Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství

Diplomová práce

Návrh komunikačního protokolu pro sběr telemetrických dat a řízení senzorických systémů vagónů v železniční dopravě

Šimon KAVAN

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Šimon Kavan

Informatika

Název práce

Návrh komunikačního protokolu pro sběr telemetrických dat a řízení senzorických systémů vagónů v železniční dopravě

Název anglicky

Proposal of communication protocol for collection of telemetric data and management of sensoric systems in rail transportation

---

Cíle práce

Cílem práce je provést analýzu požadavků a navrhnout koncepční řešení pro backendový informační systém sbírající data z telemetrických jednotek ve vagónech železniční dopravy. Kromě zajištění funkcionality bude také brán ohled na bezpečnost. Výsledek bude sloužit jako jeden ze vstupů zadávací dokumentace k realizaci softwaru.

Metodika

Práce bude rozdělena do dvou částí. První část práce bude teoretická a bude popisovat a citovat nástroje a techniky použité v druhé praktické části práce. Druhá část práce bude projektová dokumentace podle standardů UML, BPMN, TOGAF, Archimate a ISO-OSI, která bude obsahovat:

1) Sběr a organizaci funkčních i nefunkčních požadavků.

2) Rešerši existujících možných variant řešení ve světě IoT.

3) Návrh systémového rozhraní mezi telemetrickými IoT jednotkami a budoucím backend systémem včetně jeho provozního zabezpečení.

Doporučený rozsah práce

60-80 stran

Klíčová slova

IoT; security; telemetry; protocol; backend system

Doporučené zdroje informací

FOWLER, M. *UML distilled : a brief guide to the standard object modeling language*. Boston: Addison-Wesley, 2004. ISBN 0321193687.

GUTMANN, P. *Cryptographic security architecture : design and verification*. Berlin: Springer, 2004. ISBN 978-1-4419-2980-8.

SALAM, A. *Internet of things for sustainable community development : wireless communications, sensing, and systems*. Cham: Springer, 2020. ISBN 978-3030352905.

SCHEDLBAUER, M. *The art of business process modeling : the business analyst's guide to process modeling with UML & BPMN*. Sudbury: Cathris Group, 2010. ISBN 978-1-4505-4166-4.

Předběžný termín obhajoby

2022/23 LS – PEF

Vedoucí práce

doc. Ing. Vojtěch Merunka, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 7. 3. 2023

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 13. 3. 2023

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 14. 03. 2023

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Návrh komunikačního protokolu pro sběr telemetrických dat a řízení senzorických systémů vagónů v železniční dopravě" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 29.3.2023 _____

Poděkování

Rád bych touto cestou poděkoval vedoucímu mé práce doc. Ing. Vojtěchu Merunkovi, Ph.D. za podporu a cenné připomínky v průběhu jejího zpracování.

Návrh komunikačního protokolu pro sběr telemetrických dat a řízení senzorických systémů vagónů v železniční dopravě

Abstrakt

Diplomová práce řeší možnosti a specifika komunikačního protokolu pro sběr telemetrických dat a řízení senzorických systémů vozů v nákladní železniční dopravě. Práce se zabývá analýzou a sběrem požadavků, koncepcí a návrhem centrálního systému a komunikačního protokolu tak, aby se její výstup dal využít jako součást zadávací dokumentace. Součástí práce je rešerše postupů, dílčích komponentů a doporučení. Práce zohledňuje bezpečnostní stránku problematiky, včetně již existujícího hardware komunikačních jednotek.

Klíčová slova: IoT, sběr požadavků, železniční nákladní doprava, komunikace, zabezpečení, TOGAF, ArchiMate, UML, CoAP, telemetrie, protokol, centrální systém

Proposal of communication protocols for collection of telemetric data and management of sensoric systems in rail transportation

Abstract

Thesis aim at analysis and proposal of specific communication protocol for telemetric data collection and sensorics system management in cargo rail transportation.

Thesis starts with business analysis of stakeholders and their requirements, high-level proposal of concept and architecture of central management system. Outputs of this thesis should help build technical part of contract specification for the developer.

Thesis should also cover basic security topics of the overall design and communication protocols, also with regard to already existing hardware of telemetry units.

Keywords: IoT, requirements, cargo rail transportation, communication, security, TOGAF, ArchiMate, UML, CoAP, telemetry, protocol, backend system

Obsah

1 Úvod

Komunikace a sběr telemetrických dat je v dnešní době důležitým nástrojem pro optimalizaci nákladů v železniční nákladní dopravě.

Informace o poloze nákladního železničního vozu (dále jen vozu) a stavu jeho nákladu mohou pomoci optimalizovat navazující logistiku přepravovaného zboží. Informace o fyzickém stavu vozu mohou umožnit plánovat jeho servisní zásahy. Informace o času a lokalitě jednotlivých událostí hlídaných datovým modulem mohou pomoci řešit pojistné události. Tyto a další údaje lze sbírat za pomoci autonomních datových jednotek umístěných na vozech, zpracovávat a dále distribuovat v rámci centrálního řídícího systému.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem této práce je provést analýzu požadavků, navrhnout koncepční řešení pro backendový informační systém sbírající data z telemetrických jednotek ve vozech a doporučit komunikační protokol pro výměnu dat. Kromě zajištění funkcionality bude také brán ohled na bezpečnost. Výsledek bude sloužit jako jeden ze vstupů zadávací dokumentace k realizaci softwaru.

2.2 Metodika

Práce bude rozdělena do dvou částí. První část bude teoretická a bude popisovat a citovat nástroje a techniky použité v druhé praktické části práce. Druhá část práce bude projektová dokumentace dle standardů UML, BPMN, TOGAF, ArchiMate a ISO-OSI, která bude obsahovat:

1) Sběr a organizace funkčních a nefunkčních požadavků.
2) Rešerši existujících možných variant řešení ve světě IoT.
3) Návrh systémového rozhraní mezi telemetrickými IoT jednotkami a budoucím backend systémem včetně provozního zabezpečení.

2.2.1 Postup zpracování

1) Pro samotný návrh je nejprve potřeba zjistit, čí vstupy budeme v rámci analýzy zpracovávat – identifikace účastníků systému (stakeholders).
2) Dále bude potřeba sebrat požadavky od jednotlivých účastníků takového systému.
3) V dalším kroku je potřeba zpracovat požadavky a provést nad nimi základní analýzu výsledných potřeb.
4) Následně bude potřeba připravit rešerši možných způsobů řešení těchto potřeb.
5) V praktické části se bude řešit:
   a) Návrh koncepce systému jako celku
   b) Návrh konceptu jednotlivých komponent

c) Návrh konceptu zabezpečení
d) Návrh komunikačního protokolu

Tyto budou dále rozpracovány do základních variant ve formě koncepčních diagramů, soupisu funkčních a nefunkčních požadavků a dalších vstupů pro dodavatele takového systému.

6) Před dokončením a odevzdáním bude práce konzultována se zadavatelem a vedoucím práce.

2.2.2 Použité nástroje a rámce

- Pro sběr a třídění požadavků bude použito pravidel business analýzy dle BABOK (BRENNAN, 2009).
- Pro pochopení a optimalizaci obchodních procesů bude využito notace BPMN, protože bez formálního popisu nelze procesy formalizovat a následně optimalizovat (SCHEDLEBAUER, 2010).
- Návrh a koncepty systému a jeho komponentů bude realizován pomocí modelovacího jazyku ArchiMate a metodiky TOGAF.
- Zabezpečení bude zvažováno pro každý aspekt architektury (GUTMANN, 2019) již od počátku a navrženo bude s přihlédnutím k moderním trendům v bezpečnostní architektuře aplikací (OWASP, PKI a dle obecně platných doporučených pravidel).

3 Teoretická východiska

Pro dosažení cíle práce (návrhu komunikačního protokolu telemetrických datových jednotek a koncepční návrh centrálního systému) je potřeba nejprve porozumět prostředí a uživatelům, kteří budou tento systém používat. Pro samotný návrh je nutno vzít v potaz identifikované požadavky a také zvážit technické možnosti a omezení komunikace vyplývající z prostředí nákladní železniční dopravy.

3.1 Analýza

3.1.1 Požadavek

Pro sběr požadavků, jak uvádí BABOK je požadavek definován jako:

1) Schopnost požadovaná zainteresovaným účastníkem, jak vyřešit problém nebo dosáhnout splnění cíle.
2) Podmínka nebo schopnost, která musí být dosažena či zajištěna řešením ke splnění smlouvy, standardu, specifikace nebo jiného formálně definovaného dokumentů nebo jeho části.
3) Dokumentovaná reprezentace stavu nebo schopnosti jako v 1) nebo 2).

3.1.2 Řízení sběru a komunikace požadavků

Dle BABOK začíná sběr a řízení požadavků u identifikace zadavatelů a jejich požadavků. Identifikují se jejich seznamy, role a zodpovědnosti. Následně se řeší sběr požadavků, jejich formalizace a validace napříč všemi zadavateli.

3.1.3 Analýza požadavků

Analýza požadavků dle BABOK slouží zejména k vyhodnocení sebraných požadavků do dostatečné úrovně detailu tak, aby efektivně popisovala požadavek v jeho specifikovaném rozsahu. Zároveň slouží k validaci, že zapsané požadavky popisují požadavky zadavatele a kvalitu jednotlivých požadavků.

3.2 Rešerše možných konceptů

Rešerše konceptů by měla vycházet z koncepcí dnes běžných a osvědčených v oblasti architektury a návrhu informačních systémů. Nejprve si vybereme rámec pro návrh, modelovací nástroje, koncepty komunikace v prostředí IoT a také základních vstupů pro návrh centrálního evidenčního systému. Vzhledem k omezením danými již existujícím prototypy modulu řídící jednotky vozu, která je napájena z baterie a vybavena GSM/GPS moduly, bude rešerše zaměřena zejména na standardy komunikace ISO-OSI přes datové spojení nebo binární SMS komunikaci.

3.3 Definice IoT

Internet věcí (Internet of Things – IoT) je termín používaný organizacemi jako Institute of Electrical and Electronic Engineers (IEEE) i International Telecommunication Union (ITU-T) a je definován jako:

- Síť zařízení s vestavěnými senzory, jež jsou propojeny s Internetem.
- Globální infrastruktura informační společnosti, která umožňuje vznik pokročilých služeb propojováním informačních a komunikačních technologií.

Podle těchto organizací je hlavními vlastnostmi IoT schopnost snímat a komunikovat, při zajištění bezpečí a soukromí (SALAM, 2019). Hlavní elementy IoT jsou:

- Zařízení – jednotka vybavená komunikačními a senzorickými schopnostmi
- Věci – fyzické objekty v reálném světě

3.4 Návrh konceptu

Pro samotné návrhy bude použito formy abstrakce, kterou poskytuje TOGAF, ArchiMate, UML a BPMN. V oblasti technického návrhu se bude práce zabývat modelem ISO-OSI.

3.4.1 TOGAF

TOGAF patří mezi základní rámce v oblasti informační podnikové architektury. Je vyvíjen a udržován v rámci The Open Group již od roku 1995, s původem ve Spojených státech amerických, v oblasti Ministerstva obrany (DoD).

TOGAF se zaměřuje na oblast architektury v prostředí více firem, které mají některé své cíle shodné. Vhodná informační podniková architektura umožňuje optimalizaci různorodých a často neslučitelných procesů do jednotného integrovaného prostředí, s cílem podpořit dosahování cílů firemní obchodní strategie.

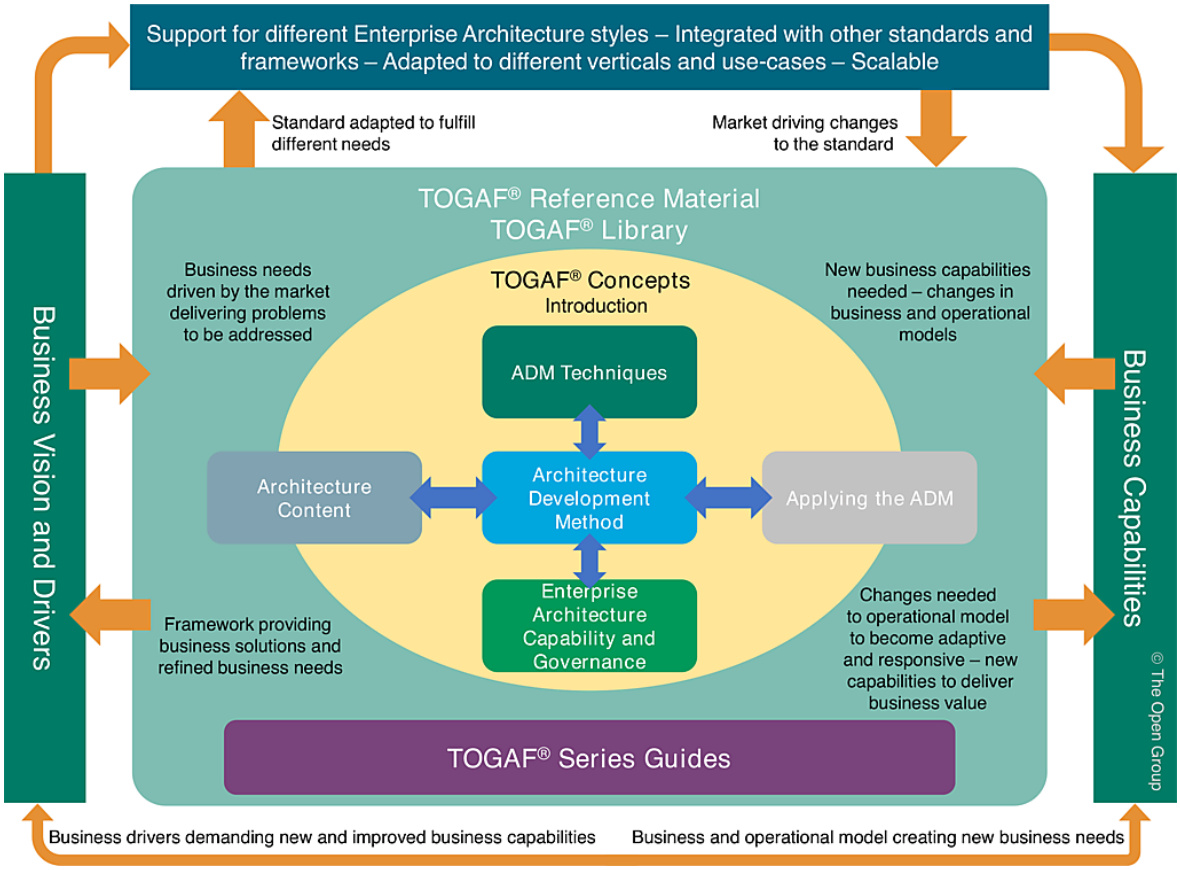
Mezi základní přínosy informační podnikové architektury by se daly vyjmenovat například:

- Efektivnější strategické rozhodování vedení
- Efektivnější provoz
- Snadnější a účelnější digitální transformace
- Lepší návratnost investic a snížení rizika pro budoucí investice
- Rychlejší, snadnější a levnější pořizování nových IT systémů

Architektonický rámec je základní struktura, případně soubor struktur, které mohou být použity pro vývoj dalších konkrétních architektur. Měl by pomoci s popisem současného stavu a cílového stavu ve společnostech pomocí popisu stavebních bloků a jejich plánované interakce. Každý rámec je potřeba přizpůsobit na míru konkrétní skupině organizací.

