

Systém pro sběr a zpracování dat v zemědělství

Bc. Jakub Trubka

Diplomová práce
2024

 Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Jakub Trubka**
Osobní číslo: **A21510**
Studijní program: **N0613A140022 Informační technologie**
Specializace: **Softwarové inženýrství**
Forma studia: **Kombinovaná**
Téma práce: **Systém pro sběr a zpracování dat v zemědělství**
Téma práce anglicky: **System for Data Collection and Processing in Agriculture**

Zásady pro vypracování

1. Seznamte se s metodami sběru dat v zemědělství, jejich bezpečností filtrováním, vyhodnocením a vizualizací.
2. Navrhněte systém pro sběr, filtrování a vizualizaci dat vhodný pro využití v zemědělství.
3. Navrhněte zařízení, které bude kompatibilní s navrženým systémem pro sběr dat určené pro sběr dat v zemědělství.
4. Implementujte navržený systém.
5. Vytvořte zařízení pro sběr dat.
6. Otestujte vytvořené zařízení a systém v reálném provozu.
7. Vyhodnotte chování systému a možnosti integrace. Diskutujte robustnost, škálovatelnost a možné využití realizovaného systému.

Seznam doporučené literatury:

1. BUYYA, Rajkumar; CALHEIROS, Rodrigo N.; DASTJERDI, Amir Vahid (ed.). *Big data: principles and paradigms*. Morgan Kaufmann, 2016. ISBN 978-0128053942.
2. Dunning, T. *Time series databases : new ways to store and access data*. CA: O'Reilly Media, 2014. ISBN 978-1491914724.
3. OSINGA, Sjoukje A., et al. *Big data in agriculture: Between opportunity and solution*. Agricultural Systems, 2022, 195: 103298.
4. LAVRIC, Alexandru; PETRARIU, Adrian Ioan. *LoRaWAN communication protocol: The new era of IoT*. In: 2018 International Conference on Development and Application Systems (DAS). IEEE, 2018. p. 74-77.
5. ÇORAK, Burak H., et al. *Comparative analysis of IoT communication protocols*. In: 2018 International symposium on networks, computers and communications (ISNCC). IEEE, 2018. p. 1-6.
6. BUYYA, Rajkumar; DASTJERDI, Amir Vahid (ed.). *Internet of Things: Principles and paradigms*. Elsevier, 2016. ISBN 978-0-12-805395-9

Vedoucí diplomové práce:

Ing. Bc. Pavel Vařacha, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

5. listopadu 2023

Termín odevzdání diplomové práce: **13. května 2024**

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan



prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 5. ledna 2024

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomové práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této současti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

.....

podpis studenta

ABSTRAKT

Tato diplomová práce se zabývá návrhem a implementací systému pro sběr, zpracování, uchovávání a vizualizaci dat získaných v oblasti zemědělství, primárně v oblasti sadařství a vinařství. Výsledný systém umožňuje uživateli monitorování podmínek v místě umístění zařízení se senzory. Systém klade důraz na nízkou cenu, odolnost a rozšiřitelnost. V rámci teoretické části jsou popsána existující řešení a rozebrána problematika sběru, zpracování a vizualizace dat. Praktická část se věnuje návrhu architektury systému, včetně návrhu jednotlivých zařízení pro sběr dat a komunikačních protokolů. Dále se praktická část zabývá implementací celého systému a jeho jednotlivých částí. V práci jsou též diskutovány výsledky získané experimentováním s vytvořeným systémem a diskutovány další možnosti zlepšení systému jako takového.

Klíčová slova: IoT, sběr dat, analýza dat, vizualizace dat, LoRa, MQTT, InfluxDB, zemědělství, senzory

ABSTRACT

This thesis deals with the design and implementation of a system for collection, processing, storage and visualization of data obtained in the field of agriculture, primarily in the field of orchards and viticulture. The resulting system allows the user to monitor the conditions at the location of the device with sensors. The system emphasizes low cost, durability, and expandability. In the theoretical part, existing solutions are described and the issues of data collection, processing and visualization are analyzed. The practical part is devoted to the design of the system architecture, including the design of individual devices for data collection and communication protocols. Furthermore, the practical part deals with the implementation of the entire system and its individual parts. The work also discusses the results obtained by experimenting with the created system and discusses other possibilities for improving the system as such.

Keywords: IoT, data collection, data analysis, data visualization, LoRa, MQTT, InfluxDB, agriculture, sensors

Děkuji vedoucímu diplomové práce Ing. Pavlovi Vařachovi, Ph.D. za rady, trpělivost a ochotu při konzultacích. Poděkování patří také Vinařskému spolku Uherčice za poskytnutí prostor k testování systému.

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 AKTUÁLNÍ SITUACE	12
1.1 SBĚR DAT V ZEMĚDĚLSTVÍ	12
1.1.1 Problémy	13
1.1.2 Metody sběru dat	13
1.2 SENZORY V ZEMĚDĚLSTVÍ	15
1.2.1 Metody získání dat	16
1.2.2 Umístění senzorů	17
1.2.3 Způsob nakládání s daty	17
1.3 MOŽNÉ VYUŽITÍ SYSTÉMU PRO SBĚR DAT	17
1.4 EXISTUJÍCÍ ŘEŠENÍ	18
1.4.1 Planet - Data Driven Precision Agriculture	18
1.4.2 BeeHero	18
1.4.3 Swift Sensors	19
1.4.4 Precision AG technology	19
2 POUŽÍVANÉ TECHNOLOGIE	21
2.1 IoT V ZEMĚDĚLSTVÍ	21
2.2 KOMUNIKAČNÍ PROTOKOLY	22
2.2.1 LoRa	24
2.2.2 Bluetooth Low Energy	27
2.2.3 MQTT	29
2.2.4 Mobilní síť	30
2.3 ZPRACOVÁNÍ DAT	32
2.3.1 Čištění a normalizace dat	32
2.3.2 Analýza dat	34
2.4 UKLÁDÁNÍ DAT	35
2.4.1 InfluxDB	36
2.5 VIZUALIZACE DAT	37
2.5.1 Metody vizualizace	37
2.5.2 Příklad vizualizační aplikace	39
2.6 ZABEZPEČENÍ	40
2.6.1 LoRa	40
2.6.2 Bluetooth Low Energy	41

2.6.3	MQTT	42
2.7	SERVOVÁ ŘEŠENÍ.....	43
II	PRAKTICKÁ ČÁST.....	45
3	NÁVRH SYSTÉMU	46
3.1	PŘÍPADY UŽITÍ	46
3.2	SPECIFIKACE POŽADAVKŮ SOFTWAROVÉ PLATFORMY.....	47
3.3	SPECIFIKACE NÁVRHU PLATFORMY	49
3.4	NÁVRH BRÁNY PRO ZAŘÍZENÍ	50
3.4.1	Vstupní protokol.....	52
3.4.2	Výstupní protokol.....	53
3.5	NÁVRH APLIKACE PRO PŘÍJEM A FILTROVÁNÍ DAT	57
3.6	NÁVRH RELAČNÍ DATABÁZE	59
3.7	NÁVRH ČASOVÉ DATABÁZE	60
3.8	NÁVRH APLIKACE PRO PŘÍSTUP K DATŮM	61
3.9	NÁVRH MOBILNÍ APLIKACE.....	62
4	NÁVRH ZAŘÍZENÍ	65
4.1	PŘÍPADY UŽITÍ	65
4.2	SPECIFIKACE POŽADAVKŮ ZAŘÍZENÍ PRO SBĚR DAT	65
4.3	SPECIFIKACE NÁVRHU MĚŘÍCÍHO ZAŘÍZENÍ PRO VINAŘE.....	66
4.3.1	Specializace vstupního protokolu	66
4.3.2	Návrh hardwaru	67
4.4	SPECIFIKACE NÁVRHU MĚŘÍCÍHO ZAŘÍZENÍ PRO VČELAŘE	69
4.4.1	Specializace vstupního protokolu	69
4.4.2	Návrh hardwaru	69
5	IMPLEMENTACE	71
5.1	STRUKTURA REPOZITÁŘE.....	71
5.1.1	Brána pro zařízení	71
5.1.2	Aplikace pro nahrávání dat	73
5.1.3	Aplikace pro získání dat	73
5.1.4	Databáze a nasazení systému	74
5.1.5	Mobilní aplikace	76
5.1.6	Zařízení pro vinaře	78
5.1.7	Zařízení pro včelaře	80
6	TESTOVÁNÍ.....	83
6.1	TESTOVÁNÍ POŽADAVKŮ NA SYSTÉM	83

6.1.1	Periodické měření více zařízení a ukládání dat	83
6.1.2	Test upozornění uživatele a filtrace dat	84
6.1.3	Test vizualizace dat a autorizace uživatele	86
6.1.4	Test nasazení systému	87
6.2	TESTOVÁNÍ POŽADAVKŮ NA ZAŘÍZENÍ	87
6.2.1	Testování zařízení pro vinaře.....	87
6.2.2	Testování zařízení pro včelaře.....	88
6.3	TESTOVÁNÍ V REÁLNÝCH PODMÍNKÁCH.....	89
ZÁVĚR	92
SEZNAM POUŽITÉ LITERATURY	94
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	97
SEZNAM OBRÁZKŮ	98
SEZNAM TABULEK	100
SEZNAM PŘÍLOH	101

ÚVOD

Zemědělství reprezentuje jedno z klíčových odvětví světové ekonomiky, zajišťuje potraviny, krmivo a paliva pro miliardy lidí. Nicméně čelí řadě komplexních výzev, včetně růstu populace, změny klimatu, nedostatku zdrojů, degradace životního prostředí a nestálosti na trhu. Prostředkem k řešením těchto výzev může být i zavádění moderních technologií a inovativních přístupů, které by mohly vylepšit produktivitu, efektivitu a udržitelnost zemědělství.

Jedním z moderních přístupů je sběr a analýza dat v zemědělství. Tento proces zahrnuje shromažďování, zpracování a interpretaci různých dat spojených se zemědělskými aktivitami. Mezi tato data patří například údaje o růstu plodin, stavu půdy, povětrnostních podmírkách, škůdcích a řízení farmy. Sběr a analýza dat mohou výrazně podpořit rozhodovací procesy zemědělců a dalších zainteresovaných stran, optimalizovat vstupy, snižovat náklady, zvyšovat kvalitu a výnosy. Taktéž mohou přispět k tvorbě politik, vědeckému výzkumu a vzdělávání v oblasti zemědělství. Ač se tyto technologie stávají stále dostupnější, jsou velmi nákladné. To není problém pro velké zemědělské provozy, které zavedením moderních procesů mohou zvýšením efektivity ušetřit nemalé sumy, problém ovšem nastává u menších zemědělců. Zvýšení efektivity u nich nevyrovnaná náklady na pořízení moderních systémů pro sběr a analýzu dat. Existující systémy jsou však zaměřené především na větší provozy.

Hlavním cílem této práce je vyvinout systém pro sběr a analýzu dat v zemědělství, který nabídne spolehlivé, bezpečné a uživatelsky přívětivé řešení pro sběr, filtrování a vizualizaci zemědělských dat. Práce si bere zároveň za cíl, aby celý systém byl transparentní, modulární, veřejně dostupný a levný na údržbu i nasazení. Systém se skládá ze dvou hlavních částí, zařízení pro sběr dat a platformy pro analýzu dat. Zařízení pro sběr dat je fyzické zařízení osazené senzory, určené k měření. Platforma pro analýzu dat je softwarová platforma, která přijímá, ukládá, filtruje a vizualizuje data shromážděná zařízením.

Práce obsahuje teoretickou část, ve které je popsána aktuální situace 1 a používané technologie 2. Sekce o aktuální situaci si bere za cíl seznámit čtenáře s problematikou sběru dat v zemědělství, používanými senzory a existujícími řešeními. Sekce věnující se používaným technologiím seznamuje čtenáře s problematikou IoT a analýzou dat.

Dále práce obsahuje praktickou část, která obsahuje sekce věnující se návrhu 3, implementaci 5 a testování systému 6. Návrh systému obsahuje mimo jiné i případy užití a specifikace požadavků, ze kterých vychází celý návrh, implementace i testování.

I. TEORETICKÁ ČÁST

1 AKTUÁLNÍ SITUACE

V dnešní době zaujímá získávání a správa dat klíčovou roli ve vývoji a optimalizaci různých odvětví lidské činnosti. Nasbíraná data umožňují analyzovat a optimalizovat různé výrobní procesy, zlepšovat kvalitu produktů a služeb, předvídat a řešit problémy a zvyšovat konkurenceschopnost. Například monitorování průmyslového stroje pomocí senzorů může včas odhalit jeho poruchu a zabránit tak zbytečným výpadkům ve výrobě, které by mohly způsobit ztrátu času, peněz či ohrožení lidských životů.

Zemědělství je jedním z odvětví, kde se sběr a zpracování dat stává stále důležitějším a rozšířenějším. Data mohou poskytnout zemědělcům cenné informace, které jim pomáhají při rozhodování o různých aspektech zemědělské produkce, jako je výběr vhodných rostlin, stanovení optimálních termínů výsevu a sklizně, aplikace hnojiv a pesticidů, zavlažování a další. Data také umožňují sledovat a hodnotit podmínky růstu plodin, jejich výnosnost, odolnost vůči škůdcům a chorobám. Správné využití získaných dat může výrazně zvýšit efektivitu a udržitelnost zemědělského hospodaření a zpracování plodin. Podrobnější informace o sběru dat v zemědělství jsou uvedeny v podkapitole 1.1.

Zdroje dat v zemědělství mohou být různého druhu. Mezi nejčastější patří různé typy senzorů, které měří například teplotu, vlhkost, kyselost a složení půdy, koncentraci živin, intenzitu světla a další parametry. Dalším zdrojem dat jsou satelitní a letecké snímky, které poskytují informace o rozloze, typu a stavu plodin, erozi půdy, znečištění vody a ovzduší a dalších faktorech. Kromě toho existují i administrativní záznamy, jako jsou statistiky, smlouvy, certifikáty, dotace a další dokumenty, které se týkají zemědělské činnosti. Získání dat z těchto zdrojů je v současnosti snadné a dostupné. Ceny senzorů a dalších zařízení pro sběr dat neustále klesají, zatímco jejich kvalita a spolehlivost rostou. Na trhu se objevuje stále více společností, které se specializují na sběr a zpracování dat v zemědělství a nabízejí zemědělcům různé služby a aplikace. Existují i veřejně dostupné služby, které umožňují získávat satelitní snímky půdy zdarma nebo za nízký poplatek. Příklady takovýchto společností a služeb jsou popsány v podkapitole 1.4.

1.1 Sběr dat v zemědělství

Sběr dat v zemědělství je složitý proces, který zahrnuje mnoho různých typů dat. Zemědělec obvykle nemá k dispozici jediné řešení, které by pokrývalo všechny jeho potřeby. Zemědělská technika obsahuje spoustu senzorů, ovšem data z nich se nesbírájí. Při sklizni se sbírá informace o výnosu z daného místa, ale tyto informace nejsou spojeny s množstvím srážek či se složením půdy. To, že neexistují kompletní systémy pro sběr a zpracování dat, omezuje potenciál využití moderních digitálních technologií [1]. V oblasti zemědělství se setkáváme s několika problémy, které brání plnému využití

potenciálu dat a informací, které lze z nich získat. Tyto problémy se týkají zejména fragmentace, standardizace, sdílení a dostupnosti dat [2][3].

1.1.1 Problémy

Informace z této podsekce jsou čerpány z [2]. Prvním problémem je fragmentace dat. Zemědělská data zahrnují údaje o půdě, plodinách, podmírkách, škůdcích logistice a podobně. Moderní technologie a přístupy postupně umožňují sledování všech těchto informací, problémem ovšem i nadále zůstává izolovanost a kompatibilita těchto datových sad a systémů, které datové sady spravují. Například údaje o výnosu plodiny získané při sklizni sbírá pro zemědělce výrobce strojů provádějících sklizeň, zatímco sběr informací o úhrnu srážek obstarává jiný systém a jejich propojení není triviální. Tato fragmentace dat stěžuje jejich analýzu, která by mohla poskytnout ucelený pohled.

Druhým problémem je nedostatečná standardizace. Data jsou ukládána různými systémy, různými způsoby a v různých formátech. Tento problém zásadně znesnadňuje integraci datových sad do rozsáhlejších systémů a spolupráci různých systémů. Nedostatek standardů také zvyšuje nároky na kvalitu a spolehlivost dat, která musí být ověřována a validována před jejich použitím. Standardizace dat by usnadnila jejich výměnu, porovnávání a kombinování.

Tímto se dostáváme ke třetímu problému a tím je omezení sdílení dat. Nasbíraná data jsou často pouze lokálního rázu a neumožňují pohled v širším měřítku. Pokud by byla data sdílena mezi různými zemědělci, výzkumníky, poradci, úřady a dalšími zainteresovanými stranami, bylo by možné vytvářet přesnější modely a predikce, které by pomohly zlepšit zemědělskou praxi a politiku. Sdílení dat by také podporovalo spolupráci a inovaci v zemědělství.

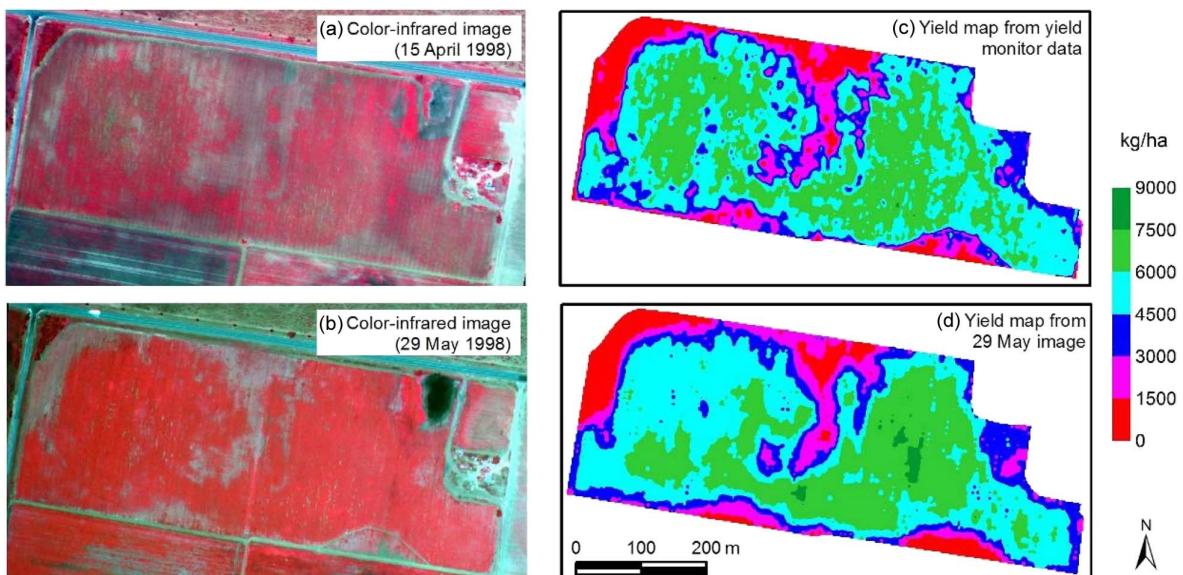
Posledním problémem je také dostupnost vyhodnocovacích systémů. I přes to, že různé senzory a data jsou dostupnější než kdy dřív, je potřeba tato data vyhodnocovat. Toto může být problém pro menší zemědělce, kterým se nevyplatí kvůli drobnému zvýšení efektivity platit vyšší částky za analýzu dat. Tito drobní zemědělci nejsou pro společnosti zabývající se zpracováním těchto dat lukrativní.

1.1.2 Metody sběru dat

Ke sběru dat v zemědělství se primárně používají tradiční techniky, jako jsou terénní průzkumy, odběry vzorků půdy, laboratorní analýzy a experimenty. Tyto metody jsou užitečné pro získání průměrů a rozptylů parametrů plodin a půdy, ale mají některá omezení, jako jsou vysoké náklady, pracnost, spotřeba času a nízké prostorové rozlišení. Již delší dobu se využívají přesné metody sběru dat založeny na nových technologiích, jako je dálkový průzkum země, globální polohový systém (GPS), geografické informační

systémy (GIS) a senzory. Tyto metody jsou schopny lépe zachytit časovou a prostorovou změnu podmínek a stavů a zlepšit produktivitu v zemědělství. [4]

Při dálkovém průzkumu země se získávají informace bez fyzického kontaktu s objektem. Dálkový průzkum lze provádět pomocí různých platform, jako jsou například družice či letadla. Dálkový průzkum poskytuje informace o různých parametrech plodin a půdy v zemědělství. Data pořízená při leteckém či satelitním snímkování se využívají specializovanými algoritmy pro zpracování obrazu či nově umělou inteligencí. Pomocí těchto metod je možné detektovat změny a anomálie a využívat dopady počasí, klimatu a přírodních katastrof a predikovat výnos a zdraví plodin. Satelitní snímkování má však také určitá omezení, jako je dostupnost a kvalita snímků. Není například možné pořizovat snímky při vyšším výskytu oblačnosti. Na obrázku 1.1 je vidět infračervený snímek pole při růstu plodiny a při zrání s odhadem výnosu při sklizni vytvořeným pomocí regresní rovnice z původního snímku. [5]



Obrázek 1.1 Infračervené snímky pole při prvním růstu (a) a při zrání (b) s odhadem výnosu získaným regresní rovnicí z původních snímků (c, d) [5].

Tam, kde není možnost získat data dálkovým průzkumem, se mohou využít senzory, které jsou fyzicky umístěny v místě měření. Senzory jsou zařízení, která detekují a měří fyzikální nebo chemické vlastnosti objektu nebo prostředí v zemědělství. Senzory mohou poskytovat informace o obecných parametrech jako je vlhkost, teplota, úhrn srážek, výška podzemní vody. Zařízení pro měření mohou být také specializovaná a sbírat informace o parametrech, které nelze získat pomocí dálkového průzkumu, jako je pH, živiny, biomasa, chlorofyl a kvalita ovoce. Pomocí senzorů lze řídit zavlažování, hnojení a systémy ochrany proti škůdcům. Senzory však mají také určité problémy, jako je cena, údržba, nutnost kalibrace či bezpečnost přenosu dat. [6]

Data shromažďovaná v administrativě jsou další metodou sběru dat v zemědělství.

Metoda používá úředních a administrativních záznamů a dokumentů z různých zdrojů, jako jsou vládní úřady, statistické úřady, obchodní svazy a soukromé společnosti. Data shromážděná v administraci mohou poskytnout informace o různých aspektech zemědělství, jako je výroba, spotřeba, obchod či ceny. Data shromážděná v administraci však mají také určitá omezení, jako je dostupnost, kvalita či úplnost. Informace sbírané v Evropské unii lze získat například z portálu *Agri-food Data Portal*¹⁾.

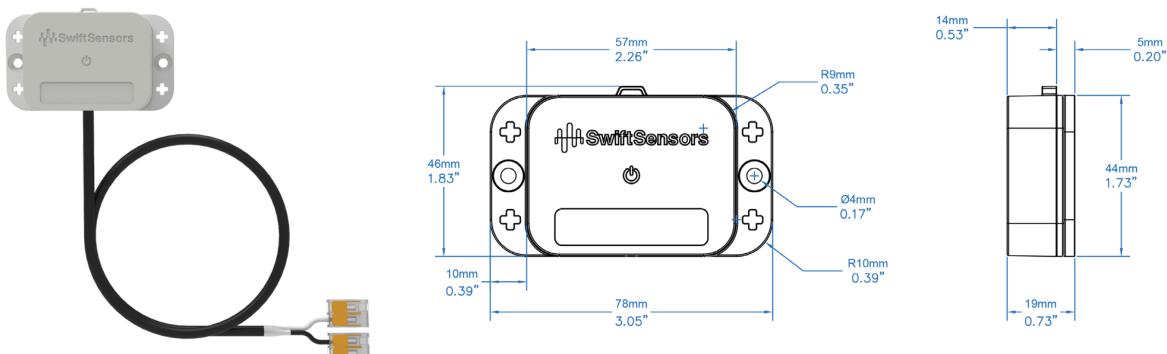
Všechny zmíněné metody mají určitá omezení a měly by být kombinovány a jejich výstupy musí být spojovány pro přesnější výstupy. Kombinace různých metod může zvýšit přesnost, spolehlivost a komplexnost dat a informací. Sběr dat je prvním krokem k analýze a vizualizaci dat, které jsou nezbytné pro vytváření znalostí a náhledů, které mohou zlepšit zemědělský sektor.

