TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Vizualizace utažených spojů na repasním pracovišti za automatickou zástavbou podvozku

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie

Autor práce: **Vratislav Dutka**
Vedoucí práce: Ing. Igor Kopetschke
Konzultant: Ing. Jan Žmolík

Liberec 2020

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

Visualization of tightened joints in the refurbishment workplace behind automatic chassis installation

Bachelor thesis

Study programme: B2646 – Information technologies
Study branch: 1802R007 – Information technologies

Author: **Vratislav Dutka**
Supervisor: Ing. Igor Kopetschke
Consultant: Ing. Jan Žmolík

Liberec 2020

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Zadání bakalářské práce

Vizualizace utažených spojů na repasním pracovišti za automatickou zástavbou podvozku

Jméno a příjmení: **Vratislav Dutka**
Osobní číslo: M17000074
Studijní program: B2646 Informační technologie
Studijní obor: Informační technologie
Zadávající katedra: Ústav nových technologií a aplikované informatiky
Akademický rok: **2019/2020**

Zásady pro vypracování:

1. Seznamte se s problematikou automatického utahování šroubových spojů v prostředí Škoda Auto a.s.
2. Navrhněte vhodné řešení pro vizualizaci stavu automaticky utažených spojů na podvozku.
3. Implementujte řešení v prostředí .NET CORE za využití technologií SignalR, WPF (MVVM Pattern).
4. Ověřte správnost implementovaného řešení nasazením do produkčního režimu v prostředí Škoda Auto a.s.

Rozsah grafických prací: dle potřeby
Rozsah pracovní zprávy: 30-40 stran
Forma zpracování práce: tištěná/elektronická
Jazyk práce: Čeština

Seznam odborné literatury:

[1]BORY, Pavel. C# bez předchozích znalostí. Brno: Computer Press, 2016. ISBN 9788025146866.
[2]MICHAELIS, Mark. Essential C# 7.0. Boston, MA: Addison-Wesley, [2018]. ISBN 978-1509303588.
[3]ASP.NET documentation SignalR [online]. Microsoft, 2019 [cit. 2019-09-06]. Dostupné z:https://docs.microsoft.com/en-us/aspnet/signalr/
[4]Getting started with Model-View-ViewModel (MVVM) pattern using Windows Presentation Framework (WPF)[online]. IntelliTect, 2019 [cit. 2019-09-06]. Dostupné z: https://intellitect.com/getting-started-model-view-viewmodel-mvvm-pattern-using-windows-pre
[5]Introduction to ASP.NET Core SignalR [online]. Microsoft, 2018 [cit. 2019-09-06]. Dostupné z:https://docs.microsoft.com/en-us/aspnet/core/signalr/introduction?view=aspnetcore-2.2

Vedoucí práce: Ing. Igor Kopetschke
Ústav nových technologií a aplikované informatiky

Datum zadání práce: 9. října 2019
Předpokládaný termín odevzdání: 18. května 2020

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

Ing. Josef Novák, Ph.D.
vedoucí ústavu

V Liberci dne 17. října 2019

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

1. 6. 2020 Vratislav Dutka

Poděkování

Chtěl bych poděkovat panu Ing. Igoru Kopetschkemu za odborné vedení práce. Dále bych chtěl poděkovat Ing. Janu Žmolíkovi za cenné rady, které mi pomohly tuto práci vytvořit. Také bych chtěl poděkovat firmám CMS s. r. o. a Škoda Auto a. s., které mi umožnily práci vytvořit.

Vizualizace utažených spojů na repasním pracovišti za automatickou zástavbou podvozku

Abstrakt

Na montážní hale M13 v závodě Škoda Auto v Mladé Boleslavi jsou šroubové spoje na podvozku vozu utahovány automatickou stanicí. Na následujícím pracovišti je operátory montáže prováděna kontrola a případná repase těchto spojů. Pro snadnější navigaci a optimalizaci celého procesu manuální kontroly operátorem bude na toto pracoviště dodána vizualizace utažených spojů. Na vizualizačních tabulích budou zobrazeny stavy všech utažení pro aktuálně montovaný vůz. Operátor dle této vizualizace následně provede manuální repasi spojů.

Klíčová slova: C#, .NET, ASP .NET CORE, vizualizace, repase, utahované spoje

Visualization of tightened joints in the refurbishment workplace behind automatic chassis installation

Abstract

At the M13 assembly hall at the Škoda Auto company in Mladá Boleslav, the screw connections on the chassis of the car are tightened by automatic station. At the following workplace, assembly operators performs inspection and possible repairs of these connections. For easier navigation and optimization of the entire process of manual inspection by the operator, the visualization of tightened connections will be delivered to this workstation. The visualization tables will show the status of all tightenings for the currently assembled car. The operator then performs manual overhaul of the connections based on this visualization.

Keywords: C#, .NET, ASP .NET CORE, visualization, repase, tightening joints

Obsah

Seznam obrázků

Seznam zkratek

VUS	Vizualizace utahovaných spojů za podvozkovou zástavbou (název systému)
KNR	Kennungsnummer (výrobní identifikační číslo)
VIN	Vehicle identification number (mezinárodní identifikační číslo motorového vozidla)
PC	Počítač
SQL	Structured query language (standardizovaný strukturovaný dotazovací jazyk, který se používá pro práci s daty v relačních databázích)
API	Application programming interface (rozhraní pro programování aplikací)
v2	Druhá verze
OSI model	Open systems interconnection model (model standardizace počítačových sítí)
TCP	Transmission control protocol (jeden ze základních protokolů sady protokolů Internetu, konkrétně představuje transportní vrstvu)
HTTP	Hypertext transfer protocol (internetový protokol určený pro komunikaci s WWW servery)
HTTPS	Hypertext transfer protocol secure (zabezpečený HTTP)
WPF	Windows presentation foundation (knihovna tříd pro tvorbu grafického rozhraní, která je součástí .NET frameworku)
MVVM Pattern	Model–view–viewmodel pattern (softwarový architektonický vzor pro vývoj grafického rozhraní implementovaný ve WPF)
PLC	Programmable logic controller (programovatelný logický automat)
px	Pixel (obrazový bod)
IO	In Ordnung (v pořádku)
NIO	Nicht In Ordnung (ne v pořádku)

Úvod

Vizualizace utažených spojů na repasním pracovišti za automatickou zástavbou podvozku ve společnosti Škoda Auto na montážní hale M13 v Mladé Boleslavi vizualizuje správnost utažení jednotlivých spojů, které jsou prováděné automatickými utahovacími zařízeními každého podvozku. Tento systém funguje z důvodu kontroly pro případné provedení repase jednotlivých spojů podvozku obsluhou.

Předpokládaný projekt je novou verzí vizualizace na tomto pracovišti. Stará verze nesplňuje nové bezpečnostní předpisy a standardy pro aplikace ve společnosti Škoda Auto. V původním systému vizualizace je problém s konfigurační částí – pomalé vytváření nových typů podvozků. V dnešním rychlém výrobním tempu, kdy je automobil na pracovišti zhruba minutu, není vizualizační část přehledná. Obě části jsou aplikacemi a nelze je jednoduše přenášet mezi zařízeními. Především konfigurační část není připravena na nástup mobilních zařízení (tabletů, mobilních telefonů, atd.).

Všechna tato omezení bylo potřeba vzít v úvahu při vytváření nové verze systému, který byl implementován v rámci této práce.

1 Cíle práce

Hlavním cílem mé práce je vytvořit novou verzi aplikace, která ze zdrojových systémů převezme potřebná data, upraví je a zobrazí v přehledné formě. Požadavkem je skvělá přehlednost celého systému a vzhledová podobnost předchozí verze z důvodu rychlé orientace pracovníku na repasním pracovišti při přechodu mezi verzemi. Také je potřeba, aby celý proces fungoval v reálném čase. Aplikace musí být rozdělena na dvě základní části: vizualizační část (část, která bude přímo interpretovat načtená data pracovníkům na repasním pracovišti na vizualizační tabule) a konfigurační část (část, ve které se budou vytvářet schémata podvozků určená k vizualizaci).

Vizualizace podle zadání bude rozdělena na dvě vizualizační tabule (levá polovina podvozku se vizualizuje na jedné obrazovce a pravá část na druhé). Musí být ovšem zachován základní vzhledový layout.

U konfigurace bude také zachován základní layout stránky, přičemž musí odstranit nedostatky původní vizualizace, jako např. složitou přenositelnost mezi zařízeními, nahrávání obrázků pouze jedné velikost v aplikaci třetí strany nebo omezení maximálního počtu vizualizovaných spojů. Celá tato část by měla být jednoduše a intuitivně ovládatelná a přehledná.

Předpokladem pro dosažení daných cílů je seznámit se a porozumět příchozím datům především z utahovacích zařízení, která jsou používána na montážní hale M13 na automatické utahovací stanici, ale také se zbylými zdrojovými daty (záznamy z výrobního plánu a informace o změně polohy automobilů na montážní lince).

2 Podrobnější představení problematiky

Toto téma jsem si vybral, jelikož se zajímám o vývoj aplikací a bylo možné ho vytvořit pomocí nástrojů .NET, .NET CORE a programovacího jazyka C#, které mi jsou taktéž velmi blízké. Práce mi umožňuje získat důležité zkušenosti a především praxi do budoucnosti. Díky tomuto tématu se blíže seznámím s realtimovými komunikačními technologiemi, sběrem informací (přes připravené Api) ze zdrojových služeb a s utahovacími zařízeními na montážní hale M13.

2.1 Popis pracoviště

Podvozková zástavba je jednou z důležitých montážních operací při montáži automobilu. Jedná se o proces, kdy se karoserie automobilu sesazuje a následně spojuje s předpřipraveným podvozkem. Podvozek je osazen pohonným agregátem, převodovkou, přední nápravou s tlumiči a dalšími díly. Zde k takto osazenému podvozku se připevňují díly zadní nápravy s tlumícími jednotkami, výfuk, palivová nádrž atd. Zkompletovaný podvozek přijíždí na rámu pod hlavní linku, kde se po závěsech dopravují karoserie.

Celý proces se odehrává na celkem šesti taktech (takt je úsek montážní linky, na němž se provádí jedna nebo více podobných operací) automatickými utahovacími zařízeními. Jelikož je proces vykonáván bez zásahu obsluhy, hrozí zde nedotažení jednotlivých spojů. Automobil s nedotaženými spoji na podvozku by mohl ohrozit všechny účastníky silničního provozu.

