



# Standardy vývoje informačních systémů Správy železnic

Březen 2022

# Historie verzí

Verze	Popis	Platnost od	Předchozí verze
0.1	Draft	22. 3. 2022	
1.0	První verze dokumentu	31. 3. 2022	

# Obsah

Seznam zkratk a pojmů.....	3
1 Standardy vývoje informačních systémů Správy železnic .....	4
1.1 Dvouvrstvá architektura .....	4
1.1.1 Datová vrstva.....	4
1.1.2 Aplikační vrstva .....	4
1.2 Třívrstvá a vícevrstvá architektura .....	4
1.2.1 Datová vrstva.....	5
1.2.2 Aplikační vrstva .....	5
1.2.3 Prezentační vrstva .....	5
1.2.4 Integrovaná vrstva .....	5
1.3 Požadavky na prezentační vrstvu .....	6
1.3.1 Uživatelské rozhraní (User Interface, UI) .....	6
1.3.2 Uživatelský prožitek (User Experience, UX) .....	6
1.4 Bezpečnost .....	7
1.4.1 Zabezpečení aplikací .....	7
1.4.2 Autentizace a autorizace .....	7
1.4.3 GDPR .....	8
1.5 Dokumentace .....	8
1.5.1 Technická dokumentace jádra systému.....	8
1.5.2 E-R modely databáze .....	8
1.5.3 Objektový model pro aplikace .....	8
1.5.4 Procesní diagramy, schémata toků dat .....	8
1.5.5 Komunikační rozhraní.....	8
1.5.6 Drátové modely všech obrazovek uživatelského rozhraní aplikací.....	8
1.5.7 Popis konfigurace provozního prostředí.....	9
1.5.8 Uživatelská příručka .....	9
1.5.9 Příručka administrátora .....	9
1.6 Předávání vývoje do provozu .....	9

# Seznam zkratek a pojmů

<b>3NF</b>	Třetí normální forma
<b>API</b>	<i>z angl. Application Programming Interface</i> , rozhraní pro programování aplikací
<b>APP</b>	Aplikační vrstva
<b>AS</b>	Aplikační server
<b>DB</b>	Databáze
<b>DBMS</b>	<i>z angl. Database Management System</i> , Systém řízení databáze
<b>DC</b>	Datové centrum
<b>DDL</b>	<i>z angl. Data Definition Language</i>
<b>DR</b>	<i>z angl. Disaster Recovery</i> , Obnova po havárii
<b>HA</b>	<i>z angl. High Availability</i> , Vysoká dostupnost
<b>HW</b>	Hardware označuje veškeré fyzicky existující technické vybavení počítače
<b>JSON</b>	<i>z angl. JavaScript Object Notation</i> , JavaScriptový objektový zápis
<b>OS</b>	Operační systém
<b>SQL</b>	Structured Query Language, standardizovaný dotazovací jazyk pro práci v relačních databázích
<b>SW</b>	Software je sada všech počítačových programů používaných v počítači, které provádějí nějakou činnost
<b>SŽ</b>	Správa železnic, státní organizace
<b>WS</b>	Webový server
<b>XML</b>	<i>z angl. Extensible Markup Language</i> , obecný značkovací jazyk

# 1 Standardy vývoje informačních systémů

## Správy železnic

Při vývoji software ve Správě železnic je požadováno, aby byly plně respektovány obvyklé metodiky a best-practice pro návrh a vývoj software pomocí vícevrstvé architektury. Konkrétní užití jednotlivých vzorů se řídí vhodností, plánovanou zátěží a požadavky na dostupnost vyvíjeného software.

### 1.1 Dvouvrstvá architektura

Dvouvrstvou architekturu při vývoji software lze využít v případě, kdy se jedná o menší, samostatný software, který nebude integrován na další informační systémy, nebo datové zdroje Správy železnic. Užití takového software je plánováno pro menší desítky uživatelů, bez požadavku na vysokou dostupnost a možnosti škálování výkonu a rozložení zátěže prostřednictvím clusterování. U tohoto typu software nejsou definovány požadavky na vysokou odolnost proti chybám, rychlou reakci systému, nebo správu dat pro velké sítě.

