze adresářová struktura -> pracnější pozdější využití
* Use-case: machine-learning, deep analýzy
* Plusy: rychlé, jednoduchý přístup k datům, pro velké množství dat, snadná změna
* Mínusy: žádná vize – nevíme, proč to vlastně děláme, data governance - ve společnosti musí být jasně nastavené procesy a přístupy, kdo má za co zodpovědnost - u data lake tohle nejde jasně kontrolovat
* Data swamp (bažina) – když se data lake zvrtne

**Data mesh (datová síť)**

* 3.generace– chyba není v transformaci, ale v ukládání všeho na jedno místo, centralizované úložiště je zdrojem problémů -> přesouváme se k distribuovaným úložištím
* Micro-service pro data, Rychle odpovídá na změny
* Na business úrovni existuje mapování na zdrojové údaje. Musí mít vhodnou infrastrukturu pro zpracování dat.
* Principy:
  + Doménově orientované decentralizované vlastnictví dat a architektura
  + Máme vlastní tým, který řeší infrastrukturu, která je všude stejná
  + Data jsou ten produkt

**Data jako produkt**

* Dají se najít – data katalog, Dostupné – jak často systém padá, Důvěryhodný a pravdivý, Sémantika a syntaxe
* Interoperabilita – měli bychom mít schémata, …, Kontrola přístupu

**Data (pre)processing**

**Data transformace**

* Data wrangling – příprava dat
* Zpravidla transformace dat z jednoho formátu do druhého, z jednoho schématu do druhého
* Hlavně pochopení zdrojových dat, schématu, formátu a znalost knihoven
* Data selection – filtrování řádků v sloupcích: SELECT \* FROM customer WHERE city = 'Prague’;
* Data projection – filtrování sloupců v řádcích: SELECT city, state, zipcode FROM customer;
* Data sumarizace – velký obrázek, vizualizace – grafy, …, metriky – sum, count, min, max, mean, …
* Redukce dat – data cube agregace, redukce dimenzí, výběr pouze některých atributů, regrese, histogramy, sampling (získávání reprezentativních vzorků),
* Data cleaning – základem je kvalita dat
* Integrace dat
  + Identifikace entit - chceme identifikovat stejné objekty – 2 instance stejného uživatele spojit do jednoho
  + Zbavujeme se redundancí, integrace schámat
  + Rozpoznávání pojmenovaných entit - identifikace čehokoliv z datového skladu vůči něčemu
* Data Lowering / lifting (zvyšování/snižování významu) - Z pdf -> xls -> csv -> rdf -> lod
  + Technický viewpoint – pouze o formátu, ale sémanticky to nedává smysl
  + Semantický viewpoint – přidávám tomu smysl, toto je entita zákazníka atd.

**Kroky transformace dat**

* Data cleaning – redukovat šum, odstranění nesrovnalostí …
* Data integration – spojení více zdrojů dat
* Data transformation – normalizace, redukce, formát
* Atributy procesu: automatizace, výkonnost, škálovatelnost, robustnost, provenance (původ)

**Diskretizace a generování hierarchie pojmů**

* Hierarchie dat umožňují snadno provádět sumarizaci a vidět data z různých perspektiv / konceptuálních úrovní
* Nejsou vždy dány, řešením je jejich vytvoření z dat
* Pro potřeby business analýzy, nástroje pro data mining
* Generování diskretizace: Členění, diskretizace založená na entropii, intuitivní rozložení, shlukování (clustering)
* Generování hierarchie pojmů: manuální definice, částečné uspořádání atributů, soubor atributů, sémantika

**Alternativy ETL**

* Extract Transform Load
  + Pull přístup, pracuje na vyžádání nebo podle rozvrhu
  + Pipeline / workflow / proces
  + Komponenty / pluginy
  + +: vizualizace toku dat, paralelizace, monitoring, failover features, přehlednost, rychlost, …
* Extract Load Transform – pro data lake
* Enterprise Service Bus – Push model, získává data podle potřeby

**Kvalita dat a provenance**

**Proč?**

* Špatná kvalita dat nám bere více času, peněz, může naštvat zákazníky, vytváří další navazující chyby
* Provozní level: správně psaná jména, adresy, Taktická: čím lepší data, tím lepší rozhodnutí, Strategická

**Kvalita dat**

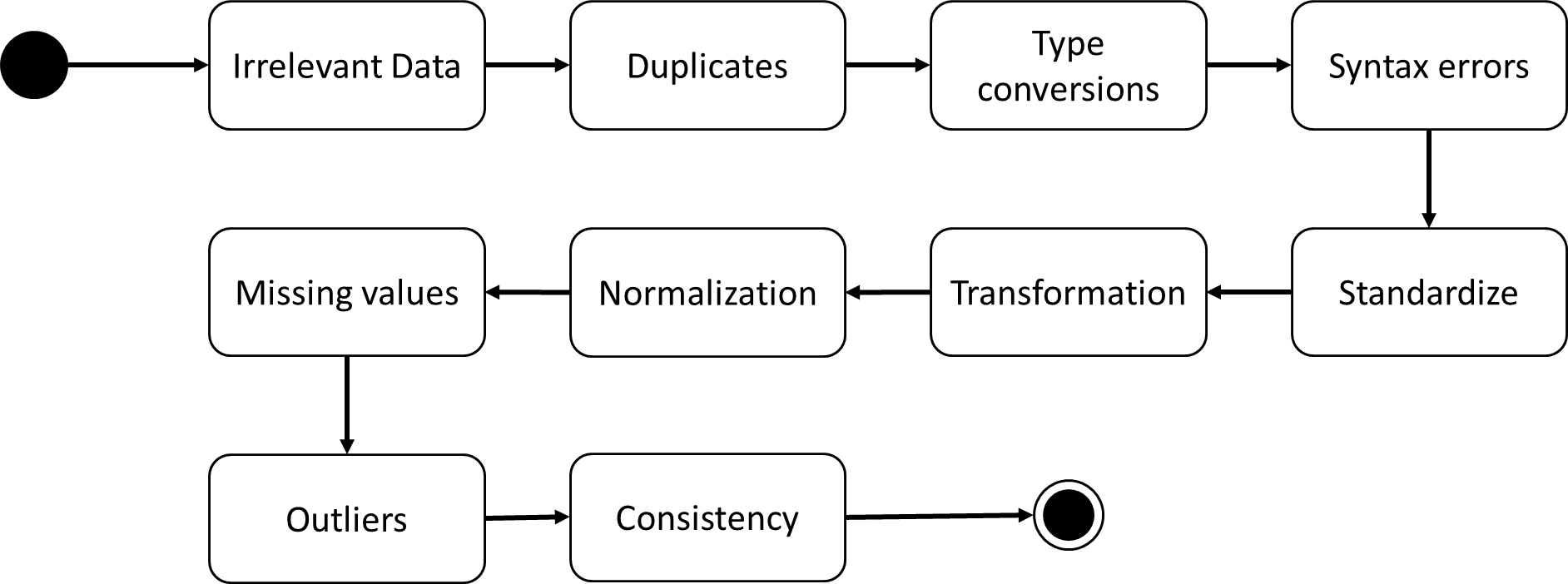
* Kvalita dat odpovídá jejich schopnosti uspokojit požadavky na jejich použití, neexistuje přesná definice kvality
* Abychom mohli data zlepšit, musíme umět změřit rozdíl
* Čištění dat – součást procesu kvality dat – odstranění odlehlých hodnot, chybějících hodnot, vyhlazení šumu
* Proces nikdy nekončí – měl by být součástí správy dat a jejich životního cyklu

