# Smlouva o dílo

č. UKRUK/129169/2023

(dále jen „***Smlouva***“)

## Smluvní strany:

## Univerzita Karlova

se sídlem: Ovocný trh 560/5, 116 36 Praha 1

zastoupená: Mgr. Martinem Maňáskem, kvestorem

IČO: 00216208, DIČ: CZ00216208

bank. spojení: Česká spořitelna, a.s., pobočka v Praze 1, č. účtu: 909909339/0800

ID datové schránky: piyj9b4

(dále jen „***Objednatel***“)

a

## MEDIA FACTORY Czech Republic a.s.

se sídlem: Žerotínova 1133/32, 130 00 Praha 3 - Žižkov

registrovaný: Městským soudem v Praze

zastoupený: Liborem Olexou, členem představenstva a Dianou Novotnou, členkou představenstva

IČO: 26288311, DIČ: CZ26288311

tel.: 602 321 870

bank. spojení: ......, a. s., č. účtu: .....

ID datové schránky: d34e5j5

(dále jen „***Dodavatel***“)

(dále společně Objednatel a Dodavatel jako „***smluvní strany***“)

uzavřely v souladu s ustanovením § 1746 odst. 2 a § 2358 an. zákona č. 89/2012 Sb., občanského zákoníku, ve znění pozdějších předpisů (dále jen „***občanský zákoník***“), a podle ust. § 61 an. zákona č. 121/2000 Sb., autorský zákon, v platném znění (dále jen „***autorský zákon***“), tuto Smlouvu takto:

## 1. Úvodní prohlášení

1.1. Univerzita Karlova (dále též „***UK***“), v souladu se zákonem č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů, v platném znění (zákon o vysokých školách), jako Objednatel prohlašuje, že je oprávněna uzavřít a řádně plnit tuto Smlouvu a závazky v ní obsažené.

1.2. Společnost MEDIA FACTORY Czech Republic a.s. jako Dodavatel prohlašuje, že je právnickou osobou řádně založenou a zapsanou podle českého právního řádu v obchodním rejstříku vedeném Městským soudem v Praze oddíl B vložka 10575, a že splňuje veškeré podmínky a požadavky v této Smlouvě stanovené a je oprávněn tuto Smlouvu uzavřít a řádně plnit závazky v ní obsažené.

1.3. Dodavatel prohlašuje, že ve smyslu § 4b) zákona č. 159/2006 Sb., o střetu zájmů, ve znění pozdějších předpisů (dále jen „***zákon o střetu zájmů***“), není a ani jeho poddodavatelé nejsou obchodní společností, ve které veřejný funkcionář uvedený v § 2 odst. 1 písm. c) zákona o střetu zájmů nebo jím ovládaná osoba, vlastní podíl představující alespoň 25% účasti společníka v obchodní společnosti.

1.4. Dodavatel prohlašuje, že se na něj, jeho poddodavatele a ani na jím nabízené plnění nevztahují mezinárodní sankce ve smyslu § 2 zákona č. 69/2006 Sb., o provádění mezinárodních sankcí, ve znění pozdějších předpisů.

## 2. Východiska a účel Smlouvy

2.1. Tato Smlouva je uzavřena na základě výsledku veřejné zakázky nazvané *„RUK - ÚVT - Dodávka modulu Osobní údaje Studijního informačního systému na Univerzitě Karlově“* (dále jen „***Veřejná zakázka***“) zadávané v dynamickém nákupním systému s názvem *„RUK – ÚVT – Dynamický nákupní systém na rozvoj studijního informačního systému Univerzity Karlovy“* zavedeném v souladu s § 139 zákona č. 134/2016 Sb., o zadávání veřejných zakázek, ve znění pozdějších předpisů (dále jen „***ZZVZ***“).

2.2. Záměrem Objednatele je dodávka modulu Studijního informačního systému UK (dále též „SIS“ nebo též „SIS UK“) Osobní údaje (dále jen „Software“ nebo též „dílo“), včetně získání potřebných licencí k jeho užití, pro účely popsané v Příloze č. 1a, 1b a 3 Smlouvy a jeho uvedení do provozu na UK a dodání dokumentace, jak je uvedeno dále v této Smlouvě.

2.3. Objednatel prohlašuje, že Software je součástí SIS UK a představuje tedy významný informační systém dle § 2 písm. d) zákona č. 181/2014 Sb., o kybernetické bezpečnosti (dále jen „ZoKB").

2.4. Účelem této Smlouvy je určit podmínky vytvoření díla a rozsah služeb, které Dodavatel dodá Objednateli a které Dodavatel bude poskytovat Objednateli v souvislosti s převzetím a nasazením Software u Objednatele.

## 3. Předmět a místo plnění

3.1. Dílo, které se Dodavatel touto Smlouvou zavazuje pro Objednatele zpracovat, a služby, které se Dodavatele zavazuje poskytovat Objednateli, dle bodu 2.4. Smlouvy, budou zahrnovat:

a) dodání Software splňujícího požadavky Objednatele specifikované v Příloze č. 1a, č. 1b a č. 3 Smlouvy, včetně implementace datových rozhraní a integrace, unit testů, jeho dokumentaci a jeho instalaci a konfiguraci na hardwarovém vybavení Objednatele, včetně dodání, instalace a konfigurace programového vybavení třetích stran, které je nezbytné pro provoz Software u Objednatele,

b) systémové a systémové integrační testování, zátěžové testování a uživatelské akceptační testování,

c) součinnost s Objednatelem a třetími stranami při vytváření rozhraní Software na informační systémy Objednatele,

d) technická pomoc a konzultace poskytované Objednateli při testování Software a jeho rozhraní,

e) zaškolování správců Software v souladu s požadavky definovanými v Příloze č. 1a této Smlouvy,

f) případné další služby, které byly součástí nabídky Dodavatele,

g) záruku na celé dílo dle technických požadavků.

3.2. Objednatel se touto Smlouvou zavazuje poskytnout součinnost uvedenou v této Smlouvě a zavazuje se platit Dodavateli cenu podle této Smlouvy.

## 4. Závazky Dodavatele

4.1. Dodavatel zaručuje, že dílo jím vytvořené pro Objednatele a služby jím poskytované podle této Smlouvy budou na profesionální úrovni, budou v souladu se zadáním uvedeném v bodě 3.1. této Smlouvy a budou odpovídat všeobecně uznávanému standardu a legislativním povinnostem relevantním pro Objednatele. Dodavatel se zavazuje, že jeho pracovníci budou při plnění předmětu Smlouvy na pracovištích Objednatele dodržovat relevantní vnitřní předpisy a normy Objednatele za předpokladu, že s nimi byl Dodavatel prokazatelně seznámen.

4.2. Dodavatel odpovídá za časové a obsahové plnění této Smlouvy, pokud Objednatel včas splní své závazky dle čl. 5. této Smlouvy.

4.3. Dodavatel neodpovídá za poruchy způsobené třetí osobou nemající žádný vztah k Dodavateli nebo událostí, za kterou tato osoba odpovídá, nebo za poruchy způsobené okolnostmi vylučujícími odpovědnost podle § 2913 odst. 2 občanského zákoníku.

4.4. Dodavatel může k částem plnění předmětu Smlouvy použít třetí strany (poddodavatele). V tomto případě je Dodavatel:

a) povinen písemně sdělit Objednateli předem identifikační údaje každého poddodavatele a jeho úkoly dříve, než příslušný poddodavatel zahájí svou činnost,

b) odpovědný Objednateli za příslušnou část plnění a dodržování všech k příslušné části plnění vztahujících se závazků uvedených v této smlouvě stejně, jako kdyby příslušnou část plnění zajišťoval sám.

4.5. Dodavatel se zavazuje poskytnout potřebnou instruktáž týkající se práce se Software správcům Software. Podrobné podmínky školení jsou vymezeny v Příloze č. 1a Smlouvy.

4.6. Dodavatel bere na vědomí, že podle § 2 písm. e) zákona č. 320/2001 Sb., o finanční kontrole ve veřejné správě, v platném znění, je osobou povinnou spolupůsobit při výkonu finanční kontroly. Tato povinnost se týká rovněž těch částí nabídky Dodavatele, Smlouvy a souvisejících dokumentů, které podléhají ochraně podle zvláštních právních předpisů (např. jako obchodní tajemství, utajované informace), za předpokladu, že budou splněny požadavky kladené právními předpisy (např. zákonem č. 255/2012 Sb., o kontrole /kontrolní řád/, v platném znění). Dodavatel bere na vědomí, že obdobnou povinností je dodavatel povinen smluvně zavázat také své poddodavatele.

4.7. Dodavatel se zavazuje, že poskytne součinnost při kontrolách subjektům, které jsou oprávněny ke kontrole dotačních prostředků.

4.8. Dodavatel přebírá nebezpečí změny okolností ve smyslu ustanovení § 2620 odst. 2 zákona č. 89/2012 Sb., občanského zákoníku.

4.9. Dodavatel se zavazuje udržovat v platnosti a účinnosti po celou dobu účinnosti této smlouvy pojistnou smlouvu, jejímž předmětem je pojištění odpovědnosti za škodu způsobenou Dodavatelem třetí osobě (zejména Objednateli) a to tak, že limit pojistného plnění vyplývající z pojistné smlouvy nesmí být nižší než 10 000 000 Kč (slovy: deset milionů korun českých). Objednatel si v průběhu platnosti této Smlouvy může kdykoli vyžádat prokázání splnění této povinnosti předložením pojistné smlouvy nebo pojistného certifikátu, přičemž Dodavatel je v takovém případě povinen předložit doklady prokazující splnění této podmínky do 5 kalendářních dnů od písemného vyžádání Objednatelem.

4.10. Dodavatel je povinen zajistit, že plnění předmětu Smlouvy bude realizovat za podmínek uvedených v Příloze č. 3 Smlouvy, tj. zejména, že na plnění předmětu Smlouvy se budou podílet veškeré osoby Dodavatele identifikované v Příloze č. 3 Smlouvy, že tyto osoby Dodavatele budou plnit činnosti uvedené v Příloze č. 3 Smlouvy a že při plnění předmětu této Smlouvy bude postupovat v souladu s informacemi uvedenými v Příloze č. 3 Smlouvy.

4.11. Dodavatel je povinen zajistit, aby skutečnosti deklarované v prohlášeních dle bodů 1.3. a 1.4. této Smlouvy platily po celou dobu trvání Smlouvy. Objednatel je v případě kolize kterékoliv skutečnosti deklarované v prohlášení dle bodu 1.3. nebo 1.4 této Smlouvy se skutečným stavem podle své volby oprávněn:

a) stanovit Dodavateli písemně lhůtu ke zjednání nápravy,

b) vypovědět Smlouvu s účinností výpovědi okamžikem jejího doručení Dodavateli nebo

c) odstoupit od Smlouvy.

Výše uvedená oprávnění dle písm. b) a c) je Objednatel oprávněn uplatnit i v případě, že v jím dle písm. a) stanovené lhůtě Dodavatel nápravu nezjedná.

4.12. Pokud by došlo ke kolizi skutečného stavu s prohlášeními dle bodů 1.3. a 1.4 této Smlouvy, zavazuje se Dodavatel o této skutečnosti písemně informovat Objednatele, a to ve lhůtě 2 pracovních dnů ode dne, kdy se o této skutečnosti dozvěděl nebo při vynaložení rozumně očekávatelného úsilí dozvědět měl.

4.13. Dodavatel souhlasí, že pokud porušením povinností stanovených v bodech 4.10. nebo 4.11. vznikne Objednateli škoda nebo újma, je povinen tuto v plném rozsahu Objednateli nahradit. Dodavatel dále v této souvislosti souhlasí, že pokud by úhradou sjednané ceny nebo dotčené části ceny měl Objednatel porušit povinnost stanovenou obecně závazným právním předpisem včetně přímo aplikovatelného práva EU, pak Dodavatel nemá na

úhradu sjednané ceny nebo její dotčené části nárok, a to bez ohledu na to, zda a v jakém rozsahu již poskytnul Objednateli plnění.

## 5. Závazky Objednatele

5.1. Pro úspěšný průběh poskytování služeb dle této Smlouvy se Objednatel zavazuje k poskytnutí součinnosti podle zdůvodněných požadavků Dodavatele. Součinnost Objednatele bude zahrnovat:

a) zpřístupnění a zajištění provozu hardwarového vybavení pro instalaci a provoz Software u Objednatele, a to minimálně v testovací instanci, a to včetně poskytnutí licencí Operačních systémů a virtualizační platformy v případě využití technologií podporovaných ze strany Objednatele, které jsou definovány v Příloze č. 2 Smlouvy.

b) spolupráci na řízení prací, blíže specifikovanou v čl. 6. této Smlouvy,

c) spolupráci při přípravě, provádění a vyhodnocení testů Software,

d) další součinnosti, na kterých se Objednatel a Dodavatel písemně dohodnou.

5.2. Objednatel se zavazuje, že poskytne Dodavateli dostatečný a bezpečný přístup do prostor a k systémům Objednatele tak, aby Dodavatel mohl plnit své povinnosti a současně aby Objednatel byl o tomto přístupu informován.

5.3. Objednatel bude udržovat testovací prostředí Software, do něhož bude instalován opravený nebo upravený Software dodaný Dodavatelem pro účely akceptačních testů.

5.4. Objednatel je oprávněn k poskytnutí součinnosti dle bodu 5.1. této Smlouvy, včetně řízení prací na straně Objednatele, využít třetí strany, s nimiž je ve smluvním vztahu. V případě využití třetí strany je Objednatel:

a) povinen písemně sdělit Dodavateli předem identifikační údaje třetí strany a její úkoly dříve, než třetí strana zahájí svou činnost,

b) oprávněn předat této třetí straně důvěrné informace dle bodu 5.1. této Smlouvy nezbytné pro poskytování součinnosti, je však povinen zakotvit povinnost utajení takto předaných důvěrných informací ve smlouvě uzavřené s touto třetí stranou.

## 6. Spolupráce Objednatele a Dodavatele

6.1. Každá ze smluvních stran jmenuje svého zplnomocněného zástupce, Vedoucího týmu, který ji bude výlučně zastupovat v realizačních záležitostech souvisejících s plněním této Smlouvy. Jména a kontakty Vedoucích týmů za Objednatele a Poskytovatele, složení týmu Poskytovatele jsou uvedeny v Příloze č. 4 Smlouvy.

6.2. Vedoucí týmu na straně Dodavatele odpovídá za řízení činnosti případných poddodavatelů.

6.3. Vedoucí týmu na straně Objednatele odpovídá za řízení činnosti případných třetích stran, jejichž součinnost je nezbytná pro úspěšné plnění závazků Objednatele dle této Smlouvy.

6.4. Smluvní strany zajistí svým zplnomocněným zástupcům dle bodu 6.1. této Smlouvy dostatečné pravomoci pro výkon jejich činností.

6.5. Objednatel je oprávněn jmenování svého zplnomocněného zástupce dle bodu 6.1. této Smlouvy změnit, je však povinen o takové změně předem písemně informovat Dodavatele.

6.6. Změnu ve jmenování svého zplnomocněného zástupce dle bodu 6.1. této Smlouvy a náhradu dalších členů realizačního týmu Dodavatele identifikovaných v Příloze č. 3 Smlouvy je Dodavatel oprávněn provést pouze po vzájemné písemné dohodě smluvních stran a po předchozím doložení zkušeností nově jmenovaného zástupce nebo nového člena realizačního týmu Dodavatele. Zkušenosti nově jmenovaného zástupce nebo nového člena realizačního týmu Dodavatele musí odpovídat v celém rozsahu doloženým zkušenostem nahrazovaného zástupce/člena realizačního týmu Dodavatele, jenž jsou uvedeny v nabídce Dodavatele předložené ve Veřejné zakázce a/nebo v Příloze č. 3 Smlouvy.

6.7. Objednatel si vyhrazuje právo zřizovat po dobu platnosti této Smlouvy podle potřeby organizační struktury projektu a v případech, kdy to bude nezbytné pro plnění závazků Dodavatele vyplývajících z této Smlouvy, požadovat zastoupení Dodavatele v těchto strukturách.

## 7. Komunikace mezi smluvními stranami

7.1. Nebude-li smluvními stranami dohodnuto jinak, budou spolu smluvní strany komunikovat:

a) písemně poštou na adresy uvedené v záhlaví této Smlouvy,

b) elektronickou poštou mezi zplnomocněnými zástupci,

c) osobně prostřednictvím zplnomocněných zástupců,

7.2. Všechna oznámení mezi smluvními stranami, která se vztahují k této Smlouvě nebo která mají být učiněna na základě této Smlouvy, musí být učiněna v písemné podobě a druhé straně doručena buď osobně, nebo doporučeným dopisem či jinou formou doporučeného poštovního styku, nebo prostřednictvím informačního systému datových schránek není-li dohodnuto mezi smluvními stranami jinak.

7.3. Písemnost, která má být dle této smlouvy doručena druhé straně (oznámení, výpověď, odstoupení od smlouvy, reklamace vad apod.), je doručena dnem jejího převzetí Vedoucím týmu druhé smluvní strany nebo dnem, kdy byla doručena osobně nebo prostřednictvím držitele poštovní licence do sídla této smluvní strany, nebo doručením do datové schránky Objednatele.

7.4. Komunikace elektronickou poštou je považována za doručenou okamžikem potvrzení převzetí přijímající stranou, resp. okamžikem odpovědi přejímající strany.

7.5. Komunikace mezi smluvními stranami bude probíhat v českém jazyce.

## 8. Termíny plnění a přejímací postupy

8.1. Klíčové termíny jsou stanoveny v Příloze č. 3 této Smlouvy.

8.2. Dodavatel se zavazuje písemně upozornit Objednatele na skutečnost, že bez poskytnutí součinnosti ve smyslu odst. 5.1. této Smlouvy ze strany Objednatele není schopen pokračovat v plnění této Smlouvy a z tohoto důvodu hrozí nedodržení klíčových termínů stanovených v odst. 8.1. této Smlouvy. Písemné upozornění musí obsahovat konkrétní popis činností, které je Objednatele povinen v rámci své součinnosti vykonat, a odůvodnění, proč bez poskytnutí této součinnosti ze strany Objednatele není Dodavatel schopen v plnění této Smlouvy pokračovat. V případě, že ani do 5 pracovních dnů od doručení upozornění ze strany Dodavatele Objednatel neposkytne požadovanou součinnost, má Dodavatel nárok na prodloužení klíčových termínů stanovených v odst.

8.1. této Smlouvy, a to o dobu, po kterou je Objednatel v prodlení s poskytnutím požadované součinnosti.

8.3. Předání licencí a Software provede Dodavatel tak, že Software nainstaluje v testovacím prostředí Objednatele. Součástí předání bude rovněž předání vybraných částí dokumentace Software dle Přílohy č. 1a Zadávací dokumentace.

8.4. Počínaje datem předání dle bodu 8.3. této Smlouvy začíná běžet akceptační lhůta v délce třiceti (30) kalendářních dnů. V této lhůtě proběhnou uživatelské akceptační testy Software a revize dodané dokumentace. Na závěr Objednavatel vypracuje protokol o provedených testech, který bude podkladem pro vydání akceptačního rozhodnutí.

8.5. Podmínkou pro akceptaci Software předávaného jako součást plnění je, že předávaný Software nevykazuje chybu bránící používání základních funkcí Softwaru (lze projít bezchybně všechny základní funkce Software), vykazuje nejvýše jednu (1) chybu v základní funkčnosti, kde lze využit jiný způsobem využití Softwaru se stejným výsledkem, a nejvýše pět (5) chyb drobného rozsahu (názvy polí, tlačítek, barevné schéma aj.). Součástí akceptačního protokolu bude výčet chyb nebránících akceptaci spolu s termíny dohodnutými smluvními stranami pro jejich odstranění.

8.6. Do uplynutí akceptační lhůty může Objednatel předané plnění odmítnout, pokud se prokáže jeho nesoulad s legislativními požadavky nebo požadavky Objednatele uvedenými v Příloze č. 1a, č. 1b a č. 3 této Smlouvy, nebo nebudou splněny podmínky uvedené v bodě 8.5. této Smlouvy,

8.7. Rozhodnutí o odmítnutí dle bodu 8.6. této Smlouvy bude mít písemnou formu a bude doručeno v akceptační lhůtě Dodavateli. Neobdrží-li Dodavatel v akceptační lhůtě písemné vyrozumění Objednatele o odmítnutí, je Objednatel povinen podepsat ke dni uplynutí akceptační lhůty akceptační protokol s rozhodnutím ve významu „akceptováno“.

8.8. Počínaje datem doručení písemného vyrozumění Objednatele o odmítnutí dle bodu 8.7. této Smlouvy začíná běžet opravná lhůta v délce čtrnácti (14) kalendářních dnů. Jestliže během opravné lhůty Dodavatel odstraní vady plnění, které byly důvodem odmítnutí, a prokáže tuto skutečnost Objednateli, je Objednatel povinen podepsat akceptační protokol. Marné uplynutí opravné lhůty se považuje za porušení této Smlouvy podstatným způsobem.

## 9. Cenové a platební podmínky

9.1. Cena za dílo a za poskytnutí služeb dle této Smlouvy je cenou pevnou, maximální a nejvýše přípustnou, zahrnuje všechny náklady Dodavatele, jeho veškeré práce, včetně přiměřeného zisku.

9.2. Cena je uvedena v Příloze č. 2 této Smlouvy.

9.3. Fakturu doručí Objednateli po podepsání akceptačního protokolu dle bodu 8.5. této Smlouvy za příslušné plnění. Přílohou faktury bude akceptační protokol podepsaný zástupci obou smluvních stran. Dodavatel bude fakturovat jednorázově, a to po akceptaci plnění dle bodu 8.5. této Smlouvy.

9.4. Každá faktura vystavená Dodavatelem dle této Smlouvy bude vystavena jako daňový doklad se zúčtováním DPH podle předpisů platných k datu zdanitelného plnění a musí mít náležitosti stanovené příslušnými právními předpisy pro daňový doklad. Splatnost faktury bude třicet (30) kalendářních dnů od prokazatelného doručení faktury

Objednateli. Dnem uhrazení faktury je den, kdy byla příslušná částka odepsána z účtu Objednatele.

9.5. Faktura Dodavatele musí být vystavena v souladu s touto Smlouvou a musí mít náležitosti daňového dokladu dle zákona č. 235/2004 Sb., ve znění pozdějších předpisů, zejména:

a) evidenční číslo daňového dokladu,
b) název a sídlo Objednatele a Dodavatele,
c) číslo Smlouvy a den jejího uzavření,
d) název projektu: Transformace pro VŠ na UK, registrační číslo: NPO_UK_MSMT16602/2022, SC A1
e) datum vystavení daňového dokladu a datum uskutečnění zdanitelného plnění,
f) označení banky a číslo účtu, na který má být zaplaceno a který je registrován u příslušného správce daně a je zveřejněn způsobem umožňujícím dálkový přístup ve smyslu zákona č. 235/2004 Sb., o dani z přidané hodnoty, v platném znění,
g) jednotkovou cenu bez daně a slevu, není-li obsažena v jednotkové ceně, základ daně, sazbu daně a její výše, pokud nejde o plnění dle ust. § 92e zákona č. 235/2004 Sb., o dani z přidané hodnoty, v platném znění,
h) číselný kód klasifikace CZ – CPA, a v případě plnění dle ust. § 92e zákona o DPH poznámku „daň odvede zákazník“,
i) čísla a data vyhotovení soupisů skutečně a řádně provedených prací a zjišťovacích protokolů,
j) IČO a DIČ Dodavatele a Objednatele.

9.6. Objednatel se zavazuje proplatit v termínu každou fakturu vystavenou Dodavatelem v souladu s ustanovením bodů 9.4. a 9.5. této Smlouvy. Nesprávně nebo neúplně vyplněnou fakturu je Objednatel oprávněn vrátit Dodavateli k opravě, po tuto dobu neběží doba splatnosti faktury. Po prokazatelném doručení bezchybné faktury Objednateli počíná běžet nová lhůta splatnosti.

9.7. V případě, že se Dodavatel stane nespolehlivým plátcem ve smyslu § 106a zákona č. 235/2004 Sb., o dani z přidané hodnoty, v platném znění, je povinen o tom neprodleně písemně informovat Objednatele. Bude-li Dodavatel ke dni uskutečnění zdanitelného plnění veden jako nespolehlivý plátce, bude část ceny za služby dle této Smlouvy odpovídající dani z přidané hodnoty uhrazena přímo na účet správce daně v souladu s ust. § 109a zákona č. 235/2004 Sb., o dani z přidané hodnoty, v platném znění. O tuto částku bude ponížena celková cena a Dodavatel obdrží cenu dle této Smlouvy bez DPH. V případě, že se Dodavatel stane nespolehlivým plátcem ve smyslu tohoto bodu, má Objednatel současně právo od této Smlouvy odstoupit.

## 10.Sankce

10.1. V případě, že Dodavatel nesplní některý z klíčových příloha ů dle bodu 8.1. této Smlouvy z důvodů ležících na straně Dodavatele (tj. v případě, kdy část plnění nebyla Dodavatelem předána nebo byla Objednatelem odmítnuta), je Objednatel oprávněn účtovat Dodavateli smluvní pokutu ve výši 10 % z ceny za příslušné plnění dle bodu 9.2. této Smlouvy, a to za každý i započatý kalendářní den uplynuvší mezi příslušným termínem a dnem akceptace příslušné části plnění nebo dnem odstoupení Objednatele od Smlouvy dle bodů 18.2. nebo 18.3. této Smlouvy. V případě, že k nesplnění termínu

došlo z důvodu odmítnutí části plnění Objednatelem, se do doby rozhodné pro výpočet sankce nezapočítávají dny, kdy probíhalo testování nebo posuzování dodané části plnění na straně Objednatele.

10.2. Smluvní pokuty dle bodu 10.1. této Smlouvy mohou být uplatněny nejvýše do souhrnné částky ve výši 100 % z ceny, která je uvedena v Příloze č. 2 této Smlouvy.

10.3. Pokud dodavatel nepředloží platnou pojistnou smlouvu nebo pojistný certifikát prokazující podmínku dle bodu 4.14. této Smlouvy ve stanovené lhůtě 5 kalendářních dnů, je Objednatel oprávněn účtovat Dodavateli smluvní pokutu 10 000,- Kč (slovy deset tisíc korun českých) za každý i započatý kalendářní den prodlení. Výše smluvní pokuty dle tohoto bodu není omezena.

10.4. V případě, že Dodavatel poruší bod 4.10 této Smlouvy je Objednatel oprávněn účtovat Dodavateli smluvní pokutu ve výši 10 000,- Kč (slovy deset tisíc korun českých) za každé jednotlivé porušení.

10.5. Uplatní-li Objednatel svá práva dle bodu 10.1. nebo 10.3. této Smlouvy, bude smluvní pokutu uplatňovat na Dodavateli s lhůtou splatnosti třicet (30) dnů ode dne doručení výzvy Objednatele k zaplacení smluvní pokuty Dodavateli.

10.6. Uplynutím lhůty dle bodu 10.5. této Smlouvy je Objednatel oprávněn smluvní pokutu uplatnit formou automatického zápočtu proti první přijaté faktuře od Dodavatele.

## 11.Kybernetická bezpečnost

11.1. Dodavatel tímto bere na vědomí, že Objednatel je osobou povinnou dle zákona č. 181/2014 Sb., o kybernetické bezpečnosti a o změně souvisejících zákonů (dále jen „ZoKB“) a plní povinnosti vyhlášky č. 82/2018 Sb., o bezpečnostních opatřeních, kybernetických bezpečnostních incidentech, reaktivních opatřeních, náležitostech podání v oblasti kybernetické bezpečnosti a likvidaci dat (dále jen „VyKB“).

11.2. Dodavatel tímto bere na vědomí, že je pro Objednatele při zajišťování smluvního vztahu založeného touto Smlouvou v pozici významného dodavatele ve smyslu § 2 písm. n) a § 8 odst. 1 písm. f) a odst. 2 VyKB a v pozici možného budoucího provozovatele ve smyslu § 2 písm. g) VyKB. V případě, že se Dodavatel stane provozovatelem ve smyslu shora uvedeného, bude o tomto informován Objednatelem.

11.3. Objednatel je v souladu s ustanovením § 4 odst. 4 ZoKB a ve spojení s přílohou č. 7 VyKB povinen stanovit závazná bezpečnostní opatření, která se vztahují na Dodavatele při plnění předmětu této smlouvy (dále jen „Bezpečnostní opatření“).

11.4. Dodavatel je povinen v rozsahu plnění této Smlouvy naplnit všechna Bezpečnostní opatření uvedená v Příloze č. 5 této smlouvy – Bezpečnostní požadavky.

11.5. Dodavatel se dále zavazuje:

a. poskytnout na vyžádání Objednateli dokumenty, zprávy, a obdobné vstupy, které budou prokazovat naplnění Bezpečnostních požadavků;
b. na požádání s Objednatelem konzultovat kdykoli v průběhu účinnosti této Smlouvy detailní nastavení bezpečnostních opatření k naplnění Bezpečnostních požadavků a pro takovéto konzultace zajistit účast kvalifikovaných pracovníků;
c. neprodleně informovat Objednatele o všech významných změnách v naplnění Bezpečnostních požadavků, které nastanou kdykoli v průběhu účinnosti této Smlouvy;
d. bezodkladně a s vyvinutím nejlepšího úsilí zajistit náhradní způsob naplnění Bezpečnostních požadavků, pokud stávající řešení přestalo být funkční a efektivní;

e. bezodkladně a prokazatelně informovat Objednatele o kybernetických bezpečnostních událostech a incidentech, které mohou ovlivnit poskytování služeb dle této Smlouvy;
f. při výkonu své činnosti včas a prokazatelně upozornit Objednatele na zřejmou nevhodnost jeho příkazů či doporučení vztahujících se k Bezpečnostním opatřením a jejichž následkem může vzniknout újma nebo nesoulad s platnými a účinnými právními předpisy či jinými předpisy vztahujícími se k poskytování služeb dle této Smlouvy.

11.6. Dodavatel bere na vědomí, že veškeré aktivity Dodavatele a jeho plnění realizované v prostředí Objednatele mohou být monitorovány a vyhodnocovány v rozsahu předmětu plnění.

## 12.Ochrana a utajení informací

12.1. Za důvěrné informace se bez ohledu na formu jejich zachycení považují informace tvořící obchodní tajemství podle jiných právních předpisů a dále informace, které Objednatel za důvěrné označí.

12.2. Pro nakládání s osobními údaji, s nimiž Dodavatel přijde do styku v průběhu plnění, a pro ochranu těchto údajů při jejich zpracování platí v plném rozsahu ustanovení zákona č. 110/2019 Sb., o zpracování osobních údajů, ve znění pozdějších předpisů.

12.3. Objednatel poskytne Dodavateli pro plnění předmětu Smlouvy zabezpečený vzdálený přístup do databáze SIS UK, která je umístěna u Objednatele. V této databázi jsou zpracovávány osobní údaje osob, studenty či zaměstnanci Objednatele, a dále ostatní uživatelé SIS UK. Dodavateli bude do této databáze poskytnut přístup nezbytný k plnění předmětu Smlouvy, a to po dobu platnosti a účinnosti Smlouvy. Pracovníci Dodavatele nejsou oprávněni nahlížet na data ve zpřístupněné databázi bez vědomí a bez výslovného souhlasu Objednatele. Dodavatel je povinen zajistit, že vzdálený přístup do databáze budou mít pouze osoby, které jsou v době trvání této smlouvy v pracovněprávním nebo obdobném smluvním vztahu s Dodavatelem, podílejí se na plnění předmětu Smlouvy a jsou písemně zavázány vůči Objednateli povinností mlčenlivosti. Zabezpečený vzdálený přístup Dodavatele do databáze bude omezen na vyjmenované pracovníky Dodavatele, kterým bude Objednatelem přiděleno přístupové jméno a heslo. Zabezpečený vzdálený přístup bude možný pouze z předem dohodnutých síťových adres a omezen na přístupové protokoly dohodnuté mezi zplnomocněnými zástupci Objednatele a Dodavatele.

12.4. Smluvní strany jsou povinny zajistit utajení získaných důvěrných informací způsobem obvyklým pro utajování takových informací, není-li dále v tomto článku výslovně sjednáno jinak. Zavazují se tímto, že podniknou všechny kroky k zabezpečení těchto informací.

12.5. Povinnost oboustranného utajení důvěrných informací platí bez ohledu na ukončení účinnosti této Smlouvy.

12.6. Smluvní strany mají právo požadovat navzájem doložení dostatečnosti utajení důvěrných informací.

12.7. Smluvní strany jsou povinny respektovat veškerá práva a oprávněné zájmy druhé smluvní strany a její obchodní značky, loga a ochranné známky v souladu s právními předpisy a vnitřními předpisy Objednatele.

12.8. Dodavatel se zavazuje, že každou tiskovou zprávu nebo jinou informaci určenou ke zveřejnění a týkající se uzavření této Smlouvy a průběhu jejího plnění předloží ke

schválení a korektuře Objednateli a nebude ji publikovat bez předchozího písemného schválení Objednatelem.

12.9. Objednatel se zavazuje, že návrhy na zveřejnění, které mu Dodavatel předloží v souladu s ustanovením bodu 12.8. této Smlouvy, posoudí bez zbytečného odkladu a nebude Dodavateli bezdůvodně bránit v tom, aby využil skutečnost uzavření této Smlouvy a významné události v průběhu jejího plnění k propagačním účelům.

12.10. Žádné ustanovení této Smlouvy nebrání žádné ze stran v poskytnutí informací třetí straně či ve zveřejnění informací, pokud povinnost poskytnutí těchto informací vyplývá z platných právních předpisů.

12.11. Veškerá data Objednatele, k nimž Dodavatel získá v průběhu platnosti této Smlouvy přístup, jsou považována za důvěrné informace ve smyslu bodu 12.1. této Smlouvy a Dodavatel je nesmí použít k jiným účelům než k plnění předmětu Smlouvy.

## 13.Duševní vlastnictví, práva třetích osob

13.1. Dodavatel poskytuje plněním této Smlouvy Objednateli výhradní časově neomezená užívací práva (licenci) k užití Software podle ust. § 12 autorského zákona.

13.2. Software dle této Smlouvy, včetně jeho oprav, úprav a rozšíření provedených v rámci plnění této Smlouvy, a jeho dokumentace mohou být užity smluvními stranami ke všem způsobům užití potřebným pro provoz Software u Objednatele a pro jeho další rozvoj pro potřeby Objednatele, a to včetně vytváření děl odvozených a začleňování do děl souborných, bez časového omezení a s územním omezením dle bodu 13.1. této Smlouvy.

13.3. Zdrojový kód Software je psán tak, že nekomplikuje provádění modifikací Software, není úmyslně zatemnělý nebo zmatený.