TOGAF je popsán v šesti základních dokumentech:

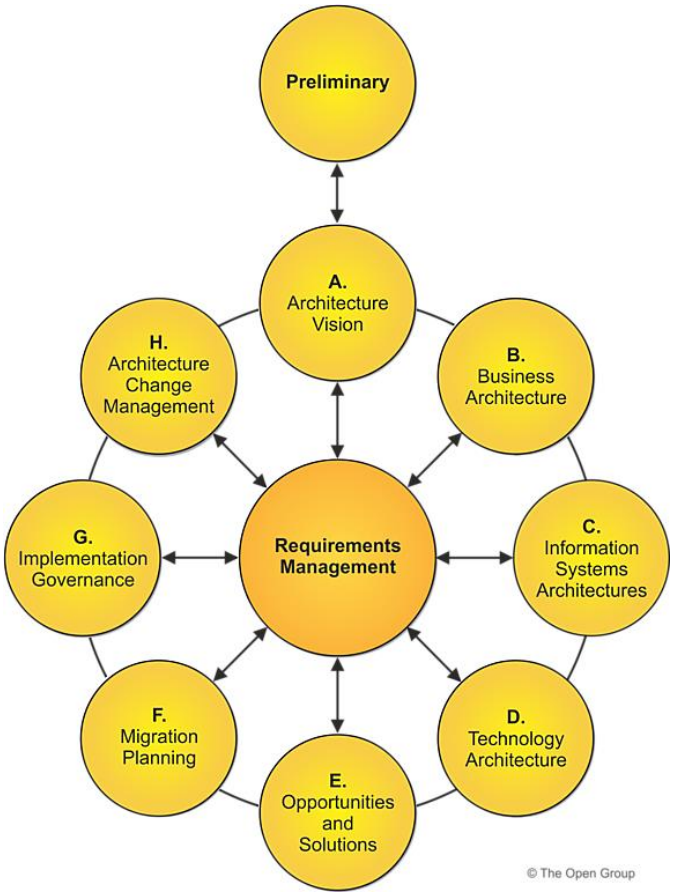
1) Představení a cílové koncepty
2) Architektura vývojových metod
3) Soubor technik TOGAFu pro architektury vývojových metod
4) Aplikace architektonických vývojových metod v praktickém kontextu
5) Architektonický obsah – popis artefaktů a základních stavebních bloků
6) Možnosti a řízení informační podnikové architektury



Obrázek 1 Struktura TOGAFu, (zdroj: The Open Group)

3.4.1.1 Architektura vývojových metod (ADM)

Architektura vývojových metod (ADM cyklus) popisuje metody pro vývoj a řízení životního cyklu v rámci podnikové informační architektury a je základem TOGAF rámce. V průběhu ADM cyklu je potřeba pravidelně ověřovat výsledky oproti očekáváním, a to jak pro celý ADM cyklus, tak i pro každou fázi procesu. Pro potřeby této práce budeme pracovat s fázemi A až D.



Obrázek 2 ADM - Architecture Development Method, (zdroj: The Open Group)

Preliminary - předběžná fáze má za cíle zejména:

- Zjistit možnosti požadované organizací – kontext, rozsah a jednotlivé prvky podnikové informační architektury (EA).
- Identifikovat a ohraničit prvky organizace, kterých se EA dotkne.
- Vybrat a nastavit rámce, metody a procesy, které jsou ovlivněny EA.
- Nastavit cíle.

A. Architektonická vize má za cíle zejména:

- Vytvořit rámcovou vizi schopností a obchodních hodnot, které vzniknou v rámci EA.
- Získat potvrzení k seznamu prací (Statement of Work - SoW), které budou potřeba k dosažení této vize.

B. Obchodní architektura má za cíle zejména:

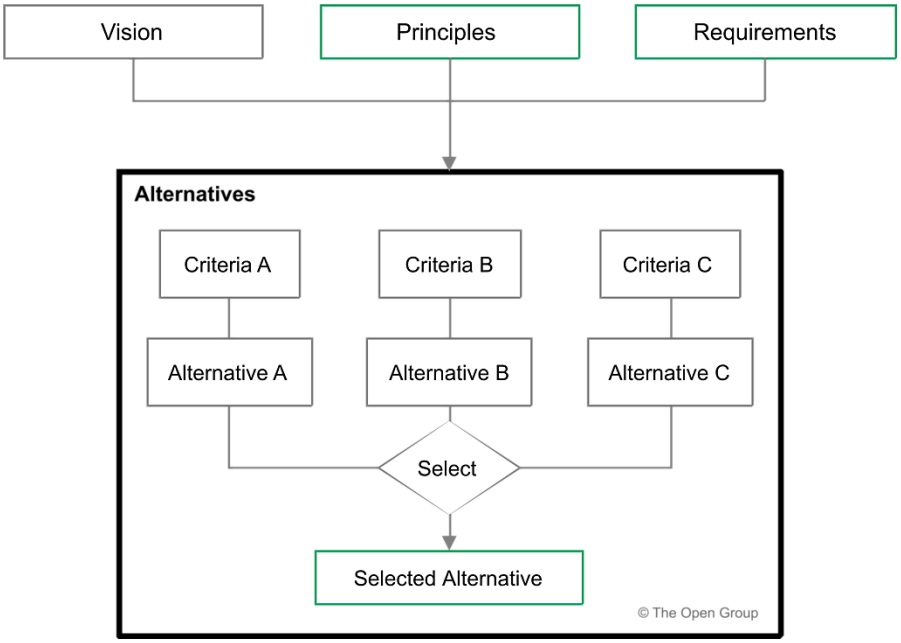
- Vytvořit cílovou obchodní architekturu a popsat, jak organizace funguje při zajišťování svých obchodních cílů.

- Odpovídá na strategické cíle v architektonické vizi a ukotvuje seznam prací (SoW) a jednotlivé účastníky (stakeholders).
- Identifikuje kandidáty pro hlavní komponenty cílové architektonické vize jako rozdíl mezi základní a cílovou obchodní vizí.

C. Architektura informačních systémů má za cíle zejména:
   - Vytvořit cílovou architekturu informačních systémů popisující, jak tato architektura umožní dosáhnout obchodních cílů a v rámci definovaného seznamu prací (SoW) způsobem, který je pro jednotlivé účastníky srozumitelný a přijatelný.
   - Identifikuje bloky funkcionalit jako vstup pro plánování vývoje tak, aby pokrýval mezery mezi současným a cílovým stavem architektury.
D. Architektura technologií má za cíle zejména:
   - Vyvinout cílovou technologickou architekturu, která umožní dosáhnout cílové vize, společně se stavebními bloky, jež budou poskytovat technologické komponenty a služby tak, aby došlo k zajištění SoW a potřeb jednotlivých účastníků.
   - Identifikovat kandidáty na komponenty, které zaplní mezery mezi současnou a cílovou architekturou.
E. Příležitosti a řešení mají za cíle zejména:
   - Vytvořit úvodní verzi kompletního architektonického plánu založeného na analýze chybějících komponent z fáze B, C a D.
   - Rozhodnout, zda je vhodné postupovat po jednotlivých krocích, a pokud ano, navrhnout přechodnou architekturu, která bude průběžně dodávat obchodní hodnotu.
   - Definovat cílové stavební bloky a finalizovat cílovou architekturu.

3.4.1.2 Metody pro výběr alternativ

Pro výběr a návrh konkrétní alternativy se budeme řídit pomocí metodologie ADM, která kombinuje vizi, pravidla a požadavky. Z nich vycházejí možné alternativy řešení a v rámci hodnocení je pak vybrána ta nejvhodnější.



Obrázek 3 ADM Metody, (zdroj: The Open Group)

3.4.2 ArchiMate Enterprise Architecture modeling language

ArchiMate Enterprise Architecture modeling language představuje standard The Open Group, jehož cílem je vytvořit vizuální nástroj pro popis, analýzu a sdílení potřeb a problémů, které vznikají v podnikové informační architektuře a které se také v průběhu času mění.

ArchiMate poskytuje rámec pro zobrazení a popis podnikové informační architektury a jejich změn. Součástí jsou nástroje pro vizualizaci a specifikaci závislostí, bodů pohledu jednotlivých účastníků (stakeholders) a popis nástrojů pro přizpůsobování. ArchiMate je založen na pohledech přes služby, aplikace a technologie (včetně fyzických řešení).

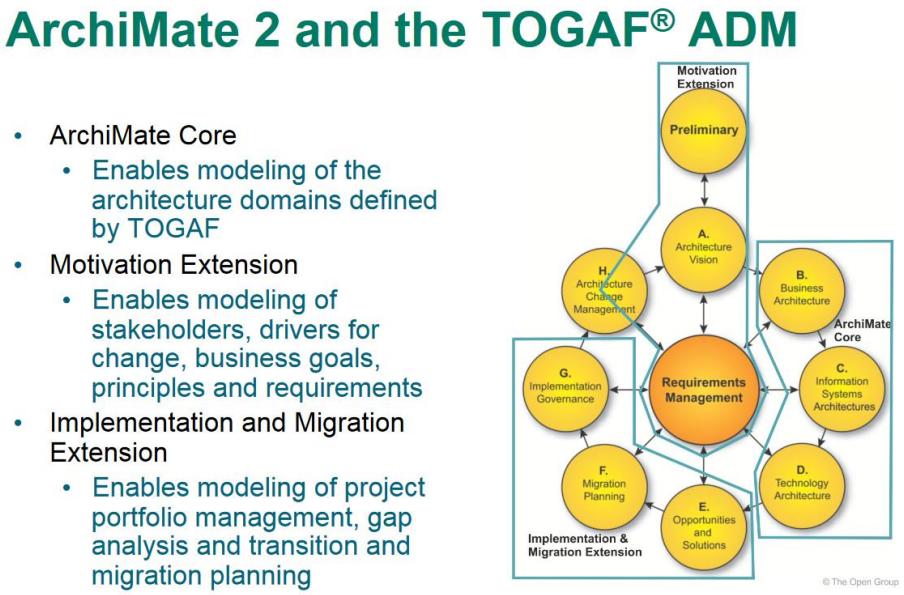
ArchiMate 2.1 je složen z několika částí:

- ArchiMate CORE – umožňuje modelovat architekturu jednotlivých domén definovanou pomocí TOGAFu.
- Motivation Extension – umožňuje modelovat účastníky, hybatele změn, obchodní cíle, principy a požadavky.
- Implementation and Migration Extension – umožňuje modelovat projektové portfolio, analýzu rozdílů mezi jednotlivými stavy a plánovat přechod a migrace.

3.4.2.1 ArchiMate a TOGAF

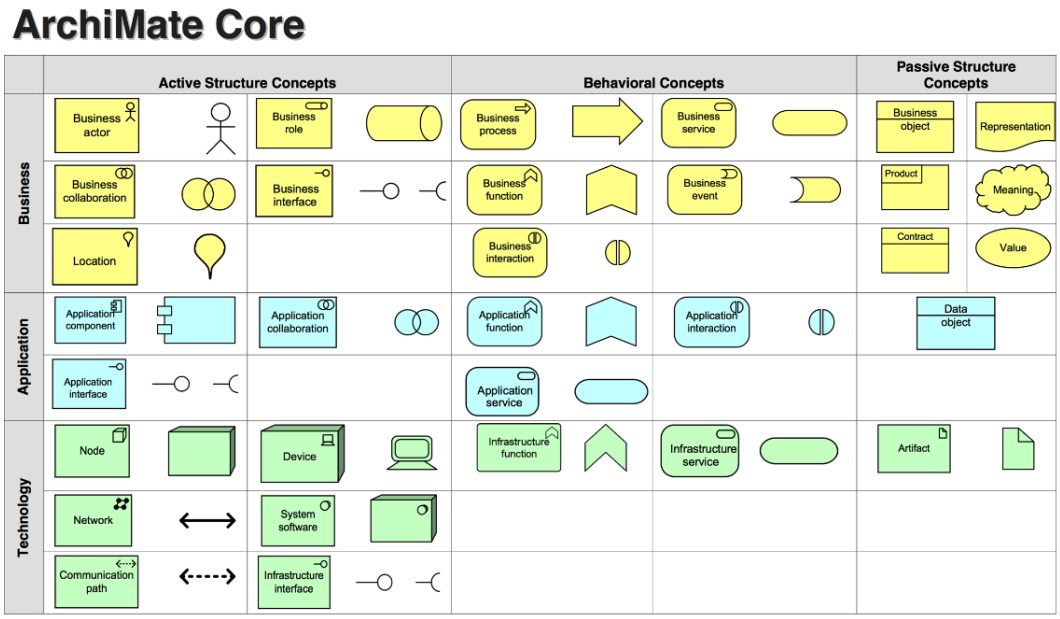
ArchiMate je navázán na TOGAF a adresuje jednotlivé fáze z cyklu ADM.