**Hodnocení / inspekce**

* Musíme definovat kvalitu dat, k měření kvality dat můžeme použít dimenze kvality, alternativně inspekci = uživatel interaktivně kontroluje data namísto použití technik pevného měření
* Dimenze kvality:
  + Dostupnost – aspekty související s přístupem, autenticitou, vyhledávání dat
  + Vnitřní dimenze – nezávislé na kontextu uživatele, společné pro více uživatelů, např. úplnost, konzistence
  + Kontext – vysoce závislé na kontextu daného úkolu, jako relevance, aktuálnost
  + Reprezentační dimenze - aspekty související s návrhem dat, jako je interoperabilita.
* Inspekce
  + Manuální kontrola dat
  + 1. profilování dat – souhrnné statistiky, průměr, rozptyl, …
  + 2. vizualizace – rozšíření profilování -> vytváříme grafické prvky
  + Snaha najít vzory nebo hodnoty, které jsou neočekávané, a tedy chybné

**Dimenze**

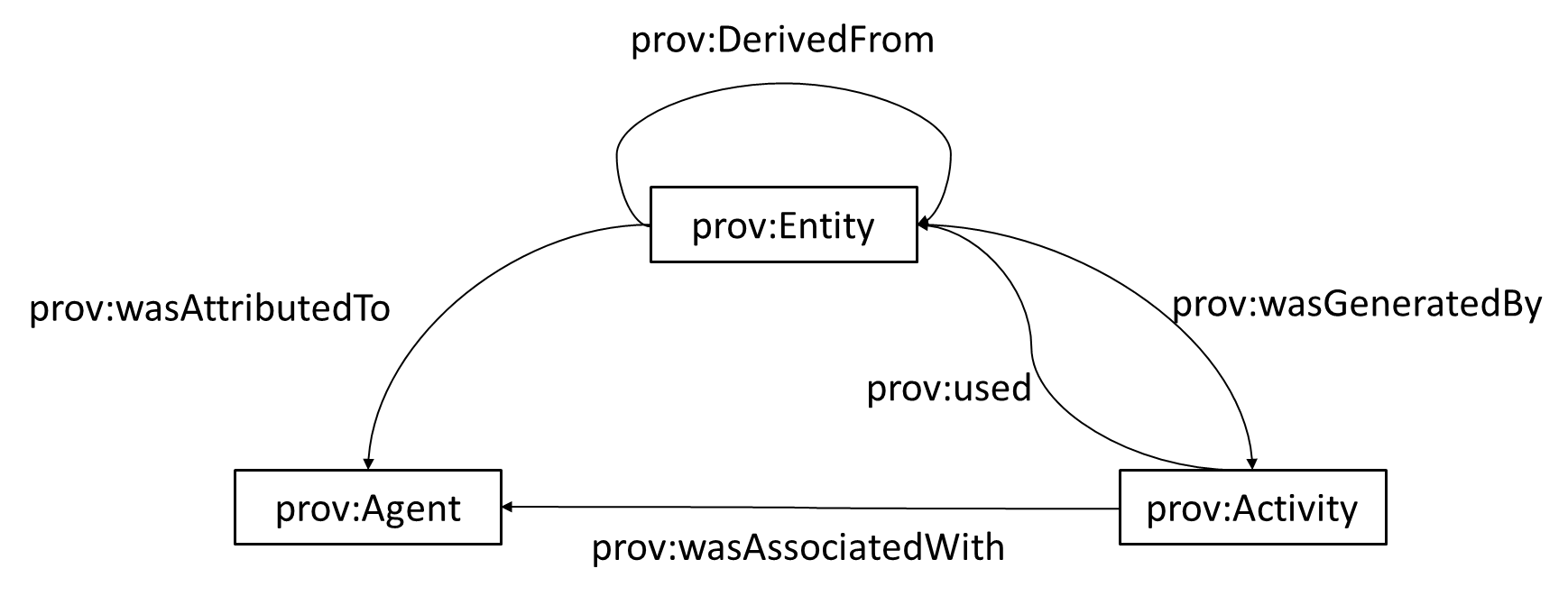
1. Přesnost – míra, do které jsou data správná, spolehlivá, certifikovaná, hodnoty dat uložené v databázi odpovídají reálným hodnotám – např. čísla na daný počet deset. míst, adresy – musím znát i ulici, nejen město
2. Úplnost - míra, do které jsou data dostatečně hluboká a rozsáhlá pro daný úkol – chybějící datové záznamy
3. Konzistence – neporušení sémantických pravidel definovaných nad sadou datových položek – negativní věk
4. Dostupnost - míra, do jaké jsou data (nebo jejich část) přítomna, dostupná a připravená k použití
5. Licencování - povolení spotřebiteli k opakovanému použití datové sady za stanovených podmínek – GDPR
6. Aktuálnost - jak aktuální jsou data ve vztahu ke konkrétnímu úkolu
7. Srozumitelnost - snadnost, s jakou lze data pochopit bez nejasností a použít je pro lidské konzumenty – null???
8. Vzájemné propojení - míra, do jaké jsou entity, které reprezentují stejný koncept, vzájemně propojeny, ať už v rámci dvou nebo více zdrojů dat

**Data cleaning**

* String – odstranění bílých znaků, new line, standardizace kódování, odstranění stop slov
* Datum a čas – formát, časová pásma
* Chybějící hodnoty
  + Vynechání řádků anebo sloupců s chybějícími hodnotami
  + Hledání vzorů v chybějících hodnotách
  + Doplnění chybějících hodnot: ručně, globální konstanta, N/A, Unknown, metody imputace
  + Metody imputace = proces odhadu chybějících údajů pozorování na základě platných hodnot jiných proměnných – např. k-nejbližší sousedé, na základě průměru
* Šum dat = náhodná chyba nebo rozptyl v měřené veličině
  + Odlehlé hodnoty: odstranění, mít extra skupinu / statistické metody
  + Techniky vyhlazování dat: Rozdělení do skupin (Binning), Shlukování, Regrese
* Nevalidní / nekonzistentní data
  + -> Znalost domény/businessu, Levenshteinova vzdálenost, Asociační pravidlo, Shlukování, ..., Odstranit

**Linie/Provenance dat**

* Level detailu: technická linie, business linie
* Účel: WHERE linie – z jakých vstupů pochází výstup, HOW linie – dáno výstupem, jak bylo se vstupy manipulováno, aby byl výstup vytvořen
* Cíl nástroje pro linii dat je správa životního cyklu dat od jejich vzniku až po zánik
* Cíl data provenance je konkrétně sledovat vznik dat a segregaci dat ve třech klíčových fázích - data-in-motion, data-in-process, a data-in-rest