13.4. Dodavatel se zavazuje aktualizovat uživatelskou dokumentaci a technickou dokumentaci Software po změnách provedených v rámci oprav a úprav Software podle této Smlouvy.

13.5. Uživatelskou dokumentaci Software bude Dodavatel po provedení jejích změn dle bodu 13.4. této Smlouvy předávat Objednateli, a to současně s předáním upravené verze Software.

13.6. Objednatel je oprávněn při vytváření děl odvozených a začleňování Software do děl souborných dle bodu 13.2. této Smlouvy využít služeb třetích stran. Toto ustanovení není porušením ustanovení čl. 12. této Smlouvy.

13.7. Dle výslovné dohody Objednatele a Dodavatele je odměna za veškerá licenční oprávnění k autorským dílům, která jsou součástí či příslušenstvím díla či jeho části, již zahrnuta v ceně díla, tzn., že Dodavateli nenáleží žádná další odměna, cena či jakákoli jiná platba nebo náhrada za poskytnutí licence, a to ani v případě její aktualizace, upgradu nebo opravy (service pack).

13.8. Dodavatel se zavazuje nahradit Objednateli škodu za všechny důvodné a přiměřené nároky třetích osob z titulu porušení jejich chráněných práv souvisejících s plněním Dodavatele podle této Smlouvy, pokud Objednatel:

a) oznámí Dodavateli bez zbytečného odkladu písemně a uceleně uplatnění jakéhokoliv podobného nároku třetích osob,

b) neuzná sám předmětný nárok,

c) zplnomocní Dodavatele k vypořádání takového nároku soudní nebo mimosoudní cestou,
d) neučiní bez předchozí konzultace s Dodavatelem jakékoliv právní úkony ve věci předmětných nároků.

## 14.Záruky

14.1. Dodavatel zaručuje, že dílo dle této Smlouvy a každá služba, kterou Dodavatel poskytuje, bude provedena s vynaložením přiměřené odborné péče, znalostí a dovedností a bude odpovídat aktuálnímu popisu příslušné služby (včetně kritérií plnění) obsaženému v této Smlouvě, jejích přílohách nebo jiném příslušném dokumentu.

## 15.Odpovědnost za škodu

15.1. Uplatněním sankce podle čl. 10. této Smlouvy není dotčeno právo poškozené smluvní strany na náhradu škody způsobené porušením povinnosti sankcionované smluvní pokutou, a to i ve výši přesahující tuto smluvní pokutu.

15.2. Smluvní strany se zavazují upozornit druhou smluvní stranu bez zbytečného odkladu na vzniklé okolnosti vylučující odpovědnost a bránící řádnému plnění této Smlouvy. Smluvní strany se dále zavazují k vyvinutí maximálního úsilí k odvrácení a překonání okolností vylučujících odpovědnost.

15.3. Dodavatel odpovídá za škody na zdraví (včetně usmrcení) a na nemovitém a movitém majetku v plné výši.

15.4. Dodavatel odpovídá za jinou skutečnou škodu než výše uvedenou (viz bod 15.3. této Smlouvy), která může vzniknout v rámci plnění Dodavatele podle této Smlouvy.

15.5. Za žádných okolností nebude Dodavatel odpovědný za ztrátu nebo škodu na záznamech či datech Objednatele nebo vadnost těchto záznamů či dat, které prokazatelně nebyly způsobeny vadou plnění Dodavatele či osob využitých Dodavatelem k plnění této smlouvy, a za případné následné škody či újmy takto vzniklé.

15.6. Objednatel je oprávněn prokazatelnou výši škody, za kterou Dodavatel odpovídá dle bodu 15.3. nebo 15.4. této Smlouvy, uplatnit formou automatického zápočtu proti přijatým fakturám od Dodavatele.

## 16.Řešení sporů

16.1. Smluvní strany se zavazují vyvinout maximální úsilí k odstranění vzájemných sporů vzniklých na základě této Smlouvy nebo v souvislosti s touto Smlouvou a k jejich vyřešení zejména prostřednictvím jednání zplnomocněných zástupců dle bodu 6.1. této Smlouvy.

16.2. Jestliže se spory nepodaří vyřešit smírnou cestou, může každá ze stran postoupit spor nejvyšším představitelům smluvních stran. Nejvyšší představitelé se pokusí vyřešit spor smírnou cestou. Případný soudní spor bude řešen věcně a místně příslušným soudem. Rozhodčí řízení se nepřipouští. Rozhodným právem je právo ČR.

## 17. Platnost a účinnost Smlouvy

17.1. Smlouva nabývá platnosti dnem jejího podpisu oběma smluvními stranami, přičemž platí datum posledního podpisu, a účinnosti dnem uveřejnění v registru smluv podle zákona č. 340/2015 Sb., v platném znění. Smlouva se uzavírá na dobu určitou, a to na dobu do okamžiku akceptace díla Objednatele.

17.2. Objednatel i Dodavatel jsou oprávněni odstoupit od této Smlouvy v plném rozsahu v případě porušení některého bodu této Smlouvy druhou smluvní stranou, pokud na toto porušení písemně upozorní a druhá smluvní strana do čtrnácti (14) kalendářních dnů uspokojivě nevysvětlí vzniklou nesrovnalost nebo ji neodstraní. Účinnost Smlouvy je v tomto případě ukončena okamžikem prokazatelného doručení písemného sdělení o odstoupení od Smlouvy druhé smluvní straně.

17.3. Objednatel je oprávněn odstoupit od Smlouvy s okamžitou platností také i v případě porušení Smlouvy podstatným způsobem ze strany Dodavatele (viz body 1.5., 8.8. a 9.7. této Smlouvy).

17.4. Ukončením účinnosti této Smlouvy nejsou dotčena ustanovení týkající se ochrany informací (viz čl. 12. této Smlouvy), ochrany práv Objednatele (viz čl. 13. této Smlouvy), záruky (viz čl. 14. této Smlouvy), řešení sporů ani splatné závazky smluvních stran.

17.5. V případě odstoupení od Smlouvy jednou smluvní stranou dle bodu 17.2. nebo 17.3. této Smlouvy se obě smluvní strany zavazují vyvinout maximální úsilí k dosažení dohody na vzájemném vyrovnání. Tímto ustanovením nejsou dotčena ustanovení týkající se sankcí (viz čl. 10. této Smlouvy) ani odpovědnosti za škodu (viz čl. 15. této Smlouvy).

## 18. Výhrada změny dodavatele

18.1. Objednatel si ve smyslu § 100 odst. 2 ZZVZ vyhrazuje právo realizovat změnu v osobě Dodavatele v průběhu plnění dle této Smlouvy, budou-li splněny následující podmínky:

a) nastanou důvody umožňující Objednateli ukončit tento smluvní vztah výpovědí nebo odstoupením od této Smlouvy pro důvody na straně Dodavatele nebo bude-li tato Smlouva ukončena ze strany Dodavatele před uplynutím doby jejího trvání,

b) Dodavatel bude nahrazen dodavatelem, jehož nabídka se umístila jako další v pořadí při hodnocení nabídek ve Veřejné zakázce (dále jen „Nahrazující dodavatel“),

c) Nahrazující dodavatel prokáže Objednateli splnění všech podmínek účasti ve smyslu § 36 ZZVZ, které byly ve Veřejné zakázce stanoveny,

d) Nahrazující dodavatel splní veškeré další podmínky pro uzavření této Smlouvy ve smyslu § 104 ZZVZ, pokud byly ve Veřejné zakázce tyto další podmínky stanoveny,

e) Nahrazující dodavatel předloží Objednateli originály dokladů o své kvalifikaci, pokud je Objednatel již nemá k dispozici,

f) Objednatel u Nahrazujícího dodavatele zjistí údaje o jeho skutečném majiteli, a to postupem dle § 122 odst. 4 nebo 5 ZZVZ,

g) Nahrazující dodavatel nebo jeho případní poddodavatelé nejsou obchodními společnostmi, ve které veřejný funkcionář uvedený v § 2 odst. 1 písm. c) zákona č. 159/2006 sb., o střetu zájmů, ve znění pozdějších předpisů, nebo, jím ovládaná

osoba, vlastní podíl představující alespoň 25% účasti společníka v obchodní společnosti a

h) na Nahrazujícího dodavatele, na jeho případné poddodavatele ani na jím nabízené plnění se nevztahují mezinárodní sankce ve smyslu § 2 zákona č. 69/2006 Sb., o provádění mezinárodních sankcí, ve znění pozdějších předpisů.

18.2. Pokud se postupem dle bodu 18.1. této Smlouvy nepodaří uzavřít smlouvu s Nahrazujícím dodavatelem, může Objednatel nahradit Dodavatele dodavatelem, jehož nabídka se umístila jako další v pořadí za Nahrazujícím dodavatelem při hodnocení nabídek ve Veřejné zakázce. Při tom musí být rovnocenným způsobem splněny podmínky uvedené v bodě 18.1. této Smlouvy.

18.3. Dodavatel, který nahradí Dodavatele postupem dle bodu 18.1. nebo 18.2. této Smlouvy, bude pokračovat v plnění dle této Smlouvy za podmínek odpovídajících své nabídce předložené ve Veřejné zakázce.

18.4. Realizace změny v osobě Dodavatele proběhne buď cestou ukončení této Smlouvy a uzavřením smlouvy nové, nebo cestou postoupení pohledávky z této Smlouvy ve smyslu § 1879 a násl. zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů.

## 19.Závěrečná ustanovení

19.1. Výkon práv a povinností plynoucích z této Smlouvy se řídí příslušnými ustanoveními občanského zákoníku a dále autorským zákonem.

19.2. Bude-li některé z ustanovení této Smlouvy shledáno jako neplatné nebo nevymahatelné, nemá taková skutečnost vliv na platnost nebo vymahatelnost zbývajících ustanovení této Smlouvy.

19.3. Jestliže smluvní strana v případě neplnění či porušení této Smlouvy neuplatní všechna svá práva v takovém případě jí náležející, nelze takové jednání v žádném případě vykládat jako vzdání se takových práv pro případ jiného či následného neplnění či porušení sjednaných smluvních povinností.

19.4. Žádná smluvní strana není odpovědná druhé smluvní straně za vynaložení nákladů, rizika nebo za závazky vyplývající z činnosti této smluvní strany v souvislosti s předmětem plnění. Každá ze smluvních stran bude jednat jako nezávislý právní subjekt, nikoliv jako zmocněnec druhé smluvní strany.

19.5. V případě změny nebo ukončení závazku z této Smlouvy jsou smluvní strany povinny postupovat v souladu s § 222 a 223 ZZVZ.

19.6. Smlouvu lze měnit pouze oboustranně odsouhlasenými číslovanými dodatky podepsanými oběma smluvními stranami. Žádný jiný protokol, dokument, obvyklá praxe nebo zvyk nebudou považovány za dodatek ke Smlouvě nebo za její pozměnění.

19.7. Obě smluvní strany souhlasí, že:

a) na základě této Smlouvy neuděluje žádná strana druhé smluvní straně právo užívat její ochranné známky či jiná označení (včetně ochranných známek či označení v rámci podniku) pro účely propagace nebo publikování, není-li to oběma smluvními stranami předem písemně dohodnuto;

b) obě smluvní strany jsou oprávněny uzavírat obdobné smlouvy s třetími stranami, za předpokladu, že uzavřením nebo plněním takové smlouvy nebude jakkoliv dotčeno plnění dle této Smlouvy;

c) žádná ze smluvních stran neodpovídá za nesplnění svých závazků, pokud k takovému nesplnění došlo z důvodů okolností vylučujících odpovědnost podle § 2913 odst. 2 občanského zákoníku.

19.8. Dodavatel se zavazuje neuzavírat smlouvy týkající se Software anebo služeb dle této Smlouvy týkající se fakult nebo dalších součástí UK bez předchozí dohody se zplnomocněným zástupcem Objednatele dle bodu 6.1. této Smlouvy.

19.9. Smluvní strany berou na vědomí, že tato Smlouva vyžaduje uveřejnění v registru smluv podle zákona č. 340/2015 Sb., ve znění pozdějších předpisů, a s tímto uveřejněním souhlasí. Zaslání Smlouvy do registru smluv zajistí Objednatel neprodleně po podpisu Smlouvy. Smluvní stany se dohodly, že v rámci smluvního jednání dojde před zveřejněním Smlouvy k dohodě o anonymizaci těch údajů Smlouvy, které nelze zveřejnit postupem dle zákona č. 340/2015 Sb., protože jejich zveřejněním by došlo k porušení jiných právních předpisů. Objednatel se současně zavazuje informovat druhou smluvní stranu o provedení registrace tak, že zašle druhé smluvní straně kopii potvrzení správce registru smluv o uveřejnění Smlouvy bez zbytečného odkladu poté, kdy sám potvrzení obdrží, popř. již v průvodním formuláři vyplní příslušnou kolonku s ID datové schránky druhé smluvní strany (v takovém případě potvrzení od správce registru smluv o provedení registrace Smlouvy obdrží obě smluvní strany zároveň).

19.10. Nedílnou součástí této Smlouvy jsou následující přílohy:

a) Příloha č. 1a – Specifikace a popis předmětu plnění
b) Příloha č. 1b – Obecná technická specifikace
c) Příloha č. 2 – Položkový rozpočet
d) Příloha č. 3 – Koncepce nabízeného řešení
e) Příloha č. 4 – Složení týmu za Objednatele a Poskytovatele
f) Příloha č. 5 – Bezpečnostní požadavky

19.11. Smlouva je uzavírána elektronicky, a to tak, že je opatřena elektronickými podpisy (zaručeným elektronickým podpisem založeným na kvalifikovaném certifikátu nebo kvalifikovaným elektronickým podpisem) oprávněných zástupců smluvních stran.

Smluvní strany prohlašují, že si tuto Smlouvu přečetly, že s jejím obsahem souhlasí a na důkaz toho k ní připojují svoje podpisy.

| | |
|---|---|
| Za MEDIA FACTORY Czech Republic a.s. | Za Univerzitu Karlovu |
| V Praze dne (viz elektronický podpis) | V Praze dne (viz elektronický podpis) |
| ..... | ..... |
| **Libor Olexa, člen představenstva** | **Mgr. Martin Maňásek, kvestor** |

.....

**Diana Novotná, členka představenstva**

# Příloha č. 1a - Specifikace a popis předmětu plnění

Obsah

# 1 Předmět plnění

Modul Osobní údaje je referenčním zdrojem základních informací o všech osobách, které jsou ve studijním informačním systému evidovány. Jde zejména o studenty, účastníky programů celoživotního vzdělávání, uchazeče o studium, vyučující či zaměstnance.

Modul slouží uživatelům k zobrazení jejich osobních údajů, k editaci povolené podmnožiny těchto údajů a také nabízí i základní funkce vyhledávání jiných osob. Jako jeden ze základních modulů je zdrojem dat ostatním modulům, které potřebují evidované osobní údaje pro svou agendu. Modul se skládá z vlastní databáze, uživatelského rozhraní a z aplikačních rozhraní, která slouží ostatním modulům i vlastnímu uživatelskému rozhraní.

Součástí dodávky je implementace řešení, které splňuje požadavky specifikované dokumentací, realizace potřebných testů řešení a jeho nasazení v prostředí UK.

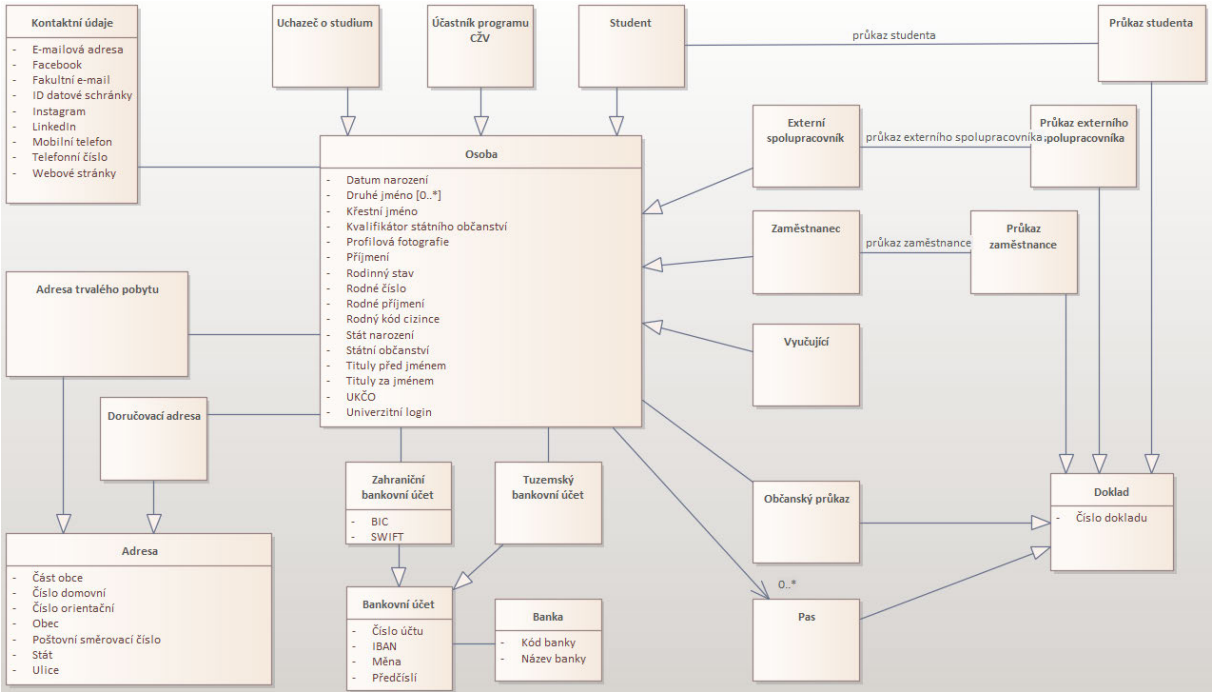
# 2 Byznys analýza modulu Osobní údaje

## 3 Popis modulu Osobní údaje

Modul Osobní údaje je referenčním zdrojem základních informací o všech osobách, které jsou ve studijním informačním systému evidovány. Jde zejména o studenty, účastníky programů celoživotního vzdělávání, uchazeče o studium, vyučující či zaměstnance. Slouží jednak uživatelům k zobrazení jejich osobních údajů, k editaci povolené podmnožiny těchto údajů a také nabízí i základní funkce vyhledávání jiných osob. Dále slouží i ostatním modulům, které potřebují evidované osobní údaje pro svou agendu. Modul se skládá z vlastní databáze, uživatelského rozhraní a z aplikačních rozhraní, která slouží ostatním modulům i vlastnímu uživatelskému rozhraní.

## 4 Konceptuální model

Níže zobrazený konceptuální model graficky znázorňuje entity, jejich atributy a vztahy mezi nimi pro výsek reality realizovaný modulem Osobní údaje.

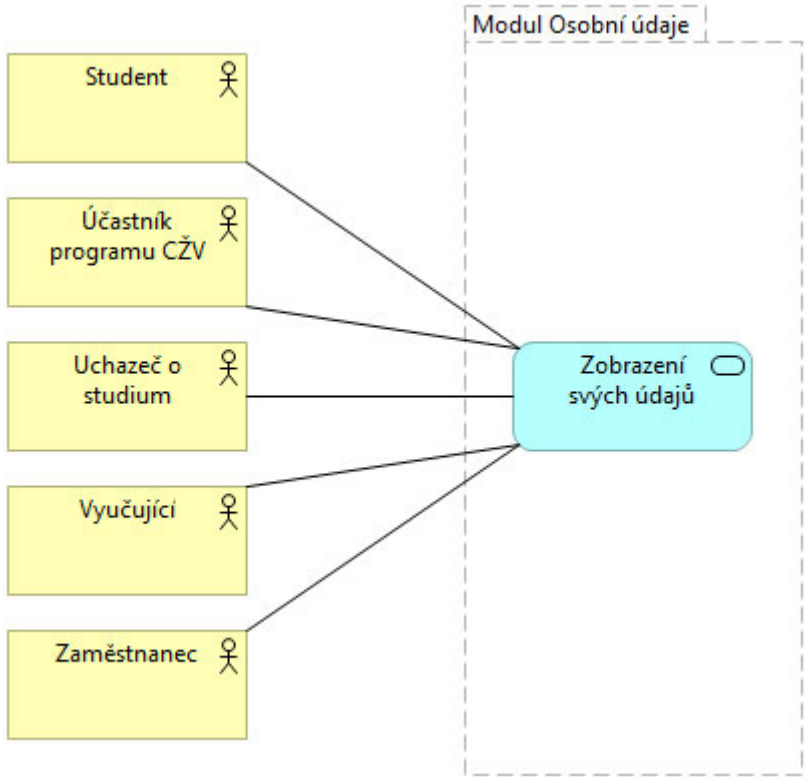



## 5 Zákazníci modulu a jejich potřeby

### 5.1.1 Zobrazení svých údajů

Jednou ze základních potřeb uživatelů ve vztahu ke svým osobním údajům je možnost si tyto o své osobě vedené údaje zobrazit. Rozsah zobrazení údajů by měl vždy co možná nejširší, a to v souladu se zásadou transparentnosti a pravidly stanovenými obecným nařízením o ochraně osobních údajů (GDPR).

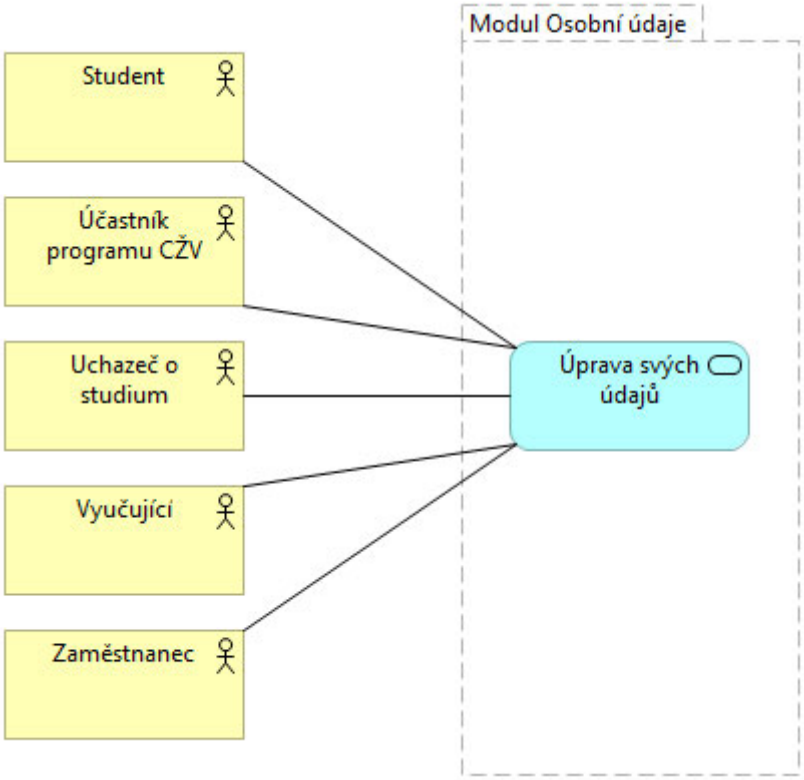
Rozsah údajů pokrytý touto analýzou popisuje pouze výsek informací, které by měl v cílovém stavu uživatel o sobě vidět. Tento výsek se bude rozšiřovat společně s rozvojem dalších částí studijního informačního systému.




### 5.1.2 Editace svých údajů

Návaznou potřebou uživatele je možnost si údaje o své osobě, tedy data v modulu Osoba editovat. Modul Osobní údaje umožňuje editaci stejným typům uživatelů jako u funkce zobrazení údajů. Následující tabulka uvádí přehled údajů, které mají uživatelé právo editovat.

| |
|---|
| Doručovací adresa |
| Bankovní účet |
| Kontaktní údaje<br>• E-mailová adresa<br>• Facebook<br>• Instagram<br>• LinkedIn<br>• Mobilní telefon<br>• Telefonní číslo<br>• Univerzitní e-mail<br>• Webové stránky |



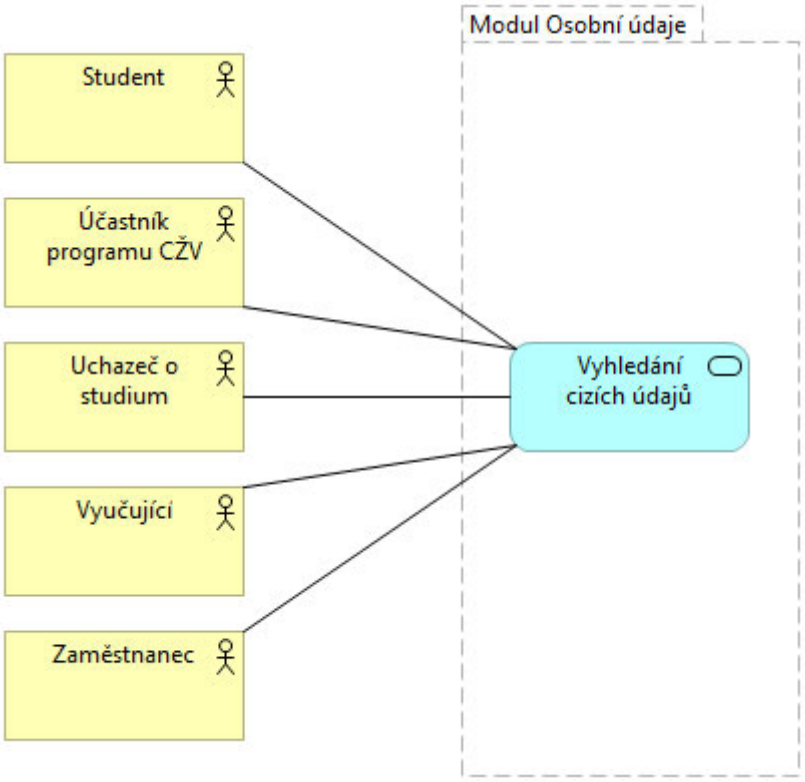

### 5.1.3 Vyhledávání cizích údajů

Vyhledávání probíhá na základě vyhledávacích parametrů:

Jméno – vyhledávání na základě částečné shody
Příjmení – vyhledávání na základě částečné shody
Číslo osoby přidělené univerzitou – vyhledávání na základě přesné shody

Při vyhledávání cizích údajů jsou zobrazeny následující údaje:

| Údaj | Zaměstnanec/Vyučující | Anonymní uživatel |
|---|---|---|
| Jméno | ✓ | ✓ |
| Příjmení | ✓ | ✓ |
| Tituly před jménem | ✓ | ✓ |
| Tituly za jménem | ✓ | ✓ |
| Fotografie | ✓ | ✓ |
| Další údaje ke zveřejnění dle individuálního nastavení uživatelem | ✓ | ✓ |




#### 5.1.4 Založení osoby

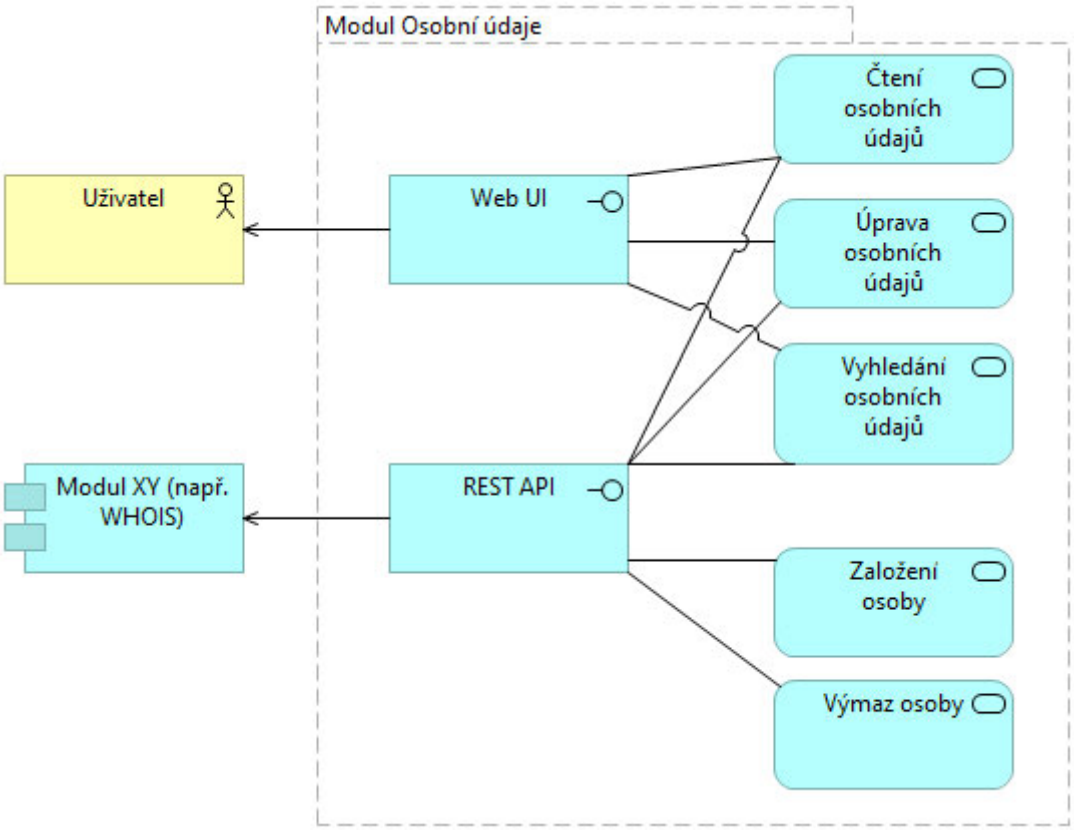
Funkce založení osoby není součástí rozsahu realizovaného touto zakázkou.

#### 5.1.5 Výmaz osoby

Funkce smazání osoby není součástí rozsahu realizovaného touto zakázkou.

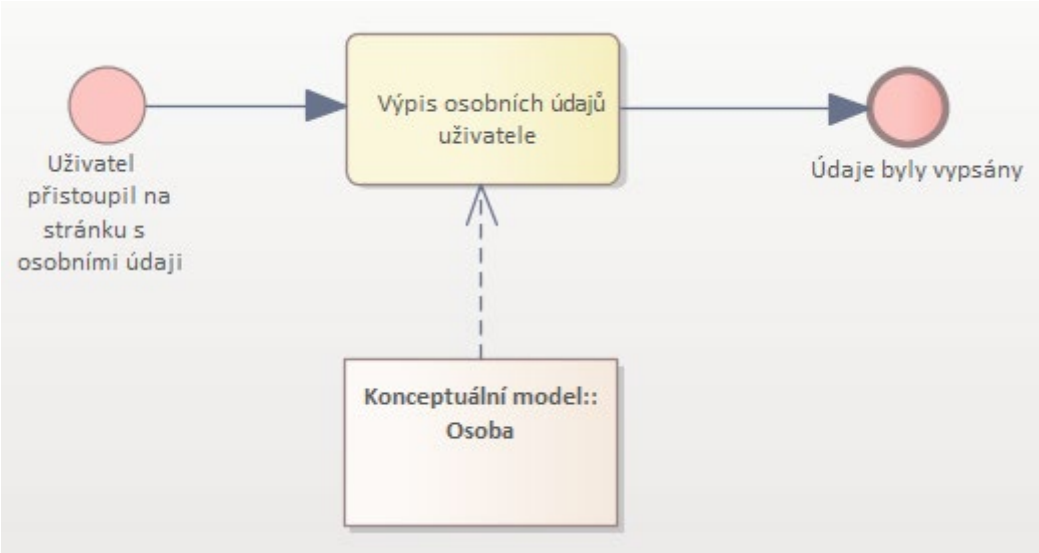
## 6 Modely a diagramy

#### 6.1.1 Využití funkcí modulu různými kanály.

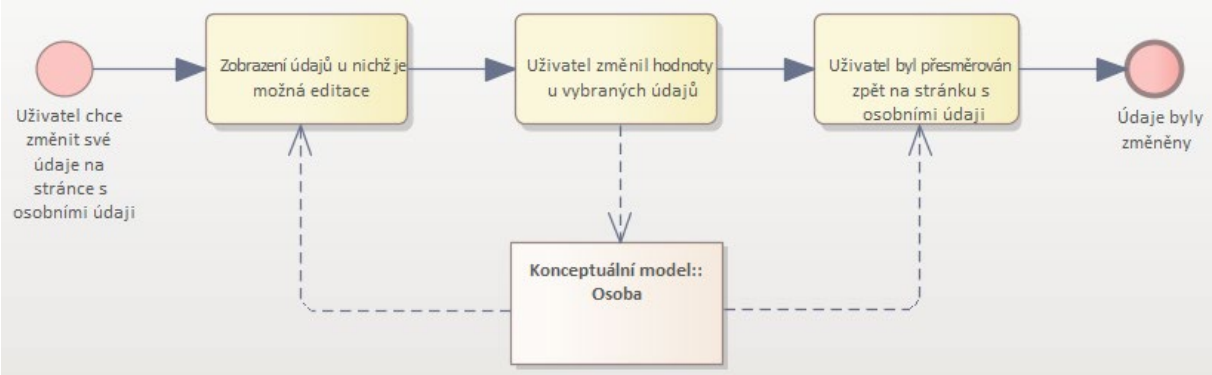



# 7 Procesní modely

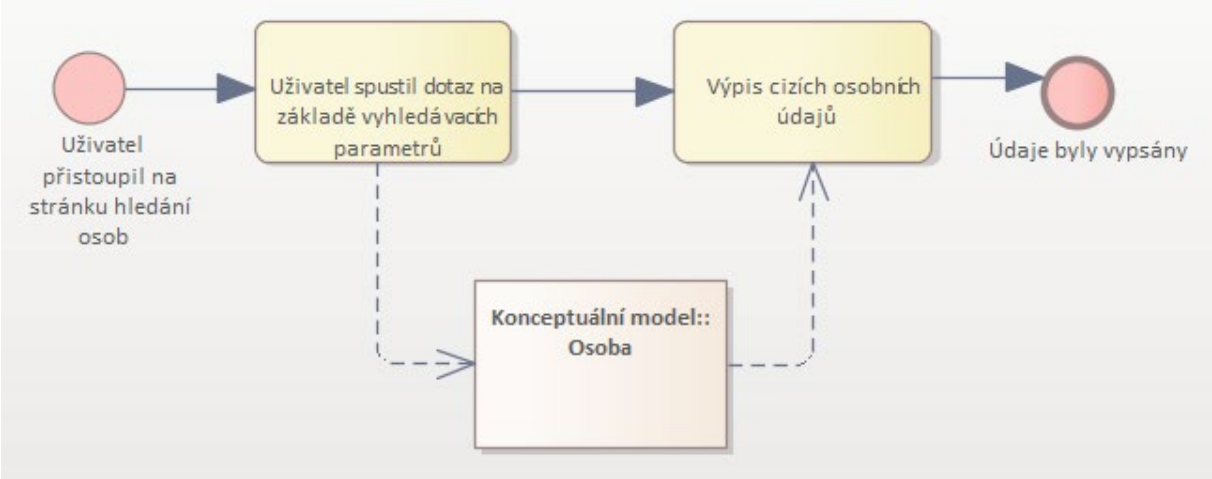
## 7.1.1 Zobrazení svých údajů



## 8 Editace svých údajů



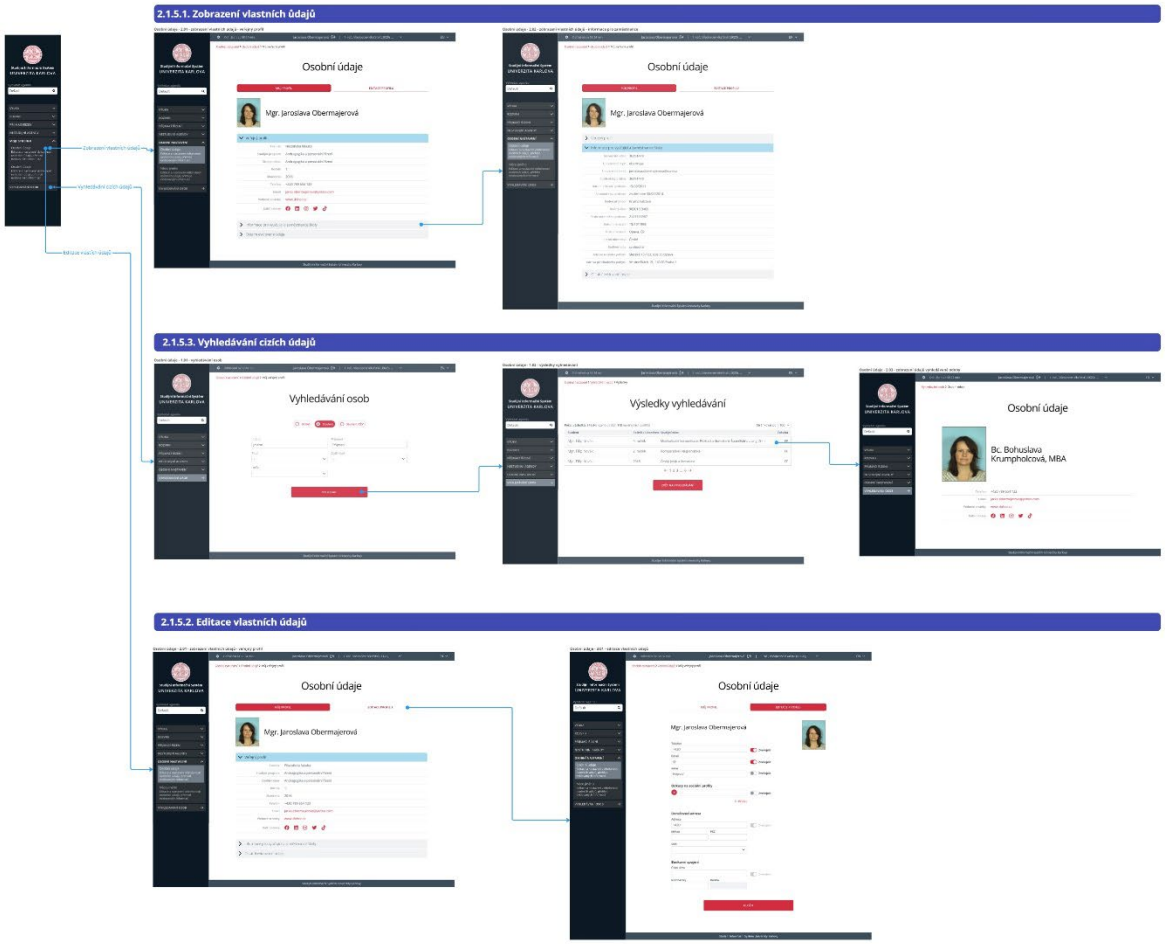

## 9 Vyhledávání cizích údajů



## 10 Uživatelské rozhraní modulu Osobní údaje

### 11 Flowchart

Následující schéma znázorňuje mapování UI na procesní modely (viz kapitolu 2.1.5. Procesní modely v tomto dokumentu):




PDF tohoto schématu a exporty jednotlivých obrazovek UI jsou součástí přílohy.