1.2 Senzory v zemědělství

Tato podsekce se tedy věnuje aktuálnímu stavu využití senzorů v zemědělství a jejich rozdělení.

Moderní zemědělství již delší dobu silně využívá technologie založené na datech a jejich zpracování. Jak již bylo zmíněno, existuje mnoho způsobů jak sbírat data. Ovšem dominantní metodou je sběr pomocí senzorů. Pomocí senzorů je můžeme data sbírat a vyhodnocovat v reálném čase. To ostatní metody neumožňují. Sběr satelitních či leteckých záběrů je sice možné provádět kontinuálně, ale to by bylo velmi nákladné. Přístup k aktuálním hodnotám kdykoliv, a to v reálném čase, umožňuje využití těchto dat okamžitě - například pro zavlažování, či detekci ohně a podobně.

Na obrázku 1.2 je vidět vnější pouzdro senzoru od společnosti Swift používaný v zemědělství. Společnost nabízí certifikované senzory do všech myslitelných odvětví. Existující řešení této společnosti je popsáno v 1.4.3.



Obrázek 1.2 Příklad vnějšího pouzdra senzoru používaného v zemědělství [7].

Senzory můžeme dělit různými způsoby. Následující podsekce popisuje různé způ-

¹⁾<https://agridata.ec.europa.eu/extensions/DataPortal/home.html>

soby rozdelení senzorů.

1.2.1 Metody získání dat

Rozdelení je založeno na principu měření hodnot. To znamená jak probíhá detekce a jak následně snímač převádí měřený stav na naměřenou fyzikální veličinu.

Mezi způsoby detekce patří elektrická, biologická, chemická a radioaktivní. Elektrická detekce může využívat elektrod, biologická detekce zase biologické složky, chemická detekce chemické reakce a radioaktivní detekce sleduje emise radioaktivního záření.

Dále se rozdelení senzorů prohlubuje prostřednictvím typů konverze, tj. procesu transformace vstupních signálů na výstupní data. Mezi běžné konverzní metody patří následující.

- Fotoelektrická konverze spočívá ve využívání světla k vytváření elektrického signálu, příkladem je fotodioda. Tento senzor může být umístěn ve vinohradu za účelem měření množství slunečního záření.
- Termoelektrická konverze využívá rozdílu teplot k vytváření elektrického napětí. Příkladem termoelektrického senzoru je termoelektrický článek, který generuje elektrický proud na základě teplotního rozdílu mezi dvěma spojenými materiály. Senzor najde využití pro monitorování teploty.
- Elektrochemická konverze zahrnuje chemické reakce, které generují elektrický proud. Příkladem elektrochemického senzoru může být glukózový senzor, který využívá elektrochemického měření pro detekci hladiny glukózy. Tento senzor je možné využít pro monitorování množství cukru v moštu.
- Elektromagnetická konverze využívá změn v elektromagnetickém poli k produkci elektrického napětí. Příkladem senzoru s elektromagnetickým principem může být RFID čtečka, která generuje elektrický signál při působení elektromagnetického pole. Senzor najde využití například při automatizovaném rozpoznávání včelích úlů.
- Termooptická konverze využívá změny v optických vlastnostech materiálu v reakci na teplotní změny. Příkladem termooptického senzoru může být termografická kamera, která vizualizuje teplotní rozdíly pomocí změn v infračerveném záření. Výstupem těchto senzorů mohou být podklady pro interaktivní mapy podobné jako 2.6.

Co se týče způsobu získání dat, může být senzor dále pasivní či aktivní. V případě pasivního senzoru není nutné dodávat energii pro měření, v případě aktivního senzoru

je potřeba energii dodat. Senzor také může být analogový nebo digitální. V případě analogového senzoru je naměřená hodnota udávána jinou fyzikální veličinou (například odporem), kterou je nutné dále převést do digitální podoby pomocí analogově-digitálního převodníku. [8]

1.2.2 Umístění senzorů

Senzory je dále možné dělit podle jejich umístění, a to na statické a dynamické. Statické senzory jsou pevně umístěny na určitém místě a nedochází k jejich pohybu. Oproti tomu dynamické senzory jsou umístěny například na zemědělské technice, jsou tedy v pohybu. Aby bylo možné efektivně využít hodnoty z nich, musí být u každé naměřené hodnoty zaznamenána i aktuální pozice. To se děje hlavně pomocí zaznamenání GPS souřadnic.

1.2.3 Způsob nakládání s daty

Senzory lze rozdělit podle způsobu, jakým nakládají s naměřenými daty. Senzory mohou data ukládat na lokální úložiště či data posílat na nějaký typ brány dále do systému. Tento přístup může být i kombinovaný, kdy zařízení naměřené hodnoty zasílá dále do systému, ale jako zálohu si udržuje hodnoty za určité období.

1.3 Možné využití systému pro sběr dat

Systém pro sběr dat v zemědělství představuje evoluci způsobu plánování a automatizace v zemědělství. Možných případů užití je mnoho, zde je několik reálných situací.

- Efektivní zavlažování - systém umožňuje sledovat a řídit spotřebu vody na základě množství srážek, vlhkosti půdy a stavu plodin.
- Inteligentní zemědělské stroje - na základě dat ze senzorů je možné využívat autonomní zemědělskou techniku s přesně přednastavenými trasami a cílem, jako je například aplikace postríku podle nutnosti v závislosti na zdraví plodin.
- Inteligentní včelí úly - mohou přinést zemědělci informace o nutnosti dokrmení včelstev, či nutnosti stáčení medu.
- Detekce škůdců s preventivním opatřením - senzory ve vinohradech mohou sloužit k detekci škůdců, jako jsou třeba hejna špačků, a na základě detekce spouštět preventivní opatření ve formě zvukových děl.

Způsobů využití je nepřeberné množství. Těmito způsoby může přinést systém pro sběr a zpracování dat v zemědělství skutečnou hodnotu prostřednictvím optimalizace, informovaného rozhodování a monitorování. [9]

1.4 Existující řešení

V současné době neexistuje jednotné řešení, které by zahrnovalo veškeré aplikace v zemědělství, tedy aby bylo by schopné kombinovat veškeré existující systémy a formáty dat. Existuje mnoho řešení a společností, které nabízejí produkty a služby pro různé datové potřeby a aplikace v zemědělství. Tyto produkty a služby využívají různé metody a technologie, jako je satelitní zobrazování, senzory, GPS, GIS, analýzu dat, umělou inteligenci a strojové učení. Každé řešení a společnost má své výhody a nevýhody. Tato podsekce popisuje některé příklady existujících řešení pro vzdálený sběr dat, sběr dat ze senzorů a i existující řešení určené, pro menší zemědělce pro které je systém navržený v této práci zaměřen.

1.4.1 Planet - Data Driven Precision Agriculture

Jedná se o příklad produktu využívající jako zdroj dat satelitní snímkování. Produkt od společnosti Planet poskytuje satelitní snímky s vysokým rozlišením a vysokou frekvencí. Snímky jsou zpracovávány pomocí specializovaných algoritmů a je z nich extrahováno velké množství informací. Mimo jiné poskytuje informace o zdraví plodin a půdy, umožňuje predikovat úrodu či detekovat působení škůdců. Služba poskytuje rozhraní pro integraci a analýzu dat. Služba má přístup k více než 200 satelitům. Cena služby se v roce 2023 pohybovaly od 1,44 amerických dolarů až po 2,88 dolarů za km^2 , v závislosti na vybrané službě. [10]

1.4.2 BeeHero

Produkt BeeHero je příklad úzce specializovaného řešení. Produkt je určen pro včelaře a shromažďuje a analyzuje data z úlů, jako je teplota, vlhkost, zvuk, hmotnost a aktivita. Poskytuje včelařům upozornění a informace v reálném čase. Produkt slouží pro určení optimální doby stáčení medu nebo dokrmení, detekci nemocí a správu včelstev. Systém také umožňuje spojení zemědělců se včelaři pro účely opylení rostlin. Podobně jako 1.4.3 architektura systému je postavena na množství senzorů, bráně a cloutu. Senzory slouží k měření fyzikálních a chemických vlastností uvnitř úlů. Dochází k monitorování teploty, váhy, aktivity, zvuků a vlhkosti. Data jsou bezdrátově přenášena do brány, která podporuje až 100 úlů do vzdálenosti jednoho kilometru. Aktuální stav úlů lze sledovat pomocí mobilní či webové aplikace. Stejně jako předchozí řešení není možnost lokálního nasazení systému. Cena produktu není na stránkách výrobce dostupná. Na obrázku 1.3 je vidět novou generaci senzorů do úlů. [11]



Obrázek 1.3 BeeHero senzor umístěný v úlu [11].

1.4.3 Swift Sensors

Swift Senzors využívá jako jediný zdroj dat data ze senzorů. Systém sleduje pomocí bezdrátových senzorů různé parametry a umožňuje uživatelům přistupovat k datům v reálném čase pomocí webové či mobilní aplikace. Systém dále umožňuje nastavovat výstrahy a upozornění na kritické události a generuje reporty a analýzy. Architektura systému je postavena na rozsáhlém spektru bezdrátových senzorů různých typů, propojených bránou, jež slouží k přenosu dat do cloudu. Výrobce nabízí přes 30 druhů senzorů s cenou od 100 amerických dolarů za kus. Komunikační bránu mezi senzory a cloudem lze pořídit za 330 amerických dolarů. Tato brána je schopna obsloužit až 150 senzorů. Současně je nutné hradit roční poplatek ve výši 65 amerických dolarů za každý senzor. Řešení bohužel neumožňuje lokální nasazení a tak je zemědělec odkázán na cloudové služby výrobce a pravidelné poplatky. Příklad senzoru je možné vidět na obrázku 1.2. Společnost nabízí nepřeberné množství certifikovaných senzorů pro všechna možná odvětví. [7]

1.4.4 Precision AG technology

Jedná se o příklad produktu využívající fúzi dat z více zdrojů. Produkt je vytvořen přímo výrobcem zemědělské techniky John Deere. Systém se skládá z různých nástrojů a funkcí, které umožňují správu dat, automatické navádění zemědělské techniky, plánování a monitorování. Shromažďuje data z různých zdrojů, jako jsou senzory a kamery na zemědělské technice, satelity, drony, GPS, GIS a administrativní dokumenty, a in-

tegruje je do jediné platformy. Může pomoci zemědělcům a agronomům monitorovat, plánovat a optimalizovat jejich zemědělské operace. Tento systém je nejblíže plnohodnotnému systému zahrnujícímu všeobecné aplikace v zemědělství, ale přináší i některé problémy. Jelikož se jedná o systém vytvořený přímo výrobcem zemědělské techniky, který není otevřený, je zemědělec uzamčen v ekosystému daného výrobce a kvůli kompatibilitě zařízení nemůže nakupovat levnější alternativy. Cena osazení jednoho kusu zemědělské techniky takovým systémem se v roce 2023 pohybovala od 2000 amerických dolarů. Dále je potřeba platit pravidelné poplatky za využívání cloudových služeb které provádí zpracování samotných dat. [12]

2 POUŽÍVANÉ TECHNOLOGIE

V prostředí moderního zemědělství dochází k rychlé integraci nových technologií do tradičních metod sběru, zpracování a vizualizace dat. Tyto nové technologie přináší zvýšení přesnosti, efektivity a udržitelnosti v zemědělství. Nové technologie se neustále vyvíjí a tudíž téměř každý den vznikají nové zařízení, systémy či další vylepšení stávajících technologií, umožňující lepších výsledků, než bylo doposud dosažitelné. Systémy využívané ke sběru a zpracování dat v zemědělství lze označit za systémy internetu věcí, jedná se totiž o propojené sítě sledující fyzické objekty.

Tato sekce je věnována popisu nejčastěji používaných technologií v systémech pro sběr, vyhodnocení a vizualizaci dat jako takových, se zaměřením na zemědělství. Sekce obsahuje popis architektury a částí, ze kterých se systémy skládají, a to od jednotlivých zařízení přes komunikační protokoly a zabezpečení až po vizualizační aplikace. Sekce dále obsahuje teoretický popis technologií v systému pro sběr a zpracování dat v zemědělství navrženém a implementovaném v rámci této diplomové práce.

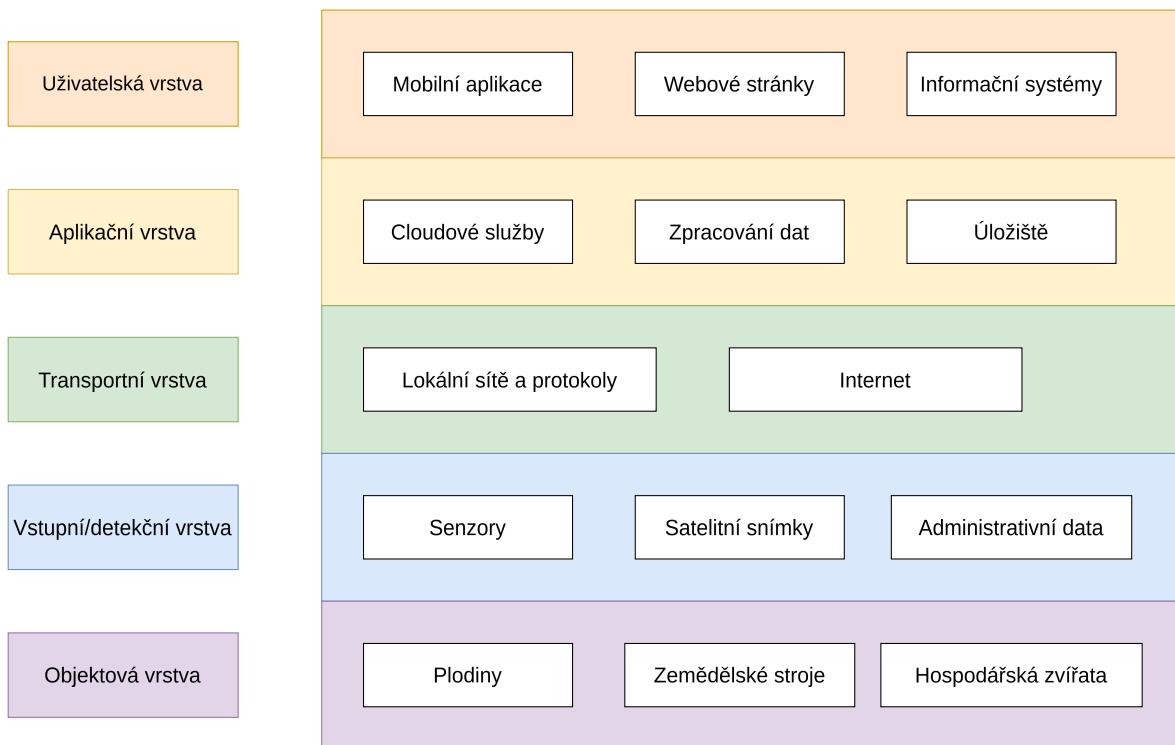
2.1 IoT v zemědělství

IoT neboli internet věcí jsou systémy umožňující propojení fyzických zařízení a senzorů do jedné sítě. Termín IoT v zemědělství tedy zastřešuje právě zmíněné systémy. IoT v zemědělství zahrnuje široké spektrum systémů specializovaných na sběr, zpracování a sdílení informací. Tyto systémy se skládají z několika klíčových prvků.

- Sledované fyzické objekty, mezi které patří plodiny, zemědělské stroje, hospodářská zvířata, půda a další prvky, které jsou sledovány v rámci zemědělství.
- Monitorovací zařízení, jako jsou senzory a detekční systémy, které monitorují fyzické objekty a shromažďují relevantní data.
- Přenosové protokoly, které slouží k propojení monitorovacích zařízení a umožňují přenos dat od senzorů až k uživateli.
- Brány, které jsou klíčové pro propojení celé sítě monitorovacích zařízení s internetem a dalšími částmi systému.
- Programy pro zpracování dat, které hrají klíčovou roli ve zpracování a interpretaci dat, která jsou následně využita pro rozhodování.
- Úložiště, sloužící k ukládání naměřených hodnot a zpracovaných dat, poskytující historický kontext a analýzu.
- Aplikace pro vizualizaci dat, které poskytují uživatelům přehledné a intuitivní rozhraní k vizualizaci a interpretaci informací.

- Uživatelé využívají data a informace poskytované IoT systémem k informovaným rozhodnutím a řízení aktivit.

Prvky systému IoT v zemědělství můžeme rozdělit do několika vrstev. Vizualizované vrstvy je možné vidět v diagramu 2.1. První vrstvou je vrstva fyzických objektů. Následuje vrstva detekční, do které spadají senzory, detekční a vstupní systémy. Následuje vrstva přenosová, která je zodpovědná za propojení všech částí systému, od jednotlivých senzorů až po cloudové aplikace. Poté přichází na řadu vrstva aplikační obsahující aplikace na zpracování dat, úložiště a aplikace pro zobrazování. Poslední vrstvou je vrstva uživatelská sloužící pro předání informací uživatelům, ta obsahuje například mobilní či webové aplikace. [9]



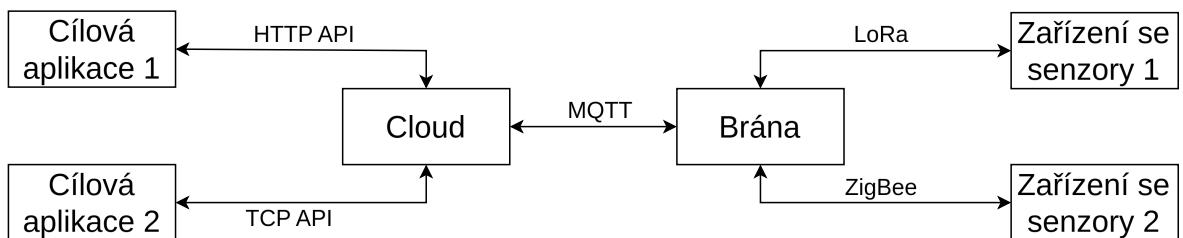
Obrázek 2.1 Vrstvy IoT systémy v zemědělství [9].

2.2 Komunikační protokoly

Tato sekce popisuje komunikační protokoly používané v IoT systémech v zemědělství. Komunikační protokoly popsáne v této sekci pokrývají široké spektrum funkcionality. Ve většině případů nelze zvolit jeden protokol, který by mohl pokrýt veškeré případy užití v celém systému. Každý protokol přináší určité výhody a nevýhody a vždy je potřeba zvolit vhodný protokol na základě požadavků na systém a jeho komunikaci. Je tedy běžné, že systém IoT může využívat celou škálu protokolů pro různé účely, a to i z důvodu kompatibility se zařízeními a systémy od různých výrobců. Pokud se

systémy skládají ze součástí s různými účely a od různých výrobců, je pravděpodobné, že množství využitých komunikačních protokolů velmi rychle vzroste.

Jako příklad poslouží jednoduchý systém z diagramu 2.2. Systém obsahuje dvě zařízení se senzory, bránu, cloudovou část a dvě cílové aplikace. Systém pro své fungování potřebuje dvě zařízení se senzory od různých výrobců. První zařízení využívá pro komunikaci protokol LoRa a druhý ZigBee. Pro správné fungování je tedy potřebná brána která umí přijímat oba protokoly. Brána dále posílá získaná data do cloudové aplikace přes internet pomocí protokolu MQTT. Data si lze prohlížet ve dvou cílových aplikacích. Každá z cílových aplikací je již existující systém s daným rozhraním, které využívá různé protokoly. Z toho vyplývá, že i takto jednoduchý systém využívá kvůli požadavkům minimálně 5 protokolů běžných v IoT.



Obrázek 2.2 Příklad jednoduchého systému s různými komunikačními protokoly.

Při volbě komunikačního protokolu je také nutné uvědomit si, jaké vrstvy referenčního modelu ISO/OSI protokol pokrývá. Zde je důležité zmínit, že se jedná o jiné rozdělení vrstvy IoT z obrázku 2.1. Některé protokoly mohou pokrývat pouze jednu vrstvu, zatímco některé více vrstev. Výsledný systém může kombinovat různé protokoly na různých vrstvách. Vrstvy a zástupci protokolů často se vyskytujících v IoT jsou zobrazeny v tabulce 2.1. Při přenosu dat je také důležité dbát na zabezpečení, data je potřeba zabezpečit proti zneužití. Některé protokoly mají určitou úroveň zabezpečení vestavěnou, jiné nikoli. Popis zabezpečení i z pohledu protokolů je popsán v podsekci 2.6.

OSI model	Protokoly internetu včetně
7. Aplikační vrstva	
6. Prezentační vrstva	HTTPS, XMPP, CoAP, MQTT, AMQ
5. Relační vrstva	
4. Transportní vrstva	UDP, TCP
3. Síťová vrstva	IPv6, 6LoWPAN, RPL
2. Linková vrstva	
1. Fyzická vrstva	IEEE 802.15.4, Wifi, Ethernet, GSM, LoRa

Tabulka 2.1 Protokoly používané v IoT na modelu ISO/OSI [13].

Komunikační protokoly pokrývající první tři vrstvy modelu ISO/OSI je vhodné rozdělit na protokoly bezdrátové a protokoly vyžadující kably pro komunikaci. V IoT se

při komunikaci s branami nejčastěji bavíme o bezdrátových protokolech, které lze dále rozdělit do tří skupin:

- protokoly lokálního charakteru určené pro komunikaci na kratší vzdálenosti, jako je například Bluetooth, WiFi či Zigbee,
- nové protokoly navržené pro komunikaci na delší vzdálenosti zvané LPWAN, zástupcem je LoRa,
- mobilní a satelitní sítě.

Každý komunikační protokol, který je v této sekci probrán, nese své vlastní výhody a nevýhody. Některé jsou vhodnější pro rychlé přenosy dat v reálném čase, zatímco jiné se mohou vynikajícím způsobem hodit pro přenos objemnějších dat nebo pro operace, kde je klíčová nízká spotřeba energie. Výběr správného komunikačního protokolu je klíčovým faktorem pro stabilitu a funkčnost systému.

2.2.1 LoRa

Technické informace z této podsekce jsou čerpány z [14] a [15]. Protokol LoRa neboli *Long Range* je bezdrátový komunikační protokol vyvinutý pro komunikaci na větší vzdálenosti. Protokol byl určen již z pravopřátku vývoje pro zařízení internetu věcí umístěných na nedostupných místech. Tato zařízení jsou často závislá na napájení z bateriových článků a proto je protokol a jeho moduly navržen tak, aby byl co nejméně energeticky náročný. LoRa umožňuje zařízením vysílat a přijímat data na vzdáleností více než 10 kilometrů v ideálních podmírkách. Protokol je vhodný pro aplikace, které vyžadují nízké zpoždění a je pro ně dostatečná nízká přenosová rychlosť.

LoRa pokrývá vrstvy 1 až 3 modelu OSI, přesněji stejnojmenná přenosová část protokolu pokrývá fyzickou a linkovou vrstvu a technologie LoRaWAN pokrývá vrstvu síťovou. Technologii fyzické přenosové vrstvy zastřešuje společnost Semtech, zatímco technologie LoRaWAN je pod správou aliance výrobců zvané LoRa Alliance.

Fyzická vrstva protokolu využívá patentovanou metodu modulace dat s rozprostřeným spektrem do elektromagnetických vln. Komunikační síť se skládá z komunikačních bran a komunikujících zařízení a využívá síťovou topologii hvězdy, kdy každé zařízení komunikuje přímo s bránou. Protokol využívá nelicencovaná pásmo, není tudíž nutné platit licenční poplatky za využívání komunikačního pásmo. Pásma je zvoleno podle oblasti využití, protože různé státy umožňují nelicencované použití jiných pásem. Zároveň tato pásmo mají omezení, tzv. pracovní cyklus (*duty cycle*), který říká po jakou dobu může zařízení využívat pásmo k přenosu dat. Pásma využívaná technologií LoRa ve světě s omezením pracovního cyklu je možné vidět v tabulce 2.2. Z tabulky je patrné,

že například v Evropě v pásmu $868 - 868.6\text{MHz}$ lze využívat pracovní cyklus 1 %, což znamená že zařízení může být aktivní (vysílat zprávy) pouze 1 % času.

Oblast	Frekvence (MHz)	Pracovní cyklus (%)
Evropa	863 - 865	0.1
Evropa	865 - 868	1
Evropa	868 - 868.6	1
Evropa	868.7 - 869.2	0.1
Evropa	869.4 - 869.65	10
Evropa	869.7 - 870	1
USA	902 - 928	1
Čína	779 - 787	1
Čína	470 - 510	1
Austrálie	915 - 928	1

Tabulka 2.2 Frekvenční pásma LoRa s povolenými pracovními cykly [16].