**Data provenance – PROV**

* = informace o entitách, aktivitách a lidech, které se podíleli na vzniku dat nebo věcí, které lze použít k posouzení jejich kvality, spolehlivosti nebo důvěryhodnosti
* Rodina dokumentů PROV definuje model, odpovídající serializace a další podpůrné definice, které umožňují interoperabilní výměnu provenance v heterogenních prostředích, jako je web
* Provenance digitálních objektů reprezentuje jejich původ
* PROV je specifikace k vyjádření provenance záznamů, které obsahují popisy subjektů a činností podílejících se na jejich tvorbě nebo dodání

**Data management & FAIR**

**Data asset**

* Ekonomický zdroj, který můžeme vlastnit a kontrolovat, dává nám/produkuje nějakou hodnotu
* Může být převeden na peníze, hodnota je rozdíl mezi cenou a benefity – cena za uložení dat, za nahrazení dat pokud je ztratím, cena za zlepšení dat, benefity z vyšší kvality dat, jak data mohu prodat

**Data management**

* = vývoj, realizace a dohled nad plány, zásadami, programy a postupy, které zajišťují, kontrolují, chrání a zvyšují hodnotu dat a informačních aktiv v průběhu jejich životního cyklu
* Data management odborník: každý, kdo se zabývá jakýmkoliv aspektem správy dat
  + Business: data stewared, data strategists, chief data officers, data governance council
  + Technical: database administrátor, network administrátor, developeři

**Data management frameworks**

* Pokud chci dělat nějakou věc, toto jsou kroky, na které bych neměla zapomenou, způsoby, jak postupovat, …
* Strategic alignment model
* Amsterdam Information Model
* Hexagon, Context diagram

**Aikenova pyramida**

* Fáze 1: Společnost, která se rozhodla pořídit si databázi
* Fáze 2: Začínáme řešit kvalitu data
* Fáze 3: vytváříme management, vytváříme data governance
* Fáze 4: Zisk

**Životní cyklus softwaru**

* Vytvoření dat / zisk dat -> Uložení dat – bezpečí dat -> Údržba (maintenance) dat – aktualizace -> Syntéza dat - např. predikce, doporučení -> Použití dat -> Publikace dat -> Archivace dat -> Destrukce dat

**Data governance**

* Všechny organizace rozhodují o datech bez ohledu na to, zda mají formální funkci data governance
* = rámec lidí, zásad, procesů a technologií, které definují způsob správy dat vaší organizace
* -> strategie, politika, normy a kvalita, dohled, dodržování předpisů, management problémů, hodnocení data asssetů, projekty data managementu
* Je nutné rozhodnout: zásady zpracování dat, kvalitu dat, metadata, přístup k datům, životní cyklus dat

**Organizace data governance**

* Funkce: definování zásad, standardů a podnikové datové architektury, soudní funkce – správa a eskalace problémů, výkonné – ochrana a služba, administrativní povinnosti
* Řízení dat je nejčastější označení pro odpovědnost a zodpovědnost za data a procesy, které zajišťují efektivní kontrolu a využívání datových aktiv.
* Typy provozních modelů: Centralizovaný / Federativní

**FAIR**

* nutné zlepšit infrastrukturu podporující opakované použití vědeckých dat -> Zásady FAIR pro vědecká data
* Dobrá data management je spíše klíčovým prostředkem vedoucím k objevování znalostí a inovací a k následné integraci a opětovnému využití dat a znalostí komunitou po procesu zveřejnění dat
* FINDABILITY - Nalezitelnost - metadata a data by měla být snadno nalezitelná pro lidi i počítače.
* ACCESSIBILITY - Přístupnost - Uživatel musí vědět, jak k nim může přistupovat včetně ověřování a autorizace.
* INTEROPERABILITA - Data je obvykle třeba integrovat s jinými daty / pracovními postupy.
* REUSABILITY – Znovupoužitelnost - Metadata a data by měla být dobře popsána

**Typy zdrojů -** Data nebo digitální objekty, metadata – info o digitálních objektech, infrastruktura

**FAIR rules:**

* Findability
  + F1. (Meta)datům je přiřazen globálně jedinečný a trvalý identifikátor - IRI
  + F2. data jsou popsána bohatými metadaty (definovanými v R1)
  + F3. metadata jasně a explicitně obsahují identifikátor dat, která popisují - IRI
  + F4. (meta)data jsou registrována nebo indexována ve vyhledávacím zdroji
* Accessibility
  + A1. (meta)data lze vyhledat podle jejich identifikátoru pomocí standardizovaného komunikačního protokolu – HTTP, FTP
  + A1.1 protokol je otevřený, bezplatný a univerzálně implementovatelný – HTTP, FTP
  + A1.2 protokol umožňuje v případě potřeby postup autentizace a autorizace – HTTP, FTP
  + A2. metadata jsou přístupná, i když data již nejsou k dispozici
* Interoperabilita
  + I1. (meta)data používají formální, přístupný, sdílený a široce použitelný jazyk pro reprezentaci znalostí – JSON-LD, RDF
  + I2. (meta)data používají slovníky, které se řídí zásadami FAIR.
  + I3. (meta)data obsahují kvalifikované odkazy na jiná (meta)data – IRI
* Reusability
  + R1. meta(data) jsou bohatě popsána množstvím přesných a relevantních atributů
  + R1.1. (meta)data jsou zveřejňována s jasnou a přístupnou licencí na používání dat – MIT, CC-BY
  + R1.2. (meta)data jsou spojena s podrobným původem – PROV-O
  + R1.3 (meta)data splňují komunitní standardy relevantní pro danou oblast – SDMX

**Academic Publishers (ACM)**

* Experimentální výsledek není plně prokázán, pokud jej nelze nezávisle reprodukovat
* Repeatability (opakovatelnost) – stejný tým, stejný experimentální setup
* Reprodukovatelnost – jiný tým, stejný experimentální setup
* Replikovatelnost – jiný tým, jiný experimentální setup

**FAIR software**

* **F**indability – Nalezitelnost - Jednoznačná identifikace pro běžné vyhledávací strategie.
* **A**ccessibility - Dostupnost - Pracovní verze musí být buď ke stažení a/nebo přístupná např. přes webové rozhraní spolu s požadovanou dokumentací a licenčními informacemi.
* **I**nteroperabilita - Důraz je kladen na výměnu informací.
* **R**eusability - znovu použitelnost - Usiluje o to, aby někdo mohl opakovaně používat software

**FAIR principy pro software**

* Findability:
  + F1. Software a související metadata mají pro každou vydanou verzi globální, jedinečný a trvalý identifikátor.
  + F2. Software je popsán bohatými metadaty.
  + F3. Metadata jasně a explicitně obsahují identifikátory všech verzí softwaru, které popisují.
  + F4. Software a s ním spojená metadata jsou zahrnuty do registru softwaru s možností vyhledávání.
* Accessibility
  + A1. Software a s ním spojená metadata jsou přístupné podle svého identifikátoru pomocí standardizovaného komunikačního protokolu.
  + A1.1 Protokol je otevřený, bezplatný a univerzálně implementovatelný.
  + A1.2 Protokol umožňuje v případě potřeby postup autentizace a autorizace.
  + A2. Metadata softwaru jsou přístupná, i když software již není k dispozici.
* Interoperabilita
  + I1. Software a související metadata používají formální, přístupný, sdílený a široce použitelný jazyk, který usnadňuje strojovou čitelnost a výměnu dat.
  + I2S.1. Software a související metadata jsou formálně popsány pomocí řízených slovníků, které se řídí zásadami FAIR.
  + I2S.2 Software používá a vytváří data v typech a formátech, které jsou formálně popsány pomocí řízených slovníků, jež se řídí zásadami FAIR.
  + I4S. Závislosti softwaru jsou zdokumentovány a existují mechanismy pro přístup k nim.
* Reusability
  + R1. Software a související metadata jsou bohatě popsány množstvím přesných a relevantních atributů.
  + R1.1. Software a s ním spojená metadata mají nezávislé, jasné a přístupné licence k použití kompatibilní se závislostmi softwaru.
  + R1.2. Metadata softwaru obsahují podrobnou provenienci, úroveň podrobnosti by měla být dohodnuta komunitou.
  + R1.3. Metadata a dokumentace softwaru splňují komunitní standardy relevantní pro danou oblast.

