

# Správní informační systém

## *Technická specifikace*

# Obsah

1. Úvod	1
2. Struktura uživatelského rozhraní	2
2.1. Obecné principy	2
2.1.1. Seznamy položek	2
2.1.2. Historie	3
2.1.3. Tisk	3
2.1.4. Export	3
2.2. Konkrétní součásti	3
2.2.1. Pracovní přehled	3
2.2.2. Přehled událostí	4
2.2.3. Sekce	4
2.2.4. Témata	4
2.2.5. Pohledy	4
2.2.6. Sestavy	4
2.2.7. Případy	5
2.2.8. Dokumenty	5
2.2.9. Předvolby uživatele	5
2.2.10. Předvolby systému	5
3. Pohledy, komponenty a zdroje dat	7
3.1. Příklady typů komponent	7
3.1.1. Mapy	7
3.1.2. Číselníky	8
3.1.3. Diagramy	8
3.1.4. Tabulky	8
3.1.5. Speciální	8
4. Případy	9
4.1. Typy případů	10
4.2. Účast na případech	10
4.3. Žádost o práci na dokumentu	11
4.4. Lhůta pro vyhotovení dokumentu	11
5. Dokumenty	12
5.1. Typy dokumentů	12
5.1.1. Závislosti	12
5.1.2. Kvantifikace	13
5.1.3. Typové třídy	14
5.1.4. Schéma	15
5.2. Komponenty	18
5.3. Verze a revize	18

5.4. Podepisování .....	18
5.4.1. Elektronické .....	19
5.4.2. Vlastnoruční .....	19
6. Model oprávnění .....	20
6.1. Uživatelské role .....	21
6.2. Způsoby manipulace .....	21
7. Upozornění na událost .....	22
7.1. Události .....	22
8. Implementační detaily .....	24
8.1. Architektura .....	24
8.2. Datová vrstva .....	24
8.2.1. PostgreSQL .....	24
8.2.2. Dokumenty .....	24
8.2.3. Zdroje dat .....	24
8.3. Uživatelské rozhraní .....	26

# 1. Úvod

Správní informační systém, zkráceně *SpIS* je otevřený nástroj pro vedení elektronických evidencí. Měl by zpřehlednit výkon organizačních úkonů, mezi které patří například příprava a jednotlivé kroky realizace veřejných zakázek včetně zpracování podkladů pro fakturaci, plánování rozpočtu a jednotlivých výdajových záměrů nebo nastavení pracovních procesů a schvalovacích mechanismů dle rolí přidělených pracovníkům dané organizace. Systém bude tedy v praxi evidovat doklady a další dokumenty.



Bez ohledu na specifikaci níže platí, že SpIS nebo jeho nasazení musí vyhovět stávající legislativě. Například v oblasti ochrany osobních údajů či bezpečnosti.

Výběr z funkcí, které očekáváme v prvních verzích:

## Hierarchické pracovní struktury

Rozklad případu na jednotlivé dodávané výsledky (což jsou pro nás dokumenty) v předepsané hierarchii (dle typu případu). Tento postup se jinde označuje pojmem [WBS](#).

## Editor dokumentů

Nástroj pro vytváření a úpravy dokumentů, představující uživatelské rozhraní uzpůsobené jejich konkrétním typům. Využívá zásuvné moduly, které realizují dílčí komponenty rozhraní. Každý typ dokumentu popisuje, jaké komponenty se mají pro jeho pole využít.

## Přehledy

Vhodně nastavené přehledy čerpající z případů a dalších evidencí. Pro jejich zobrazení se využívají komponenty realizované jako zásuvné moduly. Tyto komponenty se provazují s konkrétními daty, čímž vznikají pohledy, které se dále seskupují do témat a sekcí.

## Oprávnění

Pružný systém oprávnění založený na uživatelských rolích a organizačních jednotkách umožňuje pracovat pouze s těmi dokumenty a na těch případech, které se uživatelů týkají. Uživatelské účty se vždy přebírají z vnější evidence uživatelů.

## Export dat a tisk

Systém bude obsahovat nástroje pro export dat, tisk dokumentů, pohledů a vybraných přehledů formou jednoduchého tisku a tiskových šablon.

SpIS bude napojen na vybrané další informační systémy, ze kterých bude čerpat podklady a do kterých bude předávat výstupy nebo dílčí dokumenty. Napojení bude realizováno prostřednictvím datové vrstvy, tedy v databázi.

## 2. Struktura uživatelského rozhraní

Uživatelské rozhraní je realizováno formou interaktivního webového portálu. Každý dokument, případ, každé téma či sekce, každá část nastavení je webovou stránkou s unikátní adresou.

Jednotlivé stránky mohou být vylepšeny pomocí klientských skriptů za účelem dosažení vyšší míry uživatelské přívětivosti, ale musí svou funkci plnit i se zakázaným skriptováním. Výjimku tvoří pohledy. Ty ke svému ztvárnění vyžadují pokročilé funkce (vykreslování grafů) na straně klienta.

### 2.1. Obecné principy

#### 2.1.1. Seznamy položek

Systém v řadě případů zobrazuje různě dlouhé seznamy položek. Pokud si má uživatel ze seznamu některé položky vybrat, nebo může na některé položce vyvolat akci (například *upravit*, *smazat*), pak pro tyto seznamy platí následující pravidla.

##### Filtrování

Seznam umožňuje filtrování položek. Tím se rozumí přechodně omezit množství zobrazených položek zadáním filtračního výrazu.

- Vždy je přípustné filtrování pomocí jednoduchého fulltextového hledání.
- V případě seznamů nebo tabulek o více sloupcích je přípustné filtrování jednotlivých sloupců zvlášť, pokud je možné kombinovat více kritérií.

##### Řazení

Pokud seznam obsahuje 1 nebo více sloupců, systém umožňuje řádky řadit podle vybraného z nich a ve výchozím zobrazení upřednostňuje (je-li to možné a podstatné):

- Řazení dle času od nejnovějšího po nejstarší.
- Vzestupné abecední řazení dle hlavního názvu.

##### Výběr sloupců

Pokud seznam obsahuje 2 nebo více sloupců, systém umožňuje uspořádat tyto sloupce v libovolném pořadí. Také umožňuje libovolné sloupce skrýt nebo zobrazit takové sloupce, které se běžně nezobrazují.

Veškerá nastavení zobrazení seznamů jsou trvalá a společná pro daný druh zobrazení. Vždy je možné se u daného zobrazení vrátit k výchozímu nastavení. Změny v nastavení se ukládají odděleně alespoň pro přehledy uvedené níže a aplikují se vždy pouze pro konkrétního uživatele.

- Pracovní přehled
- Přehledy případů (společné nastavení pro všechna témata a typy případů)
- Jednotlivé přehledy dokumentů (pro každý typ dokumentu zvlášť)

### 2.1.2. Historie

U dokumentů a případů se uchovávají předcházející verze jejich stavu. Uživatel si může zobrazit anotovaný výpis těchto verzí, ze kterého jsou patrné alespoň míra a povaha změny a dále také její autor. Dále si může libovolnou předešlou verzi zobrazit, případně ji porovnat s jinou.

Pro rychlou orientaci je také vždy možné zobrazit porovnání aktuální verze dokumentu s verzí předchozí, případně aktuální revize s předchozí. V případě otevření dokumentu skrze odkaz získaný automatickým upozorněním na změnu je uživateli zobrazen popis změny mezi revizemi.

### 2.1.3. Tisk

Každá stránka zobrazující uživatelská data (například dokumenty, případy či přehledy) podporuje stažení PDF verze pro tisk. Ta vzniká na serveru pomocí zvláštní HTML šablony a CSS stylů. Jak šablonu, tak styly je možné přizpůsobit a tím dosáhnout souladu vzhledu tištěných výstupů s vizuálním stylem organizace.

Kromě centrálních předloh je možné nastavit oddělené předlohy i pro jednotlivé typy dokumentů a případů. Předlohy pro tisk komponent jsou vždy vlastní danému typu komponenty.

V případě potřeby je možné pro konkrétní typ dokumentu stanovit další šablony pro tisk. Tyto šablony z dokumentu čerpají po datové stránce, ale formu určují zcela samostatně. Typickým příkladem využití je například příprava posudku k odeslání poštou.