Za automatickou zástavbou podvozku je repasní pracoviště. Zde se případně špatně utažené spoje z automatického utahování repasují a následně se vystavuje potvrzení o správném provedení celé série utahovacích úkonů.

Utahovací zařízení po dokončení operace vracejí výsledky utahování (konečný utahovací moment, úhel utahování, atd.). Tyto informace se přenášejí a ukládají ve formátu JSON - pro běžného člověka čitelný formát, pro pracovníka na montážní lince, kde čas průjezdu jednoho automobilu je cca 1 minuta, jsou to data v nesrozumitelné formě.

VUS tato data zpracovává a interpretuje ve srozumitelné/rychle čitelné (vizuální) podobě.

2.1.1 Utahovací zařízení

Na pracovišti podvozkové zástavby se používají utahovací zařízení od firmy AtlasCopco. Jedná se o švédskou firmu, která se zabývá výrobou a distribucí průmyslových strojů a vybavení od kompresorů přes obráběcí a stavební stroje po elektrické a montážní vybavení. Používají se zde dva typy utahovacích zařízení: strojní PowerMacs a ruční Power Focus.

Power Focus jsou ruční utahovací zařízení, která hlídají správnost utahování (moment, úhel – to jsou dva hlavní ukazatelé správnosti utahování). Utahování do určitého momentu znamená, že se utahovací zařízení vypne po dosažení správného (předem nastaveného) utahovacího momentu. Během tohoto utahování zařízení hlídá správné úhlové utahování – v zařízení je umístěn gyroskop, který měří záporný úhel utahování. Pokud je tento úhel překročen po dobu delší než 30 otáček, zařízení se automaticky vypne a vrátí status NIO.

PowerMacs jsou strojní utahovací zařízení. Zde může být jedno utahovací zařízení nastaveno na utahování více než jednoho šroubu. Každé utahovací zařízení je osazeno více než jedním senzorem utahovacího momentu a úhlu, pod kterým je utahování prováděno, aby spoje byly co nepřesnější. I zde může dojít k chybě především v té podobě, že v zásobníku bude špatně vyskladněn šroub, atd. PowerMacs vrací stejná data jako zařízení Power Focus.

Oba typy utahovacích zařízení komunikují s ToolsNetem pomocí Open Protokolu (OP). OP je protokol vyvinutý firmou Atlas Copco pro komunikaci mezi utahovacími zařízeními a PLC, počítačem (s operačním systémem Linux nebo Windows), tiskárnou nebo právě ToolsNetem (systém pro ukládání výsledků utahování). Utahovací zařízení posílají data jako řetězce znaků – převážně čísel a musí se odtud potřebná data vyparsovat. O předzpracování dat se stará zdrojová služba SQS, z které bude systém VUS načítat data. Výsledky utahování už budou přicházet ve struktuře, nikoliv v OP. [1]

2.2 Existující implementace vizualizačního systému

Nynější verze vizualizace nesplňuje aktuální požadavky na modernizaci a zrychlování výroby. I konfigurační prostředí je zastaralé a není připravené na nástup elektromobilismu (při němž je potřeba konfigurovat více vizualizačních spojů ze všech stran podvozku – aktuální verze umí ukládat popisy pouze *nad* nebo *pod* model nikoliv *napravo* či *nalevo*

od modelu). Konfigurace se provádí pouze v předem nainstalované aplikaci na PC, což omezuje konfiguraci pouze na předem připravené počítače.

Systém nesplňuje požadavky pro rozšíření o novou funkcionalitu – není k dispozici dokumentace ani zdrojové kódy (jedná se o takzvaný black box).

2.2.1 Obrázky

V zastaralé podobě musí potřebné obrázky podvozků nahrát jiná pověřená osoba (jedná se o pracovníka, který tuto operaci musí dělat - v původní verzi neexistuje jiný způsob), která má přístup k aplikaci třetí strany (aplikace, která umožňovala nahrání obrázku). Tento mezikrok zdržoval celou konfiguraci a nástup nového modelu na výrobní linku musel být ohlašován a konfigurován s předstihem.

Dalším omezením nahrávaných podvozků je jejich velikost. Obrázek podvozku, který se nahrával pro konfiguraci a následnou vizualizaci, musí mít přesné rozlišení. S jiným rozlišením aplikace neumí pracovat.

2.2.2 Popisy s vodícími linkami

V předchozí verzi mají popisy předem nastavenou velikost, která omezuje délku textu. Tyto popisy také maji pouze předdefinované umístění a je stanoven maximální počet popisů. Při aktuálním výrobním procesu toto omezení nevadí, ale s přibývajícím množstvím spojů by bylo potřeba více boxů.

V předchozí verzi vodící čáry mohou být rovné nebo jednou zalomené, což by opět s přibývajícím množstvím spojů nemuselo stačit.

2.2.3 Změna polohy

Stará verze vizualizace není napojena na systém CarRFID, která sleduje polohu jednotlivých vozů na montážní lince a umožňuje zaslání těchto dat do jiných systémů. K vizualizaci je připojen pouze ruční scanner a pracovníci si musí každý vůz ručně načíst, následně je vizualizován podvozek s utaženými spoji. Tato verze byla vyhovující v momentě, kdy na podvozku nebylo správně utaženo více spojů, ale pokud jsou všechna utažení v pořádku, manuální načtení vozu je nadbytečnou operací.

2.3 Nová verze vizualizace

Nová verze by měla odstranit všechny výše zmíněné nedostatky. Jednodušší přístup ke konfiguraci je multiplatformní a přizpůsobený moderním mobilním zařízením. Toto přizpůsobení bude formou webové aplikace. Pro přístup ke konfiguraci bude poté stačit přístup pomocí webového prohlížeče, který je implementovaný nebo se dá doinstalovat do téměř každého operačního systému. Zjednodušení konfigurace především ze strany nahrávání obrázků podvozků vede k tomu, že obsluha nemusí jednotlivé obrázky přeškálovávat na specifické rozlišení a zároveň s ním nahrání a konfiguraci podvozku mohl obstarávat jeden pracovník. Tento bod zadání je vytvořen pomocí dynamického dopočítávání velikosti obrázku, umístění a velikosti popisů. Všechna data se dopočítávají pomocí procentuální velikosti simulované vizualizační obrazovky.

Vizualizační část se z jedné přehlcené obrazovky rozdělí na dvě. Řez podvozkem je zvolen podélný od přední po zadní část podvozku, přičemž v jedné polovině obrazovky bude polovina podvozku s popisy jednotlivých spojů, vodícími linkami a přesným označením, kde se na podvozku nachází. V druhé části bude informativní podbarvené pole (podle barvy pracovník zjistí, jestli je na dané části podvozku nesprávně dotažený spoj), ve kterém se budou nacházet informace o automobilu např. KNR, VIN, …

Všechna potřebná data k chodu VUS se budou načítat ze zdrojových služeb, které fungují na montážní hale M13:

- FIS – načítání výrobního plánu. Výrobní plán je sekvenční posloupnost vyráběných vozů obsahující jejich popis pro účely montáže.
- CarRFID – systém poskytující informace o změně polohy vozidel mezi jednotlivými takty na montážní lince
- SQS – systém, přes které jsou přístupné výsledky utahování z jednotlivých utahovacích zařízení, protože má přístup do databáze ToolsNetu.

Systém bude přijímat data ze služby CarRFID, čímž zjednoduší práci pro pracovníky na repasním pracovišti – nebudou muset každé vozidlo načítat ručně scannerem (jak tomu bylo doposud), ale bude se vizualizace zobrazovat automaticky.

Aby se zdrojové služby nevytěžovaly opětovným dotazováním, budou data (ze zdrojových služeb) ukládána do vlastní databáze systému VUS. Nebudou se muset načítat data z SQS při opětovné vizualizaci jednoho automobilu. Pokud by se nepoužila databáze, musel by se při každém pohybu linky stahovat výrobní plán (slouží jako propojení mezi KNR, které přichází z CarRFID a PR podmínkam, které jsou zadány ve VUS). Následně by nebylo kam ukládat data z konfigurace a při každém restartování by se data musela konfigurovat znovu.

PR podmínky jsou tvořeny PR čísly (PR číslo = vlastnost vozu pro účely montáže, například typ motoru, pravé / levé řízení atp.) PR podmínky se skládají z PR čísel pomocí AND, OR, NOT logických operandů.

Nově VUS bude poskytovat testovací vizualizace – pověřený pracovník se bude moci podívat na to, jak byl každý podvozek utažen (tato funkcionalita bude přístupná pouze mimo repasní pracoviště).

Systém nebude řešen jako stará verze vizualizace, že bude tzv. black boxem, ale bude plně přístupný a dokumentovaný pro možnost případného rozšíření o nové funkcionality v budoucnu. Nebude vytvářený jako monolitická aplikace, ale bude rozdělen do více mikroslužeb, které budou plnit pouze jednu úlohu. Rozdělení bude také přívětivější pro rozšiřování funkcionality.

Systém by měl splňovat bezpečnostní požadavky ve společnosti Škoda Auto. Komunikace mezi službami, které nejsou na stejném serveru, musí být ověřená. Šifrovaná musí být také komunikace se zdrojovými službami, které fungují na montážní hale M13.

Zástupnost tohoto systému je podle bezpečnostních požadavků jednotná. Ovšem zástupnost zde není řešena typicky/standardně, ale bude zde aplikace spuštěna na dvou uzlech, které budou přístupné přes balancer (balancer bude rozdělovat rovnoměrně zátěž mezi jednotlivé uzly tak, aby v ideálním případě nedošlo k výpadku obou uzlů).

3 Návrh řešení

Nemají-li být služby se zdrojovými daty zatěžovány opakovaným dotazováním na jeden automobil, musely být zvoleny vhodné metody pro ukládání a přístup k datům. Následně byly vybrány technologie pro komunikaci mezi jednotlivými službami a byl vytvořen vývojový diagram.

3.1 Ukládání dat

Pro jednoduché, rychlé načítání a vkládání potřebných dat bylo zvoleno ukládání do databáze. O druhu databáze bylo rozhodnuto ze strany zadavatele – relační databáze Microsoft SQL Server.