**Data API**

* Potřebujeme mít možnost data nějak zveřejňovat

**Data dump - výpis**

* Nejjednodušší, neměnné - když se mi data za půl hodiny změní už to nezohledním, např. vygeneruju csv

**HTTP API** – jedna z možností, ale neexistuje žádná standardizace a není jasné, jak API navrhnout

**REST**

* architektonický vzor pro návrh webových aplikací, není to API, např. HATEOAS
* **Bezstavový**: viditelnost, spolehlivost a škálovatelnost, viditelnost se zlepšuje
* **Cache**: vyžadováno, aby data v odpovědi na požadavek byla implicitně / explicitně označena jako je/není cachovatelná, pokud je cachovatelná, pak má klient cache právo znovu použít tato data odpovědi pro pozdější, rovnocenné požadavky.
* **Layer systém**: Pro požadavky internetového rozsahu. Takové chování komponent, že každá komponenta "nevidí" dál než do bezprostřední vrstvy, se kterou komunikuje. Latenci kompenzuje cache
* **Code-on-Demand**: REST umožňuje rozšíření funkčnosti klienta stažením a spuštěním kódu ve formě appletů nebo skriptů. To zjednodušuje klienty tím, že snižuje počet funkcí, které je třeba předem implementovat

**Vrstvy RESTu**

* + Level 0: the Swamp of POX
    - Plain old XML (POX)
    - Jednotlivé URL interakce
    - URL design
  + Level 1: Resources
    - Místo jednoho end pointu, chceme vystavit nějaké zdroje
    - For example: /article, /articles/1, /articles/1/comments
    - Resource design
    - Něco co chceme přenášet klientovi
  + Level 2: HTTP Verbs
    - Použití HTTP slov: GET, POST, PUT, DELETE, …
    - Status code: 200 OK, 304 Not Modified, …
    - URL query arguments - filtrování, …
    - Non-RESTful akce jako RPC -> sub-resources, fronta akcí
  + Level 3: Hypermedia Content
    - JSON Hypermedia API Language
    - JSON-LD

**Linked Data Platform**

* + Vezme resources (dokumenty na url) a umožníme uživateli provádět základní kroky - pomocí protokolu HTTP: vyhledávání zdrojů, sledovat odkazy, publikovat nové zdroje, upravovat zdroje, mazat zdroje
  + Máme už přesně řečeno, jak to implementovat
  + Blízko k rdf

**OpenAPI (swagger)**

* máme data a potřebujeme poskytnout ostatním přístup k vašim datům, aby z nich mohlo těžit více uživatelů
* zveřejníme je pomocí OpenAPI = specifikace umožňující popsat rozhraní založené na protokolu HTTP
* např. YAML

**Command-based přístup: GraphQL**

**Data katalogy a metadata**

**Hairball architecture**

* Systém zkombinovaný z menších částí, které jsou vzájemně propojené
* Ve chvíli, kdy v jedné komponentě jednu věc změním, může úplně jiná komponenta přestat fungovat

**Data warehouse**

* Sloučení spousty datových zdrojů do jednoho
* ETL scripty napíšeme jen mezi primárními zdroji a data warehousem => sloučíme data do jedné struktury
* Entropie (míra složitosti) v čase roste – nedá se tomu vyhnout
  + Nový analytik -> chce napojit nový datový zdroj – už si to tak nepřipravíme, nepromyslíme, lepíme to
  + Potřebujeme jinou strukturu dat -> vytvoří se nová instance warehousu -> čím dál větší míra entropie
* New data warehouse = new data silo
  + Více ostrůvků, které o sobě vzájemně nevědí
  + New data warehouse se snaží všechna sila propojit, ale vzniká nám nové datové silo

**Datový katalog**

* Dataset = kolekce dat, publikovaná jako datový zdroj, a dostupná pro přístup a stažení v nějakém formátu
* Data source = systém, které umožňuje ukládat jednotlivé datasety a přístup uživatelů k datům
* Datový katalog = metadata management nástroj, k organizaci, vyhledávání a popisu datasetů
* Active metadata = metadata vytvářená strojově

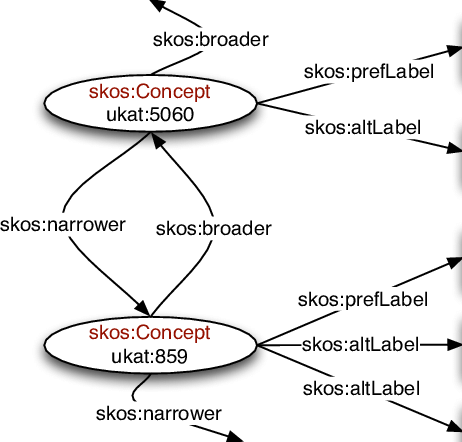
**Metadata**

* **Technická metadata**
  + Zásadní, popisují jak je dataset dostupný, jak k němu přistoupit - kde je uložený, jaký je přístupový protokol, jaký je jejich formát
* **Strukturové metadata**
  + Říká, kde jsou dostupná další data, jak jsou propojena, jakou mají strukturu, jaké jsou tu datové typy, jak se jmenují sloupečky, …
  + Většinou jako strojově čitelné schéma - JSON Schema, XSD, CSVW, …
* **Provenance metadata**
  + Odkud data pocházejí, kdo a kdy je vytvořil, za jakých podmínek, jsou kompatibilní s nějakými daty, …
* **Doménová metadata** 
  + Popisují sémantiku, význam uložených dat, co který název znamená, …
  + Title, popis, klíčová slova, témata, jakého časového úseku se obsah datové sady týká, prostorové omezení
* **Business metadata**
  + Upozornění pro jednotlivá data - např. u osobních dat, nemůžeme zveřejňovat původní data, …
  + Zákon o ochraně osobních údajů
  + Licence, přístupová práva a politiky
* **RDF je standartní model pro sdílení a publikování (meta)dat**