### 2.1.4. Export

Uživatel může získat některá data, ke kterým má přístup, i ve strojově čitelné formě. V případě dokumentů je možné si stáhnout jejich data ve formátech JSON, YAML a XML. Data pohledů je možné získat ve formátech JSON, YAML a pokud to umožňuje druh datového zdroje (data představují soubor tabulek), tak i ve formátech CSV (jednotlivé tabulky) nebo XLSX (tabulky jako jednotlivé listy).

## 2.2. Konkrétní součásti

### 2.2.1. Pracovní přehled

Pracovní přehled je výchozí stránka uživatele po jeho přihlášení, nesměruje-li do jiné části systému. Hlavním prvkem stránky je seznam případů, do kterých má uživatel přístup, seřazený dle jeho schopnosti přispět a dále dle jejich naléhavosti.

*Řazení případů dle možnosti na nich pracovat*

1. Případy, ve kterých je uživatel aktuálně **žádán o práci** na některém dokumentu.
2. Případy, ve kterých je uživatel oprávněn **dokončit** některý dokument.
3. Případy, ve kterých je uživatel oprávněn **vytvořit** některý dokument.
4. Ostatní případy, do kterých má uživatel přístup.

V rámci prvních dvou bodů jsou případy řazeny podle nejčasnějšího termínu dokončení

dokumentů, díky kterým je případ na dané pozici. Dokumenty z třetího bodu neexistují a není tedy možné u nich sledovat termíny a zbylé případy může uživatel pouze komentovat.

### 2.2.2. Přehled událostí

Na zvláštní stránce přehledu událostí jsou každému uživateli předloženy ty události, na které byl, nebo mohl být upozorněn. Jednotlivé řádky umožňují uživateli přihlásit se k nebo odhlásit se z odběru daných událostí.

Přehled by měl sloužit pro rychlou orientaci v běžících případech a získání přehledu o situaci.

### 2.2.3. Sekce

Za účelem snazší orientace uživatelů jsou veškeré objekty v systému roztrženy do témat a tematických sekcí. Toto členění nemá žádné důsledky pro funkci, ani pro řízení přístupu. Také nijak nesouvisí s organizačním členěním.

Sekce slouží k seskupení několika témat. Není běžné, aby se věci opakovaly v různých sekcích, ale není to vyloučeno. Příkladem sekcí mohou být například **Finance**, **Legislativa** a **Vnější ukazatele**. V sekci **Finance** se setkáme s vícero tématy, v sekci **Legislativa** najdeme jen hlavní téma s případy pro práci s **Vnitřními předpisy** a konečně v sekci **Vnější ukazatele** nenajdeme sice žádné případy, ale zato se zde v hlavním tématu setkáme s řadou pohledů na data získaná při provozu instituce, například **Návštěvnost**, nebo **Skladbu uživatelů**.

### 2.2.4. Témata

Téma seskupuje pohledy a sestavy, které mají za úkol poskytnout informace o stavu věcí, se správou případů a dokumentů, které umožňují tento stav ovlivňovat. Vhodná témata mohou být například **Rozpočet** nebo **Smlouvy**.

Pokud to dává smysl, mohou se věci v tématech opakovat. Například určité pohledy mohou být přínosné pro více témat, některé případy mohou souviset s vícero tématy a podobně. Názorným příkladem budiž **Nákup**, který zasahuje jak do **Rozpočtu**, tak do **Smluv**.

### 2.2.5. Pohledy

Pohledům je v rámci daného tématu věnován hlavní prostor. Některé pohledy jsou interaktivní, takže je možné s nimi manipulovat a dovídat se tak nové souvislosti. Například pokud pohled zobrazuje mapu, je možné s ní pohybovat, přibližovat ji nebo ji oddalovat. Pokud jsou na mapě interaktivní objekty, je možné je aktivovat a zkoumat jejich detaily.

Některé pohledy mohou umožňovat přímo spustit nový případ, nebo se odkazovat na existující případy a dokumenty. Tyto vazby bývají obvykle realizovány odkazy, které může uživatel otevřít v nové záložce prohlížeče.

### 2.2.6. Sestavy

Na stránce každého tématu je vyhrazen prostor, ve kterém se nachází odkazy jednak na systémové a dále také na daným uživatelem vytvořené sestavy pohledů.

Ze stránky tématu je možné aktivovat editor sestav, který umožňuje na stránku uspořádat výběr z pohledů, ke kterým má uživatel přístup. Těmto pohledům je následně možné interaktivně nastavit potřebné zobrazení. Takto připravenou sestavu je následně možné uložit, nebo případně jen vytisknout.

### **2.2.7. Případy**

Pokud téma obsahuje alespoň jeden typ případu, je na jeho stránce k dispozici přehled případů, ke kterým má aktuální uživatel přístup. Je-li k tomu uživatel oprávněn, tak se zde také nachází prvek, který umožňuje zahájit nový případ.

Z přehledu jsou patrné základní údaje o případech a zda-li do nich může uživatel zasáhnout, případně je-li to po něm požadováno. Typy případů se rozumí zpracování obchodního případu, jako jsou nákupy a veřejné zakázky, žádost o vyjádření k území nebo společná příprava složitých stanovisek.

### **2.2.8. Dokumenty**

Pokud téma obsahuje prostřednictvím přiřazených typů případů alespoň jeden typ dokumentu, je na jeho stránce k dispozici přehled dokumentů, ke kterým má aktuální uživatel přístup. Pokud je k tomu uživatel oprávněn, může z tohoto přehledu zahájit změnu dokumentu či vytvořit nový.

Z přehledu jsou patrné základní údaje o dokumentech. Dá se z něj také zjistit, do kterých případů vstupují, má-li k nim aktuální uživatel přístup.

### **2.2.9. Předvolby uživatele**

Část systému umožňující uživateli přizpůsobit si systém dle svých potřeb. Uživatel si zde může zvolit jazyk uživatelského rozhraní a způsob upozorňování na události v systému. Upozornění týkající se dokumentů a případů je možné omezit na jejich konkrétní typy.

Více o upozorněních naleznete v kapitole [Upozornění na událost](#).

### **2.2.10. Předvolby systému**

Část systému umožňující správci nastavit jednak výchozí uživatelské předvolby, ale také případná obecná nastavení.

#### **2.2.10.1. Integroční můstky**

Za účelem řízení komunikace s navázanými cizími systémy je možné zde nastavovat a měnit předvolby integračních můstků. Děje se tak pomocí rozhraní, které poskytují zásuvné moduly daných můstků.

Pokud by existoval například můstek pro komunikaci s veřejnými rejstříky, mohli bychom mu pomocí tohoto rozhraní sdělit třeba to, že má záznamy převzaté ze systému ARES kontrolovat nejdříve po 48 hodinách.



### **2.2.10.2. Správa modelu oprávnění**

Součástí předvoleb systému je pak také správa modelu oprávnění, v rámci které je možné například nastavit párování skupin či rolí z evidence uživatelů s rolemi systému nebo upravit rozsah oprávnění jednotlivých rolí.

Více o modelu naleznete v kapitole [Model oprávnění](#).

## 3. Pohledy, komponenty a zdroje dat

Každý **pohled** nebo **část dokumentu** využívá pro zobrazení určitou komponentu. Pokud je komponenta určena pro zobrazení mapových vrstev s vyznačenými body, je možné ji použít pro zobrazení prostorových dat. Pokud se jedná o komponentu pro zobrazení tabulky, je možné pomocí ní zobrazovat data relační, pokud je určena k manipulaci s krátkým textovým popiskem, je jejím prostřednictvím možné měnit určenou část dokumentu a tak dále.

To, jaká data pohled představuje, je kromě komponenty dáno zejména souborem datových zdrojů, ze kterých pohled čerpá. Tyto zdroje mohou být při použití interaktivních komponent ovlivněny hodnotami, které uživatel při prohlížení mění. Obdobně je možné, aby komponenty využívané k prezentaci nebo manipulaci s částmi dokumentů zprostředkovávaly povědomí o jeho kontextu.

Komponentu tvoří popis jejího rozhraní, včetně popisu zdrojů dat, ze kterých dokáže čerpat a dále kód pro její zobrazení. Komponenta je totiž v pohledu navázána na sadu datových zdrojů, což jsou fakticky procedury uložené v databázi. Popis rozhraní komponenty umožňuje provést automatickou kontrolu shody s rozhraním procedur při jejím využití v politice.

### 3.1. Příklady typů komponent

#### 3.1.1. Mapy

Mapové komponenty jsou tvořeny mapovým oknem, sloužím k zobrazování podkladových map a tematických vrstev, navigaci v mapě a zakresu nové datové vrstvy. Jejich funkce se liší dle konkrétního účelu a použití. Některé mapové komponenty mohou sloužit například pro účely zakresu zájmového území, jiné pak pouze k zobrazení vybraných dat v mapě, další pak k práci s body zájmu.

