

**Katedra počítačov a informatiky
Fakulta elektrotechniky a informatiky
Technická univerzita v Košiciach**



WorkBridge

Cloudové systémy

Špecifikácia a dizajn

2025/2026

Bc. Štefan Jaklovský

Bc. Jakub Janok

Bc. Kristián Kandra

Bc. Matej Maňkoš

Bc. Kamil Škurla

Obsah

1	Úvod	1
1.1	Cieľ projektu	1
1.2	Kontext a motivácia	2
1.3	Stručný prehľad riešenia	3
2	Požiadavky a funkčný návrh	4
2.1	Aktéri, funkčné požiadavky a use cases	4
2.1.1	Príklad rozdelenia do rolí	4
2.1.2	Funkčné požiadavky	5
2.1.3	Kľúčové use cases	7
2.1.4	Zhrnutie funkčného návrhu	8
2.2	Dáta, integrácie a prevádzkové požiadavky	9
2.2.1	Dátový model a hlavné entity	9
2.2.2	Charakter dát a ich uloženie	10
2.2.3	Prevádzkové požiadavky	10
2.2.4	Záver kapitoly	12
3	Architektúra systému	13
3.1	Celková architektúra a distribúcia mikroservisov	13
3.2	Komunikácia služieb a konzistencia dát	17
3.3	Backend služby, API a dátový model	19
3.3.1	Prehľad backendových služieb	19
3.3.2	Návrh API rozhraní	20
3.3.3	Dátový model	21
3.4	Frontend návrh a interakcia s backendom	24
4	Nasadenie a technológie	31
4.1	Cloud platforma, deployment a použité služby	31
4.2	Dopad nasadenia do produkcie	33
4.3	Použité technológie a modelovacie nástroje	35

Zoznam obrázkov

3.1	Vysokourovňová architektúra	15
3.2	Rozdelenie na mikroservisy	16
3.3	Návrh dátovej vrstvy systému a rozdelenie údajov medzi relačnú da- tabázu a objektové úložisko	22
3.4	Konceptuálny a databázový model hlavných entít systému	23
3.5	Postupnosť wireframov scenára UC1	26
3.6	Postupnosť wireframov scenára UC2	27
3.7	Postupnosť wireframov scenára UC3	28
3.8	Postupnosť wireframov scenára UC4	29
3.9	Postupnosť wireframov scenára UC5	30

1 Úvod

1.1 Cieľ projektu

Cieľom projektu je navrhnúť a implementovať cloudovú aplikáciu *WorkBridge*, ktorá bude slúžiť firmám na efektívnu správu pracovníkov nasadených na pracovných projektoch. Navrhované riešenie má poskytnúť centralizovanú platformu, prostredníctvom ktorej bude možné spravovať firemné profily, evidovať zamestnancov, priradovať ich na konkrétne projekty a sprístupniť im všetky dôležité informácie súvisiace s výkonom práce.

Súčasťou cieľa projektu je aj digitalizácia vybraných administratívnych procesov, ktoré sú v mnohých firmách stále riešené neefektívne, napríklad pomocou e-mailovej komunikácie, papierových dokumentov alebo rôznych neprepojených tabuliek a systémov. Aplikácia má preto umožniť pracovníkom nahrávať faktúry, výkazy odpracovaných hodín a ďalšie dokumenty priamo do systému, zatiaľ čo firma bude môcť tieto podklady kontrolovať, schvaľovať a spravovať na jednom mieste.

Z technického hľadiska je cieľom projektu vytvoriť riešenie založené na princípoch modernej cloudovej architektúry, s dôrazom na škálovateľnosť, dostupnosť a oddelenie jednotlivých funkčných častí systému. Projekt sa zároveň zameriava na návrh vhodného dátového modelu, komunikačných rozhraní medzi službami a využitie cloudových služieb na ukladanie dát, správu dokumentov a autentifikáciu používateľov.

1.2 Kontext a motivácia

Mnohé firmy pôsobiace v oblastiach ako elektroinštalácie, montážne práce, stavebníctvo alebo technické služby vysielajú svojich pracovníkov na zákazky mimo sídla firmy, často aj do zahraničia. V takomto prostredí vzniká potreba efektívne riadiť nielen samotné projekty, ale aj pracovníkov, ich dokumentáciu, pracovné výkazy, fakturáciu a organizačné informácie, ako je napríklad ubytovanie alebo kontakty na kolegov.

V praxi sú tieto procesy často rozdelené medzi viacero nástrojov. Komunikácia prebieha cez e-mail alebo mobilné aplikácie, pracovné hodiny sa evidujú v tabuľkách alebo na papieri, faktúry sa posielajú ako prílohy v správach a informácie o projekte či ubytovaní sú roztrúsené medzi rôznymi osobami alebo systémami. Takýto spôsob práce je neprehľadný, náchylný na chyby a zároveň zvyšuje administratívnu záťaž pre firmu aj pracovníkov.

Motiváciou projektu *WorkBridge* je vytvoriť jednotné riešenie, ktoré tieto procesy prepojí do jednej centrálnej cloudovej platformy. Firma získa prehľad o svojich pracovníkoch, projektoch a dokumentoch, zatiaľ čo pracovníci budú mať jednoduchý prístup k informáciám, ktoré pri práci reálne potrebujú. Dôležitým prínosom je aj zvýšenie transparentnosti, keďže pracovník bude môcť sledovať stav nahraných faktúr alebo potvrdenie odpracovaných hodín bez potreby dodatočnej komunikácie cez e-mail alebo telefón.

Projekt je motivovaný aj snahou navrhnúť riešenie, ktoré bude zodpovedať reálnym požiadavkám menších a stredných firiem. Ide o interný alebo firemný systém na operatívne riadenie pracovníkov a súvisiacich procesov. Práve zameranie na zahraničné pracovné projekty, prepojenie administratívy s projektovým riadením a dôraz na jednoduchý prístup pre pracovníkov predstavujú hlavnú pridanú hodnotu navrhovaného riešenia.

1.3 Stručný prehľad riešenia

Navrhované riešenie *WorkBridge* bude pozostávať z webovej cloudovej aplikácie s oddeleným frontendom a backendom, pričom backend bude rozdelený do viacerých samostatných služieb podľa hlavných funkčných oblastí systému. Medzi tieto oblasti patrí autentifikácia a správa používateľov, evidencia firiem a pracovníkov, správa projektov, evidencia odpracovaných hodín, správa faktúr, nahrávanie dokumentov a notifikačný mechanizmus.

Používateľmi systému budú najmä dve hlavné skupiny. Prvou skupinou budú zástupcovia firmy, ktorí budú spravovať firemný profil, vytvárať účty pracovníkov, zakladať projekty, pridelať pracovníkov na projekty a kontrolovať nahrané dokumenty. Druhou skupinou budú samotní pracovníci, ktorí sa po prihlásení do systému dostanú k prehľadu svojho projektu, informáciám o kolegoch, kontaktným údajom, údajom o ubytovaní a k funkciám na nahrávanie faktúr a výkazov odpracovaných hodín.

Architektúra systému bude navrhnutá ako mikroservisová, aby bolo možné oddeliť jednotlivé časti aplikácie podľa ich zodpovednosti a zabezpečiť ich jednoduchšiu údržbu a prípadné nezávislé rozširovanie. Hlavné business dáta budú ukladané v relačnej databáze, zatiaľ čo súbory, ako sú PDF faktúry alebo podpísané dokumenty, budú ukladané do objektového úložiska. Komunikácia medzi vybranými službami bude kombinovať synchronné API volania a asynchrónne spracovanie udalostí, napríklad pri notifikáciách alebo schvaľovacích procesoch.

Výsledkom projektu má byť návrh a realizácia systému, ktorý ukáže, ako možno pomocou cloudových technológií efektívne digitalizovať interné firemné procesy spojené s riadením pracovníkov na projektoch. Riešenie má zároveň demonštrovať praktické využitie moderných princípov návrhu cloudových aplikácií, ako sú viacvrstvomá architektúra, mikroservisy, centralizovaná správa dokumentov a podpora viacerých tenantov v jednom systéme.

2 Požiadavky a funkčný návrh

2.1 Aktéri, funkčné požiadavky a use cases

Navrhovaný systém *WorkBridge* je určený na podporu interných procesov firiem, ktoré riadia pracovníkov nasadených na projektoch, často aj mimo domovskej krajiny. Z funkčného hľadiska ide o viacpoužívateľský informačný systém s oddelenými rolami a prístupovými právami, pričom jednotlivé skupiny používateľov prístupujú k rozdielnym častiam funkcionality podľa svojich kompetencií. Namiesto pevne definovaných používateľských rolí systém využíva konfigurovateľný model rolí a oprávnení, ktorý umožňuje prispôbiť správanie systému konkrétnym potrebám každej firmy. Každá firma má možnosť vytvárať vlastné roly a priradovať im oprávnenia podľa svojich potrieb.

2.1.1 Príklad rozdelenia do rolí

Na základe analýzy domény, rozdelenie do rolí môže vyzeráť nasledovne:

- **Hlavný administrátor firmy** – zakladá firemný účet, spravuje základné údaje o firme, vytvára používateľské účty manažérov a pracovníkov a má najvyššiu úroveň oprávnení v rámci organizácie. Zároveň môže vykonávať aj operácie súvisiace s projektmi, pracovníkmi a schvaľovaním dokumentov.
- **Manažér / poverená administratívna osoba** – spravuje operatívnu agendu firmy, napríklad vytvára projekty, eviduje a upravuje pracovníkov, priraduje ich na projekty, kontroluje odpracované hodiny a spracúva faktúry. Touto rolou môže byť napríklad projektový manažér alebo ekonomický pracovník firmy.
- **Pracovník** – prístupuje k informáciám o svojom projekte, kolegoch, kontaktných údajoch, ubytovaní a odosiela podklady súvisiace s výkonom práce, najmä výkazy odpracovaných hodín a údaje potrebné na vytvorenie faktúry.

V prvej verzii riešenia sa predpokladá, že hlavný administrátor a manažér budú reprezentovať samostatné roly s rozdielnou úrovňou oprávnení, pričom časť funkcií manažéra môže byť dostupná aj hlavnému administrátorovi. Oprávnenia je možné kombinovať a prispôbovať podľa potrieb firmy.