### 12 Přístupy k návrhům UI v prostředí Figma

Soubory UI návrhů ve Figmě jsou přístupné pro prohlížení na těchto adresách.

**Komponenty**:
https://www.figma.com/file/Ye5Fy9jLvd4o0eG3TA9jnb/SoSIS-Design-system?node-id=173%3A2026&t=44tCjbchs4WV8Vp0-1

**Wireframy obrazovek**:
https://www.figma.com/file/Ye5Fy9jLvd4o0eG3TA9jnb/SoSIS-Design-system?node-id=173%3A2026&t=44tCjbchs4WV8Vp0-1

**Prototyp**:
https://www.figma.com/proto/Ye5Fy9jLvd4o0eG3TA9jnb/SoSIS-Design-system?page-id=0%3A1&node-id=1%3A2&viewport=275%2C1823%2C0.25&scaling=contain&starting-point-node-id=187%3A2711

## 13 Strojová rozhraní modulu

14 Popis vnějších rozhraní modulu Osobní údaje a vazeb na jiné moduly

Detailní specifikace je obsažena v příloze „Veřejné API - základní.yaml“.

## 15 Specifické požadavky na modul Osobní údaje

16 Datový interface pro integraci se stávajícím SIS

### 16.1.1 Datový interface pro stávající databázi SISu

Pro integraci nových modulů bude ve stávající Oracle databázi vytvořeno nové servisní schéma modul_sosis v němž budou postupně vytvářeny potřebná views pro čtení a proc(func) pro zápis dat do stávající databáze. Pro každý z modulů pak bude v databázi vytvořen další uživatel(schéma) pomocí nějž se bude nová aplikace připojovat, zároveň zde připravíme synonyma na používané objekty a potřebná oprávnění. Pro modul Osobní údaje bude vytvořen db uživatel modul_osobniudaje. Ve schématu tohoto uživatele bude možné vytvářet další tabulky pro uložení vlastních dat modulu.

### 16.1.2 Popis modulem používaných tabulek

## *17 Osoba*

- tabulka osobních údajů

| sloupec | typ | délka | popis | vazba |
|---|---|---|---|---|
| OIDENT | number | 10 | ID osoby (SIS) | |
| OIDOS | varchar2 | 10 | **ID osoby (centrální za školu - UKČO)** | |
| OLOGIN | varchar2 | 30 | **univerzitní login** | |
| OPRUKAZ | varchar2 | 20 | číslo průkazu | |
| OCIPCISLO | varchar2 | 20 | číslo čipu z průkazu | |
| OPRUKSTAV | varchar2 | 1 | průkaz - stav | PRUKST.KOD |
| OISIC | varchar2 | 0 | má průkaz ISIC (A/N) | ANONE.KOD |
| ORODC | varchar2 | 10 | **rodné číslo/rodný kód cizince** | |
| ORODCPOZN | varchar2 | 50 | poznámka k rodnému číslu | |
| ODATNAR | date | 0 | **datum narození** | |
| OPOHL | varchar2 | 1 | pohlaví (1=muž; 2=žena) | POHL.KOD |
| OPRIJMENI | varchar2 | 50 | **příjmení** | |
| OJMENO | varchar2 | 50 | **křestní jméno** | |
| OSTREDJMENO | varchar2 | 100 | prostřední jméno | |
| OROZEN | varchar2 | 50 | **rodné příjmení** | |
| OTITUL | varchar2 | 100 | **tituly před jménem** | |
| OTITULZA | varchar2 | 100 | **tituly za jménem** | |
| OOBC | varchar2 | 3 | **státní občanství** | ZEM.KOD |
| OOBCQ | varchar2 | 1 | kvalifikátor občanství | OBCQ.KOD |
| OTRVCR | varchar2 | 0 | má trvalý pobyt v ČR? (A/N) | ANONE.KOD |
| ORODST | varchar2 | 2 | **rodinný stav** | RODST.KOD |
| OOP | varchar2 | 50 | **občanský průkaz - číslo průkazu** | |
| OCPAS1 | varchar2 | 50 | **pas - číslo průkazu** | |
| OCPAS2 | varchar2 | 50 | pas (další průkaz) - číslo průkazu | |
| ORMISTO | varchar2 | 50 | místo narození | |
| ORSTAT | varchar2 | 3 | **stát narození** | ZEM.KOD |
| ORKROK | varchar2 | 4 | okres narození | KROK.KOD |

| | | | | |
|---|---|---|---|---|
| OBYT | varchar2 | 80 | adresa trvalá - upřesnění bytu | |
| OULICE | varchar2 | 75 | **adresa trvalá - ulice** | |
| OCPORI | varchar2 | 10 | **adresa trvalá - číslo popisné** | |
| OMESTO | varchar2 | 50 | **adresa trvalá - obec** | |
| OOBEC | varchar2 | 6 | **adresa trvalá - kód obce (RUIAN)** | OBEC.KOD |
| OCOBEC | varchar2 | 6 | **adresa trvalá - kód části obce (RUIAN)** | COBEC.KOD |
| OCASTOBCE | varchar2 | 50 | **adresa trvalá - část obce** | |
| OPSC | varchar2 | 10 | **adresa trvalá - PSČ** | PSC.KOD |
| OPOBOX | varchar2 | 10 | adresa trvalá - P.O. box | |
| OPOSTA | varchar2 | 50 | adresa trvalá - pošta | |
| OKROK | varchar2 | 4 | **adresa trvalá - okres (RUIAN)** | KROK.KOD |
| OSTAT | varchar2 | 3 | **adresa trvalá - stát** | ZEM.KOD |
| OPOZNADR | varchar2 | 100 | adresa trvalá - poznámka k adrese | |
| OBYTP | varchar2 | 80 | doručovací adresa - upřesnění bytu | |
| OULICEP | varchar2 | 75 | **doručovací adresa - ulice** | |
| OCPORIP | varchar2 | 10 | **doručovací adresa - číslo popisné** | |
| OMESTOP | varchar2 | 50 | **doručovací adresa - obec** | |
| OOBECP | varchar2 | 6 | **doručovací adresa - kód obce (RUIAN)** | OBEC.KOD |
| OCOBECP | varchar2 | 6 | **doručovací adresa - kód části obce (RUIAN)** | COBEC.KOD |
| OCASTOBCEP | varchar2 | 50 | **doručovací adresa - část obce** | |
| OPSCP | varchar2 | 10 | **doručovací adresa - PSČ** | PSC.KOD |
| OPOBOXP | varchar2 | 10 | doručovací adresa - P.O. box | |
| OPOSTAP | varchar2 | 50 | doručovací adresa - pošta | |
| OKROKP | varchar2 | 4 | **doručovací adresa - okres (RUIAN)** | KROK.KOD |
| OSTATP | varchar2 | 3 | **doručovací adresa - stát** | ZEM.KOD |
| OPOZNADRP | varchar2 | 100 | poznámka k další adrese | |
| OBYTP2 | varchar2 | 80 | doručovací adresa uchazeč - upřesnění bytu | |
| OULICEP2 | varchar2 | 75 | **doručovací adresa uchazeč - ulice** | |
| OCPORIP2 | varchar2 | 10 | **doručovací adresa uchazeč - číslo popisné** | |
| OMESTOP2 | varchar2 | 50 | **doručovací adresa uchazeč - obec** | |
| OOBECP2 | varchar2 | 6 | **doručovací adresa uchazeč - kód obce (RUIAN)** | OBEC.KOD |
| OCOBECP2 | varchar2 | 6 | **doručovací adresa uchazeč - kód části obce (RUIAN)** | COBEC.KOD |
| OPSCP2 | varchar2 | 10 | **doručovací adresa uchazeč - PSČ** | PSC.KOD |
| OKROKP2 | varchar2 | 4 | **doručovací adresa uchazeč - okres (RUIAN)** | KROK.KOD |
| OSTATP2 | varchar2 | 3 | **doručovací adresa uchazeč - stát** | ZEM.KOD |
| OJAKAADR | varchar2 | 1 | Student - kontaktní adresa (T-trvalá / P-doruč. 1 / S-dat. schránka) | |
| OUJAKAADR | varchar2 | 1 | Uchazeč - kontaktní adresa (T/2) | |
| OPREDTELEF1 | varchar2 | 9 | **telefonní číslo 1 - přečíslí** | ZEM.PREDCISLI |
| OTELEF1 | varchar2 | 15 | **telefonní číslo 1** | |
| OLINKATELEF1 | varchar2 | 6 | **telefonní číslo 1 - linka** | |

| | | | | |
|---|---|---|---|---|
| OPOZNTELEF1 | varchar2 | 100 | telefonní číslo 1 - poznámka | |
| OPREDTELEF2 | varchar2 | 9 | telefonní číslo 2 - přečíslí | ZEM.PREDCISLI |
| OTELEF2 | varchar2 | 15 | telefonní číslo 2 | |
| OLINKATELEF2 | varchar2 | 6 | telefonní číslo 2 - linka | |
| OPOZNTELEF2 | varchar2 | 100 | telefonní číslo 2 - poznámka | |
| OPREDMOBIL | varchar2 | 9 | mobil - přečíslí | ZEM.PREDCISLI |
| OMOBIL | varchar2 | 15 | mobil | |
| OLINKAMOBIL | varchar2 | 6 | mobil - linka | |
| OPOZNMOBIL | varchar2 | 100 | mobil - poznámka | |
| OMAIL | varchar2 | 100 | **email** | |
| OPOZNMAIL | varchar2 | 100 | poznámka k mailu | |
| OCHMAIL | varchar2 | | chybná e-mailová adresa | ANONE.KOD |
| OMAILNE | varchar2 | | nepřeje si dostávat e-maily | ANONE.KOD |
| OURL | varchar2 | 250 | **webové stránky** | |
| OPOZNURL | varchar2 | 100 | poznámka k URL | |
| OZPOJISTOVNA | varchar2 | 10 | zdravotní pojišťovna | ZPOJ.KOD |
| OZTP | varchar2 | 5 | změněná pracovní schopnost | ZTP.KOD |
| OZDRAVI | varchar2 | 5 | **zdravotní omezení** | ZDRAVI.KOD |
| OZDRAVI2 | varchar2 | 5 | **zdravotní omezení 2** | ZDRAVI.KOD |
| OZDRAVI3 | varchar2 | 5 | **zdravotní omezení 3** | ZDRAVI.KOD |
| OPOZDRAVI | varchar2 | 500 | poznámka ke zdravotnímu stavu | |
| OBANKA | varchar2 | 4 | **bankovní účet v ČR - banka** | BANKA.KOD |
| OUCET | varchar2 | 20 | **bankovní účet v ČR - číslo účtu** | |
| OSPECSYM | varchar2 | 10 | **bankovní účet v ČR - specifický symbol** | |
| OPOZNAMKA | varchar2 | 4000 | poznámka k osobě | |
| OPODEPSALISOUHLAS | varchar2 | 1 | zdravotní omezení - podepsal informovaný souhlas | ANONE.KOD |
| ODPODEPSALISOUHLAS | date | 0 | **zdravotní omezení - datum podpisu informovaného souhlasu** | |
| ODPREZKOUSENIZSTAVU | date | 0 | **zdravotní omezení - datum provedení funkční diagnostiky** | |
| OSDELENI | varchar2 | 0 | souhlas se zasíláním ne/komerčních sdělení (A/N/null) | ANONE.KOD |
| OOBCIDOS | varchar2 | 25 | identifikátor osoby v zemi občanství | |
| ODS | varchar2 | 7 | **id datové schránky** | |
| OZEMREL | varchar2 | 1 | zda daná osoba zemřela | ANONE.KOD |
| OOBC2 | varchar2 | 3 | další státní občanství | ZEM.KOD |
| OVZPREGC | varchar2 | 10 | registrační číslo VZP | |

Ve sloupci vazba jsou uvedeny vazby na další tabulky.

Znaková sada celé databáze je AL32UTF8. Množina znaků použitelných pro textové položky jméno, příjmení, názvy obcí, ulic apod. je dále omezena přesně vymezeným okruhem použitelných znaků z důvodu vazby na další personální a ekonomické systémy univerzity.

## 18 OSFOTO

- profilová fotografie (osobní/průkazková)

| sloupec | typ | délka | popis | vazba |
|---|---|---|---|---|
| ID | number | 10 | id záznamu | |
| OIDOS | varchar2 | 10 | ID osoby (centrální za školu) | OSOBA.OIDOS |
| FORMAT | varchar2 | 5 | typ obrázku (JPG nebo BMP) | |
| FOTO | blob | | fotka | |
| TYP | varchar2 | 1 | velikost (typ) fotky S-mall, L-arge | |
| PLATIOD | date | | platí od | |
| PLATIDO | date | | platí do | |
| DT | date | | kdy bylo změněno | |
| DTV | date | | kdy bylo vytvořeno | |

## 19 STUDIUM

- vybrané údaje z evidence studií (aktuální studia)

| sloupec | typ | délka | popis | vazba |
|---|---|---|---|---|
| SIDENT | number | 10 | ID studia | |
| SOIDENT | number | 10 | ID osoby | OSOBA.OIDENT |
| SFAK | varchar2 | 5 | Fakulta | FAK.KOD |
| SFAK2 | varchar2 | 5 | fakulta 2 | FAK.KOD |
| SDRUH | varchar2 | 2 | druh studia | DRUH.KOD |
| SFST | varchar2 | 2 | forma studia | FST.KOD |
| SSTUPR | varchar2 | 20 | studijní program | STUPR.KOD |
| SSTUPR2 | varchar2 | 20 | studijní program 2 | STUPR.KOD |
| SDSTUPR | date | 0 | datum zápisu do studijního programu | |
| SNOBOR1 | varchar2 | 20 | kód 1.oboru SIMS | NOBOR.KOD |
| SNOBOR2 | varchar2 | 20 | kód 2.oboru SIMS | NOBOR.KOD |

Dle číselníku druh lze určit typ studia – student (dle zákona) – typ magisterský, bakalářský, navazující magisterský, doktorský; účastníci celoživotního vzdělávání - typ program celoživotního vzdělávání

## 20 PRIHLASKA

- vybrané údaje aktuálních přihlášek (uchazečů o studium)

| sloupec | typ | délka | popis | vazba |
|---|---|---|---|---|
| UIDENT | number | 10 | ID přihlášky | |
| UOIDENT | number | 10 | ID osoby | OSOBA.OIDENT |
| USKR | varchar2 | 4 | akademický rok | |
| UFAK | varchar2 | 5 | fakulta | FAK.KOD |
| UFAK2 | varchar2 | 5 | fakulta 2 | FAK.KOD |
| UDRUH | varchar2 | 2 | druh studia | DRUH.KOD |
| UFST | varchar2 | 2 | forma studia | FST.KOD |
| USTUPR | varchar2 | 20 | studijní program | STUPR.KOD |

| | | | | |
|---|---|---|---|---|
| USTUPR2 | varchar2 | 20 | studijní program 2 | STUPR.KOD |
| UNOBOR1 | varchar2 | 20 | kód 1.oboru SIMS | NOBOR.KOD |
| UNOBOR2 | varchar2 | 20 | kód 2.oboru SIMS | NOBOR.KOD |
| UVYJAZYK | varchar2 | 6 | jazyk vyuky | JAZYK.KOD |
| UDPRIHL | date | 0 | datum přijetí přihlášky | |

## *21 Číselníky*

V našem případě jsou v tabulce OSOBA vazby pouze na číselníky, které mají vždy stejnou základní strukturu:

| sloupec | typ | popis |
|---|---|---|
| KOD | varchar2 | Kód číselníku |
| NAZEV | varchar2 | název |
| ANAZEV | varchar2 | anglický název |
| DOD | date | datum od kdy platí |
| DDO | date | datum do kdy platí |
| NEPLATNOST | varchar2 | neplatný záznam = not null |
| | | |

## *22 Zápisové operace*

Pomocí aplikace bude možné měnit údaje popsané v části 2.1.3.2 Editace svých údajů dokumentu.

Vše ve struktuře výše popsané tabulky OSOBA. Pro zápis údajů budou připraveny databázové procedury přijímající jako parametry měněné údaje a identifikaci pachatele změny. Bližší specifikace bude dohodnuta během implementace.

# 23 Konfigurační parametry a langy

### 23.1.1 Konfigurační parametry

Konfigurační parametry jsou v tabulce WCONFIG, mohou být globální nebo fakultní. Globální konfigurační parametry ovlivňují chování celého systému (příp. ovlivňují chování systému pro nepřihlášené uživatele), fakultní konfigurační parametry ovlivňují chování systému dle příslušnosti přihlášeného uživatele k dané fakultě. Globální parametry jsou v datové struktuře uloženy jako konfigurační parametry za organizační jednotku UK (kód 11000). Rozlišujeme několik typů konfiguračních parametrů, např. MO – 0/1, A – text, N – číslo, L – list, apod. Výchozí nastavení parametrů najdeme v tabulce WTEMPLATE_CONFIG, v tabulce WMODUL je pak seznam všech používaných modulů. Více viz struktura:

| **sloupec** | **typ** | **délka** | **popis** | **vazba** |
|---|---|---|---|---|
| CID | number | 0 | id | |
| CFAK | varchar2 | 5 | fakulta | FAK.KOD |
| CMODUL | varchar2 | 50 | modul | WMODUL.KOD |
| CNAME | varchar2 | 50 | nazev configu | |
| CTYP | varchar2 | 2 | typ configu | |
| CLEVEL1 | varchar2 | 50 | uroven | |
| CTYP1 | varchar2 | 2 | podtyp | |
| CVALUE | varchar2 | 1000 | hodnota | |

Modul Osobní údaje nemá žádné konfigurační parametry. (Zde je tedy popsána jen základní logika práce s konfiguračními parametry.)

### 23.1.2 Langy

V obdobné struktuře jsou v systému uloženy popisky/langy, najdeme je v tabulce WLANG. Celý systém je koncipován dvojjazyčně cs a en. Každá lang položka obsahuje výchozí a příp. změněné hodnoty, umožňuje

odděleně nastavit hodnoty samotného popisku a hodnoty nápovědy/hintu. Vše je možné opět nastavovat buď jako globální hodnoty nebo jako fakultní hodnoty, totožně jako u konfiguračních parametrů.

| sloupec | typ | delka | popis | vazba |
|---|---|---|---|---|
| LFAK | varchar2 | 5 | fakulta | FAK.KOD |
| LMODUL | varchar2 | 50 | modul | WMODUL.KOD |
| LNAME | varchar2 | 50 | nazev | |
| LDEF_CS | varchar2 | 2000 | defaultni hodnota cs | |
| LDEF_EN | varchar2 | 2000 | defaultni hodnota en | |
| LVALUE_CS | varchar2 | 2500 | hodnota cs | |
| LVALUE_EN | varchar2 | 2500 | hodnota en | |
| LHINT_DEF_CS | varchar2 | 2500 | nápověda - defaultní, cs | |
| LHINT_DEF_EN | varchar2 | 2500 | nápověda - defaultní, en | |
| LHINT_CS | varchar2 | 2500 | nápověda - vlastní text, cs | |
| LHINT_EN | varchar2 | 2500 | nápověda - vlastní text, en | |
| LPOZNAMKA | varchar2 | 1333 | fakultní poznámka | |

24 Požadavky na audit a observability pro modul Osobní údaje

24.1.1 Audit

## *25 Seznam a struktura požadovaných auditních událostí backendového modulu*

Backendový modul bude provádět audit událostí:

- Start/stop modulu
- Poskytnutí (čtení) detailu osoby (get) – evidence čísla osoby, jejíž detail byl zobrazen; zdroje a pachatele,
- Záznam změny definovaných údajů (put) – záznam měněných údajů; zdroje a pachatele

## *26 Seznam a struktura požadovaných auditních událostí frontendového modulu*

Frontendový modul bude provádět audit událostí:

- Start/stop modulu
- Poskytnutí HTML pohledu – evidence typu pohledu (seznam vs. detail), pro detail čísla osoby, jejíž detail byl zobrazen; zdroje a pachatele,
- Vyřízení změnového požadavku přes UI-záznam měněných údajů, zdroje a pachatele

## *27 Struktura požadovaných auditních událostí*

Příklady auditních záznamů ve formátu JSON:


```
[
  {
    "timestamp": "2022.10.14 08:41:32 -02:00",
    "audit_type": "info",
    "module_id": "osobniudaje",
    "instance_id": "6349049cddccb7effcd31efa",
    "client_url": "146.102.152.22",
    "client_apl": "Google Chrome 105.0.0.0",
    "client_os": "Windows 10 (x64)",
    "request_id": 0,
    "correlation_id": "6dc7c803-3d46-490b-a0bd-36a540055151",
    "user": 34546313,
```

```
    "action": "get",
    "data": [
      {
        "user": 34546313
      }
    ]
  }
]

[
  {
    "timestamp": "2022.10.14 08:50:34 -02:00",
    "audit_type": "info",
    "module_id": "osobniudaje",
    "instance_id": "634906ba8047e431a437ba1d",
    "client_url": "2001:718:1e03:5128:3963:2579:3368:ab4e",
    "client_apl": "Mozilla Firefox 105.0",
    "client_os": "Windows 10 (x64)",
    "request_id": 489,
    "correlation_id": "c81f96c6-c681-4d92-ae20-f841566b9111",
    "user": 56076348,
    "action": "post",
    "data": [
      {
        "user": 48136232,
        "type": "set_phone",
        "field1": "+421",
        "field2": "927 503 337",
        "field3": "nevolat před 11 hod."
      }
    ]
  }
]
```


### 27.1.1 Metriky

## *28 Seznam poskytovaných metrik backendového modulu*

Backendový modul bude poskytovat metriky o vlastním využití za časový okamžik:

- počet přijatých HTTP requestů na HTTP rozhraní
- počet neúspěšných HTTP requestů
- počet úspěšných HTTP requestů
- velikost příchozích HTTP requestů v bytech – min, max, avg
- velikost odchozích HTTP responsů v bytech – min, max, avg
- počet operací čtení detailu osoby (get)
- doba odezvy čtení detailu osoby (get) - min, max, avg
- počet úspěšného čtení detailu osoby
- počet selhání čtecích operací (např. pokus o get neexistující osoby, systémova chyba atd.)
- počet změnových operací definovaných údajů (put)
- počet úspěšných změnových operací

- počet selhání změnových operací
- doba odezvy změnových operací (put) - min, max, avg
- doba odezvy práce s databází - min, max, avg – pro každý typ databázové interakce (prepared statement) zvlášť

## *29 Seznam poskytovaných metrik frontendového modulu*

Frontendový modul bude poskytovat metriky o vlastním využití za časový okamžik:

- počet přijatých HTTP requestů na HTTP rozhraní
- počet neúspěšných HTTP requestů
- počet úspěšných HTTP requestů
- velikost příchozích HTTP requestů v bytech – min, max, avg
- velikost odchozích HTTP responsů v bytech – min, max, avg
- počet UI requestů na zobrazení detailu osoby
- doba odezvy vyřízení UI requestů na zobrazení detailu osoby – min, max, avg
- počet UI requestů na vyhledávání
- doba odezvy vyřízení UI requestů na vyhledávání – min, max, avg
- počet UI requestů na změnu údajů osoby
- doba odezvy vyřízení UI requestů na změnu údajů osoby – min, max, avg
- doba odezvy interakcí s backendovým modulem – min, max, avg - zvlášť pro každý typ API volaní

### 29.1.1 Tracing

## *30 Seznam poskytovaných traců backendového modulu*

Modul bude poskytovat trace spany zejména pro celou dobu vyhodnocování příchozích požadavků (pomocí API), a také podřízený span pro interakci s databázi.

## *31 Seznam poskytovaných traců frontendového modulu*

Modul bude poskytovat trace spany zejména pro celou dobu vyhodnocování příchozích požadavků (pomocí UI), a také podřízený span pro interakci s backendovým modulem.

## 32 Požadavky na výkon a dostupnost pro modul Osobní údaje

### 32.1.1 Frontendový modul

Odhadovaný běžný provoz modulu je zpracování cca 1000 požadavků za minutu, špičkovou zátěž modulu lze očekávat cca 10x vyšší. Pro plynulou práci s aplikací je nutné očekávat základní response systému cca do 200ms (max 1s). Doba response musí být garantována pro 98% požadavků při špičkové zátěži.

Do základní response je potřeba zahrnout většinu základních operací typu zobrazení detailu osobních údajů, uložení upravených údajů, vyhledání cizí osoby na základě osobního čísla (get osoby) apod.

Pro vyhledávací operace na základě částečné shody jména/příjmení může být doba k zobrazení výsledků prodloužena na 60 s; základní response modulu však musí být shodná, kdy modul poskytuje uživateli zpětnou vazbu během provádění déle trvající akce. Do této kategorie déle trvajících akcí lze zahrnout zobrazení náhledů fotografií osob nebo další dílčí informace zjišťované z jiných modulů prostřednictvím api.

### 32.1.2 Backendový modul

Odhadovaná zátěž backendového modulu je cca 1000 požadavků za minutu, špičkovou zátěž modulu lze očekávat cca 10x vyšší. Základní response backendového modulu poskytnutí dat je do 200ms. Pro fulltextové vyhledávací operace je požadovaná response do 60 s.

### 32.1.3 Vysoká dostupnost (HA)

Všechny funkce frontendového i backendového modulu mají podporovat plnohodnotnou vysokou dostupnost (tzv. režim hot-hot).

## **33 Zajištění jakosti (QA) a dokumentace pro modul Osobní údaje**

## 34 Pokrytí kódu unit testy

Pro unit testy je minimální požadované pokrytí kódu 80 %. U složitějších procesů a algoritmů je očekávané pokrytí 100 %. Všechny testy musí proběhnout úspěšně.

35 Seznam požadovaných systémových (end-to-end) testů

100 % uživatelských cest musí být pokryto systémovými testy. Všechny testy musí proběhnout úspěšně. Seznam požadovaných testovacích scénářů:

1. Zobrazení osobních údajů přihlášeného uživatele
2. Změna údajů přihlášeného uživatele
3. Nemožnost změny údajů jiných osob neoprávněným uživatelem
4. Vyhledání seznamu osob na základě zadaných parametrů
5. Zobrazení detailu jiné osoby oprávněným uživatelem

36 Seznam požadovaných výkonnostních testů

Je požadován následující seznam výkonnostních testů níže. Všechny testy musí proběhnout úspěšně:

1. Zobrazení detailu osoby pomocí HTML API
    - alespoň 10 paralelních požadavků
    - celkově 100 požadavků za sekundu
    - po dobu 10 minut
2. Čtení detailu osoby pomocí REST API
    - alespoň 10 paralelních požadavků
    - celkově 100 požadavků za sekundu
    - po dobu 10 minut
3. Změna údajů osoby pomocí HTML API
    - alespoň 10 paralelních požadavků
    - celkově 50 požadavků za sekundu
    - po dobu 10 minut
4. Změna údajů osoby pomocí REST API (50 požadavků za sekundu)
    - alespoň 10 paralelních požadavků
    - celkově 50 požadavků za sekundu
    - po dobu 10 minut
5. Vyhledání seznamu osob na základě zadaných parametrů pomocí HTML API
    - alespoň 10 paralelních požadavků
    - celkově 100 požadavků za sekundu
    - po dobu 10 minut
6. Vyhledání seznamu osob na základě zadaných parametrů pomocí REST API
    - alespoň 10 paralelních požadavků
    - celkově 100 požadavků za sekundu
    - po dobu 10 minut
7. Kombinovaný provoz HTML API transakcí
    - alespoň 10 paralelních požadavků
    - celkově 50 čtecích, 30 změnových a 20 vyhledávacích požadavků za sekundu
    - požadavky různých typů generovány v náhodném pořadí
    - po dobu 10 min
8. Kombinovaný provoz REST API transakcí
    - alespoň 10 paralelních požadavků
    - celkově 50 čtecích, 30 změnových a 20 vyhledávacích požadavků za sekundu
    - požadavky různých typů generovány v náhodném pořadí
    - po dobu 10 min

Všechny výše uvedené testy musí využívat dostatečnou různorodost vstupních parametrů klientských požadavků.

37 Seznam požadovaných testovacích nástrojů (mock, generátory, simulátory)

Je požadován následující seznam testovacích nástrojů:

1. Mock backendového modulu (vracející jeden ze tří fixních objektů osoby)
2. Generátor klientských HTML API požadavků (pro potřeby výkonnostních testů)
3. Generátor klientských REST API požadavků (pro potřeby výkonnostních testů)
4. Generátor randomizovaných osob pro vyplnění databázové reprezentace seznamu osob

- Generátor by měl být schopen vytvořit SQL proceduru pro naplnění databázové tabulky reprezentující seznam osob pomocí randomizovaných dat o objemu 50 tisíc osob

## 38 Způsob řízení projektu a administrace

39 Požadavky na způsob řízení, komunikaci a podporu

### 39.1.1 Způsob řízení

- Požadujeme ustanovení řešitelského týmu, jehož členem je zástupce zadavatele.
- Řešitelský tým se schází na pravidelné bázi, frekvence dohodnuta se zástupcem dodavatele (např. jednou týdně).
- Smyslem setkání je
  - monitoring a koordinace vnitřní realizace modulu s cílem zajistit dodržení
    - požadované architektury
    - vnitřního rozhraní pro přístup k datům ve staré databází SIS
    - vnějších rozhraní poskytovaných modulem (veřejné aplikační rozhraní, datová rozhraní)
    - vnějších rozhraní jiných modulů využívaných modulem
    - napojení na infrastrukturu
    - funkční specifikace v kontextu byznys analýzy
    - test coverage (funkční i kvalitativní /performance, …/)
  - koordinace s paralelní realizací jiných modulů
  - definice průběžných milestones dle potřeb zadavatele/dodavatele
  - monitoring plnění termínu dodávky, případně termínů průběžných milestones
- Na setkání musí proběhnout alespoň:
  - kontrola postupu prací v uplynulém období
  - stanovení postupu prací na následující období
  - rozprava o stávajících překážkách a nejasnostech

### 39.1.2 Způsob komunikace

- Komunikace prostřednictvím komunikační platformy zadavatele (Redmine)
- Groupware pro rychlou týmovou ad-hoc komunikaci
- Issue&tasks tracking (Redmine)
- Zápisy setkání týmu
- Monitoring plnění termínu dodávky, monitoring termínů průběžných milestones

### 39.1.3 Školení administrátorů

Cílem školení administrátorů systému je připravit pověřené pracovníky zadavatele pro výkon funkce správy systémových parametrů IS.