- ArchiMate CORE adresuje fáze B-D z TOGAF ADM.
- Implementation and Migration Extension adresuje fáze E-G z cyklu TOGAF ADM.
- Motivation Extension adresuje zbytek cyklu TOGAF ADM.



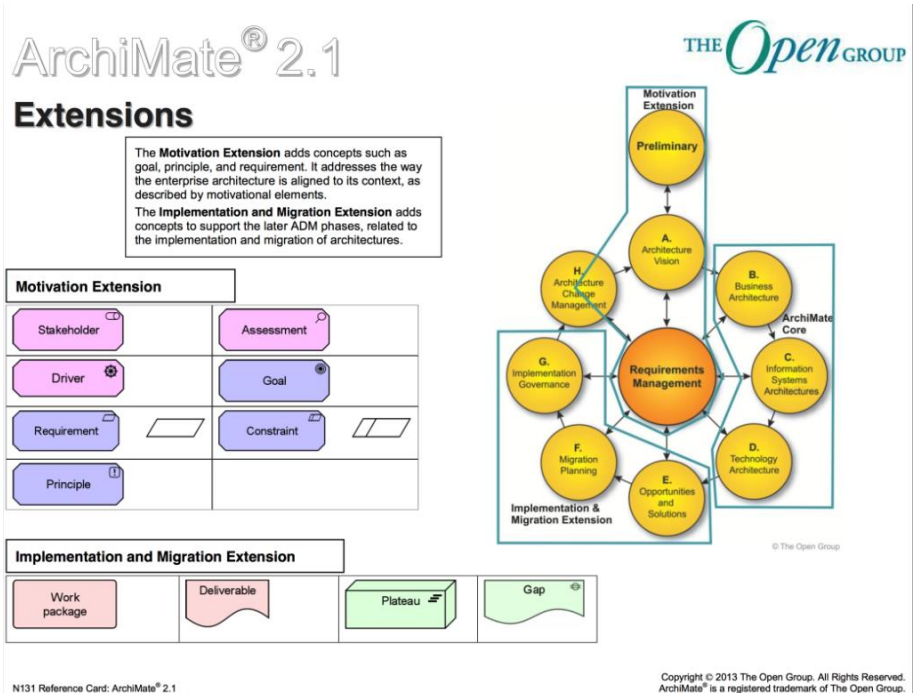
Obrázek 4 Pokrytí fází TOGAF pomocí ArchiMate, (zdroj: The Open Group)

3.4.2.2 ArchiMate Core



Obrázek 5 ArchiMate Core, (zdroj: The Open Group)

3.4.2.3 ArchiMate Extensions



Obrázek 6 ArchiMate a jeho rozšíření, (zdroj: The Open Group)

3.4.3 UML

UML znamená Unified Modelling Language a jde o modelovací jazyk (FOWLER, 2004) používaný pro vizualizaci, specifikaci, návrh a dokumentaci zejména informačních systémů. Je od roku 2005 standardizován ISO standard (konkrétně ISO/IEC 19505-1:2012) a vyvíjen v rámci Object Management Group.

UML umožňuje popisovat:

- způsoby užití (use case)
- aktivity
- jednotlivé komponenty systému
- jak systém funguje
- jak jednotlivé části reagují na ostatní
- rozhraní systému

UML bylo původně zamýšleno jako nástroj pro popis objektově orientovaných systémů, nicméně jeho využití se rozšířilo s přesahem do mnoha dalších oblastí návrhu a vývoje. Jeho využití je zejména v detailnějším popisu konkrétního systému a jeho chování (v porovnáním například s ArchiMate).
UML umožňuje modelovat statické i dynamické pohledy.

3.4.3.1 Statické pohledy

- diagramy užití
- diagramy tříd
- diagramy komponent
- objektové komponenty
- relace mezi komponenty
- a další

3.4.3.2 Dynamické pohledy

- diagram aktivit
- diagram interakcí
- diagram stavů
- diagram použití

- a další

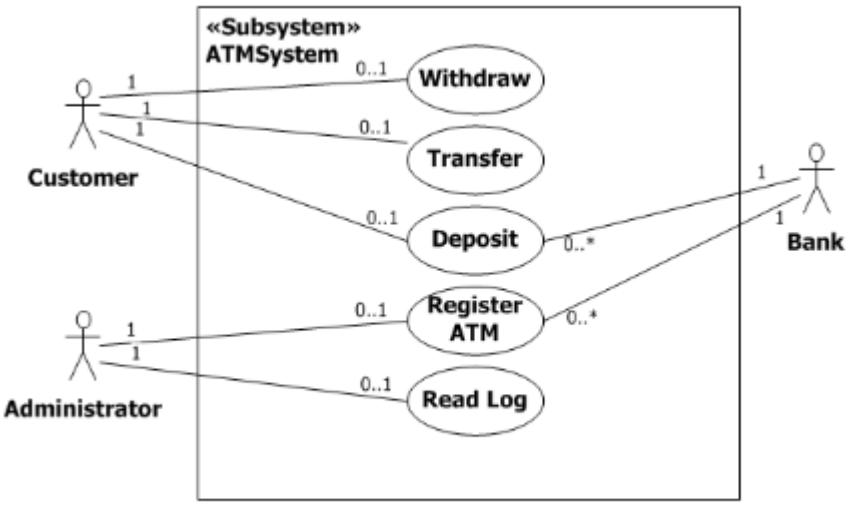


Figure 18.10 Example ATM system with UseCases and Actors

-

Obrázek 7 Ukázkový diagram užití UML, (zdroj: Open Management Group)

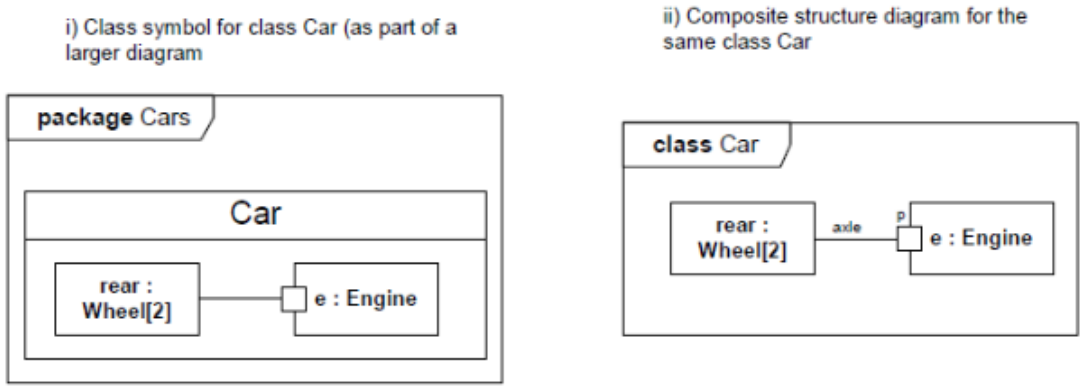


Figure A.4 A class diagram and a composite structure diagram

Obrázek 8 Ukázkový diagram tříd UML, (zdroj: Open Management Group)

3.4.4 Business Process Model and Notation

Business Process Model and Notation (BPMN) je grafická notace sloužící k modelování procesů pomocí procesních diagramů. Je standardizován ISO standard (konkrétně ISO/IEC 19510 -2013) a vyvíjen od roku 2005 v rámci Object Management Group.

Primárním cílem BPMN je poskytnout notaci, která je snadno srozumitelná všem uživatelům – od analytiků, kteří navrhují proces, přes pro vývojáře, kteří proces implementují, a také pro obchodní uživatele, kteří řídí a monitorují tyto procesy. BPMN vytváří most mezi obchodní stranou procesu a jeho technickou implementací. Umožňuje vytvářet diagramy spolupráce, procesní diagramy a diagramy choreografie (posloupností) a další.

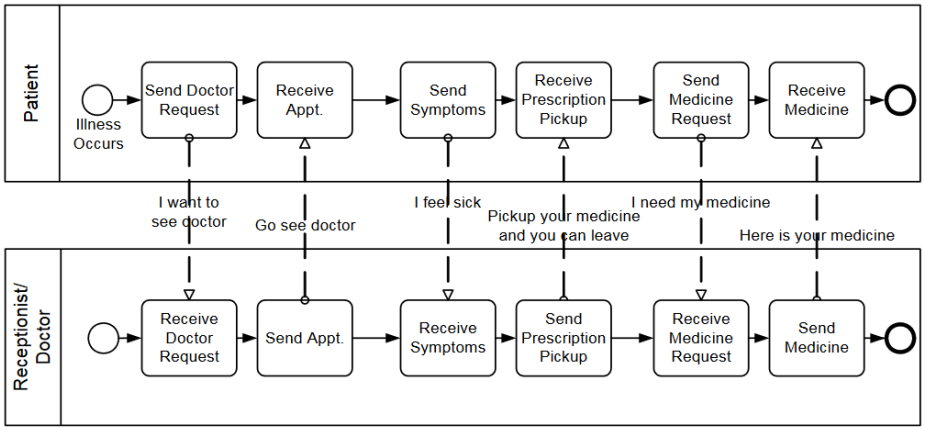
3.4.4.1 BPMN Elementy

- Objekty toku
    - události
    - aktivity
    - rozdělovače
- Data
    - datové objekty
    - datové vstupy
    - datové výstupy
    - datová úložiště
- Spojnice objektů
    - sekvenční toky
    - toky zpráv
    - asociace
    - datové asociace
- Plavecké dráhy
    - bazény
    - dráhy
- Artefakty
    - skupiny

- textové poznámky

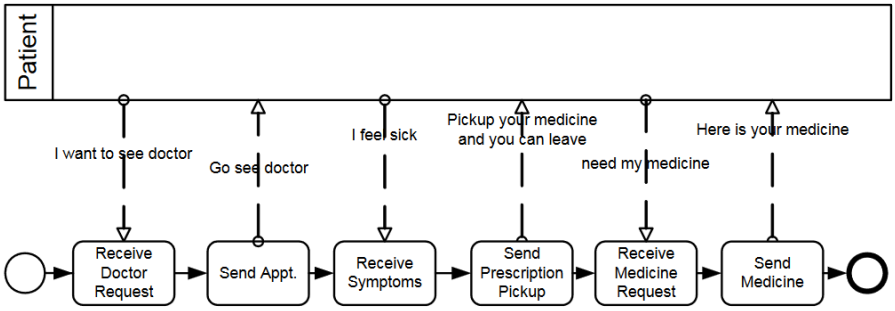
3.4.4.2 Ukázkové diagramy

Diagram spolupráce



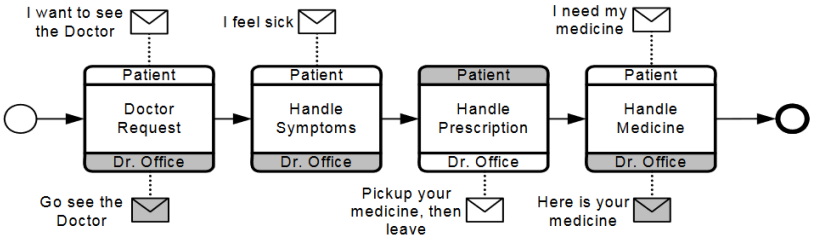
Obrázek 9 Ukázka diagramu spolupráce, (zdroj: Open Management Group)

Procesní diagram



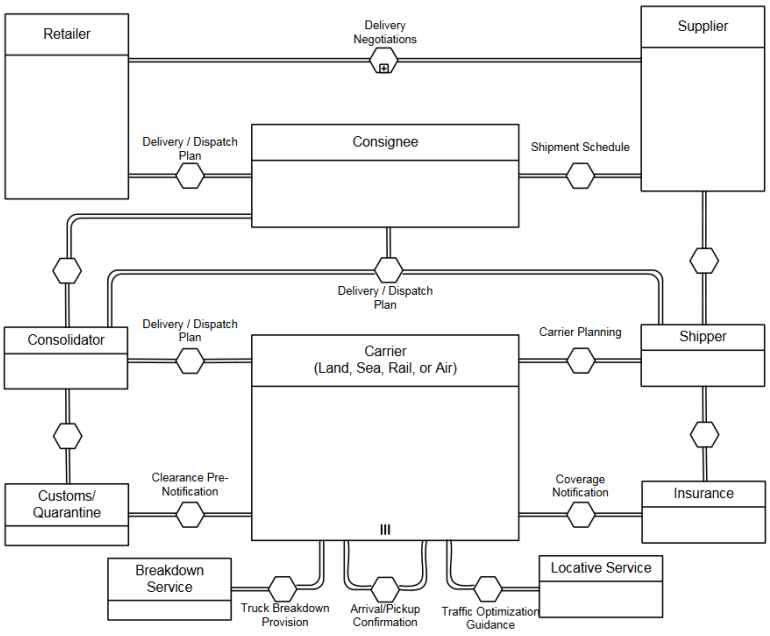
Obrázek 10 Ukázkova veřejného procesu, (zdroj: Open Management Group)

Diagram choreografie



Obrázek 11 Ukázka diagramu choreografie, (zdroj: Open Management Group)

Diagram konverzace



Obrázek 12 Ukázka diagramu konverzace, (zdroj: Open Management Group)

3.4.5 ISO-OSI model

Podle ISO-OSI (International Organization for Standardization – Open Systems Interconnection) je referenční model pro komunikaci mezi jednotlivými zařízeními propojených do sítě.