2.1.2 Funkčné požiadavky

Funkčné požiadavky boli odvodené z navrhovaného spôsobu používania systému a z identifikovaných potrieb cieľovej skupiny. Systém musí podporovať nasledujúce hlavné oblasti funkcionality.

Správa používateľov a autentifikácia

Systém musí umožniť:

- registráciu a správu firemného účtu,
- vytváranie a správu používateľských účtov manažérov a pracovníkov,
- bezpečné prihlasovanie používateľov do systému,
- správu rolí a oprávnení podľa typu používateľa,
- obmedzenie prístupu používateľa len na dáta patriace jeho firme.

Správa firiem a pracovníkov

Systém musí umožniť:

- evidenciu základných údajov o firme,
- vytváranie a úpravu profilov pracovníkov,
- uchovávanie kontaktných a pracovných údajov pracovníka,
- správu väzieb medzi firmou a pracovníkmi,
- evidenciu aktívneho alebo neaktívneho stavu pracovníka.

Správa projektov

Systém musí umožniť:

- vytváranie projektov a evidenciu ich základných parametrov,
- priradovanie pracovníkov ku konkrétnym projektom,

- evidenciu lokality projektu a organizačných údajov,
- sprístupnenie detailu projektu pracovníkom, ktorí sú na projekt priradení,
- zobrazenie zoznamu kolegov priradených na rovnaký projekt.

Správa ubytovania a organizačných informácií

System musí umožniť:

- evidenciu údajov o ubytovaní súvisiacom s projektom alebo pracovníkom,
- sprístupnenie adresy, kontaktu a ďalších poznámok pracovníkom,
- priradenie údajov o ubytovaní ku konkrétnemu projektu alebo skupine pracovníkov.

Evidencia odpracovaných hodín

System musí umožniť:

- odosielanie výkazov odpracovaných hodín pracovníkom,
- nahrávanie príloh k výkazu, napríklad podpísaných dokumentov,
- evidenciu stavu výkazu (napr. odoslaný, schválený, zamietnutý),
- kontrolu a schvaľovanie výkazov oprávnenou osobou na strane firmy,
- spätné zobrazenie výsledku spracovania pracovníkovi.

Správa faktúr a dokumentov

System musí umožniť:

- automatizované vytvorenie faktúry na základe údajov vyplnených pracovníkom,
- evidenciu základných údajov o faktúre,
- generovanie jednotného PDF výstupu faktúry podľa firemne definovaného formátu,
- uchovávanie príloh vo forme dokumentov,
- spracovanie schvaľovacieho procesu faktúr,
- zobrazenie aktuálneho stavu faktúry pracovníkovi aj firme.

Notifikácie a spätná väzba

System musí umožniť:

- informovať používateľov o zmene stavu vybraných procesov,
- generovať notifikácie pri schválení alebo zamietnutí dokumentov,
- upozorniť používateľa na priradenie na projekt alebo zmenu údajov.

2.1.3 Kľúčové use cases

UC1 – Vytvorenie pracovníka a jeho priradenie na projekt

Primárnym aktérom je hlavný administrátor firmy alebo manažér. Používateľ vytvorí profil pracovníka, doplní jeho identifikačné a kontaktné údaje a následne pracovníka priradí na konkrétny projekt. Výsledkom je, že pracovník po prihlásení vidí informácie o svojom pridelení, projekte a kolegoch.

Tento use case pokrýva základné entity systému a predstavuje jeden z hlavných vstupných scenárov pri každodennom používaní aplikácie.

UC2 – Zobrazenie detailu projektu pracovníkom

Primárnym aktérom je pracovník. Po prihlásení do systému si otvorí detail svojho aktuálneho projektu. System mu zobrazí názov projektu, lokalitu, zoznam kolegov, kontaktné údaje a prípadné organizačné informácie, napríklad údaje o ubytovaní.

Tento use case reprezentuje hlavnú používateľskú hodnotu systému z pohľadu pracovníka, pretože centralizuje roztrúsené informácie do jedného rozhrania.

UC3 – Odoslanie odpracovaných hodín a vytvorenie faktúry

Primárnym aktérom je pracovník. Používateľ vyplní výkaz odpracovaných hodín za zvolené obdobie, zadá počet hodín a v prípade potreby priloží podpísaný dokument. V rámci rovnakého procesu zároveň doplní údaje potrebné na vystavenie faktúry. System na základe týchto údajov automatizovane vytvorí faktúru v jednotnom PDF formáte, zaeviduje ju a nastaví počiatočný stav spracovania.

Pracovník má následne možnosť priebežne sledovať stav spracovania oboch podkladov.

Tento use case prepája dve úzko súvisiace časti domény, a to pracovný výkaz a fakturáciu, ktoré v praxi tvoria jeden nadväzujúci administratívny proces.

UC4 – Spracovanie a schválenie faktúry a výkazu na strane firmy

Primárnym aktérom je manažér alebo hlavný administrátor firmy. Po doručení podkladov od pracovníka používateľ otvorí detail výkazu a vygenerovanej faktúry, skontroluje zadané údaje, prípadné prílohy a rozhodne o schválení alebo zamietnutí. V prípade zamietnutia môže doplniť poznámku alebo dôvod zamietnutia.

Výsledkom use case je aktualizácia stavov dokumentov v systéme a sprístupnenie tejto informácie pracovníkovi. Súčasne môže byť vyvolaná notifikácia o zmene stavu spracovania.

UC5 – Vytvorenie projektu a priradenie pracovníkov

Primárnym aktérom je hlavný administrátor firmy alebo manažér. Používateľ vytvorí nový projekt, vyplní jeho základné údaje, lokalitu a termíny, prípadne doplní aj informácie o ubytovaní. V rámci rovnakého procesu následne vyberie a priradí pracovníkov, ktorí budú na projekte pracovať.

Výsledkom use case je založenie nového projektu v systéme a vytvorenie väzieb medzi projektom, priradenými pracovníkmi a súvisiacimi organizačnými údajmi. Tento use case predstavuje jeden z hlavných administratívnych scenárov na strane firmy, keďže prepája správu projektov, pracovníkov a ubytovania do jedného pracovného toku.

2.1.4 Zhrnutie funkčného návrhu

Z pohľadu funkčného návrhu je systém koncipovaný ako centralizovaná platforma, ktorá pokrýva hlavné pracovné a administratívne procesy medzi firmou a pracovníkmi. Dôraz sa kladie na oddelenie používateľských rolí, transparentnosť stavov dokumentov a schopnosť sprístupniť pracovníkovi všetky dôležité údaje o projekte v jednom prostredí.

Funkčný návrh zároveň vytvára základ pre následný architektonický návrh založený na samostatných službách pre autentifikáciu, správu používateľov, projekty, dokumenty, faktúry, výkazy práce a notifikácie.

2.2 Dáta, integrácie a prevádzkové požiadavky

Návrh dátovej vrstvy a prevádzkových požiadaviek vychádza z charakteru riešeného problému. Systém *WorkBridge* pracuje s kombináciou štruktúrovaných business dát, dokumentových príloh a udalostí vznikajúcich pri spracovaní používateľských operácií. Z tohto dôvodu je potrebné rozlišovať medzi transakčnými dátami, dokumentmi a integračnými alebo notifikačnými dátami.

2.2.1 Dátový model a hlavné entity

Základ dátového modelu tvoria relačné entity reprezentujúce hlavné doménové objekty systému. Medzi kľúčové entity patria:

- **Company** – reprezentuje firmu ako tenant systému,
- **UserAccount** – reprezentuje autentifikačný účet používateľa,
- **EmployeeProfile** – reprezentuje pracovný profil pracovníka,
- **Project** – reprezentuje pracovný projekt alebo zákazku,
- **ProjectAssignment** – reprezentuje priradenie pracovníka na projekt,
- **Accommodation** – reprezentuje údaje o ubytovaní,
- **Timesheet** – reprezentuje výkaz odpracovaných hodín,
- **Invoice** – reprezentuje faktúru nahranú alebo vygenerovanú v systéme,
- **DocumentMetadata** – reprezentuje metadata dokumentu uloženého mimo relačnej databázy,
- **Notification** – reprezentuje systémové upozornenie pre používateľa,
- **ApprovalHistory** – reprezentuje históriu rozhodnutí nad timesheetmi alebo faktúrami.

Dátový model musí zohľadňovať viacnájomnícky charakter riešenia. Každá hlavná entita preto musí byť naviazaná na firmu, resp. tenant kontext, aby bolo možné zabezpečiť logickú izoláciu dát medzi jednotlivými organizáciami. V praxi to znamená, že entity ako pracovník, projekt, faktúra alebo výkaz hodín budú obsahovať identifikátor firmy, ku ktorej patria.

2.2.2 Charakter dát a ich uloženie

Systém pracuje s dvoma základnými typmi dát:

1. **Štruktúrované transakčné dáta** – údaje o firmách, používateľoch, projektoch, priradeniach, faktúrach, timesheetoch a stavoch schvaľovania.
2. **Neštruktúrované alebo polostruktúrované dokumenty** – PDF faktúry, podpísané výkazy práce, prílohy a ďalšie nahrané súbory.

Pre transakčné dáta je vhodné použiť relačnú databázu, keďže medzi entitami existujú jasné vzťahy a zároveň je potrebné podporiť konzistentné aktualizácie, filtrovanie, reporting a auditovateľnosť. Relačný model je vhodný najmä vzhľadom na väzby medzi pracovníkmi, projektmi, dokumentmi a schvaľovacími procesmi.

Dokumenty by nemali byť ukladané priamo do relačnej databázy. Vhodnejším prístupom je ich uloženie do objektového úložiska, pričom v databáze budú uložené iba ich metadata, napríklad názov súboru, typ dokumentu, umiestnenie súboru, čas nahrania, vlastník a väzba na doménovú entitu.

V prípade faktúr systém nepredpokladá výhradne nahrávanie hotového dokumentu od používateľa, ale aj automatizované generovanie PDF faktúry na základe údajov zadaných pracovníkom. Tým sa zabezpečí jednotná štruktúra dokumentov v rámci jednej firmy a zároveň sa zjednoduší následné spracovanie a kontrola.

2.2.3 Prevádzkové požiadavky

Z pohľadu prevádzky musí systém spĺňať viacero základných vlastností typických pre modernú cloudovú aplikáciu.

Dostupnosť

Systém musí byť dostupný pre používateľov z rôznych lokalít a zariadení. Keďže pracovníci môžu pristupovať k aplikácii aj zo zahraničia, požaduje sa stabilná dostupnosť webového rozhrania a backendových služieb cez internet.