**DCAT (Data Catalog Vocabulary)**

* Definuje W3C, na tomto standartu je postavena interoperabilita katalogu EU
* V případě že potřebuju dataset sdílet -> katalogy si musí rozumět
* Slovník navržený pro usnadnění interoperability metadat datasetů
* Definuje strukturu metadat, vztahy mezi datasety, …
* Basic DCAT koncept:
  + dcat:Resource
    - Dataset, datová služba nebo jiný zdroj, který můžeme popsat metadaty
  + dcat:Dataset
    - Reprezentovaný dataset je kolekce dat publikovaná nebo sdílená
  + dcat:DataService
    - Kolekce operací, které můžeme volat na jednom nebo více datasetu
  + dcat:distribution
    - Každý data set má jednu nebo více distribucí – reprezentace přístupné formy datasetu jako soubor ke stažení
  + dcat:Catalog
    - Reprezentuje katalog, který je popisuje dataset, kde každý item je zdrojem

**Controlled vocabularies** (řízená slovní zásoba)

* = standardizovaná a organizovaná uspořádání slov a frází s definovaným významem (sémantikou)
* Hlavní cíl: jednotnost a jasnost komunikace -> všichni uživatelé jsou nuceni používat předem definované termíny, což minimalizuje nejednoznačnost a různorodost výrazů
* Obsahuje synonyma, hierarchické vztahy mezi termíny (nadřazené/podřazené) a další relace, které usnadňují strukturování informací a jejich vyhledávání
* **Doménová metadata:** title, description, keywords, themes, temporal a spatial resolution
  + - Title a description jsou nestrukturované textové charakteristiky významu
    - Keywords jsou volné textové charakteristiky významu strukturovaného do frází
    - Themas jsou koncepty z daného controlled vocabulary, které charakterizují význam
  + **Problém s klíčovými slovy:**
    - Lidé mají tendenci charakterizovat nejen význam (např. technické formáty, zdroj dat atd.)
    - Různí lidé mají tendenci volit klíčová slova na různých úrovních podrobnosti
    - Různí lidé volí různá klíčová slova pro stejný význam
    - Různí lidé volí stejné klíčové slovo pro různé významy
    - Klíčová slova jsou pouze slova, která je potřeba přeložit do jiných jazyků
    - Klíčová slova jsou pouze izolované ostrovy s pouze implicitní sémantikou
  + **Controlled list** = jednoduchý seznam pojmů bez jakýchkoli vztahů mezi pojmy
    - Každý koncept (pojem) je unikátní
    - Každý koncept má obvykle pouze jeden termín nebo více termínů ekvivalentních v různých jazycích
    - Koncepty se významově nepřekrývají
  + **Taxonomie** = controlled vocabulary, který uspořádává koncepty (pojmy) do hierarchické struktury
    - Každý koncept je jedinečný
    - Každý koncept má obvykle pouze jeden termín nebo více termínů ekvivalentních v různých jazycích
    - Koncepty se mohou svým významem překrývat
    - Hierarchické vztahy mohou být různého typu: širší/užší, rodič/dítě, celek/část, obecný/specifický, …
  + **Tezaurus =** controlled vocabulary, který definuje preferované a alternativní (synonymní) termíny a který propojuje pojmy pomocí hierarchických užších/širších vztahů a nehierarchických asociativních vztahů
    - Každý koncept je unikátní
    - Každý koncept má obvykle preferovaný termín a jeden nebo více alternativních termínů, každý v jiném jazyce
    - Koncepty se mohou svým významem překrývat
    - Koncepty souvisejí s: hierarchickými širšími/užšími vztahy, nehierarchickými asociativními vztahy
  + **Klasifikační schéma** = controlled vocabulary, který definuje pojmy pro uspořádání rozdělení objektů do skupin na základě vlastností, které mají objekty společné
    - Používá se pro klasifikaci statických pozorování, např. počet zaměstnanců podle ekonomické aktivity
    - Je strukturováno do klasifikačních úrovní, kde hierarchické vztahy jsou pouze mezi pojmy ze dvou sousedních úrovní
    - Je podobné taxonomii

**Reprezentace a použití controlled vocabularies**

**SKOS – simple knowledge organization systém**

* model pro vyjádření základní struktury a obsahu controlled vocabularies
* Concept, conceptScheme, prefLabel
* pro sdílení a opakované použití řízených slovníků jejich vyjádřením v datovém modelu RDF a jejich publikování na WWW

**Jak použít controlled vocabulary k popisu dat**

* Zdroj může být propojen s koncepty z jednoho nebo více controlled vocabulary, aby bylo specifikováno, že se zdroj týká těchto pojmů
* Dataset lze propojit s koncepty (pojmy) z jednoho nebo více controlled vocabulary a určit tak, že se zdroj týká těchto pojmů

**Jak se dotazovat na popisy dat**

1. Vyhledávání nebo vytvoření controlled vocabulary, např. EuroVoc
2. Vytvořit záznam metadat s anotacemi ze slovníku
3. Přístup nebo dotazování se na záznamy metadat a slovník

**Specifikace controlled vocabulary vyjádřených jako SKOS concept schema**

* Zveřejněny na WWW jako množina RDF trojic (výrazů)
  + - Kdokoli může koncept znovu použít ve vlastním controlled vocabulary
    - Kdokoli může přidat koncept do controlled vocabulary zveřejněného někým jiným
  + => SKOS concept schémata jsou založena na předpokladu otevřeného světa
    - Nikdy nemáte k dispozici kompletní informace
    - Chybějící výrok neznamená, že neexistuje, ale že o něm jen nevíme
  + Různé koncepty z různých pojmových schémat lze vzájemně propojovat / integrovat různá pojmová schémata

**Ontologie = sdílená konceptualizace věcí**

* poskytuje formální rámec pro popis jednotlivých entit v dané doméně, jejich vlastností a vztahů mezi nimi
* konceptualizace je odsouhlasena a sdílena skupinou lidí, formálně je založena na společné souboru pojmů, které definují základní význam
* důležité věci jsou reprezentovány jako vzájemně související pojmy, patří sem konkrétní nebo abstraktní věci, jejich typy, vlastnosti a vztahy mezi nimi
* používají ji např. OWL (Web Ontology Language) nebo RDF -> strukturovaný způsob vyjádření ontologických modelů -> umožňují definovat koncepty, třídy, vlastnosti, hierarchie, omezení a další aspekty ontologie