ISO-OSI model rozděluje sedm vrstev v modelu:

1) Fyzická vrstva (Physical layer) – definuje fyzické vlastnosti přenosového média, jako jsou elektrické signály, optické signály nebo bezdrátové vysílání. Zajišťuje fyzický přenos mezi zařízeními.
2) Linková vrstva (Data link layer) – zajišťuje přenos mezi sousedními síťovými zařízeními a řídí tok dat na fyzické vrstvě. Zajišťuje detekci a opravu chyb při přenosu dat.
3) Síťová vrstva (Network layer) – umožňuje komunikaci mezi sítěmi a zajišťuje směrování a přeposílání dat přes různé síťové prvky (routery atp).
4) Transportní vrstva (Transport layer) – řídí tok mezi koncovými body a zajišťuje, aby data byla doručena ve stejném pořadí, v jakém byla odeslána. Poskytuje mechanismy pro řízení rychlosti přenosu a zabezpečení proti ztrátě dat.
5) Relační vrstva (Session layer) – zajišťuje navázání, udržení a ukončení relace mezi koncovými body. Poskytuje mechanismy pro synchronizaci a řízení dialogu mezi aplikacemi.
6) Prezentační vrstva (Presention layer) – zajišťuje správné formátování a kódování dat pro přenos mezi aplikacemi na různých koncových bodech. Zajišťuje kompatibilitu mezi různými formáty dat.
7) Aplikační vrstva (Application layer) – poskytuje rozhraní pro aplikace, které komunikují prostřednictvím sítě. Zahrnuje aplikace jako elektronická pošta, prohlížeč webových stránek a další.

3.4.6 Transportní vrstva síťové komunikace

Vzhledem k rozhodnutí zadavatele využít pro komunikaci datový modul GSM lze pro komunikaci využít zejména následující způsoby:

3.4.6.1 SMS (Short Message Service)

GSM 03.40 a 3GPP TS 23.040 je standard popisující formát protokolových datových jednotek protokolu Short Message Transfer Protocol (SM-TP) používaných pro přenos dat v sítích GSM. Umožňuje přenos 1120 bitů v těle zprávy. Je k dispozici na všech typech mobilních datových sítí (2G/3G/4G/5G).

3.4.6.2 SMPP (Short Message Peer to Peer)

SMPP je otevřený průmyslový standard pro přenos krátkých textových zpráv pomocí TCP protokolu, fungující na transportní (čtvrté) vrstvě modelu OSI. Používán je zejména pro serverové aplikace, které potřebují komunikovat s jednotlivými uživateli GSM sítě.

3.4.6.3 HTTP/JSON/REST

Tato kombinace protokolu (HTTP), formátu (JSON) a architektury (REST) je v současné době de facto standardem pro webové služby a aplikace. Jsou standardizovány skrze různé organizace a skupiny, jako například Internet Engineering Task Force (IETF), World Wide Web Consortium (W3C) nebo JSON-RPC Working Group. Tyto organizace definují specifikace a doporučení pro použití těchto standard tak, aby byla zajištěna interoperabilita mezi různými implementacemi. Standardy jsou často aktualizovány a upravovány v závislosti na vývoji technologií a potřebách uživatelů.

1) HTTP (Hypertext Transfer Protocol) se používá pro komunikaci mezi klientem a serverem při požadavcích na zdroje poskytované serverem. HTTP standardizuje způsob, jakým jsou požadavky a odpovědi na tyto požadavky vyměňovány.
2) REST (Representional State Transfer) je architektura pro návrh webových služeb, která využívá HTTP pro přenos dat. REST definuje způsob, jakým jsou zdroje identifikovány a jak jsou dostupné pomocí standardních HTTP požadavků.
3) JSON je formát datový formát používaný pro výměnu dat mezi klientem a serverem. JSON definuje způsob, jakým jsou data strukturována a jak jsou kódována pro přenos.

3.4.7 Protokoly aplikační vrstvy

Pro SMS komunikaci by bylo vhodné použít proprietární binární protokol, který maximálně využije omezenou datovou kapacitu jedné SMS zprávy. Pro mobilní datovou komunikaci lze využít moderní protokoly jako HTTP.

Pro SMS i mobilní datovou komunikaci lze zvažovat například:

- MQTT (Message Queuing Telemetry Transport) - je lehký a škálovatelný protokol pro IoT (Internet of Things) komunikaci, který umožňuje komunikaci mezi zařízeními a centrálním serverem.
- LwM2M (Lightweight Machine-to-Machine) – je protokol pro správu a komunikaci IoT zařízení s centrálním systémem – platformou.

Pro mobilní datovou komunikaci lze zvažovat také:

- HTTP (Hypertext Transfer Protocol) - je standardní protokol pro webovou komunikaci. Lze ho použít pro odesílání dat z lokálních senzorů do centrálního serveru.
- CoAP (Constrained Application Protocol) - je protokol pro komunikaci IoT zařízení s omezenou kapacitou, jako jsou senzory a aktuátory, přes IP sítě.

3.4.7.1 CoAP

CoAP se profiluje jako protokol se zaměřením na přenos dat mezi jednotkami s limitovanými komunikačními a výpočetními možnostmi v prostředích, kde existuje omezení z hlediska napájení a ztrátové komunikaci na síti (RFC 7252). Komunikační jednotky mají malé kapacity ROM a RAM pamětí a protokol je vytvořen s ohledem na komunikaci mezi zařízeními - M2M (Machine-to-Machine), jako jsou chytré budovy a automatizované jednotky.

CoAP poskytuje funkcionalitu dotaz/odpověď mezi aplikačními koncovými body s nízkou úrovní režie, s důrazem na provoz v prostředích s omezeným výpočetním výkonem a limitem v komunikaci.

CoAP cílí na realizaci REST architektury v (na zdroje omezeném) IoT prostředí.

Hlavní vlastnosti CoAP

CoAP má následující vlastnosti:

- Web protokol pro M2M požadavky v IoT omezeném prostředí

- Podpora UDP
- Asynchronní výměna zpráv
- Nízká režie
- podpora URI Content-type
- Bezstavové mapování HTTP
- Schopnost podpory Datagram Transport Layer Security (DTLS)
- Podpora potvrzovaných i nepotvrzovaných zpráv

Na rozdíl od HTTP, podporuje CoAP výměnu asynchronně pomocí datagramového transportu, jako je UDP. Toho je dosaženo pomocí vrstvy podporující volitelnou spolehlivost.

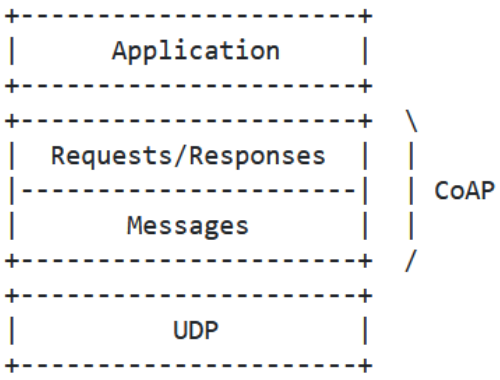


Figure 1: Abstract Layering of CoAP

Obrázek 13 Abstraktní vrstvy CoAP, (zdroj: IETF RFC 7252)

Typy zpráv CoAP

CoAP definuje čtyři typy zpráv:

- Potvrzované

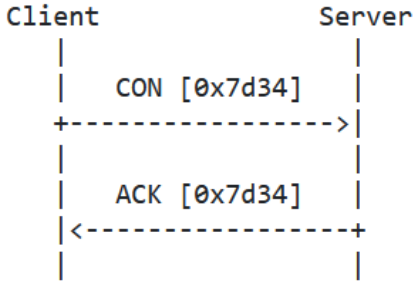


Figure 2: Reliable Message Transmission

Obrázek 14 Potvrzovaný přenos zpráv, (zdroj: IETF RFC 7252)

- Nepotvrzované

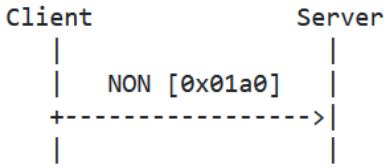


Figure 3: Unreliable Message Transmission

Obrázek 15 Nepotvrzovaný přenos zpráv, (zdroj: IETF RFC 7252)

- Potvrzení
- Reset

Model Request/Response CoAP

Pokud je server schopen odpovědět ihned, může být součástí potvrzení již také odpověď (Piggyback):

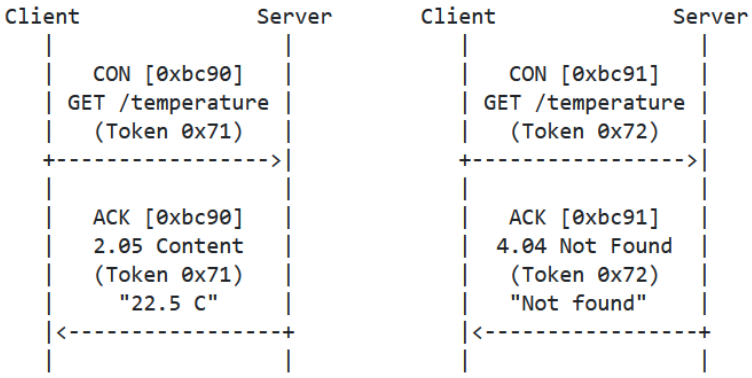


Figure 4: Two GET Requests with Piggybacked Responses

Obrázek 16 Ukázka požadavků s odpovědí, (zdroj: IETF RFC 7252)

Pokud server není schopen okamžitě odeslat odpověď, může server zaslat separátní odpověď později (ve variantě s vyžádaným potvrzením):

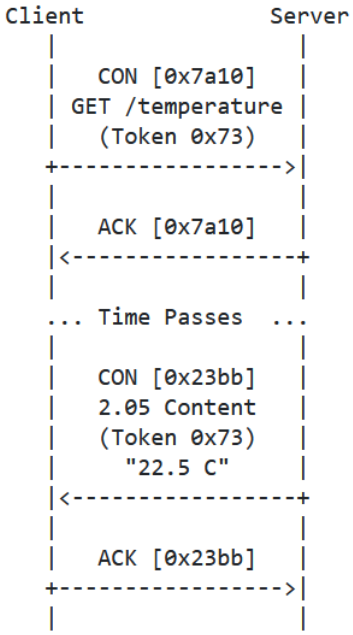


Figure 5: A GET Request with a Separate Response

Obrázek 17 Dotaz se zpožděnou odpovědí a potvrzením, (zdroj: IETF RFC 7252)

Nebo ve variantě bez potvrzení:

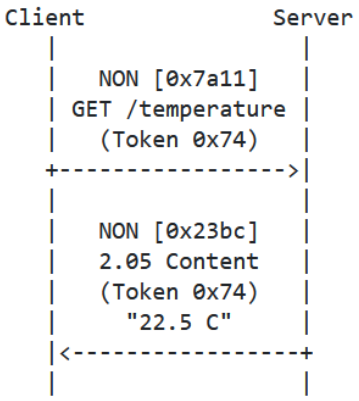


Figure 6: A Request and a Response Carried in Non-confirmable Messages

Obrázek 18 Požadavek a zpožděná odpověď bez potvrzení, (zdroj: IETF RFC 7252)

Metody CoAP

CoAP používá metody GET, PUT, POST a DELETE v podobné formě jako HTTP. Je založen na výměně kompaktních zpráv, které jsou defaultně zasílány přes UDP. Pro potřeby komunikace telemetrických datových jednotek je však zásadní, že lze také použít i jiné transportní vrstvy, jako je SMS, TCP nebo SCTP. Tato vlastnost umožňuje systému přecházet bez dodatečné implementace mezi transportní vrstvou řešenou

pomocí SMS (tam, kde to má smysl nebo i tam, kde není jiná možnost komunikace) a mobilního datového spojení (UDP či TCP).

Code	Name	Reference
0.01	GET	[RFC7252]
0.02	POST	[RFC7252]
0.03	PUT	[RFC7252]
0.04	DELETE	[RFC7252]

Table 5: CoAP Method Codes

Obrázek 19 Kódy metody CoAP , (zdroj: IETF RFC 7252)

Formát zprávy CoAP

Zprávy jsou kódované v jednoduchém binárním formátu.

- Hlavička (vždy, 4 byte)
- Token (variabilní délka, 0-8 byte)
- Options in Type-Lenght-Value (TLV) formátu (0 a více)
- Payload (po zbytek datagramu)

CoAP zpráva vypadá následovně:

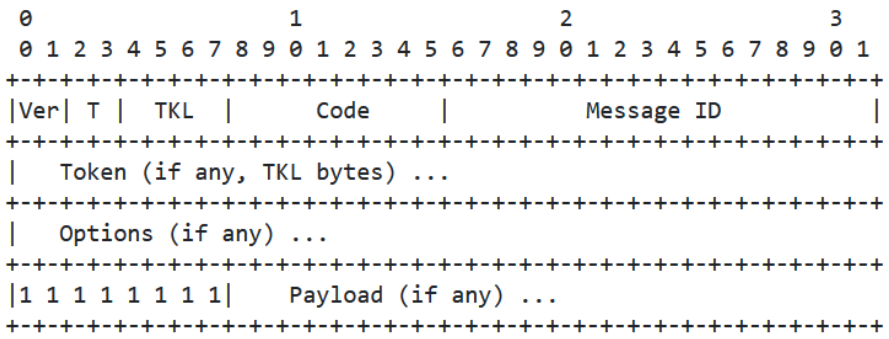


Figure 7: Message Format

Obrázek 20 Formát CoAP zprávy, (zdroj: IETF RFC 7252)

Popis CoAP zprávy:

- Version CoAP (Ver) – 2-bit integer
- Type (T): 2-bit integer. Potvrzovaný 0, Nepotvrzovaný (1), Potvrzení (2) nebo Reset (3)
- Token Length (TKL) – 4-bit integer – definuje délku tokenu (0-8 bytů)

- Code – 8-bit integer – definuje typ požadavku: dotaz, odpověď v pořádku, klientská chybná odpověď, serverová chybná odpověď, případně prázdná zpráva
- Message ID – 16 bit integer pro sledování pořadí zpráv
- Token - po hlavičce následuje Token value (0-8 bytů), dle specifikace nastavení v TKL, token se využívá na párování požadavků a odpovědí.
- Options – (volitelný, nula nebo více výskytů) – specifikuje parametry požadavku

Number	Name	Reference
0	(Reserved)	[RFC7252]
1	If-Match	[RFC7252]
3	Uri-Host	[RFC7252]
4	ETag	[RFC7252]
5	If-None-Match	[RFC7252]
7	Uri-Port	[RFC7252]
8	Location-Path	[RFC7252]
11	Uri-Path	[RFC7252]
12	Content-Format	[RFC7252]
14	Max-Age	[RFC7252]
15	Uri-Query	[RFC7252]
17	Accept	[RFC7252]
20	Location-Query	[RFC7252]
35	Proxy-Uri	[RFC7252]
39	Proxy-Scheme	[RFC7252]
60	Size1	[RFC7252]
128	(Reserved)	[RFC7252]
132	(Reserved)	[RFC7252]
136	(Reserved)	[RFC7252]
140	(Reserved)	[RFC7252]

Table 7: CoAP Option Numbers

Obrázek 21 Základní registr CoAP Options, (zdroj: IETF RFC 7252)

- Payload – (volitelný), pokud nenulový, je uveden jedním bytem Payload-marker (0xFF), který indikuje konec Options a začátek Payload. Payload pokračuje od markeru do konce velikosti datagramu. Délka Payload je vypočtena z délky datagramu.