Součástí školení administrace systému bude přehled všech funkčností, vazeb mezi funkčnostmi a implementovaných rozhraní, včetně monitoringu a správy těchto rozhraní.

V rámci tohoto školení zajistí dodavatel vyškolení max. 15 uživatelů.

40 Časový harmonogram

Harmonogram realizace modulu Osobní údaje Tento by měl splňovat kromě jiného následující minimální požadavky:

- Datum začátku realizace, a to nejpozději do jednoho kalendářního měsíce od podpisu smlouvy. Za zahájení realizace se považuje iniciální schůzka dodavatele se zadavatelem.
- Datum konce realizace, a to nejpozději do čtyř kalendářních měsíců od podpisu smlouvy. Za konec realizace je považováno datum, ke kterému dodavatel předá bez výhrad zadavateli všechny smluvní výstupy a požadované přílohy a doplňky.
- Data předání průběžných výstupů dodavatelem zadavateli. Těmi jsou zejména:
  a. backendovou část implementující API s REST testovacím klientem
  b. frontendovou část modulu proti mock backendové části
  c. plně naimplementovaná observability a audit
  d. plné QA
  e. knowledge transfer

Výše uvedený výčet je nutným minimem, nicméně harmonogram předložený dodavatelem by měl samozřejmě obsahovat i další, co možná nejpodrobnější informace popisující, jakým způsobem a v jakých etapách dodavatel plánuje zamýšlenou realizaci.

41 Dokumentace a požadavky na dodávku

Seznam požadovaných výstupů v oblasti dokumentace a QA:

1. Instalační a konfigurační příručka
    - Znalost systémových požadavků
    - Znalost způsobu konfigurace modulu
    - Znalost postupu instalace
2. Dokumentace funkčních a nefunkčních požadavků
    - Znalost happy paths i unhappy paths
    - Napomáhá odhalení regresních závad na úrovni modulu
3. Dokumentace privátního (frontendového) i veřejného API standardizovaným strojově čitelným formátem
    - Verzované API v gitu
    - Znalost API zachycená pro vývojáře ostatních modulů
    - Možnost generovat klienty a mock servery pro dané API automaticky
4. Dokumentace umístění dat používaných modulem a možnosti jejich anonymizace (tam kde je to aplikovatelné)
    - Pro potřeby zálohování a správy dat využívaných modulem
    - Splnění požadavků dle GDPR před provedením kopií, záloh atd.
5. Dokumentace technické architektury modulu/aplikace
    - Interní architektura modulu
    - Komunikace s jinými moduly
    - Možnosti provozu modulu v HA
    - Popis autorizačních rozhodnutí prováděných modulem
6. Dokumentace generovaných auditních událostí
7. Dokumentace generovaných metrik
    - Popis všech metrik (jednotky, způsob měření, přesná sémantika atd.)
    - Doporučení pro nastavení alertů pomocí PromQL
8. Dokumentace vlastních přidaných datových polí (fieldů) do strukturovaných logovacích záznamů
9. Popis všech použitých knihoven třetích stran a zdůvodnění jejich použití
10. Technická dokumentace netriviálních algoritmů
11. Popis SQL skriptů pro vytvoření potřebné databázové struktury a iniciální naplnění dat/číselníků
12. Popis datových migračních skriptů
13. Dokumentace testovací strategie, testovacích scénářů, výkonnostních testů a vytvořených testovacích, simulačních a mockovacích nástrojů
14. Doporučená systémová (hardwarová) konfigurace pro nasazení do testovacího (Stage) i produkčního (Prod) prostředí, odpovídající požadovaným výkonnostním parametrům
15. Uživatelská příručka pro administraci pomocí administračního UI rozhraní (jestli nějaké existuje)
16. Uživatelské příručka pro běžné uživatele

# Příloha č. 1b – Obecná technická specifikace

Obsah

# 42 Úvod a obecná architektura

43 Popis problému a motivace

Univerzita Karlova využívá studijní informační systém (dále jen „SIS“) vyvinutý firmou ERUDIO s r.o. Jádro tohoto systému pochází z 90. let 20. století a je již technologicky zastaralé. Systém má velké množství modulů, které však od sebe nejsou odděleny, což (mimo jiné) významně ztěžuje možnost škálovat výkon systému a také realizovat rozvoj systému více dodavateli. Výkonové problémy se pak projevují zejména při hromadných elektronických zápisech do předmětů a přihlašování studentů na zkoušky. Část modulů má podobu tzv. těžkých klientů, většina z nich má podobu webové aplikace. Jednou z velkých nevýhod stávajících webových modulů SISu je však jejich velmi problematické až nemožné využívání z mobilních zařízení.

V roce 2021 uzavřela UK dodatky smluv s firmou ERUDIO s.r.o., které umožňují SIS dále rozvíjet vlastními silami nebo s využitím třetích stran. To otevřelo univerzitě cestu k tomu, aby se vyvázala ze stávající tzv. vendor lock-in pozice a otevřela vývoj SISu směrem k většímu počtu dodavatelů, mezi nimiž by plnila úlohu integrátora.

Záměrem je převést SIS do podoby moderního informačního systému, který má modulární a tzv. servisně orientovanou architekturu, kde jednotlivé moduly spolu komunikují formou volání webových služeb, a ne sdílením dat ve společné databázi. Tento architektonický model umožňuje lépe škálovat výkon systému a usnadňuje jeho vývoj více různými dodavateli, kteří realizují jednotlivé moduly nebo jejich části, které spolu komunikují prostřednictvím těchto webových služeb.

Současně je cílem formálně popsat procesy (provést tzv. byznys analýzu), pro které SIS poskytuje podporu, a funkcionality obsažené v systému tak, aby tento (průběžně aktualizovaný) popis mohl sloužit jako podklad pro další rozvoj systému – pro komunikaci mezi univerzitou a jednotlivými dodavateli a také mezi dodavateli navzájem.

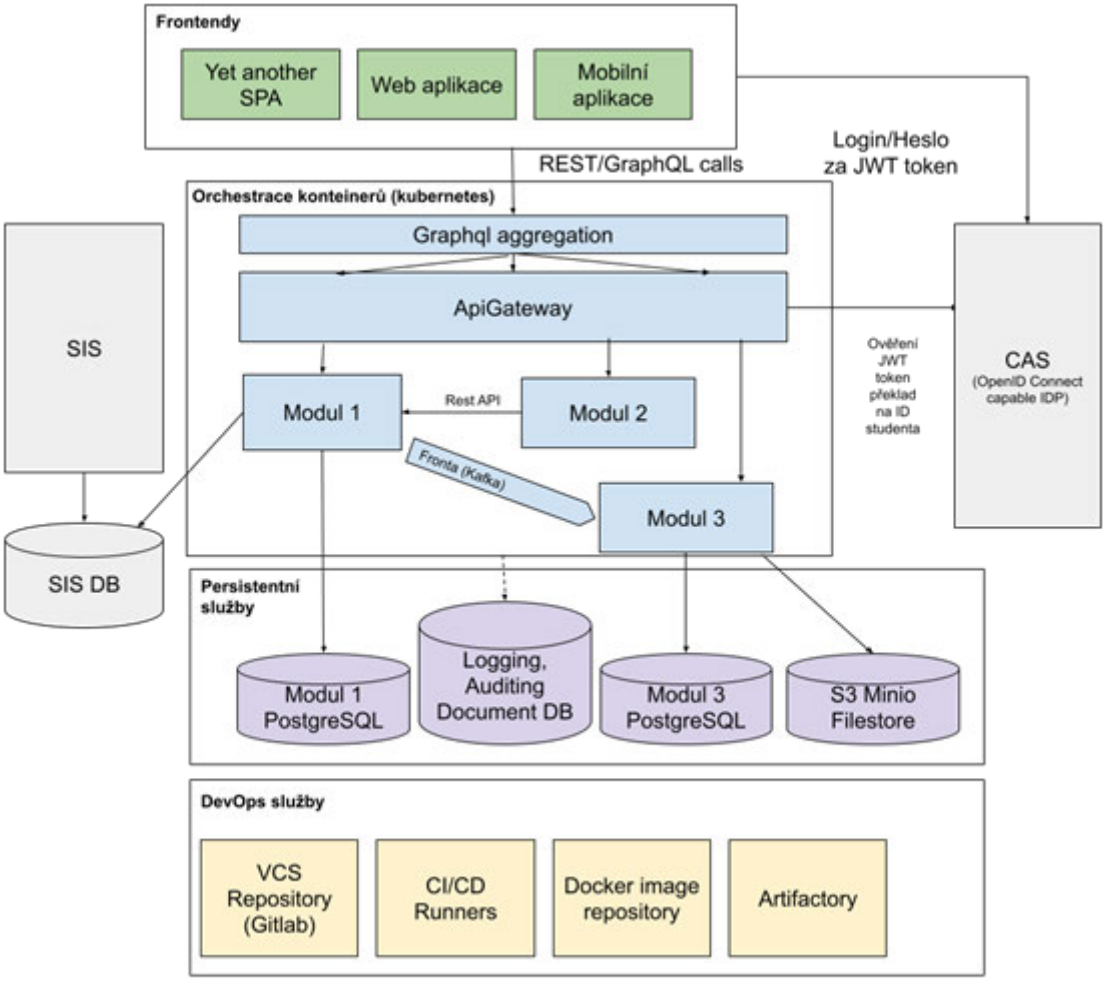
## 44 Obecné infrastrukturní požadavky

### 44.1.1 Koncept nové architektury

Nová architektura se vyznačuje svojí modulárností. Infrastruktura poskytuje sdílené služby aplikačním modulům, jako je autentizace, persistence, logování, monitorování, audit, atd.

Cílem této architektury je umožnit zejména:

- Paralelní vývoj několika dodavateli
- Postupnou reimplementaci Studentského Informačního Systému (SIS)
- Škálování výkonu při zátěži
- Continuous deployment
- Rozsáhlé možnosti QA na různých úrovních systému
- V případě zastarání jednoho modulu není třeba přepisovat celý systém




### 44.1.2 Prostředí, kontejnerizace

Veškeré aplikace budou nasazovány pomocí kontejnerů. Zvolenou technologií pro orchestraci je Kubernetes. Správa Kubernetes probíhá pomocí platformy Rancher. Kubernetes je napojený na Centrální Autentizační Službu (CAS) UK.

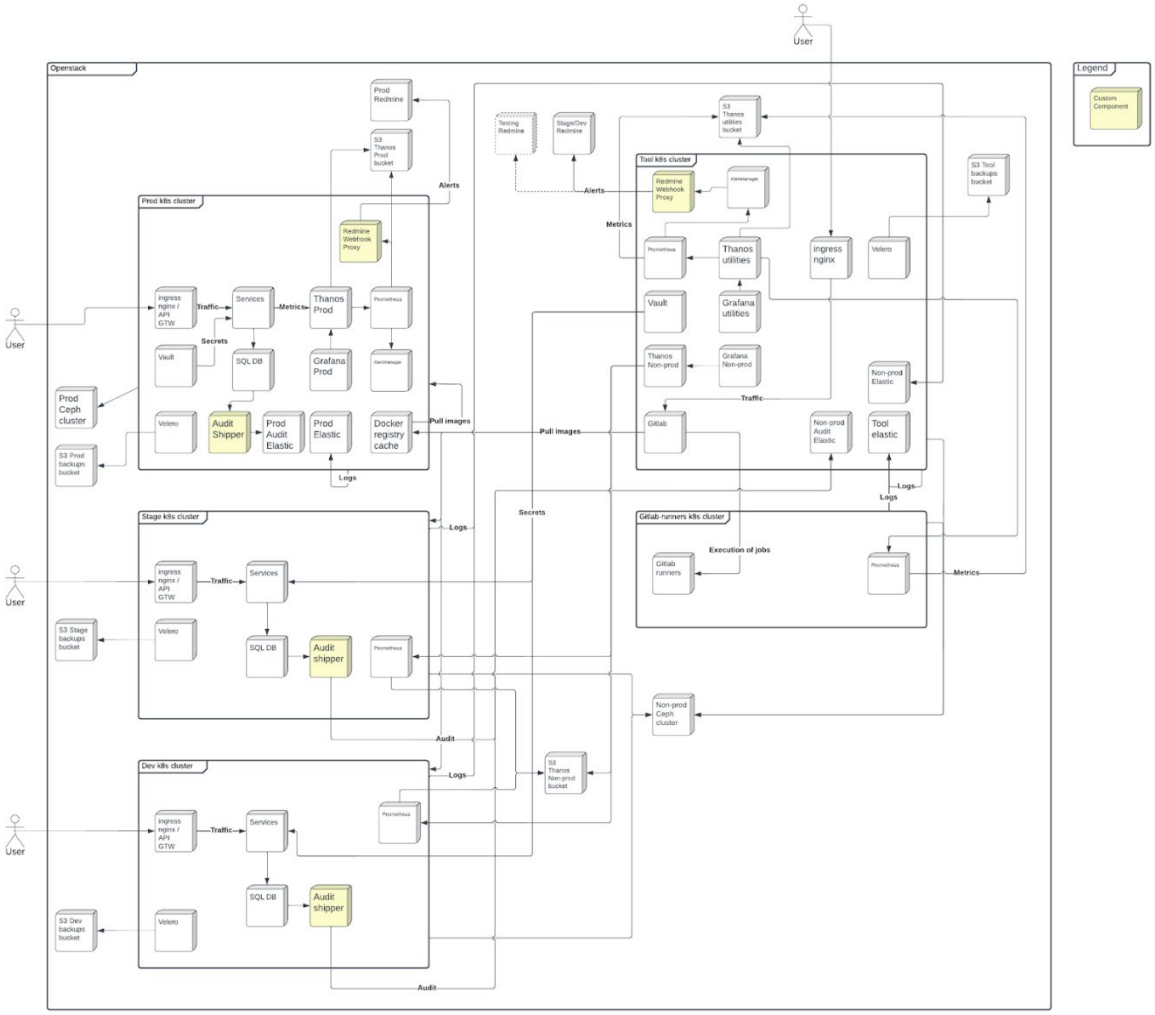
Jsou připravená tři prostředí:

1. **DEV (Development)**
    - Vývojářské prostředí
    - Toto prostředí bude dostupné dodavatelům, vývojáři zde mohou nasazovat své moduly bez asistence pomocí předpřipravených CI/CD pipelines
    - Možnost testovat integraci s ostatními moduly při vývoji
2. **STAGE (Staging)**
    - Testovací prostředí pro UK
    - Možnost spouštět zátěžové testy
    - Nasazování modulů je spravováno UK
3. **PROD (Production)**
    - Prostředí dostupné cílovým uživatelům

Každé z výše uvedených prostředí má vlastní Kubernetes cluster. Pro zjednodušení správy prostředí STAGE a DEV používají tato prostředí také sdílený TOOLS cluster, který obsahuje zejména sdílený Ceph cluster a sdílené nástroje pro monitoring a logování. Mezi další neprodukční systémové clustery patří také GitLab Runners cluster a Rancher management cluster.

Jedním z cílů vytvořeného prostředí je nasazení modulů v režimu vysoké dostupnosti (high availability) společně s automatickým škálováním dle aktuální zátěže.

Následující diagram znázorňuje přiřazení jednotlivých systémových komponent do zmiňovaných clusterů. Vlastní komponenty (vytvářeny interně na UK) jsou zvýrazněny žlutě. Moduly vyvíjené dodavateli jsou shrnuty pod komponentu “Services”.




**Legenda**

- Kubernetes – nástroj pro orchestraci kontejnerů nad clusterem serverů
- NGINX Ingress – obecná reverzní proxy
- Kong API Gateway – specializovaná rozšiřitelná reverzní proxy s pokročilými funkcemi
- PostgreSQL – relační databáze
- Kafka – asynchronní fronta pro komunikaci mezi moduly
- Ceph – platforma spravující datová úložiště
- S3 – běžné rozhraní pro úložiště souborů/objektů
- GitLab – nástroj pro správu zdrojových kódů a automatizaci
- GitLab Runner – komponenta GitLabu, která zprostředkovává automatizaci
- Docker Registry – úložiště obrazů kontejnerů
- ArgoCD – nástroj pro synchronizaci nasazování aplikací a infrastruktury
- Velero – zálohovací nástroj pro Kubernetes
- Prometheus – nástroj pro sběr a krátkodobé úložiště metrik
- Alertmanager – nástroj pro správu automatizovaných upozornění
- Thanos – nástroj pro dlouhodobé ukládání metrik
- Grafana – nástroj pro zobrazování metrik a grafů
- Redmine – wiki a správa incidentů
- Redmine Webhook Proxy – vlastní komponenta pro napojení Alertmanageru a Redmine
- Elasticsearch – úložiště logů a audit záznamů
- Logstash – nástroj pro transformaci logů a audit záznamů

- Filebeat – nástroj pro sběr logů z jednotlivých komponent
- Audit Shipper – vlastní komponenta pro přesun audit záznamů z modulových DB schémat do centrálního úložiště auditních záznamů, napr. Elasticsearch
- Kibana – nástroj pro zobrazení a vyhledávání logů a audit záznamů
- Gatekeeper – nástroj pro vynucování pravidel pro všechny součásti Kubernetes

#### 44.1.3 Git repositář kódu, CI/CD

Pro správu zdrojového kódu jednotlivých modulů je používána platforma GitLab, provozována v rámci výše popsaného kontejnerového prostředí. GitLab je sdílený pro všechna tři běhová prostředí. Slouží také zároveň jako Container Repository (nicméně produkční prostředí má vlastní Docker Registry Cache).

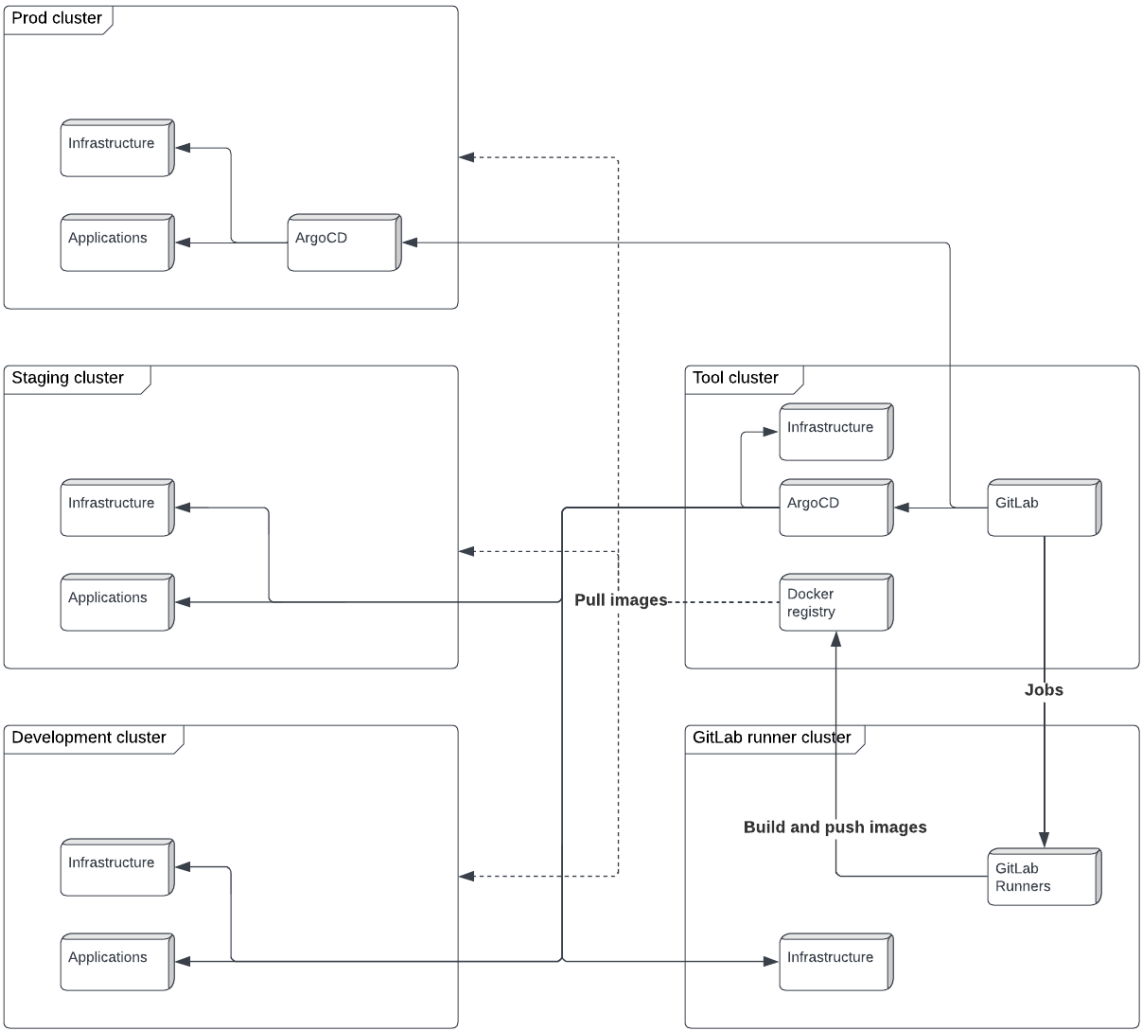
Pro každý modul bude v rámci prostředí GitLab připraven separátní projekt (skupina projektů), ke kterému obdrží dodavatel daného modulu přístup. Zadavatel také vytvoří základní CI/CD pipelines pomocí nástrojů GitLab, ArgoCD a Helm Charts pro automatizované nasazování modulu v prostředí DEV. Správa a následný vývoj těchto CI/CD pipelines bude následně probíhat ve spolupráci s dodavatelem. Finální CI/CD pipelines budou zadavatelem aplikovány také do STAGE a PROD prostředí.

Každý modul musí být reprezentovaný kontejnerem s následujícími požadavky:

- Kód musí být kompatibilní s rootless Docker image
- Každý modul používá standardní šablony pro GitLab CI a standardní Helm chart
- Base images kontejnerů: poslední stable Alpine, Debian nebo Ubuntu

Kromě samotného vyvíjeného modulu budou dodavateli zpřístupněné v rámci prostředí GitLab i vzorové moduly.

Následující CI/CD diagram znázorňuje 2 ArgoCD instance, které obsluhují všechny clustery a nasazení infrastruktury a modulů. Produkční instance je kompletně oddělená od neprodukční. ArgoCD je napojené na GitLab a synchronizuje stav několika repositářů s nastavením aplikací do jednotlivých clusterů.




### 44.1.4 Správa hesel a přístupových údajů (Secrets)

Veškeré přístupové údaje (URL služeb, hesla, tokeny atd.) budou každému modulu poskytnuty pomoci GitLab proměnných v odpovídajícím GitLab projektu a poté pomocí CI/CD pipelines dostupná jako proměnné běhového prostředí (OS environment variables). Každý modul tedy musí tyto informace načítat při startu z proměnných prostředí. **Je explicitně zakázáno ukládat podobné informace, zejména hesla a podobné přístupové údaje v jakýchkoliv konfiguračních souborech modulu.**

### 44.1.5 Principy komunikace mezi službami/moduly

Pro komunikaci mezi jednotlivými moduly budou použité následující technologie

- REST(-like) HTTP API
- GraphQL
- RPC
- Fronta
- Webhooks

## 45 *Synchronní komunikace*

Jako primární technologie určená pro synchronní komunikaci mezi moduly bude použito REST-like HTTP API. V některých případech v budoucnu může být také předepsána implementace GraphQL nebo RPC rozhraní. Konkrétní specifikace typu API (REST vs. GraphQL vs. RPC), včetně specifikace požadovaných operací, datových formátů, vstupních a výstupních dat bude detailně určena v zadávací dokumentaci každého modulu.

Kromě předpisu API rozhraní, které má modul vystavit pro ostatní moduly bude součástí zadávací dokumentace každého modulu také seznam a popis API rozhraní jiných modulů, které bude požadovaný modul potřebovat pro implementaci svých funkcionalit.

Pro zamezení snižování výsledné spolehlivosti modulů by moduly mezi sebou neměly v rámci zpracovávání jednoho požadavku od uživatele provést víc než dvě synchronní volání. Moduly by navíc neměly mezi sebou mít cyklickou závislost synchronních volání.

## 46 *Asynchronní komunikace*

Jako primární technologie určenou pro asynchronní komunikaci mezi moduly bude použita fronta, implementována pomocí platformy Apache Kafka. V rámci jednoho prostředí bude poskytnuta jedna sdílená instance Kafky. Přístupové informace modul obdrží v rámci sady odevzdávaných přístupových údajů pomocí CI/CD, jak bylo popsáno výše.

### 46.1.1 Webhooks

Další z dostupných alternativ pro komunikaci mezi moduly je také využití Webhooks u případů, kde charakter komunikace implementovaní touto technologii bude výhodnější než použití standardního REST API rozhraní. Požadavek na použití Webhooks bude pro daný modul specifikován v rámci zadávací dokumentace modulu.

### 46.1.2 Síťový provoz a napojení, reverzní proxy, API brána

Veškerý síťový provoz bude směrován na vstupní komunikační body (endpointy) modulu pomocí reverzní proxy a API brány. Tyto prvky budou poskytovat terminaci příchozího TLS spojení, load-balancing, základní autorizaci a směrování datového provozu na odpovídající moduly. Pro propagaci zdrojových IP adres klientů příchozích HTTP spojení budou přidávané hlavičky **X-Forwarded-For**.

Použité technologie pro reverzní proxy i API bránu jsou zároveň kompatibilní s logovacím, monitorovacím a tracing řešením, takže transparentně probíhá např. zaznamenávání požadavků do logu včetně cesty volání, status kódu a doby trvání, korelace logů, latence a statistických indikátorů s ostatními moduly.

Korelace jednotlivých auditních a logovacích zpráv je zajištěna pomocí přidávání identifikátoru požadavku Correlation-ID do všech logovacích a auditních zpráv. Tento identifikátor je automaticky vytvořen pro každý nový požadavek vstupující do systému přes API bránu, a to pomocí přidání HTTP hlavičky **X-Correlation-Id** s hodnotou vygenerovaného identifikátoru.

### 46.1.3 Autentizace a autorizace

Z pohledu autentizace a autorizace počítá architektura systému se dvěma hlavními případy užití:

- Uživatel přistupující k systému pomocí prohlížeče skrze webový Portál
- Jiný systém/uživatel přistupující přímo k veřejnému API systému

V prvním případě proběhne autentizace a autorizace uživatele pomocí odpovídajícího flow OpenID Connect/OAuth, s využitím stávající Centrální Autentizační Služby (CAS) UK. Z pohledů aplikačních modulů bude výsledkem přihlášení access token, který obdrží Portál SIS (jakožto vstupní uživatelské webová brána do SIS).

Na základě přijatého access tokenu (a z něj odvozených informací o přihlášeném uživateli) může Portál rozhodnout, ke kterým portálovým aplikacím bude uživateli poskytnutý přístup. Pro vstup do specifické portálové aplikace musí Portál následně požádat o poskytnutí odpovídajícího HTML UI specifický modul. V rámci tohoto volání Portál poskytne modulu také access token kontextu transakce. Jestliže tento modul potřebuje navíc kontaktovat další modul, musí mu v rámci odpovídajícího API volaní také přeposlat obdržený access token.

Podobně ve druhém případě, při použití odpovídajícího flow OpenID Connect/OAuth, modul obdrží v rámci přijatého požadavku také odpovídající access token, který v případě volání dalších modulů jim bude také přeposílat.

Jednotlivé moduly můžou na základě obdrženého access tokenu dále přesněji vyhodnocovat interní autorizační pravidla v závislosti na specifikách své aplikační logiky. Detaily autorizační logiky specifické pro daný modul budou upřesněné v zadávací dokumentaci.

### 46.1.4 Perzistentní datová úložiště

Pro perzistentní ukládání dat má každý modul k dispozici následující tři druhy úložišť:

- RDBMS – Relační databáze podporující transakce
- File/blob úložiště – S3
- Fulltext indexovaná dokumentová databáze (Document Store)

Pro všechny výše uvedené nástroje pro perzistenci dat zabezpečí zadavatel zároveň odpovídající řešení pro zálohu dat.

Všechny tyto nástroje jsou zároveň spravovány centrálně a přístupové údaje k nim jsou jednotlivým modulům poskytovány dle výše popsaného mechanismu sdílení přístupových údajů.

Veškeré změny nutné pro instalaci nebo upgrade se provádí automatizovanými migračními skripty poskytnutými dodavatelem. Úkolem těchto skriptů je zabezpečit reprodukovatelnost instalace na dalších prostředích.

Z důvodu použití kontejnerizace a automatického horizontálního škálování nesmí moduly ukládat žádná persistentní data na lokální disk. Jakákoliv potřeba ukládat data na disk dočasně musí být předem konzultována a odsouhlasena zadavatelem. Detaily takové případné implementaci pak musí být popsány v dodavatelem doložené technické dokumentaci.

## 47 Relační databáze

Standardní relační databáze bude v drtivé většině případů použitá jako primární nástroj pro perzistentní uložení dat. Jedním z hlavních důvodů je nutnost integrace nově vyvíjených modulů s původním Studijním informačním systémem na datové úrovni, protože integrace na úrovni databáze je jediný původním systémem podporovaný způsob napojení. Původní systém využívá databázi Oracle.

Pro každý modul tedy bude vytvořeno nové databázové schéma, v rámci kterého budou vytvořené pohledy (views) pro čtení dat sdílených s původním systémem, a také uložené databázové procedury (stored procedures) pro vykonávání zápisových transakcí. Detailní popis dostupných databázových prostředků je dodán jako součást zadávací dokumentace daného modulu.

Kromě práce se sdílenými (starými) daty budou některé moduly potřebovat také ukládat data odpovídající novým, doposud neevidovaným byznys objektům. Pro tyto datové entity bude zadavatelem předepsána datová struktura databázových tabulek, které modul v rámci jemu dostupného databázového schématu bude využívat pro perzistenci datových entit odpovídajícího typu.

Vedle specifikací předepsaných datových objektů budou moduly typicky potřebovat i další pomocné databázové tabulky pro ukládání implementačně specifických perzistentních dat. Strukturu potřebných tabulek sdělí dodavatel zadavateli v průběhu implementace, a to v podobě automaticky spustitelných SQL skriptů. Vytvoření potřebných databázových struktur poté zabezpečí zadavatel.

Každý modul pracující s relační databází musí tento zdroj využívat optimalizovaným způsobem, dle dobrých zvyklostí. Nutností je používání kvalitního Database Connection Pool nástroje (pro Javu např. c3p0) umožňujícího vysokou míru nastavitelnosti a sdílení databázových spojení. Nastavení Database Connection Poolu pak musí být dostupné v rámci konfigurace modulu, který ho využívá. Samozřejmostí je pak používání prepared statements a dalších standardních doporučení při práci s relační DB.

## 48 File/blob úložiště – S3

Pro potřeby ukládání souborů nebo blobů je pro každý modul k dispozici objektové úložiště dostupné pomocí S3 API, poskytované systémem Ceph. V případě potřeby je možné vytvořit pro daný modul i vícero S3 bucketů.

Vytváření S3 bucketů je možné na základě konzultace a schválení zadavatelem. Dodavatel musí kromě účelu a typu užití poskytnout také očekávaný objem ukládaných dat, včetně odhadované četnosti čtecích a zápisových operací.

## 49 Fulltext indexovaná dokumentová databáze

V odůvodněných případech je možné pro modul vytvořit separátní index v rámci dokumentové databáze Elasticsearch. Tento typ perzistence je vhodný zejména v případech, kdy je potřebná fulltextová indexace dokumentově orientovaných dat.

Vytváření indexů je možné na základě konzultace a schválení zadavatelem. Dodavatel musí kromě účelu a typu užití poskytnout také očekávaný objem ukládaných dat, včetně odhadované četnosti čtecích a zápisových operací.

### 49.1.1 Audit

Nedílnou součástí procesování většiny příchozích požadavků daným modulem je synchronní, transakčně korektní a dostatečně průkazné vytváření odpovídajících auditovacích záznamů popisujících důležité kontextuální podrobnosti procesovaných požadavků. Vytvářené auditní záznamy jsou pak neměnné a slouží primárně jako zdroj dat pro naplnění legislativních požadavků, nebo také pro účely reportingu.

Součástí zadávací dokumentace pro každý modul je také seznam auditních událostí, které nastávají v rámci daného modulu po dobu jeho aplikačního běhu. Příkladem takových událostí může být například již zmiňovaná obsluha příchozích požadavků, generování odchozích požadavků, ale také například start/stop modulu, runtime změna konfigurace, přihlášení uživatele, import dat, zpuštění důležitých automatizovaných procesů na pozadí, detekce alertu nebo chyby systému, zaslání e-mailu, atd.

Každý záznam o auditní události musí obsahovat:

- Časovou značku (timestamp) události
- Typ události
- ID služby (modulu)
- ID instance
- ID uživatele
- ID požadavku
- Correlation ID (viz sekce Logování)
- ID nebo jméno vykonávané akce
- Další položky specifické pro daný typ události specifikované v zadávací dokumentaci modulu (typicky vyžadované legislativou a platnými směrnicemi)

Přesný seznam všech pro modul předepsaných auditních událostí, které modul musí implementovat, včetně požadované struktury dat, je součástí zadávací dokumentace daného modulu.

Auditní záznam pro předepsanou událost musí být uložen perzistentním a transakčně korektním způsobem jako nedílná součást odpovídajícího výpočetního procesu. Například, nemělo by být možné vrátit data v rámci obsluhy požadavku ale přitom nevytvořit odpovídající auditní záznam, nebo naopak, neobsloužit požadavek ale přitom vytvořit auditní záznam o jeho obsluze. Tato vlastnost se typicky dosahuje pomocí vytváření auditních záznamů v rámci stejné databázové transakce, uvnitř které probíhá i obsluha procesovaného požadavku a zápis aplikačních dat. Jinými slovy, pro vytváření auditních záznamů bude použit tzv. Outbox Pattern, s outbox tabulkou v databázi modulu.

Kromě samotného sémantického významu jednotlivých auditních událostí je tyto možné dělit do dvou hlavních kategorií na a) párové, a b) nepárové. Příkladem nepárových událostí je např. start systému nebo pokus o přihlášení uživatele do systému. Párové události typicky popisují začátek a konec odpovídajícího procesu nebo transakce, přičemž pro daný proces platí, že buď má delší trvání, nebo v jeho průběhu je zvýšená pravděpodobnost možného pádu nebo zaseknutí systému (např. z důvodu nedostatku systémových procesů, čekaní na externí zdroje atd.). Párové události jsou tedy typicky používané pro zaznamenání procesování uživatelských požadavků, přičemž jeden auditní záznam je vytvořen při přijetí požadavku (včetně detailů popisujících „kdo se ptá na co“), a druhý auditní záznam je vytvořen při zaslání odpovědi na daný požadavek (detaily zde zachycují „co bylo poskytnuto“, případně úspešnost (status) dané transakce, dobu trvání, a podobně.