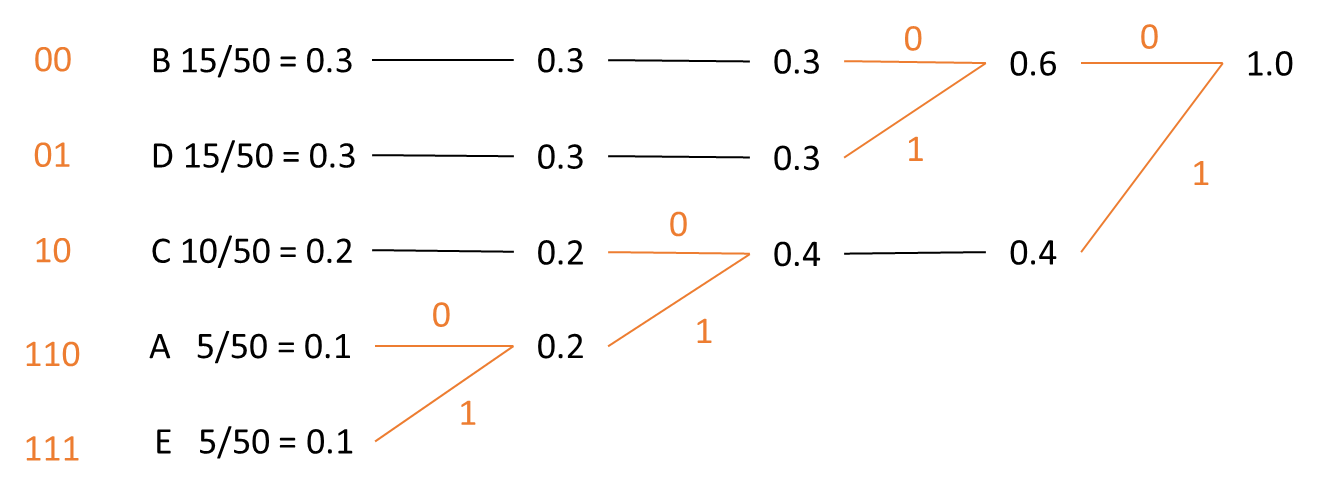
**Informační teorie**

* Komunikace dat – sdílení, ukládání, archivování -> snaha o udržení kvality dat
* Efektivní a důvěryhodná komunikace
  + Zdroj -> transmitter (kabel) -> kanál + šum -> receiver (přijímač) -> cílová destinace
  + Ve zdrojovém kódu odstraňujeme redundanci, aby byla data co nejmenší, kanál ale může dodat šum, a tak zde přidáváme jistou míru redundance by bylo možné detekovat chyby
  + Shannon entropie (v bitech) H(X) = míra neurčitosti, pokud existuje pouze jeden výsledek dostaneme 0
* **Shannonův teorém o kódování zdroje:** N výsledků ze zdroje X lze zkomprimovat zhruba do N \* H(X) bitů
* Optimální metoda komprese odstraní veškerou redundanci z dat

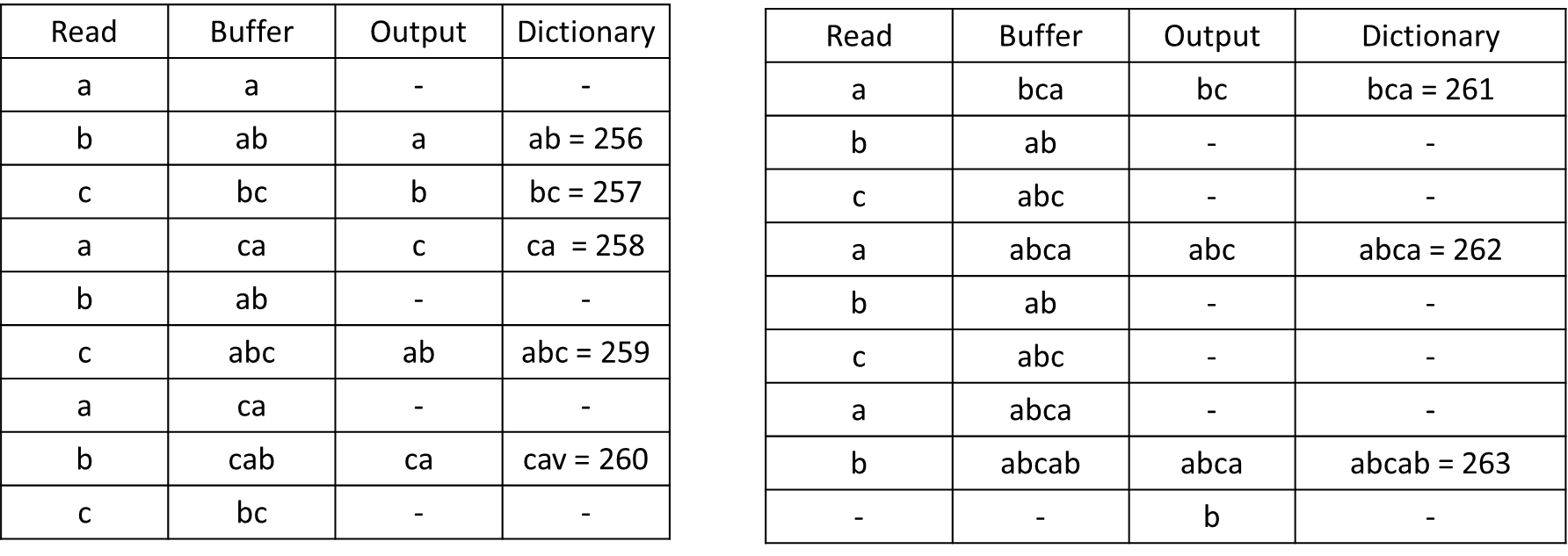
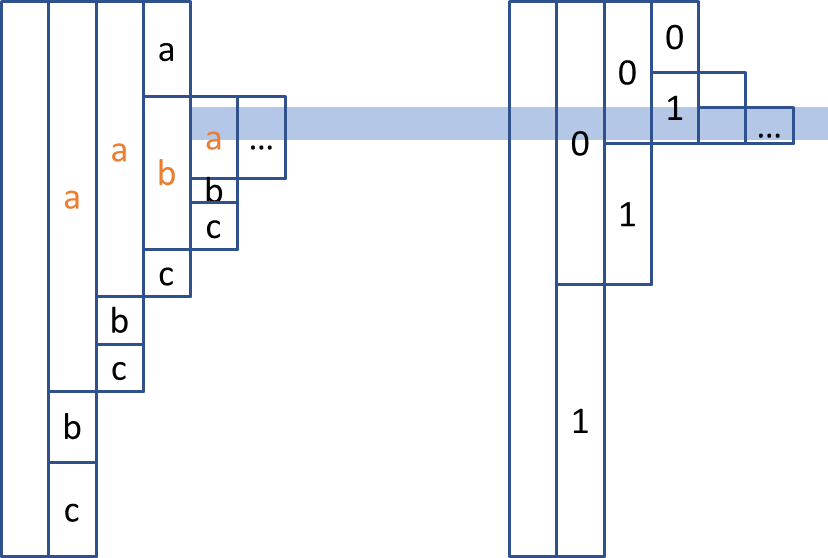
**Run-length encoding**

* Input: CCCCCBBBBBDDDDBBBBBAAABBBBBDDDDEEEEEDDDDDDDCCCCCAA
* Output: 5C5B4D5B3A5B4D5E7D5C2A
* Bezeztrátová komprese dat, při níž jsou datové řady (sekvence, v nichž se stejná hodnota dat vyskytuje v mnoha po sobě jdoucích datových prvcích) uloženy jako jediná hodnota dat a počet, nikoli jako původní řada.
* Myšlenka je jednoduchá, ale nemusí fungovat, protože nemusíme mít tak pěkné bloky informací

**Huffmanovo kódování**

* Input: CCCCCBBBBBDDDDBBBBBAAABBBBBDDDDEEEEEDDDDDDDCCCCCAA
* ****Sloučíme dva uzly s nejmenší pravděpodobností.
* Jakmile je strom vytvořen, stačí přidat štítky od kořene.
* Uzly s největší pravděpodobností mají nejkratší kódy. Vidíme také, že žádné kódové slovo není prefixem jiného kódu, takže je velmi snadné to rozluštit.
* Nutné znát pravděpodobnosti -> musíme si dokument přečíst -> statická metoda -> dynamické verze, kdy se pravděpodobnosti konstruují v průběhu komprese datového toku