Typy obsahu (CoAP Payload)

CoAP podporuje rozdílné typy obsahu - textové, binární, JSON, xml a další. Vzhledem k minimalizaci režie jsou tyto definovány již ve standardu. Pro použití v rámci

komunikačního protokolu mezi telemetrickou datovou jednotkou a centrálním systémem lze zvolit buď binární octet-stream (zejména pokud se bude uvažovat o možnosti komunikovat pomocí SMS transportu) nebo json (pokud budou hlavním komunikačním kanálem mobilní data).

Media type	Encoding	ID	Reference
text/plain; charset=utf-8	-	0	[RFC2046] [RFC3676] [RFC5147]
application/link-format	-	40	[RFC6690]
application/xml	-	41	[RFC3023]
application/octet-stream	-	42	[RFC2045] [RFC2046]
application/exi	-	47	[REC-exi-20140211]
application/json	-	50	[RFC7159]

Table 9: CoAP Content-Formats

Obrázek 22 Typy obsahu CoAP, (zdroj: IETF RFC 7252)

Kódy odpovědí CoAP

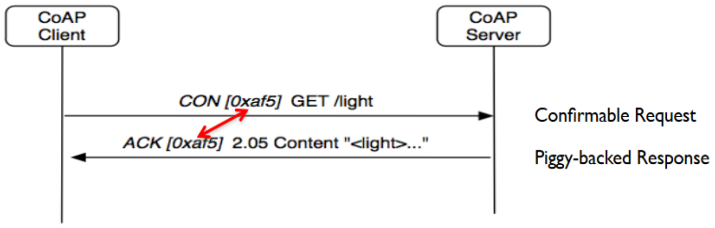
Obdobně jako u HTTP jsou u CoAP definovány kódy odpovědí. Z důvodu minimalizace režie protokolu jsou opět řešeny referenční tabulkou:

Code	Description	Reference
2.01	Created	[RFC7252]
2.02	Deleted	[RFC7252]
2.03	Valid	[RFC7252]
2.04	Changed	[RFC7252]
2.05	Content	[RFC7252]
4.00	Bad Request	[RFC7252]
4.01	Unauthorized	[RFC7252]
4.02	Bad Option	[RFC7252]
4.03	Forbidden	[RFC7252]
4.04	Not Found	[RFC7252]
4.05	Method Not Allowed	[RFC7252]
4.06	Not Acceptable	[RFC7252]
4.12	Precondition Failed	[RFC7252]
4.13	Request Entity Too Large	[RFC7252]
4.15	Unsupported Content-Format	[RFC7252]
5.00	Internal Server Error	[RFC7252]
5.01	Not Implemented	[RFC7252]
5.02	Bad Gateway	[RFC7252]
5.03	Service Unavailable	[RFC7252]
5.04	Gateway Timeout	[RFC7252]
5.05	Proxying Not Supported	[RFC7252]

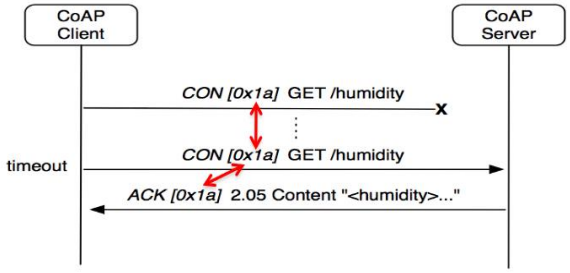
Table 6: CoAP Response Codes

Obrázek 23 CoAP kódy odpovědí, (zdroj: IETF RFC 7252)

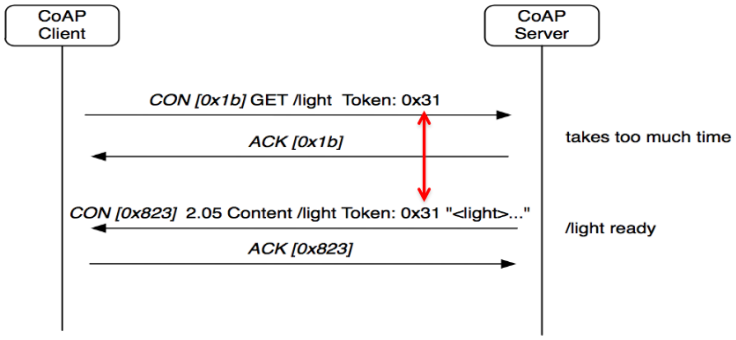
Ukázky protokolu CoAP



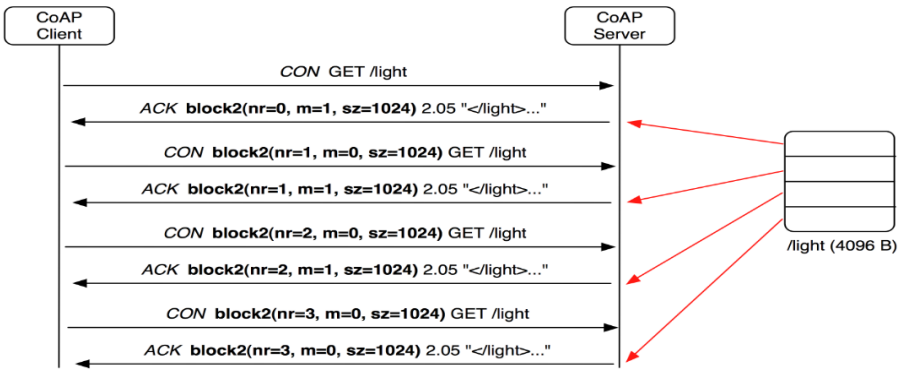
Obrázek 24 Ukázka požadavku, (zdroj: SHELBY, Z.)



Obrázek 25 Ukázka řešení ztráty packetu, (zdroj: SHELBY, Z.)



Obrázek 26 Ukázka separátní odpovědi, (zdroj: SHELBY, Z.)



Obrázek 27 Ukázka blokového transferu dat, (zdroj: SHELBY, Z.)

Aktualizace firmware telemetrických komunikačních jednotek

Firmware je možné taktéž distribuovat pomocí CoAP protokolu. Lze využít metodu PUT, která umožňuje aktualizovat zdrojový kód zařízení, tedy také firmware. Tím je umožněno aktualizovat firmware bez nutnosti připojení k síti pomocí kabelu nebo manuálního přeprogramování zařízení.

Z hlediska bezpečnosti je vhodné zvážit specifický PreSharedKey, kterým se bude pouze ověřovat konzistence a zdroj aktualizovaného firmware pro každou jednotku zvlášť.

Bezpečnost CoAP

CoAP podporuje několik způsobů zabezpečení:

- NoSec – DTLS zakázáno
- PreSharedKey – DTLS povoleno, existuje seznam sdílených klíčů a jednotek, které je mohou použít
- RawPublicKey – DTLS je povoleno a jednotka má asymetrický pár klíčů (bez certifikátu)
- Certificate: DTLS je povoleno a zařízení má asymetrický pár klíčů ve formě X.509 certifikátu

DTLS je obdoba TLS protokolu HTTP pro CoAP, definovaná v RFC 6347.

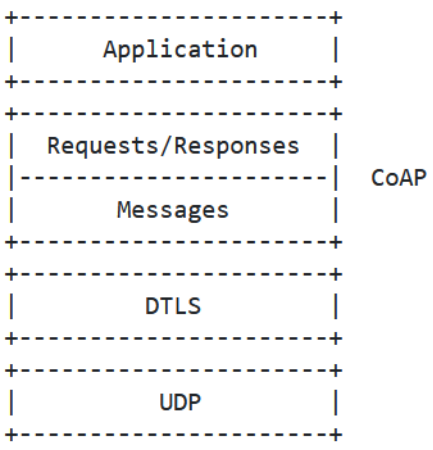


Figure 13: Abstract Layering of DTLS-Secured CoAP

Obrázek 28 Vrstvy CoAP zabezpečeného pomocí DTLS, (zdroj: IETF RFC 7252)

Pro zabezpečení komunikace by bylo z hlediska bezpečnostní architektury (GUTMANN, 2019) vhodnější využít certifikáty, nicméně vzhledem ke komplexnosti takového systému (nezbytnosti zajistit pro takový systém kompletní PKI

infrastrukturu včetně všech relevantních procesů) bude vhodné spíše použít snáze implementovatelnou variantu PreSharedKey.

3.4.8 Centrální systém

Vzhledem k tomu, že se očekává průběžný růst počtu monitorovaných vozů i klientských přístupů k datům o nich, je z hlediska pružnosti a škálovatelnosti vhodné zvážit nasazení v Cloud infrastruktuře. V systému by se neměla nacházet citlivá či osobní data. Z tohoto důvodu lze bez problému doporučit hlavní poskytovatele těchto služeb:

1) Amazon Web Services (AWS)
2) Google Cloud Platform (GCP)
3) Microsoft Azure

Z technologií pro vývoj převážně HTTP/JSON/REST aplikací lze zvažovat více rámců a technologií. V současné době lze zařadit mezi nejpoužívanějších například tyto:

1) Node.js – je technologie s otevřeným zdrojovým kódem, multiplatformní a běhové prostředí pro vývoj serverových aplikací v JavaScriptu.
2) Spring Boot – je technologie s otevřeným kódem pro vývoj Java REST mikroslužeb a aplikací.
3) Django – je technologie s otevřeným kódem pro vývoj webových aplikací v jazyce Python. Obsahuje mnoho knihoven a nástrojů pro vývoj mikroslužeb včetně Django REST frameworku.
4) Ruby on Rails – je technologie s otevřeným kódem pro vývoj webových aplikací v jazyce Ruby. Obsahuje mnoho knihoven a nástrojů pro vývoj mikroslužeb.
5) Flask - je technologie s otevřeným kódem pro vývoj webových aplikací v jazyce Python.

4 Výsledky práce

4.1 Návrh koncepce

Návrh koncepce (TOGAF – Preliminary) říká, že je nejprve potřeba definovat:

- Kontext, rozsah a prvky.
- Identifikovat organizace, kterých se architektura dotkne.
- Vybrat a nastavit rámce, metody a procesy, které jsou ovlivněny EA.
- Nastavit cíle.

4.1.1 Identifikace zainteresovaných účastníků

4.1.1.1 Provozovatel systému

Provozovatel systému má na starosti vývoj a provoz systému, jeho obchodní i technickou stránku. Sbírá požadavky od zákazníků, vyvíjí a vyrábí komunikační moduly, zabezpečuje datové sítě pro přenos informací, vyvíjí a provozuje centrální systém.

4.1.1.2 Provozovatel vozu

Provozovatel vozu ze systému získává zejména telemetrické informace k opotřebení jednotlivých komponent vozu tak, aby mohl efektivně plánovat servisní zásahy. V současné době se počítá zejména se servisem brzdné soustavy. Účastníka lze označit jako „zákazníka“ z pohledu obchodního modelu.

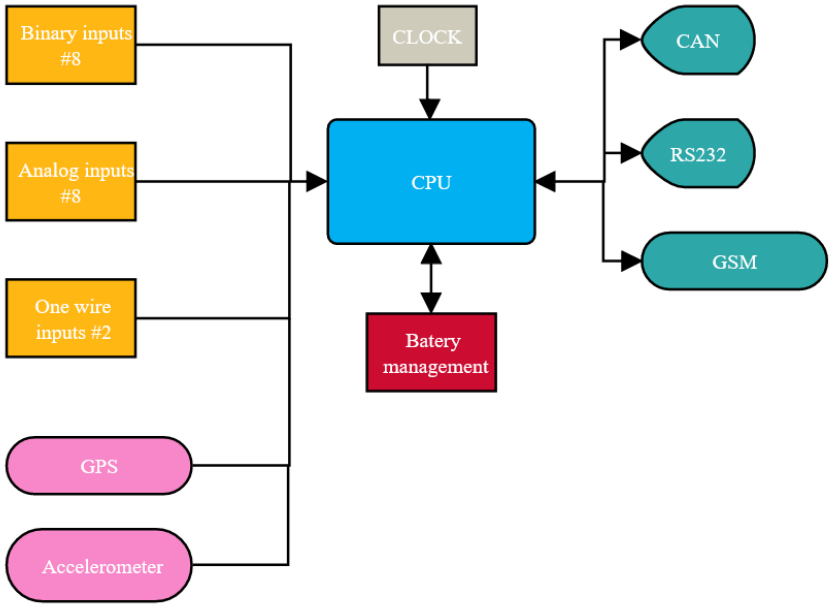
4.1.1.3 Dopravce

Dopravce ze systému získává zejména informace o lokaci vozu, o provozních podmínkách vozu (teplota) a o způsobu zacházení s nákladem (akcelerometr, otevírání dveří atd.). Účastníka lze označit jako „zákazníka“ z pohledu obchodního modelu.