Auditní záznamy vytvářené všemi aplikačními moduly v rámci jejich transakčních databázových schémat jsou kontinuálně přesouvány a agregovány do dlouhodobého centrálního perzistentního úložiště auditních záznamů pomocí interně vyvíjené komponenty Audit Shipper.

#### 49.1.2 Logování

Všechny moduly zapisují logovací záznamy do standardních výstupů, tj. do **stdout** a **stderr**, přičemž do stderr by měly být zapisovány pouze chybové zprávy (zprávy úrovně ERROR).

Logy jsou agregovány, zpracovány a uchovávány pomocí platformy Elastic Stack (Filebeat, Logstash, Elasticsearch, Kibana) za účelem možnosti vzájemné korelace událostí ze všech modulů a pro zajištění dostupnosti logů i po kritickém selhání a odstranění kontejneru. Webové rozhraní aplikace Kibana v DEV prostředí bude dodavateli zpřístupněno pro umožnění vyhledávání událostí týkajících se konkrétního modulu, kontejneru, časového rozmezí, ID požadavku či úrovně atd. Do budoucna se plánuje vytvoření vlastní aplikace pro prohlížení logovacích záznamů, který by poskytoval ještě větší pohodlí a přehled pro náhled do logů objemných víceúrovňových transakcí.

Z výše uvedených důvodů musí moduly vytvářet logovací záznamy ve strojově čitelné strukturované formě. Jako výchozí formát bylo zvoleno Elastic Common Schema (ECS) ve formátu JSON (https://www.elastic.co/guide/en/ecs-logging/overview/current/intro.html). Tento formát byl dále rozšířen o několik přídavných atributů. Následující tabulka uvádí přehled povinných atributů (ECS fields), které musí obsahovat každý logovací záznam. Samozřejmě, je možné logovat i další metadata dle ECS standardu, nicméně níže uvedené položky jsou povinné.

| **Název atributu** | **Popis** | **Příklad** |
|---|---|---|
| @timestamp | Časová značka logovacího záznamu | 2022-10-11T20:48:18.054Z |
| custom.nano | Počet nanosekund v rámci aktuální sekundy z časové značky logovací události. Slouží pro rozlišení pořadí událostí/zpráv vytvořených v rámci stejné milisekundy. | 54230932 |
| log.level | Log level logovací zprávy. Možné hodnoty jsou ERROR, WARN, INFO, DEBUG, TRACE. Tabulka uvedená níže obsahuje popis logovacích zpráv, které má modul vytvářet pro jednotlivé logovací levely. | INFO |
| service.name | Jméno nebo identifikátor modulu, stanovený na základě domluvy se zadavatelem | portal |
| message | Obsah textu logovací zprávy | Received request: /portal/ |
| error.stack_trace | Stacktrace logovací zprávy (je-li dostupný), určený zejména pro chybové zprávy | java.lang. NullPointerException at java.base/ java.util.Objects. requireNonNull (Objects.java:208) at ... |
| custom.correlation.id | Unikátní ID zpracovávaného požadavku, které bylo přirazeno pomocí platformy API Gateway při vstupu do systému. Všechny moduly v rámci architektury si při následných voláních tento identifikátor odevzdávají v HTTP hlavičkách. | 2b7e26cc-fcdb-429f-b685-f5150c2bb95a#1765 |
| custom.uuid | ID logovacího záznamu ve tvaru UUID, vygenerováno modulem při vzniku záznamu | cb9321ba-b30f-47ad-90b9-781fb310576a |

| Název atributu | Popis | Příklad |
|---|---|---|
| custom.has_children | Boolean příznak indikující, zda-li daná logovací událost je počátkem logicky podřazeného výpočetního podstromu obsahujícího podřazené logovací záznamy (které můžou být vytvořené jiným modulem nebo volanou komponentou modulu). Při stromovém zobrazení logovacích záznamů si je možné logovací záznamy s hodnotou tohoto příznaku nastavenou na „true“ možné představit jako rozkliknutelný podstrom podřazených logovacích událostí. | false |
| custom.parent | ID přímo nadřazeného rodičovské logovací zprávy. Jestliže je aktuální požadavek vykonáván jako podřazená operace nějaké nadřazené části aplikační logiky, pošle nadřazený modul nebo komponenta modulu identifikátor rodičovské logovací zprávy (kořen logovacího podstromu) pomocí HTTP hlavičky **X-Parent-Id**, nebo pomocí interních komunikačních cest mezi interními komponenty moduly. Od podřazené aplikační logiky se pak očekává vyplňování tohoto identifikátoru do všech vytvářených zpráv až do momentu kdy nevznikne nový logovací podstrom. Způsob vytváření a užití parent id je také demonstrován pomocí zdrojového kódu vzorových modulů, které budou dodavateli zpřístupněny v rámci přístupu do GitLab. | b0da9c30-b7f4-4210-8605-0b7d797bcf20 |

Následující tabulka popisuje, jaký typ logovacích zpráv je vyžadováno vytvářet v rámci jednotlivých logovacích levelů.

| Log level | Typ požadovaných logovacích zpráv |
|---|---|
| ERROR | • Zalogování chybových stavů, z nichž se modul (proces zpracování požadavku) nezotavil, včetně stacktrace |
| WARN | • Zalogování chybových a "deprecated" stavů, z nichž se modul (proces zpracování požadavku) zotavil |
| INFO | • Zalogování zahájení a ukončení každé transakce/requestu včetně ID uživatele a status kódu (transakcí se rozumí každá naplánovaná aktivita - job, aktivita iniciovaná uživatelem i aktivita iniciovaná jinou komponentou systému)<br>• Modul zajistí dostupnost informací o činnosti modulu pro potřeby zpětné diagnostiky neočekávaných a náhodných stavů |
| DEBUG | • Zalogování zahájení a ukončení komunikace s ostatními moduly a systémy třetích stran s úrovní DEBUG<br>• Zalogování interních kroků aplikační logiky z pohledu procesního flow takovým způsobem, aby bylo možné ověřit korektnost procesního algoritmu<br>• Zalogování všech ostatních zpráv, které pomohou případné diagnostice neočekávaných stavů |
| TRACE | • Zalogování podrobností o průběhu vykonávání aplikační logiky, obsahujících zejména objemný obsah, např. payload transakcí, TLS handshake, velmi jemné krokování složitějších algoritmů, apod. |

Následující ukázka kódu v jazyce Java demonstruje jeden z možných způsobu vytváření logovacího zázamu ve formátu JSON na základe dodaných strukturovaných hodnot jednotlivých atributů (fieldů).

**Ukázka Java kódu – příklad serializace logovacího záznamu**

```
public String toJson() {
cr
   JSONObject jsonLog = new JSONObject();

    jsonLog.put("@timestamp",  DateTimeFormatter.ISO_INSTANT.withZone(ZoneOffset.UTC).format(instant));
                        jsonLog.put("custom.nano",                  Integer.toString(instant.getNano()));
                              jsonLog.put("custom.correlation.id",                  correlationId);
            if          (uuid          !=          null)          jsonLog.put("custom.uuid",          uuid);
           if          (parent          !=          null)          jsonLog.put("custom.parent",          parent);
         if          (stackTrace          !=          null)          jsonLog.put("error.stack_trace",          stackTrace);
                                    jsonLog.put("log.level",                  level.name());
                                     jsonLog.put("message",                  message);
         jsonLog.put("custom.has_children",          hasChildren          ?          "true"          :          "false");
                                  jsonLog.put("service.name",                  serviceName);

                                          return                  jsonLog.toString();
}
```

V některých případech bude nutné citlivé údaje vkládané do logovacích záznamů šifrovat. Každé šifrované zprávě musí předcházet stejná nešifrovaná zpráva bez citlivých údajů. Seznam citlivých údajů bude pro každý modul upřesněn v zadávací dokumentaci.

Defaultní log level je nastavitelný pomocí konfigurace modulu (v proměnné prostředí). Nicméně komponenty systému předřazené před volání samotného modulu (např. Portál, nebo Centrální API Modul) poskytují schopnost nastavovat požadovaný dynamický log level pro vykonávanou transakci (požadavek) na základě kontextuálních informací (např. potřeba logování v levelu DEBUG pouze pro uživatele XY). Dynamicky určený log level požadavku tedy jednotlivé aplikační moduly mohou pro danou transakci obdržet v HTTP hlavičce s názvem **X-Log-Level** daného volání. V případě podřazeného navazujícího volání dalších modulů musí modul tuto hlavičku dalším modulům také přeposlat.

Aplikační logika modulu musí zabezpečit, aby se pro logovací zprávy, pro které je potřeba náročnější příprava výstupních dat (např. serializace větších objektů do paměti atd.), tato příprava výstupních dat nevykonávala v případě, kdy odpovídající logovací zpráva nebude dle aktuálního log levelu zaslána na výstup (ať již z konfigurace nebo dle dynamicky vyžadované úrovně logování). Například, v úrovni DEBUG může v kódu probíhat logování celého obsahu HTTP transakcí. Je nežádoucí, aby se v případech, kdy je aktuální logovací úroveň nižší (ERROR až INFO), prováděla serializace obsahu HTTP transakcí do paměti, jelikož by to zcela zbytečně zabíralo jak procesorové, tak i paměťové prostředky systému při frekventovaném běhu takových transakcí.

Při použití některých externích knihoven může být komplikované zapisovat na výstup všechny zprávy v požadovaném strukturovaném formátu (knihovny můžou nějaké zprávy již generovat). Dodavatel by ale měl v součinnosti a na základě dohody se zadavatelem rozhodnout o odchytávání těchto zpráv a jejich zapisování na výstup v požadovaném formátu, pakliže to nebude představovat nepřiměřenou vývojářskou náročnost.

HTML/JS frontend aplikace standardně logují do vývojářské konzole a do bufferu, v produkci je log do konzole vypnutý.

#### 49.1.3 Tracing

Mezi hlavní cíle nasazení tracingu patří zobrazení trvání dotazů a souvisejících metadat, korelace latence napříč všemi moduly, dostupnost těchto informací i po kritickém selhání a odstranění kontejneru, nebo také vyhledávání informací týkajících se konkrétního modulu, kontejneru, časového rozmezí či požadavku.

Jako hlavní technologie pro implementaci tracingu v této architektuře systému jsou použité:

- OpenTelemetry
- Elastic APM

Je doporučeno pro implementaci modulů využít dostupnou instrumentaci pomocí OpenTelemetry SDK. Pro důležité logické celky aplikační logiky může být v zadávací dokumentaci modulu předepsána vlastní implementace nových měřených úseků (spanů). Dodavatel může taktéž navrhnout zadavateli vytvoření vlastních nových měřených úseků.

Sbírané tracing záznamy jsou odesílané knihovnou OpenTelemetry SDK do Elastic APM, kde jsou uchovávany, analyzovány a zobrazovány pro cílové uživatele těchto údajů. Přístup do Elastic APM (Kibana) bude v rámci DEV prostředí poskytnutý také dodavateli.

V případě nemožnosti použít OpenTelemetry SDK je možné vytvářet vlastní datové zprávy ve standardním otevřeném formátu OpenTelemetry a zasílat je pomocí standardních protokolů (OTLP/gRPC) do Elastic APM.

Trace musí obsahovat Correlation ID (viz sekce Síťový provoz), časovou značku, ID instance, unikátní ID v rámci celého systému, ID uživatele, status code, a informace o požadavcích vyvolaných tímto požadavkem. Případné další požadavky na obsah traců může být upřesněn v zadávací dokumentaci modulu.

Tracing daného modulu musí být vypínatelný a nastavitelný pomocí proměnných prostředí (konfigurace modulu).

#### 49.1.4 Metriky a monitoring

Mezi důležité vlastnosti zvolené architektury patří také sběr metrik ze všech provozovaných modulů, kde metrikou se rozumí číselná řada popisující nějakou vlastnost systému/modulu v čase. Příkladem mohou být počty přijatých a odeslaných transakcí, jejich průměrná/minimální/maximální délka trvání, počet otevřených uživatelských sezení, rychlost vyřizování jednotlivých databázových dotazů, četnost spouštění jobů na pozadí atd. Tyto metriky jsou sbírané (nebo odesílané modulem) v pravidelných časových intervalech (např. jednou za minutu). Průměrně složitý modul může poskytovat stovky různých metrik.

Mezi hlavní důvody sběru metrik patří:

- **diagnostika systému/modulů**
- rutinní sledování provozu systému pomocí administrátorských dashboardů zobrazujících aktuální (živý) stav systému a přehled o jeho aktivitě
- analýza a sledování selhání a neočekávaných stavů
- zobrazení statistických informací o četnosti selhaní v jednotlivých modulech
- poskytnutí informací, jež koncovému uživateli pomohou určit, zda chybný stav, jenž nastal, je způsoben chybou na straně systému nebo je externího původu
- vzájemná korelace statistických údajů všech modulů, instancí a HW prostředků
- **automatické notifikace v případě překročení povolených mezí vybraných metrik (alertování)**
- různé pohledy pro vlastníky systému, provozovatelé, správce infrastruktury a vývojáře
- **reporting**

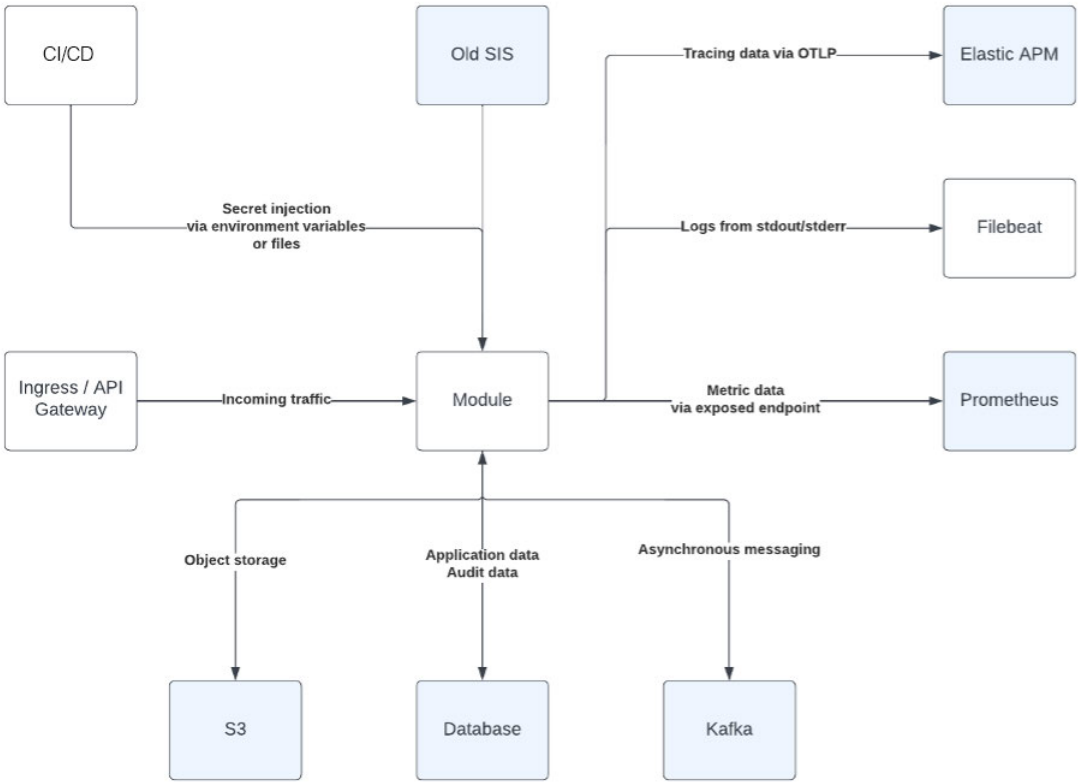
Tyto metriky jsou sbírané a ukládané ve specifické databázi určené pro ukládání numerických časových řad. Pro tento účel byla zvolena platforma Prometheus/Thanos. Agregované statistiky jsou automaticky analyzovány systémem Alertmanager, který v případě překročení očekávaných limitů automaticky vytvoří nový záznam (ticket) v systému Redmine a upozorní administrátory v aktuální on-call směně. Metriky je možné prohlížet pomocí aplikace Grafana, ke které v rámci DEV prostředí obdrží dodavatel přístup.

Pro automatický sběr musí každý modul vystavit endpoint s metrikami v Prometheus formátu. Alternativou je aktivní posílání metrik na Pushgateway. Forma odevzdávání metrik bude domluvena mezi zadavatelem a dodavatelem na začátku implementačního projektu daného modulu.

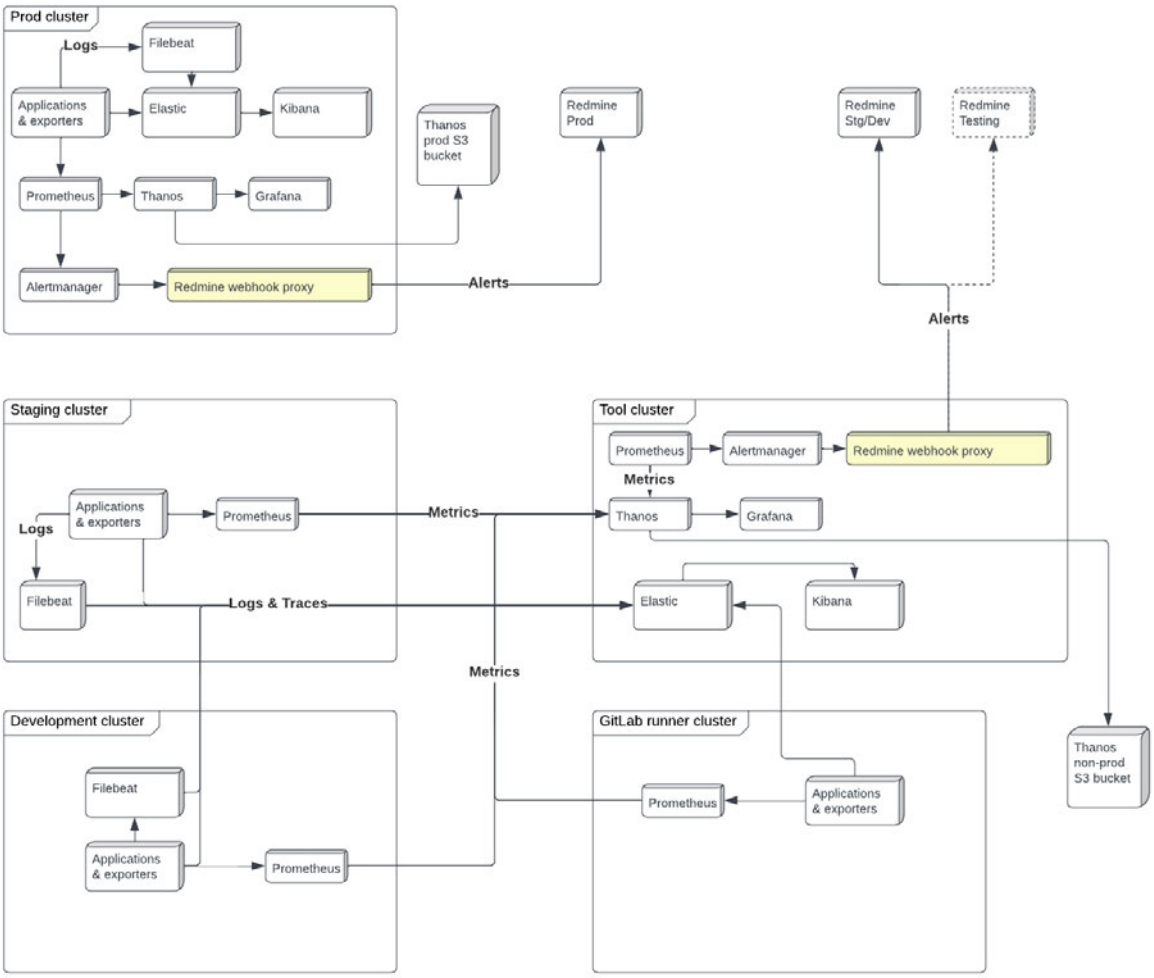
Mezi hlavní druh metrik budou patřit zejména informace o stavu, výkonu a vytížení modulu. Požadované metriky budou specifikovány v zadávací dokumentaci modulu. Dodavatel může po domluvě se zadavatelem dodat i další metriky související s implementací daného modulu pro informovanější monitoring a ladění systému. **Dodavatel musí dodat dokumentaci poskytovaných metrik (jednotky, způsob měření, přesná sémantika atd.) a doporučení pro nastavení alertů pomocí PromQL.**

#### 49.1.5 Diagramy primárních interakcí modulu

Následující diagram shrnuje hlavní komponenty infrastruktury, se kterými budou aplikační moduly přímo interagovat (mimo komunikaci s ostatními aplikačními moduly). Modře zvýrazněné komponenty vyžadují přímou integraci do kódu modulu.




Následující diagram popisuje tok metrik, logů, traců a alertů celým systémem. Sdílený monitoring pro Development a Staging cluster se nachází v Tool clusteru. Produkční cluster obsahuje svůj oddělený monitoring. Komponenty vyvíjeny interně na UK jsou zvýrazněny žlutě. Jednotlivé vyvíjené moduly jsou na diagramu zahrnuty pod komponentou “Applications & exporters”.




### 49.1.6 Reporting

Každý aplikační modul musí poskytovat data pro sdílenou reporting infrastrukturu v takové míře, aby bylo možné splnit požadavky na reporting od stakeholderů – typicky se bude jednat o data z monitoringu a auditu, nicméně ve vybraných případech i o jiný způsob vhodného poskytnutí dat pro potřeby požadovaných reportů, např. ve formě API definovaného k tomuto účelu. Veškeré detaily budou upřesněny v zadávací dokumentaci modulu.

### 49.1.7 Dostupnost a spolehlivost (High Availability)

Všechny aplikační moduly by měly podporovat provoz ve vysoké dostupnosti (HA). Každý modul by tedy mělo být možné provozovat ve vícero instancích zároveň a umožnit tak i horizontální škálování systému. Přesnější výkonové požadavky a propustnost jednotlivých endpointů (množství požadavků za časové okno, rychlost odezvy, atd.) budou specifikovány v zadávací dokumentaci daného modulu. V ideálním případě by všechny moduly měly být implementovány bezstavově (stateless) z důvodu jednoduché škálovatelnosti.

V některých konkrétních případech lze uvažovat i o implementaci bez podpory vysoké dostupnosti (např. kvůli zvýšení celkové propustnosti pro danou transakci za předpokladu, že požadovaný výkon není technologicky možné jinak docílit), nebo o implementaci vysoké dostupnosti modulů vyžadujících stavovost pomocí tzv. sticky sessions (nikoliv plně stateless). Každá taková výjimka však musí být schválena zadavatelem, a to pouze ve specifických a dobře odůvodnitelných případech.

Změnu konfigurace modulu a nasazení nové verze by mělo být ve většině případů možné bez vypnutí modulu (rolling update) pro zabezpečení co nejvyšší dostupnosti systému. Samozřejmě v odůvodněných případech je možné nasazení nové verze také po čas plánovaného výpadku (downtime), například v situacích vyžadujících náročnější změnu struktury relační databáze, migrace dat, atd.

## 50 Obecné technické požadavky na moduly

Každá aplikační doména (z pohledu koncového uživatele modul v Portálu SIS) bude implementována nejméně pomocí dvou modulů z pohledu runtime:

- **„Backendový“ modul** – zabezpečující implementaci byznys logiky, přístup k perzistentním datům, logickou autorizaci specifickou pro danou doménu, poskytování strukturovaných dat pro „frontendový“ modul generující uživatelské HTML rozhraní
- „**Frontendový“ modul** – zabezpečující HTML reprezentaci uživatelského rozhraní vloženého do hlavního Portálu SIS

Všechny moduly poběží v kontejnerizovaném prostředí Kubernetes jako nezávislé kontejnery. Pro každý modul (kontejner) bude vytvořen separátní GitLab projekt a CI/CD pipeline.

### 50.1.1 Technické požadavky na backendové moduly

Pro implementaci backendových modulů jsou povoleny následující technologie:

- Java/Kotlin (+ Spring Boot)
- TypeScript + Next.js/Nest.js
- Python + WSGI/ASGI
- C# + ASP.NET Core

### 50.1.2 Technické požadavky na frontendové moduly

Stěžejní základ pro implementaci frontendu pro specifickou portálovou aplikaci je zadavatelem dodána UX/UI analýza, jakožto součást zadávací dokumentace modulu. Pro dosažení jednotného designu bude dodavateli poskytnutá také knihovna společných komponent, vybudovaná s použitím konceptu Atomic Design[1], která bude postupně rozšiřována o nové komponenty a šablony.

Pro server-side část implementace frontendových modulů jsou povoleny stejné technologie jako pro implementaci backendových modulů, uvedené výše.

Pro browser-side část implementace frontendových modulů jsou povoleny následující technologie:

- HTML/CSS + Vanilla JS
- TypeScript + React

Uživatelská rozhraní musí podporovat zobrazování v různých jazykových mutacích. Jazyk zobrazení bude vyplývat z nastavení uživatelského profilu v rámci centrálního modulu Portál, který informaci o požadovaném jazyku přepošle specifickému frontend modulu pomocí cookie s názvem **sis_lang** v HTTP hlavičce požadavku. Není-li určeno jinak, modul musí poskytovat českou i anglickou mutaci uživatelského rozhraní. Případné další požadované jazykové mutace budou specifikovány v zadávací dokumentaci modulu.

Kromě jazyku zobrazení bude Portál posílat i následující HTTP hlavičky:

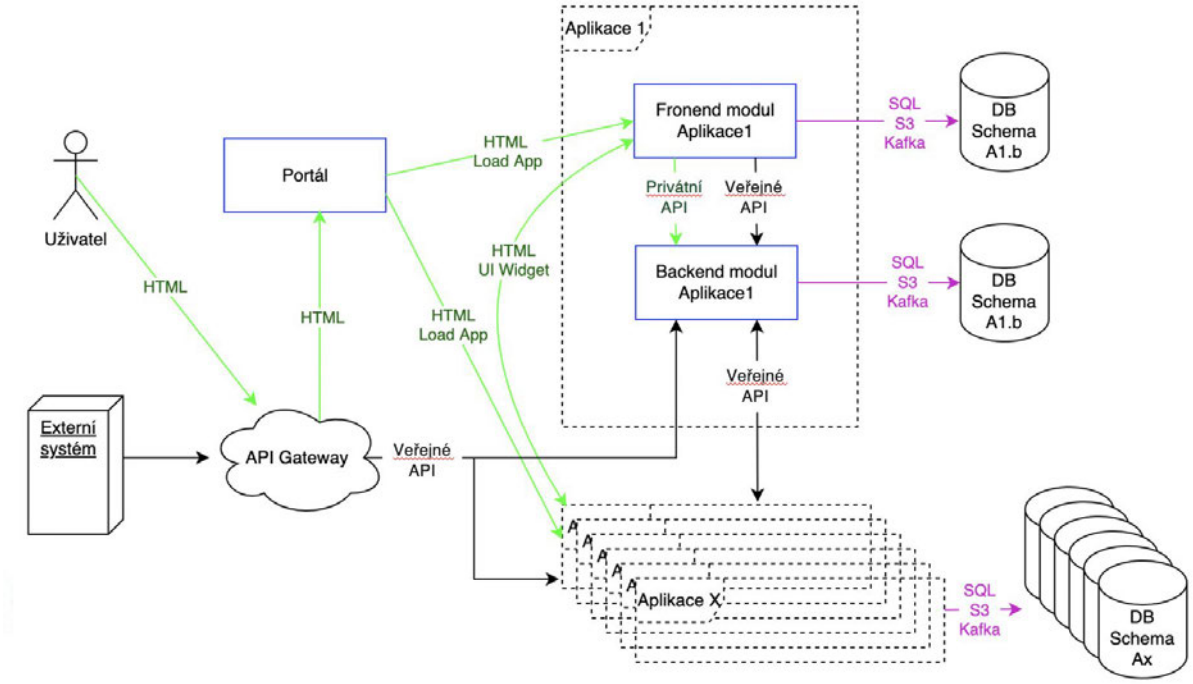
| Jméno HTTP hlavičky | Popis |
|---|---|
| X-Correlation-Id | Unikátní ID zpracovávaného požadavku, které bylo přirazeno pomocí platformy API brány při vstupu do systému. Všechny moduly v rámci architektury si při následných voláních tento identifikátor odevzdávají v HTTP hlavičkách. |
| X-Base-Path | Každá portálová aplikace bude mít přidělený nějaký namespace pro vytváření unikátních cest pro context-path HTTP requestů. Prefixem namespacu aplikace bude vždy namespace centrálního Portálu, jehož hodnota je portálovým aplikacím odevzdávaná pomocí této HTTP hlavičky. Příklad zasílané basePath: „/portal“. Jestliže přiděleným identifikátorem portálové aplikace je například „app1“, bude finální namespace aplikace vypadat následovným způsobem: „/portal/app1/“. Veškeré další zdroje, které prohlížet bude potřebovat pro Aplikaci1 dotáhnout musí jako prefix context-path používat tento namespace, na základě kterého Portál ví, které portálové aplikaci má přeposlat požadavek na daný zdroj. |
| X-Log-Level | Dynamicky určený logovací level pomocí kterého má modul logovat aktuálně procesovaný požadavek. V případě podřazeného navazujícího volání dalších modulů musí modul tuto hlavičku dalším modulům také přeposlat. Možné hodnoty jsou ERROR, WARN, INFO, DEBUG, TRACE. |
| X-Parent-Id | ID přímo nadřazeného rodičovské logovací zprávy. Jestliže je aktuální požadavek vykonáván jako podřazená operace nejaké nadřazené části aplikační logiky, pošle nadřazený modul nebo komponenta modulu identifikátor rodičovské logovací zprávy (kořen logovacího podstromu) pomocí HTTP hlavičky **X-Parent-Id**, nebo pomocí interních komunikačních cest mezi interními komponenty moduly. Od podřazené aplikační logiky se pak očekává vyplňování tohoto identifikátoru do všech vytvářených zpráv až do momentu kdy nevznikne nový logovací podstrom. Způsob vytváření a užití parent id je také demonstrován pomocí zdrojového kódu vzorových modulů, které budou dodavateli zpřístupněny v rámci přístupu do GitLab. |

### 50.1.3 Interakce mezi moduly

Následující obrázek ukazuje schéma jednotlivých API volání pro dva různé případy užití.

1. Prvním je uživatel (osoba) přistupující na centrální Portál SIS s cílem vstoupit do portálové Aplikace1. Moduly Aplikace1 poskytnou Portálu HTML kód, který Portál vloží do celkového rámce HTML stránky vrácené prohlížeči uživatele. Úkolem frontendového modulu Aplikace1 je zabezpečit uživatelské HTML rozhraní pro Portál, a to primárně za pomoci backendového modulu Aplikace1, implementujícího aplikační logiku Aplikace1, případně jiných modulů systému (ať již backendových poskytujících např. REST API, nebo frontendových poskytujících embedovatelné HTML widgety). Volání jejichž návratovou hodnotou je uživatelské rozhraní v HTML formátu je vyznačeno zelenou barvou. Na tomto místě je potřeba zmínit, že každý backendový modul může pro svůj frontendový modul vystavovat kromě zadávací specifikací předepsaného veřejného (aplikačního) API také privátní „frontendové“ API. Aplikační (veřejné) i neveřejné API pak společně poskytují frontendovému modulu potřebné funkce a data.

2. Druhým případem užití je přímé volání veřejného API systému externí aplikací. Volání veřejného (aplikačního) API jednotlivých modulů je vyznačeno černou barvou.




Příklad jednoduché ukázkové portálové aplikace poskytující uživatelské HTML rozhraní pro integraci do centrálního Portálu bude dodavateli zpřístupněno v rámci přidělení přístupu do platformy GitLab.

# 51 Obecné požadavky na uživatelská rozhraní

## 52 Obecné principy uživatelského rozhraní

Součástí zadávací dokumentace pro uživatelské rozhraní jsou:

1. Graficky zpracované náhledy pro rozpracovaný modul.
2. Samostatně zpracované komponenty, které se v modulu vyskytují

Grafická podoba zpracovaných náhledů není finální grafikou, ale pouze pracovním grafickým návrhem UI. Rozpracování finalní grafiky není součástí zadání jednotlivých modulů. Přesto obrazovky a komponenty budou mít jednotný vzhled a společně budou tvořit jednotný design systému, jehož správná a důsledná implementace je potřebná pro rychlé a správné nasazení finální grafiky.

Komponenty UI, ze kterých se budou skládát moduly v jednotlivých zadáních budou tvořit společnou knihovnu, která se bude postupně doplňovat s každým nově zadaným modulem. Všechny komponenty UI, přitomné v návrzích UI budou mít svůj zdroj v knihovně komponent.

Součástí knihovny budou "placeholdery" prvků určené pro poskytování zpětné vazby UI v případě delší odezvy modulů. Finální podoba těchto prvků bude řešená v rámci grafického návrhů. Základní zpětná vazba (spinner) má být uživateli poskytnutá všude tam kde se očekává odezva delší než 1000 ms.

## 53 Předání grafických podkladů pro UI

Zadávací dokmentace pro UI konkrétního modulu bude obsahovat:

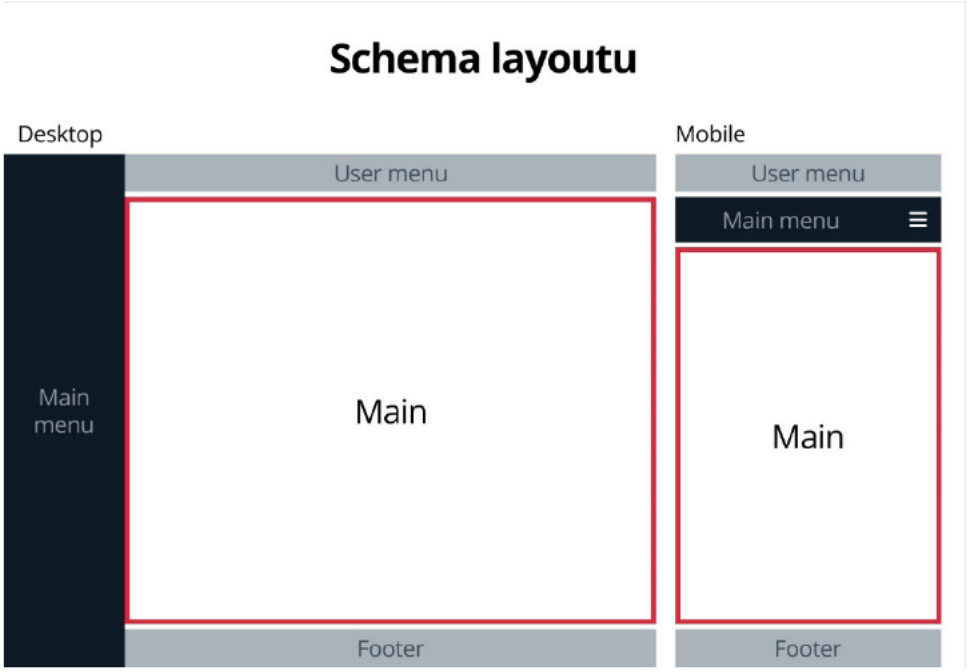
1. Exporty návrhů ve dvou variantách - desktop (šířka 1512 px) a mobil (šířka 414 px) a ve dvou formátech JPG a PDF.