Jak již bylo zmíněno, LoRa využívá metodu rozprostřeného spektra k modulaci, a frekvenční modulaci signálu ke kódování dat. Přenosovou jednotkou je jedno slovo neboli *chirp*. Jedno slovo je signál, jehož frekvence se mění s časem v jednom směru. Buď dochází ke zvyšování, nebo ke snižování frekvence. Data jsou zakódována právě v těchto slovech. Součástí protokolu je parametr *Spreading factor (SF)* který udává množství dat kódovaných v jednom slově. Hodnota parametru SF má tedy vliv na přenosové rychlosti. Hodnota SF má ovšem také vliv na dosah zařízení. Platí, že čím vyšší přenosovou rychlosť zařízení využívá, tím nižšího dosahu kvůli různému rušení může dosáhnout.

Je důležité brát v potaz omezení pracovního cyklu. Pokud máme zařízení vysílající zprávy přes protokol LoRa, jsme omezeni v počtu zpráv, které můžeme za den odeslat. Je důležité vědět kolik zpráv můžeme poslat. Množství zpráv závisí na velikosti zprávy, zvoleném SF a šířce pásma. Pro výpočet využijeme časové období jednoho dne, délku datové části zprávy 64 bajtů, šířku pásma 125 kHz a pracovní cyklus z evropského pásma 868 - 868.6 MHz o velikosti 1 %.

Pro výpočet počtu zpráv potřebujeme znát dobu odeslání jedné zprávy značené jako T_{packet} a dobu, po kterou za dané období můžeme využívat pásmo k přenosu dat. Pokud vezmeme tedy jedno procento dne, máme k dispozici 864 vteřin k přenosu. Dobu nutnou k přenosu jedné zprávy můžeme vypočítat pomocí vzorce 2.1 kde $T_{preamble}$ je čas potřebný k odeslání hlavičky paketu a $T_{payload}$ je čas potřebný k odeslání datové části [17].

$$T_{packet} = T_{preamble} + T_{payload} \quad (2.1)$$

Doba potřebná k odeslání hlavičky zprávy se počítá podle vzorce 2.2, kde T_{sym} je čas potřebný k odeslání jednoho slova a $n_{preamble}$ je velikost hlavičky zprávy [17].

$$T_{preamble} = (n_{preamble} + 4, 25) * T_{sym} \quad (2.2)$$

Doba potřebná k odeslání datové části zprávy se počítá podle vzorce 2.3, kde T_{sym} je čas potřebný k odeslání jednoho slova a $n_{payload}$ je velikost datové části zprávy [17].

$$T_{payload} = n_{payload} * T_{sym} \quad (2.3)$$

Doba k odeslání jednoho slova se vypočítá podle vzorce 2.4, kde SF je *Spreading Factor* a BW je šířka pásma, která může být 125 kHz, 250 kHz nebo 500 kHz [17].

$$T_{sym} = \frac{2^{SF}}{BW} \quad (2.4)$$

Velikost datové části správy v bitech se vypočítá podle vzorce 2.5. Ve vzorci představuje proměnná PL samotnou délku dat, SF představuje *Spreading Factor*, H indikuje implicitní či explicitní hlavičku, DE indikuje přítomnost optimalizace datového přenosu a CR představuje položku *Coding rate* což je hodnota určující množství redundantních bitů [17].

$$n_{payload} = 8 + \max(\text{ceil}(\frac{8PL - 4SF + 28 + 16 - 20H}{4 * (SF - 2DE)} * (CR + 4), 0)) \quad (2.5)$$

Pokud zvolíme hodnotu *Coding Rate* jako 4/5 (4 nosné bity a jeden redundantní) hodnotu implicitní hlavičky na 0 a optimalizaci datového přenosu 0 tak následný výpočet délky přenosu vypadá následovně:

$$T_{sym} = \frac{2^8}{125000} = 0,002048s$$

$$T_{preamble} = (8 + 4, 25) * 0,002048 = 0,025088s$$

$$n_{payload} = 8 + \max(\text{ceil}(\frac{8 * 64 - 4 * 8 + 28 + 16 - 20}{4 * (8 - 0)} * (4 + 4), 0)) = 80b$$

$$T_{payload} = 0,002048 * 80,5 = 0,180224s$$

$$T_{packet} = 0,025088 + 0,180224s = 0,205311$$

$$n_{mpd} = 864/0,205311 = 4208$$

Z výpočtu vyplývá, že doba nutná k odeslání jedné zprávy T_{packet} je 205 milisekund a pokud chceme plně využít omezení 1 % můžeme za den z daného zařízení odeslat 4208 zpráv (hodnota n_{mpd}).

LoRaWAN definuje tři třídy koncových zařízení: třídu A, třídu B a třídu C. Všechna zařízení LoRaWAN musí implementovat třídu A, zatímco třída B a třída C jsou rozšířením specifikace zařízení třídy A. Všechny třídy zařízení podporují obousměrnou komunikaci.

- Třída A nabízí energeticky nejúspornější režim. Zařízení může kdykoliv odeslat zprávu a po odeslání zprávy otevří dva sloty pro příjem odpovědi ze serveru. Pokud server nedokáže odpovědět po danou dobu, musí počkat na další zprávu vyslanou zařízením, než je zařízení schopno přijímat další zprávy - další zpráva otevře další dva sloty pro příjem. Tato třída se používá u zařízení které nevyžadují pravidelné aktualizace ze strany serveru.
- Třída B rozšiřuje třídu A o pravidelné otevřívání slotů pro příjem zpráv. Tato třída pro svou funkci vyžaduje více energie. Využívá se u zařízení, které vyžadují pravidelné zprávy ze serveru, například kvůli získání nové verze softwaru ze strany serveru.
- Třída C ponechává slot pro příjem zpráv neustále otevřený pokud nedochází k odeslání zprávy. Třída se využívá pro zařízení vyžadující rychlou reakci, jako jsou například spínače.

Výhodami protokolu LoRa jsou: možná komunikace na velké vzdálenosti, nízká spotřeba energie, dobrá průchodnost překázkami, množství zařízení připojených k síti, nízká cena. Nevhodami jsou: omezená přenosová rychlosť, limitace délky zprávy, vysoká odezva, nemožnost komunikace v reálném čase a zákonem limitované využití přenosového pásma.

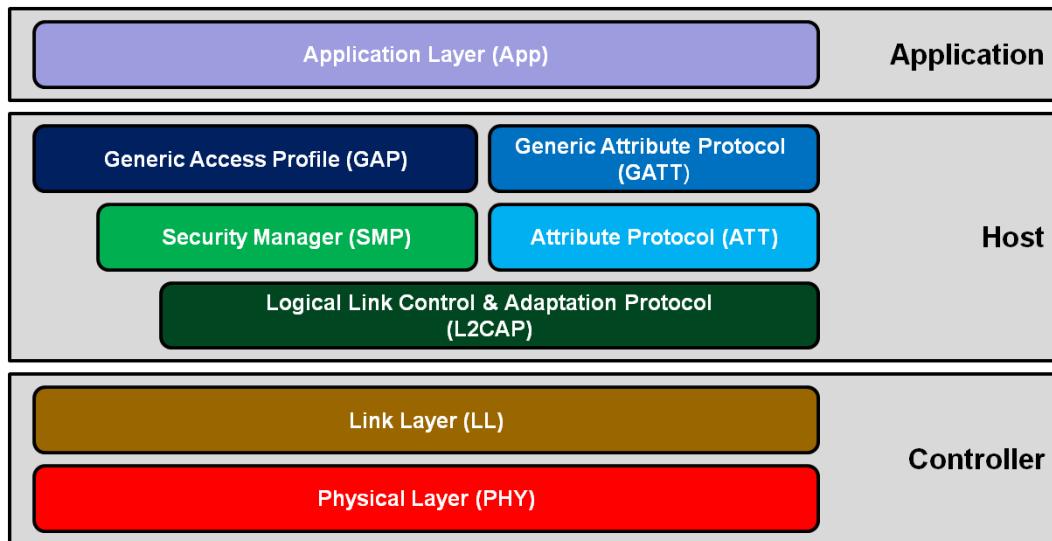
2.2.2 Bluetooth Low Energy

Teoretické informace z této sekce jsou čerpány z [18]. *Bluetooth Low Energy* (BLE) je bezdrátová komunikační technologie a je založena na protokolu *Bluetooth*. BLE spadá do sekce protokolů lokálního charakteru, umožňuje přenos dat na krátké vzdálenosti s nízkou spotřebou. BLE je navrženo pro aplikace, které vyžadují přerušovaný přenos dat a nepotřebují velkou šířku pásma nebo nepřetržité připojení.

BLE pracuje ve stejném frekvenčním pásmu 2,4 GHz jako klasické Bluetooth, ale využívá jiné modulační schéma a jednodušší strukturu paketů. BLE podporuje více

síťových topologií, jako je hvězda, mesh a broadcast. Zařízení BLE lze rozdělit do dvou typů: centrální a periferní. Centrální zařízení zahajuje a řídí komunikaci s jedním nebo více periferními zařízeními. Periferní zařízení oznamují svou přítomnost centrálnímu zařízení a čekají na připojení. Centrální zařízení může také vyhledávat a přijímat zprávy z periferních zařízení bez navazování spojení.

BLE využívá pro komunikaci tzv. *Generic Attribute Profile* (GAP) k definování struktury, výměny a přístupu k datům. *Generic Attribute Protocol* a *Attribute Protocol* (ATT) jsou dvě vrstvy, které pomáhají zařízením ukládat a vyměňovat data. ATT definuje základní jednotku dat (atribut) a způsob jejich přenosu mezi zařízeními. Atribut má jedinečný identifikátor, hodnotu a některé vlastnosti a oprávnění. Atribut může například ukládat stav baterie zařízení a mít vlastnost, která umožňuje ostatním zařízením jej číst. GATT organizuje atributy do služeb a charakteristik a definuje pravidla a operace pro komunikaci. Služba je skupina souvisejících atributů, které poskytují určitou vlastnost nebo funkci. Charakteristika je atribut, který představuje část dat nebo parametr v rámci služby. Deskriptor je atribut, který poskytuje další informace nebo konfiguraci pro charakteristiku. Zařízení může mít například službu srdeční frekvence, která obsahuje charakteristiku měření srdeční frekvence a deskriptor konfigurace klientské charakteristiky. GATT a ATT zajišťují, že zařízení mohou vzájemně komunikovat konzistentním a bezpečným způsobem. Architekturu protokolu BLE je možné vidět na obrázku 2.3.



Obrázek 2.3 Architektura protokolu BLE [18].

BLE je vhodný pro přenos malého množství dat informujících o stavu či událostech ze zařízení s nízkou spotřebou energie, jako jsou senzory nebo čidla. Výhodami BLE jsou: nízká spotřeba energie, cena a podpora je již vestavěna do zařízení využívající *Bluetooth*. Nevýhodami BLE jsou hlavně limit velikosti zpráv a nízký dosah.

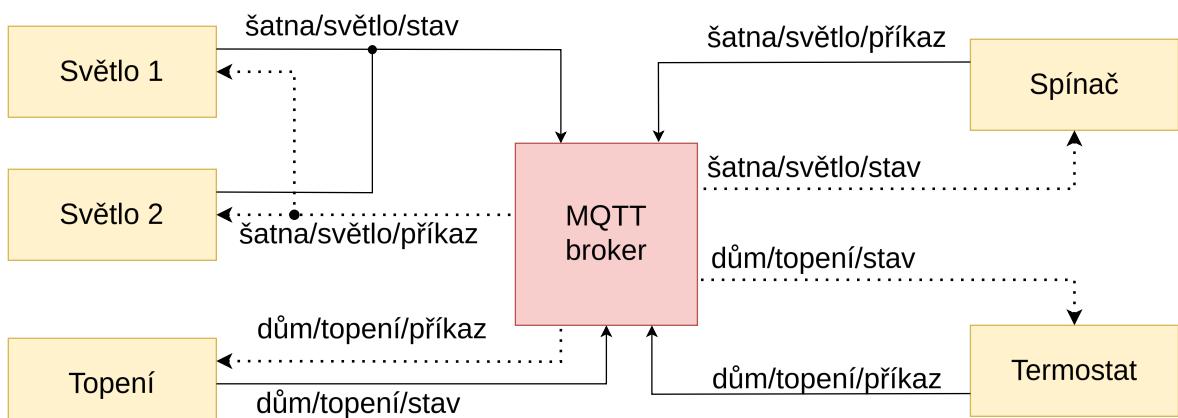
2.2.3 MQTT

Technické informace této podsekce jsou čerpány z [19]. MQTT neboli *Message Queuing Telemetry Transport* je jednoduchý komunikační protokol navržený primárně pro internet věcí. Protokol pracuje na principu publikování/odběru zpráv. Tento princip umožňuje dynamicky kombinovat modely komunikace typu jeden na jednoho, jeden na více a více na jednoho. Protokol MQTT primárně využívá pro přenos dat protokol TCP, ale je možné využívat i spojení přes UDP.

Určitá zařízení se chovají jako vydavatelé zpráv pro určité téma (v názvosloví MQTT tzv *topic*). Zprávy zasílají na zařízení, které funguje jako MQTT broker. MQTT broker dále zprávy rozesílá zařízením, která o dané téma mají zájem.

MQTT má jednoduchou strukturu paketů a jednoduché chování, což umožňuje snadnou implementaci. To je důležité pro zařízení s nízkým výkonem používaných v IoT. Protokol nabízí tři úrovně QoS neboli *Quality of Service* při doručení zpráv. Tyto úrovně jsou:

- úroveň 0 - zpráva je doručena maximálně jednou, je možné že zpráva není nikdy nikomu doručena,
- úroveň 1 - zpráva je doručena minimálně jednou, je možné že zpráva je doručena víckrát,
- úroveň 2 - zpráva je doručena právě jednou, dochází k ukládání zpráv dokud nejsou doručeny.



Obrázek 2.4 Příklad jednoduchého zapojení zařízení do sítě MQTT.

Příklad systému využívající MQTT je možné vidět na diagramu 2.4. V reálném systému na diagramu je například chytrý spínač, který je připojen k centrální MQTT broker aplikaci a publikuje zprávy na téma “šatna/světlo/příkaz“, v síti se dále nachází

inteligentní žárovka, která odebírá zprávy z tématu “šatna/světlo/příkaz“. Tímto způsobem lze zasílat žárovce příkazy k zapnutí či vypnutí bez nutnosti přímého spojení. Na stejném principu fungují v ukázkovém systému zařízení termostatu a topení. Termostat podle nastavené teploty publikuje příkazy pro topení, které je přijímá a zpět reportuje stavové informace. Z příkladu vyplývá, že se téma skládá z posloupnosti znaků se separátorem v podobě lomítka. Jednotlivé části představují úrovně a využívají se například jako v našem příkladu k určení umístění zařízení. Díky tomuto protokolu je možno ovládat i více zařízení najednou a lze provádět komunikaci i oběma směry. Zařízení je tedy schopné nejen přijímat příkazy, ale i odesílat svůj stav. Veškerá spojení závisí pouze na tom, na které téma dané zařízení publikuje a z kterého odebírá zprávy.

Výhodami MQTT jsou: možnost volby QoS, využití modelu publikování/odběru, jednoduchost. Nevýhodami MQTT jsou využití TCP komunikace, úzkého místa v podobě brokera (kvůli zahlcení sítě či zotavení při výpadku) [20].

2.2.4 Mobilní síť

Sítě LPWAN nejsou omezeny pouze na vybudování zcela nové infrastruktury přenosových vrstev. Vybudování nové infrastruktury s cílem co největšího pokrytí je velmi drahé. Některé společnosti a protokoly jako je LoRaWan se spoléhají na sdílení bran protokolu a tím dosažení vyššího pokrytí. Některé technologie ovšem vznikly s úmyslem využití již stávající infrastruktury mobilních operátorů. Hlavními zástupci těchto LPWAN protokolů jsou NB IoT a LTE-M. Využití již existující infrastruktury nabízí enormní úsporu nákladů, efektivně řeší otázku bezpečnosti a zároveň otevírá nové možnosti mobilním operátorům nabízet služby IoT. Protokoly operují v licencovaných pásmech.

NB IoT je bezdrátová technologie s nízkým odběrem a přenosem na velké vzdálenosti, tedy zástupce protokolů typu LPWAN. Technologie byla vyvinuta skupinou 3GPP a umožňuje zařízením využívat existující infrastrukturu mobilních operátorů pro přenos dat. Přesněji využívá technologii GSM, případně nepoužívané kanály v rámci LTE. NB IoT si dává za cíl zvětšit dosah stávajících mobilních sítí za cenu opakování zaslání zpráv, nižších rychlostí a vyšší odezvy. Cílem protokolu je také snížit spotřebu energie, díky čemuž zařízení využívající tento komunikační protokol může fungovat na baterii i několik let. Protokol je schopný koexistovat se sítěmi 2G, 3G, 4G, LTE-M i 5G a je součástí standardu 5G. K přenosu dat využívá úzké pásmo o frekvenci 200 kHz. Největšími výhodami NB IoT je malý odběr, dobrá penetrace signálu skrz stěny, koexistence více sítí v jedné oblasti díky využití úzkého pásma, nízká cena. Hlavními nevýhodami jsou nízké rychlosti, vysoká odezva a špatná podpora pro pohyblivé objekty. [21]

Další dnes již běžně rozšířenou technologií je LTE-M. Jedná se o síť LTE pro stroje. Technologie je součástí standardu 5G. V porovnání s technologií NB IoT nabízí desetkrát vyšší rychlosť, která je až až 1 $Mbps$. Odezva je také řádově nižší a pohybuje se v rozmezí 10 – 15 ms, přičemž u NB IoT sítě se pohybuje v řádu vteřin. Technologie umožňuje zařízením s nízkým odběrem proudu přímé připojení na existující 4G síť. Technologie využívá mnohem širší pásmo 1,4 – 5 MHz.

Technologii NB IoT nabízí v České republice mobilní operátor Vodafone a O2, mobilní operátor T-Mobile v České republice poskytuje IoT služby pomocí protokolu SigFox [22].

IoT application considerations				
	4G LTE (CAT 1+)	LTE-M (CAT M1 AND CAT M2)	NB-IoT	LoRa
Data throughput (Mbps)	██████	██████	██████	██████
Extra-long battery life	🔋	🔋	🔋	🔋
Mobility management	✓	✓	✗	✓
Remote update capabilities	✓	✓	✓	✓
Voice	✓	✓	✗	✗
Extended coverage	✓	✓	✓	✓
Latency	Low	Low-medium	Medium	Medium
Radio cost	\$\$\$	\$	\$	\$\$
Time to deploy	⌚	⌚	⌚	⌚
Part of the 5G standard				Operates in unlicensed spectrum

Obrázek 2.5 Porovnání LPWAN protokolů s 4G [21].

Na obrázku 2.5 můžeme vidět porovnání protokolů a to: 4G, který není určen pro IoT, LTE-M, NB IoT a LoRa. Porovnání ukazuje dominanci 4G LTE technologie, která má oproti ostatním protokolům obrovskou nevýhodu, a to vysokou spotřebu energie a vysokou cenu. Pro velké množství bateriově napájených zařízení je tato technologie nevhodná. Z porovnání je patrná dominance LTE-M vůči NB IoT. Technologie LTE-M umožňuje nasazení u pohyblivých zařízení. Největší výhodou sítě LoRa je využití neliencovaných pásem. Pokud jsou dodržena omezení na využití sítě, není pro komunikaci přes síť LoRa nutné platit žádné poplatky.

2.3 Zpracování dat

Technické informace z této sekce jsou čerpány z [23]. Zpracování dat je zásadní částí systémů pro sběr dat. Senzory a zařízení jimi osazené mohou generovat tisíce hodnot za relativně krátkou dobu. Když vezmeme v potaz, že takových zařízení může být v systému zapojeno desítky, stovky, či až tisíce, dochází ke generování obrovského množství dat. Tato data v nezpracované podobě není možné využít bez předzpracování, a to nejen kvůli samotnému množství, ale i případné chybovosti. Zpracování dat je kritické pro možnosti následné prezentace výsledků koncovému uživateli. Zpracování dat se skládá z následujících kroků.

- Čistění dat, při kterém dochází k odstranění všech irrelevantních bodů z datové sady. Například pokud senzor teploty při své inicializaci naměří hodnotu výrazně vyšší než hodnoty následující, je vhodné tuto hodnotu odstranit.
- Agregace dat, při které dochází ke kombinaci více datových bodů do jednoho bodu. Například pokud máme sto hodnot jdoucích za sebou se stejnou hodnotou stačí nám pouze první a poslední pro zachování informace.
- Normalizace dat - jedná se o škálování dat do společného rozsahu za účelem zjednodušení srovnání.
- Analýza dat, která slouží k extrahování poznatků za pomocí statistických technik či strojového učení. Takto je možné například identifikovat vzorce v růstu plodin a následně optimalizovat hnojení a zavlažování.

Následuje krok vizualizace dat který je popsáný v sekci 2.5.

2.3.1 Čištění a normalizace dat

Data sbíraná v zemědělství pochází z mnoha zdrojů, jako jsou senzory, satelitní snímky či manuální záznamy. Tyto zdroje poskytují data různých typů a mnohdy i rozdílné kvality. Krok čištění dat je zásadní pro následné správné vyhodnocení dat, zanechání vadných dat může mít totiž zásadní vliv na výsledné analýzy. Před jakýmkoli prvotním zpracováním dat je tedy nutné provést jejich čištění. V tabulce 2.3 je příklad nevyčištěné datové sady, na které si ukážeme typy chyb a jejich čištění. Datová sada obsahuje hodnoty naměřené zařízením se senzory teploty, vlhkosti a úrovně oxidu uhličitého umístěném ve sklepě. Čištění dat je proces zahrnující dva kroky. [24]

Prvním krokem je detekce vadných hodnot. Chyb v datasetech může být mnoho a různých typů. Mezi nejčastější problémy patří duplicitní hodnoty, extrémy, překlepy, číselné hodnoty uložené v textové podobě, chybějící hodnoty, směs datových typů, špatné

ID	Místo	Čas (Unix)	Teplota (°C)	Vlhkost (%)	CO2 (ppm)
1	Sklp	1705138819	16,4	80	556
2	Sklep	1705138820	99	80	556
3	Sklep	13.1.2024 09:40:20	16,5		556
4	Sklep	1705138822	16,4	80	5a7
5	Sklep	1705138823	16,4	81	557
6	Vinice	1605138823	33,5	16	421
7	Sklep	1705138823	16,4	81	557

Tabulka 2.3 Příklad nevyčištěného datasetu.

formátování a podobně. Detekci je možné provést automatizovaně či manuálně. Manuální detekce dává smysl pouze u menšího množství dat a dnes se již téměř nepoužívá. Automatická detekce může probíhat prostou kontrolou přednastavených povolených hodnot či složitějšími algoritmy sledujícími i trendy v datech.

V příkladu datasetu 2.3 lze provést manuální kontrolu, jelikož se jedná o příklad s velmi malým množstvím hodnot. Problémy datové sady jsou následující:

1. řádek 1 obsahuje překlep v hodnotě *Sklp* při porovnání s ostatními záznamy,
2. řádek 2 obsahuje extrémní hodnotu teploty,
3. na řádku 3 je hodnota času v jiném formátu, než je dáno, a chybí zde údaj vlhkosti,
4. na řádku 4 je nevalidní hodnota úrovně CO2,
5. řádek číslo 6 je záznam z jiného místa,
6. řádky číslo 5 a 7 jsou duplicitní,

Druhým krokem je oprava či odstranění vadných záznamů z datového setu. Metody tohoto kroku je vhodné prezentovat na příkladu transformace nevyčištěného datasetu 2.3 na dataset 2.4 který obsahuje vyčištěné záznamy. V případě prvního problému lze předpokládat překlep na základě ostatních záznamů a hodnotu *Sklp* opravit na *Sklep*. Druhý problém je řešitelný odstraněním záznamu nebo jeho úpravou. V tomto případě byla zvolena úprava teploty zprůměrováním předchozí a následující hodnoty. Vada číslo 3 je vyřešena transformací hodnoty *13.1.2024 09:40:20* na ekvivalentní hodnotu Unixové časové značky. Nevalidní hodnota úrovně oxidu uhličitého je opravena na základě okolních hodnot. Řádek číslo 6 je odstraněn, protože hodnoty byly naměřeny na jiném místě a pravděpodobně patří do jiné datové sady. Řádek číslo 7 byl z důvodu duplicitnosti odstraněn.