Škálovateľnosť

Riešenie musí byť navrhnuté tak, aby bolo možné postupne zvyšovať kapacitu systému podľa počtu firiem, používateľov a nahrávaných dokumentov. Škálovanie sa týka najmä aplikačnej vrstvy a úložiska dokumentov.

Bezpečnosť

Keďže systém spracúva firemné a osobné údaje, je nevyhnutné zabezpečiť:

- autentifikáciu a autorizáciu používateľov,
- izoláciu dát medzi jednotlivými firmami,
- ochranu nahratých dokumentov,
- šifrovanie komunikácie medzi klientom a backendom,
- auditovateľnosť kritických zmien a schvaľovacích rozhodnutí.

Spôľahlivosť a konzistencia

Zápisy údajov o faktúrach, timesheetoch a stavoch schvaľovania musia byť spracované spoľahlivo a konzistentne. Systém musí minimalizovať riziko straty dát a musí korektne evidovať jednotlivé zmeny stavov.

Monitorovanie a prevádzková diagnostika

Prevádzka systému musí umožňovať:

- zber aplikačných logov,
- sledovanie chybovosti a výkonnostných metrík,
- dohľad nad stavom služieb,
- identifikáciu problémov pri nahrávaní dokumentov alebo spracovaní požiadaviek.

Rozšíriteľnosť

Architektúra systému musí byť pripravená na ďalšie funkčné rozšírenia, napríklad:

- pokročilejší workflow schvaľovania,
- rozšírené reporty a exporty,
- integráciu s účtovným systémom,
- mobilnú aplikáciu,
- podporu ďalších typov dokumentov a interných procesov.

2.2.4 Záver kapitoly

Požiadavky a funkčný návrh systému *WorkBridge* ukazujú, že ide o cloudovú aplikáciu zameranú najmä na správu používateľov, projektov, dokumentov a schvaľovacích procesov. Charakter doménových entít, potreba izolácie dát medzi firmami a kombinácia štruktúrovaných údajov s dokumentovými prílohami priamo ovplyvňujú aj následný architektonický návrh, výber databázových technológií a spôsob rozdelenia systému na samostatné služby.

3 Architektúra systému

3.1 Celková architektúra a distribúcia mikroservisov

System *WorkBridge* je navrhnutý ako cloudová viacvrstvová aplikácia využívajúca mikroservisový architektonický štýl. Hlavným cieľom zvoleného návrhu je oddeliť jednotlivé funkčné oblasti systému do samostatných služieb tak, aby bolo možné nezávisle rozvíjať, nasadzovať a škálovať tie časti riešenia, ktoré majú odlišnú doménovú zodpovednosť a prevádzkové nároky.

Z pohľadu nasadenia je architektúra rozdelená do viacerých vrstiev. Na strane klienta vystupujú používatelia systému, teda hlavný administrátor firmy, manažér a pracovník. Tí prístupujú k systému prostredníctvom webového rozhrania. Frontendová časť je navrhnutá ako samostatná webová aplikácia, distribuovaná cez službu typu CDN, pričom statické súbory používateľského rozhrania sú uložené v objektovom úložisku. Prístup k backendovým funkciám je realizovaný prostredníctvom centrálného API gateway komponentu, ktorý predstavuje jednotný vstupný bod do systému.

Vstupná vrstva backendu je doplnená o autentifikačný a autorizačný mechanizmus. Ten zabezpečuje overenie identity používateľa, správu rolí a vydávanie prístupových tokenov. Na úrovni aplikačnej logiky sú funkcionality systému rozdelené do samostatných mikroservisov podľa hlavných domén systému. Toto rozdelenie zodpovedá požiadavke na oddelenie správy firiem, pracovníkov, projektov, dokumentov, faktúr, pracovných výkazov a notifikácií.

Mikroservisová vrstva pozostáva najmä z nasledujúcich služieb:

- **Company Service** – správa firmy, firemného profilu, nastavení a tenant konfigurácie,
- **Employee Service** – správa profilov pracovníkov, kontaktných údajov a pracovných informácií,
- **Project Service** – evidencia projektov, ich stavov a priradení pracovníkov,

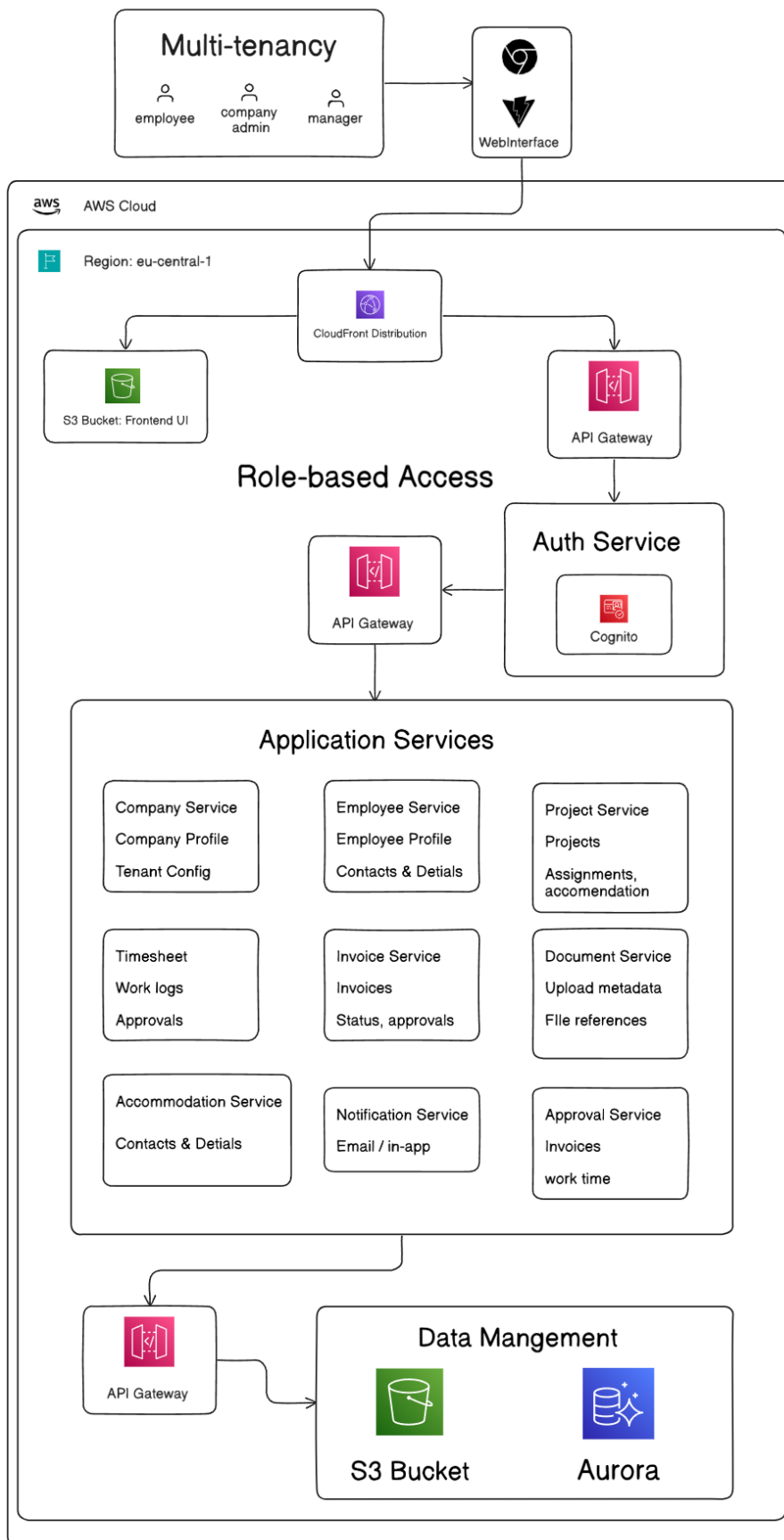
- **Accommodation Service** – evidencia ubytovania a súvisiacich organizačných údajov,
- **Timesheet / Worklog Service** – evidencia odpracovaných hodín, pracovných výkazov a ich stavov,
- **Invoice Service** – správa faktúr, ich generovania, spracovania a sledovania stavov,
- **Document Service** – správa metadát dokumentov a väzieb na súbory uložené v objektovom úložisku,
- **Approval Workflow Service** – schvaľovanie faktúr a pracovných výkazov vrátane histórie rozhodnutí,
- **Notification Service** – zasielanie e-mailových a interných notifikácií.

Samostatné rozdelenie týchto služieb umožňuje lepšie mapovať business doménu na technickú architektúru. Zároveň znižuje priame väzby medzi jednotlivými časťami systému a vytvára priestor na postupné rozširovanie funkcionality. Napríklad generovanie faktúr, spracovanie pracovných výkazov a schvaľovacie workflowy sú navrhnuté ako oddelené zodpovednosti, čo umožňuje neskoršiu úpravu procesov bez zásahu do ostatných častí riešenia.

Na technologickej úrovni je backend navrhnutý v serverless štýle s využitím samostatných vykonávacích komponentov pre jednotlivé služby. Takýto prístup znižuje prevádzkovú zložitosť, umožňuje pružné škálovanie a je vhodný pre aplikáciu s nepravidelnou alebo postupne rastúcou záťažou. Úložná vrstva je oddelená na relačnú databázu pre štruktúrované business dáta a objektové úložisko pre dokumenty a súbory.