Využití dvouvrstvé architektury musí být předem diskutováno s Oddělením IT architektury, které v odůvodněných případech vydá příslušnou výjimku.

#### 1.1.1 Datová vrstva

Realizace datové vrstvy je požadována prostřednictvím preferované relační databáze (dle služeb Platformy) a respektováním metodiky 3NF. Je požadován jednoznačný datový model s minimální redundancí dat a datové struktury budou modelovány a popsány jazykovými konstrukcemi DDL, které jsou kompatibilní s určeným databázovým systémem.

Celá struktura dat bude popsána formálně prostředky E-R modelování. K datovému modelu je požadováno dodat korespondující SQL DDL skripty, který budou plně odpovídat dodané databázi. Je požadováno, aby správnost, úplnost a optimalizace datového modelu byla řešena již v rámci návrhu řešení.

V rámci dvouvrstvé architektury je umožněno, aby logika byla rozprostřena částečně v databázi a částečně v aplikační, resp. prezentační vrstvě.

#### 1.1.2 Aplikační vrstva

Aplikační vrstva a prezentační vrstva je ve dvouvrstvé architektuře realizována jako jedna, společná a nedělitelná vrstva. Je požadováno, aby tato vrstva byla realizována v souladu s principy objektově orientovaného programování a komunikace mezi vrstvami byla realizována standardními zabezpečenými a šifrovanými protokoly. Je požadováno, aby uživatelské identity nebyly z aplikační vrstvy prezentovány do datové vrstvy, přičemž tyto vrstvy musí mezi sebou komunikovat technickým účtem, k tomu účelu v databázi vytvořeném.

Je požadováno, aby aplikační vrstva podporovala Multitasking, tedy umožňovala provádění několika procesů současně a systém byl již v rámci návrhu a vývoje optimalizován plánovaný výkon.

V rámci vývoje musí být ošetřena všechna bezpečnostní rizika popsaná v kapitole 1.4.

### 1.2 Třívrstvá a vícevrstvá architektura

Třívrstvá a vícevrstvá architektura je požadována při vývoji software ve všech případech mimo výjimky definované v kap. 1.1. Specifikace řešení vyžadující třívrstvou architekturu tak může disponovat následujícími vlastnostmi:

- Má být integrován na jiný software Správy železnic, nebo software třetích stran, a to z důvodu jednotného přístupu k datům a procesům vyvíjeného software
- Je plánováno využití pro větší počty uživatelů
- Je požadována vysoká dostupnost (HA)

- Je požadován Clustering pro rozložení zátěže a škálování výkonu
- Je požadována vysoká odolnost proti chybám, rychlá reakce systému, nebo správa dat pro velké sítě

### 1.2.1 Datová vrstva

Realizace datové vrstvy je požadována prostřednictvím preferované relační databáze (dle služeb Platformy) a respektováním metodiky 3NF. Je požadován jednoznačný datový model s minimální redundancí dat, datové struktury budou modelovány a popsány jazykovými konstrukcemi DDL, které jsou kompatibilní s určeným databázovým systémem.

Celá struktura dat bude popsána formálně prostředky E-R modelování. K datovému modelu je požadováno dodat korespondující SQL DDL skripty, který budou plně odpovídat dodané databázi. Je požadováno, aby správnost, úplnost a optimalizace datového modelu byla řešena již v rámci návrhu řešení.

V rámci třívrstvé a vícevrstvé architektury není umožněno, aby logika byla rozprostřena částečně v databázi a částečně v aplikační vrstvě. Aplikační logika je tak striktně pouze v aplikační vrstvě.

### 1.2.2 Aplikační vrstva

Je požadováno, aby tato vrstva byla realizována v souladu s principy objektově orientovaného programování a komunikace mezi vrstvami byla realizována standardními zabezpečenými a šifrovanými protokoly. Je požadováno, aby uživatelské identity nebyly z aplikační vrstvy prezentovány do datové vrstvy, přičemž tyto dvě vrstvy musí mezi sebou komunikovat technickým účtem, k tomu účelu v databázi vytvořeném.