Z důvodu zachování otevřenosti systému je doporučeno mapové komponenty implementovat pomocí nástrojů [OpenLayers](#).

#### Vlastnosti mapového okna

Mapové okno podporuje zobrazení mapových vrstev formou [WMS](#), které je možné zapnout buď jako mapu základní, nebo tematickou (lze zapnout více vrstev najednou). Systém umožňuje napojit libovolná data publikovaná mapovými službami dle standardů [Open Geospatial Consortium](#). Pro účely IPR Praha se využijí mapy a tematické vrstvy z datového skladu IPR Praha.

Okno dále obsahuje základní navigační prvky pro pohyb v mapě, jako například změnu měřítka, návrat do výchozího pohledu a funkci pro vyhledání adresy či parcely.

#### Vyhledávání

Vyhledání oblasti umožňují navigační prvky v okně.

Vyhledání podle adresy je řešeno fulltextovým vyhledáváním v rejstříku ulic a čísel orientačních či popisných. Hledání parcely je řešeno vyhledáním katastrálního území ze seznamu a zadáním čísla parcely (je možné zadat pouze část před lomítkem). Po vyhledání se mapa přiblíží a vycentruje na nalezenou adresu nebo parcelu.

Při nálezu více prvků (adresní místa v ulici, parcely se shodným kořenovým číslem) jsou prvky zobrazeny v mapě a v seznamu, po výběru konkrétního prvku se mapa přiblíží a vystředí na konkrétní místo v mapě.

### **Tvorba nové vektorové vrstvy**

V rámci mapového okna je možné vytvářet novou vektorovou vrstvu, která se stane součástí dokumentu s využitím formátu [GeoJSON](#). Vrstva může být vytvořena zadáním bodu, linie, obdélníku či obecného polygonu v rámci mapového okna a dále převzetím hranic vybraných (nalezených) územních jednotek (katastrální území, městská část) a hranic parcel.

### **3.1.2. Číselníky**

Dalším typem komponent jsou tak zvané číselníky. V jiných systémech se využívá zvláštních tabulek, které spravuje správce systému, které se využívají jako zdroj voleb pro určité vlastnosti dokumentů.

SpIS počítá s potřebou spravovat životní cyklus položek číselníku a proto tento mechanismus odděleně nezavádí. Pokud určitý dokument obsahuje vlastnost, která nabývá hodnoty z určité množiny, využije se prostý odkaz na jiný dokument patřičného typu.

### **3.1.3. Diagramy**

Řada pohledů využije interaktivní grafické komponenty jakými jsou běžné typy grafů, ale i kartogramy, kartodiagramy a časové osy. Pro všechny tyto druhy komponent platí, že je možné přepínat zahrnutí hodnot dle předem nastavených dimenzí.

### **3.1.4. Tabulky**

V tabulkových zobrazeních je možné omezit zobrazované řádky na takové, které hodnotami v určitých sloupcích vyhovují nastavenému filtru. Je možné použít jak výčet hodnot, tak i vzory včetně těch regulárních. Je možné hledat i fulltextově. Řádky je také možné řadit dle libovolného počtu sloupců.

Pro určené sloupce je v patičce tabulky přítomný součet všech hodnot v daném sloupci a dále také součet hodnot, které jsou aktuálně viditelné.

### **3.1.5. Speciální**

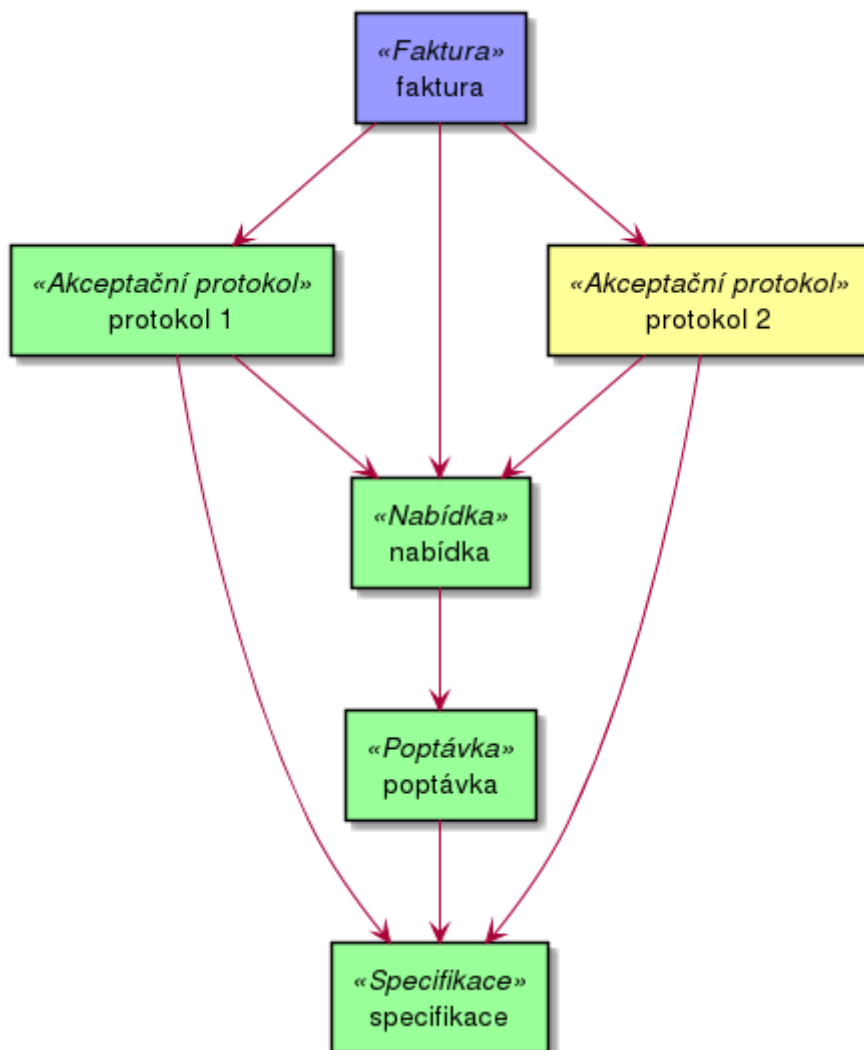
Pro přípravu rozpočtových dokumentů existuje několik zvláštních komponent, které umožňují sestavovat rozpočet na základě podkladů (závislostí) a ručních oprav. Součástí těchto specializovaných komponent může být souhrnná tabulka nebo jeden či více diagramů, které poskytnou rychlou orientaci ve výsledné finanční situaci.

V rámci implementační analýzy budou upřesněny požadavky na vývoj speciálních komponent.

## 4. Případy

Každý případ je **hierarchickou pracovní strukturou**. Práce na něm spočívá v postupném vytváření a schvalování dokumentů, které dokládají provedené úkony. Dokumenty mohou být v různých stavech a na základě svých závislostí tvoří acyklický orientovaný graf.

*Příklad 1. Hierarchická pracovní struktura (šipky značí závislost)*



V příkladu je **protokol 2** nedokončený, ale pracovník zodpovědný za schválení faktury ho označil za provizorní. To proto, aby mohl vytvořit provizorní fakturu a mohla proběhnout platba. Pokud se zmýlil a pracovník zodpovědný za **protokol 2** tento nedokončí, mohou být vyvozeny důsledky.

### Požadované

Dokumenty prázdné, u kterých se požaduje jejich naplnění a dokončení.

### Rozpracované

Dokumenty, které ještě nejsou hotové a nelze se na ně spoléhat.