3.1.1 Microsoft SQL server

Microsoft SQL Server je systém pro správu relačních databází vyvinutý společností Microsoft. Jako databázový server se jedná o softwarový produkt s primární funkcí ukládání a načítání dat podle požadavků jiných softwarových aplikací. Výhodou je, že může být na stejném stroji jako je aplikace, která data načítá/ukládá, nebo může být úplně decentralizovaná a přístupná po vnitřní síti nebo pomocí internetu z externích serverů/-počítačů. [6]

3.1.2 Přístup k databázi

Pro přístup k databázi z prostředí VUS byla zvolena varianta přístupu bez jakéhokoliv frameworku. Takto bylo rozhodnuto z důvodu, že frameworky zpomalují složitější SQL dotazy a musejí se poté přepisovat ručně. Následně z důvodu zabezpečení a uchování konzistence dat bylo rozhodnuto, že přístup k databázi bude mít pouze jedna mikroslu-

ba, která bude obstarávat pouze komunikaci s databází. Přímý přístup k databázi nebude mít ani konfigurační část systému.

3.2 Programovací jazyk

Programovací jazyk byl připomínkován ze strany CMS – používání C# s podporou .NET frameworku nebo .NET CORE. Tyto připomínky byly z důvodu, že softwarové oddělení, které vyvíjí a obstarává servis aplikací a systémů, tyto technologie využívá. Dalším požadavkem bylo vyhnout se technologii WCF (Windows Communication Foundation) pro komunici mezi komponentami, která je na základě dřívějších zkušeností z vývoje nevhodná pro použití při vývoji nových systémů pro zákazníka Škoda Auto.

3.3 Komunikace systému VUS

Systém již od začátku neměl být koncipován jako jedna monolitická aplikace (z důvodu jednodušší rozšiřitelnosti, údržby a servisu), ale jako více mikroslužeb. Aby služby mohly mezi sebou komunikovat a zpracovávat data, musela se určit technologie ke komunikaci. Podle zadání celý systém musel komunikovat a pracovat v reálném čase, proto byly vybírány pouze realtimové komunikační technologie. Bylo možno vybrat více technologií, ale podle zadání jsem používal SignalR.

3.3.1 SignalR

SignalR byl zvolen, protože podporuje realtimovou komunikaci, jak mezi aplikacemi psanými v C# s podporou .NET/.NET CORE, tak i mezi webovým serverem (ASP .NET/ASP .NET CORE) a webovým klientem (JavaScript). Tato technologie komunikace byla už od počátku vyvíjena pro aplikace psané v .NET CORE/.NET. Původně byla autory (David Fowler a Damian Edwards) navržena pro aplikace ASP .NET/ASP .NET CORE, aby umožnila programátorům vyvíjet webové aplikace, kde mohli klienti dostávat informace ze strany serveru (od jiného klienta přes server) v reálném čase, tomu slouží API pro provádění RPC server-klient. [5] Poté, co autoři technologii vyvinuli, Microsoft ji odkoupil. Následně v dalších verzích byla technologie upravena pro podporu klientské části aplikace vyvíjené v :NET/.NET CORE, nikoliv pouze v JavaScriptu. Technologie poskytuje tzv. API hooks pro snadnější práci, například vyvolání události při připojení/odpojení klienta, seskupení připojení, autorizaci.

Tato technologie funguje podle schématu klient-server. V tomto případě server funguje jako pouhý hub (rozbočovač), který, pokud dostane data, je rozešle všem aktivně

připojeným klientům. Ve verzi SignalR v2 byla přidána funkcionalita zasílání dat omezené skupině klientů nebo pouze jednotlivcům.

Klient (JavaScriptový nebo psaný v .NET/.NET CORE) se připojí k serveru a udržuje tzv. aplikační keep alive (v pravidelných intervalech posílá serveru informaci, že je připojen). Naslouchá příjem dat ze serveru, který není omezen pouze na naslouchání na určitém portu, ale i na určitém názvu metody, která se volá – nepřijímá všechna data, která jsou po portu posílána. Ke komunikaci technologie využívá aplikační protokol WebSocket. [7]

WebSocket

WebSocket (WS) je aplikační protokol, který funguje na 7. vrstvě síťového modelu OSI. Tento protokol podporuje plně duplexní komunikaci pro jedno TCP spojení. WebSocket je plně kompatibilní s protokolem HTTP. Kompatibilita je zajištěna tím, že k navázání spojení se používá HTTP handshake a následně HTTP upgrade, pomocí něhož se protokoly přepnou a komunikace už probíhá přes WS. [3]

3.4 Konfigurační a vizualizační část

Po vytvořeném novém systému bylo požadováno, aby k němu byl co nejjednodušší přístup. Z dostupných možných řešení byla zvolena varianta webové aplikace původně pouze pro konfiguraci. K jejímu spuštění ze strany uživatele vyžadujeme pouze webový prohlížeč, kde pracovník, pověřený ke konfiguraci, může udělat všechny nutné kroky sám. Nakonec byla také vizualizační část přidána do webové aplikace a vyvíjena jako tenký klient.

Tyto dvě části celého projektu jsou programované jako jedna webová aplikace psaná v ASP .NET CORE 3.0. Tato verze plně podporuje technologii SignalR pro komunikaci s webovými klienty.

3.4.1 ASP .NET CORE 3.0

ASP .NET CORE 3.0 je open source webový framework vyvinutý společností Microsoft a komunitou programátorů. Plně podporuje WPF (MVVM Pattern), který byl jedním z prvků mého zadání. Mezi jeho přednosti dále patří jeho modularita (pomocí NuGet balíčků), vývojář nemusí spouštět jednotlivé kompilace (kompilace projektu je nepřetržitá). [8]

3.5 Načítání dat ze zdrojových služeb

VUS potřebuje přijímat data z jednotlivých zdrojových služeb: FIS, CarRFID a SQS. Všechna data jsou přijímána pomocí platformy PSB (tato služba slouží ve společnosti Škoda Auto jako tzv. bridge (most) ke zdrojovým službám – je to bezpečnostní politika společnosti, aby aplikace třetích stran nevyužívaly přímý přístup do zdrojových databází nebo nevytvářely specifické nestandardní komunikační kanály mezi sebou navzájem).

3.5.1 FISové záznamy

V těchto záznamech je uložen výrobní plán jednotlivých automobilů na různých montážních linkách. Jsou zde uložené informace:

- KNR – jedinečné identifikační číslo vozu na montážní lince. Toto číslo je předáváno službou CarRFID – slouží ke spojení automobilu a podvozku.
- PR čísla – v těchto číslech je zapsána přesná výbava celého vozu (od typu motoru, přes jednotlivé asistenty automobilu až po barvu sedadel). Při konfiguraci obsluha zadá PR podmínku, která spojí přesně vizualizační podvozek s vozem na montážní lince.
- Model – označení modelu automobilu – slouží k vizualizaci.
- VIN – mezinárodní identifikační číslo motorového vozidla. Slouží k vizualizaci.
- Lfdnr -sekvenční číslo (číslo závěsu) vozidla, pomocí tohoto čísla služba žádá o další nové záznamy FIS.

Data jsou získávána pomocí REST API ve formátu JSONu.

3.5.2 CarRFID

Automobily na montážní lince jsou osazené RFID čipy (Radio Frequency IDentification, identifikace na rádiové frekvenci). Tyto čipy slouží k informaci o posunu vozu po montážní lince. V telegramech přicházejí následující informace:

- Hall – informace, na které montážní hale se vozidlo pohybuje (tento projekt bere automobily pouze na hale M13)

- Location – informace na jakém stanovišti/taktu se automobil aktuálně nachází (v projektu budou potřeba data pouze z taktu č. 49)
- KNR – jedinečné identifikační číslo vozu na montážní lince. Slouží k hledání ve FISových záznamech

Získávání dat je pomocí IBM MQ, a jsou zabalena v JSON formátu.

3.5.3 SQS

SQS je služba, ve které předzpracuje data z utahovacích zařízení a uloží je. Data zde jsou ve formátu:

- KNR – jedinečné identifikační číslo vozu na montážní lince.
- IPAddress – IP adresa utahovacího zařízení
- Bolt – číslo utahovaného spoje
- ST – číslo vřetene utahovacího zařízení
- Status – informace jestli byl spoj utažen správně nebo ne

Data jsou získávána pomocí REST API v JSONu.

3.5.4 REST API

REST API (REpresentational State Transfer Application Programming Interface) je rozhraní pro čtení, editaci nebo mazání dat ze serveru pomocí HTTP volání. K tomu slouží základní ovládací metody (GET, POST, PUT, DELETE, ...). [4] Oproti jiným architekturám navržených pro získávání dat je tato orientována datově, nikoliv procedurálně. [9]

3.5.5 JSON

JSON (JavaScript Object Notation) je nízkozátěžový datový formát pro výměnu dat. Je to formát, který je navržený jako plně textový (jednoduše psatelný i čitelný pro člověka). Vyskytuje se ve dvou základních strukturách:

- párová – název následovaný hodnotou
- listová – název následovaným listem (polem) hodnot

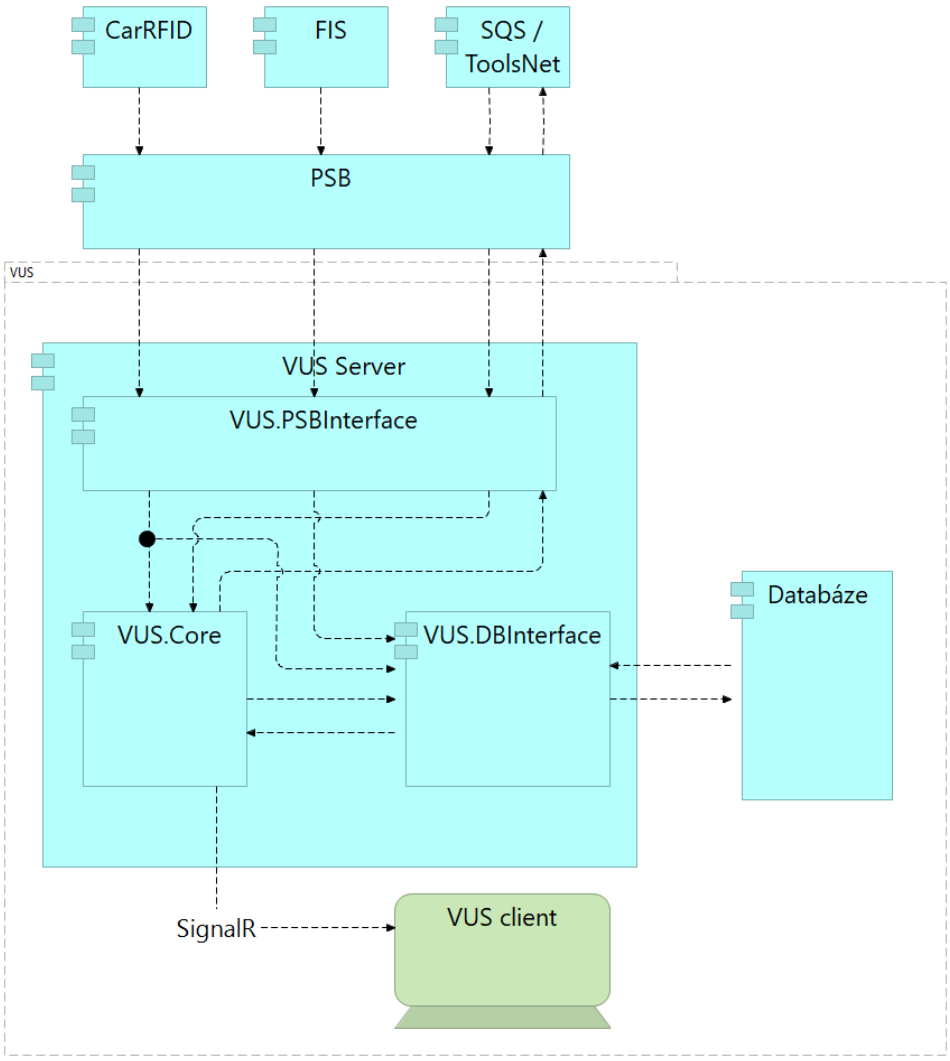
Celý datový formát může být také kombinací pávové a listové struktury. [2]