Je požadováno, aby aplikační vrstva podporovala Multitasking, tedy umožňovala provádění několika procesů současně a v již rámci návrhu a vývoje optimalizovat plánovaný výkon.

V rámci vývoje musí být ošetřena všechna bezpečnostní rizika popsaná v kapitole 1.4.

### 1.2.3 Prezentační vrstva

Pro interakci s uživatelem je požadováno, aby prezentační vrstva byla realizována desktopovým klientem (tlustým), nebo webovým klientem (tenkým), a to v závislosti na vhodnosti použití a požadavcích na software kladených. Komunikace mezi prezentační a aplikační vrstvou musí být realizována standardními zabezpečenými a šifrovanými protokoly.

V rámci prezentační vrstvy a desktopového klienta je možné přenesením části aplikační logiky na klienta, tedy využití prostředků klientské stanice ke zvýšení výkonu systému, ale pouze za předpokladu, že tento systém bude zabezpečovat konzistenci aplikační logiky, napříč všemi desktopovými klienty.

Bez aktualizčních mechanismů, které zajistí stejné verze software, na všech klientských stanicích v reálném čase není tato možnost povolena.

### 1.2.4 Integrační vrstva

V případě, kdy vyvíjený software má být integrován na jiný software Správy železnic, nebo software třetích stran, je požadováno, aby tato integrační vrstva byla realizována jako samostatná vrstva, umožňující škálování výkonu a rozložení zátěže.

Realizace integrací mezi aplikačními komponentami musí splňovat principy SOA. Veškerá komunikace tedy musí probíhat prostřednictvím definovaných služeb rozhraní, a není tedy povolena výměna dat prostřednictvím přímých vazeb, jako je sdílení paměti, souborů, nebo databází. Pokud je k dispozici, komunikace probíhá prostřednictvím k tomu určené sběrnice (ESB) nebo integrační platformy.

V případě, že má být vyvíjena komponenta integrována se **spisovou službou SŽ**, musí splňovat požadavky na integraci prostřednictvím Národního standardu pro elektronické systémy spisové služby<sup>1</sup> a integrace musí být rozhraními definovanými v tomto standardu také realizována.

V případě, že má být vyvíjena aplikace integrována s programovým prostředím komponent **systému SAP**, musí být realizována prostřednictvím určené integrační platformy (SAP Cloud Platform, příp. produktu, která jej nahradí). Detailní parametry požadavku na integraci budou definovány v příslušných případech.

## 1.3 Požadavky na prezentační vrstvu

### 1.3.1 Uživatelské rozhraní (User Interface, UI)

Pomocí uživatelského rozhraní může uživatel komunikovat se zařízením, počítačem a programy. Při navrhování vysoce kvalitního uživatelského rozhraní je požadováno zohlednit nejen vzhled rozhraní, ale také jeho logickou strukturu, aby s ním uživatel mohl snadno a rychle komunikovat a dosáhnout požadovaného výsledku bez zbytečného úsilí. Cílem je vytvořit rozhraní, které poskytuje jednoduchou, srozumitelnou a pohodlnou interakci uživatele s informačním systémem.

Pro návrh UI informačních systémů SŽ platí následující zásady:

- standardní ovládací prvky
- uživatelské rozhraní jednoduché a přehledné
- konzistentní prostředí
- účelné rozvržení obrazovek
- barvy a písma dle grafického manuálu
- hierarchie daná typograficky
- informování uživatele, co systém právě dělá
- odpovídající tvar a velikost ovládacích prvků
- kódování znaků UNICODE
- datumové položky dle českého standardu „DD.MM.RRRR“
- jednotný vizuální styl (pro některé projekty dle korporátní identity)
- responzivní design webových aplikací

### 1.3.2 Uživatelský prožitek (User Experience, UX)

UX je to, co uživatel pocítí a pamatuje si v důsledku použití aplikace, systému nebo webu. UX musí být bráno v úvahu při vývoji uživatelského rozhraní, vytváření informační architektury a testování použitelnosti informačních systémů SŽ. Po určení cílového publika a charakteristiky uživatelů je požadováno vytvořit seznam UX požadavků na projekt.