2. Přístup ke zdrojovým souborům návrhů (Figma)Odkaz na soubor s komponenty společnými pro všechny doposud zpracované moduly v rámci projektu (Figma).
3. Flowchart obrazovek
4. Interaktivní prototyp ve Figmě

## 54 Uživatelské rozhraní - popis základního layoutu

V této části jsou popsané vlastnosti layoutu z hlediska responzivního chování aplikace.

Uživatelské rozhraní modulů Studijního Informačního Systému vychází z modulární architektury systému a je navrženo tak, aby umožňovalo postupné začleňování nově vytvořených modulů do aplikace. Skládá se ze dvou ovládacích panelů – jeden pro volby uživatelského profilu a druhý pro navigaci v modulech, a ze sekce, určené pro zobrazení zvoleného modulu. Tyto prvky tvoří jednotný základní layout aplikace.




Pro účely zadání je podstatná definice responzivního chování sekce, do níž bude umístěn HTML kód modulu. Obsah zbývajících sekcí layoutu dodá centrální webový Portál.

### 54.1.1 Základní prvky uživatelského prostředí

- Hlavní navigační panel (MAIN MENU), který obsahuje branding a hlavní nabídku, která slouží k výběru jednotlivých modulů.
- Panel uživatelského menu (USER MENU) – obsahuje informace o uživateli, volby pro přepínání mezi rolemi uživatele a volby pro nastavení dalších preferencí na uživatelské úrovni
- Jednotná patička (FOOTER)
- Hlavní panel modulu (MAIN), ve kterém se zobrazuje aktuální modul

### 54.1.2 Responzivita a breakpoints

Uživatelské rozhraní je založené na fluidním layoutu a počítá s intenzivním využitím flex CSS komponenty. Breakpoints (BP) jsou minimalizované a jsou uplatněné pro každou ze základních komponent zvlášť. Nevztahují se nutně k velikostem konkrétních zařízení, ale vychází spíše z potřeby komponent. BP pro základní sekce layoutu jsou následovné:

1. MAIN MENU BP: **max-width: 1199px**
    a. Pod tímto BP se panel zobrazuje nad hlavním obsahem a nabídka je schovaná pod ikonou.
    b. Nad tímto BP se panel zobrazuje vlevo od hlavního obsahu
2. USER MENU BP**: max-width: 799px**
    A. Pod tímto bodem jsou některé textové položky sdružené pod ikonku s dropdown menu
    B. Nad tímto bodem jsou tyto položky umístěné přímo v liště
3. MAIN BP: **max-width: 499px** – tento BP se týká především chování flex prvků. Od tohoto bodu dolů se ruší sloupce ve formulářích a maximální počet sloupců v ostatních komponentách se redukuje na 2.

Potřeba dalších BP může přibývat s rozpracováním konkrétních modulů, přesto snaha bude držet se co nejvíce již zmíněních.

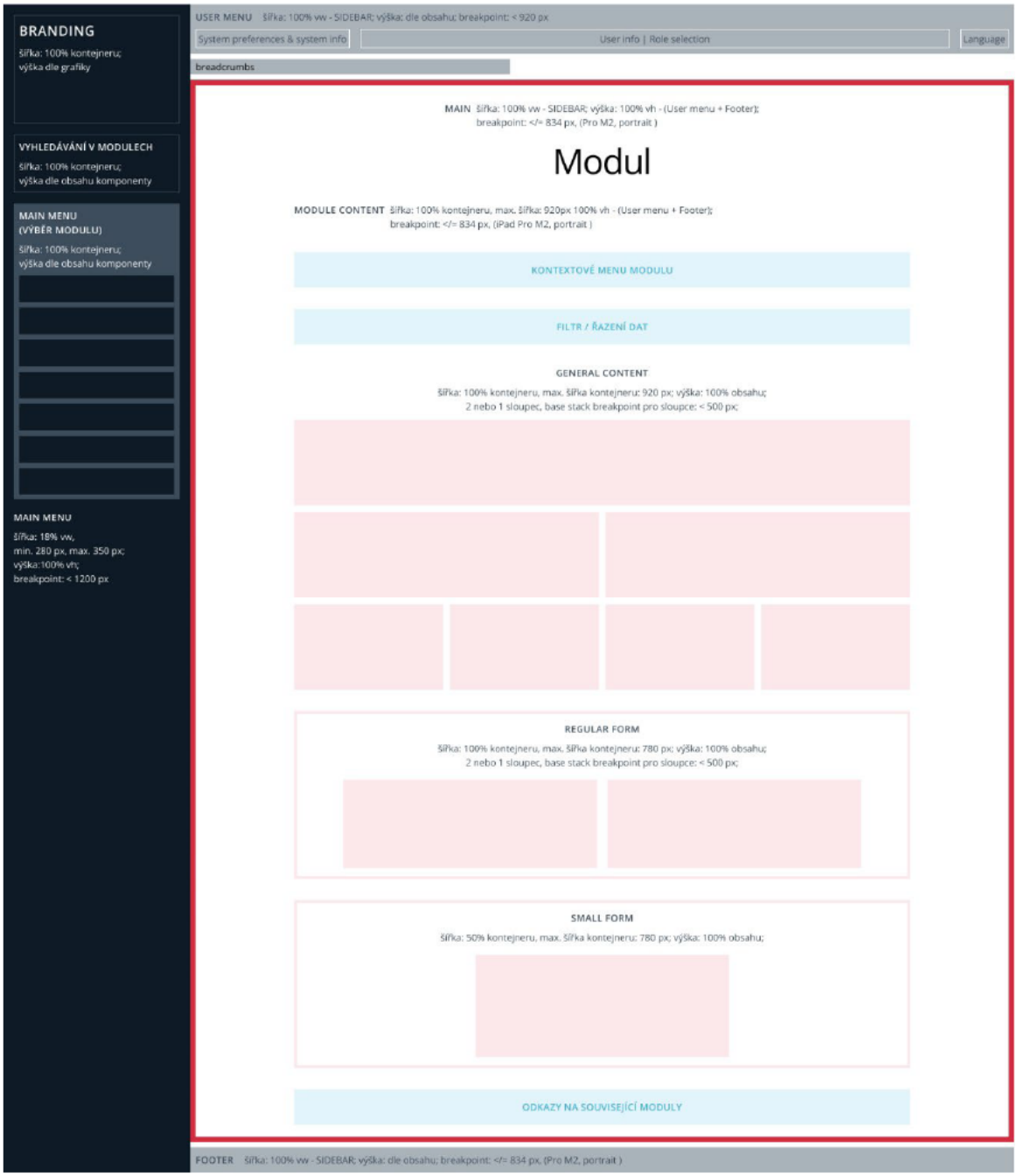
#### 54.1.3 Dimenze základních sekcí layoutu

1. MAIN MENU **18% vw, max. šířka 280px**
2. Komponenty v sekci MAIN **100% nadřazeného prvku, max. šířka 920px**
3. Formuláře **100 % nadřazeného prvku, max. šířka 780px**
4. Malé formuláře **100 % nadřazeného prvku, max. šířka 460px**

#### 54.1.4 Grafické náhledy

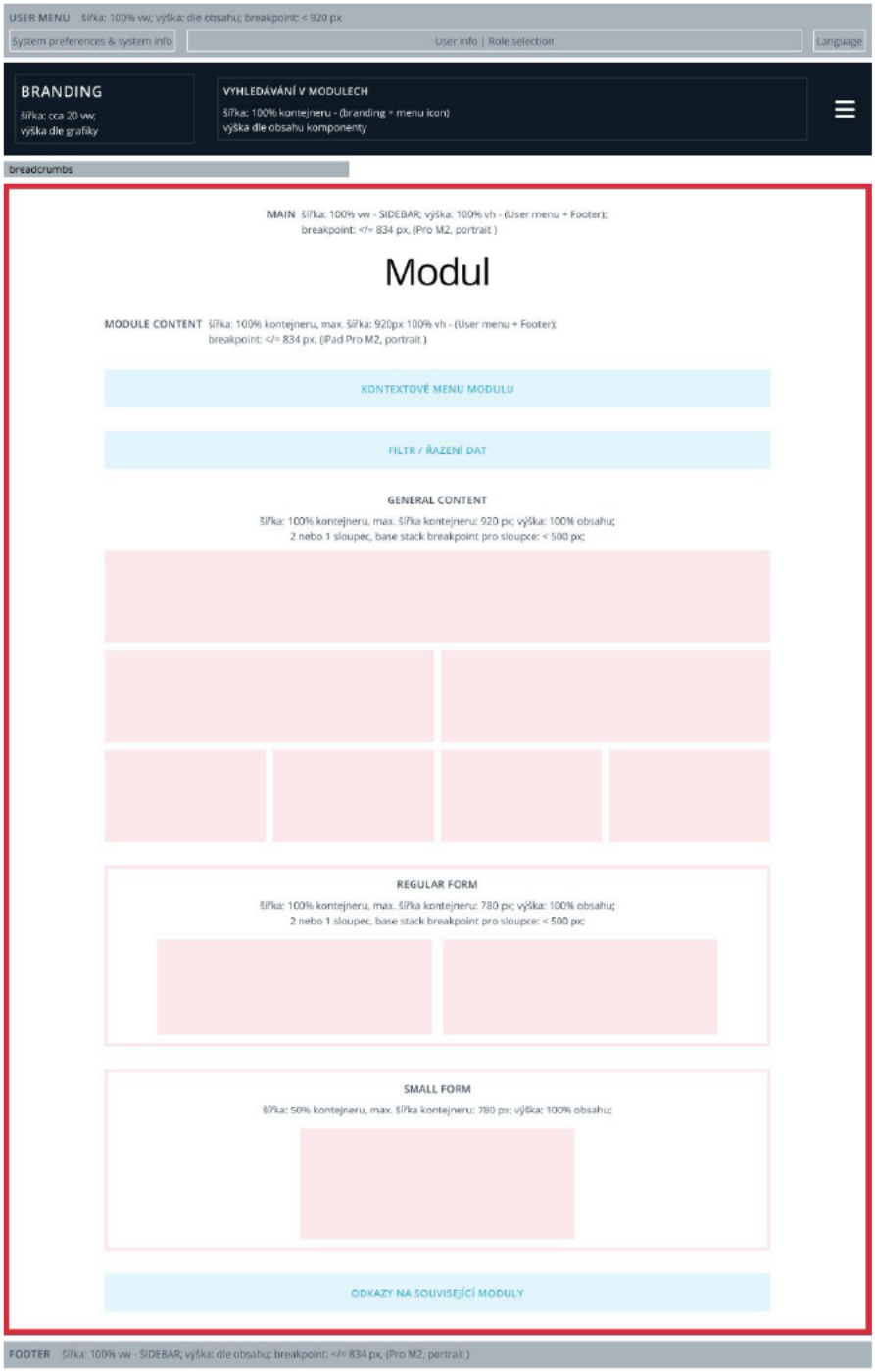
# Základní layout - desktop

šířka obrazovky  min. 1200 px



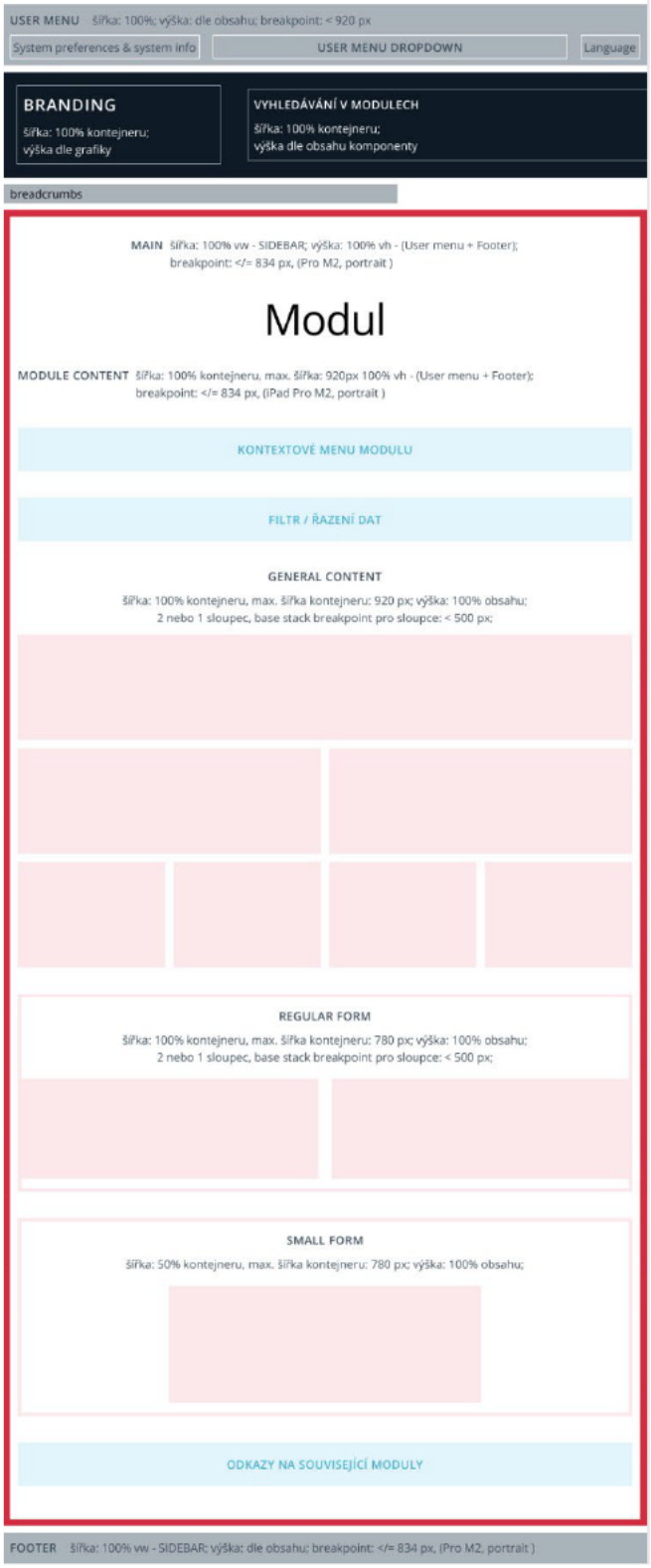
# Základní layout - tablet

šířka obrazovky < 1200 px



# Základní layout - small screen

šířka obrazovky < 800 px



# Základní layout - mobile

šířka obrazovky
</= 500 px

55 Požadavky na optimalizaci

Návrh uživatelského rozhraní (UI) aplikace musí brát v potaz fakt, že aplikace bude sloužit uživatelům s velmi odlišnými požadavky na zařízení a prohlížeče (zaměstnanci školy vs. studenti), proto je optimalizace většiny případu užití pro mobilní zařízení i pro desktop počítač/notebook stejně důležitá.

1) Desktop
   a) Windows / Chrome
   b) Windows / Edge
   c) Windows / Firefox
   d) OS X / Safari 15+
   e) OS X / Chrome
   f) OS X / Firefox
2) Mobile
   a) Android 10+ / Chrome
   b) iOS 13+ / Safari
   c) Android 10+ / Samsung internet

Tento seznam se zakládá na veřejných datech o nejpoužívanějších OS/prohlížečů v ČR a na interních uživatelských statistikách UK.

Minimální velikost obrazovky, pro kterou je aplikace určená je 375 x 667 pro mobilní zařízení a 768 x 1024 pro tablet. Aplikace se má zobrazovat s v souladu s grafických návrhem jak na desktopovém zařízení, tak na obrazovkách těchto velikostí.

56 Požadavky na přístupnost a použitelnost

Aplikace musí splňovat úroveň shody **AA** dle WCAG 2.0WCAG 2.0**.** Splněné požadavky na přístupnost je potřeba doložit protokolem z automatické kontroly specializované on-line služby, která bude specifikována dodatečně s dodavatelem. V rámci protokolu jsou pro dodavatele závazné parametry, které jsou předmětem jeho kompetencí a součástí jemu určeného zadání.

Zde je minimální závazný seznam parametrů schody s WCAG 2.0, relevantní pro dodavatele sw:

1. Správná sémantická struktura HTML kódu
2. Správné zvětšení obsahu při zvětšení/zmenšení
3. Podpora navigace pomocí klávesnice
4. Správné typy formulářových polí dle HTML5
5. Pokud se obsah automaticky mění, uživatel musí mít možnost pozastavit automatické změny, případně přepnout na jejích manuální ovládání
6. Možnost procházet obsah pomocí čtečky obrazovky
7. Možnost přeskočit navigaci pomocí klávesnice
8. Automaticky přesouvat focus na modální okna, vyskakovací okna, upozornění atd.
9. Prevence automatického odhlášení uživatele.
10. Prevence automatického přehrávání zvuku a videa, pokud to není žádoucí chování.
11. Důsledné použití atributu HTML lang
12. Používat atributy ARIA tam, kde struktura HTML není jednoznačná
13. Alternativní způsoby konzumace multimediálních souborů (transkripce dialogů a popis obsahu)
14. Data pro grafy, tabulky, mapy, SVG atd. konzumovatelná prostřednictvím asistenčních technologií

## 57 Obecné požadavky na strojová rozhraní modulů

Z důvodu vysoké modularizace systému je nutné sjednotit podobu strojových (programovacích) rozhraní jednotlivých modulů. Pod pojmem strojové rozhraní modulu rozumíme rozhraní umožňující strojovou (programovanou) interakci s modulem. Strojovou interakcí s modulem rozumíme komunikaci modulu s jinými

částmi systému (jiné moduly, front-end) za účelem výměny dat a volání operací. Rozlišujeme několik druhů

| Popis druhu strojového rozhraní | Název druhu strojového rozhraní | Vlastník | Specifikace |
|---|---|---|---|
| Volání aplikačních operací poskytovaných modulem pro potřeby výkonu funkcionalit systému zahrnujících CRUD operace nad daty spravovanými modulem a operace implementující části byznys logiky zajišťované modulem. | Veřejné aplikační strojové rozhraní (veřejné API) – určené pro jiné moduly systému | UK | OpenAPI + JSON Schema + mapování na sémantický byznys slovník |
| | Privátní aplikační strojové rozhraní (privátní API) – určené pro front-end část modulu | Dodavatel | OpenAPI + JSON Schema |
| Šíření a příjem notifikací o událostech, které nastaly uvnitř modulu a mají být pozorovatelné vně modulu, zahrnující události s významem odpovídajícím událostem nastávajícím v rámci byznys logiky zajišťované modulem. | Notifikační rozhraní | UK | JSON Schema + mapování na sémantický byznys slovník |
| Přístup k datům spravovaných modulem pro potřeby datově řízené univerzity | Datové rozhraní | UK | Datové schéma (CSVW, JSON Schema, …) + mapování na sémantický byznys slovník |

strojových rozhraní popsaných v následující tabulce.

Modul musí poskytovat alespoň jedno veřejné API. Může obsahovat více různých veřejných API lišících se specifikací, soukromých API, notifikačních rozhraní a datových rozhraní.

Vlastník rozhraní je plně odpovědný za jeho specifikaci sestávající z definice struktury programovacího rozhraní (definice operací, definice datových struktur) a dokumentace programovacího rozhraní. Nikdo jiný než vlastník rozhraní není oprávněn jeho specifikaci změnit.

Následují konkrétní obecné požadavky na jednotlivé druhy programovacích rozhraní. Požadavky jsou doplněny příkladem hypotetického modulu Zábava (Amusement) umožňujícího studentům nakupovat vstupenky na koncerty. Jedná se čistě o hypotetický příklad modulu, který není součástí zadání.

## 58 Obecná pravidla

Nejprve uvedeme obecná pravidla, která platí pro všechny druhy rozhraní.

### 58.1.1 Jmenné konvence

- Všechna URL endpointů rozhraní jsou definována relativně vůči základnímu URL SIS UK (tzv. base URL). Toto základní URL se může lišit dle prostředí, a proto jej neuvádíme a není pro návrh rozhraní podstatné.
- Všechny složky relativních URL (kroky cest, parametry) jsou uváděny v angličtině v kebab-case-notaci.
- Jediným povoleným formátem používaným veřejnými, privátními a notifikačními rozhraními je formát JSON. Jediným povoleným jazykem pro definici datových struktur pro formát JSON je jazyk JSON Schema ve verzi 2020-12 nebo pozdější.
- Názvy prvků datových struktur (klíče JSON, případně elementy a atributy XML a sloupce CSV) jsou uváděny v angličtině v kebab-case-notaci.

### 58.1.2 Chování klientů rozhraní

- Všechna URL endpointů rozhraní slouží jako identifikátory i lokátory. Nejsou však nositeli aplikační/byznysové sémantiky. V žádném případě není možné je na klientovi parsovat a na výsledku parsování stavět aplikační logiku.

## 59 Obecné požadavky na veřejná API

Veřejné API nabízí sadu operací, které umožňují vykonávání byznys operací, ze kterých sestávají byznys procesy/případy užití definované v kapitole byznys analýzy.

Veřejné API je z technického hlediska RESTful API, tj.

- reprezentuje byznys entity jako zdroje
- umožňuje CRUD manipulaci se zdroji,
- používá JSON pro reprezentaci zdrojů,
- používá URL pro identifikaci zdrojů,
- je bezstavové,
- používá HTTP metody POST, GET, PUT, PATCH, DELETE, OPTIONS a HEAD pro CRUD manipulaci se zdroji,
- implementuje HATEOAS princip (Hypertext As The Engine of Application State).

Veřejná API jsou verzovaná. Verze číslujeme sémanticky (semantic versioning).

### 59.1.1 Zdroje a jejich typy

Zdroj je základním technickým konceptem REST. Zdrojem je konkrétní nebo abstraktní věc, která je byznys entitou vyplývající z byznys analýzy, o níž chceme prostřednictvím veřejného API zpřístupnit údaje a umožňovat s ní CRUD manipulaci.

Uvažujeme následující typy zdrojů:

- zdroj typu objekt
  - reprezentuje byznys entitu nebo její část
    - např. student, koncert, interpret, vstupenka
- zdroj typu kolekce objektů
  - reprezentuje kolekci všech byznys entity daného typu
    - např. kolekci všech studentů, kolekci všech koncertů, kolekci všech interpretů, kolekci všech vstupenek
  - pro objekty v kolekci obsahuje pouze základní údaje, nikoliv jejich kompletní detail. Mezi základními údaji je vždy:
    - URL objektu (jedná se o zdroj)
    - typ případně typy objektu
    - byznysový identifikátor objektu
- zdroj typu vztah
  - reprezentuje byznys vztah dané byznys entity s jinou byznys entitou
    - např. vstupenka studenta (vztah studenta s jeho vstupenkou), majitel vstupenky (vztah vstupenky s jejím majitelem-studentem) nebo interpret vystupující na koncertu (vztah koncertu s interpretem, který na koncertu vystupuje)
- zdroj typu kolekce vztahů
  - reprezentuje kolekci vztahů, které reprezentují pro danou byznys entitu všechny její byznys vztahy odpovídající dané asociaci v byznys modelu
    - např. kolekce všech vstupenek daného studenta (0..*), kolekce všech majitelů dané vstupenky (0..1) nebo kolekce všech interpretů vystupujících na koncertu (1..*)
- zdroj typu ovladač (controller)
  - reprezentuje specifickou byznys operaci (akci) se zdrojem typu objekt, kterou není možné realizovat jako CRUD operaci
    - např. zneplatnění vstupenky.

Zdroj typu kolekce vztahů je kolekcí, jejíž prvky nereprezentují byznys entity vztažené k dané byznys entitě, ale vztahy samotné.

Zdroje typu ovladač by měly být využívány co nejméně – pouze v případech, kdy není možné danou byznys operaci realizovat jako CRUD operaci nad zdrojem typu objekt nebo kolekce. Např. byznys operaci změny majitele vstupenky na koncert může být realizována jako operace Update (HTTP PUT) na zdroji typu objekt reprezentujícím majitele vstupenky. Nezavádíme proto zdroj typu ovladač pro změnu majitele vstupenky. Zdroj typu ovladač může být nutný pro změnu stavu byznys entity, která není přímo realizovatelná jako prostý Update nějaké její části. Např. se může jednat o změnu stavu byznys entity na stav zneplatněno, tj. operace zneplatnění

vstupenky. Alternativou by bylo vytvoření zdroje typu objekt reprezentujícím platnost vstupenky. Pokud ale takový zdroj nemáme a nemůžeme jej z nějakého důvodu vytvořit, musíme vytvořit zdroj typu operace. Vždy ale preferujeme variantu vytvoření potřebných zdrojů, je-li to možné.

#### 59.1.2 Pravidla pro určování zdrojů a tvorbu jejich URL

Každý zdroj je vystaven prostřednictvím endpointu na URL, které je veřejné v rámci systému, tj. lze na něj přistoupit z jiných částí systému.

URL zdroje je identifikátorem zdroje v rámci celého systému a zároveň jeho lokátorem. URL zdroje má následující tvar:

```
[základní URL]/[relativní URL modulu]/public-api/[relativní URL veřejného API]/[verze API specifikace]/[relativní URL v rámci API]
```

kde

- `[základní URL]` je HTTP URL (tzv. base URL) společné pro všechny endpointy všech služeb a je určeno technickou infrastrukturou systému
- `[relativní URL modulu]` je relativní URL modulu, obvykle v podobě názvu modulu
- `[relativní URL veřejného API]` je relativní URL veřejného API v rámci modulu, obvykle v podobě názvu veřejného API, který musí být v rámci modulu unikátní. Ve výjimečných případech může být vynecháno, ale je to považováno za antipattern. Pokud v budoucnu dojde k rozhodnutí vytvořit druhé veřejné API, bude toto druhé muset být pojmenováno, čímž dojde k nekonzistenci v konvenci tvorby URL
- `[verze API specifikace]` dle semantic versioning. Změna verze znamená změnu veřejného API. Změna modulu při zachování API nemění její verzi.
- `[relativní URL v rámci API]` je relativní URL daného zdroje, s nímž je možné prostřednictvím veřejného API manipulovat. Je unikátní v rámci veřejného API.

Základními pravidly při určování zdrojů jsou následující pravidla.

- Pro typ byznys entity popsaný v byznys analýze určíme
  - zdroj typu kolekce objektů, který reprezentuje kolekci všech instancí tohoto typu. Platí, že `[relativní URL v rámci API]` = `[název typu v množném čísle]`, např.:
    - `students`
    - `interprets`
    - `tickets`
    - `concerts`
  - zdroje typu objekt, které reprezentují jednotlivé instance typu (tj. jednotlivé byznys entity). Platí, že `[relativní URL v rámci API]` = `[název typu v množném čísle]/{byznysový identifikátor pro tento typ byznys entity}`, např.:
    - `students/12345678`, kde 12345678 je číslo osoby (UKČO) studenta
    - `interprets/pokac`, kde pokac vyplývá z názvu interpreta, který je unikátní
    - `tickets/v-xzfhrd`, kde v-xzfhrd je kód vstupenky
    - `concerts/4758`, kde 4758 je číslo koncertu
- Pro asociaci mezi dvěma typy byznys entit popsanou v byznys analýze a její navigovatelný konec (v UML diagramech označeno šipkou, T je typ navigovatelného konce, S je druhý typ v asociaci) určíme
  - zdroj typu kolekce vztahů, který reprezentuje kolekci všech vztahů modelovaných asociací mezi danou instancí typu S a instancemi typu T. Platí, že `[relativní URL v rámci API] = [relativní URL v rámci API zdroje typu objekt reprezentujícího danou instanci typu S]/[název role asociačního konce nebo název asociace jako podstatné jméno]`. V případě maximální kardinality asociačního konce == 1 je název v relativním URL uveden v jednotném čísle. V případě > 1 je uveden v množném čísle. Např.:
    - `students/12345678/tickets`
    - `tickets/k-xzfhrd/owner`
    - `concerts/4758/interprets`
  - zdroje typu vztah, které reprezentují jednotlivé vztahy modelované asociací ve směru navigovatelného konce. Platí, že `[relativní URL v rámci API] = [relativní URL`

`v rámci API zdroje typu objekt reprezentujícího danou instanci typu S]/[název role asociačního konce nebo název asociace jako podstatné jméno]/{byznysový identifikátor pro typ T}`. V případě maximální kardinality asociačního konce == 1 je název role v relativním URL uveden v jednotném čísle. V případě > 1 je uveden v množném čísle, např.:
    - `students/12345678/tickets/k-xzfhrd`
    - `tickets/k-xzfhrd/owner/12345678`
    - `concerts/4758/interprets/pokac`
- Pro specifickou byznys operaci/akci se zdrojem typu objekt, kterou není možné realizovat jako CRUD operaci, určíme zdroj typu ovladač. Platí, že `[relativní URL v rámci API] = [relativní URL v rámci API zdroje typu objekt]/[název byznys operace/akce jako podstatné jméno v jednotném čísle]`, např.:
    - `tickets/k-xzfhrd/invalidation`

Výše uvedené příklady jsou relativní URL v rámci API. Jejich plné relativní URL v rámci systému je pak tvořeno dle výše uvedeného předpisu relativním URL modulu, relativním URL API v rámci modulu, verzí API a relativním URL v rámci API. Náš příklad je v kontextu modulu Zábava, tj. plná relativní URL jsou např.:

- `amusement/public-api/basic/1.0.0/students`
- `amusement/public-api/basic/1.0.0/students/12345678`
- `amusement/public-api/basic/1.0.0/students/12345678/tickets`

Výše uvedené příklady ukazují, že byznysové identifikátory v relativních URL mohou být nečíselné. Pokud je to možné, preferujeme číselné byznysové identifikátory. Jejich (ne)existence však vyplývá z byznys analýzy. Pokud není v byznys analýze identifikován žádný číselný identifikátor, volíme nečíselný.

Použití byznysových identifikátorů je nutné. Pokud bychom použili umělé databázové primární klíče, hrozilo by pronikání interní databázové logiky modulu do systému, což je zakázáno.

Výše uvedená pravidla neaplikujeme maximalisticky. Není nutné vytvářet zdroje pro každý typ byznys entity a pro každou byznys asociaci. Pravidla aplikujeme dle potřeby. Není např. nutné vytvářet zdroje pro každý navigovatelný konec každé asociace. V případě jednodušších API můžeme mít dokonce jen jeden zdroj typu kolekce objektů (např. kolekce studentů) a pro každý objekt v kolekci zdroj (např. jednotlivé studenty). Prvky vnitřní struktury zdrojů (např. kontaktní údaje, bankovní účty) nemusíme nutně reprezentovat jako samostatné zdroje. Vždy vycházíme z byznys analýzy a v ní identifikovaných byznys operací.

Pokud ale nějaký zdroj určíme, musíme ho určit podle jednoho z výše uvedených pravidel. Určování jiných zdrojů podle jiných pravidel není povoleno.

#### 59.1.3 Pravidla pro návrh operací pro manipulaci se zdroji

Definice CRUD operací musí vyplývat z byznys analýzy. Každá určená CRUD operace je mapována na odpovídající byznys operaci/akci popsané v byznys analýze. Při určování CRUD operací pro manipulaci s jednotlivými typy zdrojů se řídíme následující tabulkou.

| | **Kolekce objektů** | **Objekt** | **Kolekce vztahů** | **Vztah** | **Ovladač** |
|---|---|---|---|---|---|
| POST (Create) | Vytvoření nového objektu | N/A | N/A | N/A | Vykonání ne-idempotentní operace na daném objektu |
| GET (Read) | Čtení kolekce objektů | Čtení detailu objektu | Čtení kolekce se základními údaji o vztažených objektech | N/A | N/A |
| PUT (Update, příp. idempotentní Create) | N/A | Aktualizace objektu poskytnutím jeho nové reprezentace | Vložení vztahu s objektem do kolekce. Pro kardinality ..1 to znamená zároveň odstranění existujícího vztahu, pokud existuje. | N/A | Vykonání idempotentní operace na daném objektu |
| PATCH (Update) | N/A | Aktualizace části údajů o objektu poskytnutím nových hodnot vybraných údajů | N/A | N/A | Vykonání idempotentní operace na daném objektu |
| DELETE (Delete) | N/A | Smazání objektu | N/A | Smazání vztahu | N/A |

#### 59.1.4 Pravidla pro stránkování a filtrování kolekcí pomocí parametrů v URL

Operace pro čtení kolekce (GET) může bez dalšího znamenat načtení dat o velkém množství byznys entit. Pokud je potřeba množství omezit, je možné URL zdroje typu kolekce doplnit stránkovacími a filtrovacími parametry.

Stránkovacími parametry jsou

- `_offset`, jehož hodnotou je celé číslo >= 0, určuje pozici prvního prvku kolekce, který bude vypsán ve výsledku,
- `_limit`, jehož hodnotou je celé číslo > 0 které je shora omezeno (každou službou zvlášť), určuje počet prvků, které budou vypsány ve výsledku,
- `_order-by`, jehož hodnotou je název vlastnosti prvků kolekce s primitivní hodnotou, podle které má být kolekce seřazena pro účely stránkování
    - pokud není parametr uveden, vyplývá seřazení z interní reprezentace dat v modulu, přičemž je takové seřazení zdokumentováno v dokumentaci modulu
    - seznam požadovaných atributů dle kterých je možné řadit bude explicitně stanoven v rámci definice API

Dokumentace zdroje typu kolekce uvádí, zda podporuje stránkování a stránkování s explicitní specifikací seřazení vč. seznamu vlastností, podle kterých je možné třídit. Musí se jednat o vlastnosti definované ve specifikaci datové struktury pro reprezentaci kolekce.

Filtrovacími parametry jsou

- `_type`, jehož hodnotou je seznam řetězců, které reprezentují typy prvků, které se mohou vyskytovat v kolekci a dle kterých je možné kolekci filtrovat
- další parametry odpovídající vlastnostem prvků kolekce s primitivní hodnotou, názvy parametrů nesmí začínat znakem _, který je jako prefix rezervovaný

Dokumentace zdroje typu kolekce uvádí, zda podporuje filtrování podle typu, včetně výčtu možných typů, jejich popisu a vazby na typy byznys entit. Dále uvádí, zda podporuje filtrování podle dalších parametrů, včetně výčtu vlastností, které lze použít jako filtrovací parametry. Musí se jednat o vlastnosti definované ve specifikaci datové struktury pro reprezentaci kolekce.