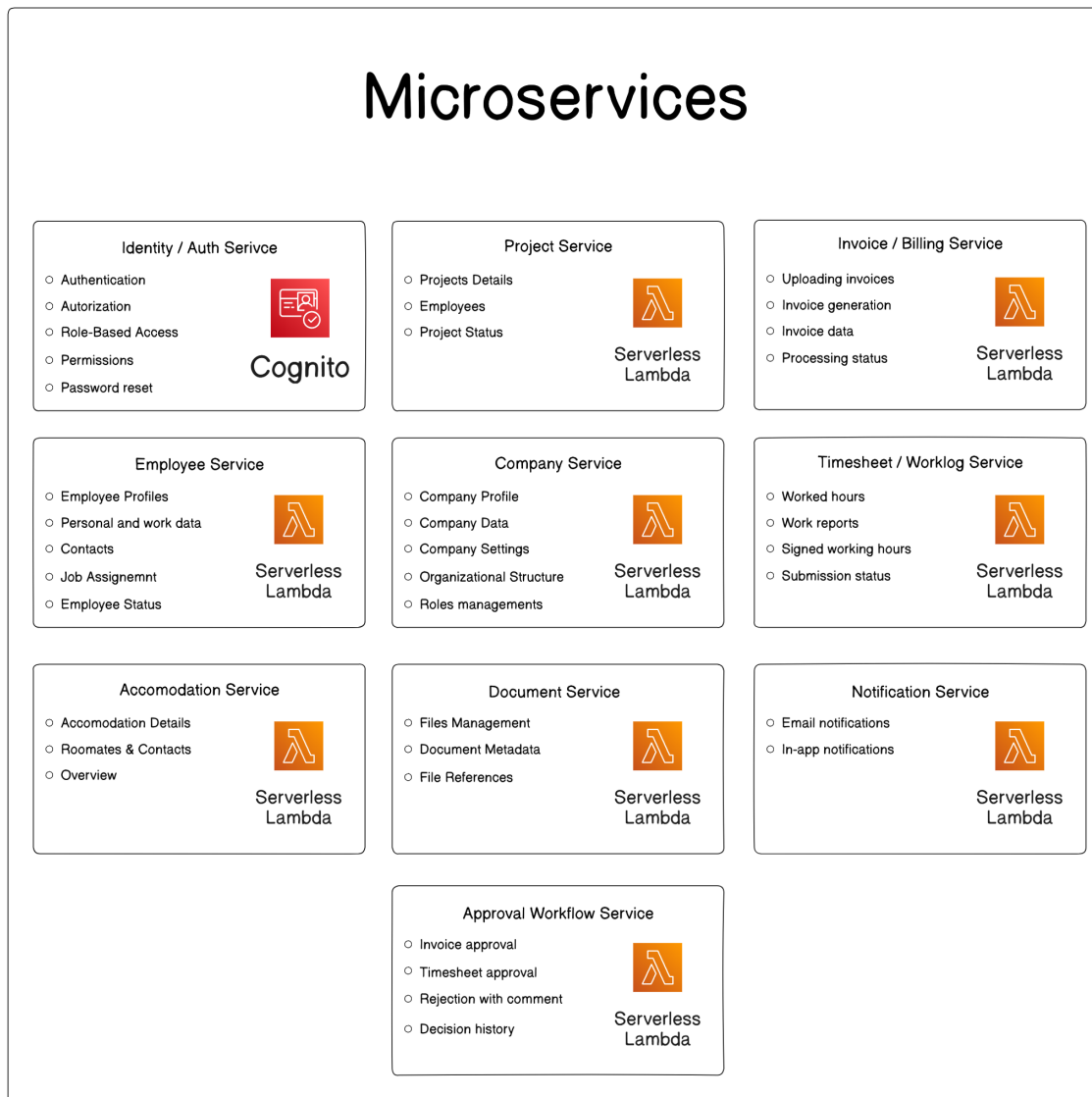
Architektúra zároveň zohľadňuje viacnájomnícky charakter riešenia. Každá firma predstavuje samostatný tenant systému a jednotlivé dáta sú logicky izolované podľa identifikátora firmy. Táto vlastnosť sa uplatňuje na úrovni používateľských účtov, projektov, faktúr, timesheetov aj dokumentov.

Obrázok 3.1 znázorňuje vysokú úroveň architektúry systému vrátane používateľských rolí, viacnájomníckeho prístupu, frontendovej vrstvy, autentifikačného mechanizmu, aplikačných služieb a dátovej vrstvy. Diagram slúži ako prehľad hlavného rozdelenia systému do logických a technologických vrstiev.



Obrázok 3.1: Vysokoúrovňová architektúra

Obrázok 3.2 detailnejšie zobrazuje navrhnuté mikroservisy a ich hlavné zodpovednosti. Tento pohľad je dôležitý najmä z hľadiska obhájenia distribúcie funkcionality medzi jednotlivé backendové služby.



Obrázok 3.2: Rozdelenie na mikroservisy

Detailnejší architektúra systému je dostupná v samostatnom externom SVG súbore `workbridge_architecture.svg` priloženom k dokumentácii. Tento diagram zachytáva jemnejšiu vnútornú štruktúru jednotlivých služieb a ich väzieb.

3.2 Komunikácia služieb a konzistencia dát

Komunikácia medzi jednotlivými časťami systému je navrhnutá kombináciou synchronného a asynchrónneho prístupu. Synchronná komunikácia sa používa najmä pri spracovaní požiadaviek z frontendovej aplikácie, keď používateľ očakáva okamžitú odpoveď systému. Typickým príkladom je načítanie zoznamu projektov, detailu pracovníka, stavu faktúry alebo uloženie nového pracovného výkazu.

Vstupným bodom do backendu je API gateway, ktorá plní niekoľko dôležitých úloh. Predovšetkým centralizuje prístup klienta k backendovým službám, zjednocuje smerovanie požiadaviek a vytvára priestor pre aplikovanie bezpečnostných mechanizmov. Prostredníctvom API gateway sú požiadavky smerované do konkrétnych mikroservisov podľa typu operácie a doménovej oblasti. Zároveň je táto vrstva vhodným miestom na validáciu prístupových tokenov, aplikovanie CORS pravidiel a prípadné obmedzovanie počtu požiadaviek.

Autentifikácia používateľov je riešená prostredníctvom samostatnej identity služby. Po úspešnom prihlásení používateľ získa token obsahujúci informácie o identite, role a tenant kontexte. Backendové služby potom na základe týchto údajov určujú, ku ktorým zdrojom má používateľ prístup. Týmto spôsobom sa zabezpečuje nielen role-based access control, ale aj izolácia dát medzi jednotlivými firmami.

Na úrovni business služieb je vhodné odlišiť operácie, ktoré si vyžadujú okamžitú odpoveď, od tých, ktoré môžu byť spracované asynchrónne. Napríklad vytvorenie alebo aktualizácia projektu je typickým synchronným scenárom. Naopak, po schválení faktúry alebo pracovného výkazu môže systém asynchrónne vygenerovať notifikáciu, zapísať udalosť do histórie a pripraviť ďalšie spracovanie bez toho, aby používateľ čakal na dokončenie týchto sekundárnych operácií.

Z hľadiska dátovej konzistencie je systém navrhnutý ako distribuovaná aplikácia s oddelenými službami, ktoré spravujú vlastnú doménovú logiku. Keďže nie všetky údaje budú menené v rámci jednej transakcie cez celý systém, návrh prirodzene smeruje k prístupu založenému na lokálnej konzistencii v rámci jednotlivých služieb a na koordinácii nadväzujúcich operácií prostredníctvom stavových zmien a udalostí.

Tento prístup je vhodný najmä pri workflowoch, kde jednotlivé kroky nadväzujú v čase, napríklad:

1. pracovník odošle pracovný výkaz a údaje pre vytvorenie faktúry,
2. systém vytvorí alebo aktualizuje príslušné záznamy,
3. manažér dokumenty skontroluje,
4. schvaľovací modul rozhodne o výslednom stave,
5. notifikačný modul informuje používateľa o výsledku.

Stavy entít, ako sú *Timesheet*, *Invoice* alebo *ApprovalHistory*, preto predstavujú dôležitý mechanizmus koordinácie procesov. Namiesto zložitej distribuovanej transakcie naprieč viacerými službami systém pracuje so zmenami stavov, auditovateľnou históriou rozhodnutí a opakovaním bezpečných operácií tam, kde je to potrebné.

V rámci návrhu API je vhodné uplatniť princíp, že každá služba vystavuje rozhranie len pre svoju vlastnú doménu. Napríklad *Project Service* sprístupňuje operácie súvisiace s projektmi a priradeniami, zatiaľ čo *Document Service* sa sústreďuje na evidenciu dokumentov a väzbu na súborové úložisko. Takéto oddelenie znižuje riziko prílišného prepojenia služieb a podporuje ich samostatnú evolúciu.

3.3 Backend služby, API a dátový model

Backendová časť systému *WorkBridge* je navrhnutá ako súbor samostatných služieb, pričom každá služba pokrýva konkrétnu doménovú oblasť. Tento prístup znižuje komplexitu jednotlivých komponentov a zároveň podporuje lepšie oddelenie zodpovedností. Z hľadiska návrhu API sa predpokladá využitie REST rozhraní, cez ktoré bude frontend komunikovať s aplikačnými službami.

3.3.1 Prehľad backendových služieb

Company Service spravuje firemný profil, základné identifikačné údaje, nastavenia organizácie a tenant konfiguráciu. Ide o službu, ktorá reprezentuje najvyšší organizačný kontext systému.

Employee Service pokrýva správu profilov pracovníkov, ich osobných a kontaktných údajov, pracovného zaradenia a základných väzieb na firmu a projektové priradenia.

Project Service zabezpečuje evidenciu projektov, ich hlavných atribútov, časových parametrov, lokality a priradenia pracovníkov. V tejto oblasti sa nachádza aj logika súvisiaca so zobrazením projektového kontextu pracovníkovi.

Accommodation Service spracúva údaje o ubytovaní vrátane adresy, kontaktu a poznámok. Hoci ide o užšiu doménu, v navrhovanom systéme má význam ako samostatná funkčná oblasť vzhľadom na potrebu sprístupniť tieto údaje pracovníkom na zahraničných projektoch.

Timesheet / Worklog Service zabezpečuje evidenciu odpracovaných hodín, popisu vykonanej práce a príloh k pracovným výkazom. Ide o jednu z kľúčových služieb systému, pretože poskytuje podklady pre naviazujúce spracovanie faktúr a schvaľovanie na strane firmy.

Invoice Service spravuje údaje o faktúrach, ich metadáta, stavové prechody a generovanie jednotného fakturačného výstupu. Faktúra môže byť v systéme vytvorená automatizovane na základe údajov doplnených pracovníkom.

Document Service oddeľuje prácu so súbormi od ostatnej doménovej logiky. Eviduje metadata dokumentov a väzbu na fyzické súbory uložené v objektovom úložisku.

Approval Workflow Service centralizuje schvaľovacie procesy nad pracovnými výkazmi a faktúrami. Uchováva rozhodnutia, dôvody zamietnutia a históriu spracovania.

Notification Service zabezpečuje informovanie používateľov o významných udalostiach v systéme, napríklad o schválení alebo zamietnutí dokumentov.