### Dokončené

Dokumenty, které jsou hotové a schválené zodpovědnou osobou. Tato osoba ručí za jejich

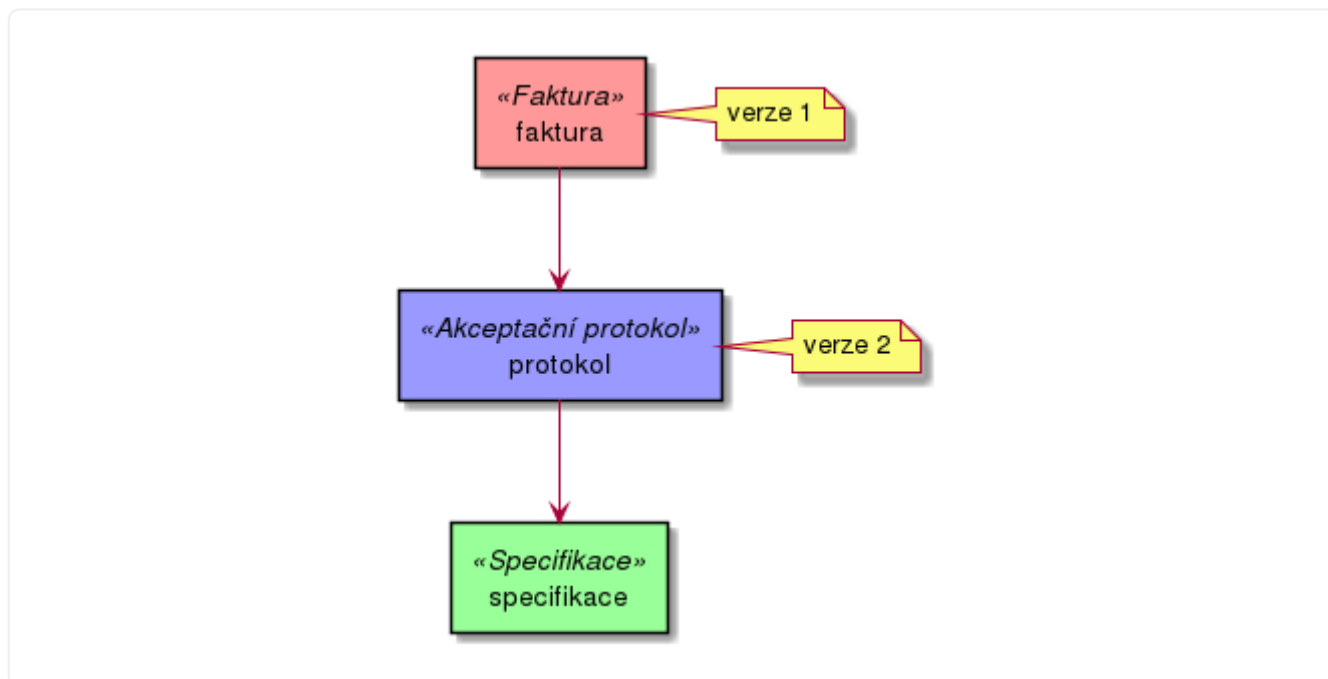
správnost a platnost. Ostatní se na ně mohou obvykle bez obav spolehnout.

### Provizorní

Dokumenty, které jsou hotové, ale nejsou schválené zodpovědnou osobou. Pokud se na ně pracovníci spoléhají, přebírají tím na sebe riziko, které jinak leží na zodpovědné osobě.

Z důvodu průkaznosti nelze dokončené dokumenty revidovat. Pokud by se tak mělo stát, je vytvořena nová verze, která ten původní nahradí v jeho roli. Pokud se na předchozí verzi jiný dokument spoléhá, je považován za neplatný.

#### Příklad 2. Zneplatnění dokumentu revizí jeho závislosti



V příkladu došlo k tomu, že byla v důsledku aktualizace dokumentu **protokol** zneplatněna aktuální verze dokumentu **faktura**. V této situaci jsou v podstatě jen dvě možnosti jak postupovat správně:

1. Vrátit se k předchozí verzi dokumentu **protokol**, kterou systém uchovává.
2. Schválit novou verzi dokumentu **protokol** a následně vytvořit novou verzi dokumentu **faktura**.

## 4.1. Typy případů

Každý případ je opatřen popisem své hierarchické struktury. Protože si jsou některé případy ve své struktuře velmi podobné, definujeme pojmenované typy případů. Tyto typy reprezentují strukturu, které se dokumenty v případě přizpůsobují.

V případě potřeby je možné typ aktivního případu změnit a dokumenty přeuspořádat do nových rolí. Stejně tak je možné do struktury vložit již existující dokumenty, pokud tak nedojde k porušení jejich vazeb na závislosti.

## 4.2. Účast na případech

Po vytvoření případu lze na základě typů dokumentů přítomných v typu případu sestavit seznam rolí, které na případě musejí spolupracovat, aby jej bylo možné dokončit. Účastníci případu mohou

v jakékoliv chvíli pozvat vybrané role (nejen) z tohoto seznamu, čímž dojde k otevření případu uživatelům v této roli. K účasti na případě je možné uživatele přizvat i jmenovitě.

Uživatelé přístupem do případu nezískávají žádná dodatečná oprávnění. Stále mohou manipulovat pouze s dokumenty, ke kterým mají oprávnění vyplývající z uživatelských rolí. V řadě případů je možné do případu zahrnout libovolné dokumenty, což ovšem nelze produktivně zohlednit pozváním všech uživatelů v organizaci.

Dalším důvodem k individuálnímu pozvání může být například zajištění procesu nákupu napříč útvary, kdy si útvar požadující nákup přizve odborníky z jiného útvaru pro spolupráci na zadání ve společném oboru působnosti. Například tehdy, když si **Oddělení pro styk s veřejností** přizve pracovníky z **Odboru ICT** do případu v oboru **Technické vybavení pro produkci multimédií**.

### 4.3. Žádost o práci na dokumentu

Pokud je k tomu uživatel oprávněn, smí v rámci případu požádat o práci na některém dokumentu. Žádat je možné buď jednoho uživatele nebo roli, která byla do případu přizvána. Úkolování nemá přímou vazbu na termín požadovaného dokončení dokumentu, který je stanoven odděleně a na základě jiného oprávnění.

Oprávnění úkolovat jiné vyplývá z oprávnění dané roli uživatele vzhledem ke konkrétnímu typu dokumentu. Dále jím však disponují i uživatelé oprávnění manipulovat libovolným způsobem se závislými dokumenty. Tito uživatelé jsou oprávnění také vytvářet nové, požadované dokumenty právě za účelem možnosti vznést požadavek na jejich vyhotovení.

Ve zkratce platí, že potřebuje-li uživatel něco od ostatních, je mu umožněno si o to říci.

### 4.4. Lhůta pro vyhotovení dokumentu

Pokud je k tomu uživatel oprávněn, smí v rámci případu nastavit nebo změnit lhůtu pro dokončení určitého dokumentu. Tato lhůta je všem viditelná a zohledňuje se například při řazení případů na stránce pracovního přehledu.

Každý typ dokumentu má možnost upravit výchozí lhůtou pro dokončení. Pokud tak neučiní, při vytvoření dokumentu daného typu není žádná lhůta nastavena. Pokud dokument neexistuje alespoň jako požadovaný, není možné mu nastavit lhůtu žádnou.

## 5. Dokumenty

Pod pojmem dokument rozumíme strukturovaná strojová data, se kterými uživatel nakládá přímo v systému. Tyto dokumenty často mívají přílohy, což jsou nestrukturovaná data (často různé formátované texty, tabulky či obrázky), které vznikají mimo systém a jsou do něj pouze vkládány.

Dokumenty mají kromě vnitřní struktury také takzvané závislosti a díky nim pak tvoří určité vnější struktury. Tyto závislosti jsou dokumentům vlastní ve všech případech, kterých jsou součástí.