3.5.6 IBM MQ

Je zprávově orientovaný protokol vyvinutý společností IBM. Umožňuje bezpečně komunikovat nezávislým aplikacím mezi sebou pomocí zpráv. Je možné jeho využití na velkém množství platform (Linux, Microsoft Windows, OS/400, UNIX, ...). Mezi hlavní komunikační součásti patří: zpráva (nese samotná data nebo řídící informace), fronta (místo, kde se skladují/jsou čteny zprávy) a manager front (systém pro řízení jednotlivých front a zpráv). [10]

3.6 Návrh VUS

Systém VUS byl koncipován, jak jsem již zmínil, nikoliv jako monolitická aplikace, ale jako systém složený z mikroslužeb a tenkého klienta, který bude sloužit pouze k vizualizaci dat. Vývoj projektu byl řízen agilně - na začátku vývoje byly stanoveny dílčí cíle. Jednotlivé vývojové části byly prezentovány koncovému uživateli k připomínkám. V následující iteraci byly zpracovány připomínky s další částí projektu. Díky této strategii se předešlo složitým rollbackům během vývojových cyklů, ale prvotní návrh počítající s rozvržením na tři mikroslužby, konfigurační web a tenkoklientskou aplikaci neuspěl.



Obrázek 3.1: Původní návrh VUS

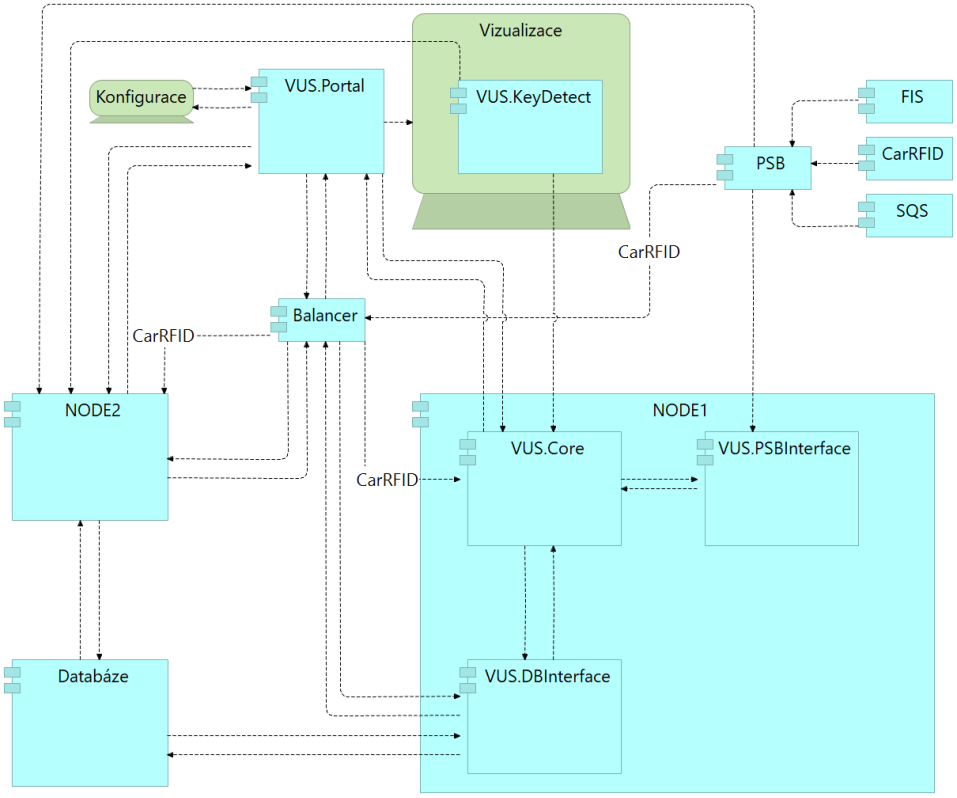
VUS.PSBinterface v tomto případě měl být SignalR hub a zároveň přebírat data z PSB (FIS, CarRFID a data z utahovaček z SQS). VUS.DBInterface komunikoval s databází. VUS.Poratál sloužil pouze ke konfiguraci vizualizace. Komunikoval s VUS.DBInterfacem pomocí REST API. VUS.Core modifikoval data pro klienta, aby klient byl pouze tenkým klientem a jen je vykresloval.

V průběhu implementace prvního řešení bylo zjištěno, že toto je slepá ulička. Jedna z prvních připomínek ze strany zadavatele ohrozila jednoduchou rozšiřitelnost systému a bylo potřeba architekturu pozměnit. V tento moment ovšem nešlo o vývoj jako takový, spíše se zde prověřovaly a zprovozňovaly komunikační technologie (RERT Api a SignalR). Prvotní připomínkou bylo, že systém nepočítal se záložním načítáním CarRFID pomocí scanneru. Načítání mělo probíhat v době, kdy služba CarRFID nebude dostupná, nebo pokud bude potřeba opravit více spojů a na pozici repasního pracoviště již přije-

de nový automobil. Ze strany CMS bylo zamítnuto vytváření tlustého nebo hybridního klienta, byl prosazován pouze tenký klient.

V tomto kroku se architektura předělala do podoby, kdy se řídící komponentou stala služba VUS.Core (do této služby se přesunul SignalR hub). VUS.Client byl zrušen a přesunut na webový portál. Načítání scannerem mělo být ve službě VUS.Core, což se ukázalo také jako špatná volba. V tomto případě vhodnému řešení bránily dvě překážky: firewall služby zakazuje načítání vstupů do služeb. Tento problém se dá obejít implementací nízkoúrovňového záznamníku úderů klávesnice, ovšem druhým požadavkem bylo získávat data pouze z jednoho určeného scanneru, nikoliv z více. Nízkoúrovňový záznamník úderů kláves neumí rozpoznat, z jakého zařízení data dorazila. Tím vznikla třetí verze vizualizace, do něhož byla začleněna aplikace, která běží na pozadí a která načítá vstup ze scanneru.

Následné malé změny se odehrávaly především ve změnách zdrojových služeb s daty. Hlavně s daty z utahovacích zařízení. Prvotně se data měla získávat přes PSB ze služby ToolsNet, ale ten nemá rozhraní pro PSB - proto byl zamítnut. Poté se odsouhlasila služba ŠUT. V této službě bylo zjištěno, že data jsou nekonzistentní a nejsou zde uložena všechna. Tím se načítání dat vrátilo zpět ke službě SQS. Finální vývojový datagram vypadá následovně:



Obrázek 3.2: Aktuální návrh VUS

Zde je vidět, zástupnost systému VUS – je umístěn na dva uzly (rozkreslený je pouze jeden, ale oba uzly jsou totožné), které jsou za balancerem –rozděluje zátěž na oba uzly stejně a při výpadku jednoho z uzlů celý provoz přesměruje pouze na jeden funkční. Služba PSB a další zdrojové služby (FIS, CarRFID, SQS) nejsou součástí systému VUS.

Je tu také naznačeno, že vizualizace a konfigurace jsou rozděleny pouze oprávněními, nikoliv fyzicky.

4 Realizace řešení

VUS je rozdělený na webovou aplikaci (vizualizační a konfigurační část) a mikroslužby. Celý systém komunikuje pomocí technologie SignalR a REST API.

4.1 VUS.Core

Jedná se o mikroslužbu, která řídí komunikaci a výměnu dat mezi jednotlivými službami a webovým portálem (zde se jedná pouze o komunikaci týkající se vizualizace). O komunikaci se stará SignalR hub. Hub, jakožto rozbočovač, standardně rozesílá data, která přijdou. Jsou rozesílána jako multicast zprávy (všem připojeným klientům). Toto byl první problém, protože bezpečnostní politika ve společnosti Škoda Auto výslovně zakazuje zasílání zpráv formou broadcastu nebo multicastu. Veškerá komunikace mezi službami musí být řešená pouze unicastově. Bohužel v momentě, kdy by bylo jednodušší využití muslticastu, musí se rozesílání řešit postupně jednotlivým klientům.

VUS.Core dále hlídá připojení a nepřipojení jednotlivých klientů. Nedostatkem technologie SignalR je, že si ukládá pouze connId (identifikátor spojení). Programátor si nemůže nadefinovat alias k tomuto identifikátoru. Toto bylo řešeno ukládáním spojení aliasu a id do slovníku. Problém nastával, když se klient odpojil a znovu připojil v době kratší než 30 sekund (po tuto dobu běží u SignalR reconnecting time), při níž nenastane odpojení klienta ze strany hubu, ale pouze ze strany klienta. Na událost odpojení reaguje slovník tak, že daný záznam ze slovníku smaže, ale pouze až po uplynutí třiceti sekundového reconnecting time. Problém zde byl ohledně nasimulování této události – stávalo se to pouze při extrémním vytížení počítače, ale ne pravidelně. Nakonec byla teto komplikace vyřešena při znovupřipojení, kdy se aktualizuje id spojení.