3.3.2 Návrh API rozhraní

Na úrovni API sa predpokladá zoskupenie endpointov podľa doménových oblastí. Príklady logických skupín rozhraní sú uvedené nižšie:

- /companies – operácie nad firemným profilom a nastaveniami,
- /employees – správa pracovníkov, profilov a kontaktov,
- /projects – evidencia projektov a projektových priradení,
- /accommodations – údaje o ubytovaní,
- /timesheets – pracovné výkazy a odpracované hodiny,
- /invoices – faktúry, ich vytváranie, stav a detail,
- /documents – metadáta dokumentov a väzby na súbory,
- /approvals – schvaľovanie a história rozhodnutí,
- /notifications – používateľské notifikácie.

Takto navrhnuté API vytvára prirodzené rozhrania medzi klientskou a serverovou časťou systému a zároveň zodpovedá rozdeleniu backendu na mikroservisy.

3.3.3 Dátový model

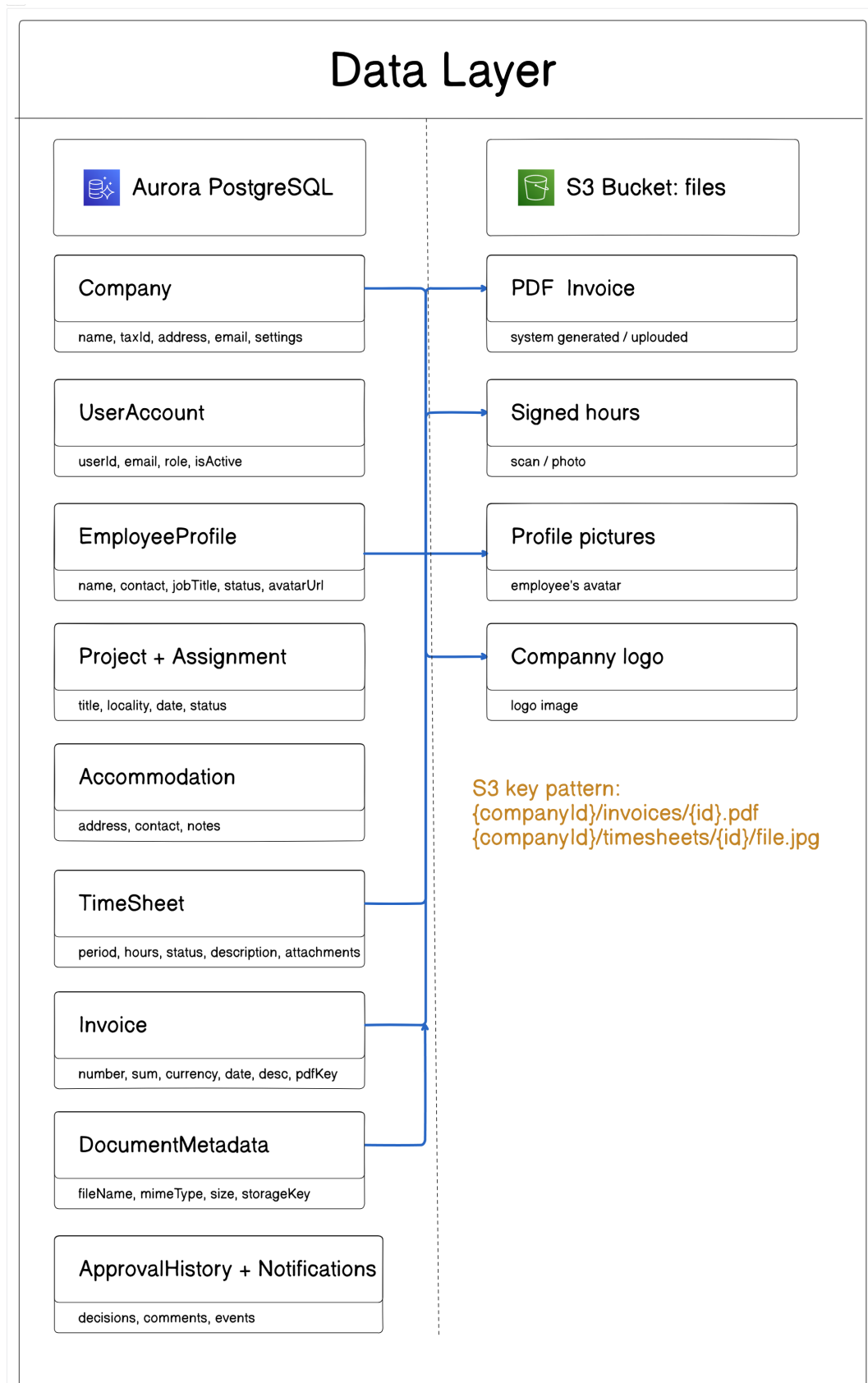
Dátová vrstva systému je rozdelená na relačné úložisko a objektové úložisko. Relačná databáza uchováva štruktúrované business dáta, zatiaľ čo objektové úložisko je určené na správu dokumentov a súborových príloh. Takéto rozdelenie vychádza z povahy údajov spracovávaných systémom.

Medzi hlavné relačné entity patria najmä:

- **Company,**
- **UserAccount,**
- **EmployeeProfile,**
- **Project,**
- **ProjectAssignment,**
- **Accommodation,**
- **Timesheet,**
- **Invoice,**
- **DocumentMetadata,**
- **Notification,**
- **ApprovalHistory.**

Každá z týchto entít je naviazaná na konkrétnu firmu, čím sa zabezpečuje tenant izolácia dát. Významnú úlohu zohrávajú aj väzby medzi pracovným výkazom, faktúrou, dokumentmi a schvaľovacou históriou, keďže práve tieto entity spolu tvoria hlavný administratívny proces systému.

Obrázok 3.3 zobrazuje základné rozdelenie dát medzi relačnú databázu Aurora PostgreSQL a objektové úložisko S3. Diagram zároveň ilustruje, ktoré typy údajov sú uchovávané ako štruktúrované entity a ktoré ako dokumenty alebo binárne súbory.



Obrázok 3.3: Návrh dátovej vrstvy systému a rozdelenie údajov medzi relačnú databázu a objektové úložisko

Obrázok 3.4 predstavuje detailnejší pohľad na hlavné entity systému a ich vzťahy. Tento model slúži ako základ pre implementáciu relačnej databázovej vrstvy a zároveň potvrdzuje, že charakter spracovávaných údajov je prevažne relačný.



Obrázok 3.4: Konceptuálny a databázový model hlavných entít systému

3.4 Frontend návrh a interakcia s backendom

Frontendová časť systému *WorkBridge* je navrhnutá ako webová aplikácia určená pre tri hlavné typy používateľov: hlavného administrátora firmy, manažéra a pracovníka. Návrh používateľského rozhrania musí zohľadňovať, že jednotlivé roly pracujú s odlišnými scenármi a potrebujú rozdielny rozsah funkcionality.

Z pohľadu používateľského rozhrania možno frontend rozdeliť na dve hlavné časti. Prvou je administratívne rozhranie pre firmu, v ktorom sa spravujú pracovníci, projekty, ubytovanie, faktúry a pracovné výkazy. Druhou je rozhranie pracovníka, ktoré je zamerané na jednoduchý prístup k osobným a projektovým informáciám a na odosielanie podkladov do systému.

Administratívna časť by mala poskytovať najmä tieto obrazovky:

- dashboard s prehľadom projektov, čakajúcich schválení a notifikácií,
- zoznam a detail pracovníkov,
- zoznam a detail projektov,
- správa projektových priradení a ubytovania,
- prehľad timesheetov a faktúr čakajúcich na spracovanie,
- detail schvaľovania s možnosťou potvrdenia alebo zamietnutia.

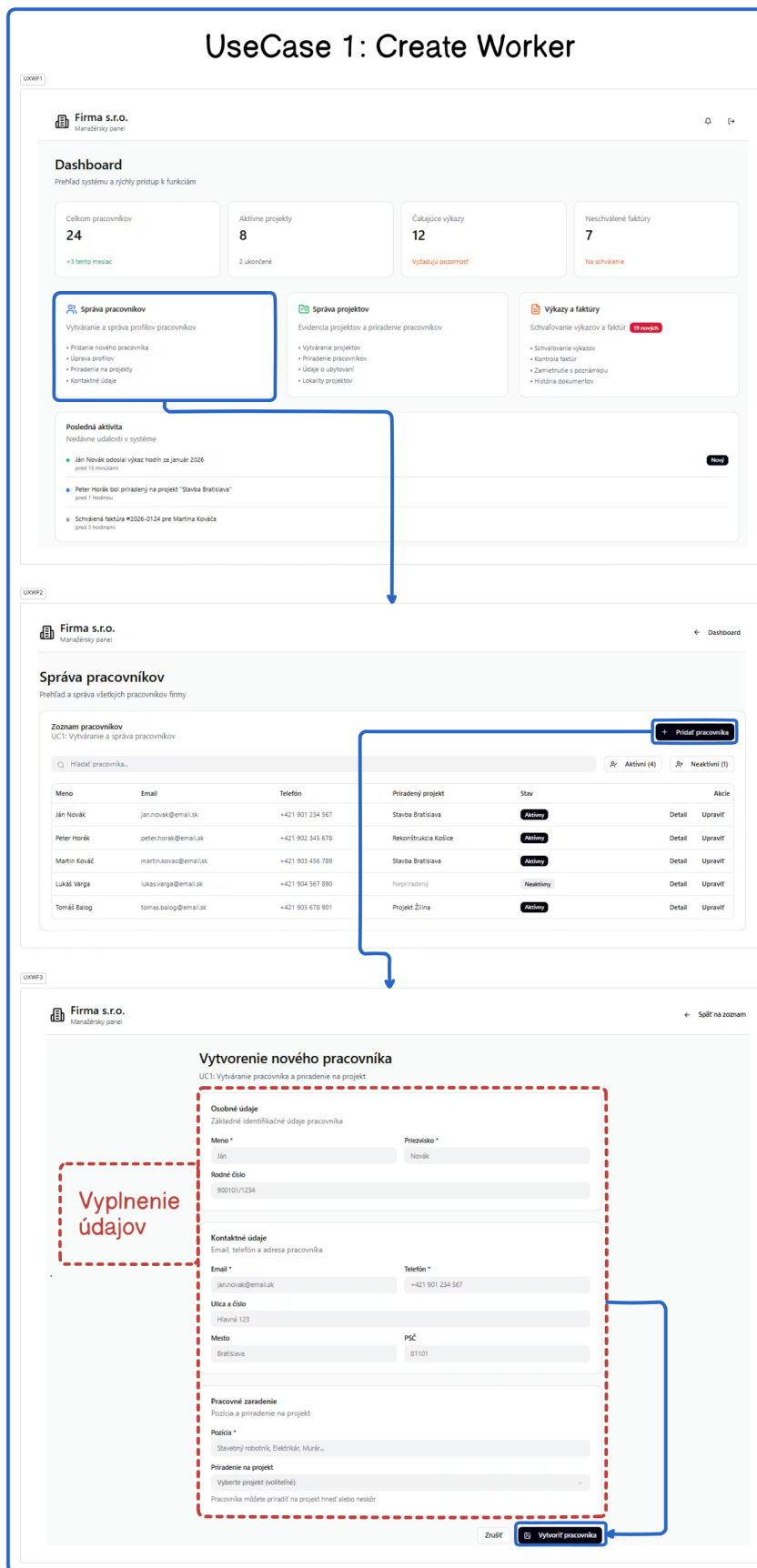
Používateľské rozhranie pracovníka by malo obsahovať najmä:

- prehľad aktuálneho projektu,
- zoznam kolegov a kontaktných údajov,
- detail ubytovania,
- formulár na zadanie odpracovaných hodín,
- formulár na doplnenie údajov potrebných na vytvorenie faktúry,
- prehľad odoslaných dokumentov a ich stavov.