#### 59.1.5 Pravidla pro definici zdrojů a operací pro manipulaci se zdroji

Veřejné API pro manipulaci se zdroji je formálně definováno s pomocí OpenAPI verze 3.0.3 nebo vyšší (dále jen Open API). Definice je vyjádřena jako YAML dokument nebo sada YAML dokumentů, která popisuje tvar URL zdrojů a možné CRUD operace nad těmito zdroji tak, jak byly navrženy dle výše uvedeného postupu. Vstupní parametry, které jsou součástí URL jsou definovány také jako součást specifikace. Vstupní parametry, které jsou předávány v těle požadavků, a výstupní reprezentace zdrojů, které jsou předávány v těle odpovědí, jsou reprezentovány jako JSON dokumenty, jejichž JSON struktura je popsána dle následující kapitoly a je obsažena v definici API nebo v samostatných souborech a je odkazována.

#### 59.1.6 Pravidla pro návrh a popis JSON datových struktur

Zdroje a vstupní parametry CRUD operací předávané v tělech požadavků jsou reprezentovány v datovém formátu JSON. Datová struktura pro reprezentaci zdroje je navržena odvozením z odpovídající části konceptuálního modelu, jehož struktuře odpovídá. Může být doplněna o technické datové prvky bez sémantického významu zachyceného v konceptuálním modelu.

Datová struktura je popsána datovým schématem vyjádřeným v jazyku JSON Schema. Datové prvky definované ve schématu, které nejsou technickými datovými prvky, jsou napojeny na prvky konceptuálního modelu. Datová schémata tak nepopisují pouze syntax ale také sémantiku pomocí pojmů definovaných v konceptuálním modelu z byznys analýzy.

Existuje katalog datových struktur, které je povinné používat při definici nových datových struktur. Katalog definuje základní datové struktury, např. pro reprezentaci specifikace časových údajů, množství, adres, apod.

Operace na zdrojích musí na vstupu (v rámci těla HTTP požadavku) i na výstupu (v rámci těla HTTP odpovědi) data o zdrojích reprezentovat v k tomu navržených datových strukturách dle odpovídajících JSON schémat. Každá operace je tak doplněna o definici vstupní a výstupní datové struktury. Vstupy a výstupy jednotlivých volání operací musí být validní vůči JSON schématům definujících tyto datové struktury. Požadavek na veřejné API s nevalidním vstupem musí být odmítnut s příslušným chybovým HTTP kódem.

#### 59.1.7 Pravidla pro implementaci principu HATEOAS

Dle principu HATEOAS musí být pro daný zdroj specifikována volání možných operací nad daným zdrojem a možných operací nad souvisejícími zdroji způsobem, který nevyžaduje, aby byla byznys logika volání operace implementována na straně klienta. K implementaci principu HATEOAS lze přistoupit staticky nebo dynamicky.

- Dynamický přístup znamená, že možné operace jsou uvedeny přímo, konkrétně a jednoznačně v reprezentacích zdrojů, které jsou vraceny v tělech odpovědí na volání operací. K tomu lze využít RDF 5988, ale konkrétní technický standard bohužel neexistuje.
- Statický přístup znamená, že možné operace jsou definovány v definici zdrojů. K tomu lze využít OpenAPI konstrukt zvaný Link. Pro získání konkrétních volání je nutné definici možných operací na klientovi interpretovat dle logiky konstruktu Link. Interpretační logika musí být tedy reprezentována na straně klienta, nicméně se nejedná o byznys logiku, takže není porušen princip HATEOAS.

V našem případě volíme následující kombinaci obou přístupů:

- Dynamický: V reprezentaci zdroje uvádíme URL zdroje a souvisejících zdrojů, které může klient použít pro volání operací čtení zdroje (GET). Pokud je URL uvedeno, operace HTTP GET je definována.
- Statický: Všechny ostatní možné operace nad daným zdrojem a nad souvisejícími zdroji definujeme jako součást OpenAPI definice veřejného API.

## *60 Dynamický HATEOAS pro operaci čtení*

Do JSON datové struktury reprezentující zdroj jakéhokoliv typu vkládáme do místa, ve kterém referencujeme související zdroj typu objekt, jeho URL. URL je buď absolutní nebo relativní vůči URL veřejného API. Je uvedeno jako hodnota JSON klíče `_id`. Tento klíč pochází ze specifikace JSON-LD, kde je používán na identifikaci zdrojů reprezentujících konkrétní nebo abstraktní entity. V případě, že klient při zpracování JSON dokumentu s reprezentací zdroje narazí na klíč `_id`, interpretuje jej jako URL zdroje typu objekt. Pomocí HTTP GET požadavku na toto URL získá detailní reprezentaci odkazovaného objektu.

Pro zvýšení jednoznačnosti interpretace doplňujeme klíč `_id` klíčem `_type`. Ten také pochází ze specifikace JSON-LD. Jeho hodnotou je sémantický typ nebo pole sémantických typů identifikovaného objektu. Sémantické typy jsou katalogizovány v katalogu sémantických typů. Katalog sémantických typů uvádí pro každý sémantický typ jeho lidsky čitelný název a popis a propojuje jej na odpovídající prvek nebo prvky z byznys analýzy, typicky na odpovídající typ byznys entit.

Následující příklad demonstruje dynamický HATEOAS v reprezentaci vstupenky zdrojem `vstupenky/v-xzfhrd`. Klient získává detailní reprezentaci vstupenky, přičemž o měně ceny vstupenky a o jejím majiteli získává pouze základní informace. Detailní informace může získat voláním operace čtení (GET) na URL definovaných v klíči `_id`. Klíč `_type` může použít pro určení správné prezentační logiky pro uvedené sémantické typy.

```
{
    "_id": "tickets/v-xzfhrd",
    "_type": "ticket",
    "code": "v-xzfhrd",
    "price": {
        "value": 799,
        "currency": {
            "_id": "https://číselníky.cuni.cz/číselník/měny/položka/czk",
            "_type": "Měna",
            "notace": "CZK"
        }
    },
    "owner": {
        "_id": " studenti/12345678",
        "_type": ["student", "doctoral-student"],
        "personal-number": "12345678",
        "full-name": "Jan Evangelista Purkyně",
    }
}
```

Následující příklad demonstruje dynamický HATEOAS v reprezentaci studenta zdrojem `students/12345678`. Klient získává detailní reprezentaci studenta, přičemž o vstupenkách studenta získává pouze základní informace. Detailní informace o vstupence může získat voláním operace čtení (GET) na URL definovaných v klíči `_id`. Klíč `_type` může použít pro určení správné prezentační logiky pro uvedený sémantický typ.

```
{
    "_id": " studenti/12345678",
    "_type": ["student", "doctoral-student"],
    "personal-number": "12345678",
    "full-name": "Jan Evangelista Purkyně",
    "birth-date": "1787-12-17",
    "tickets": [
        {
            "_id": " tickets/v-xzfhrd",
            "_type": "ticket"
        },{
            "_id": " tickets/v-zuadre",
            "_type": "ticket"
        }
    ]
}
```

Následující příklad demonstruje dynamický HATEOAS v reprezentaci kolekce vstupenek zdrojem `vstupenky`. Klient získává základní informaci o jednotlivých vstupenkách. Detailní informace může získat voláním operace čtení (GET) na URL definovaných v klíči `_id`. Klíč `_type` může použít pro určení správné prezentační logiky pro uvedené sémantické typy.

```
[
    {
        "_id": "tickets/v-xzfhrd",
        "_type": "ticket",
        "kód": "v-xzfhrd"
    },{
        "_id": "tickets/v-zuadre",
        "_type": "ticket",
        "code": "v-xzfhrd"
    }
]
```

### *61 Statický HATEOAS pro CRUD operace*

Výše uvedený přístup naplňuje princip HATEOAS pouze částečně, protože pokrývá pouze operace čtení zdrojů. Proto zavádíme ještě jeden přístup, kterým je statická specifikace volání dalších operací. V rámci OpenAPI definice veřejného API doplňujeme k definici každé operace nad zdrojem také statické definice toho, jaké další operace nad jakými dalšími souvisejícími zdroji můžeme volat poté, co obdržíme odpověď na její volání. Např.

- k definici operace čtení kolekce studentů doplníme statickou definici dalších možných operací
  - čtení detailů jednotlivých studentů
  - operace vložení nového studenta do kolekce
- k definici operace čtení studenta doplníme statickou definici dalších možných operací
  - úprava studenta
  - smazání studenta

Konkrétní sada možných operací je závislá na typu zdroje:

- pro zdroj typu objekt jsou definovány následující operace
  - pro operaci čtení objektu (GET)
    - aktualizace daného objektu (PUT), pokud je podporována
    - smazání daného objektu (DELETE), pokud je podporováno
    - pro každý typ byznys entit, kterého je objekt instancí a pro který byl určen zdroj typu kolekce objektů
      - čtení kolekce objektů, ve které jsou reprezentovány všechny instance tohoto typu byznys entit (GET)
    - pro každý konec byznys asociace, který má počátek v typu byznys entity, jehož je objekt instancí, a pro který byl určen zdroj typu kolekce vztahů
      - čtení kolekce vztahů, ve které jsou reprezentovány vztahy odpovídající byznys asociaci spojující objekt s jinými objekty (GET)
      - vytvoření nového vztahu v kolekci vztahů, pokud je podporováno (PUT)
    - pro každou specifickou byznys akci s objektem pro kterou byl určen zdroj typu ovladač
      - vykonání byznys akce (PUT/POST)
  - pro operaci aktualizace objektu (PUT/PATCH)
    - čtení objektu (GET)
- pro zdroj typu kolekce objektů reprezentujících množinu instancí daného typu byznys entit jsou definovány následující operace
  - pro operaci čtení kolekce (GET)
    - vytvoření nového objektu v kolekci (POST), pokud je podporováno
    - pro každý objekt v kolekci

            - čtení detailu objektu (GET)
        - pro operaci vytvoření nového objektu v kolekci (POST)
            - čtení objektu (GET)
- zdroj typu kolekce vztahů reprezentujících množinu vztahů odpovídajících byznys asociaci spojující objekt s jinými objekty
    - pro operaci čtení kolekce (GET)
        - vytvoření nového vztahu v kolekci vztahů, pokud je podporováno (PUT)
        - pro každý vztah v kolekci
            - smazání existujícího vztahu z kolekce vztahů, pokud je podporováno (DELETE)

Příklad specifikace veřejného API hypotetického modulu Zábava je uveden v příloze „Příklad specifikace veřejného API ve formátu OpenAPI 3.yaml“ tohoto dokumentu.

62 Obecné požadavky na privátní API

Privátní API je kolekcí operací, které naplňují specifické potřeby uživatelského front-end daného modulu. Privátní API je určeno pouze a jenom pro potřeby front-end daného modulu. Nesmí být použito pro žádné jiné účely. Nesmí zajišťovat funkcionality, které jsou poskytovány veřejným API, tj. privátní API ani žádná jeho část nesmí striktně ani volně kopírovat funkcionality veřejného API nebo jeho části.

Protože není vystaveno veřejně do systému, nejsou na něj kladeny tak striktní požadavky jako na veřejné API.

Technicky se jedná o JSON API, které

- umožňuje čtení dat reprezentujícím byznys entity,
- umožňuje operace pro čtení i zapisování dalších dat, které nutně nevyplývají z byznys analýzy, ale jsou potřebné pro správné fungování uživatelského fronte-endu daného modulu,
- používá JSON pro reprezentaci dat,
- je bezstavové.

Privátní API musí být zdokumentováno a datové struktury pro reprezentaci dat v datovém formátu JSON musí být popsány v jazyku JSON Schema. Pokud nějaká datová struktura privátního API sémanticky odpovídá nějakému typu byznys entit popsaných v byznys analýze, musí být tato souvislost zaznamenána v dokumentaci.

Zadavatel si vyhrazuje právo z privátního API nebo jeho části vytvořit veřejné API. Zodpovědnost za privátní API nebo jeho část při využití tohoto práva převezme za API odpovědnost a zajistí, aby splňovalo výše uvedené podmínky na veřejné API. Dodavatel, který privátní API nebo část využíval, musí svůj kód potřebným způsobem upravit tak, aby pracoval s vytvořeným veřejným ekvivalentem.

# 63 Zajištění jakosti (QA) a dokumentace

64 Obecné požadavky na kvalitu kódu a bezpečnost

### 64.1.1 Kvalita kódu

Zdrojový kód každého modulu musí procházet kontrolou linteru a nástroje na statickou analýzu kódu s cílem odhalení chyb již v čase kompilace. Tento nástroj musí odpovídat zvolené technologii implementace daného modulu.

Zdrojový kód každého modulu musí procházet kontrolou nástroje na formátování odpovídající standardům a dobrým zvykům pro zvolenou technologii implementace.

### 64.1.2 Bezpečnost

Požadavky na implementaci modulů:

1. Modul musí být implementovaný tak, aby byl odolný vůči známým bezpečnostním hrozbám
2. Kód modulu prochází kontrolou **Static Vulnerability Scan**
    - Neobsahuje závažné/critical chyby
3. Kód prochází kontrolou **Dependency Vulnerability Check**
    - Neobsahuje závažné/critical chyby
4. Kód prochází kontrolou **Dynamic Vulnerability Analysis**

    - Neobsahuje závažné/critical chyby
5. Modul používá knihovny třetích stran, k nimž je poskytována LTS (long term support), nebo u nichž lze předpokládat střednědobá podpora (5-10 let) – například na základě popularity použití dané knihovny v rámci komunity

Pro prokázání splnění podmínek uvedených v bodech 2, 3, 4 dodavatel dodá pro každou novou verzi reporty z úspěšného ověření bezpečnosti pro jednotlivé položky. U false-positive případů uvedených v dodaných reportech bude poskytnuto písemné odůvodnění proč dodavatel považuje daný bod reportu za false-positive. Zadavatel bude namátkově a v některých případech automaticky pravidelně provádět tuto kontrolu také.

Následující tabulka uvádí přehled typických nástrojů pro provádění požadovaných skenů a reportů.

| **Druh skenu** | **Typické nástroje** |
|---|---|
| Static Vulnerability Scan | IBM AppScan, Fortify, Veracode, FindSecBugs, Brakeman, Bandit |
| Dependency Vulnerability | OWASP Dependency Check, Bundler Audit, Safety |
| Dynamic Vulnerability Analysis | Burp Suite, OWASP ZAP, HP WebInSpect |

65 Obecné požadavky na dokumentaci

Pro každý modul musí být dodavatelem odevzdána následující dokumentace:

1. Instalační a konfigurační příručka
    - Znalost systémových požadavků
    - Znalost způsobu konfigurace modulu
    - Znalost postupu instalace na lokální stanici a ve vývojovém prostředí

2. Dokumentace funkčních a nefunkčních požadavků
    - Znalost happy paths i unhappy paths
    - Napomáhá odhalení regresních závad na úrovni modulu

3. Dokumentace privátního (frontendového) i veřejného API standardizovaným strojově čitelným formátem
    - Verzované API v gitu
    - Znalost API zachycená pro vývojáře ostatních modulů
    - Možnost generovat klienty a mock servery pro dané API automaticky

4. Dokumentace umístění dat používaných modulem a možnosti jejich anonymizace (tam kde je to aplikovatelné)
    - Pro potřeby zálohování a správy dat využívaných modulem
    - Splnění požadavků dle GDPR před provedením kopií, záloh atd.

5. Dokumentace technické architektury modulu/aplikace
    - Interní architektura modulu
    - Komunikace s jinými moduly
    - Možnosti provozu modulu v HA
    - Popis autorizačních rozhodnutí prováděných modulem

6. Dokumentace generovaných auditních událostí

7. Dokumentace generovaných metrik
    - Popis všech metrik (jednotky, způsob měření, přesná sémantika atd.)
    - Doporučení pro nastavení alertů pomocí PromQL

8. Dokumentace vlastních přidaných datových polí (fieldů) do strukturovaných logovacích záznamů

9. Popis všech použitých knihoven třetích stran a zdůvodnění jejich použití

10. Technická dokumentace netriviálních algoritmů

11. Popis SQL skriptů pro vytvoření potřebné databázové struktury a iniciální naplnění dat/číselníků

12. Popis datových migračních skriptů

13. Dokumentace testovací strategie, testovacích scénářů, výkonnostních testů a vytvořených testovacích, simulačních a mockovacích nástrojů

14. Doporučená systémová (hardwarová) konfigurace pro nasazení do testovacího (Stage) i produkčního (Prod) prostředí, odpovídající požadovaným výkonnostním parametrům

15. Uživatelská příručka pro administraci pomocí administračního UI rozhraní (jestli nějaké existuje)

16. Uživatelské příručka pro běžné uživatele

Očekávané technologie a standardy:

- Gherkin – pro popis testovacích scénářů
- OpenAPI 3.0 / GraphQL Schema – dokumentace API
- Markdown – formátovaný text
- UML diagramy
- README.md – informace pro vývojáře a operations tým, co daný modul dělá, jak ho rychle spustit ve vývojovém prostředí a jak pro něj vyvíjet

# 66 Obecné požadavky na QA

## 66.1.1 Obecné požadavky na testování

Pro potřeby ověření kvality dodaného software (Quality Assurance - QA) je kromě dodání samotné aplikace očekáváno také dodání několika druhů automatických, poloautomatických a manuálních testů, a to v následujících kategoriích:

- **Unit a integrační testy**
  - Cíl: zvýšená šance odhalení regresních závad na úrovni modulu, a to zejména při úpravách a refactoringu
  - Zejména netriviální procesy a algoritmy by měly být pokryté pomocí automatizovaných (unit) testů
  - Bezchybná komunikace mezi jednotlivými komponentami uvnitř aplikace, ale také mezi rozhraním různých systému by měly být pokryté pomocí automatizovaných (integračních) testů
  - Technická implementace testů bude závislá na zvolené technologii pro implementaci jednotlivých modulů
  - Dodavatel musí poskytnout zadavateli veškeré potřebné nástroje a dokumentaci pro automatické spouštění testů jako součásti CI/CD pipeline (build modulu)
  - Procento pokrytí testy a míra automatizace bude specifikována pro každý modul v zadávací dokumentaci modulu

- **Systémové „end-to-end“ testy**
  - Cíl: zvýšená šance odhalení regresních závad na úrovni vzájemné integrace modulů při úpravách a refactoringu
  - Cíl: dokumentace technického postupu jednotlivých uživatelských cest (user journeys) pro vzájemnou validaci korespondence s byznys analýzou modulu
  - Všechny uživatelské cesty (user journeys) a automatizované procesy by měly být pokryté pomocí end-to-end testů (testovací scénáře)
  - Testovací scénáře mohou být automatizované, poloautomatizované nebo zcela manuální, s maximální možnou (rozumnou) mírou plné automatizace co největšího počtu testovacích scénářů, s cílem zapojení testů do pravidelného (nočního) spouštění všech plně automatizovaných scénářů v testovacím prostředí zadavatele
  - Testovací scénáře budou vytvářeny ve formátu Gherkin

- Seznam minimální požadované sady dodaných testovacích scénářů bude specifikován pro každý modul v zadávací dokumentaci modulu

- **Výkonnostní testy**
  - Cíl: zvýšená šance odhalení výkonnostních nedostatků při úpravách a refactoringu
  - Pokrytí všech výkonnostně citlivých částí systému výkonnostními testy
  - Výkonnostní testy musí být automatizované v maximální možné míře
  - Společně s testy poskytne dodavatel zadavateli i detailní technickou dokumentaci popisující fungování, přípravu dat a způsob jejich spouštění
  - Výkonnostní požadavky pro jednotlivé funkce systému, případně míra automatizace testů budou specifikovány v zadávací dokumentaci modulu

#### 66.1.2 Mock, simulační nástroje a nástroje na generování dat

Testy ze všech kategorií, zejména pak testy funkčních požadavků je potřebné umět spouštět jak v rámci plně nasazeného prostředí, tak i proti lokální instanci daného modulu (pro možnost otestování před nasazením do prostředí). Pro tuto potřebu poskytne dodavatel zadavateli také odpovídající nástroje:

- Mock nástroje (nástroje simulující funkci třetích komponent pomocí definovaného rozhraní komponenty)
- Nástroje a generátory pro automatizované vytváření testovacích datových sad
- Simulační nástroje (generátory klientského provozu pomoci API rozhraní modulu, nástroje simulující uživatelský provoz atd.)

Požadovaný seznam nástrojů relevantní k funkcím daného modulu bude upřesněn v zadávací dokumentaci.

#### 66.1.3 Doporučená systémová konfigurace

Jako součást dokumentace poskytne dodavatel zadavateli také popis doporučené systémové (hardwarové) konfigurace pro nasazení do jak do testovacího (Stage) tak i do produkčního (Prod) prostředí, odpovídající požadovaným výkonnostním parametrům systému/modulu.

# Příloha č. 2 – Položkový rozpočet

| Název položky | Nabídková cena v Kč bez DPH | Výše DPH | Nabídková cena v Kč s DPH |
|---|---|---|---|
| Cena za vývoj Software včetně implementace datových rozhraní a integrace, unit testů a instalaci Software | 572 000,00 Kč | 120 120,00 Kč | 692 120,00 Kč |
| Cena za testování, testovací scénáře a nástroje | 417 131,00 Kč | 87 597,51 Kč | 504 728,51 Kč |
| Cena za provedení školení správců | 16 445,00 Kč | 3 453,45 Kč | 19 898,45 Kč |
| Cena za vytvoření dokumentace | 192 764,00 Kč | 40 480,44 Kč | 233 244,44 Kč |
| Cena za licence programového vybavení třetích stran, pokud je jejich dodání nezbytné k implementaci řešení a nejde o licence operačních systémů ani virtualizačních platforem, a jejich instalaci v prostředí Objednatele | 0,00 Kč | 0,00 Kč | 0,00 Kč |
| **Celková nabídková cena** | **1 198 340,00 Kč** | **251 651,40 Kč** | **1 449 991,40 Kč** |

**Příloha č. 3 - Koncepce nabízeného řešení**

# KONCEPCE NABÍZENÉHO PLNĚNÍ UK

MEDIAFACTORY

# ÚVOD

Tento dokument je vytvořen za účelem doložení technického návrhu, projektového plánu a personálního složení týmu pro projekt Univerzity Karlovy - modul osobní údaje.

# KAPITOLA Č. 1 – TECHNICKÝ NÁVRH ŘEŠENÍ

Hrubý návrh a seznam dodaných komponent

Dodáme modul Osobní údaje v podobě dvou aplikací v duchu microservice architektury dle zadání: Backend a Frontend. Součástí bude zdrojový kód včetně testů Dockerfile soubory pro sestavení rootless kontejnerů a docker-compose konfigurace pro lokální vývoj.

Dále dodáme testovací aplikace podle požadavků na dodání testovacích nástrojů v kapitole 6.4. Testovací nástroje budou implementovány v jazyce Java, abychom zbytečně nezvyšovali počet programovacích jazyků v projektu a tím komplexitu projektu.

- Backend

Aplikace bude v souladu s technickými požadavky implementována v jazyce Java a bude postavena nad frameworkem Spring Boot.

Aplikace bude poskytovat privátní REST api frontendovému modulu Osobní údaje. V něm bude poskytovat především data pro překlad frontendové aplikace agregací dat z tabulky WLANG popsané ve specifikaci 5.2.2 Langy, tak aby frontend mohl získat všechny lokalizované popisky co nejmenším počtem dotazů.

Dále bude aplikace poskytovat veřejné REST api rozhraní, které bude konzumováno frontendovou aplikací modulu Osobní údaje a ostatními moduly informačního systému. REST api bude obsahovat endpoint pro vyhledávání osob, endpoint pro získání detailu osoby a endpoint pro úpravu osoby.

Aplikace bude vyhledávat a vypisovat data přímo z view zdrojové databáze.

Aktualizaci dat bude aplikace provádět přímým zavoláním databázové procedury.

Pro práci s databází využijeme JDBC api, pro mapování databázových entit využijeme JPA implementaci Hibernate.

Metriky definované ve specifikaci 5.3.2 bude aplikace sbírat pomocí Spring Boot Actuators a pomocí OpenTelemetry SDK posílat do Prometheus databáze..

- Frontend

Aplikace bude v souladu s technickými požadavky implementována v jazyce TypeScript, nad frameworkem Next.js.

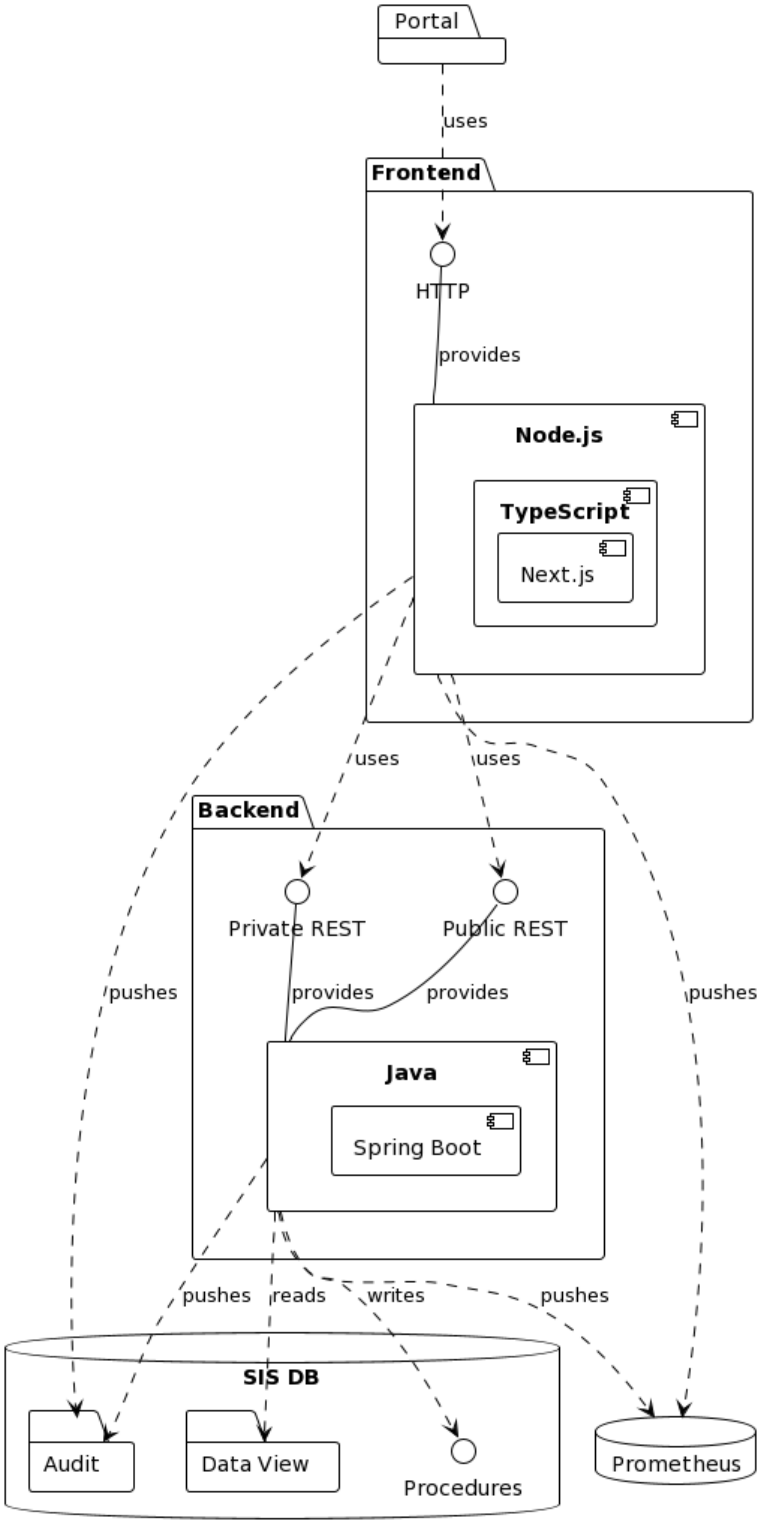
Bude poskytovat uživatelské rozhraní konzumované Portálovou aplikací pomocí standardního http protokolu.

UI bude vykreslováno pomocí komponentového frameworku React a to v co největší míře na serverové straně. Pro stavbu komponent využijeme knihovnu Tailwind UI.

Aplikace bude pro vyhledávání osob a pro zobrazení detailu osoby konzumovat veřejné REST api backend aplikace. Jen slovník přeložených popisků bude aplikace získávat z na míru připraveného privátního REST api backend aplikace.

Metriky bude aplikace sbírat pomocí knihovnen @vercel/otel a @opentelemetry implementující OpenTelemetry standardy.

Pro potřeby odesílání auditních logů bude aplikace napojena na databázi pomocí knihovny Knex.js




- Backend mock

Podle požadavků na testovací nástroje dodáme mock backend aplikace implementovaný v jazyce Java. Půjde o velmi zjednodušenou verzi aplikace, která bude implementovat veřejné i privátní REST rozhraní, ale bude vracet jen data z předem připravených json souborů.

- Generátor REST API požadavků

Vytvoříme Java aplikaci bez grafického rozhraní, která podle parametrů na příkazové řádce bude generovat požadavky na jednotlivé endpointy REST API backendu.

- Generátor HTML API požadavků

Vytvoříme Java aplikaci bez grafického rozhraní, která podle parametrů na příkazové řádce bude generovat požadavky na HTTP API frontendu.

- Generátor randomizovaných osob

Vytvoříme Java aplikaci bez grafického rozhraní, která podle parametrů na příkazové řádce bude generovat SQL importující náhodné osobní údaje do databáze.

LICENCE

S výjimkou pro Tailwind UI pro implementaci použijeme pouze otevřené technologie a knihovny s licencí umožňující volné komerční využití, modifikace i distribuci díla.

- Backend
  - Open JDK - GNU GPL v2
  - Spring Boot - Apache License 2.0
  - HikariCP - Apache License 2.0
  - Hibernate - LGPL 2.1
  - Lombok - MIT License
- Frontend
  - Typescript - Apache License 2.0
  - Next.js - MIT License
  - React - MIT License
  - @vercel/otel - MIT License
  - @opentelemetry/api - Apache License 2.0
  - @opentelemetry/auto-instrumentations-node - Apache License 2.0
  - @opentelemetry/sdk-metrics - Apache License 2.0
  - @opentelemetry/exporter-prometheus - Apache License 2.0
  - @opentelemetry/exporter-trace-otlp-http - Apache License 2.0
  - Knex.js - MIT License
  - Tailwind CSS - MIT License
  - Tailwind UI - Komerční licence v rámci technologického stacku dodavatele (licence umožňuje využití dodavatelem pro koncového uživatele)

ŠKÁLOVÁNÍ

Dodavatel neidentifikoval u backend a frontend aplikace potřebu udržovat aplikační stav, který by bránil některým formám škálování aplikace. Lze tak aplikace umístit za jakýkoliv load balancer a bez problému multiplikovat počet instancí podle aktuální potřeby.

## Kapitola č. 2 – Technologie

BACKEND

- Java (OpenJDK) v. 20, planujeme upgrade na v. 21 LTS po releasu.
- Maven jako nástroj pro správu, řízení a automatizaci buildů.
- Spring Boot v. 3.x.
  - Spring HATEOAS
  - Spring Boot Actuator
- HikariCP jako DB connection pool.
- Vzhledem k tomu, že databázový systém je předem určen (Oracle) a neplánuje se přechod na jiný systém, pro práci s DB dodavatel plánuje využít JDBC případně JPA/Hibernate.
- Oracle Driver
- Lombok za účelem usnadnění vývoje snížením psaní boilerplate kódu.

- Využití instrumentace OpenTelemetry pro sběr tracing záznamů.

FRONTEND

- Typescript 5 - nadstavba nad Javascriptem dodávající compile time typovou kontrolu kódu
- Node.js 18.16 LTS - Javascriptový runtime pro serverové aplikace
- Next.js 13 - Framework pro tvorbu webových aplikací. Řeší routing http požadavků, server side rendering stránek pomocí Reactu a hydrataci frontendových UI komponent.
- React 18 - Framework pro tvorbu UI komponent
- vercel/otel 18 - Knihovna pro sběr metrik v Next.js
- @opentelemetry - Knihovna pro tracing a sběr metrik
- Pro UI Komponenty bude využit TailwindCSS a Tailwind UI

QA TESTOVÁNÍ

- Automatizace e2e FE: Cypress
    - JavaScript
- Performance testy: Jmeter/Lighthouse
- Dokumentace, příručky: Microsoft Word/Excel/Google Docs
- Testovací scénáře ve formě gherkin ve Visual studio s koncovkou .feature
    - Propojení se cypress pomocí knihoven
        - @badeball/cypress-cucumber-preprocessor
        - @badeball/cypress-cucumber-preprocessor/browserify
- V zadávacích dokumentech je zmíněno, že je potřeba splnit bezpečností kritéria zadavatele.
    - Vzhledem k tomu, že v rámci zadání modulu nejsou ve QA specifické požadavky, tak dodavatel neočekává provedení penetračních testů a aplikace bude splňovat dodavatelovi best practices na bezpečnost aplikací.
- V případě psaní GHERKIN testů doporučuje dodavatel rovnou automatizovat testy v cypress. Výhodou je minimální nutnost lidských zdrojů a vyšší přesnost testování
    - Samotné automatizované testy nejsou naceněny v rámci nabídky a jsou tedy out-of-scope
- **Předpoklad QA dodávky za dodavatele je**:
    - **Scope:**
        - manuální testy FE,
        - testovací scénáře ve formě gherkin pro FE ,
        - performance testy
    - **Out of scope**
        - Penetrační testy
        - Automatizované FE e2e testy
        - Testy které nejsou popsané v zadávací dokumentaci
        - Dokumentaci/testovací scénáře které nejsou popsané v zadávací dokumentaci
- Dodavatel očekává testování na posledních verzí následujících zařízení a prohlížečů:
    - Desktop:
        - Windows / Chrome
        - Windows / Edge
        - Windows / Firefox
        - OS X / Safari
        - OS X / Chrome
        - OS X / Firefox
    - Mobile
        - Android - Chrome
        - Android - Samsung internet
        - iOS - Safari

## KAPITOLA Č. 3 – SPRÁVA SYSTÉMU

V souladu se zadávací dokumentaci aplikace dodavatel neočekává existenci složitého systému správy.

Dodavatel předpokládá, že základní nastavení (např. napojení na databázový systém, credentials apod.) se aplikaci budou předávat pomocí proměnných prostředí. Modul by neměl obsahovat žádné globální konfigurační parametry.