4.2 VUS.DBInterface

VUS.DBInterface je služba, jež zprostředkovává přístup do databáze VUS. Toto řešení bylo zvoleno z důvodu zabezpečení přístupu k lokální databázi VUS. Jelikož ani konfigurační část vizualizace nemá přímý přístup do databáze, musel zde být implementován přístup prostřednictvím REST API (pro komunikaci s portálem – SignálR se zde nevyužívá, aby se v případě konfigurace nezatěžovala řídící služba VUS.Core a vizualizace probíhala v reálném čase).

Do VUS.DBInterface se musela implementovat REST API serverová část a SignalR klient. Spouštění jednotlivých částí služby muselo být pouze v následujícím pořadí – nejprve SignalR klient a poté Reat Api server. V opačném pořadí se SignalR klient nespustil – RESR Api server převzal veškeré prostředky pro navazování spojení a další nepovolil. I v případě, že se jednalo o komunikaci na jiném portu a obě komunikační technologie běžely každá na vlastním vlákně.

Řešení v propojení s REST API bylo zvoleno z důvodu, že telegram projde bez problému balancerem a je rozděleno podle aktuální zátěže, kdežto komunikace SignalR neumí projít přes balancer.

4.3 VUS.PSBInterface

Jedná se o službu obstarávající komunikaci s PSB (služba, která slouží ve společnosti Škoda Auto jako tzv. bridge (most) ke zdrojovým službám). Tato komunikace probíhá po REST API.

Celý proces získávání dat z PSB musí být přenášen zabezpečeným protokolem (podle standardů společnosti o bezpečnosti). Vzhledem k tomu, že se data získávají pomocí REST API, je využíván protokol HTTPS. Jedním z problémů bylo při vývoji klienta vytvoření self-signed certifikátu a jeho následné přidání na port (pod Windows 7), aby se mohla vyzkoušet komunikace s testovací serverovou částí. S tímto je spjaté přidání klientského certifikátu REST API klientovi, který žádá o zasílání dat z PSB. Nakonec celý proces byl vyřešen jednoduchou sekvencí příkazů.

V telegramu ze služby FIS jsou uloženy tyto pro VUS důležité informace: KNR13 a KNR jsou jednoznačné identifikátory jednotlivých automobilů na montážní lince. Služba CarRFID posílá změnu polohy pro KNR, nikoliv pro automobil. VIN je celosvětové jednoznačné označení automobilu. Model nese název modelu automobilu, který se vizualizuje. V poli statuses jsou uloženy statusy automobilů (razítka, že automobil prošel všemi místy na výrobní lince). Toto pole získáváme z důvodu položky lfdnr, ve které je uloženo sekvenční číslo automobilu. Pomocí tohoto čísla se vyžadují nová data z FISo-

vé služby. PrNumbers jsou PR čísla, podle kterých VUS načítá jednotlivé vizualizační nakonfigurované obrázky podvozků. V konfiguraci jsou tzv. PR podmínky a pokud je automobil splňuje, je daný podvozek vizualizován.

CarRFID služba posílá telegramy s daty, ve kterých jsou uloženy následující informace: hall, kde je název montážní haly (pro VUS je to hala M13); KNR13, podle kterého spojuji podvozek s automobilem na montážní lince; Location je označení taktu/stanoviště, na které vozidlo přijelo (v tomto systému to je takt č. 49).

Proces získávání dat z MQQueue se odehrává paralelně ve více vláknech. Tato strategie byla zvolena z důvodu, že změna polohy vozidel nepřichází v přesných časových intervalech, ale v rozmezí 5 - 10 vteřin, kdy je volitelná rychlost montážní linky podle vytíženosti výroby. Více vláken bylo zvoleno, protože celý proces, podle předpisů společnosti Škoda Auto, získávání dat z MQQueue funguje podle hesla zeptej se a čekej na odpověď (než se fronta naplní). Kdyby se data získávala pouze sériově, mohla by se plnit jiná fronta než ta, na kterou se aktuálně dotazuje systém a aktualizace pozice by nenastala.

Data z SQS přicházejí ve struktuře: KNR jedinečné identifikační číslo vozu na montážní lince; IPAddress je IP adresa utahovacího zařízení; ST je číslo vřetene utahovacího zařízení; Bolt je číslo utahovaného spoje vřetene; Status informace o správnosti utažení spoje.

Příchozí data z jednotlivých služeb se zpracují, transformují do objektů a podle typu požadavku se odešlou jen zpět do služby Core nebo přes Core do služby DBInterface k uložení do databáze.

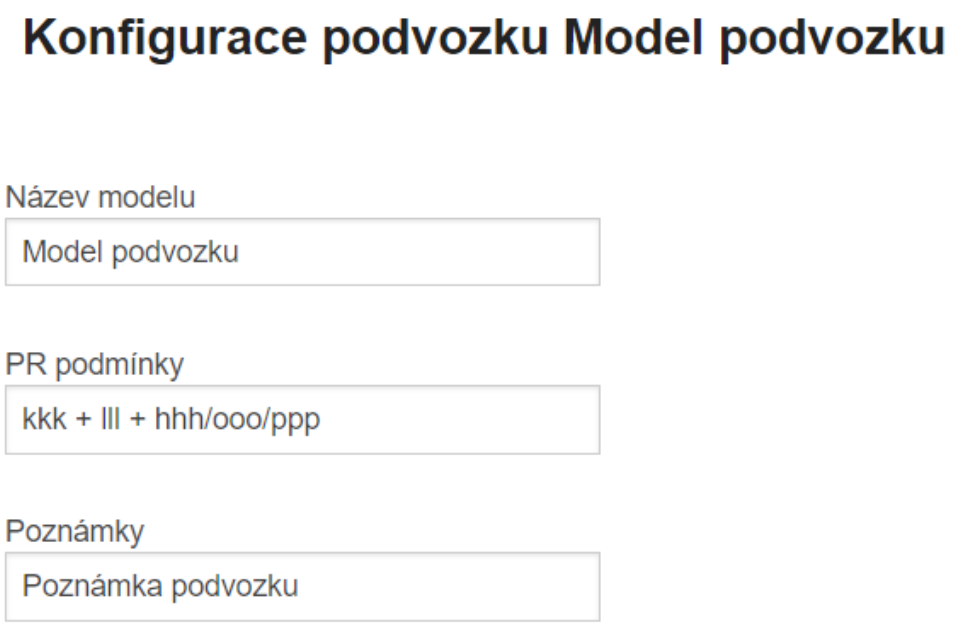
4.4 VUS.Portal

Jedná se o webovou aplikaci, která slouží ke konfiguraci vizualizačních podvozků a k samotné vizualizaci. Web byl vyvíjen pomocí technologie ASP .NET CORE 3.0, která plně podporuje připojení webového klienta (programovaného v JavaScriptu) pomocí technologie SignalR.

Layout webu byl vytvořen společností Škoda Auto – všechny weby společnosti mají dle zadání vzhledem odpovídat internímu standardu. Vzhled konfiguračního a vizualizačního podokna mohlo být vytvořen podle potřeby. Do pravého horního rohu bylo umístěno logo firmy CMS s. r. o. a vedle něj se nachází logo společnosti Škoda Auto a. s. (firemní politiky).

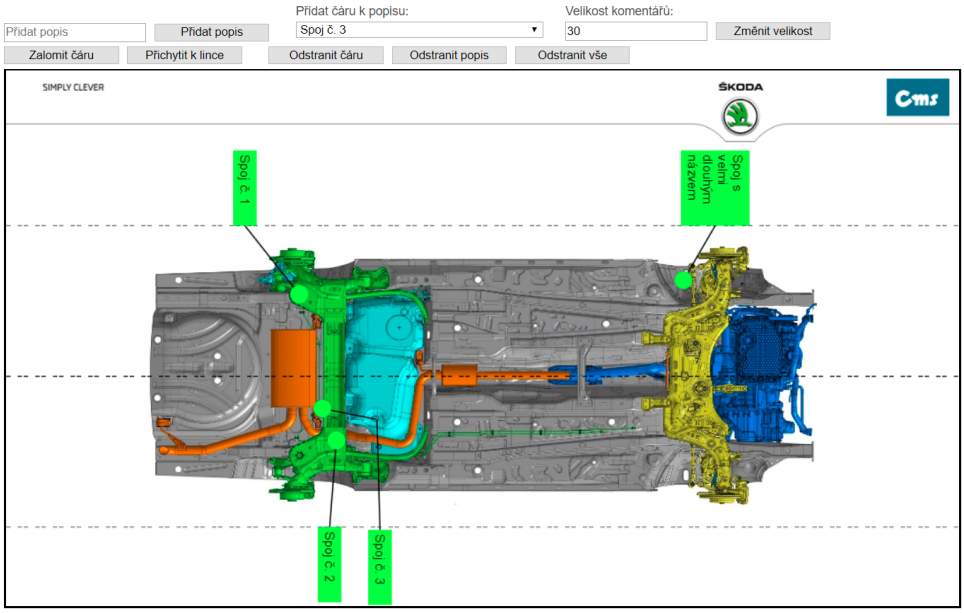
4.4.1 Konfigurace vizualizačního procesu

Konfigurační webová část se skládá z dílčích částí: přidání, editace a mazání nakonfigurovaných podvozků; a z hlavní části, v níž je možnost přidání jednotlivých popisů, spojnic a bodů jednotlivých utahovaných spojů spolu s editací základních polí.



Obrázek 4.1: Editační část přidávání popisů utahovaných spojů na webu

Zde je editační část hlavní konfigurační části, kde se edituje název modelu, PR podmínky (slouží k vyhledávání podvozků – spojovacími znaky mezi jednotlivými PR čísly jsou + (musí být zastoupeny PR čísla na obou stranách znaménka),/(jedna z obou stran znaménka musí platit) a ! (pokud je před číslem vykřičník, znamená to, že toto číslo nesmí být obsaženo)) a poznámky pro jednodušší orientaci obsluhy.