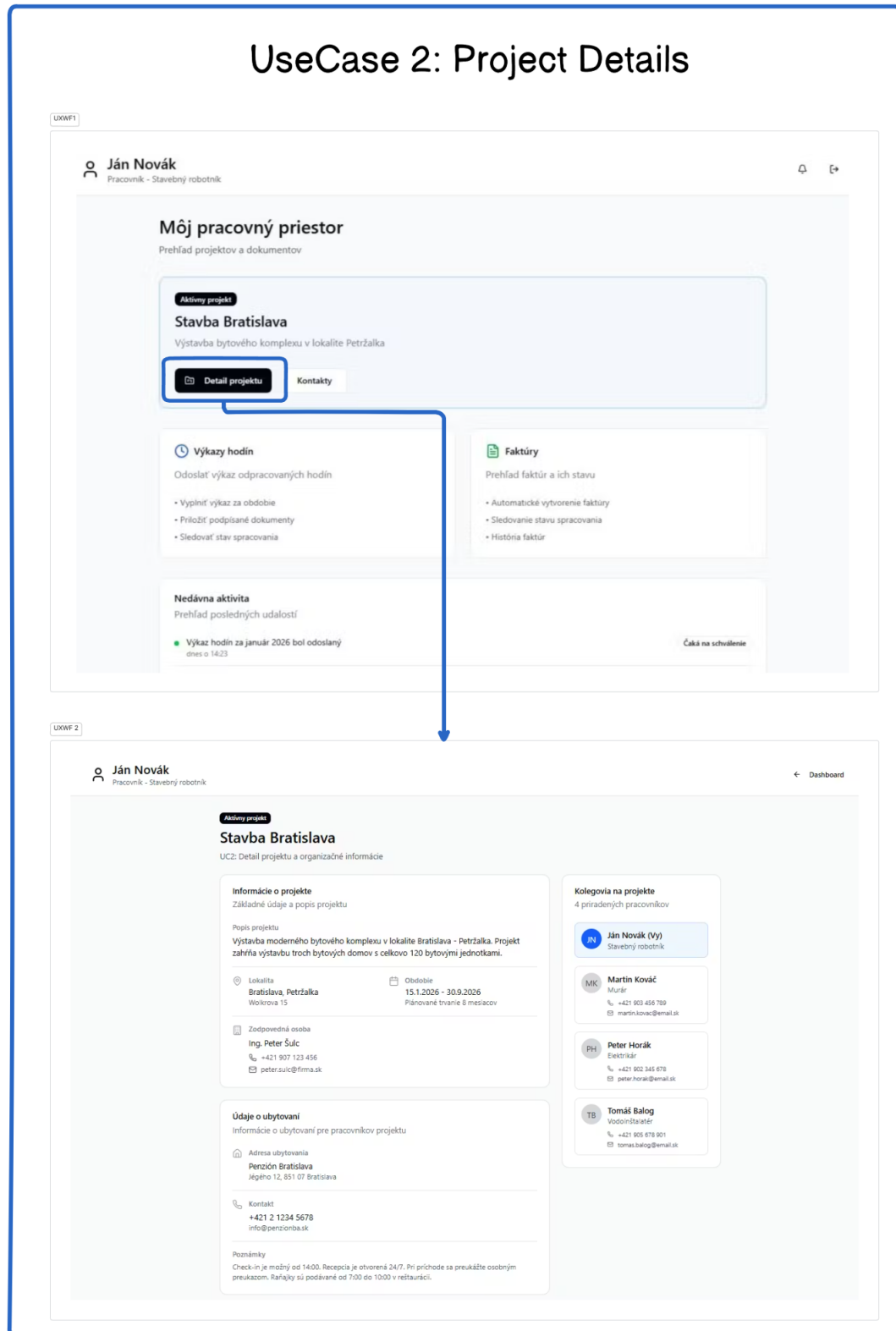
Z hľadiska interakcie s backendom frontend komunikuje cez API gateway pomocou REST rozhraní. Po úspešnom prihlásení používateľ pracuje v kontexte svojej roly a firmy. Rozhranie preto musí vedieť dynamicky meniť dostupné obrazovky a akcie podľa typu používateľa a oprávnení uložených v identitnom tokene.

Dôležitou vlastnosťou frontendového návrhu je minimalizácia zložitosti pre pracovníka. Pracovník by nemal byť nútený pracovať s komplikovanými administratívnymi formulármi. Jeho hlavné scenáre majú byť jednoduché, priamočiare a optimalizované pre rýchle zadanie údajov o práci, nahratie podkladov a sledovanie stavu spracovania. Naopak, administratívna časť môže obsahovať detailnejšie pohľady, filtre a kontrolné mechanizmy vhodné pre firemné procesy.

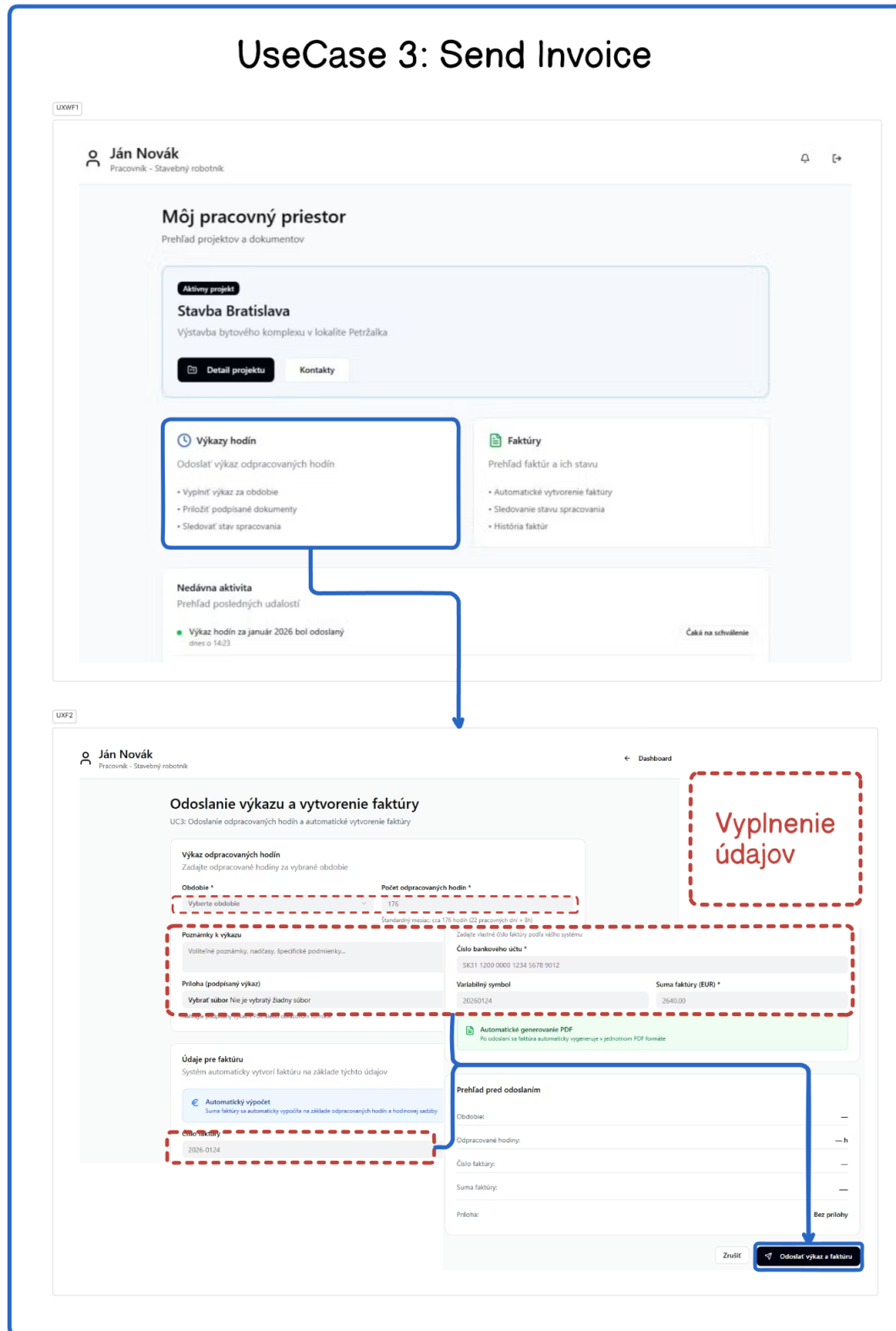
Na spresnenie návrhu používateľského rozhrania sú v tejto časti uvedené postupnosti wireframov zachytávajúce hlavné scenáre použitia systému. Tieto wireframy nereprezentujú iba jednotlivé samostatné obrazovky, ale celé používateľské toky od vstupu do systému až po dokončenie konkrétnej úlohy. Takýto pohľad umožňuje lepšie overiť logiku navigácie, nadväznosť krokov a rozdelenie funkcionality medzi jednotlivé roly. Obrázky 3.5 až 3.9 preto ilustrujú štyri kľúčové scenáre práce so systémom z pohľadu administratívnej aj pracovnickej časti rozhrania.