ID	Místo	Čas (Unix)	Teplota (°C)	Vlhkost (%)	CO2 (ppm)
1	Sklep	1705138819	16,4	80	556
2	Sklep	1705138820	16,45	80	556
3	Sklep	1705138821	16,5	80	556
4	Sklep	1705138822	16,4	80	557
5	Sklep	1705138823	16,4	81	557
7	Sklep	1705138823	16,4	81	557

Tabulka 2.4 Příklad datasetu po vyčištění.

Díky krokům detekce a opravy chyb v datasetu 2.3 je možné pokračovat v dalších krocích datové analýzy.

Normalizace datové sady je transformace dat do standardního měřítka a distribuce, takže je lze snáze a přesněji porovnávat a analyzovat. Normalizace dat může zlepšit výkon a přesnost různých algoritmů a modelů, protože snižuje vliv měřítka a umístění dat a činí je homogennějšími a symetrickými. [24]

2.3.2 Analýza dat

Technické informace z této podsekce vychází z [25]. Analýza dat v IoT je proces vyhodnocení dat pocházejících z různých senzorů. Pro analýzu se používají různé nástroje a techniky. Analyzovaná data mohou sloužit k mnoha účelům, od k upozornění uživatele na nebezpečné situace až po dlouhodobé předpovědi. Stejně jako data slouží k mnoha účelům, lze je i analyzovat nepřeberným množstvím způsobů. Přístupy k analýze dat z IoT jsou deskriptivní, diagnostické, prediktivní a preskriptivní.

Deskriptivní analýza provádí shrnutí a vizualizaci nezpracovaných dat za účelem získání přehledu o jejich charakteristikách. V IoT to může znamenat vytváření řídících či vizualizačních panelů, které zobrazují hodnoty senzorů v reálném čase, historické trendy nebo statistické souhrny. Pochopením distribuce dat můžeme identifikovat vzorce a anomálie v datech.

Diagnostická analýza má za cíl odhalit příčiny pozorovaných událostí. Když hodnota senzoru překročí předem definovanou prahovou hodnotu, diagnostická analýza nám pomůže zjistit, proč k tomu došlo. Například zda šlo o nefunkční senzor, faktory prostředí nebo skutečnou anomálii.

Prediktivní analýza využívá historická data k vytváření informovaných předpovědí budoucích událostí. Využívají se algoritmy strojového učení, které nám umožňují předpovídat hodnoty senzorů, detektovat hrozící selhání nebo odhadnout spotřebu zdrojů. Pro prediktivní úlohy se běžně používají prognózy časových řad, regresní modely a neuronové sítě.

Preskriptivní analýza navrhuje akce na základě statistik. Pokud například teplotní senzor signalizuje přehřátí, může analýza doporučit úpravu chladících systémů či upozornit personál. Využívají se optimalizační algoritmy a rozhodovací stromy.

Analýzu dat lze rozdělit také na analýzu v reálném čase a analýzu se zpožděním. Analýza v reálném čase je důležitá pro detekce anomalií, na které je nutné okamžitě reagovat, jako jsou například poruchy. Analýzu dat při které nezávisí na okamžité detekci událostí je možné provádět zpětně, a to například za účelem plánování.

2.4 Ukládání dat

Technické informace této sekce jsou čerpány z [26]. Datové úložiště je zásadní částí systému pro sběr a analýzu dat. Pokud jsou data uložena nevhodným způsobem či použitím nevhodné technologie může dojít netriviálnímu zpomalení celého systému a k problémům při získávání dat. Vezměme si příklad, kdy hodnoty naměřené senzory jsou uloženy v relační tabulce s identifikátorem v podobě zvyšujícího se čísla. Pokud budou prováděny dotazy na časové období, musí při každém dotazu proběhnout třídění dat podle času. Pokud jde o miliony záznamů, dojde k vysoké zátěži databáze.

Žádná technologie ukládání dat není univerzální. V rámci systému pro sběr dat v zemědělství je předpokládáno, že data generovaná systémem jsou různorodá a generovaná různými zdroji, jako jsou senzory, kamery, satelity, GPS a podobně. K dispozici jsou různé typy řešení pro ukládání dat. Základní typy databází používané pro sběr dat ze senzorů jsou následující.

- Relační databáze - jedná se o strukturované systémy ukládání dat, které k organizaci dat používají tabulky a vztahy mezi tabulkami. Záznamem v tabulce je řádek a typ záznamu je určen sloupcem. Relační databáze jsou vhodné pro data, která mají pevné schéma, vysokou konzistenci a nízkou složitost. Relační databáze mohou podporovat transakce, dotazy a indexování. Některé příklady relačních databází jsou MySQL, PostgreSQL a Oracle.
- NoSQL databáze - jedná se o systémy bez relací, které k ukládání a organizaci dat používají různé formáty, jako je klíč-hodnota, dokument, sloupec a graf. Databáze NoSQL jsou vhodné pro data, která mají flexibilní schéma, vysokou škálovatelnost a vysokou složitost. Databáze NoSQL mohou podporovat distribuované zpracování, paralelní zpracování a zpracování v reálném čase. Příkladem NoSQL databáze je MongoDB.
- Časové databáze - jedná se o systémy přizpůsobené pro záznamy měřené v čase. U těchto dat je běžné dotazování na časové období a klíčem je právě čas pořízení

záznamu. Databáze jsou vhodné pro sledování stavu systémů a sledování trendů v čase. Příklady databází časových řad jsou InfluxDB, Prometheus.

- Datové sklady - jedná se o centralizované systémy pro ukládání dat, které integrují data z více zdrojů a poskytují jednotný pohled na data. Datové sklady jsou vhodné pro data, která mají velký objem a velkou rozmanitost. Některé příklady datových skladů jsou Google BigQuery či Snowflake.

Výběr správného typu databáze záleží na typu dat, která jsou ukládána. Pokud je například kvalita ukládaných dat nízká (nepřesné a chybějící hodnoty), je vhodnější volba datových skladů, které umožňují čištění a validaci dat před analýzou. Pokud je důležitá integrita a dostupnost, je vhodnější relační databáze. Je běžné, že systémy využívají více technologií k ukládání dat najednou. Pro účely uložení dat obsahující relace, jako jsou údaje o uživatelích či různá metadata, je nasazena relační databáze, ale samotné hodnoty jsou uloženy v NoSQL či časové databázi. [26]

2.4.1 InfluxDB

Projekt InfluxDB je zástupcem časových databází. Je tedy orientovaná na ukládání záznamů, které jsou indexované podle časové značky. Databázi lze používat ve formátu komerčním i zdarma. Komerční řešení nabízí možnosti škálování celého systému na více zařízení a tím rozložení zátěže. InfluxDB organizuje data a přístup k nim do čtyř úrovní:

- organizace, která zastřešuje uživatele a spravuje API tokeny,
- *bucket* neboli kyblík, spadá pod organizace a jedná se o kontejnery které obsahují související data,
- měření, spadá pod *bucket* a jedná se o logickou skupinu datových bodů, tato úroveň je podobná tabulkám v relační databázi,
- datový bod, neboli samotný záznam obsahující časovou značku nebo skupinu značek obsahující metadata podle kterých se dá také indexovat a naměřené hodnoty.

Databáze umožňuje na jednotlivé kyblíky nastavovat retenční politiku, ta zajišťuje automatické mazání dat určitého stáří. InfluxDB nabízí vlastní jazyk pro psaní dotazů a umožňuje integraci do mnoha systémů, a to jak ze strany získávání, tak poskytování dat. Databáze je optimalizovaná pro zápis velkého množství dat a na dotazy na časové období, umožňuje i jednoduché agregace a filtrování dat. [27]

2.5 Vizualizace dat

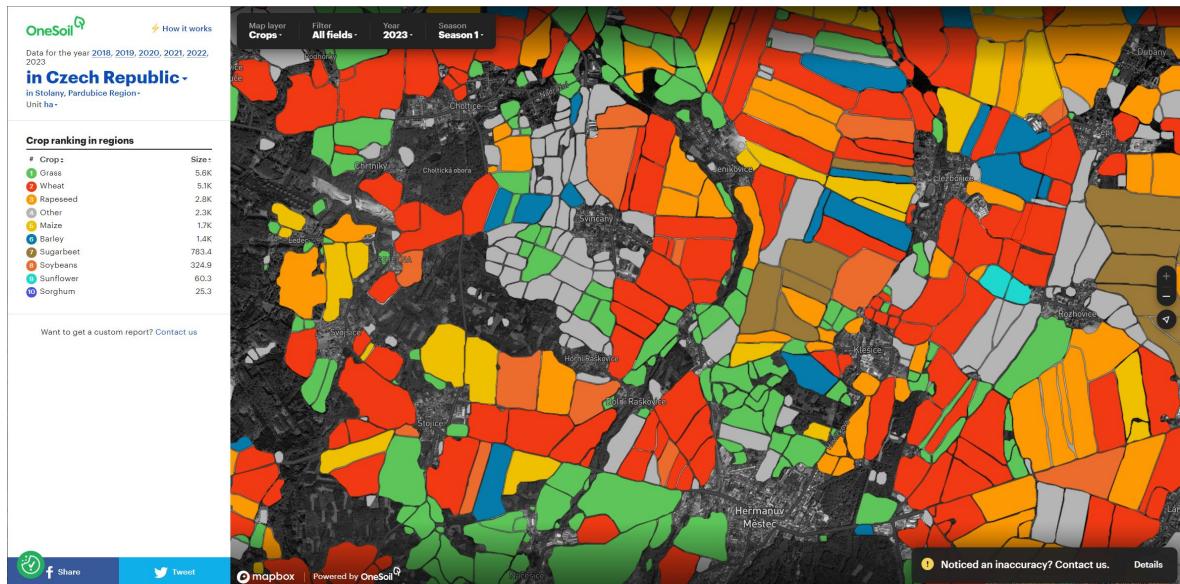
Vizualizace dat je jedním z nejlepších prostředků pro reprezentaci získaných dat koncovému uživateli. Velké agrární společnosti si mohou dovolit zaměstnávat či najímat datové analytyky. Analytici nasbíraná data zpracují a poskytují finální plány získané z dat jako je plán setby, sklizně či hnojení a podobně. U menších zemědělců je ovšem tato možnost nemyslitelná, finančně pro ně nedává smysl. Například pro včelaře vlastnícího 20 úlů je nesmyslné najímat si datového analyтика, aby mu z naměřených dat řekl, kdy je nutné včelstva dokrmovat. Je tedy velmi důležité, aby co největší množství zpracování a prezentace dat proběhlo bez lidského zásahu a byly prezentovány jen výsledné nejnutnější a nejdůležitější informace, které jsou nutné pro rozhodování. Správná kombinace zpracování dat popsaném v 2.3 a efektivního zobrazení ušetří nezanedbatelné množství času i peněz. Cílem sběru dat není to, aby se konečný uživatel probíral miliardami hodnot, ale aby získal užitečné a efektivně zobrazené informace.

K vizualizaci je využívána nejčastěji grafická reprezentace získaných dat, jako například různé typy grafů, tepelné mapy či interaktivní panely. Vizualizace a zpracování dat vyžaduje další výzkum a vývoj, aby bylo možné překonat stávající a vznikající výzvy a využít potenciál nových technologií, jako je umělá inteligence, zpracování velkých dat a *cloud computing*. Tato část uvádí některé příklady nástrojů a technik vizualizace dat, které se používají v IoT.

2.5.1 Metody vizualizace

Technické informace z této sekce jsou čerpány z [23]. K vizualizaci dat jsou nejčastěji využívány tabulky a grafy, v případě zemědělství i mapy. U vizualizace se také využívá interakce uživatele. Množství dat i po zpracování může být stále obrovské a interaktivní vizualizační prvky mohou velmi usnadnit zobrazování většího množství dat. Příkladem je interaktivní mapa, na které je možné zobrazovat a kombinovat různé vrstvy. Tyto interaktivní prvky často umožňují i dodatečné filtrování dat či vkládání dodatečných dat. Problémem u interaktivních prvků je nárok na dodatečný výkon aplikace a znovuzpracování dat. Příklad interaktivní mapy vizualizace dat ze zemědělství lze vidět na obrázku 2.6. U vizualizačních aplikací je nutné zajistit i vhodné formy exportu a sdílení dat. Při interaktivní práci uživatele s daty je vhodné uživateli umožnit uložení dodatečně upravených dat. To se děje většinou pomocí umožnění exportu do souboru či tisku vizualizace. Formát exportu je vhodné zvolit univerzální, aby bylo snadnější sdílení mezi různými aplikacemi, je vhodné využít formáty jako je *JSON* či *CSV*.

Různé formy zobrazení jsou vhodné pro reprezentaci různých typů dat či pro dosažení různých cílů. Při volbě nevhodného způsobu vizualizace může dojít ke ztrátě důležitých informací či kontextu. Například pokud je k zobrazení informací o výnosu



Obrázek 2.6 Interaktivní mapa zemědělských plodin [28].

za jednotlivé roky použit pouze jednoduchý sloupcový graf, zemědělec nemusí ze samotných hodnot poznat, že je něco v nepořádku. Z takového grafu například nepozná, že jedno pole je nadprůměrně výnosné a jiné podprůměrně výnosné. Tyto dvě anomálie se ve výsledné vizualizaci navzájem vyruší. Pro vizualizaci výnosu je mnohem vhodnější vizualizace pomocí mapy, na které je možné identifikovat méně či více výnosná místa a porovnat tyto hodnoty s předchozími roky.

Dalším důležitým bodem vizualizace je forma vizualizační aplikace. Formou aplikace rozumíme to, jakým způsobem je finální vizualizace distribuována. V dnešní době se nejčastěji využívají tři formy distribuce dat, a to skrz mobilní aplikace, aplikace pro stolní počítače a notebooky a webové aplikace.

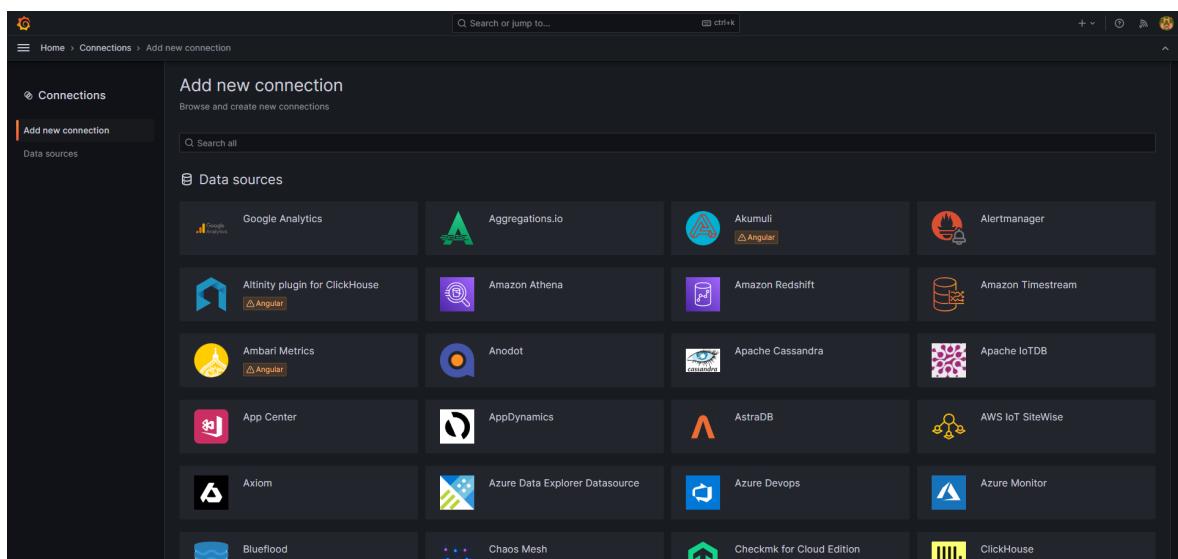
- Webová aplikace poskytuje přístup k vizualizovaným datům pomocí webového prohlížeče. Omezením webové aplikace je nutnost připojení k internetu pro její fungování, aplikace i data jsou totiž poskytována ze serverů. Velkou výhodou je snadná aktualizace. Aplikace je umístěna na serverech poskytovatele a aktualizace tudíž může provést sám poskytovatel bez nutnosti zásahu uživatele. Další výhodou je také dostupnost, k webové aplikaci je možné se dostat z jakéhokoliv zařízení s aktualizovaným prohlížečem a přístupem k internetu.
- Počítačová aplikace je oproti aplikaci webové umístěna přímo na zařízení uživatele. Tato vlastnost uživateli umožňuje s historickými daty pracovat i bez aktivního připojení k internetu, což je velkou výhodou. Problémem tohoto řešení je horší přenositelnost. Na každém zařízení využívající toto řešení je nutná instalace aplikace. Problémová může také být také aktualizace, kterou musí provádět sám uživatel.

- Mobilní aplikace umožňuje přenositelnost i ukládání dat. Aplikace je sice instalována, ovšem do mobilního přenosného zařízení. Stejně jako u počítačové aplikace může být logika umístěna přímo na zařízení a tím pádem není nutné připojení k internetu. Zároveň je aplikace uzpůsobena pro menší obrazovky což u webové aplikace není pravidlem.

Pro distribuci je náročné zvolit pouze jedno z výše zmíněných řešení. Každé má své pro a proti a žádná metoda není dominantní. Spousta existujících řešení tedy kombinuje všechny tři formy. Například nástroj Grafana popsaný v 2.5.2 nabízí všechny tři formy - webovou, mobilní i počítačovou aplikaci. To je umožněno existencí moderních vývojových nástrojů, které dovolují snadný vývoj pro všechny tři platformy najednou. Je tak možné sdílet kód mezi všemi třemi platformami, aniž by byl potřeba separátní vývoj pro každou z nich. Možnost sdílení základních prvků aplikací mezi platformami také poskytuje pohodlí uživateli, protože aplikace pro každou platformu vypadá a chová se stejně.

2.5.2 Příklad vizualizační aplikace

Řešení pro vizualizaci dat existuje nepřeberné množství. Dá se říci, že každé databázové řešení a každá společnost zabývající se velkými daty vyvíjí aplikaci pro vizualizaci dat. Existuje ovšem i velké množství obecných a snadno konfigurovatelných řešení vhodných pro různé aplikace nezávisle na datech. Jedním příkladem je nástroj Grafana.



Obrázek 2.7 Dostupné zdroje dat v základní instanci Grafany.

Grafana je platforma pro analýzu a monitorování, která má podporu pro vizualizaci, monitorování a analýzu dat a notifikace uživatele. Grafana je vyvíjena jako *open source* projekt pod licencí AGPL. Grafana umožňuje integraci širokého spektra zdrojů,

jako je Prometheus či InfluxDB, a poskytuje flexibilní a přizpůsobitelné vizualizace, jako jsou řídicí panely, grafy, mapy a různá rozšíření. Grafana také umožňuje vytvářet upozornění a správu dat. Grafana je skvělým příkladem řešení, které je vysoce konfigurovatelné a přizpůsobitelné potřebám IoT. Instanci tohoto projektu lze získat a nasadit během pár minut pomocí nástroje Docker. Následná základní konfigurace Grafany je pouze o napojení zdroje dat. Na obrázku 2.7 je možné vidět vestavěné zdroje dat podporované základní instalací Grafany. Není uskutečnitelné, aby základní funkcionalita pokryvala vše, co různí uživatelé mohou potřebovat. Grafana proto nabízí možnost využití zásuvných modulů. Tyto moduly nabízí snadnou možnost rozšíření funkcionality instance. Existují tři typy zásuvných modulů.

- Moduly poskytující napojení na zdroje dat, pokud Grafana neobsahuje napojení na nějakou specifickou databázi, například InfluxDB je možné vytvořit zásuvný modul který toto napojení poskytne.
- Aplikační moduly umožňují vytvořit specializované instance, například pro monitorování systémů, jako je Prometheus či Kubernetes.
- Panelové moduly umožňují přidávat nové typy vizualizací na řídicí panel, jako jsou mapy, hodiny, koláčové grafy, seznamy a další.

Jelikož se jedná o *open source* řešení, zásuvné moduly může vyvíjet kdokoliv, což značně zvyšuje využitelnost řešení. [29]

2.6 Zabezpečení

Zařízení IoT čelí bezpečnostním problémům, způsobeným kybernetickými útoky a úniky dat. Bezpečná komunikace v IoT využívá kryptografické protokoly a techniky k ochraně a ověřování dat a zařízení. Tato sekce se věnuje formám zabezpečení protokolů popsaných v 2.2. [13]

2.6.1 LoRa

Technické informace z této sekce jsou čerpány z [15]. LoRaWAN komunikace je zabezpečená pomocí standardu šifrování AES s šifrovacími klíči o délce 128 bitů. Pro šifrování komunikace jsou používány dva klíče. První klíč zvaný *NwkSKey* (network session key) je určen k šifrování meta dat v přenosové vrstvě a data jsou rozšifrována síťovým serverem. Druhý klíč zvaný *AppSKey* (application session key) je oproti prvnímu určen k šifrování samotného obsahu zprávy a je rozšifrován až cílovou aplikací.

Pro připojení do sítě LoRaWAN je potřeba takzvaná aktivační procedura. LoRaWan tyto procedury nabízí dvě:

- OTAA (over-the-air activation) neboli vzdálená aktivace je jednodušší metodou. Zařízení, které se chce připojit k síti LoRaWan pošle zprávu typu *Join Request* obsahující 64bitový identifikátor zařízení, 64bitový identifikátor aplikace se kterou chce komunikovat a pseudonáhodné číslo. Tato zpráva je podepsána třetím klíčem, který se v protokolu LoRaWan používá (AppKey), který je předsdílený na zařízeních a aplikačním serveru. Po úspěšné autentizaci server odesílá zprávu *Join Accept*, která je také zašifrována pomocí AppKey. Obě zařízení si na základě vyměněných informací vygenerují klíče AppSKey a NwkSKey, které jsou dále používány pro komunikaci až do ukončení relace.
- ABP (Activation By Personalization) - v této proceduře jsou již klíče AppSKey a NewSKey i 32 bitový identifikátor koncového zařízení předsdíleny při instalaci zařízení. Toto umožňuje okamžité započetí šifrované komunikace bez nutnosti autentizačních kroků.

První procedura je doporučena, protože každá relace má vygenerované rozdílné klíče, což zvyšuje bezpečnost. V druhém případě záleží životnost klíčů na uživateli. Všechny vysílané zprávy lze kdykoliv zachytit. Díky šifrování pomocí AppSKey nelze obsah rozšifrovat a díky šifrování pomocí NwkSKey nelze zprávy ani upravovat. V případě útoku je ovšem možné zprávy znova zachytávat a duplicitně odesílat. Pro prevenci tohoto typu útoku obsahuje LoRaWAN možnost čítačů, u nichž dochází k inkrementu čísla u každé zprávy. Každá zpráva ve špatném pořadí je pak ignorována.

2.6.2 Bluetooth Low Energy

Tato podsekce čerpá technické informace z [18]. *Bluetooth Low Energy* obsahuje dva bezpečnostní módy. První mód zajišťuje bezpečnost komunikace jejím šifrováním a obsahuje 4 úrovně:

- komunikace bez zabezpečení,
- neověřené párování s šifrováním,
- ověřené párování s šifrováním,
- ověřené zabezpečené spojení s šifrováním.

Druhý mód zajišťuje bezpečnost pomocí podepisování dat a obsahuje dvě úrovně:

- neověřené párování s podpisy dat,
- ověřené párování s podpisy dat.

Zařízení zahajuje komunikaci v módu 1 úrovni 1. Vyžadovaná úroveň zabezpečení se mění během času. Úroveň zabezpečení se tedy postupně zvyšuje. Pro navázání komunikace je nutné provést párování. Při párování dochází k autentizaci zařízení a zabezpečení linky pomocí krátkodobého klíče *STK*. Po autentizaci jsou distribuovány dlouhodobé klíče *LTK* pro šifrování. Díky přenosu klíčů je možné v budoucnu navázat spojení rychleji. Pokud jsou klíče uloženy pro pozdější využití, jsou zařízení spojena. Šifrování je postaveno na 128 bitovém AES standardu.

2.6.3 MQTT

Technické informace této podsekce jsou čerpány z [19]. Technologie spoléhá na externí mechanismy k zajištění zabezpečení, dostupnosti a integrity dat. Zabezpečení komunikace přes protokol MQTT je řešeno dvěma mechanismy.