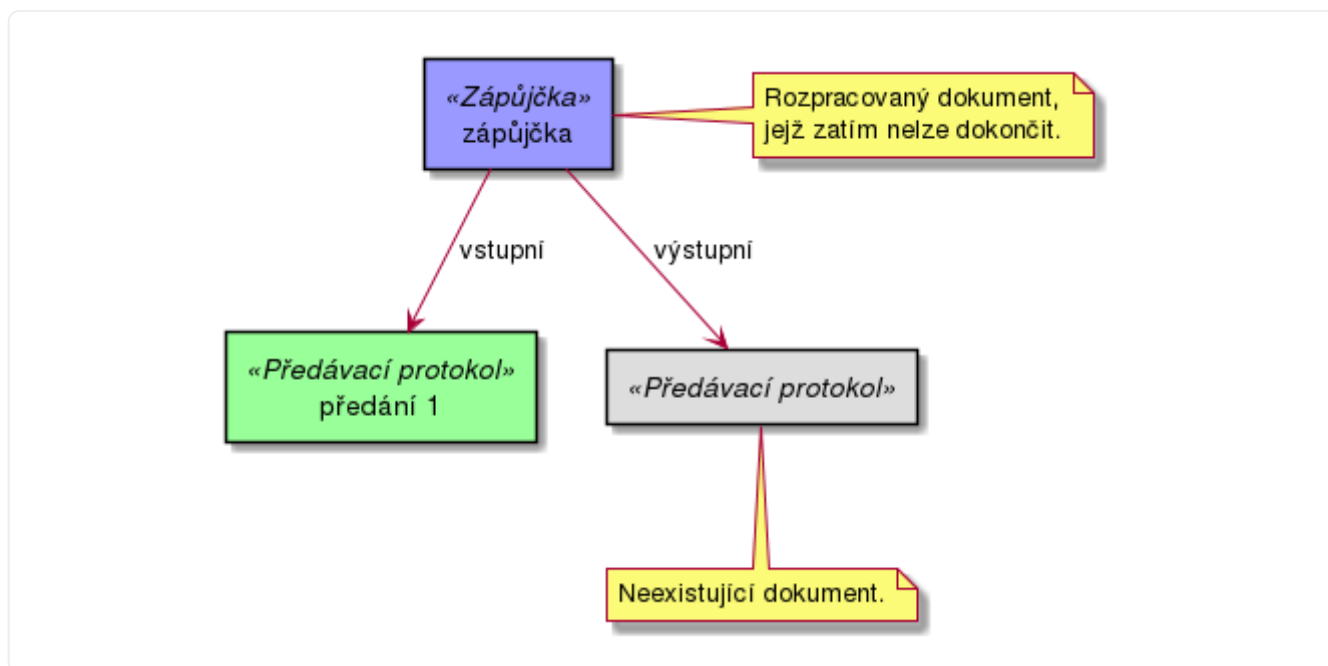
### 5.1. Typy dokumentů

Typy dokumentů shrnují vnitřní strukturu, závislosti, použité komponenty uživatelského rozhraní a další vlastnosti instancí (tedy dokumentů) do jednoznačného pojmenování. V příkladech výše jsme zatím viděli například typy nazvané **Faktura** nebo **Akceptační protokol**.

#### 5.1.1. Závislosti

Dokumenty mohou záviset pouze na jiných dokumentech, a to vždy v určitých rolích.

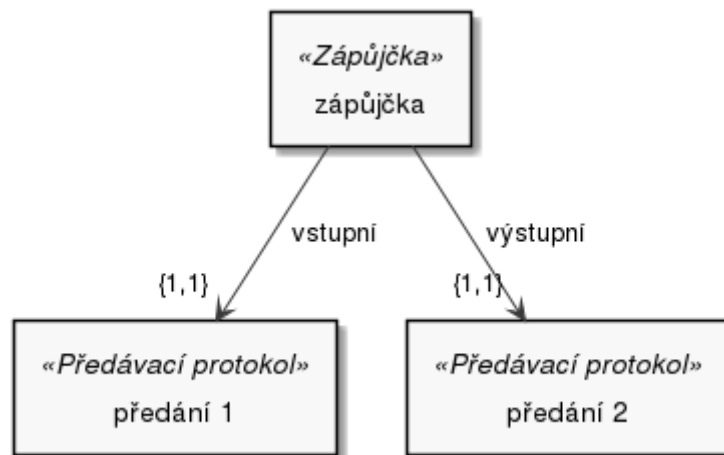
*Příklad 3. Závislosti mají konkrétní role*



V příkladu je předávací protokol použit nejprve pro dokumentaci zahájení zápůjčky věci a následně bude použit pro dokumentaci jejího navrácení. Pro vyložení závislého dokumentu **zápůjčka** je podstatné tyto dva protokoly od sebe odlišit.

Oproti instancím (dokumentům) mohou typy záviset pouze na dalších typech (nebo typových třídách, které jsou vysvětleny níže). Pro některé situace může být užitečné, aby dokument závisel na určitém počtu zcela libovolných dokumentů, bez ohledu na jejich typ.

#### Příklad 4. Typy pro předchozí příklad

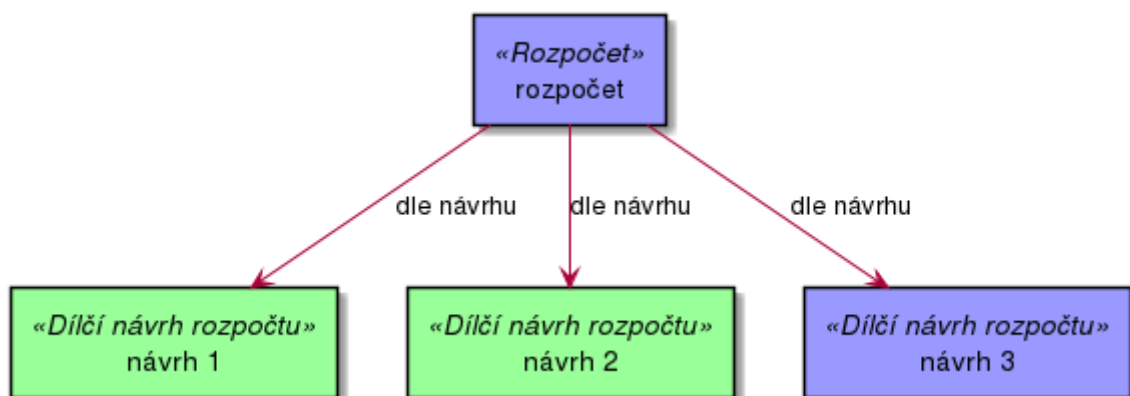


Za povšimnutí stojí uvedení dokumentů, které by měly být přítomny, ale v příkladu výše zatím schází. Také jsou zde u rolí uvedeny kvantifikátory.

### 5.1.2. Kvantifikace

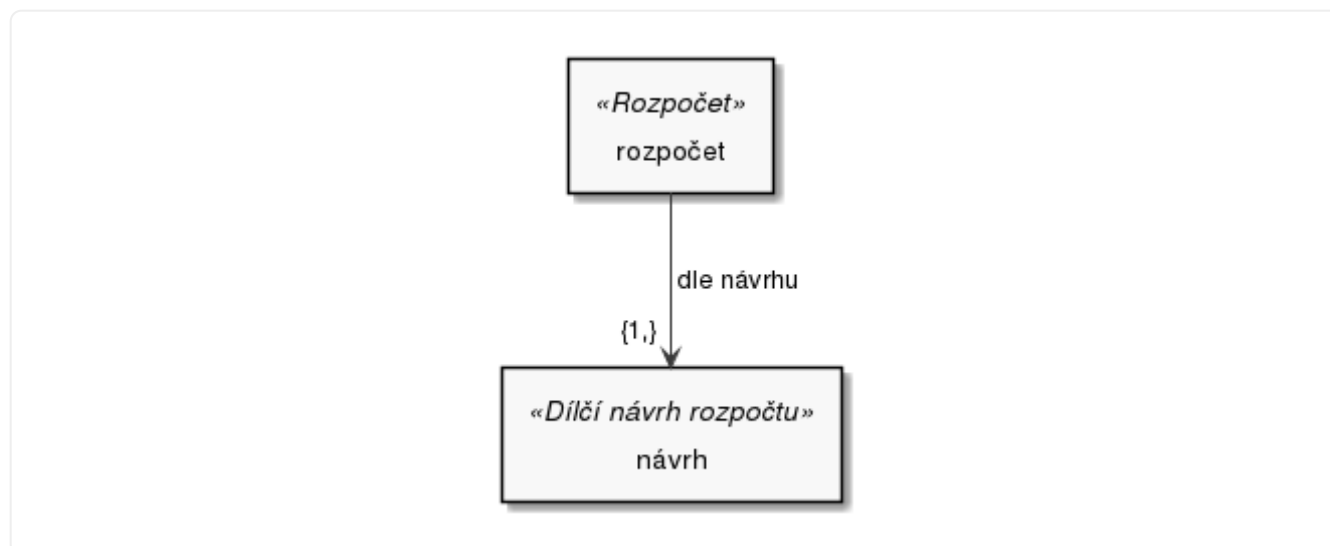
Závislosti je nutné kvantifikovat. V určité roli může vystupovat konkrétní počet dokumentů.

#### Příklad 5. Struktura se třemi dokumenty ve stejné roli



Jak budou vypadat typy pro tuto strukturu?