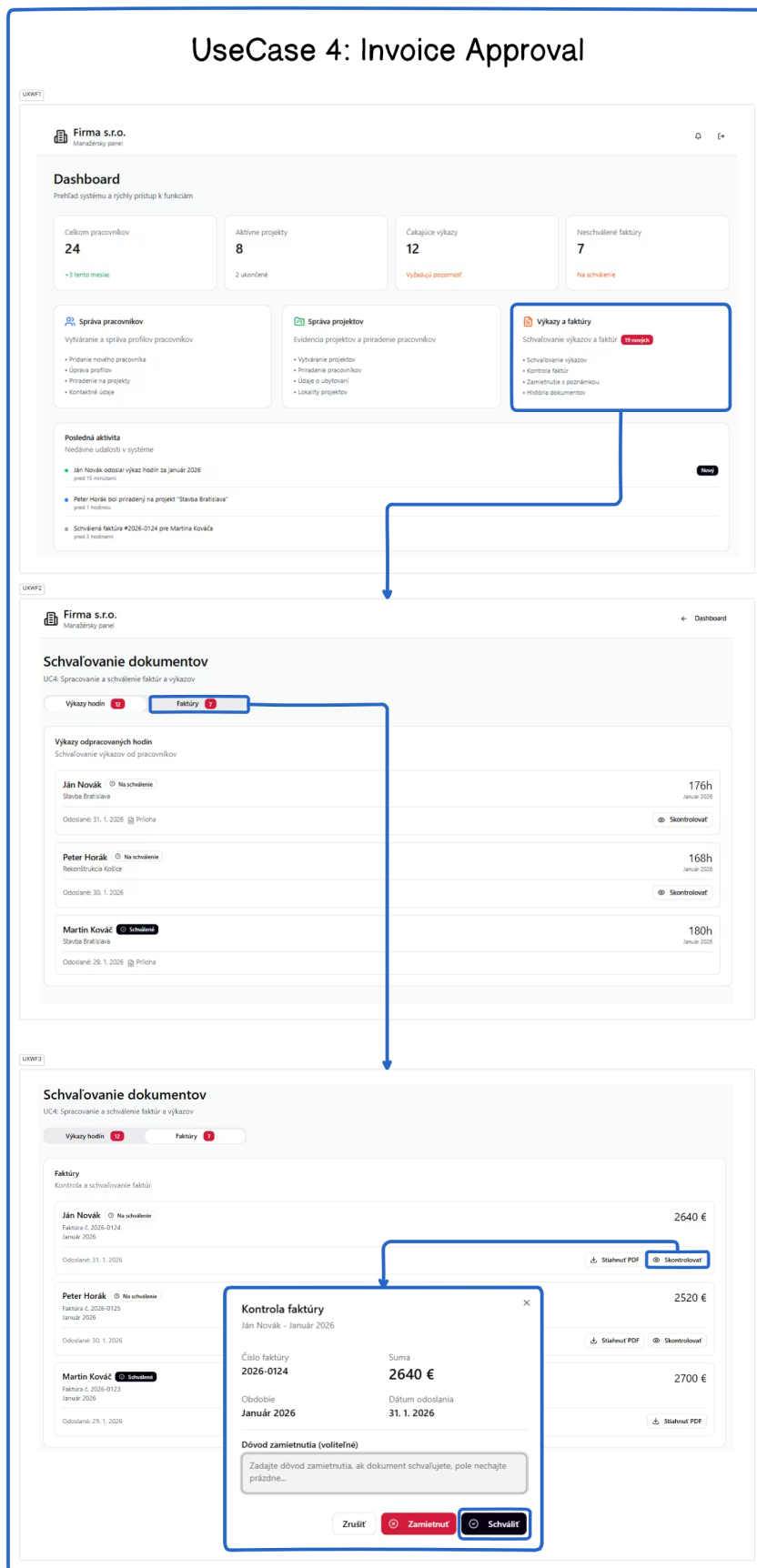
Obrázok 3.5: Postupnosť wireframov scenára UC1



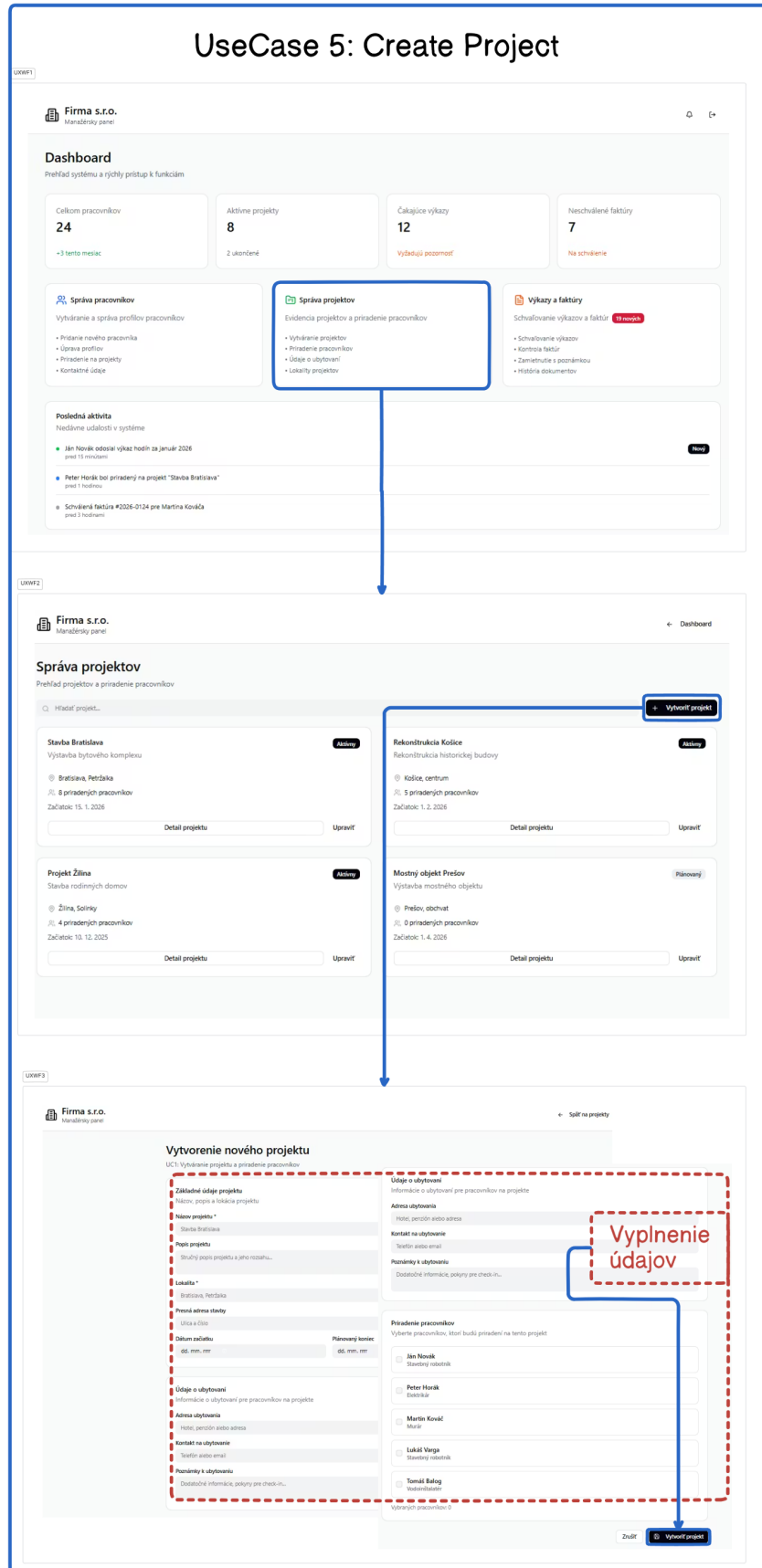
Obrázok 3.6: Postupnosť wireframov scenára UC2



Obrázok 3.7: Postupnosť wireframov scenára UC3



Obrázok 3.8: Postupnosť wireframov scenára UC4



Obrázok 3.9: Postupnosť wireframov scenára UC5

4 Nasadenie a technológie

4.1 Cloud platforma, deployment a použité služby

Pre realizáciu systému *WorkBridge* bola zvolená cloudová platforma Amazon Web Services (AWS). Toto rozhodnutie vychádza z viacerých dôvodov. AWS poskytuje široké spektrum spravovaných služieb vhodných pre návrh viacvrstvovej cloudovej aplikácie, podporuje serverless aj mikroservisový prístup a zároveň umožňuje jednoducho kombinovať aplikačnú logiku, autentifikáciu, databázové služby, objektové úložisko a monitoring v jednom prostredí. Takýto výber je v súlade s celkovým návrhom systému, ktorý počíta s oddeleným frontendom, API gateway vrstvou, samostatnými backend službami a kombináciou relačného a objektového úložiska.

Nasadenie systému je navrhnuté do regiónu `eu-central-1`, ktorý predstavuje vhodnú voľbu z pohľadu geografickej blízkosti, latencie a praktickej dostupnosti služieb pre používateľov v európskom priestore. Frontendová časť systému je navrhnutá ako statická webová aplikácia, ktorá bude uložená v službe Amazon S3 a distribuovaná cez Amazon CloudFront. Takýto model nasadenia je vhodný pre webové rozhranie typu SPA alebo modernej frontendovej aplikácie, pretože umožňuje jednoduché publikovanie novej verzie, dobrý výkon pri doručovaní statického obsahu a prirodzené škálovanie bez potreby správy vlastných serverov.

Vstupným bodom do backendu je Amazon API Gateway. Táto služba zabezpečuje centralizované smerovanie požiadaviek z klienta na jednotlivé backendové komponenty, pričom zároveň vytvára priestor pre bezpečnostné pravidlá, validáciu požiadaviek, kontrolu prístupu a jednotné vystavenie API navonok. V návrhu systému vystupuje API Gateway ako centrálna integračná vrstva medzi frontendom a mikroservismi.