Zdrojové kódy aplikace budou uloženy do GitLabu zadavatele, předpokládáme verzování pomocí SemVer, automatizované nasazení na cílové prostředí pomocí CI/CD pipelines. Případné ladění procesu nasazení, změny pipelines apod. se budou řešit v rámci vývoje objednaného modulu.

## KAPITOLA Č. 4 – SLOŽENÍ, ŘÍZENÍ A ORGANIZACE REALIZAČNÍHO TÝMU

**Řízení projektu:**

Řízení projektu bude vedeno metodou waterfall a fixtime fix price. Obecně pak:

- Kick off a kick out meeting budou provedeny osobně.
- V rámci projektu bude vytvořena komunikační matice všech členů projektu spolu s jejich rolemi a kontakty.
- Komunikace bude vedena hlavně na základě statusů/kontrolních schůzek, dokumenty předávány skrze sdílené úložiště a pomocí emailové komunikace. V případě nutnosti bude využita telefonická komunikace.
- Jednotlivé dokumenty budou ukládány na sdílené úložiště zadavatele, v případě, že nebude možné využít úložiště dodavatele, bude vystaven sdílený google drive pro odpovědné osoby zadavatele.
- Status meetingy budou vedeny vzdáleně na 14 denní bázi, kde bude probrán aktuální stav projektu, případně otevřené body nezbytné pro úspěšný postup v projektu.

**Očekávaná součinnost**

Dodavatel očekává zvýšenou součinnost zadavatele na začátku projektu, kdy dojde k vypracování low level designu. Předpokládané aktivity jsou:

- předání všech podkladů ze strany zadavatele nejpozději 3 pracovní dny před kick off projektu
- technická schůzka na vyjasnění otevřených bodů a otázek co nejdříve po kick off projektu
- průběžná mailová komunikace pro vyjasnění otevřených bodů
- ad-hoc schůzky pro operativní vyjasnění otevřených věci

Po dokončení low level designu aplikace očekává dodavatel minimální nutnost součinnosti. Standardně se bude jednat hlavně o projektové statusy a v případě nutnosti dodání náležitostí, které nebyly očekávány zadavatelem ani dodavatelem.

**Tým a zástupnost**

Tým bude složen ze zkušených a stálých zaměstnanců (hlavní lídrem projektu je CTO společnosti), kteří definují rozvoj a směr společnosti. Tím bude zajištěna kvalita projektu a zodpovědný přístup k aplikaci.

**Role jsou:**

- Architekt, garant řešení a backend vývojář - hlavní vývojář backendové vrstvy
- Seniorní full stack vývojář - sekundární vývojář backendové vrstvy, který v případě výpadku může pomoci i s frontendovým modulem
- Seniorní frontendový vývojář - hlavní vývojář frontendového modulu
- Seniorní kodér - vývojář, který s námi v případě nutnosti bude ladit UI aplikace
- QA - tester zajišťující vytvoření E2E a výkonnostních testů a sepsání dokumentace mimo programátorskou dokumentaci
- Projektový manažer - Projektový manažer zajišťující hlavní běh projektu
- Sekundární projektový manažer - pro případ výpadku hlavního projektového manažera

- Záložní full stack vývojář - vývojář evidovaný v seznamu zdrojů pro případ výpadku jednoho z vývojářů.

**Interní řízení projektu**

Interně bude projekt řešen na základě pravidelných denních stand-upů týmu, statusů pro řešení konkrétních a s využitím projektových nástrojů rodiny Atlassian - JIRA, Confluence a další.

**SLOŽENÍ TÝMU.**

V rámci organizace projektového týmu je vyřešena zastupitelnost jednotlivých rolí následovně (nemoc, dovolené apod.).

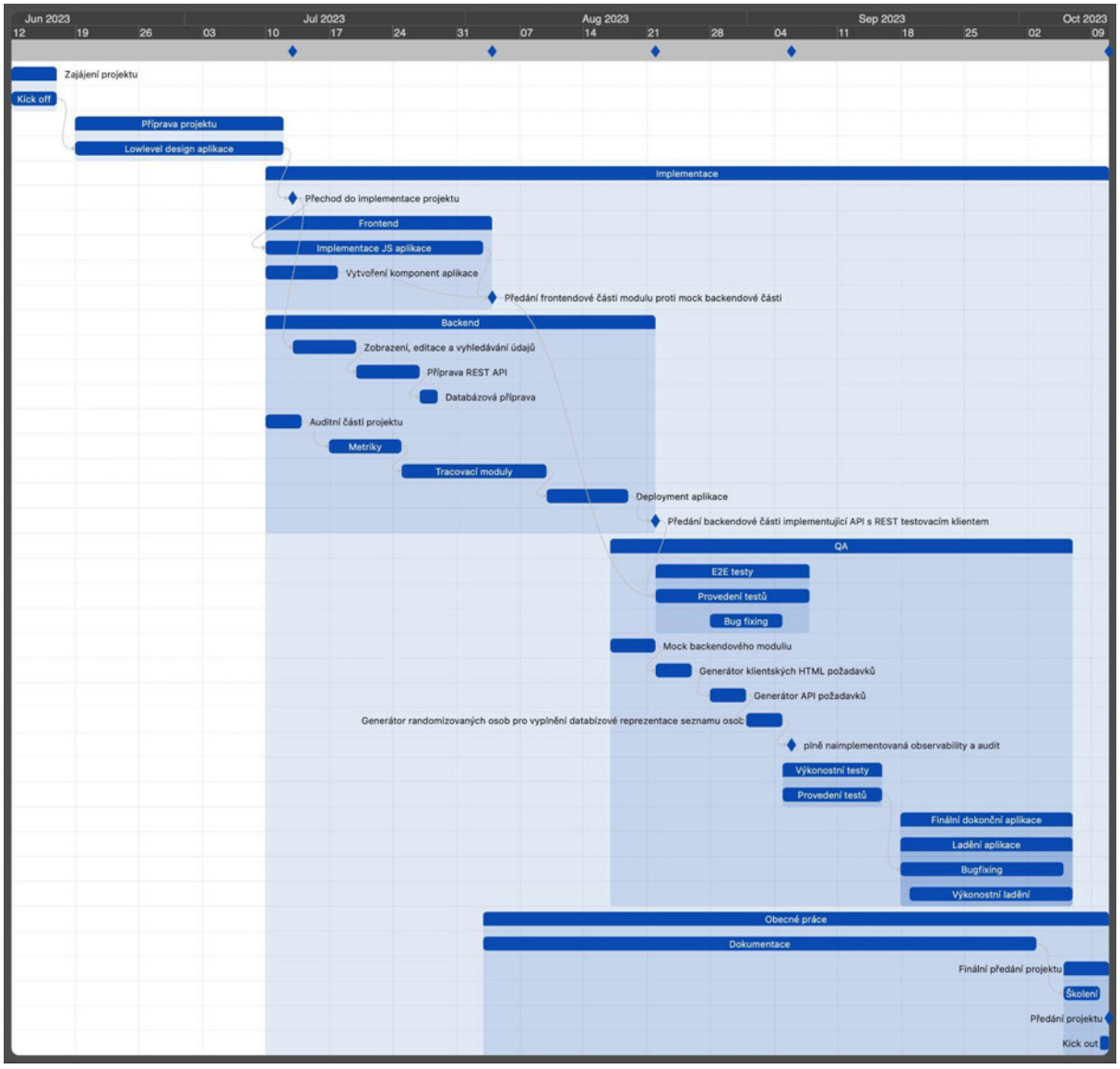
## KAPITOLA Č. 5 – HARMONOGRAM PLNĚNÍ

Dílčí části této kapitoly budou dodány jako samostatné přílohy:

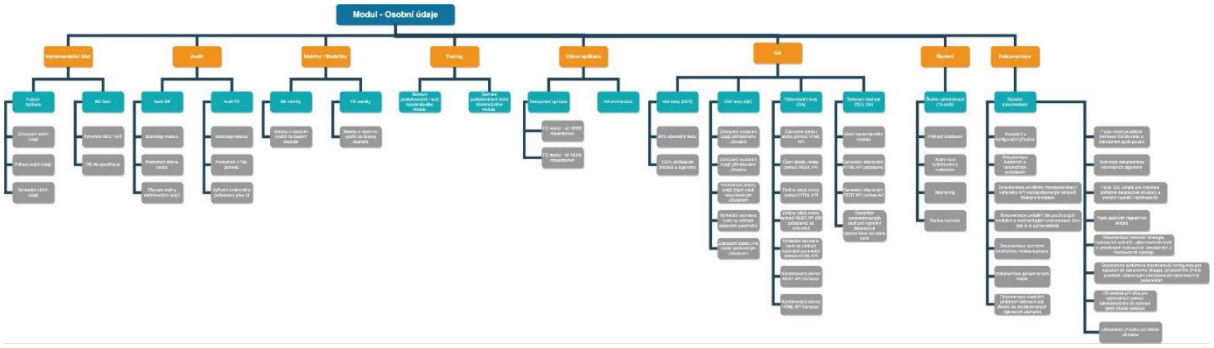
- Příloha č.1 - harmonogram projektu
- Příloha č.2 - WBS projektu

**Náhledové obrázky dodávaných příloh:**

**Příloha č.1 - harmonogram:**




**Příloha č.2 - WBS projektu**


Modul - Osobní údaje

# Příloha č. 4 - Složení týmu Objednatele a Dodavatele

(tj. projektový manažer)

e-mail: ……

telefon: ……

Další složení týmu Dodavatele je uvedeno v příloze č. 3 Smlouvy.

# Příloha č. 5 – Bezpečnostní požadavky

**Článek 1**
**Úvod**

1. Tento dokument popisuje bezpečnostní požadavky kladené na Dodavatele v rámci realizace veřejné zakázky „RUK – ÚVT - Dodávka modulu Osobní údaje Studijního informačního systému na Univerzitě Karlově“, zejména pro naplnění požadavků vyplývajících pro Dodavatele ze zákona č. 181/2014 Sb., o kybernetické bezpečnosti a o změně souvisejících zákonů (zákon o kybernetické bezpečnosti), ve znění pozdějších předpisů (dále jen „ZoKB“), a vyhlášky č. 82/2018 Sb., o bezpečnostních opatřeních, kybernetických bezpečnostních incidentech, reaktivních opatřeních, náležitostech podání v oblasti kybernetické bezpečnosti a likvidaci dat (vyhláška o kybernetické bezpečnosti (dále jen „VyKB“).
2. Dodavatel je povinen plnit relevantní povinnosti v rozsahu a způsobem, aby byl naplněn účel právní úpravy oblasti bezpečnostních opatření kybernetické bezpečnosti ve vztahu k povinnostem, které tato právní úprava stanovuje Objednateli jakožto povinné osobě dle ZoKB. V takovém případě je Objednatel oprávněn požadovat od Dodavatele přiměřenou součinnost i nad rámec povinností stanovených v této příloze, avšak vždy pouze za účelem zajištění plnění povinnosti Dodavatele z oblasti kybernetické bezpečnosti ve smyslu shora uvedeného.

**Článek 2**
**Bezpečnostní požadavky**

**2.1. Účel**
1. Tento dokument stanoví způsoby a úrovně realizace bezpečnostních opatření pro Dodavatele a určuje vzájemný vztah odpovědnosti za zavedení a kontrolu bezpečnostních opatření mezi Objednatelem a Dodavatelem. Požadavky na Dodavatele jsou definovány dle platné právní úpravy, především pak dle ZoKB, VyKB.
2. Smluvní strany se dohodly, že pokud to bude potřebné ke splnění požadavků ZoKB, VyKB, či souvisejících právních předpisů z oblasti bezpečnosti informací, uzavřou bez zbytečného odkladu po výzvě kterékoli smluvní strany písemný dodatek Smlouvy zohledňující takové požadavky.

**2.2. Obecné požadavky**
1. Dodavatel se při poskytování plnění pro Objednatele zavazuje plnit následující povinnosti:
    a. postupovat v souladu s účinnými právními předpisy, zejména pak požadavky vyplývajícími pro Dodavatele, jakožto budoucího významného dodavatele provozovatele základní služby, ze ZoKB, VyKB a reflektovat případné novely dotčených právních předpisů či novou právní úpravu;
    b. dodržovat příslušná ustanovení bezpečnostních politik, metodik a postupů předaných Dodavateli Objednatelem, resp. platné řídící dokumentace Objednatele či její části anebo platné řídící dokumentace, k jejímuž dodržování se Objednatel zavázal, pokud byl Dodavatel s takovými dokumenty nebo jejich částmi seznámen, a to bez ohledu na způsob, jakým byl s takovou dokumentací Objednatele seznámen (např. školením, protokolárním předáním příslušné

dokumentace poskytovateli, elektronickým předáním prostřednictvím e-mailu, zřízením přístupu Poskytovateli na sdílené úložiště aj.);

c. rozvíjet bezpečnostní povědomí svých zaměstnanců a příp. dalších osob, které se podílejí na plnění Smlouvy a průběžně je seznamovat s prováděnými nebo plánovanými změnami. Zaměstnanci a další osoby na straně Dodavatele podílející se na plnění Smlouvy musí být prokazatelně seznámeni s platnými předpisy a bezpečnostními požadavky Objednatele, a to ještě před zahájením jakékoli činnosti ze strany těchto osob pro Objednatele v souvislosti s plněním této Smlouvy;
d. zaznamenávat podstatné okolnosti související s poskytovaným předmětem plnění dle Smlouvy (technické záznamy, organizační záznamy o školení, pověření apod.) a informovat o nich Objednatele;
e. přidělovat svým jednotlivým pracovníkům zaměstnancům oprávnění k výkonu činností a přísně při tom dodržovat bezpečnostní zásadu tzv. „potřeba vědět“ (need-to-know principle), tedy zejména dbát o to, aby byla minimalizována rizika nežádoucího přístupu k aktivům Objednatele;
f. průběžně dokumentovat, kontrolovat a vyhodnocovat oprávněnost přístupu, jak fyzického, tak i logického, u všech osob na straně Dodavatele, které přistupují k předmětu plnění dle této Smlouvy;
g. průběžně detekovat technické zranitelnosti a konfigurační nesoulady předmětu plnění Smlouvy a o zjištěných skutečnostech bez zbytečného odkladu informovat Objednatele. Detekované technické zranitelnosti musí být vyhodnoceny s ohledem na související riziko a musí podle povahy předmětu plnění dojít k nápravným opatřením ze strany Dodavatele. Nápravná opatření musí být schválena Objednatelem;
h. realizovat bezpečnostní opatření pro ochranu dat souvisejících s plněním předmětu Smlouvy.

**2.3. Bezpečnost informací**

1. Dodavatel je povinen zajistit utajení získaných důvěrných informací objednatele způsobem obvyklým pro utajování takových informací, není-li výslovně sjednáno jinak. Tato povinnost platí bez ohledu na ukončení účinnosti této smlouvy. Dodavatel je povinen zajistit utajení důvěrných informací i u svých zaměstnanců, zástupců, jakož i jiných spolupracujících třetích stran, pokud jim takové informace byly poskytnuty.
2. Právo užívat, poskytovat a zpřístupnit důvěrné informace má Dodavatel pouze v rozsahu a za podmínek nezbytných pro řádné plnění práv a povinností vyplývajících z této Smlouvy.
3. Za důvěrné informace se bez ohledu na formu jejich zachycení považují veškeré informace, které nebyly Objednatelem označeny jako veřejné a které se týkají této Smlouvy a jejího plnění (zejména informace o právech a povinnostech smluvních stran jakož i informace o cenách), které se týkají Objednatele, jeho smluvních partnerů, pacientů, obchodní tajemství, anebo informace pro nakládání, s nimiž je stanoven právními předpisy zvláštní režim utajení. Dále se považují za důvěrné informace takové informace, které jsou jako důvěrné výslovně Objednatelem označeny.
4. Za důvěrné informace se v žádném případě nepovažují informace, které se staly veřejně přístupnými, pokud se tak nestalo porušením povinnosti jejich ochrany, dále informace získané na základě postupu nezávislého na této Smlouvě a informace poskytnuté třetí osobou, která takové informace nezískala porušením povinnosti jejich ochrany.

5. Dodavatel je povinen zajistit bezpečnost informací z pohledu dostupnosti. Informace se z pohledu dostupnosti považují za bezpečné, jestliže jsou dostupné autorizovaným uživatelům v době, kdy jsou potřeba.
6. Dodavatel je povinen zajistit bezpečnost informací z pohledu integrity. Informace se z pohledu integrity považují za bezpečné, jestliže je zaručena jejich správnost, bezchybnost a jsou vyloučeny jejich neautorizované změny.

**2.4. Oprávnění užívat data**

1. Dodavatel je při poskytování plnění pro Objednatele oprávněn užívat informace předaná Dodavateli Objednatelem za účelem plnění předmětu Smlouvy, avšak vždy pouze v rozsahu nezbytném ke splnění předmětu Smlouvy.
2. Dodavatel se při poskytování plnění pro Objednatele zavazuje nakládat s informacemi pouze v souladu se Smlouvou a příslušnými právními předpisy, zejména ZoKB, VyKB a dalšími souvisejícími právními předpisy.

**2.5. Řetězení a řízení dodavatelů**

1. Dodavatel nezapojí do poskytování plnění dle této Smlouvy žádného dalšího poddodavatele bez předchozího konkrétního nebo obecného písemného povolení Objednatele.
2. Dodavatel se zavazuje, že se bude řídit požadavky Objednatele na řízení bezpečnosti informací a poskytne Objednateli veškerou nezbytnou součinnost v otázkách řízení bezpečnosti informací a pokud využívá při poskytování plnění poddodavatele, zajistí, že bude Objednateli poskytnuta veškerá nezbytná součinnost v otázkách řízení bezpečnosti informací také od těchto poddodavatelů.
3. Dodavatel je povinen předat Objednateli kontaktní údaje všech osob dodávajících systémovou a technickou podporu pro řešení.
4. Pokud Dodavatel využívá při poskytování plnění poddodavatele, zavazuje se, že budou dodržovat bezpečnostní požadavky vč. požadavků na ochranu osobních údajů vyplývající z této Smlouvy.
5. Dodavatel se zavazuje bezodkladně doložit Objednateli na základě jeho výzvy smluvní dokumenty se svými poddodavateli, ze kterých bude vyplývat závazek poddodavatele poskytovat plnění v souladu s bezpečnostními požadavky vyplývajícími z této Smlouvy.
6. Dodavatel odpovídá za to, že jeho poddodavatelé nebudou jednat v rozporu s bezpečnostními požadavky vyplývajícími z této Smlouvy.

**2.6. Řízení změn**

1. Dodavatel se zavazuje provést hodnocení dopadů Objednatelem navrhovaných změn na termíny a cenu předmětu plnění Smlouvy. Pokud by však takovéto hodnocení vyžadovalo dodatečné náklady anebo by nepříznivě ovlivnilo pracovní vytížení zaměstnanců nebo využití jiných prostředků určených k provádění předmětu plnění, Dodavatel tuto skutečnost oznámí Objednateli a hodnocení provede pouze na základě písemného pověření Objednatelem.V takovém případě bude hodnocení hrazeno podle stráveného času v sazbách platných v době hodnocení.
2. Dodavatel u významných změn dokumentuje jejich řízení, provádí analýzu rizik, přijímá opatření za účelem snížení všech nepříznivých dopadů spojených s významnými změnami, aktualizuje bezpečnostní politiku a bezpečnostní dokumentaci, zajistí testování informačního systému a zajistí možnost navrácení do původního stavu.
3. Dodavatel má povinnost informovat Objednatele o výsledcích řízení změn, které mají dopady na plnění předmětu Smlouvy ze strany Dodavatele.

4. Dodavatel má povinnost přijmout účinná opatření ke snížení nepříznivých dopadů v souladu s výsledky řízení změn
5. Dodavatel se zavazuje poskytnout Objednateli veškerou nezbytnou součinnost při analýze souvisejících rizik, přijímání opatření za účelem snížení všech nepříznivých dopadů spojených se změnami, aktualizaci bezpečnostní dokumentace, souvisejícím testováním a zajištění možnosti navrácení do původního stavu.
6. V případě realizace penetračního testování nebo testování zranitelnosti řešení poskytne Dodavatel Objednateli veškerou potřebnou součinnost.

**2.7. Akvizice, vývoj a údržba**

1. Dodavatel se zavazuje v rozsahu plnění na své straně zajistit bezpečnou implementaci, inovaci, aktualizaci a testování technologií, které jsou předmětem plnění.
2. Dodavatel se zavazuje předat Objednateli dokumentaci předmětu plnění obsahující zejména:
    a. dokumentaci všech bezpečnostních nastavení, funkcí a mechanismů;
    b. dokumentaci obsahující popis autorizačního konceptu a oprávnění;
    c. dokumentaci obsahující instalační a konfigurační postupy.
3. V případě, že předmět plnění zahrnuje vývoj software, zavazuje se Dodavatel:
    a. dodržovat a implementovat nejlepší praktiky pro bezpečný vývoj softwaru definované na základě smluvního vztahu, nebo dodaného Objednatelem;
    b. na vyžádání umožnit Objednateli provedení auditu prováděného nebo provedeného plnění, předložit Objednateli vyvíjený kód SW;
    c. poskytnout Objednateli v termínech stanovených Objednatelem, resp. bez zbytečného odkladu požadovanou součinnost na provedení bezpečnostního testování v průběhu vývoje SW či kdykoli po jeho předání;
    d. zajistit, že plnění bude obsahovat jen ty součásti, které jsou objektivně potřebné pro řádné provozování softwaru a/nebo které jsou specifikovány výslovně ve Smlouvě (např. že SW nebude obsahovat žádné nepotřebné komponenty, programové vzorky apod.);
    e. pokud je součástí plnění i instalace operačního systému případně SW třetích stran, zajistit v průběhu jeho instalace, že budou použity předepsané verze těchto produktů kompatibilní a funkční v IT prostředí Objednatele;
    f. zajistit bezpečnost testovacího prostředí u Dodavatele a ochranu poskytnutých testovacích dat Objednatelem;
    g. zajistit, že do produkčního prostředí Objednatele bude dodán jen předmětem Smlouvy specifikovaná kompilovaný, resp. spustitelný kód a další nezbytná data pro provozování předmětu plnění;
    h. instalovat SW pouze na základě Objednatelem předem schválených migračních postupů;
    i. předat zdrojový kód Objednateli bezpečnou formou zajišťující jeho integritu;
    j. zajistit řízení verzí zdrojového kódu;
    k. zajistit zálohování zdrojového kódu a jeho uložení mimo produkční prostředí;
    l. nevyvíjet, nekompilovat a nešířit v IT prostředí Objednatele programový kód, který má za cíl nelegální ovládnutí, narušení dostupnosti, důvěrnosti nebo integrity nebo neautorizované či nelegální získání dat a informací.

**2.8. Zvládání kybernetických bezpečnostních událostí a incidentů**

1. Dodavatel se při poskytování plnění pro Objednatele zavazuje v rozsahu poskytovaných služeb dle Smlouvy, že:

a. stanoví činnosti, role a jejich odpovědnosti a pravomoci vedoucí k rychlému a účinnému zvládání kybernetických bezpečnostních událostí a incidentů, podle takto stanovených a popsaných pravidel bude postupovat, a bude hlásit všechny bezpečnostní události a incidenty neprodleně po jejich detekci Objednateli prostřednictvím ohlašovacích kanálů na osobu odpovědnou za kybernetickou bezpečnost, v případech, kdy situace nestrpí odklad telefonicky.
b. nastavená pravidla pro zvládání bezpečnostních incidentů budou respektovat požadavek na legalitu zajištění stop, tj. jejich původ a oprávněnost jejich získaní musí být v souladu s platnými zákony a standardy tak, aby bylo možné jejich následné využití v rámci forenzní analýzy a eventuální použití jako důkazní materiál;
c. navrhne řešení tak, aby byl systém detekce a zvládání bezpečnostních událostí a incidentů začleněn do procesů a systémů a realizuje opatření pro zvýšení odolnosti informačního a komunikačního systému vůči kybernetickým bezpečnostním incidentům a omezením dostupnosti;
d. provede analýzu příčin bezpečnostního incidentu a navrhne opatření s cílem zamezit jeho opakování v případě, že Dodavatel bezpečnostní incident zapříčinil nebo se na jeho vzniku podílel.

**2.9. Informační povinnost a povinnosti při výměně informací**

1. Dodavatel se během poskytování plnění pro Objednatele zavazuje Objednatele informovat o:
   a. způsobu řízení rizik, zbytkových rizicích souvisejících s plněním Smlouvy a bez zbytečného odkladu také o změnách ve způsobu řízení rizik;
   b. změně vlastnictví zásadních aktiv, využívaných Dodavatelem k plnění Smlouvy, a změně oprávnění nakládat s těmito aktivy, a to nejpozději do tří pracovních dnů po uskutečnění této změny.
2. Dodavatel se během poskytování plnění pro Objednatele zavazuje dostatečně zabezpečit veškerý přenos dat a informací z pohledu bezpečnostních požadavků na jejich důvěrnost, integritu a dostupnost.

**2.10. Řízení kontinuity činností**

1. Objednatel má oprávnění zapojit Dodavatele do řízení kontinuity činností, a to zejména oprávnění k zahrnutí Dodavatele do plánu kontinuity činností, který souvisí s informačním systémem a souvisejících služeb a/nebo zahrnutí Dodavatele do Plánu obnovy (DRP) Objednatele. Objednatel má oprávnění požadovat po Dodavateli zpracování Plánů obnovy (DRP) disaster recovery postupů.
2. Objednatel má povinnost informovat Dodavatele o způsobu zapojení do řízení kontinuity činností.

**2.11. Bezpečnost lidských zdrojů**

1. Dodavatel zajistí poučení všech svých zaměstnanců podílejících se na dodávce o bezpečnostních pravidlech uvedených v těchto Bezpečnostních požadavcích, jež se musí v průběhu dodávky dodržovat a zajistí jejich dodržování nasazením kontrolních a vynucovacích mechanismů. Rozsah poučení podléhá schválení Objednatele osobou určenou za kybernetickou bezpečnost.
2. Dodavatel se zaváže zajistit dostatečnou míru zastupitelnosti pro technické aspekty řešení (zajištění kontinuity dodávky, zastupitelnost zaměstnanců).

3. Dodavatel je povinen vést písemnou evidenci uskutečněných poučení v rozsahu předmět poučení, datum a seznam poučených osob. Dodavatel se zavazuje předložit tuto evidenci objednateli bezodkladně poté, co k tomu bude Objednatelem vyzván.

**2.12. Bezpečnost komunikace**

1. Zaměstnanci Dodavatele, kteří mají přidělen přístup do interní sítě Univerzity Karlovy (většinou na konkrétní server), odpovídají za své činnosti prováděné v rámci interní sítě Univerzity Karlovy. Z důvodu zajištění bezpečnosti zaměstnanci Dodavatele nesmí zejména:
    a. zneužívat síťové prostředky pro osobní účely a zatěžovat kapacitu sítě;
    b. šířit škodlivý kód;
    c. připojovat do sítě jiná, než schválená zařízení Dodavatelem (včetně USB zařízení, soukromých mobilních zařízení, IoT zařízení apod.);
    d. využívat nástroje sloužící k maskování identity;
    e. provádět bezdůvodné skenování portů;
    f. provádět jakoukoliv formou monitorování počítačoví sítě, které může vést k zachycení informací/dat, pokud není předmětem plnění smlouvy;
    g. obcházet autentizaci uživatele nebo obcházet zabezpečení jakéhokoliv počítače, počítačové sítě;
    h. provádět jakékoliv nepracovní aktivity vedoucí k omezování nebo odepírání služeb jiným uživatelům;
    i. užívat jakékoliv programy, skripty nebo příkazy, nebo zasílat zprávy v jakékoliv formě s úmyslem omezit nebo znemožnit poskytování služeb nebo terminálových relací lokálně nebo přes počítačovou síť, internet nebo intranet;
    j. využívat bezpečnostních mezer nebo vytvářet útoky na komunikaci v počítačových sítích (např. přístup k datům, jichž není zaměstnanec zamýšleným příjemcem, přihlašování na server nebo účet zaměstnancem, který není k tomuto přístupu výslovně oprávněn, s výjimkou případů, kdy tyto aktivity jsou součástí řádných pracovních úkolů);
    k. předávat informace o konfiguraci a topologii sítě cizím osobám; tyto informace je oprávněn předat pouze odpovědný zaměstnanec Univerzity Karlovy, pokud jsou takové informace nutné z hlediska přípravy či plnění smluvního vztahu.
2. Při práci na pracovní stanici, mobilním zařízení připojeného do sítě nebo do informačního systému Univerzity Karlovy musí Dodavatel dodržovat tyto základní zásady:
    a. umožnit přístup jen poučenému zaměstnanci Dodavatele;
    b. chránit ICT prostředky Univerzity Karlovy;
    c. po ukončení práce provést neprodleně odhlášení tak, aby se zamezilo zneužití jeho přístupových práv.

**2.13. Řízení přístupu**

1. Dodavatel bere na vědomí, že přístup k datům, informacím či zařízením souvisejícím s předmětem Smlouvy je možné povolit pouze fyzické identitě zaměstnance Dodavatele poučené o těchto Bezpečnostních požadavcích, a to na základě požadavku Dodavatele na přístup.
2. Dodavatel bere na vědomí, že přidělení oprávnění zaměstnanci Dodavatele musí být řízeno zásadou tzv. „potřeba vědět“ (need-to-know principle) a není nárokové.
3. Dodavatel se zavazuje, že udělený přístup nesmí být sdílen více zaměstnanci Dodavatele.

4. Dodavatel se zavazuje, že nebude instalovat a používat žádné nástroje, které nebyly předem písemně odsouhlaseny osobou odpovědnou za kybernetickou bezpečnost na straně Objednatele a jejichž užívání by mohlo ohrozit kybernetickou bezpečnost.
5. Dodavatel se zavazuje, že nebude vyvíjet, kompilovat a šířit v jakékoliv části technologického nebo komunikačního systému programový kód, který má za cíl nelegální ovládnutí, narušení, nebo diskreditaci technologického nebo komunikačního systému nebo nelegální získání dat a informací. Dodavatel bere na vědomí, že přístup do interní sítě a/nebo k technologickým a komunikačním systémům bude realizován výhradně s využitím zařízení Objednatele.
6. Dodavatel se zavazuje zajistit, aby osoby podílející se na poskytování plnění Objednateli, kteří přistupují do interní sítě a/nebo technologického nebo komunikačního systému chránili autentizační prostředky a údaje k systémům Objednatele. Dodavatel bere na vědomí, že v případě neúspěšných pokusů o autentizaci uživatele může být příslušný účet zablokován a řešen jako bezpečnostní incident ve smyslu příslušné řídící dokumentace a mohou být uplatněny příslušné postupy zvládání bezpečnostního incidentu.
7. Dodavatel se zavazuje, že nebude instalovat a používat zejména nástroje typu Keylogger, Sniffer, Analyzátor zranitelností a Port Scanner, Backdoor, rootkit a trojský kůň nebo jinou podobu malware.
8. Dodavatel bere na vědomí, že postup zvládání bezpečnostního incidentu či skutečnost vzniklá v důsledku porušení bezpečnostních požadavků nebude posuzována jako okolnost vylučující odpovědnost Dodavatele za prodlení s řádným a včasným plněním předmětu Smlouvy a nebude důvodem k jakékoli náhradě případné újmy Dodavateli či jiné osobě ze strany Objednatele.
9. Na základě smluvního vztahu může být konkrétním zaměstnancům Dodavatele umožněn přístup k předmětným ICT prostředkům pomocí vzdáleného přístupu přes VPN. Pracovní stanice, přenosná zařízení zaměstnanců Dodavatele přistupující k ICT prostředkům Objednatele musí mít instalován výrobcem podporovaný aktualizovaný operační systém a systém pro ochranu před škodlivým kódem (antivirový program) s nejnovější verzí virové databáze.
10. Lokální přístup Dodavatele do provozního prostředí může být povolen pouze v odůvodněných případech. Tento přístup musí probíhat ve zvláštním režimu dohledu ze strany Univerzity Karlovy.

**2.14. Ochrana před škodlivým kódem**

1. Dodavatel se zavazuje, že zajistí maximální ochranu před zavlečením škodlivého SW do interní sítě Objednatele. Zaměstnanci, resp. subdodavatelé Dodavatele mají zakázáno používat soukromou výpočetní techniku pro připojování do interní sítě Objednatele.

**2.15. Monitorování činností**

1. Dodavatel bere na vědomí, že veškeré aktivity Dodavatele a jeho plnění realizované v rámci plnění předmětu Smlouvy nebo s ním úzce související budou Objednatelem průběžně a pravidelně monitorovány a vyhodnocovány s ohledem na obsah Smlouvy.

**2.16. Kontrola souladu s požadavky bezpečnosti**

1. Dodavatel se zavazuje umožnit Objednateli nebo jím pověřené třetí osobě provést audit plnění povinností Dodavatele dle této Smlouvy, zejména způsobu plnění bezpečnostních opatření, způsobu řízení dodavatelů, nakládání s daty, způsobu identifikace a hlášení kybernetických bezpečnostních incidentů. Objednatel se zavazuje vyrozumět Dodavatele o termínu a případně osobě pověřené provedením auditu alespoň 3 pracovní dny předem.

Dodavatel se zavazuje umožnit provedení auditu ve všech prostorách, v nichž dochází k plnění předmětu této Smlouvy. Náklady na uskutečnění auditu ponese každá smluvní strana zvlášť; Dodavatel nemá právo na poskytnutí úhrady nákladů či jakéhokoliv jiného plnění v souvislosti s auditem. V případě, že výsledkem auditu budou zjištění o pochybeních na straně Dodavatel, zavazuje se Dodavatel bezodkladně po výzvě Objednatele veškeré takové nedostatky na své náklady odstranit. Dodavatel je povinen zajistit umožnění auditu za stejných podmínek i u svých poddodavatelů.

2. Dodavatel je povinen pravidelně provádět také vlastní hodnocení rizik a kontrolu zavedených bezpečnostních opatření. Tato kontrola probíhá v pravidelných intervalech stanovených Objednatelem, na žádost Objednatele nebo v případě vzniku kybernetického bezpečnostního incidentu v rámci poskytované služby nebo v případě, že se vznik bezpečnostního incidentu jeví jako pravděpodobný. O výsledku kontroly podá Dodavatel Objednateli bez zbytečného odkladu písemnou kontrolní zprávu.

**2.17. Porušení Bezpečnostních požadavků**

1. Dodavatel odpovídá za to, že jeho zaměstnanci a/nebo poddodavatelé nebudou jednat v rozporu s bezpečnostními požadavky vyplývajícími z této Smlouvy. V případě, že dojde k nedodržení těchto požadavků ze strany zaměstnance a/nebo poddodavatele Dodavatele, považuje se každé takové nedodržení požadavků za porušení povinnosti Dodavatele dle této Smlouvy.