Obrázek 4.2: Přidávání popisů utahovaných spojů na webu

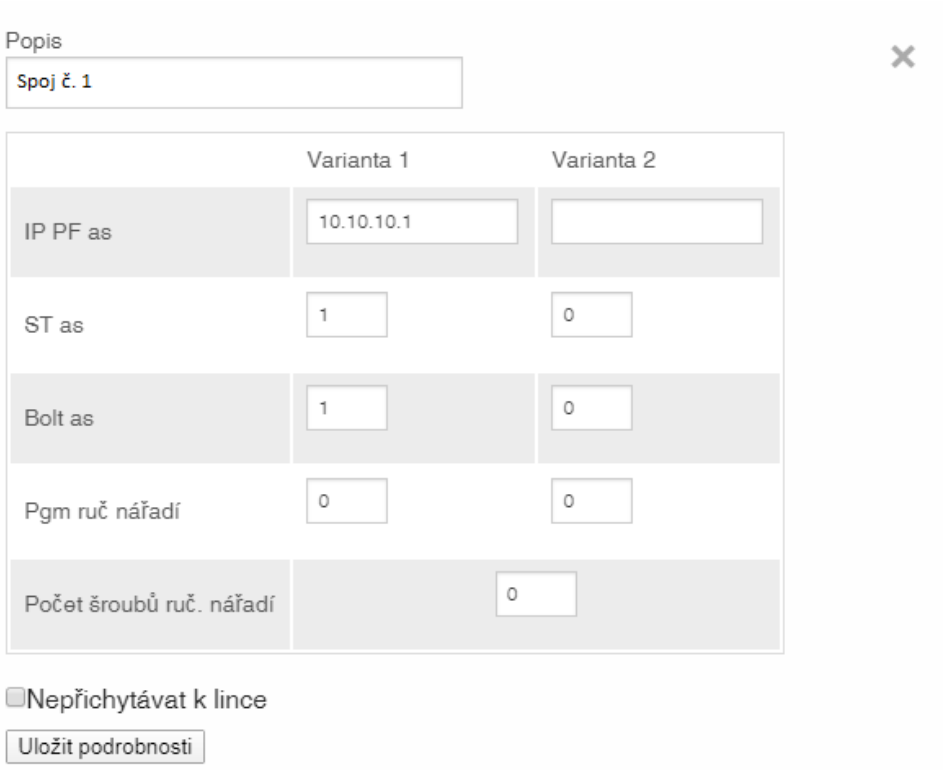
Na následujícím obrázku je ukázka konfigurace. Celá konfigurace je vykreslována do Canvasu. Popisy se přidávají jedním kliknutím do nasimulované obrazovky. Vodící linky se automaticky vygenerují po přidání komentáře. Čáry se mohou zalomit násobně. Lze nastavit velikost písma komentářů. Texty popisů jsou svisle – toto byla jedna z připomínek ze strany zadavatele.

Mohou se také přidávat samotné čáry, ale v rozbalovacím menu si obsluha musí nejprve vybrat popis, ke kterému spoji se bude čára vztahovat – při vizualizaci se podbarví i samotná čára podle barvy popisu (pokud čára bude přiřazena k jinému popisu může se stát, že linka bude mít jinou barvu než popis). Čára se přidává stejně jednoduše jako popis, jen s tím rozdílem, že se musí kliknout jak na počáteční bod, tak na koncový.

Se všemi komentáři, čárami i uzly čar se může po vytvoření posouvat podle potřeby. Pokud se s čímkoli začne hýbat, začnou se hýbat i přímo návazné čáry a vykreslení změny probíhá ihned, což je ideální pro pracovníka, který přímo vidí výsledek a nemusí si jej představovat. Toto bylo jedním z nedostatků staré verze konfigurace (pokud se měla čára zalomit – musely se vytvořit dvě čáry a po uložení program teprve dopočítal, jak bude zalomená čára vypadat).

Prostřední čárkovaná čára značí místo, kde se bude následně vizualizační část půlit. Je zde ošetřeno, aby žádný komentář ani čára nemohly vést přes tuto linku. Ve vizualizaci by část nebyla vidět. Pokud alespoň jeden komentář/linka porušuje tuto podmínku je zde zakázáno podvozek uložit a obsluha musí nejprve tuto chybu odstranit.

Horní a spodní čárkované linky okolo obrázku jsou vodícími linkami pro přichycení komentářů, aby všechny komentáře byly přehledně srovnány do dvou řad a nebyly rozházeny po celém modelu podvozku (připomínka k realizaci konfigurace ze strany zadavatele).



Obrázek 4.3: Konfigurace vizualizace jednotlivých utahovaných spojů

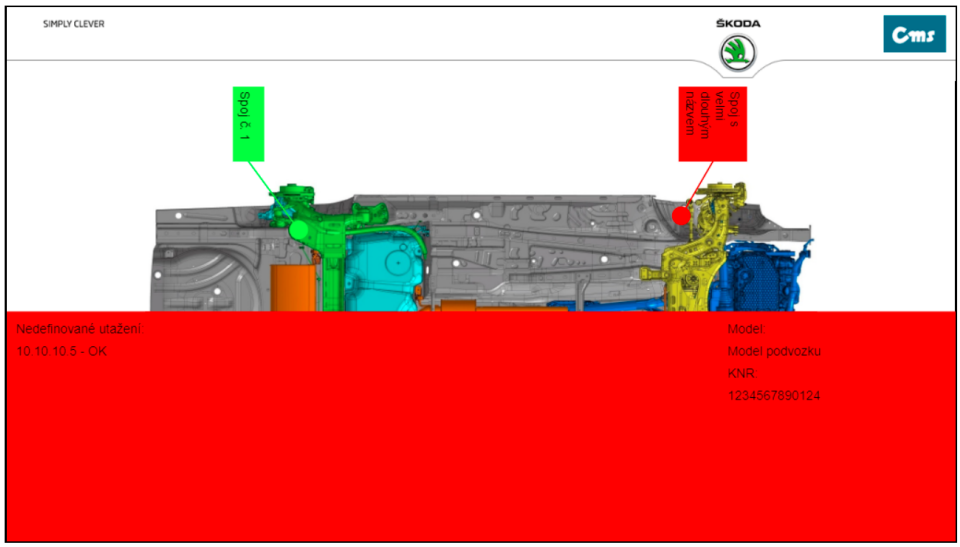
Každý popis má vlastní nastavení: IP adresy (podle této adresy se spojují data z utahovacího zařízení s popisem), ST, Bolt, Program ručního nářadí a počet šroubů. Toto vše ve 2 variantách (z důvodu zástupnosti jednotlivých utahovacích zařízení). Dalším nastavením je nepřichytávat k lince, které komentář nepřichytí k vodící lince nad obrázkem, ale nechá jej na místě, kde byl původně zanechán.

Pokud u každého popisu není vyplněna alespoň jedna IP adresa, obrázek se také neuloží. Samozřejmě nestačí pouhé vyplnění IP adresy, je potřeba navíc vyplnit i příslušnou kombinaci dalších informací.

K přichycení k lince slouží příslušné tlačítko, které posune komentáře nahoru nebo dolů podle toho, zda byl jejich spodní/horní okraj (na základě toho, jestli se komentář nachází ve spodní nebo horní polovině) přichycen k horní/spodní čárkované čáře.

Podvozek lze uložit nebo zkopírovat. Kopírování jednotlivých podvozků bylo jedním z požadavků ze strany zadavatele. Slouží k tomu, aby pracovník obstarávající konfiguraci vizualizace utahovaných spojů nemusel jeden podvozek několikrát konfigurovat pro různé PR podmínky, ale mohl na jednom podvozku nakonfigurovat základní spoje jednou a nové spoje (pro jinačí PR podmínky) přidat nebo naopak odebrat.

4.4.2 Vizualizace

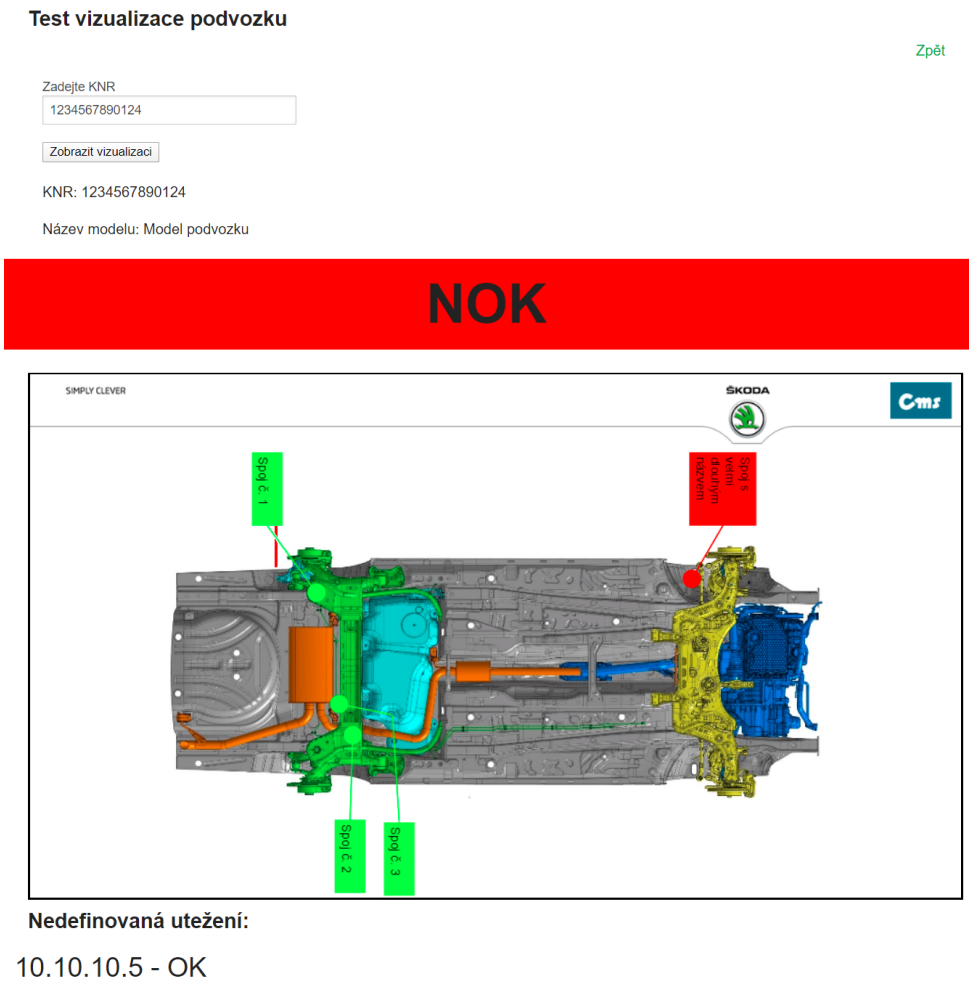


Obrázek 4.4: Vizualizační část VUS

Další částí je samotná vizualizace, která je pro jednodušší orientaci (zda se na podvozku někde nachází chybně utažený spoj) rozdělena na dvě obrazovky. Podvozek je rozdělen horizontálně na 2 poloviny (viz obrázek). V jedné polovině je podvozek s popisy a linkami a v druhé svítí zelený nebo červený obdélník – podle toho, jestli jsou všechny spoje správně utažené nebo ne. Toto řešení bylo zvoleno, aby obsluha repasního pracoviště nemusela zkoumat jednotlivé spoje na obrazovce, ale aby bylo jednoznačně vidět, zda nastala chyba při utahování nebo ne (ve staré verzi byl touto notifikací pro celý podvozek pouze malý obdélník, který se snadno přehlédl). Dále se zde zobrazuje název modelu, KNR, a nedefinovaná utažení (tj. utažení, která by podle konfigurace neměla nastat, ale nastala). Z důvodu předcházení určitým nedefinovaným utažením je možnost konfigurovat ip adresy, které se budou ignorovat.