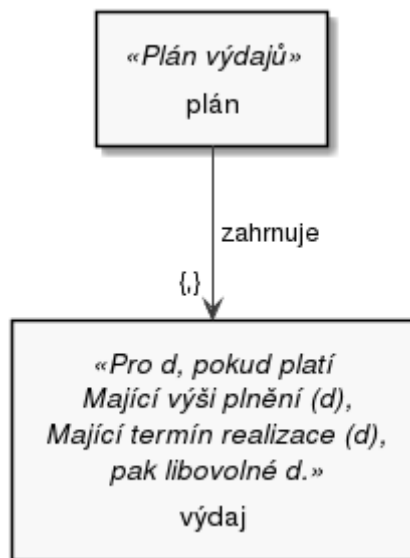
Kvantifikátor  $\{1,\}$  zde značí přítomnost alespoň jednoho dokumentu daného typu v dané roli. Je však možné specifikovat libovolný rozsah, tedy například  $\{,\}$  nebo  $\{0,\}$  pro libovolný počet dokumentů,  $\{1,3\}$  pro jeden až tři dokumenty a  $\{,\}$  nebo  $\{0,5\}$  pro až pět dokumentů.

### 5.1.3. Typové třídy

Typové třídy popisují dílčí strukturu dokumentů tak, aby mohlo dojít k použití více typů dokumentů ve stejné roli. Popisují virtuální strukturu dokumentů a upravují jejich závislosti.

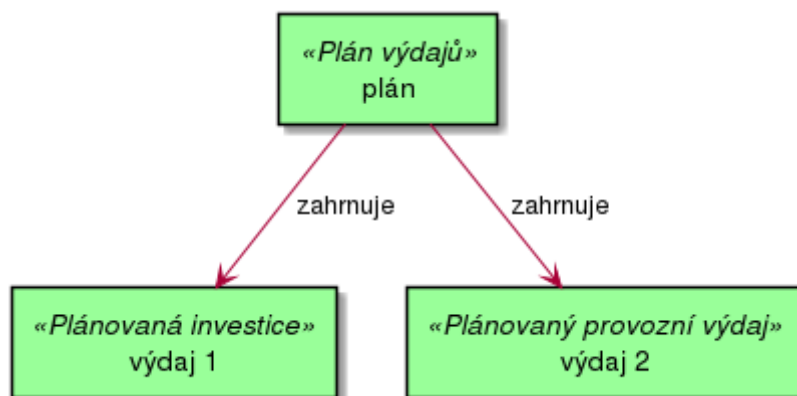
Virtuální strukturou se zde rozumí to, že definují mapování mezi sadou virtuálních vlastností připadajících dané typové třídě a vlastnostmi dokumentu typu, který danou typovou třídu implementuje. Závislé dokumenty se pak mohou odkazovat na tyto virtuální vlastnosti bez ohledu na to, kde se nachází.

Můžeme si například zavést typové třídy *Mající výši plnění (x)* a *Mající termín realizace (x)*, které popisují části schématu dokumentu, ve kterém předepisují přítomnost finanční částky s určitým významem a dále termínu, kdy dojde k nějaké realizaci. S pomocí těchto typových tříd pak můžeme popsat některé závislosti obecněji.



Pokud typy dokumentů **Plánovaná investice** a **Plánovaný provozní výdaj** implementují obě tyto typové třídy, pak je možné typy z předešlého případu splnit například následující strukturou.

Příklad 8. Implementace typů z předchozího příkladu



Termín typová třída se vyskytuje hojně v kontextu funkcionálního programování. Typové třídy tam ale obvykle neposkytují stejný druh polymorfismu jako typové třídy popsané výše. Dle tamější praxe by musely být oba výdaje v příkladu zcela stejného typu.

#### 5.1.4. Schéma

Dokument je ve své podstatě stromovou strukturou kompatibilní s formáty JSON, CBOR a YAML.

Jazyk schématu vychází z připravované normy [JSON Schema](#), kterou však rozšiřuje o další prvky. Jedná se například o popis uživatelského rozhraní pro editaci stromu, spustitelný kód v řadě významů a také o schopnost přejímat hodnoty ze závislostí.

### Příklad 9. Schéma zápůjčky (hlavička)

```
name: Lease
name[cs]: Zápůjčka

dependencies:
  lender:
    name: Lender
    name[cs]: Půjčovatel
    type: Party

  borrower:
    name: Borrower
    name[cs]: Vypůjčovatel
    type: Party

attachments:
  subject:
    name: Subject
    name[cs]: Předmět
    accept: [document, image]
    count: [0, inf]
```

V příkladu výše vidíme závislosti (**lender** a **borrower**), které dokument vyžaduje. Protože u nich není uvedena žádná kvantifikace, předpokládá se výchozí, tedy **{1,1}**. Typ obou závislostí je **Party** (smluvní strana), což bude pravděpodobně typová třída.

Dále můžeme vidět jediný typ přílohy, dokument nebo obrázek v roli **subject** (předmět). Vzhledem k neomezenému počtu se bude patrně jednat o sbírku dokladů popisujících zapůjčovanou věc a snad i další podmínky zápůjčky.

```
import:
  - from: spis.datetime
    names: [today, timedelta]

body:
  type: object
  required: [period]

  properties:
    period:
      allOf: [{ $ref: "calendar.yaml#/definitions/daterange" }]

    interface:
      component: DateRangePicker

    data:
      holidays: planning.czech_holidays

    options:
      mode: year-view

  default:
    start: !eval "today()"
    end: !eval "today() + timedelta(days=14)"

  ensure:
    - that: !eval "self.data.start < self.data.end"
      name: Period must not be zero length
      name[cs]: Období nesmí být nulové délky
```

Hned v první části příkladu vidíme import funkcí (`today` a `timedelta` z modulu `spis.datetime`). V politice je možné využívat vypočítaných hodnot a tímto způsobem je možné získat knihovní funkce. Modul `spis` a jeho součásti obsahují zabudované funkce systému.

V popisu samotného těla dokumentu vidíme, že se v tomto případě nejedná o žádnou složitou strukturu, ale pouze o doplnění příloh a závislostí o termín platnosti výpůjčky představovaný sekci `period`. Její hrubý popis je čerpán z knihovny schémat a úpravy probíhají pomocí komponenty editoru dokumentů `DateRangePicker` se zobrazením svátků.

V závěru sekce pak nalezneme určení výchozích hodnot pro nový dokument, které využívá právě knihovních funkcí importovaných výše k nastavení výpůjčky ode dnes na 14 dní a nakonec i seznam kontrol, kterým je daná komponenta podrobena a bez jejichž úspěšného absolvování není možné dokument dokončit.

Kromě kontrol připojených k jednotlivým komponentám je možné nastavit také kontroly, které platí pro dokument jako celek.

## 5.2. Komponenty

Komponenty úzce souvisí se schématem dokumentu. Právě v něm si programátor vybírá, ze kterých se prostředí pro úpravy daného typu dokumentu bude skládat.

Ve většině případů pracují komponenty se skaláry a jedná se v podstatě o dobře známé prvky uživatelských rozhraní. V jistých případech však komponenty editoru dokumentů vyžadují další informace a začínají ve svém uspořádání připomínat komponenty pohledů.

## 5.3. Verze a revize

Ačkoliv se může zdát, že dokumenty lze upravovat, ve skutečnosti tomu tak není. Při každé zdánlivé změně vzniká nová revize dokumentu, která předchozí revizi nahrazuje. Souběžně s označením revize je u každého dokumentu vedeno také označení jeho verze.

Při každém dokončení dokumentu dochází k uzamčení jeho současné verze na současné revizi. To znamená, že již není možné vytvořit revizi se shodnou verzí a musí tedy při další změně dojít k použití nového označení.

*Tabulka 1. Změny ve verzi a revizi v důsledku úprav dokumentu*

Verze	Revize	Popis změny
1	1	Dokument vytvořen
1	2	Drobná změna dokumentu
1	3	Zásadní změna dokumentu
1	4	Dokument dokončen
2	5	Drobná změna dokumentu
2	6	Dokument zrušen

Běžný postup úpravy dokumentu probíhá tak, že si uživatel otevře aktuální revizi, zanechá požadované změny a výsledek uloží jako revizi novou. Pokud dojde k souběhu a druhý uživatel (například v době kdy první stále chystá své změny) úspěšně vytvoří novou revizi, dojde při pokusu o uložení ze strany prvního uživatele ke konfliktu.

Konflikt je uživateli prezentován v plné šíři. Pokud chce se změnami pokračovat, musí se u každé komponenty rozhodnout pro některou z následujících možností:

1. Přijme změnu od konkurujícího uživatele.
2. Zamítne změnu a navrátí hodnotu do původního stavu.
3. Prosadí svou změnu, kterou případně dále přizpůsobí.