UX informačních systémů SŽ musí mít následující vlastnosti:

- cílem je efektivní uživatel
- návodné ovládání
- ergonomie
- jednoduché, intuitivní
- pravidla přístupnosti, tam kde je požadováno
- zobrazování relativních a požadovaných dat
- rychlost odezvy (doba zpracování požadavku od uživatele by na serveru neměla přesáhnout 0,5s, tak aby celková doba odezvy uživatelský ovládacích prvků byla kratší než 0,8s. V případě, že je předpokládán čas odezvy delší než 0,8s, ale kratší než 2s

---

<sup>1</sup> NSESSS, <https://www.mvcr.cz/clanek/narodni-standard-pro-elektronicke-systemy-spisove-sluzby.aspx>



bude uživateli zobrazen wait cursor a pokud bude předpokládán čas odezvy delší než 2s bude pro informaci uživatele použit progress bar zobrazující průběh operace.)

- použití lazy loading v odůvodněných případech
- jednotná terminologie v celém systému
- ne všechno na jedné obrazovce
- ne všechno v rozbalovacím menu (příliš mnoho položek)
- navigace, kde se uživatel v aplikaci nachází
- minimalizace použití dlouhých textů
- vhodné využití grafických a obrazových prvků
- nepoužívat drobný text
- pečlivé plánování dialogů (logické skupiny)
- ne překrývající se dialogy
- jednotné, stejné ovládací prvky v dialogích na stejných místech s popisky s jednotnou terminologií

## 1.4 Bezpečnost

Všechny vyvíjené aplikace musejí splňovat požadavky kladené platnou legislativou.

Z pohledu požadavků na vyvíjený software je nutné zajistit oblasti:

- Zálohování a obnova
- Bezpečnost komunikací
- Řízení přístupu
- Ochrana před škodlivým kódem
- Logování a monitoring
- Bezpečné předávání a výměna informací
- Akvizice, vývoj a údržba

### 1.4.1 Zabezpečení aplikací

Je požadováno, aby jednotlivé vrstvy splňovaly minimálně tyto požadavky:

- Ke komunikaci mezi jednotlivými vrstvami je používán systémový účet, který lze v případě ohrožení kybernetické bezpečnosti deaktivovat, nebo změnit.
- Systémový účet, který je využíván ke komunikaci mezi vrstvami není privilegovaným účtem.
- Všechny vrstvy jsou ošetřeny proti nejzávažnějším bezpečnostním rizikům jako jsou<sup>2</sup>:
  - Injection
  - Broken Authentication
  - Sensitive Data Exposure
  - XML External Entities (XXE)
  - Broken Access Control
  - Security Misconfiguration
  - Cross-Site Scripting (XSS)
  - Insecure Deserialization
  - Using Components with Known Vulnerabilities
  - Insufficient Logging&Monitoring
- Jednotlivé vrstvy uchovávají své konfigurační parametry v šifrované podobě.

### 1.4.2 Autentizace a autorizace

#### 1.4.2.1 Autentizace

Autentizace je proces ověření proklamované identity subjektu. Je požadováno, aby aplikace umožňovala následující typy autentizace:

---

<sup>2</sup> Dle aktuálního seznamu nejzávažnějších bezpečnostních rizik definovaných OWASP (<https://owasp.org/>).