Prvním mechanismem je zabezpečení MQTT pomocí protokolu *Transport Layer Security* protokolu (TLS). Protokol vytváří šifrované spojení mezi dvěma stranami za pomoci certifikátů vygenerovaných certifikační autoritou.

Druhým mechanismem zabezpečení komunikace je autentizace a autorizace. Na příkladu komunikace z obrázku 2.4 je možné vidět mnoho zařízení komunikujících přes popisovaný protokol. Hned na první pohled je viditelný problém – pokud mají všechna zařízení možnost publikovat zprávy na jakékoli téma, případně odebírat jakékoli téma, může dojít k zásadním problémům z hlediska bezpečnosti. Jakékoli zařízení na síti by reálně bylo schopné odposlouchávat veškerou komunikaci a ovládat ostatní zařízení.

Tento problém řeší protokol pomocí tří úrovní zabezpečení – identita, autentizace a autorizace. Při připojení k serveru klient zasílá svůj unikátní identifikátor (například MAC adresu) s uživatelským jménem a heslem. Heslo i uživatelské jméno je volitelné, technologie umožňuje i běh bez jakékoli autentizace či zabezpečení, což není doporučeno. MQTT broker na základě zaslanych hodnot rozhoduje o způsobilosti zařízení k připojení k síti. Po autentizaci zařízení může zařízení vykonávat publikování a odběr zpráv. MQTT broker umožňuje nastavení přístupu jednotlivých zařízení k tématům. Tohoto omezení je možné dosáhnout pomocí nastavení rolí a přiřazení různých rolí klientům či seznamem práv pro jednotlivé klienty.

Druhým způsobem autorizace přístupu k tématům je pomocí tokenů, které jsou odeslány s žádostí o přístup k MQTT síti. Protokol umožňuje kromě uživatelského jména a hesla i autentizaci pomocí certifikátů x.509 s šifrováním pomocí TLS. V tomto případě MQTT broker využívá k verifikaci certifikační autoritu. Je doporučeno využívat protokol s využitím všech metod zabezpečení, tedy identity klienta, autentizaci pomocí hesla, uživatelského jména a certifikátů a omezení přístupů pomocí tokenů či rolí.

Třetí možností, jejíž podpora ovšem není přímo vestavěna do protokolu, je šifrování obsahu zpráv. Při šifrování může být využito symetrického či asymetrického klíče a dochází k šifrování přímo samotné zprávy přenášené přes MQTT. Tuto možnost je nutné zvolit pokud TLS není dostupné a posílaná data nesmí být dostupná komukoliv.

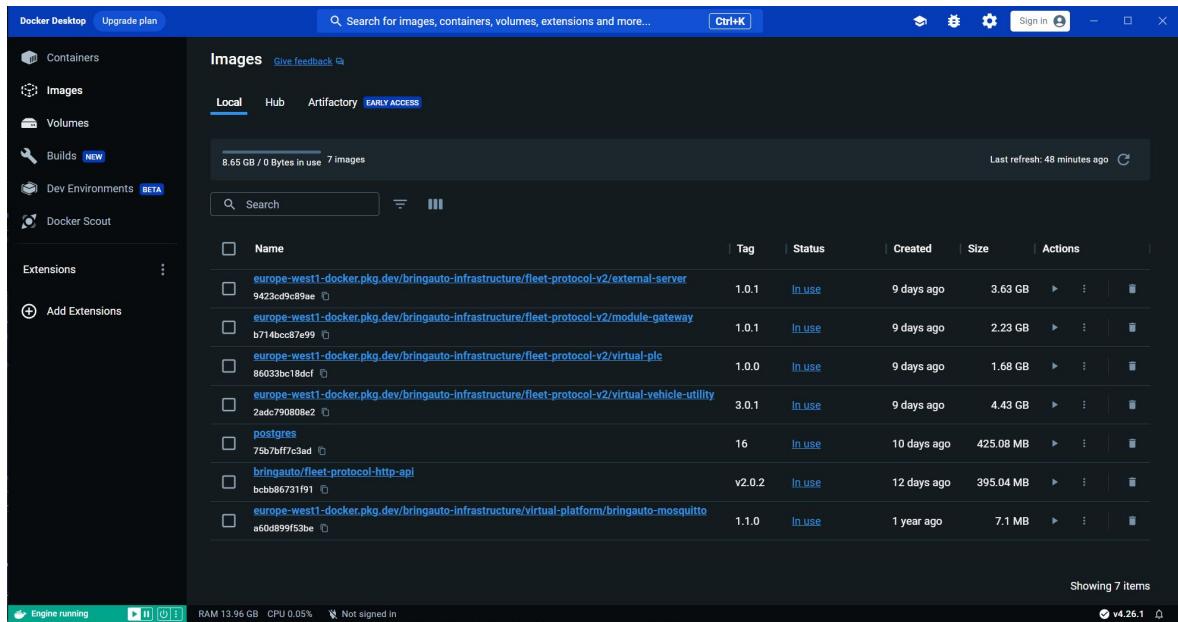
2.7 Serverová řešení

Jedním z klíčových aspektů návrhu systému pro sběr, zpracování a vizualizaci nejen v zemědělství je výběr vhodného serverového řešení. Existují dvě hlavní možnosti, a to vlastní hostování serverové části nebo využití clouдовých služeb třetích stran. Každá možnost má své výhody a nevýhody v závislosti na konkrétních požadavcích a omezeních systému.

Vlastní hostování znamená, že vlastník systému je odpovědný za nastavení, údržbu a zabezpečení hardwaru i softwaru. Tato možnost poskytuje větší kontrolu a flexibilitu nad konfigurací systému, výkonem a zabezpečením. Vyžaduje však také více technických dovedností, zdrojů a nákladů. Velkou nevýhodou je nutnost dodatečných zaměstnanců starajících se o správu systému. Toto není problém, pokud společnost již serverová pole vlastní, to ovšem v zemědělství není příliš časté. Tento typ hostování je nevhodnější pro řešení, které potřebují fungovat v takzvaném offline režimu či pro společnosti, které provozují vlastní servery.

Cloudová služba je služba, u které vlastník systému využívá služby třetí strany k hostování serverové strany systému. Tato možnost nabízí větší pohodlí a škálovatelnost, protože poskytovatel se stará o serverovou infrastrukturu, údržbu a zabezpečení. Zahrnuje však také závislost na poskytovateli clouđových služeb. Pokud je služba nedostupná, tak vlastník systému nemá příliš mnoho možností nápravy. Cloudové služby jsou ovšem většinou zálohované a ve výsledku je toto řešení stabilnější. Cloudové služby jsou vhodné pro škálovatelné systémy a společnosti bez vlastních serverů. Příklady clouđových služeb, které lze využít pro nasazení systému pro sběr analýzu a vizualizaci, jsou služba AWS od společnosti Amazon, služba Google Cloud od společnosti Google a služba IBM Cloud od společnosti IBM. [30]

Dalším velmi důležitým aspektem serverového řešení je způsob vhodného nasazení. Jednou z populárních možností je kontejnerizace, což je proces nasazení softwaru, který spojuje aplikaci a její závislosti do jednoho balíčku, který lze spustit na jakékoli infrastruktuře. Bez použití kontejnerizace může být nasazení časově velmi zdlouhavé a náročné na údržbu. Nepřeberné množství programovacích jazyků, knihoven a jejich verzí dokáže velmi znesnadnit nasazení na jeden systémový obraz. Kontejnerizace nabízí několik výhod, jako je přenositelnost, škálovatelnost, odolnost proti chybám a agilita. Kontejnerizace může také usnadnit přijetí architektury mikroslužeb, což je přístup



Obrázek 2.8 Příklad nasazení systému přes službu Docker.

k vývoji softwaru, který využívá více vzájemně závislých softwarových komponent k poskytování funkční aplikace. Některé příklady technologií kontejnerizace jsou Docker, Kubernetes a OpenShift. Tyto technologie mohou vývojářům pomoci vytvářet, nasazovat a spravovat kontejnerizované aplikace v cloudu. Kontejnerizace také dokáže velmi pomoci při vývoji. Vývojář si může pomocí kontejnerů nasimulovat kompletně celý systém bez nutnosti instalace jednotlivých závislostí a zvolit určité části, které v danou fázi vyvíjí. Příklad kontejnerizace je možné vidět na obrázku 2.8, kde ve službě Docker Desktop je spuštěný kompletní systém o sedmi kontejnerech - doba k potřebnému nasazení je méně než pár minut. [31]

II. PRAKTICKÁ ČÁST

3 NÁVRH SYSTÉMU

Pro návrh systému je důležité definovat si pojmy používané dále v této práci.

- *Zařízení* je fyzické zařízení osazené senzory a umístěné v terénu. Zařízení je zodpovědné za sběr dat a předání dat softwarové platformě.
- *Softwarová platforma* (dále jen platforma) je část zodpovědná za připojení zařízení, správu dat a přístup uživatele k datům. Jedná se tedy o všechny komponenty mimo zařízení pro měření dat. Platforma je schopná pracovat nezávisle na měřících zařízeních.

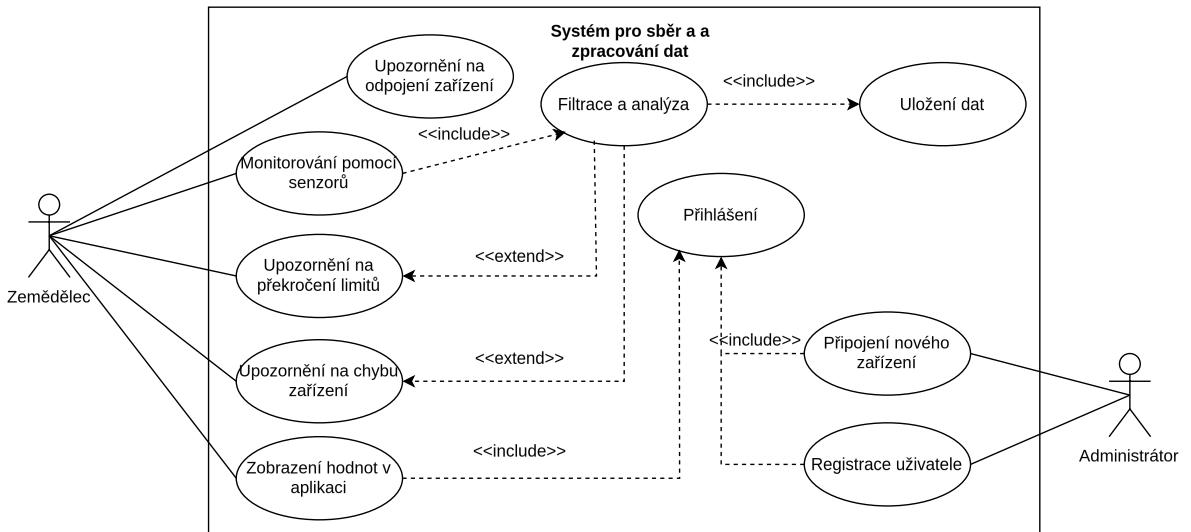
Tato část diplomové práce se zaměřuje na návrh systému obecně a návrh softwarové platformy. Sekce začíná výběrem případů užití, které platforma pokrývá. Následuje specifikace požadavků, které vycházejí z případů užití. Sekce je zaměřena na návrh systému jako celku a hlavně návrh softwarové platformy. Případy užití a specifikace požadavků a návrhu samotných zařízení pro sběr dat je popsána v sekci 4. Systém jako takový vznikl na popud samotných zemědělců. Od toho se i odvíjí případy užití, požadavky na platformu a požadavky na specifická zařízení navržená a vytvořená v rámci této práce.

3.1 Případy užití

V rámci této práce jsou uvažovány dva případy užití, na jejichž reálném základu práce vznikla. Předmětem těchto případů jsou dva aktéři, a to vinař a včelař. Systém není omezen pouze na tyto dva případy, avšak oba významně ovlivňují výběr a návrh zařízení pro sběr dat. Vizualizované případy užití je možné vidět na diagramu 3.1. Diagram případů užití zobecňuje aktéry do role zemědělce, protože aktéři sdílí požadavky na platformu, ale mají rozdílné požadavky na zařízení pro sběr dat. Rozdíl případů užití pro zařízení u aktérů je popsán až v sekci věnující se návrhu zařízení pro sběr dat 4.

Ve zvoleném případě užití je aktérem zemědělec. Cílem je sběr dat měřených senzory, jejich vizualizace a automatické upozornění při odchylce hodnot z předdefinovaných hodnot. Podmínkou pro funkčnost platformy je nainstalované a funkční zařízení, spuštěná serverová část a mobilní aplikace. Hlavní scénáře jsou následující.

- Zemědělec chce periodické záznamy ze senzorů uložené v platformě.
- Zemědělec si zobrazuje naměřené hodnoty v mobilní aplikaci.
- Zemědělec chce filtrovat nevalidní hodnoty.
- Zemědělec obdrží upozornění při překročení předem definovaných hodnot.



Obrázek 3.1 Diagram případu užití platformy.

- Zemědělec je informován při poruše zařízení.
- Zemědělec vlastní více zařízení se senzory.
- Zemědělec se musí přihlásit pro zobrazení měření.

Druhým aktérem platformy je administrátor či provozovatel systému. Hlavní scénáře pro administrátora jsou následující.

- Administrátor registruje uživatele.
- Administrátor přidává zařízení.

Případy užití je nutné převést na specifikaci požadavků platformy. Požadavky jsou specifikované v následující sekci.

3.2 Specifikace požadavků softwarové platformy

Tato podsekce se zaměřuje pouze na specifikaci požadavků na platformu, bez požadavků na samotná zařízení na kterých probíhá sběr dat. Specifikace požadavků na zařízení je v sekci 4.2. Požadavky na platformu vzniklé z případu užití jsou následující.

1. Periodické měření dat pomocí připojených zařízení, platforma musí umět periodicky získávat data z připojených zařízení.
2. Ukládání dat, platforma musí být schopná bezpečně uchovávat data pro jejich znovupoužití.
3. Podpora více zařízení, platforma musí být schopná připojit více zařízení pro sběr dat najednou, a to i různých typů.

4. Detekce odpojeného zařízení, platforma musí provést detekci odpojení zařízení a upozornit uživatele.
5. Upozornění při překročení limitů, platforma musí být schopná upozornit uživatele při překročení předem definované hodnoty.
6. Aplikace pro vizualizaci, uživatel musí mít přístup k naměřeným hodnotám pomocí vizualizační aplikace.
7. Filtrace a analýza, platforma musí umožňovat jednoduchou filtraci a analýzu dat.
8. Zabezpečení, platforma nesmí umožnit přístup k datům bez předchozí autentizace a autorizace.

Tyto požadavky na platformu vznikly ze strany uživatele. Ze strany správce systému je vhodné přidat požadavky zaměřující se na efektivitu platformy. V případě zpeněžení systému je nutné, aby bylo zapotřebí co nejmenší množství zásahů pro jeho běh. Požadavky na platformu přidané ze strany provozovatele jsou tedy následující.

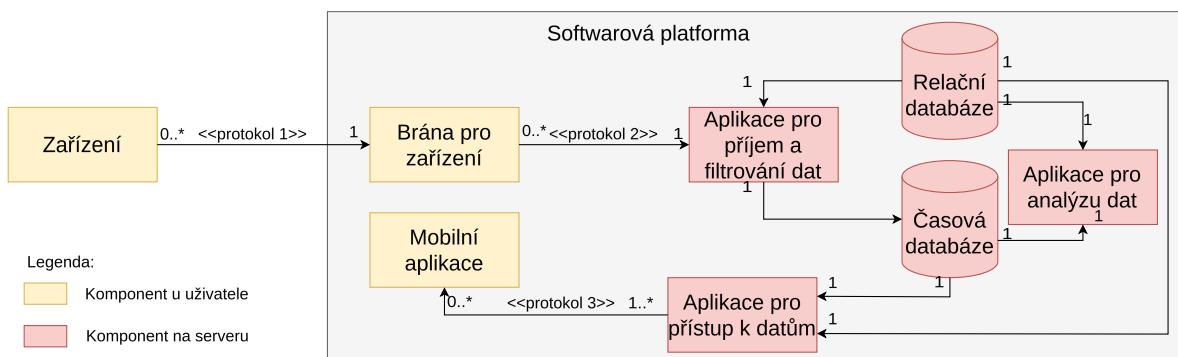
9. Snadné nasazení, platformu musí být možné jednoduše a rychle nasadit.
10. Správa uživatelů a oprávnění, platforma musí umožňovat správu uživatelů a nastavování práv pro účely sdílení platformy více uživateli.
11. Bezpečnost, platforma musí implementovat mechanismy pro zabránění neoprávněnému přístupu a obsahovat mechanismy pro ukládání auditních záznamů o přístupu k platformě.
12. Logování, platforma musí poskytovat mechanismus pro ukládání stavu celého systému.
13. Zálohování a obnova dat, platforma musí poskytovat mechanismy pro zálohu a obnovu dat.
14. Škálovatelnost, platformu a systém musí být možné škálovat za účelem přidání nových zařízení a uživatelů.

Návrh platformy bere v potaz definované požadavky. Kroky pro jejich splnění jsou popsané dále v návrhu a implementaci. Odkaz na specifické požadavky jsou řešeny pomocí jejich číslování. Pokud se tedy popis implementace odkazuje na řešení požadavku 12, řeší funkcionality požadavek na ukládání stavu systému a podobně.

3.3 Specifikace návrhu platformy

Tato podkapitola popisuje specifikaci návrhu platformy pro sběr a analýzu dat v zemědělství na základě specifikace požadavků platformy popsané v 3.2. Specifikace popisuje architekturu a vysvětluje rozhodnutí učiněná během návrhu.

Přehled komponent a jejich vztah k zařízení a umístění je možné vidět v diagramu nasazení 3.2. Platforma jako taková se skládá z brány pro zařízení, aplikace pro příjem a filtrování dat, relační databáze, časové databáze, aplikace pro přístup k datům, aplikace pro analýzu dat a mobilní databáze pro zobrazování dat.



Obrázek 3.2 Diagram komponent a jejich vztahu k zařízení.

Brána pro zařízení slouží jako přístupový bod jednotlivých zařízení do platformy. Každé zařízení může být připojeno právě k jedné bráně, ale brána umožňuje připojení více zařízení najednou. Zařízení komunikuje s branou pomocí pevně daného protokolu. Brána se stará o přeposílání dat od uživatele do serverové části. Brána je umístěna v dosahu zařízení u uživatele. Platforma může obsahovat více bran.

Aplikace pro příjem a filtrování dat je zodpovědná za přístup bran k serverové části. Aplikace je zodpovědná za filtrování nevalidních hodnot a za autorizaci a autentizaci bran. Platforma obsahuje pouze jednu aplikaci pro příjem a filtrování dat. Aplikace provádí ukládání naměřených dat do časové databáze na základě informací z relační databáze. Aplikace komunikuje s branami přes internet pomocí předem definovaného komunikačního protokolu.

Relační databáze slouží pro správu metadat z celého systému. Databáze obsahuje informace o uživatelských účtech, jejich oprávněních a dostupných zařízeních. Na základě této databáze určuje aplikace pro ukládání dat kam se data uloží a aplikace pro získání dat jestli má uživatel práva k zobrazení zařízení. Jedna instance platformy obsahuje jednu instanci relační databáze.

Časová databáze slouží jako úložiště samotných dat. Tento typ databáze je optimalizován pro ukládání dat měřených ze senzorů, kde se očekávají dotazy na určité časové období. Tento typ databáze umožňuje rychlý přístup a zároveň optimalizuje analýzy

nad tímto typem dat. Instance platformy obsahuje jednu instanci časové databáze.

Aplikace pro přístup k datům je zodpovědná za bezpečný uživatelský přístup k uloženým hodnotám. Aplikace je přístupovým bodem mobilní aplikace k serverové části a zajišťuje autentizaci a autorizaci uživatele. Aplikace poskytuje data zažádaná ze strany mobilní aplikace s využitím definovaného protokolu. Instance platformy obsahuje jednu instanci aplikace pro přístup k datům.

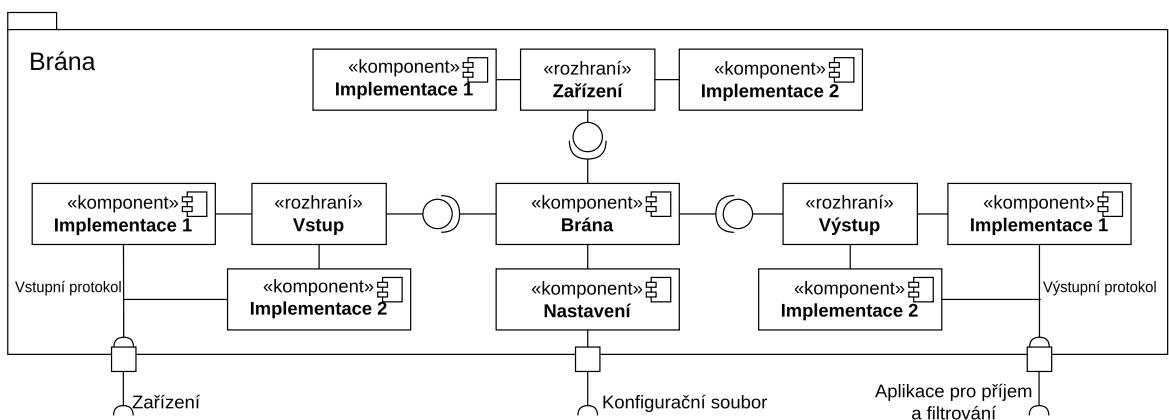
Aplikace pro analýzu dat sleduje ukládaná data a upozorňuje uživatele na předem definované události, jako je například překročení limitních hodnot. Každá instance platformy obsahuje jednu instanci této aplikace.

Mobilní aplikace slouží k vizualizaci dat naměřených zařízeními. Aplikace umožňuje přístup uživateli k platformě při zadání uživatelských údajů. Aplikace vykresluje naměřená data do grafů.

3.4 Návrh brány pro zařízení

Cílem brány pro zařízení je poskytovat přístup zařízením k platformě. Brána je tedy zodpovědná za udržování spojení se zařízeními a přenosu zpráv, které jsou dále aplikací pro příjem a filtrování dat zpracovány. Brána jako taková musí být strategicky umístěna u uživatele v místech dosažitelných signálem ze zařízení a zároveň s přístupem k internetu. Součástí návrhu brány je také komunikační protokol použitý mezi zařízením a branou a mezi branou a aplikací pro příjem a filtrování dat. Brána je zodpovědná za splnění požadavků 1 a 3.

Práce si klade za cíl vytvořit modulární řešení, což platí i pro branu. Modularita je dosaženo primárně zvolením architektury aplikace, která to umožňuje. Diagram jednotlivých komponent, jejich závislosti a rozhraní jak vnitřních tak vnějších, je vizualizovaný na diagramu 3.3.

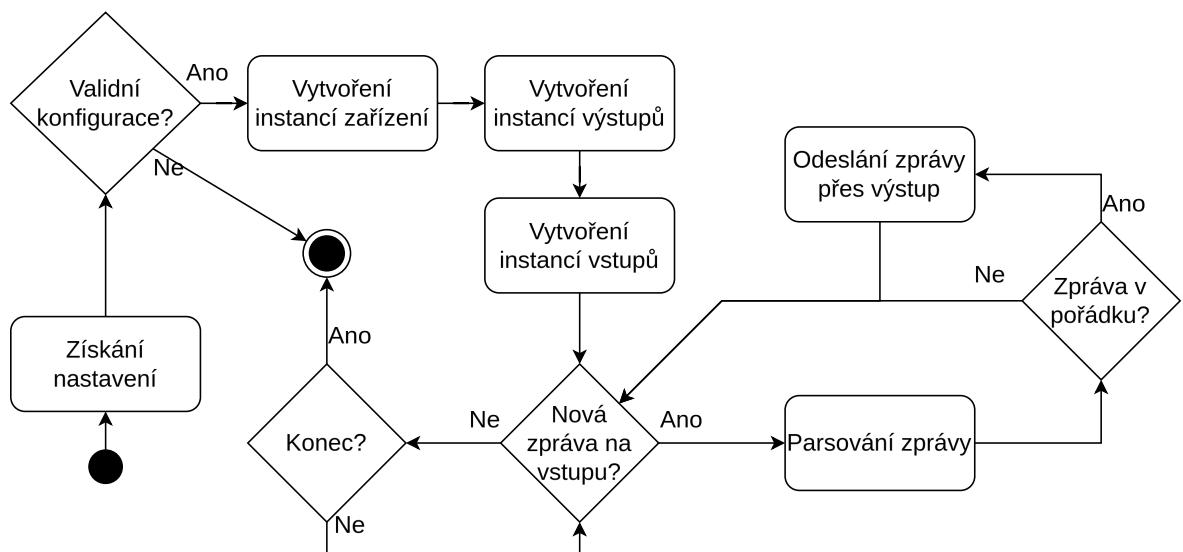


Obrázek 3.3 Diagram komponent brány pro zařízení.