Po vypořádání všech konfliktů je uživateli opět umožněno uložit novou revizi dokumentu.

## 5.4. Podepisování

Pokud to politika daného typu dokumentu vyžaduje, může být k dokončení dokumentu zapotřebí

podpisu osob v určitých rolích. Podpisy jsou na sobě vždy nezávislé a vztahují se k dokumentu včetně všech jeho příloh.

Před zahájením podepisování je nutné dokument nejprve provizorně dokončit. Dokument je tak podepisujícímu vždy předložen v celé své, v ten okamžik již neměnitelné, podobě.

Podpisy se k dokumentu připojují odděleně, podobně jako přílohy. Pokud to politika daného typu dokumentu umožňuje, je možné kombinovat podpisy elektronické a vlastnoruční.

#### **5.4.1. Elektronické**

Elektronické podpisy jsou realizovány dle normy [XAdES](#). Dokument je pro účely podpisu převeden do své kanonické XML podoby obsahující také otisky příloh a celkový otisk je následně uživatelem podepsán. Výsledný XML fragment s podpisem je uložen v databázi.

Ověření platnosti podpisu je realizováno dle shodné normy.

#### **5.4.2. Vlastnoruční**

Vlastnoruční podpisy jsou realizovány pomocí tvrzení o podpisu, které do systému vloží podepisující nebo jiná pověřená osoba, u které je k dispozici k nahlédnutí podepsaný originál. Ten vznikne vytištěním dokumentu a jeho podpisem.

Pro vyšší míru jistoty spoléhajících se uživatelů může být tvrzení doplněno skenem podepsaného dokumentu, který si mohou uživatelé zobrazit. Tento "důkaz" však platnost dokumentu nijak nezaručuje a spoléháním se na něj se uživatel může vystavovat určitému riziku.

Pokud by byla příloha z listinné podoby autorizovaně konvertována, bylo by možné se na ni spolehnout, ale přesto (dle našeho názoru) neexistuje žádný způsob, jak bezpečně spojit podepsanou listinu s nepodepsaným elektronickým originálem za účelem automatického ověřování platnosti a návaznosti úkonů ze strany systému.

## 6. Model oprávnění

Model oprávnění umožňuje určit, ke kterým datům v systému mají jací uživatelé přístup a mohou je číst nebo měnit. Model pracuje s pravidly ve tvaru **Osoba / Oprávnění / Dokument** a nejlépe jej lze pochopit s pomocí termínů, se kterými pracuje:

### Typ dokumentu

Typ je základní a neměnnou vlastností každého dokumentu popisující jeho charakter. Dokument může být například typu **Faktura**.

### Obor působnosti

Právě za účelem kontroly přístupu musí být každý dokument zařazen do takzvaného oboru působnosti. Obor může, ale nemusí vycházet z organizačního členění. Může se tedy jednat například o obor nazvaný **Odbor ICT**. Obor působnosti může zahrnovat i podobory.

### Předmětný dokument

Jedná se o dokument určitého typu, který spadá do určitého oboru působnosti. Předmětnými dokumenty určitého oprávnění (popsaného níže) mohou být například **Faktury Odboru ICT**.

### Způsob manipulace

Způsobem manipulace rozumíme to, jak bude s předmětnými dokumenty nakládáno. Součástí popisu způsobu je i detailní vymezení toho, jak bude přistupováno k jednotlivým částem předmětných dokumentů. Příkladem způsobu manipulace může být například **Podepisovat**.

### Oprávnění

Spojení způsobu manipulace s typem dokumentů a oborů působnosti. Oprávnění představují to, co uživatelé ke své práci potřebují. Například **Podepisovat Faktury Odboru ICT**.

### Role

Soubor vícero oprávnění se souhrnným názvem. Například role s názvem **Vedoucí ICT** smí **Podepisovat Faktury Odboru ICT** a **Podepisovat Předávací protokoly Oddělení provozu**.

### Odvislá role

Odvislá role je podobná běžné roli, ale vypouští obor působnosti. Ten je doplněn při jmenování. Jako příklad můžeme uvést odvislou roli **Vedoucí (oboru)**, která smí **Podepisovat Faktury (tohoto oboru)**.

### Jmenování

Díky jmenování určité osoby do některé role dojde k aplikaci všech pravidel vyplývajících z dané role na danou osobu a tedy ke vzniku oprávnění osoby nakládat vymezeným způsobem s předmětnými dokumenty. Osobu je možné jmenovat do běžné i odvislé role.

Pokud byla osoba do role jmenována s příznakem **zastupující**, nemá běžně k dispozici práva, která z role plynou. Může se jich však chopit pomocí patřičného prvku uživatelského rozhraní. Systém eviduje v protokolu informaci o tom, zda-li by danou akci mohla osoba provést aniž by se chopila práva zástupce.

## 6.1. Uživatelské role

Role představují oprávnění uživatelů, kteří jsou součástí určitých oborů působnosti k manipulaci s množinou předmětných dokumentů spadajících do stejných nebo jiných oborů.



Pro jmenování uživatelů do rolí může sloužit například jejich příslušnost do určitých skupin, která se zjistí z vnější evidence uživatelů.

Úlohou rolí je pojmenovat skupiny oprávnění, které mají určití uživatelé, a tím umožnit jejich kontrolu a změny. Běžné role se uplatní zejména při netypických situacích, jako jsou případy mimo běžný rámec organizační struktury. Odvislé role se naopak uplatní při formulování oprávnění pro pravidelné organizační struktury.

Správce může k jednotlivým rolím připojit poučení pro jejich nabyvatele. Tito pak po jmenování do daných rolí musí toto poučení vzít na vědomí, jinak jim není umožněno se systémem dále pracovat. Stejně tak musí nabyvatelé vzít poučení na vědomí při jeho dodatečné změně.

## 6.2. Způsoby manipulace

Každý dokument se může nacházet ve třech stavech. Manipulace znamená například to, že určitý uživatel u předmětného dokumentu tento stav změní. Tedy dokument rozpracuje, označí za provizorně dokončený nebo jej dokončí. Obsah dokumentů, které jsou (byť jen provizorně) dokončené, již není možné dále měnit. Provizorní dokument však dokončit lze.

Některé typy dokumentů mohou pro dokončení vyžadovat podpisy od různých uživatelů. Způsob manipulace pro tyto typy dokumentů určuje, kteří uživatelé jsou oprávněni vytvořit které podpisové doložky.

Způsob manipulace také upravuje, se kterými částmi těla dokumentu a se kterými přílohami přesně smí uživatel nakládat. Oprávnění k částem těl dokumentů jsou rekurzivní a musí tedy podporovat nejen svolení, ale také zákazy.

Tabulka 2. Popis způsobů manipulace s tělem dokumentu

Cesta	Manipulace
/	číst klíče
/název	číst a měnit hodnotu
/zdůvodnění	číst a měnit hodnotu
/odhadovaná-cena	číst hodnotu



## 7. Upozornění na událost

V rámci systému dochází k řadě událostí. Mechanismus upozorňování na ně využívá nastavené komunikační metody (email nebo Jabber) k odeslání upozornění dotčenému uživateli. K upozornění uživatele nedochází, pokud událost, na kterou by měl být upozorněn, sám vyvolal.

Kromě okamžitého upozornění si může uživatel nastavit upozornění dávkové, které se odesílá souhrnně jednou denně. Zda-li si přeje být upozorňován okamžitě, souhrnně, nebo vůbec si může uživatel pro každý druh upozornění určit sám.

Tabulka 3. Předvolby upozorňování

Událost	Nikdy	Ihned	Souhrnně
Je požadována Vaše práce na dokumentu		✓	✓
Můžete vytvořit chybějící dokument	✓		✓
Vámi naposledy upravený dokument byl změněn	✓	✓	✓
Vámi sledovaný dokument byl změněn	✓	✓	✓
Dokument případu, kterého se účastníte, byl změněn	✓	✓	✓
Případ, kterého se účastníte, byl změněn	✓	✓	✓
Změnila se Vaše účast na případu	✓	✓	✓
Změnily se Vaše role v systému	✓	✓	✓

### 7.1. Události

#### Je požadována práce uživatele na dokumentu

Někdo uložil danému uživateli pracovat na určitém dokumentu. Upozornění obsahuje odkaz na případ, v rámci kterého k události došlo a dále identifikátor dokumentu, o který se jedná. Po zobrazení odkazované stránky je uživateli naznačeno, o který dokument se jedná.