- SSO (Single Sign-On), autentizaci pomocí protokolu Kerberos, nebo OpenID proti Active Directory
- Manuální přihlášení, autentizaci pomocí vyvíjeného software, tzn. Uživatelská jména a hesla jsou uložena v databázi v šifrované podobě.
- Autentizaci pomocí protokolu LDAP, proti Active Directory
- 2FA

#### **1.4.2.2 Autorizace**

Je požadováno, aby vyvíjený software obsahoval vlastní autorizační modul, který bude minimálně umožňovat:

- Vytváření uživatelských účtů
- Vytváření rolí
- Přidělování jednotlivých uživatelských účtů k rolím
- Přidělování konkrétních oprávnění na role

V rámci naplnění povinností vyplývajících ze zákona č. 181/2014 Sb. a vyhlášky č. 82/2018 Sb. je požadováno, aby vyvíjený software umožňoval správu uživatelů a rolí pomocí externího nástroje na řízení identit, tj. Identity managementem implementovaným ve Správě železnic. Integrace mezi vyvíjeným softwarem a Identity management bude realizována prostřednictvím integrační vrstvy vyvíjeného software.

#### **1.4.3 GDPR**

Je požadováno kompletní splnění všech požadavků na zpracování osobních údajů dle zákona č. 110/2019 Sb. Analýza a návrh opatření musí být řešen již v rámci návrhu řešení.

### **1.5 Dokumentace**

Je požadováno, aby součástí dodávky vyvíjeného software byla dokumentace, a to minimálně v rozsahu:

#### **1.5.1 Technická dokumentace jádra systému**

Dokumentace jádra systému, jeho funkcí, služeb a rozhraní. Dokumentace bude obsahovat kompletní popis architektury jádra systému, výčet a podrobný popis všech jeho funkcí, přehled a popis služeb, které jádro poskytuje dalším komponentám systému, modulům a knihovnám.

#### **1.5.2 E-R modely databáze**

Kompletní dokumentace ve formě E-R schémat pro všechny implementované databáze včetně korespondujících DDL SQL skriptů.

#### **1.5.3 Objektový model pro aplikace**

Dokumentace obsahující objektové modely všech funkcí, jejich komponent, modulů, vztahů.

#### **1.5.4 Procesní diagramy, schémata toků dat**

Dokumentace obsahující procesní diagramy a mapu všech toků dat celého řešení.

#### **1.5.5 Komunikační rozhraní**

Dokumentace všech typů komunikačních rozhraní, všech jejich registrovaných služeb a všech funkcí, struktur dat a vlastností těchto služeb.

#### **1.5.6 Drátové modely všech obrazovek uživatelského rozhraní aplikací**

Dokumentace všech částí software musí obsahovat drátové modely všech obrazovek uživatelského rozhraní včetně popisu funkcí prvků každé obrazovky.

### 1.5.7 Popis konfigurace provozního prostředí

Dokumentace musí obsahovat soupis všech požadavků na nastavení hardwarových a softwarových komponent běhového prostředí jako jsou:

- mapování souborových systémů
- požadavky na operační paměť a počty jader
- konfigurační parametry jednotlivých podpůrných SW prostředků (např. specifika pro nastavení databáze, aplikačního serveru, webového serveru, apod.)

### 1.5.8 Uživatelská příručka

Příručka bude distribuována uživatelům. Musí obsahovat kompletní popis všech uživatelských funkcí pro práci se software. Příručka bude využívána jako základní materiál pro školení nových uživatelů. Příručka musí obsahovat kvalitně a jednoznačně zpracovaný popis kroků pro jednotlivé implementované funkce s vhodným doprovodným obrazovým materiálem ve formě výřezů obrazovek. Musí být napsána v českém jazyce a před finálním odevzdáním zpracovaná jazykovým korektorem.

### 1.5.9 Příručka administrátora

Příručka bude distribuována úzké skupině uživatelů, administrátorům systému. Musí obsahovat kompletní popis všech funkcí pro práci s administrací software. Příručka bude využívána jako materiál pro školení nových administrátorů. Příručka musí obsahovat kvalitně a jednoznačně zpracovaný popis kroků pro jednotlivé implementované funkce s vhodným doprovodným obrazovým materiálem ve formě výřezů obrazovek. Musí být napsána v českém jazyce a před finálním odevzdáním zpracovaná jazykovým korektorem.

## 1.6 Předávání vývoje do provozu

Pokud nebude určeno jinak, veškeré výstupy (zdrojové kódy, konfigurační soubory, testovací data, dokumentace atp.) musejí být předávány prostřednictvím určeného repozitáře.