4.1.2 Požadavky na systém

V rámci analýzy byly k dnešnímu dni identifikovány následující funkční a nefunkční požadavky jednotlivých účastníků systému:

Req 1 – Systém musí být schopen pracovat s komunikační jednotkou ve voze, která je již navržena a vyrobena. Jednotka podporuje servisní zásahy pomocí lokálního konfiguračního rozhraní (USB). Její blokové schéma je následující:



Obrázek 29 Blokové schéma komunikační jednotky, (zdroj: Ideový autor systému)

Req 2 – Systém musí být schopen kontaktovat komunikační jednotku a zaslat jí žádost o odpověď – např. o vybraná data z paměti.

Req 3 – Systém musí být schopen obdržet od komunikační jednotky odpověď a zpracovat ji.

Req 4 – Systém musí být schopen pracovat v off-line režimu, zejména s tím, že komunikační jednotky se mohou vyskytovat v místech bez pokrytí mobilním signálem.

Req 5 – Systém musí být schopen pracovat s ohledem na to, že komunikační jednotky jsou napájeny z lokální baterie, její výměna probíhá pouze v rámci servisních intervalů, a tudíž musí s energií maximálně šetřit. Jednotka tedy není trvale on-line, pouze se v pravidelných intervalech „probouzí“, komunikuje a znovu „usíná“.

Req 6 – Komunikační jednotka musí v pravidelných intervalech zasílat stavové informace – synchronní přenosy.

Req 7 – Komunikační jednotka může v nepravidelných intervalech zasílat alarmy – asynchronní události.

Req 8 – Data zaslaná komunikační jednotkou mohou obsahovat data o změně stavu (otevření dveří vozu), stavové informace (teplota, stav opotřebení brzdového systému, stav baterie apod.), lokační údaje GPS a další.

Req 9 – Data zasílaná komunikační jednotce mohou obsahovat tři typy událostí:

1) konfigurační příkazy
2) nastavování alarmů
3) aktualizace firmware

Req 10 – Celý systém by měl být zabezpečen před zneužitím. Jde zejména o přístup k datům v evidenci centrálního systému a podvrhům hlášených událostí.

Req 11 – Do systému by měli mít přístup externí uživatelé – zákazníci.

Req 12 – Externí zákazníci mohou mít různé role a jejich kombinace – např. provozovatel vozu (řeší zejména servisní zásahy), dopravce (řeší zejména polohu a stav zásilky – otevření dveří, teplotu), provozovatel systému (řeší stav baterie a aktualizaci konfigurace a firmware).

Req 13 – Systém musí být schopen pracovat s různým typem komunikačních jednotek. Ať již jde o různé funkční či senzorické výbavy (jednotky sledující výhradně stav brzd, případně jednotky s nebo bez možnosti sledování polohy atd.), hardware a jeho verze (starší či novější jednotky).

Req 14 – Komunikační jednotka musí mít synchronizovaný čas s centrálním systémem.

Req 15 – Lokální servisní připojení musí umožňovat minimálně následující funkcionalitu:

1) diagnostiku
2) kalibraci čidel
3) nastavení komunikace (IP adresa, uživatelský účet)
4) aktualizaci firmware

Req 16 – Centrální systém musí udržovat data z jednotky obdržená i data do jednotky odeslaná tak, aby byl zajištěn přístup k posledním údajům i k historii dat pro konkrétní jednotku.

Req 17 – V případě změny dat přes administrační lokální konfigurační rozhraní technikem následně dochází k odeslání dat a synchronizaci v centrálním systému.

Req 18 – Centrální systém musí poskytovat jednotlivým uživatelům data především formou API. API je hlavní produkční přístup k centrálnímu systému. Webový přístup je doplňkový a slouží zejména pro ověření funkcionalit v rámci implementace systému.

4.1.3 Rámec systému

Systém bude provozován na železnici v rámci celé evropské železniční sítě. Je třeba, aby přechod komunikační jednotky mezi jednotlivými státy nevyžadoval zásah uživatele.

4.1.4 Cíle

1) Systém spolehlivě funguje s minimálními provozními náklady.
2) Na trhu je zájem o služby provozovatele systému.
3) Provozovatel systému je komerčně úspěšný.

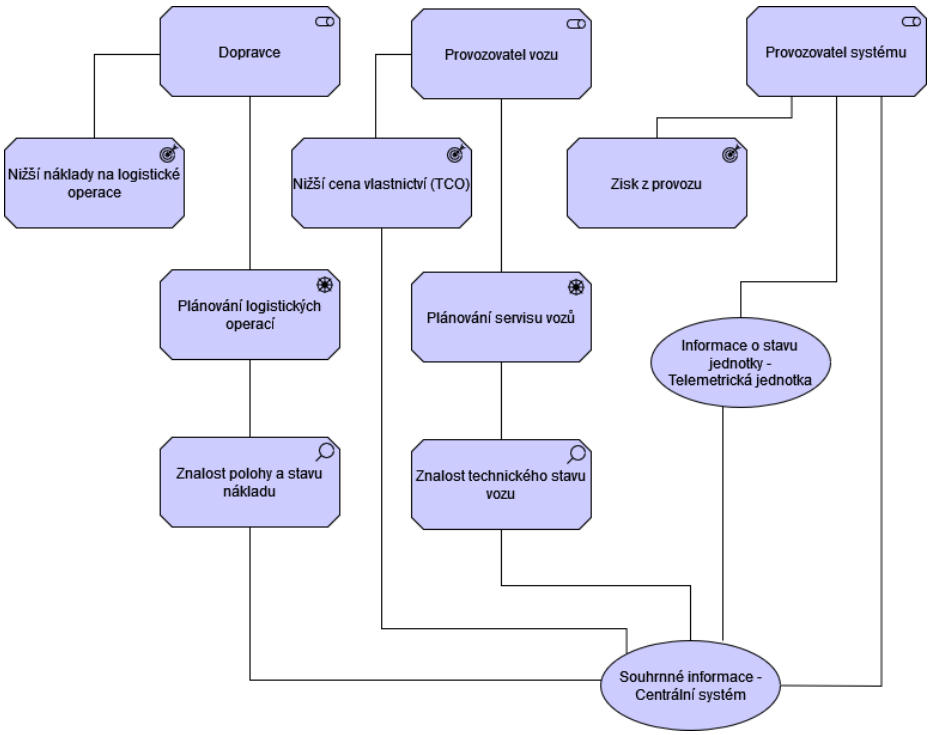
4.2 Vize architektury

Fáze vize architektury (TOGAF – A. Architecture vision) má za cíl zejména vytvořit rámcovou představu schopností a obchodních hodnot, které vzniknou v rámci EA, a získat potvrzení k seznamu prací (Statement of Work - SoW), které budou potřeba k dosažení této vize.

4.2.1 Rámcová architektonická vize

Cílem vize je systém, který umožní sbírat data z telemetrických jednotek vozů během jejich provozu. Provozní data jsou sbírána a v pravidelných intervalech odesílána do centrálního evidenčního systému, kde jsou uchovávána a archivována. Některá data mohou být odesílána taktéž ad hoc – zejména alarmy. Telemetrické jednotky lze na dálku konfigurovat zasláním konfiguračních příkazů, případně lze do jednotky také zaslat příkaz k aktualizaci jejího firmware.

Systém je optimalizovaný z hlediska nákladů na komunikaci jednotek, údržbu jednotek, provozu samotného systému a taktéž z hlediska rizik (reputační, bezpečnostní).



Obrázek 30 ArchiMate - motivace účastníků systému, (zdroj: autor práce)

4.2.2 Seznam prací k dosažení vize

V rámci této vize je potřeba vybudovat a zprovoznit Centrální systém, popsat základní procesy, datové artefakty, navrhnout možné způsoby komunikace a zabezpečení. Telemetrické komunikační jednotky budou následně výrobcem upraveny dle vybrané finální specifikace.

4.3 Obchodní architektura

Fáze obchodní architektury (TOGAF – B. Business architecture) má za cíl zejména vytvořit rámcovou obchodní architekturu pomocí definice toho, jak budou vypadat obchodní procesy k dosažení cílů. Zároveň by měla odpovědět na strategické cíle v architektonické vizi a ukotvit seznam prací (SoW) a jednotlivé účastníky (stakeholders). Měla by identifikovat kandidáty pro hlavní komponenty cílové architektonické vize, jako rozdíl mezi základní a cílovou obchodní vizí.

4.3.1 Rámcová obchodní architektura

Správce systému vyrábí a dodává telemetrické jednotky provozovatelům železničních nákladních vozů. Benefit pro správce systému je ten, že nabídne provozovateli doplněk vozu pro optimalizaci servisních návštěv.

Provozovatel vozu železniční nákladní dopravy získává s telemetrickou jednotkou benefit v možnosti optimalizovat náklady na servis a provoz nákladních železničních vozů. Cesta vozu do servisu je na železnici poměrně nákladná. Provozovatel či vlastník vozu je dle Zákona o drahách č. 266/1994 Sb. zodpovědný za pravidelnou údržbu a kontrolu technického stavu vozidel včetně brzdného systému. Cílem tohoto systému je nabídnout provozovateli detailní přehled o stavu všech systémů na jednotlivých vozech tak, aby bylo možné servisní intervaly optimalizovat.

Dopravce získává od provozovatele železniční nákladní dopravy benefit v podobě možnosti automatizovaného sledování (trackingu) jednotlivých vozů s nákladem, a tím i možnost lépe plánovat následující logistické operace zboží převáženého na železničním voze vybaveném telemetrickou komunikační jednotkou. Systém dopravce musí být schopen se na systém provozovatele automatizovaně napojit.

4.3.2 Seznam komponentů k zajištění funkce systému

Podle prvotní vize a identifikovaných požadavků by v systému měly existovat následující základní komponenty:

4.3.2.1 Centrální systém

V požadavcích se vyskytuje tzv. centrální systém a jeho jednotliví uživatelé (dopravce, provozovatel vozu, správce systému). Centrální systém je složen z API dostupného pro komunikační jednotky, datového úložiště pro ukládání stavů jednotlivých jednotek, prezentační vrstvy pro koncové uživatele (web), aplikační vrstvy zpracovávající a řídící komunikaci mezi centrálním systémem a komunikačními jednotkami a vrstvy zabezpečení a autentizace pro jednotlivé uživatele a samotné komunikační jednotky.

4.3.2.2 Telemetrická komunikační jednotka

Jedná se o jednočipovou, autonomní jednotku s vlastním bateriovým napájením, snímající data ze senzorických systémů vozu. Jednotka musí mít schopnost komunikovat s jednotlivými telemetrickými systémy vozu (dveře, brzdy, …), komunikovat s centrálním systémem (GSM) a zároveň zjišťovat svůj stav (baterie, poloha, čas, …).

Komunikačních jednotek může být více druhů, s rozdílnou funkcionalitou v rámci senzorické či komunikační sady a dalších parametrů. Tento fakt je potřeba zohlednit při návrhu samotného obsahu a formátu vyměňovaných zpráv.

V principu lze očekávat, že jednotky se budou dále vyvíjet a je potřeba dlouhodobě zajistit zpětnou kompatibilitu celého řešení.

4.3.2.3 Komunikační protokoly

Protokol komunikační jednotky pro servisní zásahy:

- Protokol pro servisní zásahy není ovlivněn žádným z požadavků a lze použít například USB (Universal Serial Bus) protokol.

Protokol komunikační jednotky s centrálním systémem:

- Protokol komunikační jednotky s centrálním systémem musí umožňovat zasílání základních stavových informací a příjem většího balíku informací, jako je například nový firmware. Objemově se může jednat o zprávy o velikost jednotek bajtů obsahu pro stavový alarm (např. baterie klesla pod 1.2V) až po jednotky megabajtů (firmware).
- Zprávy je potřeba spolehlivě doručit, nemělo by docházet k jejich ztrátám ani podvrhům, ať už ze strany serveru směrem k jednotce nebo i obráceně od jednotky k serveru. Zároveň se však nelze spolehnout na to, že komunikační jednotka bude v okamžiku komunikace ze strany serveru k dispozici – tj. on-line. Důvodem je úspora energie (jednotka je napájená z vlastní baterie) uspáváním samotné jednotky a zejména také komunikačního GSM modulu – ten se zapíná pouze v situacích, na které je nakonfigurován.

4.3.3 Rozdíl mezi současným stavem a cílovým stavem

V současné době existuje základní vize - obchodní koncepce, která je formulována budoucím provozovatelem systému. V předsériové výrobě existují prototypy telemetrické komunikační jednotky a servisní nástroje. Centrální systém se nachází v úrovní analytické fáze.

V cílovém stavu by měl být Centrální systém schopen komunikace s jednotlivými telemetrickými komunikačními jednotkami, měl by poskytovat své funkce a služby přes zabezpečené REST API. Jednotky mohou zasílat informace ad hoc nebo plánovaně, při samotném odeslání prověří frontu instrukcí na čekající a nezpracované, tyto si samy stáhnou a u vyžádaných případů i zpět potvrdí přijetí a zpracování.

Zároveň by měl být umožněn zákazníkům (dopravcům nebo provozovatelům železničních nákladních vozů) přístup do systému tak, aby z něho mohli získávat informace v rámci svých návazných procesů. Z toho důvodu bude nezbytné, aby byla v provozu konfigurace přístupových práv a aby byly vystaveny služby poskytující nezbytné klientské informace (poloha, stav senzorů, …).