#### Uživatel nově může vytvořit chybějící dokument

Obdobně jako v případě požadované práce, pouze k události dochází obvykle vytvořením všech potřebných závislostí určitého, zatím neexistujícího dokumentu v rámci případu.

#### Uživatелеm naposledy upravený dokument byl změněn

Někdo změnil dokument, který naposledy upravoval upozorňovaný uživatel. Tato událost se spustí jen při první změně jiným uživatelem. Pokud si uživatel přeje být informován o všech změnách daného dokumentu, musí ho začít sledovat. Odkaz je shodný s odkazem v předchozích událostech.

#### Uživatелеm sledovaný dokument byl změněn

Obdobně jako v případě předchozí události, jen je sledující uživatel upozorňován na všechny změny dokumentu.

#### Dokument případu, které se uživatel účastní, byl změněn

Velmi široká volba pro upozorňování, díky které bude uživatel upozorněn při libovolné změně

libovolného dokumentu ve všech případech, kterých se účastní.

#### **Případ, kterého se uživatel účastní, byl změněn**

Touto událostí se myslí takové změny případu, které se netýkají konkrétních dokumentů ani účasti uživatelů nebo rolí na případu. Jedná se tedy především o změny metadat případu. Součástí upozornění je odkaz na případ.

#### **Změnila se účast uživatele na případu**

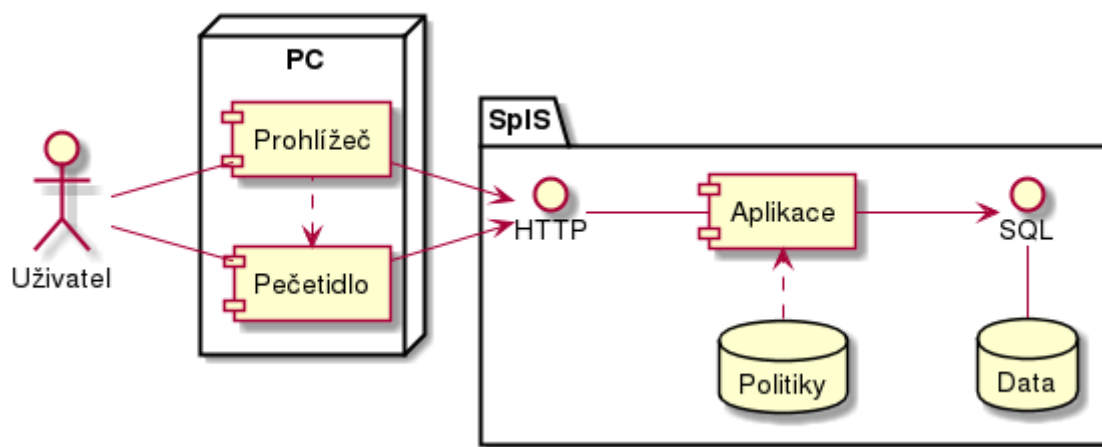
Někdo uživatele pozval do nového případu. Součástí upozornění je kromě odkazu na případ i jméno uživatele, který jej pozval a role, ve kterých v případě figuruje.

#### **Změnily se role uživatele v systému**

Správce systému udělil nebo odebral uživateli role. Součástí upozornění jsou i informace o tom, jaká role mu byly přiděleny a jaké odebrány, případně ve kterých oblastech působnosti.

## 8. Implementační detaily

### 8.1. Architektura



Uživatelé budou se SpISem komunikovat pomocí webového rozhraní spuštěném v běžném webovém prohlížeči. Pro účely elektronického podepisování bude na jejich zařízeních nasazena další aplikace, ve schématu označená jako *Pečetidlo*.

Jak naznačuje schéma výše, *Aplikace* samotná je běhovým prostředím pro určité *Politiky*. Ty vytváří programátoři a popisují jimi nejen strukturu dokumentů, které se ve SpISu vyskytují, ale také jejich vzájemné vazby v případech, použité prvky uživatelského rozhraní a další rysy systému dané organizace.

### 8.2. Datová vrstva

#### 8.2.1. PostgreSQL

Úlohu hlavního databázového systému plní otevřené řešení *PostgreSQL*. Tím se zajistí dobrá integrace do stávajícího prostředí objednatelů, dlouhodobá udržitelnost a nedojde k omezení možnosti řešení dále šířit, protože licenční podmínky *PostgreSQL* v něm nebrání.

#### 8.2.2. Dokumenty

Ústřední skupinou objektů v databázi jsou dokumenty. Tyto dokumenty, vyjádřené hodnotou typu *jsonb* (binární JSON) jsou doplněny o řadu metadat v rozsahu vyplývajícím z ostatních požadavků. Dokumenty všech typů budou uloženy v jediné tabulce a bude u nich při ukládání docházet alespoň k základní kontrole dle schématu odpovídajícímu jejich typu.

Obsah dokumentů by nemělo být vůbec možné měnit, pouze vytvářet nové revize a upravovat jejich stav směrem k dokončení a podepsání.

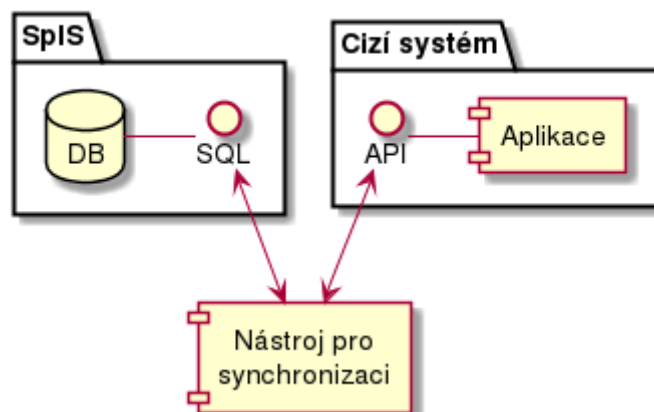
#### 8.2.3. Zdroje dat

Relační datové struktury, ze kterých uložené procedury čerpají data pro pohledy, se mohou do

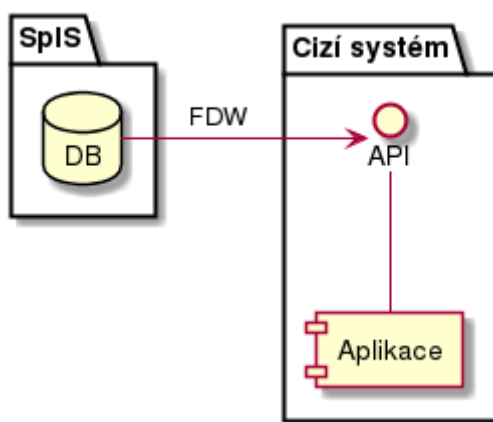
systému dostat dvěma způsoby. Může jít o data z jiných systémů nebo rozbory dokumentů.

### 8.2.3.1. Data z cizích systémů

Data z cizích systémů se do systému zanáší zpravidla pomocí zvláštních nástrojů vyvinutých na míru jednak cizímu systému a dále dle potřeb konkrétních procesů vedených ve SpISu.



V některých případech je možné využít také databázový mechanismus zapouzdření cizích dat (Foreign Data Wrappers). Díky tomu se mohou cizí data jevit jako součást vlastní databáze a je možné s nimi pracovat jako s pohledy. Tento druh provázání se hodí zejména pro čerpání z rozsáhlých databází, které není výhodné přenášet celé.



### 8.2.3.2. Data z dokumentů

Žádná existující databáze zatím neobsahuje funkci automatického, okamžitého odvozování tabulek a reakci na změny takto odvozených řádků. Pro naplnění relačních datových struktur z dokumentů se tak využívají spouště, které data z dokumentů přenášejí do tabulek ihned po vytvoření nové revize nebo změně stavu dokumentu.

Různé přehledy mohou mít různé požadavky na platnost revizí dokumentů, ze kterých čerpají. Mohou vyžadovat revizi nejnovější, poslední podepsanou, obě zároveň nebo jejich spojení. Vhodná soustava pohledů usnadní jejich vytváření.

## 8.3. Uživatelské rozhraní

Uživatelské rozhraní je postaveno na bázi projektu [PatternFly](#) a využívá v maximální míře příkladové vzory uveřejněných na webu tohoto projektu.

Pro každou část systému bude požadované chování upřesněno v rámci implementační analýzy tak, aby bylo jeho ovládání intuitivní a jednoduché. Z této analýzy a komunikace s uživateli vyplynou například požadavky na komponenty, ale také na podobu procesních modelů.