4.4.3 Testovací vizualizace

Poslední částí webu je část testovací vizualizace, kde se zadá KNR a obsluha může překontrolovat, v jakém stavu byla jednotlivá spojení před repasí. Toto slouží pouze pro obsluhu mimo montážní linku, tudíž testovací vizualizace není rozdělena na poloviny, ale vypadá podobně jako konfigurace.



Obrázek 4.5: Ukázka testovací vizualizace

Je zde podbarvené pole s popisem, v jakém stavu byl podvozek, dále tu jsou vyznačené popisy s podbarvením podle toho, jestli spoj byl správně nebo špatně utažen (v této barvě je i označení, kde se spoj nachází a vodící linka mezi popisem a bodem spoje). Zobrazuje se zde název modelu, KNR a případně i nedefinované utažení.

4.4.4 Velikost canvasů

Velikost jednotlivých canvasů se počítá dynamicky podle velikosti obrazovky, čímž je zajištěna responzivita (hlavní požadavek na možnost konfigurace a vizualizace na různých platformách – od tabletu až po vizualizační tabule, na kterých vizualizace bude fungovat).

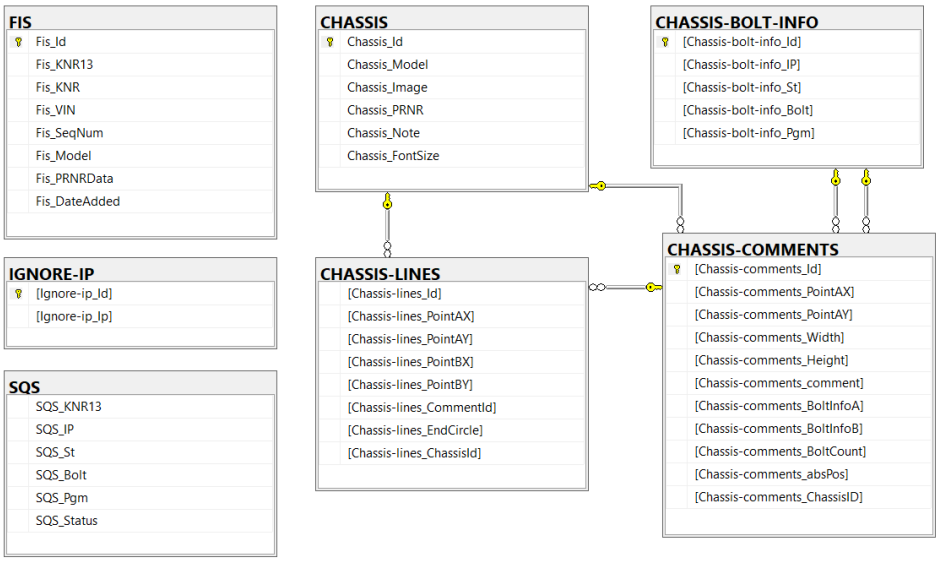
4.5 VUS.KeyDetect

Oproti ostatním službám se jedná o aplikaci, která běží na pozadí PC na repasním pracovišti. VUS.KeyDetect slouží k načítání dat z ručního scanneru. Hlavním důvodem je poskytnout obsluze pracoviště možnost znovu vizualizovat automobil, když na pracoviště přijede nový a předchozí ještě nebude dotažený nebo služba CarRFID bude mít dočasný výpadek. Tato aplikace je záložním řešením pro načítání příchozích automobilů na repasní pracoviště. Takto načtené KNR aplikace pomocí technologie SignalR přepošle do služby VUS.Core, aby se automobil vizualizoval.

Původní řešením bylo, aby se KNR ze scanneru (pouze ze scanneru, nikoliv z jiné možné klávesnice) načítalo přímo do služby VUS.Core. Toto řešení bylo zamítnuto, protože přímému načítání vstupu do služby brání firewall. Po obejití firewallu služba nedokázala určit z jakého zařízení data přišla, což byl největší podnět k přesunu načítání do aplikace.

4.6 Databáze

Databáze v tomto systému slouží do jisté míry jako forma cache paměti. Jsou tu ukládány záznamy ze všech zdrojových služeb (aby nebyly zatěžovány opětovnou komunikací). Pro zdrojovou službu SQS slouží databáze jako cache - je tu vidět, jaké informace do systému VUS byly načteny a zároveň při opětovné vizualizaci jsou data načtena z lokální databáze a nevytěžuje se zdrojová služba SQS. Ukládají se sem také všechna data z konfigurační části – názvy modelů, názvy obrázků, PR podmínky, velikost písma pro jednotlivé modely, rozložení komentářů po okně, texty komentářů, rozložení jednotlivých spojnic atd.



Obrázek 4.6: Databázový vývojový model

4.7 Komunikace mezi službami při základních operacích

Služby si vyměňují data mezi sebou pomocí technologie SignalR. Následující podkapitoly obsahují popis komunikace mezi jednotlivými moduly systému.

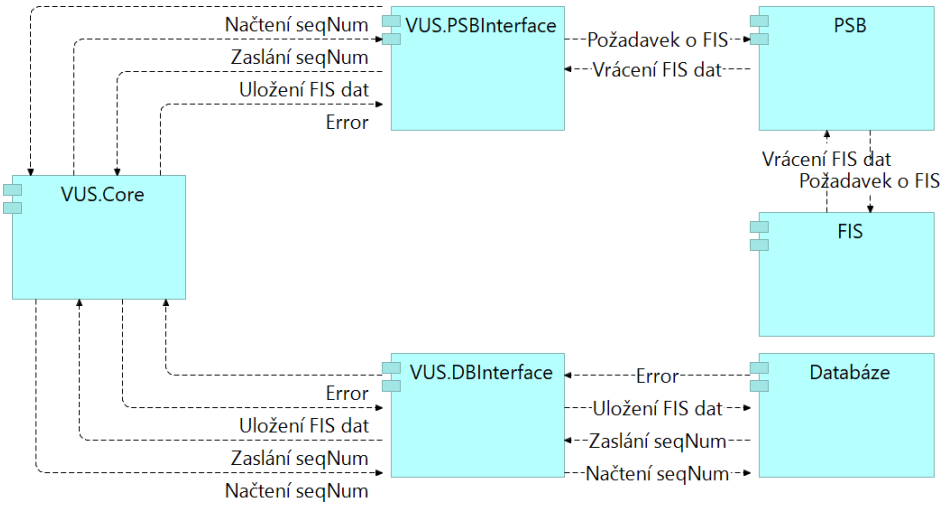
4.7.1 Ukládání/načítání konfiguračních dat

Ukládání a načítání jednotlivých dat v konfigurační části VUS.Portálu funguje pomocí REST API. Celý proces funguje jednoduše pomocí základní funkcionality této technologie: požadavek, odpověď. Požadavek je dvojího typu:

- Požadavek o poslání dat např. při zobrazení základního menu, načtení konfigurace popisů utahovaných spojů nebo editace základních údajů o podvozku
- Požadavek na uložení dat – např. přidání nového podvozku, uložení konfigurace popisů nebo uložení základních údajů o podvozku při editaci

4.7.2 Načítání FISových záznamů

O načítání a ukládání FISových záznamů ze zdrojové služby FIS přes prostředníka PSB se stará služba VUS.PSBInterface. Jelikož tyto záznamy jsou ukládány do databáze a zde jsou i uloženy záznamy o posledním sekvenčním čísle, je celý proces tvořen dvěma kroky: prvním krokem je načtení posledního sekvenčního čísla a druhým samotné získání FISových záznamů.



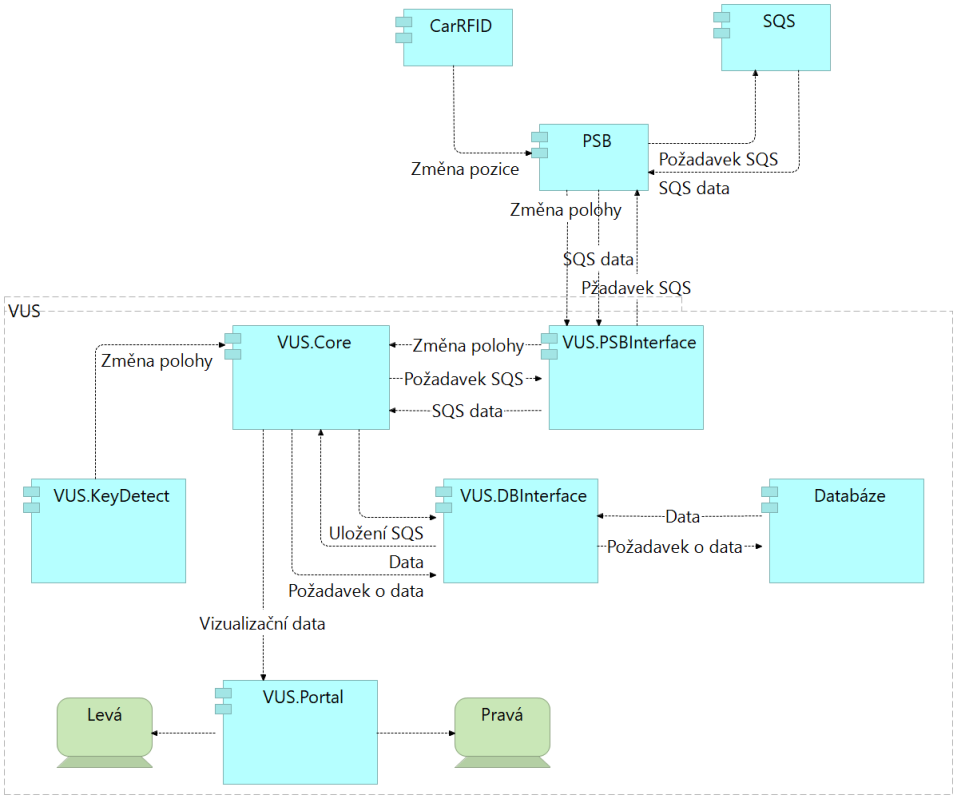
Obrázek 4.7: Proces nacítání a ukládání FISových záznamů

Zde je vidět, jak celý proces získávání dat, tak i průběh dotazování. Zdvojená komunikace přes VUS.Core je z důvodu, že technologie SignalR nepodporuje komunikaci mezi službami napřímo, ale pouze přes serverovou část (hub ve službě VUS.Core).