Hlavní částí aplikace komponent zvaný *Brána*, tento komponent se stará o práci

s konfigurací, vytváření spojení se zařízeními, vytvořením spojení s aplikací pro příjem dat a samotnou správou všech zařízení. Tato komponenta vyžaduje pouze jednu implementaci a stará se o kontrolu zpráv a jejich správné směrování. Brána na základě nastavení z konfiguračního souboru vytváří specifické instance vstupů, výstupů a zařízení, které implementují generické rozhraní. Pro příjem zpráv program obsahuje generické rozhraní zvané *Vstup*. Rozhraní obsahuje funkce nutné pro inicializaci a získání zpráv ze zařízení a umožňuje různé implementace na základě zvolené technologie či komunikačního zařízení. Z rozhraní je bráně předána zpráva od zařízení ve formátu vstupního protokolu, jehož návrh je v sekci 3.4.1. Zpráva je předána instanci *Zařízení*. Zařízení implementuje generické rozhraní a stará se parsováním zprávy z vstupního protokolu. Brána se dále stará o předání zprávy přes rozhraní zvané *Výstup* pomocí výstupního protokolu popsaném v 3.4.2. Rozhraní *Výstupu* také obsahuje funkce nutné pro inicializaci a získání zpráv ze zařízení a umožňuje různé implementace na základě zvolené technologie či komunikačního zařízení. Instancí vstupu a zařízení může být více, instance výstupu a brány je vždy pouze jedna.

Stavový diagram programu je možné vidět na diagramu 3.4. Program začíná načtením konfigurace a její validací. Pokud je konfigurace validní, na jejím základě jsou vytvořeny instance zařízení, výstupní instance a vstupní instance. Po provedení inicializace se program nalézá ve smyčce pro čtení zpráv. Každá přijatá zpráva je validována a pokud je v pořádku, je předána pro odeslání na výstup. Veškeré chyby jsou zaznamenávány.



Obrázek 3.4 Stavový diagram brány pro zařízení.

Vstupními body programu jsou interní protokol a konfigurační soubor. První implementací je generování předdefinovaných hodnot pro účely testování bez nutnosti zařízení. Další implementace jsou rozhodnuty v sekci návrhu zařízení 4. Konfigurační

soubor je ve formátu *JSON*. Tento formát byl zvolen pro svou snadnou čitelnost a všeobecné rozšíření. Konfigurační soubor obsahuje tři hlavní sekce: všeobecná nastavení, nastavení vstupů a nastavení výstupů.

Výstupním bodem programu je výstupní protokol. Přenosové vrstvy jsou vybrány v sekci věnované aplikaci pro ukládání a filtrování dat 3.5.

3.4.1 Vstupní protokol

Vstupní protokol slouží k reportování dat naměřených zařízením. U návrhu vstupního protokolu je především důležité vzít v potaz způsob jeho využití. Protokol je využit pro komunikaci mezi zařízením a bránou. K této komunikaci může být využito mnoho různých přenosových vrstev jako je LoRa, BLE či Zigbee. Tyto přenosové vrstvy ovšem sdílí jednu vlastnost, a to nižší přenosové rychlosti. V případě využití technologie LoRa je dokonce omezené množství dat, které je možné v jedné zprávě zasílat.

Zároveň není nutné zařízení se senzory ovládat, pouze z nich sbírat data. V případě, že by i zařízení muselo přijímat data, bylo by navíc nutné odebírat větší množství energie na straně zařízení. Přenosové vrstvy také nemusí zaručovat doručení a nepoškození zpráv. Navržený protokol je tedy bezstavový a počítá pouze se zasíláním zpráv o malé, pevně dané velikosti ze zařízení do brány. Protokol je ovšem možné v budoucnosti rozšířit, doplnit či nahradit, brána totiž umožňuje díky definici rozhraní podporu více protokolů i přenosových vrstev. Formát zprávy lze vidět na obrázku 3.5.

Verze 1 B	Typ jednotky 1 B	Číslo jednotky 1 B	Příznakové bity 1 B	Data 20 B	Kontrolní suma 1 B
--------------	---------------------	-----------------------	------------------------	--------------	-----------------------

Obrázek 3.5 Formát zprávy vstupního protokolu.

Data ve zprávě jsou binární ve formátu *little endian*. Zpráva protokolu ve verzi 1 má velikost 25 bajtů a obsahuje 6 položek.

- Verze, jedná se 8 bitovou celočíselnou kladnou hodnotu udávající verzi protokolu. Verze udává formát a obsah zbytku zprávy, další položky jsou obsaženy ve verzi 1.
- Typ jednotky - jedná se 8 bitovou celočíselnou kladnou hodnotu udávající typ jednotky. Typ jednotky udává formát datové části a příznakových bitů.
- Číslo jednotky - jedná se 8 bitovou celočíselnou kladnou hodnotu udávající unikátní číslo jednotky.
- Příznakové bity - jedná se o binární pole, spodní 4 bity představují 4 bitové celočíselné kladné číslo sloužící jako počítadlo zpráv, horní 4 bity jsou závislé na typu jednotky.

- Data - jedná se o 20 bajtů dat obsahujících záznam z měření, formát dat specifikuje typ jednotky.
- Kontrolní suma - 8 bitová celočíselná hodnota součtu zbytky zprávy, slouží pro detekci poškození zprávy.

Protokol ve verzi 1 počítá s maximálním počtem 65 536 zařízení u uživatele, z toho maximálně 256 zařízení jednoho typu. Protokol zároveň umožňuje detekci ztracené zprávy pomocí 4 bitového počítadla a detekci poškození zprávy pomocí kontrolní sumy. Vzorec pro výpočet aktuální hodnoty počítadla je $C_{n+1} = (C_n + 1)mod(2^4)$ když C je hodnota počítadla a n je číslo zprávy. Vzorec pro výpočet sumy je $S = \sum_{i=0}^{24} message[i]$ kde S je kontrolní suma a $message$ je pole bitů zprávy.

3.4.2 Výstupní protokol

Výstupní protokol slouží ke komunikaci mezi bránou a aplikací pro příjem a filtrování dat. Oproti vstupnímu protokolu zde není omezeno množství zaslaných dat. Zároveň protokol musí obsahovat i informace o jednotce a měl by být jednodušší pro zpracování nejen programově, ale i vizuálně pro administrátora v případě nutnosti ladění systému. Z toho důvodu byl zvolen formát zasílaných dat ne binární, jak je tomu u vstupního protokolu, ale textový, respektive formát JSON. Tento formát umožňuje jednodušší zachycení a čtení zpráv zasílaných na síti při ladění implementace. Zároveň lze pomocí tohoto formátu velmi snadno definovat zprávy o dynamické velikosti.

Jak již bylo zmíněno, zařízení, na kterém bude spuštěn program vstupní brány, musí mít přístup k internetu a vyžaduje vyšší výkon pro neustálý příjem zpráv. Z toho důvodu lze předpokládat, že z důvodu vyšší spotřeby bude napájeno přímo z elektrické sítě a ne bateriově. Tato informace umožňuje zahrnout do výstupního protokolu více výpočetně náročné funkcionality bez ohledu na spotřebu, a to mechanismus pro zabránění ztráty dat. Tento mechanismus je velmi jednoduchý. Na každou zprávu obsahující data odeslaná na server je očekávána odpověď potvrzující příjem. Pokud odpověď není obdržena, jsou neodeslaná data uložena na lokální úložiště do souboru ve formátu CSV. Příklad vizualizovaných záznamů z tohoto souboru obsahuje z tabulka 3.1. Formát záznamu je závislý na zařízení, které jej vyprodukovalo. V příkladu vidíme záznam zařízení pro měření dat ve sklepě navrženém v rámci této práce. První tři sloupce jsou společné pro všechny zařízení a to `timestamp_ms` udávající přijetí zprávy v milisekundách ve formátu UNIX a položku `flags`, jenž obsahuje příznakové bity v decimální reprezentaci. Následuje $1..N$ naměřených hodnot, v tomto příkladě se jedná o teplotu, vlhkost a úroveň oxidu uhličitého. Anglické názvy jsou použity pro usnadnění kódování znaků. Vytváření hlavičky souboru a záznamu má na starost implementace zařízení. Název souboru určuje obsah a formát sloupců. Vzor pro název souboru je `<čas>_<typ>`

timestamp_ms	flags	temperature	humidity	co2
1706952914000	1	21.2	86.3	506
1706952924000	2	22.3	87.3	507

Tabulka 3.1 Příklad dvou záznamů ve výstupním souboru.

zařízení>_<cílo zařízení>.csv. Na základě typu zařízení je následně možné určit jaké sloupce soubor obsahuje. V příkladu 3.1 je název souboru 1706952914_1_1.csv. Každé fyzické měřící zařízení tedy má vlastní sadu souborů se záznamy.

Tento přístup umožňuje zabránit ztrátě dat systému i při výpadku internetového spojení se serverovou částí platformy. V takovém případě nedojde ke ztrátě dat ale k jejich uložení na lokální úložiště. Do protokolu je také vhodné zahrnout mechanismus pro získání takto uložených dat. Není uživatelsky přívětivé ručně získávat neodeslaná data a předávat je administrátorovi pro pozdější zpracování. Z toho důvodu protokol zavádí i možnost zaslání požadavku o získání takto uložených dat serverovou stranou do zařízení. Protokol tedy obsahuje pět různých zpráv, a to zprávu typu DATA, DATA_ACK, DATA_READ, DATA_READ_RESPONSE a DATA_READ_RESPONSE_ACK.

Příklad zprávy typu DATA pro stejné zařízení z příkladu 3.1 je následující:

```
{
  "type": 1,
  "id": 1,
  "device_type": 1,
  "device_id": 1,
  "stored_data_points": true,
  "data_points": [
    {
      "timestamp_ms": 1706952923000,
      "flags": 3,
      "temperature": 22.3,
      "humidity": 83.5,
      "co2": 507
    }
  ]
}
```

Typ této zprávy je identifikován položkou `type`, která je rovna hodnotě 1. Položka `id` obsahuje inkrementující se čítdlo zpráv výstupního protokolu a slouží pro případnou detekci výpadku zpráv. Položky `device_type` a `device_id` určují zdrojové zařízení. Položka `stored_data_points` obsahuje informaci o tom, zda jsou na zařízení uloženy neodeslané zprávy pro dané zařízení. Pole `data_points` obsahuje samotné datové body ze zařízení. Každá zpráva je zastoupena jednou položkou. Formát těchto položek kopíruje sloupce z csv souboru a je tedy určen typem zdrojového zařízení. Příklad zprávy tedy přímo navazuje na položky z 3.1. Odpověď na tuto zprávu typu DATA_ACK ze serverové strany vypadá následovně:

```
{
  "type": 2,
  "id": 1,
```

```

    "response_type": 1,
    "error_data": "DeviceUnknown"
}
```

Typ této zprávy je definován položkou `type` která je rovna hodnotě 2. Položka `id` kopíruje identifikátor ze zprávy na kterou odpovídá, tímto způsobem je vyřešeno potvrzování zpráv. Položka `response_type` určuje typ odpovědi. Pokud je hodnota rovna 0 je vše v pořádku, jakákoli nenulová hodnota představuje chybu, která je popsána položkou `error_data`. V příkladu zprávy aplikace pro příjem a filtrování dat dané zařízení nezná.

Zpráva DATA_READ pro získávání uložených dat vypadá následovně:

```

{
    "type": 3,
    "id": 10,
    "device_type": 3,
    "device_id": 1
}
```

Typ této zprávy je identifikován položkou `type`, která je rovna hodnotě 3. Položka `id` je zvyšující se čítadlo zpráv. Čítadla ze zpráv DATA a DATA_READ jsou na sobě nezávislá. Položky `device_type` a `device_id` určují ze kterého zařízení mají být čtená data. Odpověď na tuto zprávu DATA_READ_RESPONSE má totožný formát jako zpráva DATA s tím rozdílem, že položka `type` má hodnotu 4 a hodnota `id` zrcadlí hodnotu z příkazu. Odpověď na výše ukázaný příklad by tedy vypadala následovně:

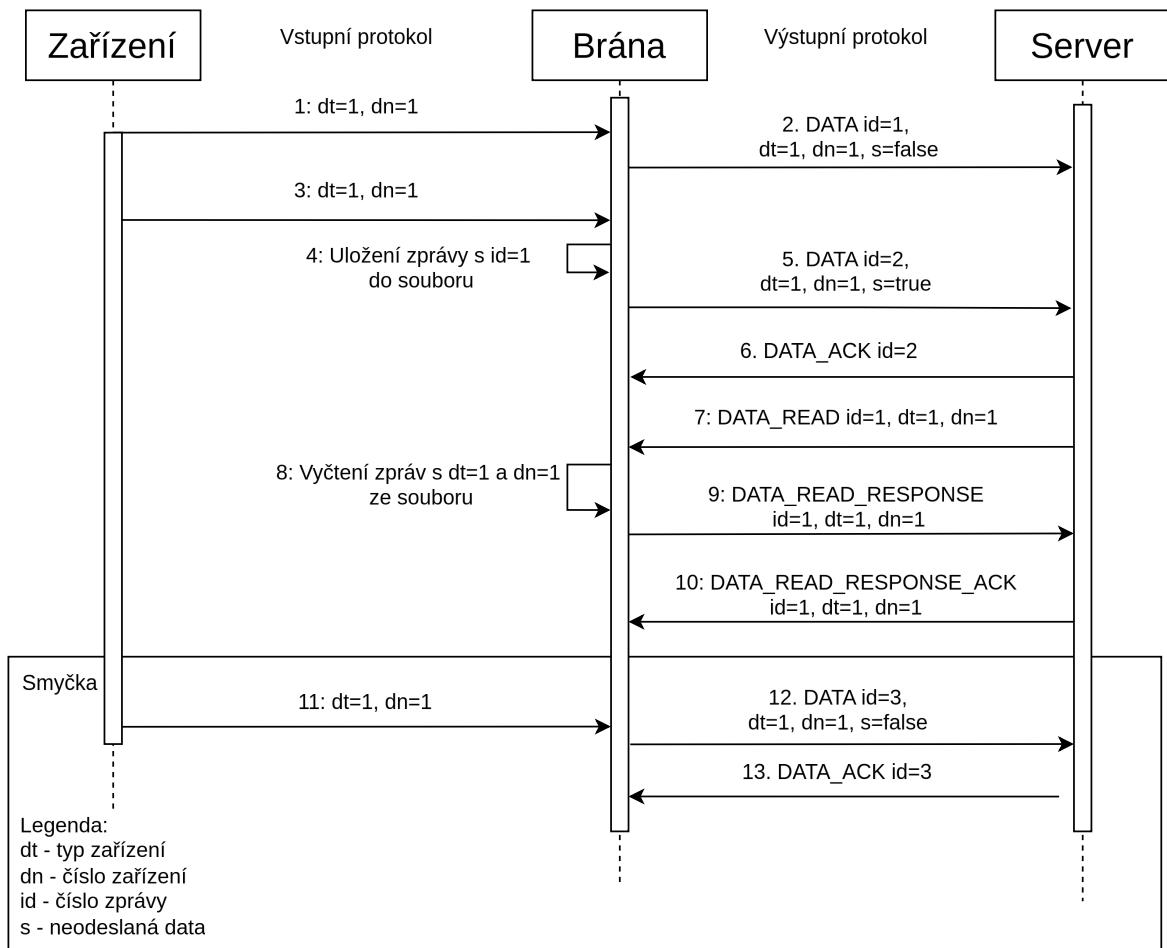
```

{
    "type": 4,
    "id": 10,
    "device_type": 3,
    "device_id": 1,
    "stored_data_points": true,
    "data_points": [
    ]
}
```

Zpráva typu DATA_READ_RESPONSE_ACK je totožná se zprávou DATA_READ_ACK a liší se jen v hodnotě `id`, má hodnotu 5. Tato zpráva slouží k potvrzení příjmu uložených dat a je vysílána směrem k bráně.

Na sekvenčním diagramu 3.6 je možné vidět příklad komunikace výstupním protokolem v případě, že se nepodaří odeslat první zprávu. U daného scénáře je posloupnost událostí následující.

1. Zařízení typu 1 a čísla 1 zašle zprávu vstupního protokolu do brány, která ji přijme.
2. Brána vytvoří zprávu výstupního protokolu typu DATA, přidělí ji identifikátor 1 a zašle ji do aplikace pro příjem a filtrování zpráv, v diagramu značené jako Server.



Obrázek 3.6 Sekvenční diagram výstupního protokolu.

3. Zařízení typu 1 a čísla 1 zašle druhou zprávu vstupního protokolu do brány, která ji přijme.
4. Předchozí zpráva nebyla potvrzena a proto je zálohována do souboru.
5. Brána vytvoří zprávu výstupního protokolu typu DATA, přidělí ji identifikátor o jedno vyšší, tedy 2 a zašle ji na server. Zpráva má nastavený příznak o neodeslaných zprávách.
6. Server obdrží zprávu DATA s identifikátorem 2 a zašle jako odpověď DATA_ACK se stejnou hodnotou identifikátoru.
7. Protože předchozí zpráva obsahovala informaci o neodeslaných zprávách ze zařízení typu 1 a čísla 1, je vygenerována zpráva DATA_READ s identifikátorem 1 a zaslána bráně.
8. Brána obdrží zprávu DATA_READ a načte data z příslušného souboru.

9. Brána vygeneruje zprávu DATA_READ_RESPONSE se stejným identifikátorem, jako měl požadavek.
10. Server potvrdí získání uložených dat zprávou DATA_READ_RESPONSE_ACK se stejným identifikátorem, jako předchozí dvě zprávy.
11. Zařízení typu 1 a čísla 1 zašle zprávu vstupního protokolu do brány, která ji příjme.
12. Brána vytvoří zprávu výstupního protokolu typu DATA, přidělí ji identifikátor o jedno vyšší a zašle ji na server.
13. Server obdrží zprávu DATA a zašle jako odpověď DATA_ACK se stejnou hodnotou identifikátoru.

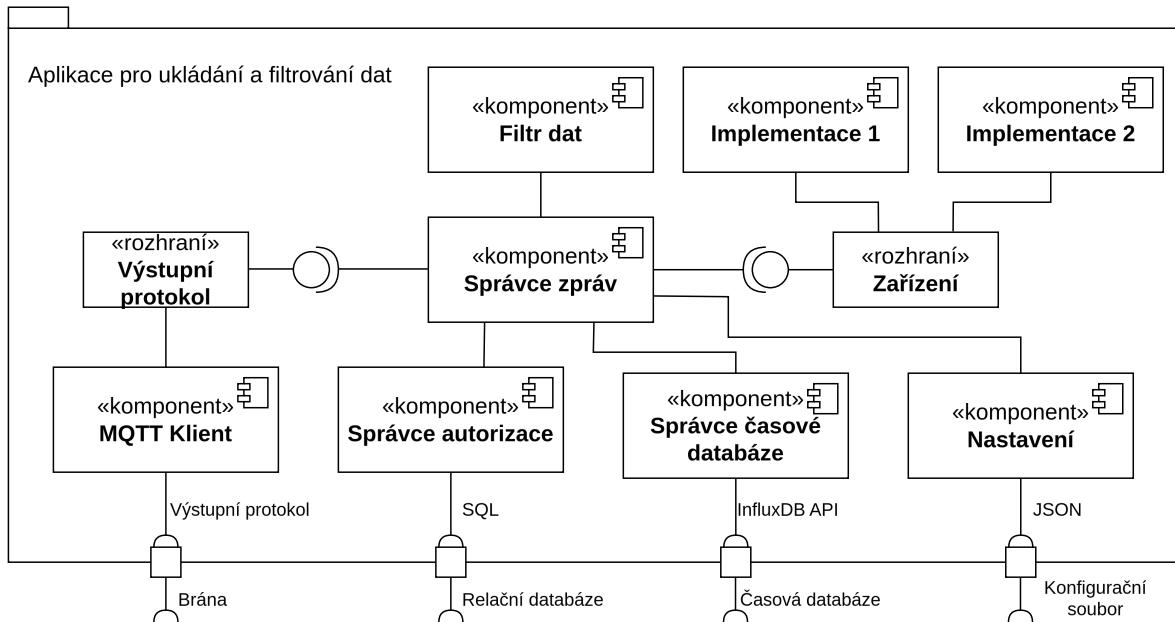
Kroky 10, 11 a 12 se opakují ve smyčce. Protokol zajišťuje, že je každá zpráva předána minimálně jednou, ale nezajišťuje, že je zpráva předána právě jednou. Je možné, že některé zprávy obdrží server duplicitně kvůli zpomalení sítě a přijetí potvrzení se zpožděním. Na to je nutné myslit u výběru a nastavení časové databáze, duplicitní hodnoty se musí zahazovat. Protokol je v budoucnu možné rozšířit o další zprávy, například pro zjišťování stavu zařízení či o zaslání konfigurace. Identifikace brány, stejně jako její autorizace a autentizace, je řešena pomocí přenosové vrstvy a popsaná v sekci 3.5.

3.5 Návrh aplikace pro příjem a filtrování dat

Cílem aplikace pro příjem a filtrování dat je poskytovat přístup branám k serverové části platformy. Aplikace je zodpovědná za autorizaci a autentizaci bran. Aplikace je dále zodpovědná za filtrování dat na základě předem určených pravidel a za ukládání dat do časové databáze na základě informací z časové databáze. Aplikace umožňuje připojení více bran najednou.

Aplikace pro příjem a filtrování dat implementuje výstupní protokol ze sekce 3.4.2 a komunikuje s množinou bran, filtruje data a ukládá je do časové databáze na základě relační databáze. Přehled jednotlivých komponent, jejich závislosti a rozhraní jak vnitřních, tak vnějších, je vizualizovaný na diagramu 3.7.

Hlavní část aplikace je komponent *Správce zpráv*. Tento komponent vytváří komponent *MQTT Klient*, který slouží k příjmu a odesílání zpráv výstupního protokolu a komunikuje s branami. Pro práci s daty ze zařízení, jako je zpracování či úpravy, obsahuje aplikace rozhraní *Zařízení*, jež má implementaci pro každý existující typ zařízení. Za filtrování zpráv je zodpovědný komponent *Filtr dat*, pro autorizaci a komunikaci s relační databází slouží komponent *Správce autorizace*, pro nahrávání dat do časové

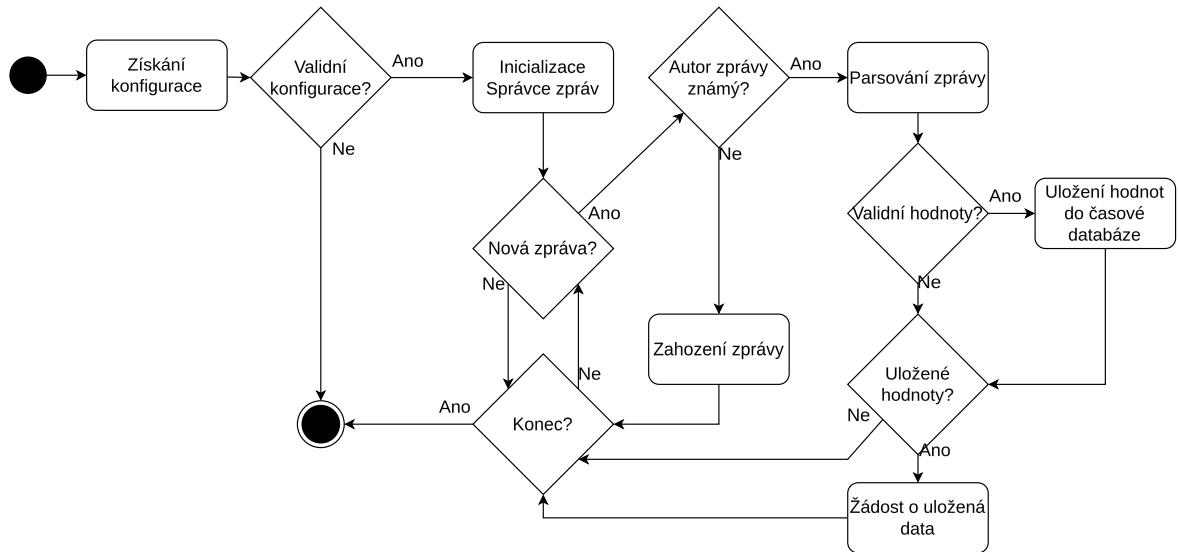


Obrázek 3.7 Diagram komponent aplikace pro ukládání a filtrování dat.

databáze program obsahuje komponent *Správce časové databáze*. Nastavení aplikace je načítáno z konfiguračního souboru. Vnějšími rozhraními aplikace jsou tedy brány, relační databáze, časová databáze a konfigurační soubor.