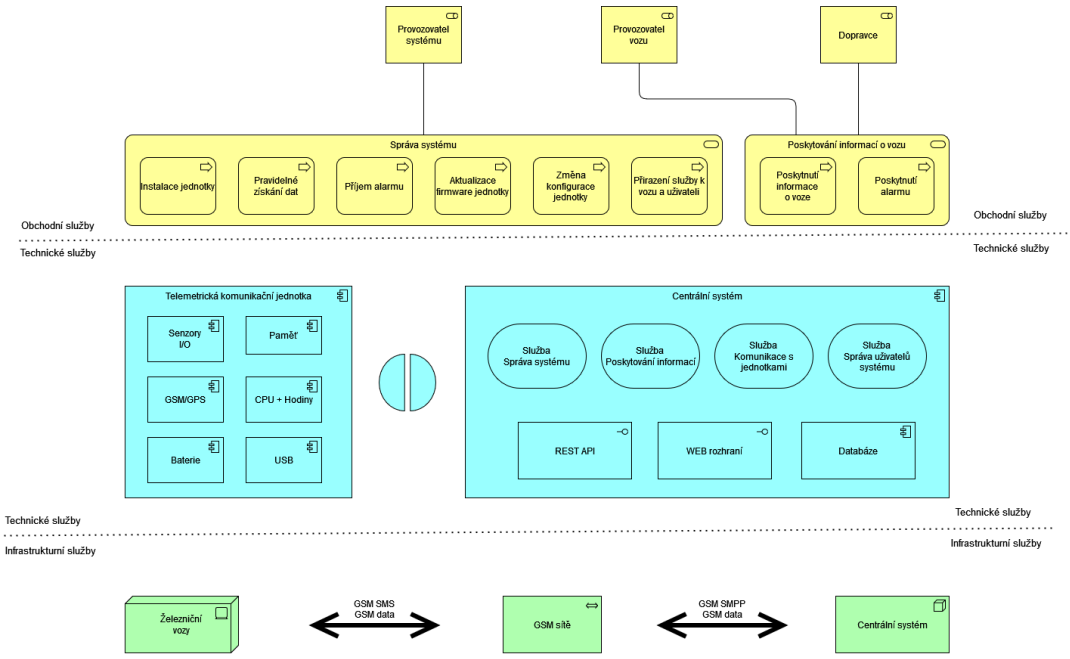
4.4 Architektura informačních systémů

Fáze architektura informačních systémů (TOGAF - C. Information systems architecture) má za cíle zejména vytvoření cílové architektury informačních systémů a popis toho, jak tato architektura umožní dosáhnout obchodních cílů v rámci definovaného seznamu prací (SoW). Mělo by dojít k identifikaci bloků funkcionalit jako vstup pro plánování vývoje tak, aby pokrýval mezery mezi současným a cílovým stavem architektury.

4.4.1 Koncepční blokové schéma cílové architektury

Koncepční blokové schéma popisuje jednotlivé primární uživatelské role v systému a jejich spojení s obchodními službami. Tyto obchodní služby („Instalace jednotky“, „Poskytnutí alarmu uživateli“, …) jsou poskytovány technickými službami na úrovni komponent centrálního systému.

Telemetrická komunikační jednotka a její komponenty získávají informace z vozu a pomocí GSM (či Data kanálu) je dávkově poskytují i získávají z centrálního systému.



Obrázek 31 ArchiMate - koncepční blokové schéma, (zdroj: autor práce)

4.4.2 Bloky funkcionalit

Veškeré služby jsou chráněny autentizací a autorizací. Přístup je chráněn Web Application Firewallem (WAF), který umožňuje komunikovat pouze předem definovanými dotazy.

4.4.2.1 Služba Správa systému

1) Správa telemetrické komunikační jednotky
    - Správa konfigurace jednotky
        - aktivní senzory
        - HW typ jednotky a její vlastnosti
        - adresa serveru
        - konfigurace komunikace s centrálním serverem
    - Správa zabezpečení jednotky
        - autentizační mechanismus
    - Synchronizace času
        - žádost o synchronizaci času
    - Správa firmware jednotky
        - žádost o aktualizaci firmware jednotky
2) Správa organizačních struktur
    - Spravuje společnosti
    - Spravuje oprávnění a role společností ke skupinám telemetrických komunikačních jednotek
3) Správa alarmů
    - Konfigurace monitorovaných událostí
    - Konfigurace akcí při typu události

4.4.2.2 Služba poskytování informací

1) Poskytuje historii a aktuální stav lokačních údajů.
2) Poskytuje historii a aktuální stav alarmů.
3) Poskytuje historii a aktuální stav konfigurace a firmware.
4) Poskytuje historii a aktuální stav jednotlivých senzorů.

4.4.2.3 Služba komunikace s jednotkami

1) Poskytuje rozhraní pro příjem informací z jednotek.
2) Poskytuje frontu zpráv pro jednotku.
3) Zabezpečuje autentizaci jednotek.
4) Zabezpečuje důvěrnost komunikace s jednotkami.

4.4.2.4 Služba správa uživatelů systému

1) Definuje jednotlivé uživatele.
2) Definuje příslušnost uživatele k firmě.
3) Definuje role a práva konkrétního uživatele.

4.5 Technologická architektura

Fáze technologické architektura (TOGAF - D. Technology architecture) má za cíle zejména vyvinout cílovou technologickou architekturu, která umožní dosáhnout cílové vize společně se stavebními bloky, jež budou poskytovat technologické komponenty a služby tak, aby došlo k zajištění SoW a potřeb jednotlivých účastníků.

4.5.1 Komunikační vrstva

Komunikace mezi síťovými zařízeními probíhá automatizovaně. Centrální systém spravuje data, konfigurace a přístupové údaje.
Telemetrická komunikační jednotka:

1) Navazuje v předem nastavených intervalech spojení se serverem a odesílá nakonfigurované údaje ze své paměti (poloha, stav baterie, stav senzorů, …).
2) Navazuje v předem nastavených situacích neplánovaná spojení za účelem oznámení konkrétní změny stavu jednotky (alarm atp.).
3) Stahuje frontu příkazů a jeden po druhém je plní, po splnění označuje odpovědí za hotové.

4.5.1.1 Dostupné přenosové standardy

- SMS – je varianta komunikace v rámci systému. Lze ji využívat dle potřeby jako primární nebo také jako záložní (fall-back) řešení. Je vhodné tyto varianty řešit pomocí konfigurace celého systému. Umožňuje optimalizovat náklady v situacích, kdy datové přenosy (např. v roamingu) znamenají vyšší cenu provozu systému. Výhodou je tedy nejen provozní cena, avšak také funkčnost, zejména v místech horšího pokrytí mobilním signálem. Nevýhodou jsou omezení způsobené omezením velikosti datového prostoru v jedné zprávě (částečně řešitelné pomocí spojených textových zpráv – Concatenated short message) a asynchronním principem SMS zpráv.
    - SMS-MO (SMS Mobile Originated) - jsou zprávy odesílané z mobilního zařízení. Vhodné pro přenos dat z telemetrické komunikační jednotky směrem k centrálnímu systému.

    - SMPP (Short message Peer-to-Peer) – je standard pro zasílání zpráv pomocí TCP spojení ze serveru směrem k mobilnímu zařízení. Vhodné pro přenos dat z centrálního systému směrem k telemetrické datové jednotce. Výhodou je možnost přímého napojení serveru do sítě operátora GSM, bez nutnosti pořizovat vlastní sadu GSM modulů na straně serveru.
- Mobilní Data – poskytují TCP/UDP konektivitu na sítích 3G a výše. Výhodou je přímé navázání spojení. Nevýhodou jsou požadavky na kvalitu pokrytí, možné poplatky (např. roaming - v režimu provozu telemetrické komunikační jednotky mimo primární síť operátora). Je použitelné v následujících sítích:
    - GPRS - General Packet Radio Service
    - EDGE – Enhanced Data Rte for GSM Evolution
    - UMTS - Universal mobile Telecommunication System
    - LTE – Long-term Evolution

4.5.2 Centrální systém

4.5.2.1 Nasazení

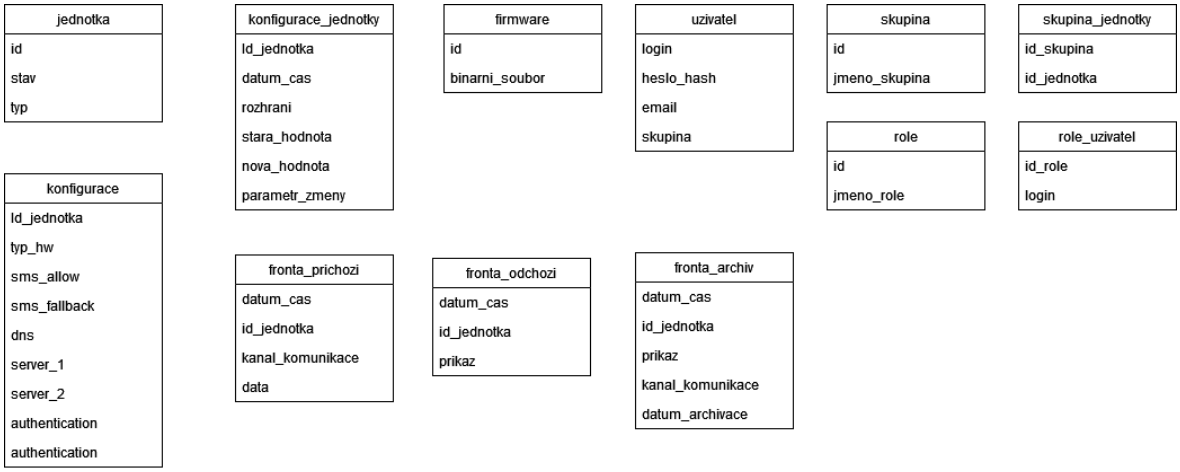
Z hlediska nasazení backendu je možno vybrat kteréhokoliv poskytovatele ze tří zmíněných, například Amazon Web Service.

4.5.2.2 Technologie

Z hlediska technologie lze doporučit kterýkoliv ze zvažovaných rámců z rešerše. Hlavními faktory pro výběr bude stabilita a dostupnost vývojářského týmu pro konkrétní technologii. Vzhledem ke stavu na trhu se může zdát vhodný například Spring Boot (Java). Pro sběr a distribuci zpráv by bylo možné koncepčně využít systémy na zprávu fronty (message broker), nicméně vzhledem k plánovanému rozsahu systému by nyní zajištění oprávnění k jednotlivým tématům a událostem mohlo přinést zbytečnou komplexitu a těžkopádnost. V budoucnu by asi nemělo nic bránit přechodu například na MQTT server/broker jako prostředníka, ale pro současný systém bude jednodušší řešit role a práva k datům na úrovni datových struktur.

4.5.2.3 Datové artefakty

Centrální systém je v této práci řešen pouze na obecné úrovni. Relační vztahy a jednotlivé procesy nejsou do detailu popisovány, protože k nim není dost vstupních informací. Obecně se dá říci, že systém bude spravovat mimo jiné přibližně následující datové artefakty:



Obrázek 32 Diagram tříd centrálního systému, (zdroj: autor práce)

1) *jednotka* – evidence samotných jednotek
    - Stav jednotky (nasazena, servisována, ve výrobě, ...).
2) *konfigurace* – evidence konfigurací jednotek
    - Typ jednotky – hardware verze, snímače, vlastnosti
    - Povolené protokoly komunikace a jejich konfigurace
        - Adresa serveru
        - Autentizační řetězec
3) *konfigurace_jednotky* – historie administrace jednotky
    - Změny konfigurace a zdroj této změny
    - Změny firmware a zdroj této změny
4) *fronta_prichozi* / *fronta_odchozi* – eviduje komunikace mezi centrálním systémem a jednotkami
    - Příkazy čekající pro jednotlivé jednotky mohou být např:
        - Synchronizuj čas
        - Nastav konfigurační parametr XY na hodnotu AB
        - Pošli stav senzoru XY (více senzorů, období historie, …)
        - Resetuj alarm XY

        - Pošli stav konfigurace jednotky
        - Stáhni nový firmware
    - Příchozí data z jednotek ke zpracování centrálním systémem
        - Jednotka, čas, poloha, stav senzorů, stav baterie, uptime na baterii, firmware
        - Jednotka, čas, poloha, alarm XY, stav baterie
5) *fronta_archiv* – historické údaje obdržené z jednotek
    - Odeslané do jednotek (stažené, potvrzené)
    - Obdržené z jednotek
6) *uzivatel* – Uživatelé Centrálního systému
    - Login
    - Heslo (hash)
    - Email
    - Skupina
    - Role
7) *skupina* / *skupina_jednotky* – vytvoření skupin pro přístup k datům z jednotek
    - Jednotky a skupiny uživatelů s přístupem k jejich datům
8) *role* / *role_uzivatel* – vytvoření rolí pro přístup ke konkrétním datům z jednotek
    - Seznam rolí v systému
9) *firmware* – registrace verzí firmware ke stažení jednotkou
    - Seznam verzí firmware ke stažení

4.5.3 Protokolu telemetrické datové jednotky

Pro samotný protokol byl vybrán CoAP, zejména pro jeho schopnost využívat asynchronní kanál, jako je UDP a SMS. Zároveň umožňuje v omezené šíři řídit datový provoz pomocí potvrzovaných či nepotvrzovaných zpráv, případně i díky možnosti do potvrzení rovnou připojit odpověď. Velikost obsahu přenášené informace se přizpůsobuje aktuální velikosti datagramu.

4.5.4 Protokolu pro zákazníky systému

Přístup k informacím pro jednotlivé zákazníky systému je potřeba realizovat prostřednictvím HTTP REST API, protože tito zákazníci se budou integrovat zejména pomocí automatizovaných způsobů – neočekává se manuální zjišťování informací z centrálního systému. Toto rozhraní by mělo být zabezpečeno alespoň pomocí JWT (JSON Web Token). Pro integraci mezi systémy zákazníků a centrálním systémem by mělo být použito mTLS pro zabezpečení důvěrnosti přenášených dat a autentizaci systému zákazníka.