Získávání nových FISových záznamů je řízeno pomocí časovače (každých 5 minut se systém snaží načíst nové záznymy). Po vypršení časovače se pošle požadavek na VUS.DBInterface, aby bylo posláno poslední sekvenční číslo (seqNum). Databázový interface se doptá databáze na poslední sekvenční číslo. Výsledek požadavku se vrátí zpět do VUS.PSBInterface. VUS.PSBInterface pošle požadavek na PSB pro nový výrobní plán počínající sekvenčním číslem o jedna větším, než bylo zasláno z databáze (FIS vrací záznamy od čísla, které přijde včetně). Přijatá data se pošlou k uložení do databáze.

4.7.3 Proces vizualizace

Pro vyvolání nového vizualizačního procesu musí přijít změna polohy ze služby CarRFID, nebo pracovníci na repasním pracovišti musí naskenovat ručním scannerem KNR (informace o změně polohy přijde přes aplikaci VUS.KeyDetect).



Obrázek 4.8: Proces komunikace služeb při vizualizování podvozku

Poté co se informace o změně polohy dostane do VUS.Core, spustí se proces získávání dat pro vizualizaci. Pokud informace přijde z CarRFID, uloží se do databáze. Z databáze se vyberou data o podvozku (model, název obrázku, ...), popisy a vodící linky, popřípadě data z utahovacích zařízení, pokud již jsou uloženy v databázi, a pošlou se zpět do služby Core. Pokud již byla v databázi data z SQS, pošle se celý balík dat na VUS.Portal, v opačném případě je potřeba ještě získat data z SQS. Core pošle požadavek na VUS.PSBInterface pro zisk dat z utahovacích zařízení. PSBInterface tato data načte z PSB a vrátí je zpět na Coru. Všechna data se pošlou na portál a informace z utahovacích zařízení se pošlou do databázového interfacu k uložení. Ve VUS.Portal se data z utahovacích zařízení spojí s daty z konfigurace podvozku a pošlou opět pomocí technologie SignalR na jednotlivé klienty k vizualizaci (na pravého a levého zobrazovacího klienta).

Testovací vizualizace funguje téměř stejně, jen pokyn k vizualizaci zadává uživatel, který má k této operaci oprávnění přes webový prohlížeč. V posledním kroku se data odesílají na testovacího klienta, nikoliv na testovací pracoviště (pravý a levý vizualizační klient).

5 Zhodnocení řešení

Při realizaci tohoto projektu jsem musel několikrát předělávat architekturu systému. Architektura systému se měnila z důvodu, že nebylo vhodné vytvářet jednu monolitickou aplikaci, ve které by bylo složité do budoucna přidávat rozšíření funkcionalit a především orientaci při servisování v případě incidentů s funkcionalitou VUS (jelikož každá komponenta obstarává pouze jednu funkcionalitu, zorientuje se člověk obstarávající servis jednoduše při nahlášeném incidentu, kde chyba nastala). Druhým důvodem předělávání architektury bylo, že prvotní jednoduchý návrh nepočítal s rozšiřitelností a byl vytvořen pouze pro konkrétní zadání, nikoliv pro celkově rozšiřitelný a do budoucna udržitelný systém.

Během vývoje projektu se zadání (pomocí připomínek od zadavatele) postupně téměř celé změnilo a zůstal pouze základní cíl – vytvoření nové verze vizualizace. Postupem času se měnily zdrojové služby poskytující data VUS, přidávalo se záložní načítání KNR při posunu linky pomocí scanneru.

Načítání pomocí scanneru přispělo k rozšíření systému o aplikaci navíc, která se stará pouze o načítání vstupu ze scanneru a vytvoření podnětu pro vizualizaci. Původně mělo načítání probíhat přímo ve službě VUS.Core. Tomuto řešení ovšem bránily dvě překážky: služby nejsou navržené pro příjem dat – firewall služby přímé načítání zakazuje. To ovšem šlo obejít pomocí nízkoúrovňové detekce stisknutí kláves, ale key logger už neumí rozpoznat z jakého zařízení informace přišla. Ovšem služba musela brát vstupy pouze z jednoho zařízení, což nevyhovovalo původnímu zadání.

Následné připomínky vedoucích pracovníků ohledně informací a představ o vzhledu konfigurační a vizualizační části a dalších jejich funkcionalit byly často diskutovány osobně při představování frontendových částí systému (vzhled VUS.Portal). Z těchto schůzek vyšly najevo připomínky týkající se vzhledu např. že je potřeba na konci přímek mít označující kolečka, pozici popisů, že text nemá být horizontálně, ale vertikálně, atd. Co se týče funkcionality VUS.Portal jsem měl připomínky přímo od koncového uživatele již v době, kdy jsem celou aplikaci vytvářel a nemusel jsem se základními připomínkami čekat až na testovací provoz aplikace.

Největším problémem vývoje bylo vytvoření a především otestování REST API klienta při zasílání požadavků šifrovaně (HTTPS). Při tomto úkonu je potřeba přidat self

signed certifikát na port a přidat parametr, že se bude vyžadovat certifikát od klienta v nastavení firewallu. Jelikož jsem celý systém zprvu vyvíjel pod Windows 7, tak mi nešel vygenerovat ani vložit na port vlastní certifikát, který byl potřeba přidat, abych tuto funkcionalitu mohl otestovat. Při přechodu na operační systém Windows 10 všechny testy proběhly v pořádku – podařilo se vygenerovat a přidat certifikát.

Po nasazení na testovací servery na montážní lince v hale M13 systém až na drobné chyby v prvotní konfiguraci fungoval bez problémů. Ovšem připomínky probíhaly ze stran pracovníků jak na repasním pracovišti, tak o od lidí, kteří konfigurují jednotlivé podvozky.

Ze strany konfigurace byly připomínky na přerovnání jednotlivých ovládacích prvků (tlačítka, textových vstupních polí a rozbalovacích nabídek jednotlivých popisů pro přidávání nových čar). Následně byla připomínka na to, že po kliknutí na tlačítko uložit (uložit konfiguraci podvozku) se přejde na hlavní stránku a pokud chce uživatel daný podvozek zkopírovat, musí znovu podvozek rozkliknou, zkopírovat, nakonfigurovat a uložit – v tomto případě byl při opravě vynechán krok přesměrování na hlavní stránku.

Vizualizační část byla připomínkována jak společností CMS, tak zaměstnanci Škoda Auto. V tomto případě se jednalo o otočení obrázku podvozku s popisky, umístění informačního pruhu a vizualizované části na obrazovce. Ze strany pracovníků na repasním pracovišti byly připomínky na velikost písma popisů, to ovšem už bylo na pracovnících, kteří se starají o konfiguraci jednotlivých podvozků. Z mé strany jsem přepsal výchozí hodnotu velikosti písma z velikosti textu 16 px na 26 px.

6 Závěr

Cílem této práce bylo vytvořit novou verzi vizualizace utahovaných spojů za podvozkovou zástavbou na repasním pracovišti ve společnosti Škoda Auto na montážní hale M13. K samotné vizualizaci utahovaných spojů bylo potřeba také vytvořit konfigurační prostředí vizualizovaných spojů. Obě části jsou přehledné a podobné předchozí verzi pro jednodušší orientaci pracovníků při přechodu mezi verzemi. Celý systém pracuje v reálném čase. Zadání bylo splněno v celém rozsahu včetně zpracování značného množství připomínek ze strany zadavatele.

Oproti původnímu zadání byla přidána testovací vizualizace (pověřený pracovník může zkontrolovat, jak byly utažené spoje na automobilech, které prošly repasním pracovištěm, případně kolik a jaká nedefinovaná utažení se zobrazují). Bylo přidáno dodatečné načítání KNR automobilů pomocí scanneru při výpadku zdrojové služby pohybu automobilů na montážní lince CarRFID.

Systém VUS byl pro jednodušší servis a přidávání funkcionalit rozdělen z monolitické aplikace do tří mikroslužeb, které se starají jen o dílčí úlohy (řízení komunikace mezi službami, načítání dat ze zdrojových služeb a o přístup k databázi), aplikace, která se stará o zachytávání dat ze scanneru a webovou aplikaci, ve které je umístěna konfigurační a vizualizační část projektu.

Do budoucna se systém bude upravovat a rozšiřovat pro nové nároky na zrychlení výroby, zjednodušování obsluhy a přidávání více možných popisů jednotlivých spojů s masivnějším nástup elektromobilismu a jiných alternativních pohonů, jejichž díly budou implementovány jakoukoliv částí do podvozkové zástavby.

Literatura

[1] *Atlas Copco*. 2020. URL: https://www.atlascopco.com/cs-cz (cit. 03. 04. 2020).

[2] *ECMA-404*. 2. vyd. Geneva: Ecma, 2017.

[3] *Tools.ietf.org*. 2011. URL: https://tools.ietf.org/html/rfc6455 (cit. 03. 04. 2020).

[4] *Tools.ietf.org*. 2014. URL: https://tools.ietf.org/html/rfc7231%5C#section-4 (cit. 03. 04. 2020).

[5] *Wikipedia.org*. 2019. URL: https://cs.wikipedia.org/wiki/Remote_procedure_call (cit. 03. 04. 2020).

[6] *Wikipedia.org*. 2020. URL: https://en.wikipedia.org/wiki/Microsoft_SQL_Server (cit. 03. 04. 2020).

[7] *Wikipedia.org*. 2020. URL: https://en.wikipedia.org/wiki/SignalR (cit. 03. 04. 2020).

[8] *Wikipedia.org*. 2020. URL: https://en.wikipedia.org/wiki/ASP.NET_Core (cit. 07. 04. 2020).

[9] *Wikipedia.org*. 2020. URL: https://en.wikipedia.org/wiki/Representational_state_transfer (cit. 03. 04. 2020).

[10] *Wikipedia.org*. 2020. URL: https://en.wikipedia.org/wiki/IBM_MQ (cit. 07. 04. 2020).