Stavový diagram programu je zobrazen na diagramu 3.8. Program začíná načtením konfigurace. Po načtení konfigurace dojde k inicializaci všech komponent a spuštění hlavní smyčky odbavování zpráv. Při příjmu zprávy je brána autorizována a dojde také k autentizaci zařízení obsažené ve zprávě. Pokud se krok nepovede, je zpráva zahozena a situace je zaznamenána. Po ověření autora zprávy dochází k jejímu parsování a vyfiltrování hodnot. V případě, že jsou hodnoty validní, jsou uloženy do časové databáze. Následně je zkонтrolován příznak uložených zpráv z výstupního protokolu. Pokud je příznak aktivní, dojde k zaslání žádosti o uložená data na bránu.

Výstupní protokol jako takový neřeší šifrování či autorizaci nebo autentizaci. Tyto kroky jsou řešeny pomocí přenosové vrstvy MQTT. MQTT umožňuje šifrování datového přenosu pomocí SSL a autorizaci s autentizací pomocí uživatelského identifikátoru a hesla. U protokolu uživatel zprávy publikuje na určité téma a jiný je odebírá. Přístup uživatelů lze omezit na určitou podmnožinu témat, ke kterým mají přístup. Uživatelé jsou v tomto případě brána a popisovaný program pro příjem dat. Aplikace má přístup ke všem tématům, brány pouze ke specifickým. Jelikož výstupní protokol počítá s komunikací v obou směrech, probíhá komunikace mezi aplikací a bránou pomocí dvou témat. Formát tématu je <název společnosti>/<identifikátor brány>/<směr>. Pokud se tedy v systému nalézá brána s identifikátorem `gate_1` společnosti `Agri`, tak brána publikuje na téma `agri/gate_1/gateway` a odebírá zprávy na téma `agri/gate_1/uploader`. Názvy jsou pro zabránění chyb z důvodu kódování bez diakritiky a malými písmeny.

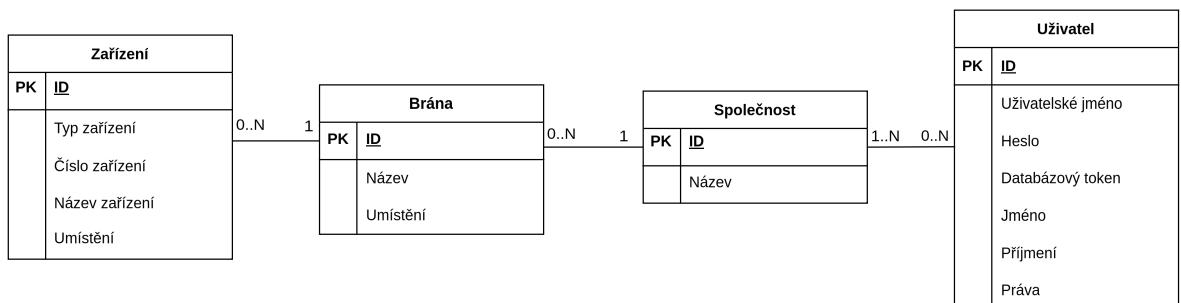


Obrázek 3.8 Stavový diagram aplikace pro ukládání a filtrování dat.

Směr gateway určuje směr z brány do aplikace pro ukládání a filtrování a směr uploader indikuje opačný směr.

3.6 Návrh relační databáze

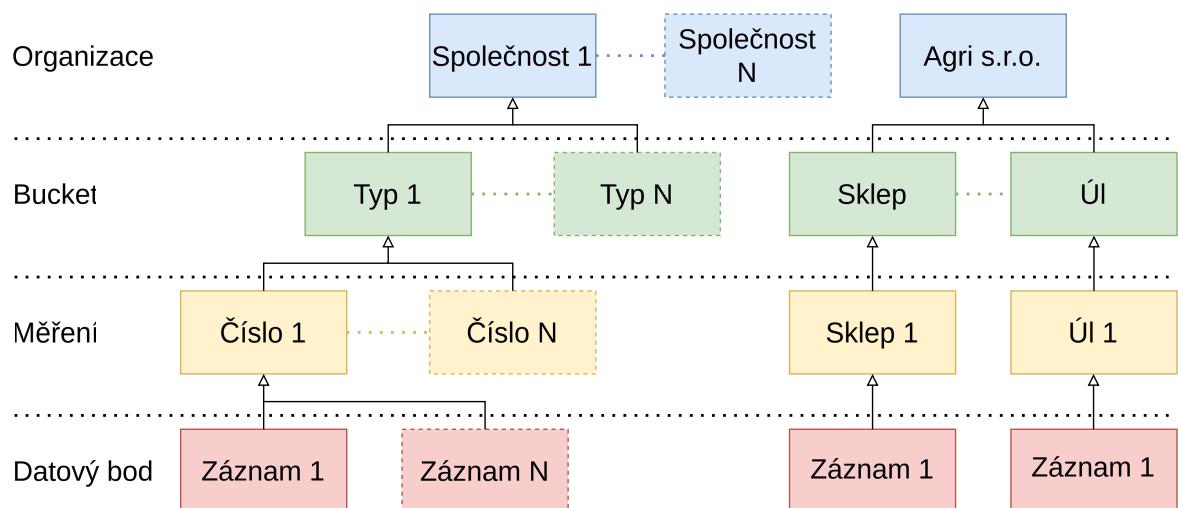
Relační databáze v rámci platformy slouží ke správě uživatelů, jejich oprávněním a ke správě bran a zařízení. Databázové schéma je zobrazeno v diagramu 3.9. Databáze je velmi jednoduchá a obsahuje 4 tabulky, a to společnost, uživatel, brána a zařízení. Zařízení náleží právě jedné bráně. Brána může obsluhovat více zařízení. Brána patří právě jedné společnosti a společnost může vlastnit více bran. Aby data obdržená od brány byla uložena do časové databáze, musí být brána zanesena v relační databázi a musí se vůči databázi autentizovat. Autentizaci a autorizaci brány řeší aplikace pro příjem a filtrování dat. Tabulka uživatelů slouží pro aplikaci pro zobrazování. Uživateli je umožněno získání dat ze všech zařízení náležící společnostem, u kterých je veden. Uživatel může být členem více společností.



Obrázek 3.9 Databázové schéma relační databáze.

3.7 Návrh časové databáze

Pro účely návrhu časové databáze je nutné znát použitou databázi. Na rozdíl od relačních databází, kde je možné návrh přetrasformovat do databáze nehledě na zvolenou technologii, nemají časové databáze přesně určenou strukturu týkající se metadat. Jednotliví zástupci přistupují trochu jinak k přístupu ukládání dat. Jediná fixní část je podpora indexování podle časové značky. Pro časovou databázi je zvolen produkt InfluxDB. Primárním důvodem pro volbu této databáze je kladná zkušenosť při práci s ní a jejím databázovým schématem určeným pro ukládání dat ze senzorů. Princip ukládání do časové databáze InfluxDB je popsán v 2.4.1. Schéma databáze i s příkladem je vizualizováno na diagramu 3.10.



Obrázek 3.10 Schéma a příklad časové databáze vůči struktuře.

Organizace v databázi InfluxDB umožňuje rozdělení přístupových práv k jednotlivým měřením. Každá společnost z relační databáze tedy má vytvořenou organizaci v časové databázi. *Bucket* v návrhu představuje typ zařízení a jednotlivá reálná zařízení již existují na úrovni měření. V příkladu se tedy vyskytuje fiktivní společnost Agri s.r.o., která vlastní dvě zařízení, první typu Sklep s číslem 1 a druhé typu Úl s číslem 1. Datové body v databázi InfluxDB obsahují tři typy hodnot: čas, značky a hodnoty. Čas je uložen vždy, jedná se o primární klíč pro vyhledávání. Vyhledávání lze provádět i pomocí značek, jedná se o textové pole které umožňuje rozeznat zdroj dat. V databázi datový záznam obsahuje minimálně název či identifikátor brány, ze kterého záznam pochází. Položky hodnot jsou závislé na typu zařízení, v případě zařízení Sklep se jedná například o úroveň oxidu uhličitého. Hodnoty měřené jednotlivými typy zařízení jsou specifikovány v sekci návrhu zařízení 4. Zabezpečení je řešeno samotnou databází, uživatel pro přímý přístup potřebuje přístupový token či přihlašovací údaje. Databáze zároveň nemusí být přístupná z internetu, protože o vkládání se stará program pro příjem a filtrování dat a o získávání dat se stará program pro přístup k datům.

Databáze InfluxDB umožňuje napojení různých vizualizačních nástrojů a nástrojů pro analýzu dat. Pro program pro analýzu dat a zasílání upozornění pro uživatele může tedy mohou být použity již existující nástroje. Jedním z možných nástrojů je Grafana. Grafana umožňuje vizualizaci dat z nepřeberného množství zdrojů. Zároveň nabízí definované rozhraní, pro které lze vytvářet vlastní zásuvné moduly. Po připojení datového zdroje je možné v Grafaně nadefinovat monitorování příchozích dat. Při překročení určeného limitu je následně možno dostat upozornění. Podobně jako u datových zdrojů je monitorování i notifikace možné rozšířit o zásuvné moduly. Uživatel tedy není omezen na využití jednoduchého překročení předem definované hodnoty a upozornění na email, ale může definovat složitější analýzy a integrace do jiných systémů, jako je například aplikace pro týmovou komunikaci Slack.

3.8 Návrh aplikace pro přístup k datům

K datům v časové databázi lze přistupovat na přímo z mobilní aplikace. To ovšem může představovat bezpečnostní riziko a mobilní aplikace by musela obsahovat velké množství logiky pro dotazování a musela by dokonale znát databázové schéma databáze. Zároveň by bylo velmi obtížné provést výměnu databáze InfluxDB za jiné řešení, velká část mobilní aplikace by se v takovém případě musela změnit. Z toho důvodu platforma obsahuje aplikaci pro přístup k datům.

Tato aplikace slouží k skrytí databázové vrstvy systému. Uživatel tedy na přímo nebude interagovat ani s relační ani s časovou databází, ale s přesně definovaným univerzálním rozhraním. Rozhraní využívá protokol HTTP, parametry jsou předávány metodou POST, odpovědi jsou ve formátu JSON a nabízí následující funkce.

- `/get_companies` - funkce slouží pro získání seznamu společností daného uživatele. Funkce očekává parametry `username` a `password`, tedy uživatelské jméno a heslo. Formát odpovědi je list společností a vypadá následovně.

```
{  
  [  
    "Company\u201d1",  
    "Company\u201d2"  
  ]  
}
```

- `/get_devices` - funkce slouží pro získání všech zařízení dané společnosti. Funkce očekává parametry `username`, `password` a `company`. Oproti předchozí funkci přibyla položka určující společnost. Funkce vrací list zařízení. Každé zařízení obsahuje položku `device_id` tedy identifikátor zařízení v rámci společnosti, dále `device_name` tedy název společnosti, `device_type` neboli typ zařízení a `fields` představující seznam měřených veličin. Příklad odpovědi obsahující jedno zařízení je následující.

```
[  
  {  
    "device_id": 1, "device_name": "sklep1",  
    "device_type": 1,  
    "fields": [  
      "temperature",  
      "humidity",  
      "co2"  
    ]  
  }  
,
```

- `/get_data` - funkce slouží pro získání bodů měření ze specifického měření pro vizualizaci. Parametry funkce jsou `username`, `password`, `company`, `device_type`, `device_id`, `fields`, `from` a `to`. Jako parametry se oproti předchozím funkcím dále specifikuje zdrojové zařízení pomocí jeho typu a identifikátoru, seznam atributů které mají být získány (například vlhkost) a časové rozmezí dotazu ve formátu unixové časové značky ve vteřinách. Odpověď je list obsahující pole atributů, kdy každý objekt atributu obsahuje položky `measurement` určující název atributu, jednotku atributu `unit`, a dále obsahuje pole `values` obsahující jednotlivé body s položkami `value` a `time` obsahující samotnou hodnotu a čas naměření ve formátu unixové časové značky ve vteřinách. Příklad odpovědi s jedním atributem a jednou naměřenou hodnotou je následující.

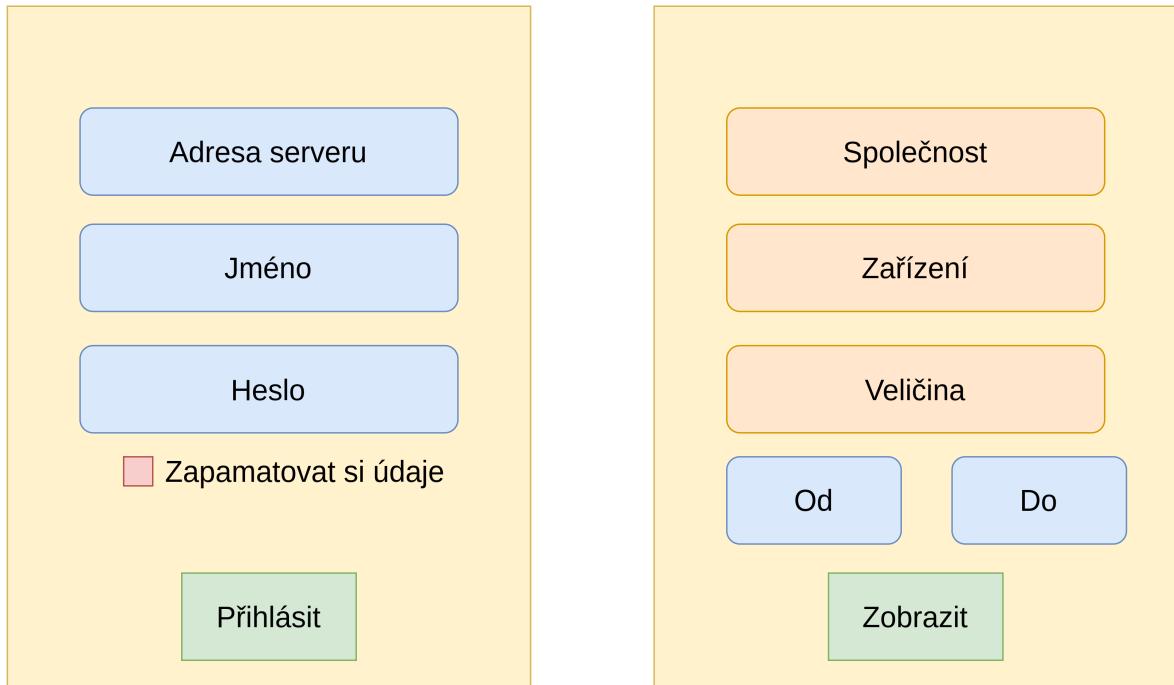
```
[  
  {  
    "measurement": "humidity",  
    "unit": "%",  
    "values": [  
      {  
        "value": 50.5,  
        "time": 1708932533  
      }  
    ]  
  }  
,
```

Rozhraní je v budoucnu možné rozšířit o další funkce, aktuální návrh je vytvořen tak, aby splňoval požadavky na systém a poskytoval mobilní aplikaci všechny informace nutné k vykreslení grafů.

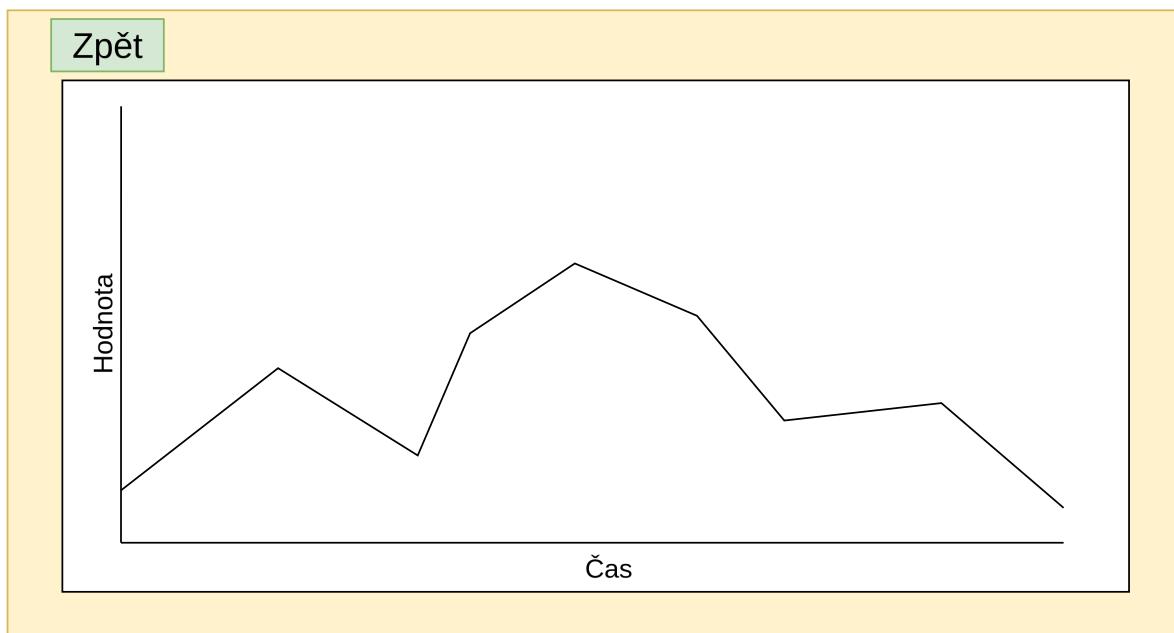
3.9 Návrh mobilní aplikace

Mobilní aplikace má zásadní význam v rámci celého systému pro uživatele. Prostřednictvím mobilní aplikace je uživateli umožněna vizualizace naměřených dat. Návrh aplikace upřednostňuje hlavně jednoduchost, použitelnost a efektivitu. Aplikace jako taková komunikuje v rámci platformy s aplikací pro přístup k hodnotám a po uživateli vyžaduje přihlášení. Návrh vzhledu a funkcionality aplikace je možné vidět na obrázcích 3.11 a 3.12. Aplikace obsahuje tři separátní okna mezi kterými na základě stavu

přechází. Jedná se o: okno pro přihlášení, okno pro volbu parametrů grafu a okno pro prezentaci grafu.



Obrázek 3.11 Návrh rozložení elementů pro přihlášení a volbu parametrů v mobilní aplikaci.



Obrázek 3.12 Návrh rozložení elementů pro okno zobrazení grafu.

První okno, ke kterému má uživatel přístup, je určeno pro přihlášení. Návrh rozložení elementů je vizualizován na obrázku 3.11 v levé části. Obrazovka obsahuje tři vstupní pole - pro adresu serveru, pro uživatelské jméno a heslo. Dále obsahuje zaškrťávací políčko umožňující zapamatování přihlašovacích údajů a tlačítko pro přihlášení.

Po úspěšném přihlášení dojde k přepnutí na okno určené pro volbu parametrů pro vizualizaci grafu. Při přihlášení uživatele dojde k načtení společnosti, kterým je uživatel přiřazen, a načtení zařízení, kterými dané společnosti disponují. Uživatel následně vybírá z rozbalovací nabídky společnost a zařízení. Dále vybírá časový interval zobrazovaných dat pomocí políček *Od* a *Do* a jakou veličinu měřenou zařízením chce zobrazit pomocí nabídky *Veličina*. Dále okno obsahuje tlačítko *Zobrazit*, které uživatele přesměruje na třetí okno pro vizualizaci grafu. Při stisku tlačítka *Zobrazit* je odeslána žádost o data do aplikace pro získání dat a data jsou vyobrazena v grafu na třetí obrazovce. Kromě samotného grafu obsahuje stránka ještě tlačítko *Zpět* pro návrat k výběru parametrů grafu. Aplikace je navržena tak, aby byla co nejjednodušší na použití. V budoucnu je možné ji rozšířit například o ukládání předdefinovaných grafů či export dat.

4 NÁVRH ZAŘÍZENÍ

Tato část diplomové práce je věnována specifikaci případů užití, specifikaci požadavků a návrhu měřících zařízení. Systém navržený v 3 je obecný a umožňuje připojení jakéhokoliv zařízení, implementující navržený protokol 3.4.1. V této části jsou navržena dvě zařízení které lze využít jako základ pro další zařízení. Návrh zařízení vychází z případů užití a specifikace požadavků.

4.1 Případy užití

Jak již bylo zmíněno v sekci 3.1, případ užití systému jako celku je mířen na zemědělce. Specifické rozdělení je řešeno na úrovni měřícího zařízení, které je připojeno k systému. Zde přichází dva případy užití z reálného života. V prvním případě je aktérem vinař. Cílem je využití systému z předchozí sekce k monitorování teploty, vlhkosti a oxidu uhličitého ve sklepě za účelem kontroly prostředí pro výrobu vína. Hlavní scénáře jsou následující:

- Vinař chce mít možnost ve sklepě sledovat teplotu, vlhkost a úroveň oxidu uhličitého.
- Vinař chce mít přístup k naměřeným hodnotám přes aplikaci i zobrazením na samotném měřícím zařízení.
- Vinař potřebuje být upozorněn při naměření nebezpečné hodnoty oxidu uhličitého.

V druhém případě je aktérem včelař. Cílem je monitorování hmotnosti včelího úlu za účelem usnadnění plánování dokrmování včelstva a stáčení medu. Hlavní scénáře jsou následující.

- Včelař chce sledovat hmotnost včelího úlu.
- Včelař chce mít přístup k naměřeným hodnotám přes aplikaci.
- Včelař nechce kupovat existující platformy, které jsou velmi drahé.

4.2 Specifikace požadavků zařízení pro sběr dat

Požadavky na zařízení vychází z případů užití ze sekce 3.1, na jejichž základě jsou navržena dvě zařízení. Požadavky na první zařízení určeného pro vinaře jsou následující.

1. Schopnost bezdrátové komunikace z podzemních prostor.
2. Měření teploty, vlhkosti a úrovně oxidu uhličitého.

3. Obrazovka s naměřenými hodnotami.
4. Schopnost okamžitého upozornění na vysoké koncentrace oxidu uhličitého bez připojení k systému.
5. Odolnost proti vlhkosti.

Požadavky na druhé zařízení určeného pro včelaře jsou následující.

1. Schopnost bezdrátové komunikace na velké vzdálenosti.
2. Měření váhy.
3. Schopnost operovat na baterie.
4. Odolnost proti počasí.

Návrh obou zařízení zohledňuje výše zmíněné požadavky.

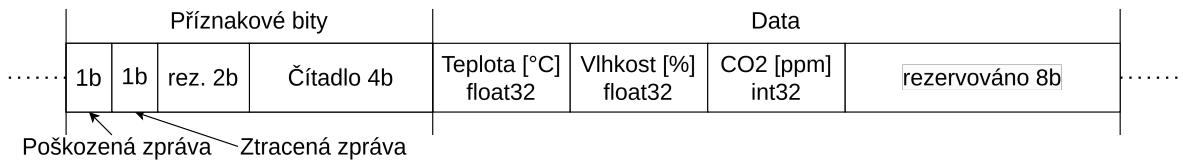
4.3 Specifikace návrhu měřícího zařízení pro vinaře

Zařízení určené pro vinaře je určeno primárně pro měření hodnot koncentrace oxidu uhličitého, který vzniká při kvašení vína, přeměnou cukrů na alkohol. Primárním úkolem tohoto zařízení je upozornit vinaře na nebezpečné koncentrace oxidu uhličitého. Sekundárním úkolem zařízení je monitorování teploty a vlhkosti. Teplotu je nutné monitorovat z důvodu skladování vína a vlhkost kvůli tvorbě plísní. Zařízení musí být schopné komunikovat s platformou, což přináší určité komplikace, a to v samotném přenosu dat do brány. Měřící zařízení bude totiž umístěno několik metrů pod zemí. Tato sekce popisuje návrh zařízení jako takového, včetně volby výpočetní jednotky, senzorů a přenosové vrstvy pro vstupní protokol a hodnot přenášených ve výstupním protokolu.