Autentifikácia a správa identít sú riešené prostredníctvom Amazon Cognito. Táto služba zabezpečuje prihlasovanie používateľov, správu účtov, vydávanie tokenov a podporu role-based access control. V kontexte systému *WorkBridge* je dôležité, že identitná vrstva zároveň prenáša informáciu o používateľskej roli a príslušnosti k firme, čo následne umožňuje izolovať dáta medzi tenantmi a riadiť oprávnenia pri prístupe k backendovým službám.

Business logika jednotlivých mikroservisov je navrhnutá v serverless štýle s využitím AWS Lambda. Takýto prístup je vhodný najmä preto, že jednotlivé funkčné oblasti systému sú relatívne dobre oddeliteľné a nevyžadujú trvalo bežiacie aplikačné uzly. Výhodou serverless prístupu je zníženie prevádzkovej reže, automatické škálovanie podľa počtu požiadaviek a jednoduchšie nasadzovanie menších samostatných komponentov. Tento výber zároveň zodpovedá odporúčaniam zadania preferovať serverless riešenia pri ľahších komponentoch a využívať existujúce cloudové služby, ako sú API gateway, blob storage, notifikácie a autentifikácia.

Na ukladanie štruktúrovaných aplikačných dát je navrhnutá relačná databáza Amazon Aurora PostgreSQL. Tento výber zodpovedá charakteru dátového modelu systému, ktorý pozostáva z previazaných entít, ako sú firmy, používateľské účty, profily pracovníkov, projekty, priradenia, timesheety, faktúry a schvaľovacia história. Relačná databáza je v tomto prípade vhodná najmä z dôvodu potreby konzistentných zápisov, väzieb medzi entitami a možnosti efektívneho dotazovania nad stavovými a administráčnymi údajmi.

Na prácu so súbormi a dokumentami je použitá služba Amazon S3. V objektovom úložisku budú ukladané najmä PDF faktúry, podpísané pracovné výkazy, profilové obrázky a prípadne ďalšie dokumentové prílohy. Samotná aplikačná databáza pritom nebude uchovávať binárny obsah súborov, ale iba metadata a referencie na ich uloženie. Tento prístup je konzistentný s dátovým návrhom systému a zároveň znižuje zaťaženie relačnej databázy.

Na zasielanie upozornení je vhodné využiť niektorú z notifikačných služieb AWS, napríklad Amazon SES pre e-mailové notifikácie alebo kombináciu služieb Amazon SNS, Amazon SQS a Amazon EventBridge pre jednoduché asynchrónne spracovanie udalostí medzi službami. V rámci návrhu systému je dôležité, aby notifikačný mechanizmus nebol pevne viazaný na synchronné spracovanie hlavnej požiadavky, ale aby bolo možné generovať notifikácie po zmene stavu dokumentu alebo po rozhodnutí schvaľovacieho workflowu asynchrónne.

Monitoring, logovanie a základná prevádzková diagnostika sú prirodzene naviazané na služby ako Amazon CloudWatch. Táto vrstva je dôležitá najmä pri

sledovaní chybovosti, výkonu jednotlivých backendových funkcií, priebehu integrácií a stability systému po nasadení.

Z pohľadu deploymentu možno nasadenie systému rozdeliť na dve hlavné časti. Frontend je publikovaný ako statická webová aplikácia, zatiaľ čo backend je nasadzovaný ako sada samostatných serverless funkcií a konfiguračných komponentov. Takýto spôsob nasadenia podporuje nezávislé verziovanie frontendovej a backendovej vrstvy, jednoduchšie aktualizácie a priebežné rozširovanie systému bez potreby zásahu do celej infraštruktúry naraz.

4.2 Dopad nasadenia do produkcie

Nasadenie systému *WorkBridge* do produkčného prostredia má významný dopad najmä z pohľadu spôsobu práce cieľových používateľov. V porovnaní s tradičným riešením založeným na e-mailoch, papierových dokumentoch a neprepojených tabuľkách prináša centralizovaná cloudová aplikácia vyššiu mieru transparentnosti, lepší prehľad nad projektmi a zníženie administratívnej záťaže. Tento prínos vyplýva z identifikovaných problémov existujúceho spôsobu práce, medzi ktoré patrí najmä neprehľadnosť a roztrúsenosť údajov.

Jedným z najdôležitejších praktických dopadov produkčného nasadenia je zjednotenie procesov medzi firmou a pracovníkmi. Firma získa centralizovaný prehľad o pracovníkoch, projektoch, odpracovaných hodinách, faktúrach a schvaľovacích stavoch. Pracovníci budú mať naopak prístup k dôležitým prevádzkovým informáciám na jednom mieste a nebudú odkázaní na komunikáciu cez viacero oddelených kanálov. Tým sa skracuje čas potrebný na odosielanie a kontrolu podkladov a zároveň sa znižuje riziko straty alebo duplicity údajov.

Nasadenie do produkcie však prináša aj technické a prevádzkové požiadavky, ktoré musia byť pri reálnom používaní zohľadnené. V prvom rade ide o dostupnosť služby. Keďže pracovníci môžu aplikáciu používať aj zo zahraničia a v rôznom čase, systém musí byť navrhnutý tak, aby bol spoľahlivo dostupný cez internet a aby zvládol výpadky alebo zvýšené zaťaženie bez zásadného dopadu na používateľský komfort. Tento aspekt je jedným z dôvodov, prečo je cloudový a serverless prístup vhodný aj z prevádzkového hľadiska.

Ďalším dopadom produkčného nasadenia je potreba dôsledného riešenia bezpečnosti. Systém pracuje s firemnými údajmi, osobnými údajmi pracovníkov a dokumentmi, ktoré môžu obsahovať citlivý obsah. Preto je nevyhnutné zabezpečiť silnú autentifikáciu, riadenie prístupových práv, šifrovanie komunikácie a logickú izoláciu dát medzi firmami. Pri produkčnom nasadení už nejde len

o funkčnosť aplikácie, ale aj o schopnosť splniť bezpečnostné a prevádzkové očakávania reálnych používateľov.

Významný dopad má aj potreba monitorovania a diagnostiky systému po nasadení. V produkcii musí byť možné sledovať stav aplikačných služieb, zlyhania pri nahrávaní dokumentov, chyby backendových funkcií, oneskorenia pri spracovaní požiadaviek a problémy pri generovaní notifikácií. Bez takejto vrstvy by bolo reálne prevádzkovanie systému výrazne komplikovanejšie, najmä pri raste počtu používateľov a tenantov.

Z prevádzkového a ekonomického hľadiska je výhodou navrhovaného riešenia aj postupná škálovateľnosť. Keďže systém je založený na spravovaných cloudových službách a serverless backendových komponentoch, nie je nutné od začiatku prevádzkovať robustnú infraštruktúru s pevne alokovanou kapacitou. To znamená, že riešenie môže byť vhodné aj pre menšie a stredné firmy, pre ktoré je dôležité začať s primeranými nákladmi a neskôr systém rozširovať podľa reálnej potreby.

Produkčné nasadenie má vplyv aj na kvalitu používateľského rozhrania a používateľských tokov. Pri internom prototypu je možné tolerovať určitú mieru nekomfortu alebo nedokončenosti, ale pri nasadení pre reálnych používateľov je dôležité, aby boli najčastejšie scenáre čo najjednoduchšie. Týka sa to najmä pracovníckej časti systému, kde musí byť formulár na odoslanie hodín a podkladov pre faktúru zrozumiteľný, rýchly a dostupný aj na menších zariadeniach.

Z pohľadu reálneho prostredia možno teda dopad nasadenia zhrnúť do troch hlavných oblastí: organizačný prínos pre firmu, vyššia transparentnosť pre pracovníka a zvýšené nároky na bezpečnosť, monitoring a spoľahlivosť systému. Práve tieto aspekty odlišujú školský návrh od reálne použiteľného cloudového riešenia.

4.3 Použité technológie a modelovacie nástroje

Pri návrhu systému *WorkBridge* boli použité technológie a modelovacie nástroje zodpovedajúce charakteru navrhovanej cloudovej aplikácie. Výber technológií vychádza z požiadavky rozdeliť riešenie na frontendovú časť, backendové služby, dátovú vrstvu a podporné cloudové komponenty. Zároveň reflektuje odporúčanie zadania využívať štandardné modelovacie nástroje a vhodné cloudové služby pri návrhu riešenia.

Z pohľadu cloudovej infraštruktúry bol zvolený ekosystém Amazon Web Services. V návrhu sa konkrétne uvažuje s využitím týchto služieb:

- **Amazon S3** – ukladanie frontendových statických súborov a dokumentových príloh,
- **Amazon CloudFront** – distribúcia frontendovej aplikácie,
- **Amazon API Gateway** – jednotný vstupný bod pre backendové API,
- **Amazon Cognito** – autentifikácia používateľov a správa identity,
- **AWS Lambda** – implementácia serverless backendových služieb,
- **Amazon Aurora PostgreSQL** – relačná databáza pre business dáta,
- **Amazon SES / SNS / SQS / EventBridge** – notifikácie a asynchrónne spracovanie udalostí,
- **Amazon CloudWatch** – monitoring, logovanie a diagnostika.

Frontendová vrstva je navrhnutá ako moderná webová aplikácia. V praktickej implementácii je vhodné použiť framework typu React alebo Next.js, ktorý umožňuje budovať komponentovo orientované používateľské rozhranie, efektívne pracovať so stavom aplikácie a jednoducho komunikovať s REST API backendu. Vzhľadom na charakter navrhovaného používateľského rozhrania ide o prirodzenú voľbu aj z implementačného hľadiska.

Backendová vrstva je založená na samostatných službách, ktoré možno implementovať ako Lambda funkcie vystavené cez API Gateway. Pri implementácii business logiky je vhodné použiť jazyk a runtime, ktorý umožňuje efektívnu prácu s AWS ekosystémom a rýchly vývoj menších služieb. Typickou voľbou je napríklad TypeScript/Node.js, prípadne Python. Vhodné je zvoliť takú technologickú kombináciu, ktorá podporuje serverless architektúru a jednoduchú integráciu s cloudovými službami.

Na úrovni databázového návrhu bola použitá relačná modelácia hlavných entít systému a ich väzieb. Dôležitým modelovacím výstupom je databázová schéma entít ako *Company*, *UserAccount*, *EmployeeProfile*, *Project*, *Timesheet*, *Invoice*, *DocumentMetadata* alebo *ApprovalHistory*. Tento model slúži ako základ implementácie databázovej vrstvy a zároveň odráža povahu dát spracovávaných systémom.