**Lempel-Ziv 1977**

* Univerzální metoda založená na slovník -> začněte základním slovníkem ~ základní znaky (ASCII).
* Aktualizace slovníku jako součást procesu komprese
* Princip: nalezení opakujících se vzorů v textu a jejich následném nahrazení kratšími symboly -> komprese
* Input: (with spaces for readability): abc abc abc abc abc ab
* Kroky komprese (odesílatele): Přečteme "a", máme "a" ve slovníku -> přečteme "b", do slovníku musíme přidat "ab", odešleme "a", ponecháme si "b" -> přečteme "c", do slovníku přidáme "bc", odešleme "b" -> Přečteme "a", a "ca" do slovníku odeslat "c" -> přečteme "b", ve slovníku máme "ab" -> přečteme "c", přidáme "abc" do slovníku, odešleme "ab" (už ve slovníku)

**Aritmetické kódování**

* Input (with spaces for readability): aab aba cba ...
* máme model, který odhadne další hodnotu, a my si jen pamatujeme počet odhadů. Pokud je tento počet menší, pak velikost dat (bajt vs. unicode) můžeme použít jako kompresi
* přidělování intervalů číselného intervalu, které reprezentují jednotlivé symboly textu

**Šum**

* **Shannonův noisy-channel coding teorém:** Přes šumový kanál je možné komunikovat s libovolně malou pravděpodobností chyby, pokud je rychlost komunikace udržována pod kapacitou kanálu (C).

**Budování důvěry**

* Symetrické: Sdílený klíč, |key| = 256 bitů, Rychlá
* Asymetrické šifry: Soukromý/veřejný klíč, |key| = 2048 bitů, Pomalý, Slouží k výměně symetrického klíče

jeden klíč potřebujeme k šifrování a druhý k dešifrování -> k posílání soukromých zpráv (bezpečná komunikace)

* **Caesarova šifra:** Fixní abeceda -> Posun abecedy o zadaný offset

**Advanced Encryption Standard (AES)**

* 128bitová symetrická šifra s klíčem o délce 128/192/256 bitů
* principu substituce a permutace -> při šifrování i dešifrování se používá stejný klíč
* Algoritmus: Použijte padding pro získání 128bitových bloků -> Uspořádání dat do matice 4 x 4 s 8 bity na buňku -> XOR s částí klíče -> Okruhy 10/12/14 podle velikosti klíče: Nahrazení bajtů pomocí předdefinované tabulky – Posunutí (shift) řádků - promíchejte sloupce - XOR s Round klíčem

**Diffie-Hellman**

* Výměna tajných klíčů, Operace s více stranami, Diffie-Hellmanova eliptická křivka, „osoba“ uprostřed

**Rivest-Shamir-Adleman (RSA)** - Asymetrická šifra, existuje alternativa založená na eliptických křivkách.

* Potřebujeme velká prvočísla a funkci φ, tj. počet menších čísel než N bez společného činitele -> Spoluprvočísla (nemají společného činitele) -> ale protože N je ze dvou prvočísel, tak to můžeme spočítat rychleji
* Dvojice soukromý/veřejný klíč umožňuje nejen bezpečně někomu poslat zprávu, ale také se ujistit, že určitou zprávu někdo vytvořil

**Hashování**

* Jednosměrná transformace dat na otisk prstu pevné velikosti
* Podobné (ale různé) vstupy mají zcela odlišné hashe
* Jeden způsob je snadný, těžko se hledají kolize

**Digitální podpis**

* **Autentičnost**: data jsou stejná jako data ze zdroje, zvláštní případ integrity, použití asymetrické šifry a podpisů
* **Integrita:** Dokument nebyl změněn, Hash dokumentu
* Podpis závisí na podepsaném dokumentu. Není tedy možné jej vytvořit předem
* Vlastník dokumentu musí zajistit jeho platnost - právní hledisko
* Certifikát používá pokročilé elektronické podpisy (AES) s privátním klíčem
* Vezmeme hash dokumentu (SHA256), přidáme metadata, opětujeme naším soukromým klíčem. To je důkaz, že dokument byl vytvořen určitým subjektem (námi). Potřebujeme ale také dát ostatním vědět, že soukromý klíč vlastníme (důvěra), proto použijeme certifikát. Běžně se certifikát přidává k podpisům

**JSON Web Tokeny**

* formát pro reprezentaci a přenos dat v bezpečném a kompaktním formátu. JWT je typicky používán pro ověřování a autorizaci uživatelů v aplikacích, zejména ve webovém prostředí
* Tři části odděleny tečkami: hlavička (header), tělo (payload) a podepsaný podpis (signature)
* Pro zabezpečení tokenu je použit digitální podpis

**Digitální certifikát**

* Poskytnout uživateli důvěru k určité autoritě
* Certifikát je veřejné prohlášení o vlastnictví soukromého klíče.
* Platnost certifikátu, potvrzení online identity v síti, měl by obsahovat veřejný klíč
* Standard digitálního certifikátu X.509

**Public key infrastruktura (PKI)**

* Existují kořenové certifikační autority, které jsou důvěryhodné: Certifikáty těchto kořenových autorit jsou dodávány jako součást operačního systému / softwaru. Spolu s ostatními certifikáty tvoří řetězec důvěryhodnosti. Pokud někdo chce certifikát, musí podat žádost a certifikační autorita jej ověří (telefonicky, osobně). Součástí žádosti je obsah certifikátu (např. veřejný klíč). Po ověření certifikační autorita certifikát podepíše a pošle zpět. Je velmi důležité, aby soukromé klíče certifikační autority byly zabezpečeny.
* Seznamy zneplatněných certifikátů (Certificate Revocation Lists)
* Úschova klíčů

**HTTPS, SSL, TLS**

* Secure Sockets Layer (SSL) nahrazen Transport Layer Security (TLS)
  + TLS je nad vrstvou TCP, pod vrstvou HTTP. Protokol TLS 3.0 lze navázat během dvou kol
  + Proč ho potřebujeme: Definování, jaké šifry lze použít, slabé šifry jsou zastaralé, Autentizace ~ prokazuje, že jedna strana je ta, za kterou se vydává
  + Odolnost vůči: Main In The Middle, Replay (opakované odeslání stejné zprávy), Downgrade (vynucení použití slabé šifry)
  + TLS používá protokol HTTP, i když ne všechny servery podporují TLS 1.3.
* HTTPS = HTTP odesílaný přes šifrovaný kanál (TLS)

**Text search**

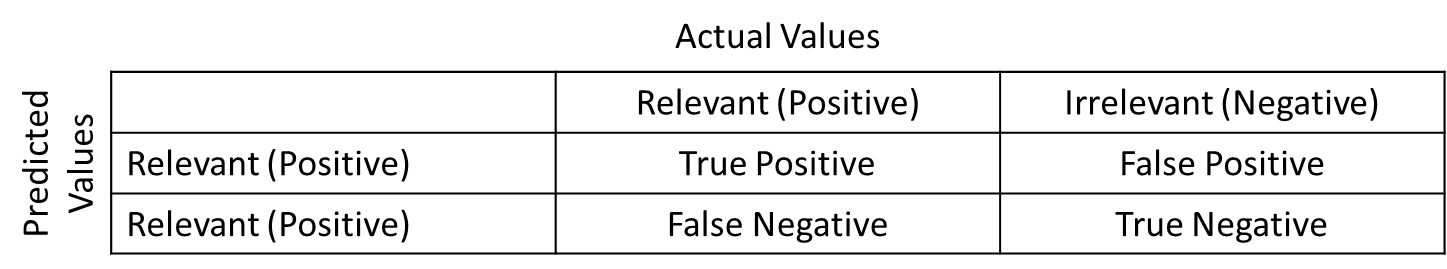
**Přístupy:**

* + Aho-Corasick – prefixový strom, musím dokument přečíst dopředu
  + Můžeme si už projít text a roztřídit podle abecedy např.