4.5.5 Koncepce zabezpečení

V praxi je nutné koncepčně zvážit rizika a rozhodnout o nastavení úrovně zabezpečení

- Autentizace a autorizace – prolomení jednoho přístupu by nemělo umožnit převzetí celého systému. Jednotlivé jednotky by se měly autentizovat pomocí individuálních přístupových klíčů. V centrálním systému musí dojít k segregaci rolí. Administrace a plný přístup k systému smí být možný pouze z administrátorské sítě. Klientské přístupy nesmí mít přístup k infrastruktuře. Každá role má přístup pouze k datům, která ji náležejí. Žádná jednotka nesmí mít přístup k datům jiné jednotky.
- Zabezpečení komunikace proti útokům – rizikem je převzetí systému nebo odstavení systému např. pomocí DDoS útoků. Systém je potřeba zabezpečit webovým aplikačním firewallem.
- Kompromitace dat v centrálním systému – rizika jsou od neoprávněného přístupu k datům až po odstavení celého systému. Centrální systém sice neobsahuje citlivá nebo osobní data, nicméně pokud má být pro jednotlivé účastníky systému považován jako referenční (např. z hlediska plánování servisu), musí data poskytovat i uchovávat spolehlivě a bezpečně.
- Technické zabezpečení centrálního systému – celá infrastruktura musí být průběžně zabezpečována a ověřována pomocí nejnovějších poznatků a doporučení. Systém musí splňovat relevantní požadavky OWASP.
- V rámci vývoje systému je nezbytné také dodržovat pravidla pro bezpečný vývoj, správu zdrojového kódu, revize kódu, dodržovat bezpečnostní pravidla,

optimálně pracovat s metodikami jako párové programování a další. Zároveň by celý systém měl průběžně procházet bezpečnostní analýzou s cílem identifikovat rizika a způsoby jejich ošetření. Je nutno podotknout, že bezpečnost je potřeba již od začátku považovat za jednu z klíčových oblastí celého systému (GUTMANN, 2019).

5 Diskuse

5.1 Průběh projektu

V průběhu realizace projektu došlo nejprve k ujasňování zvoleného tématu s vedoucím práce. Došlo ke společnému oslovení několik potencionálních odborníků v oblasti telemetrie. Pro zpracování bylo vybráno téma definované ideovým autorem koncepce celého systému telemetrických jednotek. Během prvních konzultací s ním došlo k vyjasnění vize systému, jeho současného stavu a očekávaných úkolů při realizaci. Následně se povedlo sjednotit na cílech vhodných ke zpracování v rámci diplomové práce.

Následovalo odsouhlasení cíle s vedoucím diplomové práce, včetně zapracování jeho připomínek. Po vzájemném odsouhlasení bylo možné s prací začít.

První etapou bylo posbírání požadavků a vstupů ze strany zadavatele. Během několika konzultací osobních i emailových došlo k finalizaci a odsouhlasení sebraných požadavků a přešlo se k samotné realizaci.

Realizace započala prováděním rešerší a studiem odborné literatury. Následoval výběr nástrojů a rámců pro samotnou realizaci. Ve výsledku měl asi největší vliv rámec TOGAF, podle kterého došlo k definici a zápisu obchodní vize, obchodních, technických a infrastrukturních služeb pomocí ArchiMate.

Celkově by se dalo shrnout, že fáze výběru tématu trvala 2 měsíce, odsouhlasení cíle přibližně 1 měsíc, rešerše opět jeden měsíc a samotná realizace 2 měsíce práce, včetně finální revize a připomínkování celé práce ideovým autorem koncepce a zapracováním jeho připomínek do finální verze.

Ve výsledku jsou identifikovány a definovány požadavky, účastníci systému a jejich motivace, koncepční blokové schéma systému a koncepční diagram tříd centrálního systému. Byl vybrán a doporučen protokol CoAP, který je standardizován a optimalizován do prostředí IoT a umožňuje komunikaci přes nespolehlivé kanály, navíc je i podporován na platformě ARM, použité pro vývoj telemetrických komunikačních jednotek.

5.2 Důvody pro použití protokolu CoAP

Protokol CoAP se jeví jako perspektivní řešení. Spojuje v sobě koncept IoT a dnešních moderních komunikací pomocí HTTP/REST/JSON v jednotném a dobře definovaném efektivním protokolu, vhodném jak pro komunikaci pomocí SMS, tak i přes mobilní datová spojení. Omezení daná provozními požadavky zvládne pokrýt, a zároveň díky rozumné úrovni převzetí principů HTTP umožňuje snadnější osvojení vývojáři, čímž ve výsledku podporuje efektivnější nasazení této technologie.

CoAP je podporován na platformě ARM, existuje k němu mnoho opensource implementací:

- mbed – zahrnuje CoAP podporu
- Californium a jCoAP Java knihovny
- libCoAP, OpenCoAP, Erbium C knihovny
- jCoAP Java knihovna
- Wireshark podporuje detailní náhled na protokol

Doporučení a návrh využití protokolu CoAP pro komunikaci mezi telemetrickými datovými jednotkami a centrálním systémem je založen zejména na faktu, že datová komunikace a výpočetní zdroje telemetrických komunikačních jednotek jsou (a v dohledné době budou) omezené. Zároveň je v potaz brán zdroj elektrické energie, kterým je (v každém voze umístěná) baterie. Ta musí mít životnost cca 2-5 let, podle odhadované délky servisních intervalů železničních vozů. Z toho důvodu dochází cíleně k minimalizaci datových toků, které zásadním způsobem ovlivňují spotřebu energie z baterie.

Protokol umožňuje transparentní přechod mezi přenosovou vrstvou pomocí SMS a mobilního datového připojení dle zhodnocení obchodních a cenových hledisek síťového operátora beze změny v komunikačního protokolu.

V případě, že by došlo ke změně v nastavení prostředí, jako je například lepší výpočetní výkon a zdroje jednotky, dostupnost zdroje elektrické energie nebo komunikačního kanálu z vozu, jistě by bylo vhodnější použít běžné protokoly na bázi HTTP. Takovou změnu však zatím dle komunikace se zadavatelem nelze očekávat. CoAP by však i v takovém případě byl nadále vhodným protokolem. Jedinou jeho

nevýhodou je nutnost vývojářů se seznamovat s novým druhem protokolu, byť velmi podobným standardu HTTP.

5.3 Mimo rámec této práce

Mimo rámec této práce (vzhledem k jejímu omezenému rozsahu) zůstalo několik důležitých témat, která by bylo vhodné v případě přípravy realizace obdobného systému do produkčního nasazení ještě dále řešit:

- Nebyla řešena spotřeba jednotky a požadavky na baterii. Jistě by bylo vhodné takové výpočty zajistit a prověřit taktéž porovnání spotřeby komunikace přes mobilní data oproti komunikaci pomocí SMS zpráv. Využití SMS by mělo znamenat menší energetickou náročnost komunikace, zejména pokud se zprávy datově vejdou do velikosti jedné SMS zprávy.
- Očekávaný způsob dobíjení baterie telemetrické jednotky probíhá během servisního zásahu. Jednotka sama o sobě by neměla zvyšovat požadovanou frekvenci zásahu, a proto bude potřeba baterii správně dimenzovat.
- V diplomové práci se pracuje s faktem, že cena za SMS komunikaci je nižší než cena za mobilní datové spojení, zejména v případě, že se monitorovaný vagón nachází v roamingové síti, mimo domácí datovou síť. Bylo by vhodné provést ekonomickou rozvahu založenou na nabídce služeb mobilních operátorů pro oblast IoT.
- V systému by bylo možné využít tzv. Message Broker, tedy systém zajišťující sběr a distribuci zpráv a alarmů z jednotek směrem k centrálnímu serveru a následně také k jednotlivým dalším účastníkům systému. Implementace obdobné komponenty je na zvážení zejména v okamžiku, kdy systém úspěšně roste a potřebuje rozšiřovat. V současné podobě s Message Broker komponentou v konceptu a návrhu systému není počítáno.
- Nutno zároveň podotknout, že celá práce je založena na vizi zadavatele a jeho znalostech prostředí nákladní železniční dopravy, principů a procesů v této oblasti. Samotná realita nasazení a provozu se může z mnoha objektivních či subjektivních důvodů lišit od vstupů, které tato práce identifikovala.
- V rámci centrálního systému není detailněji řešena komunikace dalších účastníků.

- Práce do detailu nerozebírá sekvenční posloupnost v rámci jednotlivých procesů, jako je životní cyklus jednotek, synchronizace času jednotek a podobně. Zároveň nebyla řešena relační vazba mezi datovými artefakty. Vše by mělo být konkrétněji definováno v další (přípravné nebo zadávací) fázi vývoje.
- Návrh obsahu zpráv (včetně zajištění kompatibility mezi jednotlivými verzemi jednotek) již bude součástí implementace protokolu v rámci detailního zadání vývoje systému.

6 Závěr

Tato práce splnila cíle, které si vytyčila. V rámci diplomové práce se podařilo nalézt zajímavý projekt z oblasti IoT ve fázi přípravy. Povedlo se identifikovat jednotlivé plánované účastníky takového systému. Byly posbírány jejich funkční a nefunkční požadavky. Pomocí analytického a architektonického rámce bylo navrženo koncepční řešení backendového systému, jeho hlavní komponenty, včetně protokolu pro komunikaci s uživateli. Pro komunikaci směrem k telemetrickým datovým jednotkám byl vybrán protokol zohledňující technická omezení daná návrhem a požadavky na samotné jednotky. V návrhu systému byla brána v úvahu kritéria bezpečnosti. Tuto práci je možné použít jako jeden ze vstupů pro zadávací dokumentaci k realizaci. Posbírané vstupy, rešerše a návrhy konceptu by měly poskytnout odrazový můstek pro další detailní specifikace pro případné zadání, pokud by se ideový tvůrce systému rozhodl pro realizaci a komerční spuštění popisovaného systému.

7 Seznam použitých zdrojů

- BRENNAN, Kevin. *A guide to the Business analysis body of knowledge (BABOK guide)*. Toronto: International Institute of Business Analysis, 2009.
- European Telecommunications Standards Institute (ETSI). *ETSI TS 123 040 - Technical realization of the Short Message Service (SMS) - Point-to-Point (PP)*. Verze 16.3.0. Dostupné z: https://www.etsi.org/deliver/etsi_ts/123000_123099/123040/16.03.00_60/ts_123040v160300p.pdf, přístupné 9. března 2023.
- FOWLER, Martin. *UML distilled : a brief guide to the standard object modelling language.* Boston: Addison-Wesley, 2004. ISBN 0321193687.
- GUTMANN, Peter. *Cryptographic security architecture: design and verification*. Berlin: Springer, 2004. ISBN 978-1-4419-2980-8.
- Internet Engineering Task Force (IETF). *JSON Web Token (JWT). RFC 7519*. Dostupné z: https://tools.ietf.org/html/rfc7519, přístupné 9. března 2023.
- Internet Engineering Task Force (IETF). *Mutual Transport Layer Security (mTLS). RFC 8705*. Dostupné z: https://tools.ietf.org/html/rfc8705, přístupné 9. března 2023.
- Internet Engineering Task Force (IETF). *RFC 7252 - The Constrained Application Protocol (CoAP).* Verze 1.0. Dostupné z: https://tools.ietf.org/html/rfc7252, přístupné 9. března 2023.
- OPEN MANAGEMENT GROUP. *Business Proces Model and Notation (BPMN), version 2.0, 2011*
- OPEN MANAGEMENT GROUP. *Unified Modeling Language (UML), version 2.5.1, 2017*
- Organization for the Advancement of Structured Information Standards (OASIS). *MQTT Version 5.0*. Dostupné z: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html, přístupné 9. března 2023.
- SALAM, Abdul. *Internet of things for sustainable community develoment : wirless communications, sensing, and systems.* Cham: Springer 2020. ISBN 978-3030352905.

- SHELBY, Z. (2018). ARM IoT Tutorial - CoAP: The Web of Things Protocol. Dostupné z: https://www.arm.com/resources/iot/coap-the-web-of-things-protocol [Citováno 29. března 2023].
- SCHEDLEBAUER, Martin. *The art of business proces modeling : the business analyst‘s guide to proces modeling with UML & BPMN.* Sudbury: Cathris Group, 2010. ISBN 978-1-4505-4166-4.
- SMPP Developers Forum. *SMPP Protocol Specification v3.4*. Dostupné z: https://smpp.org/doc/SMPP_v3_4_Issue1.pdf, přístupné 9. března 2023.
- THE OPEN GROUP. *ArchiMate, version 3.1*, 2021.
- THE OPEN GROUP. *The Open Group Architecture Framework (Togaf)*, *version 9.2,* 2018.
- Zákon č. 266/1994 Sb., o drahách.

8 Seznam obrázků

9 Seznam použitých zkratek

2G/3G/4G/5G – 2. generace / 3. generace / 4. generace / 5. generace

ADM – Architecture Development Method

API – Application Programming Interface

AWS – Amazon Web Services

BABOK – Business Analysis Body of Knowledge

BPMN – Business Process Model and Notation

CoAP – Constrained Application Protocol

DoD – Department of Defense

DDoS – Distributed Denial of Service

DTLS – Datagram Transport Layer Security

EA – Enterprise architecture

EDGE – Enhanced Data Rte for GSM Evolution

GCP – Google Cloud Platform

GPRS – General Packet Radio Service

GPS – Global Positioning System

GSM – Global System for Mobile Communications

HTTP – Hypertext Transfer Protocol

IEEE – Institute of Electrical and Electronic Engineers

IEC – International Electrotechnical Commission

IETF – Internet Engineering Task Force

IoT – Internet of Things

IP – Internet Protocol

ISO – International Organization for Standardization

JSON – JavaScript Object Notation

ITU-T – International Telecommunication Union

JWT – JSON Web Token

LTE – Long-term Evolution

LwM2M – Lightweight Machine-to-Machine

M2M – Machine-to-Machine

MQTT – Message Queuing Telemetry Transport

mTLS – Mutual Transport Layer Security

OSI – Open Systems Interconnection

OWASP – Open Web Application Security Project

PKI – Public Key Infrastructure

RAM – Random Access Memory

REST - Representional State Transfer

REQ – Requirement

ROM – Read-Only Memory

SCTP – Stream Control Transmission Protocol

SM-TP – Short Message Transfer Protocol

SMS – Short Message Service

SMS-MO – SMS Mobile Originated

SMPP – Short Message Peer to Peer

SoW – Statement of Work

TKL – Token Length

TOGAF – The Open Group Architecture Framework

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

UML – Inified Modelling Language

UMTS – Universal mobile Telecommunication System

URI – Uniform Resource Identifier

USB – Universal Serial Bus

W3C – World Wide Web Consortium

WAF – Web Application Firewall