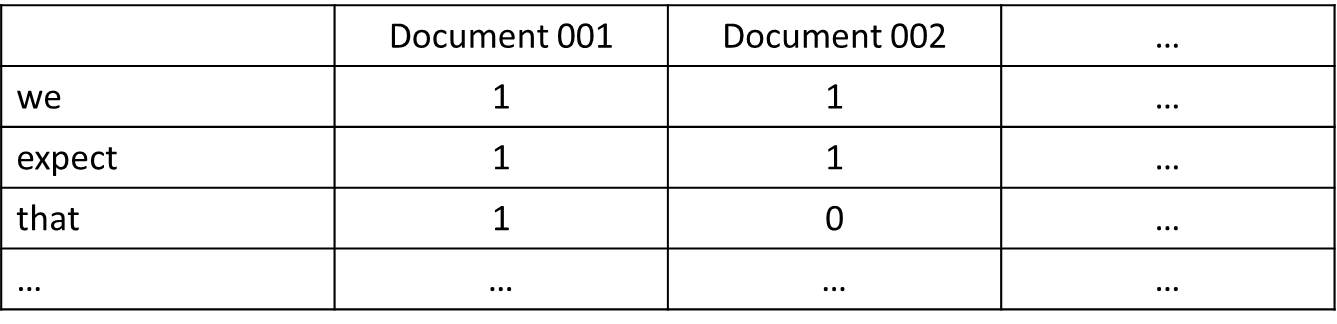
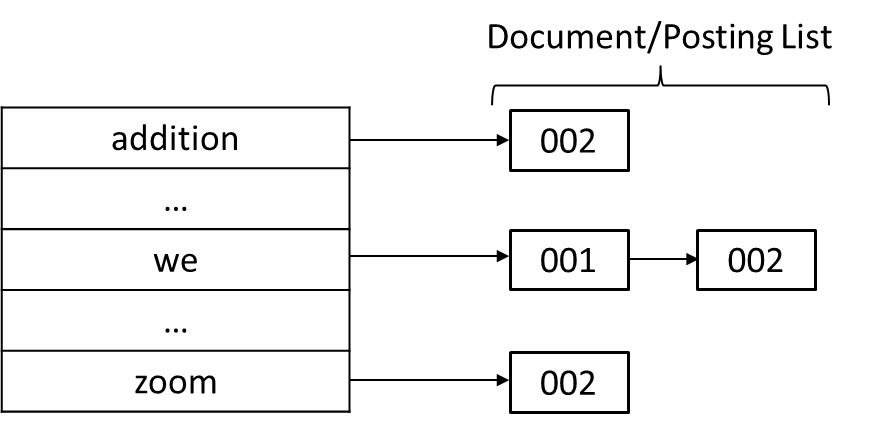
**Searching a dictionary**

* + Můžeme dokument vzít a rozdělit ho na jednotlivé části např. pomocí mezer
    - Vytvoříme si hash mapu -> ve většině případů výrazně zmenšíme objem dokumentu
    - A pak např. vytvoříme prefixový strom pro jednotlivá slova
  + Můžeme hledat jen pozici slova

**Searching:**

* + Hledáme např sousloví, dvojici slov
    - Musíme si ujasnit, co vlastně chceme vracet, jestli hledáme slova, co jsou přímo za sebou nebo hledáme dvě slova, co spolu nějak souvisí (např. jsou do nějaké vzdálenosti od sebe)
    - klasické textové vyhledávání dnes nestačí (potřebujeme kombinovat podobné významy, různé jazyky…)
  + Relevance
    - Míra toho, jak moc je uživatel spokojen s výsledky, které se mu vrátili na jeho dotaz
    - Vnášíme kontext na základě, něhož hledáme
    - Hodnocení pomocí Precision vs. Recall
      * Precision = TP / (TP + FP) = P (relevant | retrieved)
        + Kolik z toho, co jsme našli, jsme chtěli najít
      * Recall = TP / (TP + TN) = P (retrieved | relevant)
        + Kolik z toho, co jsme chtěli najít jsme našli
      * False positive, false negative, true positive, true negative

**Full-text search**

* + Uživatel poskytne nějakou malou sadu klíčových slov a cílem je vrátit dokumenty, ve kterých se nějakým způsobem tato sada slov nachází
  + Měl bych dovolit velkou variabilitu - jiné tvary slov, překlepy, synonyma
  + Vracím výsledky seřazené podle daných parametrů
  + **Realizace:**
    - **Term-document incidence matrix**
      * Dokumenty jsou sloupce, slova řádky -> vytvořím matici 1/0 - je/není, nebo 0/n – není/kolikrát je
      * Při vyhledávání poté přistupuju jen k jednotlivým řádků tabulky (slovům), která hledám
      * Velice rychlé
      * Nevýhoda, pokud slov bude hodně, matice bude velmi řídká
      * **-> Inverted index**
    - **Inverted index**
      * Pro slovo si pamatujeme jen dokumenty, v kterých slovo je = document/posting list
      * Používá se velmi často
  + Transakční zpracování?
    - Mazání, upravování a přidávání dokumentů
    - Co se pak děje s maticí

**Jak budovat indexy?**

* + Collect documents -> token extraction -> preprocessing -> build index (inverted index)
  + Collect documents - Je nutné mít přístup ke všem dokumentů (netriviální operace)
  + Token extraction
    - word – z textu, term - normalizované slovo (např. ignoruji velká malá písmena), token - instance slov
    - Full-text vs. Fulltext vs. Full text
    - U.S.A. vs. USA
    - 6/1/21 vs. Jun. 1, 2021 vs. 1.6.2021
    - Languageswithoutspaces (Chinese)
  + Preproccesing
    - Odstranění háčků a čárek
    - Normalizace malých velkých písmen např. mit vs. MIT -> ve chvíli, kdy normalizujeme, tak pak nebudeme moct najít MIT (např. v němčině)
    - Car vs. Automobile (synonyma)
    - Speling - překlepy
    - Lemmatization - car vs. cars, car's vs. cars', různé tvary, časy
    - Stemming: automat vs. Automate(s) vs. Automatic vs. Automation
    - Stop words: a, an, and, are, as, at, be, of
    - Sousloví - King of Denmark - zde se např. nechci zbavit of
  + Ve většině případů si jen musíme vybrat správnou službu, která všechno vykoná za mě