4.3.1 Specializace vstupního protokolu

Pro vstupní protokol je nutné specifikovat dvě věci - přenosovou vrstvu a obsah datové části. Sklep určený pro testování jednotky se nalézá zhruba 500 metrů od zástavby a je tedy nutné zvolit bezdrátovou přenosovou technologii, která zvládne minimálně tuto vzdálenost. Nabízejí se dvě možnosti, a to LoRa popsaná v 2.2.1 a mobilní síť pro IoT popsané v 2.2.4. Kvůli požadavku na nízkou cenu nepřipadá v úvahu volba mobilní sítě, u které by se za každé zařízení muselo platit měsíční poplatek v řádu stovek korun při menším množství zařízení. Jako přenosová vrstva je tedy zvolena technologie LoRa.

Vstupní protokol, jehož podoba je vizualizovaná v 3.5, obsahuje 20 bajtů určených pro samotné naměřené hodnoty, které je nutné specifikovat. Tato část paketu bude



Obrázek 4.1 Rozložení datové části a příznakových znaků vstupního protokolu zařízení pro vinaře.

v případě zařízení pro vinaře obsahovat 3 hodnoty - teplotu, vlhkost a úroveň oxidu uhličitého. Dále je možné určit pro zařízení smysl příznakových bitů. Jelikož je zařízení umístěno několik metrů pod zemí, je nutné data nejdřív přenést na povrch a následně odeslat pomocí vstupního protokolu. Příznakové bity proto obsahují informaci o ztrátě a poškození zprávy přenášené na povrch. Datová část a příznakové bity jsou blíže specifikované na obrázku 4.1. První 4 bajty datové části obsahují teplotu v 32 bitovém desetinném čísle. Hodnota je v jednotkách stupňů Celsia. Další 4 bajty obsahují 32 bitovou desetinnou hodnotu vzdušné vlhkosti v procentech. Následuje 32 bitů dlouhé celočíselná kladná hodnota obsahující informaci o množství častic oxidu uhličitého na milion častic. Osm bajtů je nevyužitých a je možné je později využít pro další měření. Spodní čtyři bity podle specifikace obsahují čítač zpráv. Nejvyšší příznakový bit indikuje poškození zprávy při přenosu na povrch, tedy nevalidní hodnotu kontrolní sumy. Následující bit indikuje ztrátu zprávy při přenosu na povrch. Dva příznakové bity jsou ponechány bez významu pro pozdější využití.

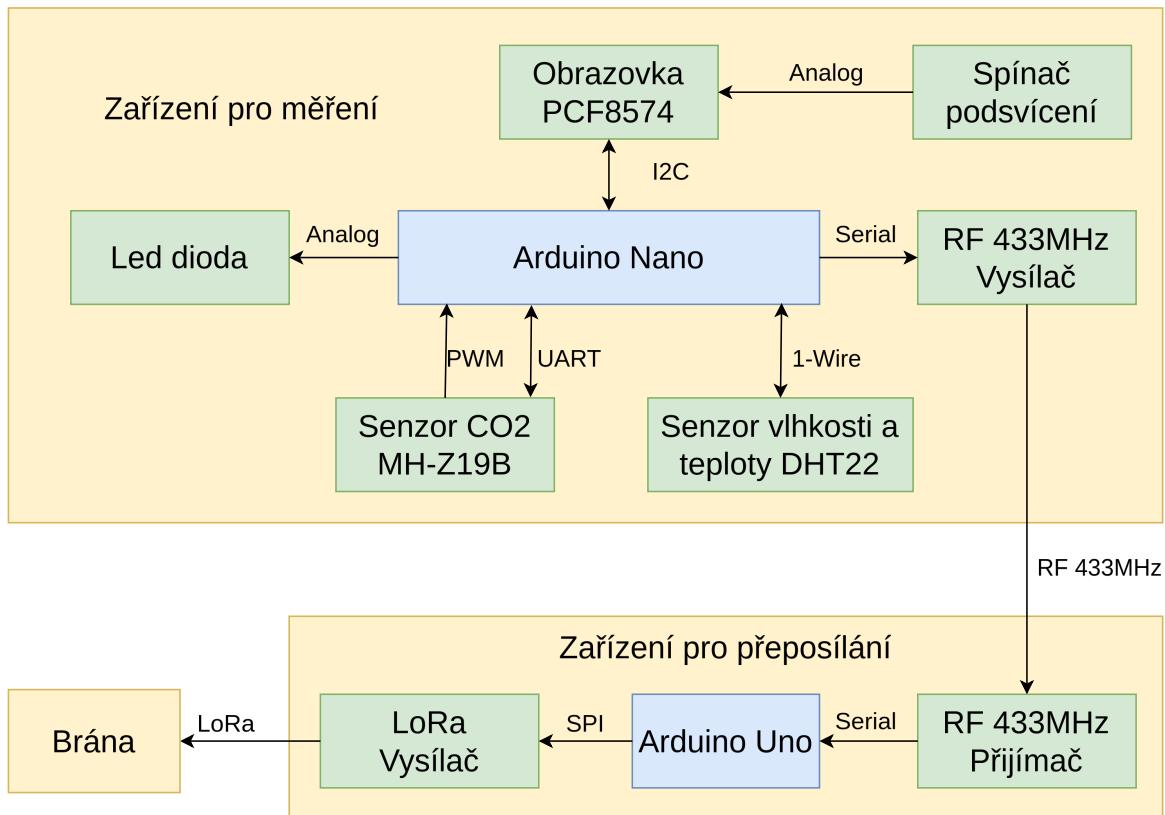
4.3.2 Návrh hardwaru

Jak již bylo zmíněno ve specializaci vstupního protokolu, zařízení provádějící samotné měření bude umístěno několik metrů podzemí. Komunikační vrstva LoRa není schopná komunikovat na velké vzdálenosti z tohoto prostředí. Je tedy nutné data nejdřív dostat na povrch jinou metodou než LoRa a následně je zaslat touto technologií na bránu. Tento problém lze řešit několika způsoby:

1. umístění antény LoRa na povrchu a zařízení ve sklepě,
2. vytvoření sekundárního zařízení určeného čistě pro přeposílání zpráv.

V případě prvního řešení nastává značný problém se sírou signálu. Je pravděpodobné, že anténa by musela být až několik desítek metrů dlouhá, aby byla umístěna až na povrchu, a bez aktivního zesílení signálu by velmi snadno docházelo k jeho rušení. Zvolena byla proto druhá možnost, ač je složitější pro implementaci. K zařízení umístěném ve sklepě bude vytvořeno sekundární zařízení určené pouze pro přeposílání dat. Komunikaci mezi těmito dvěma zařízeními lze vyřešit buď bezdrátově, nebo kabelovým spojením. V případě využití kabelu nám opět vzniká problém se vzdáleností a sírou

signálu. Zařízení proto budou komunikovat bezdrátově, pomocí technologie určené na kratší vzdálenosti s vyšší sílou signálu. Pro tuto komunikaci je zvolena pro nízkou cenu kombinace 433 MHz RF vysílače a přijímače. Bude ovšem nutné ověřit stabilitu a sílu spojení během implementace. Příznakové byty specifikované v 4.3.1 tedy indikují poškození, respektive ztrátu zprávy právě v této části komunikace. Pokud tedy brána obdrží dvě zprávy, první s čítačem o hodnotě 8 a druhou s hodnotou čítače 10, právě díky příznakovému bitu bude možné rozpoznat, zda ke ztrátě zprávy došlo pro přenosu přes LoRa mezi zařízením pro přeposílání a bránou, nebo přes RF při přenosu ze sklepa na povrch.



Obrázek 4.2 Komponenty zařízení pro vinaře a jejich komunikace.

Diagram 4.2 obsahuje přehled jednotlivých komponent a jejich zapojení. Zařízení pro měření obsahuje Arduino Nano jakožto výpočetní modul. Ten byl vybrán z důvodu relativně nízké ceny, nízkého odběru energie a celkové softwarové podpory. Pro Arduino existuje nespočet knihoven pro různé senzory, což sníží nároky na množství kódu a tím i cenu. Zařízení je osazeno senzorem DHT22 určeného k měření teploty a vlhkosti. Senzor byl zvolen primárně pro svou nízkou cenu a jednoduchost. Pro měření úrovně oxidu uhličitého byl v návrhu zvolen senzor MH-Z19. Senzor detekuje množství oxidu uhličitého pomocí infračerveného záření a měření absorpce tohoto záření plynem. Senzor také poskytuje data o teplotě. Tyto dva senzory zajišťují splnění požadavku 2. Zařízení dále obsahuje obrazovku, se kterou komunikuje pomocí protokolu I2C. Obrazovka umožňuje

zapínání podsvícení a proto je k ní umístěn i polohový spínač. Obrazovka je přidána pro splnění požadavku 3. Pro vizuální upozornění uživatele na vysokou koncentraci oxidu uhličitého je na zařízení umístěna červená led dioda, tím je splněn požadavek 4. Pro komunikaci se zařízením pro přeposílání je jednotka osazena RF vysílačem, toto spojení zajišťuje schopnost komunikace z podzemních prostor specifikovaných požadavkem 1. U zařízení musí být ve finální úpravě zajištěna vodotěsnost, aby byl splněn požadavek 5.

Zařízení pro přeposílání dat obsahuje Arduino Uno jako výpočetní modul a dále je osazeno RF přijímačem a LoRa vysílačem. Zprávy vyměňované mezi zařízeními nemají formát vstupního protokolu, obsahují pouze naměřené hodnoty, čítač a kontrolní sumu.

4.4 Specifikace návrhu měřícího zařízení pro včelaře

Zařízení pro včelaře musí být schopné pravidelného měření hmotnosti včelích úlů. Primárním účelem je tedy monitorování váhy za účelem určení doby pro dokrmení včelstva nebo stáčení medu. Zařízení bude umístěno ve volném prostranství. Tato sekce popisuje specializaci vstupního protokolu pro toto zařízení a návrh zařízení.

4.4.1 Specializace vstupního protokolu

Zařízení vyžaduje stejně jako zařízení pro vinaře komunikaci na velké vzdálenosti. Z toho důvodu je zde opět zvolena technologie LoRa jakožto přenosová vrstva pro vstupní protokol - tím je zaručeno splnění požadavku 1. Vstupní protokol obsahuje tři hodnoty - teplotu, vlhkost a hmotnost. Teplota je v datové části umístěna na prvních 4 bajtech a je reprezentována desetinnou hodnotou ve stupních Celsia. Následující 4 bajty obsahují vlhkost ve formě desetinné hodnoty v procentech. Třetí hodnota obsahuje hmotnost, jedná se opět o 4 bajty a desetinné číslo. Příznakové bity jsou u tohoto zařízení zatím nevyužity. Zařízení umožňuje rozšíření o další 8 bajtů dat a o přidání 4 příznakových bitů. Rozložení datové části a příznakových znaků vstupního protokolu je vizualizováno na obrázku 4.3.

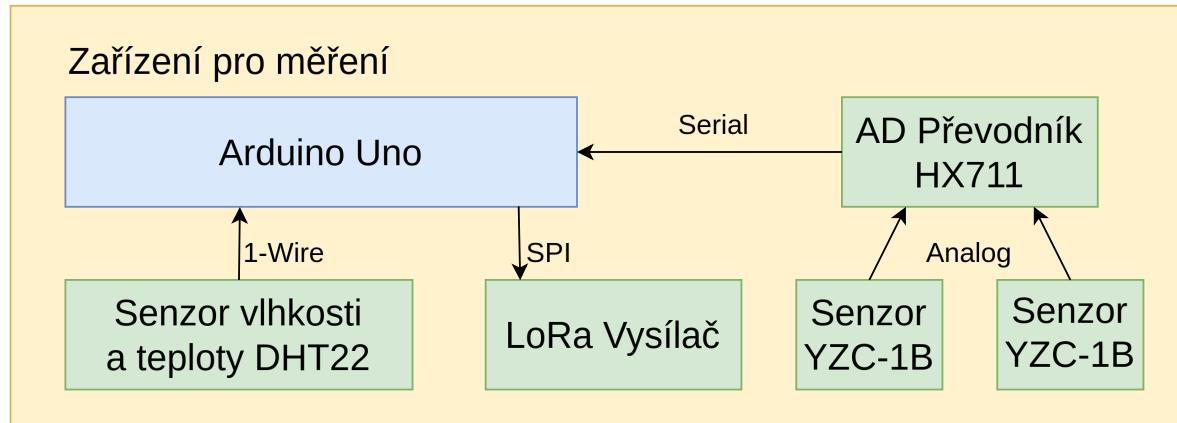
Příznakové bity		Data					
.....	rezervováno 4b	Čítadlo 4b	Teplota [°C] float32	Vlhkost [%] float32	Hmotnost [kg] float32	rezervováno 8b

Obrázek 4.3 Rozložení datové části a příznakových znaků vstupního protokolu zařízení pro včelaře.

4.4.2 Návrh hardwaru

Jako výpočetní jednotka je z důvodu ceny a dostupnosti zvolen jednočipový počítač Arduino Uno. Arduino Uno je možné napájet bateriově a tím dosáhnout splnění po-

žadavku 3. Stejně jako u zařízení pro vinaře je pro měření teploty a vlhkosti zvolen senzor DHT22. Pro váhy je použit senzor YZC-1B, který je analogový a pracuje na principu dvou slitin s různými pevnostmi v tahu a siloměru. Pro jeden úl jsou využity dva tyto senzory připojené k analogově-digitálnímu převodníku HX711. Díky těmto senzorů je splněn požadavek 2. Zařízení musí být finálně upraveno tak, aby bylo odolné proti počasí a tím byl splněn požadavek 4.



Obrázek 4.4 Komponenty zařízení pro včelaře a jejich komunikace.

5 IMPLEMENTACE

Tato kapitola se věnuje popisu implementace systému navrženém v 3 a fyzických zařízení pro měření pomocí senzorů využívajících tento systém. Zdrojové kódy k implementaci systému jsou umístěny na přiloženém paměťovém médiu. Zdrojové kódy jsou také dostupné v internetovém repozitáři GitHub¹⁾. Repozitář obsahuje veškeré zdrojové kódy, výsledky testů, testovací data a konfigurační soubory.

5.1 Struktura repozitáře

Základní složková struktura repozitáře je následující:

```
android_app
├── device
│   ├── bee_scales
│   ├── libs
│   └── wine_cellar
├── gateway
│   └── lora_gateway
└── latex
└── server
    ├── data_generator
    ├── deployment
    ├── flask_endpoint
    ├── influxdb
    └── uploader
└── testing
```

Složka *android_app* obsahuje zdrojové kódy pro mobilní aplikaci. Složka *device* obsahuje firmware pro zařízení pro včelaře, firmware pro zařízení pro vinaře a potřebné knihovny. Složka *gateway* obsahuje implementaci brány pro zařízení. Složka *latex* obsahuje zdrojové kódy pro textovou část práce. Složka *server* obsahuje veškeré zdrojové kódy pro serverovou část aplikace (platformu). Poslední složka *testing* obsahuje veškerá data vygenerovaná či potřebná pro testování celého systému.

5.1.1 Brána pro zařízení

Zdrojové soubory pro bránu se nalézají ve složce *gateway/lora_gateway*. Aplikace implementuje návrh brány specifikovaný v 3.4. Aplikace je psaná v jazyce C++ ve verzi 20 a pro sestavování využívá nástroj CMake. Aplikace je závislá na knihovnách:

- **Boost** pro práci s konfiguračními soubory ve formátu JSON,
- **PahoMqttCpp** pro implementaci MQTT komunikace,

¹⁾<https://github.com/Nothrax/dp>

- **spdlog** a **fmt** pro logování,
- a **wiringPi** pro komunikaci s LoRa modulem.

Aplikace je určena pro běh na počítači Raspberry Pi 4 s architekturou arm64, ale překlad je umožněn i pro architekturu x86-64 a může být spouštěn i pomocí nástroje Docker. Návod pro přeložení, konfiguraci a běh aplikace je popsán v souboru *README.md* v kořenovém adresáři projektu brány.

Přenosovou vrstvou pro vstupní protokol byla v návrhu zvolena technologie LoRa. Pro použití této technologie je potřebný přijímač. Jako přijímač byl zvolen modul E32-868T20D. LoRa modul komunikuje s počítačem pomocí protokolu UART a vyžaduje napájení 5V. Přenosová vrstva pro výstupní protokol je MQTT. Aplikace plně implementuje vstupní i výstupní protokol v plném rozsahu jak jsou definovány v 3.4.1 a 3.4.2.

Aplikace kromě implementace přijímání zpráv přes LoRa umožňuje pro účely testování také simulaci zařízení bez nutnosti přítomnosti fyzického zařízení či modulu LoRa. Tato simulace umožňuje integrační testování celého systému v rámci Docker kontejnerů. Simulované zařízení v aktuální implementaci poskytuje konstantní hodnoty každých několik vteřin. Součástí projektu je podpora pro automatizovaný běh aplikace pomocí linuxového nástroje **systemd**, konfigurační soubor pro tuto podporu je umístěn v *resources/systemd/gateway.service*

Aplikace při inicializaci vyžaduje konfigurační soubor ve formátu JSON. Konfigurační soubor obsahuje následující sekce.

- **generic_settings** - sekce definuje obecné nastavení programu, jako je logování a identifikátor brány a společnost.
- **device_settings** - sekce definuje nastavení pro vstupní protokol. V sekci je programu předán list známých zařízení a způsob komunikace, tedy zda jsou zprávy generovány pomocí simulace zařízení nebo přijímány pomocí protokolu LoRa. Pro protokol LoRa je v sekci definováno zapojení pinů a parametry UART komunikace.
- **output_settings** - sekce definuje nastavení výstupního protokolu, tedy parametry MQTT spojení.

Součástí projektu je příklad konfiguračního souboru pro běh s modulem LoRa a konfigurační soubor pro běh v Docker kontejneru a simulací zařízení. Konfigurační soubory jsou umístěny ve složce *resources/configuration/*. Aplikace umožňuje zabezpečení MQTT spojení pomocí technologie SSL. Příklad konfiguračního souboru využívající modul LoRa je umístěn v přílohách 1.

5.1.2 Aplikace pro nahrávání dat

Zdrojové kódy pro aplikaci pro nahrávání dat jsou umístěny ve složce *server/uploader*. Aplikace implementuje návrh specifikovaný v 3.5. Aplikace je psaná v jazyce Python pro verzi 3.10 a vyšší. Aplikace využívá knihovny:

- **paho-mqtt** pro komunikaci přes MQTT,
- **influxdb-client** pro komunikaci s databází InfluxDB a
- **mysql-connector-python** pro komunikaci s relační databází.

Aplikace je určena pro běh na serverové části systému (platformy) a stejně jako brána je možné aplikaci kontejnerizovat pomocí nástroje Docker. Zde kontejnerizace ovšem není přidána z důvodu integračního testování, ale je zamýšleno aplikaci spouštět pomocí kontejneru i v produkčním prostředí.

Aplikace plně implementuje výstupní protokol tak, jak je definován v 3.4.2. Zabezpečení výstupního protokolu je řešeno pomocí SSL. Aplikace je konfigurována pomocí konfiguračního souboru ve formátu JSON. Příklad konfiguračního souboru je umístěn ve složce *config* a je umístěn v přílohách 1. Konfigurační soubor obsahuje následující sekce:

- **mqtt_settings** pro nastavení parametrů MQTT spojení,
- **timeseries_settings** pro nastavení spojení s databází InfluxDB,
- **relation_settings** pro nastavení spojení s relační databází.

Návod pro spouštění aplikace a popis konfiguračního souboru je umístěn v souboru *README.md* v kořenovém adresáři projektu.

5.1.3 Aplikace pro získání dat

Zdrojové kódy jsou umístěny ve složce *server/flask_endpoint*. Aplikace plně implementuje návrh a HTTP rozhraní specifikované v 3.8. Aplikace je psaná v jazyce Python pro verzi 3.10 a vyšší. Aplikace využívá knihovny:

- **influxdb-client** pro komunikaci s databází InfluxDB a
- **mysql-connector-python** pro komunikaci s relační databází.
- **Flask** pro vytvoření HTTP rozhraní.

Stejně jako aplikace pro nahrávání dat je tato aplikace určena pro běh v kontejneru Docker na serverové části (platformě). Aplikace je konfigurována pomocí konfiguračního souboru ve formátu JSON. Příklad konfiguračního souboru je umístěn ve složce *config* a je umístěn v přílohách 1. Konfigurační soubor obsahuje následující sekce:

- **flask_settings** pro nastavení parametrů HTTP rozhraní,
- **timeseries_settings** pro nastavení spojení s databází InfluxDB,
- **relation_settings** pro nastavení spojení s relační databází.

Návod pro spouštění aplikace a popis konfiguračního souboru je umístěn v souboru *README.md* v kořenovém adresáři projektu.

5.1.4 Databáze a nasazení systému

Zdrojové soubory určené pro nasazení databází a také celého systému jsou umístěny ve složce *server/deployment*. Pro účely jednoduchého nasazení celého systému byl zvolen nástroj Docker, díky kterému je možné spouštět jednotlivé části systému v kontejnerech bez nutnosti instalace všech závislostí. Ve složce se nachází soubor *docker-compose.yml*, který obsahuje reference na všechny kontejnery spustitelné v rámci systému. Jediné programy, které nejsou součástí, jsou samotná zařízení pro včelaře a vinaře. Jejich existenci lze ovšem nasimulovat režimem generátoru v bráně.

Kontejnery nezaručují uchování konfigurace a dat, při jejich smazání či restartu dochází ke smazání veškerého nastavení a uložených dat. Z toho důvodu jsou přítomny dvě složky a to *configs* a *storage*. První zmiňovaná složka obsahuje konfigurační soubory pro jednotlivé instance určené pouze pro čtení. Druhá složka obsahuje data vygenerovaná aplikacemi, která musí zůstat konzistentní. Tímto přístupem je zajištěno uložení stavu systému i přes smazání kontejnerů a jeho přenositelnost.

Docker pro nasazení systému obsahuje následující kontejnery:

- **influxdb** - kontejner je zodpovědný za spuštění databáze InfluxDb ve verzi 2.1. Složka s konfiguracemi obsahuje konfigurační soubor pro instanci. Do sekce pro ukládání jsou nahrávná data ukládána do databáze a databázové schéma. Výchozí přístupové údaje jsou uloženy v souboru *.env*. Instance obsahuje i možnost přístupu přes internetovou stránku pro účely snazšího generování přístupových údajů.
- **mosquitto** - jedná se o kontejner obsahující MQTT broker. Kontejneru je poskytován konfigurační soubor s SSL certifikáty. Logy jsou ukládány do připnutého úložiště.

- **gateway** - kontejner vytvářený v rámci diplomové práce obsahuje spustitelnou verzi brány pro přístup zařízení. Pomocí konfiguračního souboru je brána nastavena pro generování dat z fiktivního zařízení. Do připojeného úložiště jsou ukládány logy.
- **uploader** - kontejner vytvořený v rámci práce, obsahuje aplikaci pro nahrávání dat do databáze. Program je nastaven pomocí konfiguračního souboru.
- **flask_endpoint** - kontejner vytvořený v rámci práce, jedná se o aplikaci pro přístup k datům nastavenou konfiguračním souborem.
- **mariadb** - kontejner obsahující relační databázi. Konfigurace a schéma databáze včetně základního uživatele je vytvořeno pomocí konfiguračního souboru. Vytvořená data jsou uložena v připnutém úložišti.
- **adminer** - webová aplikace sloužící pro jednoduchý přístup k relační databázi za účelem vytváření záznamů.
- **grafana** - aplikace určená pro správu notifikací a základní vizualizace dat v časové databázi influx.

Dochází ke spouštění tří kontejnerů, které nejsou veřejně dostupné a je nutné jejich lokální vygenerování. K tomu slouží bash skript *build-docker.sh*, který provádí vytvoření kontejnerů **flask_endpoint**, **uploader** a **gateway**.

Docker compose je spouštěn pomocí několika předdefinovaných profilů pomocí příkazu `docker compose profile=<název> up`. Každý kontejner má vlastní profil, tedy při spuštění příkazu `docker compose profile=influxdb up` je spuštěn pouze kontejner s časovou databází. Dále jsou obsaženy profily:

- **all** profil spouští všechny kontejnery, je přítomen za účelem simulace celého systému včetně zařízení,
- **platform** profil spouští platformní část, tedy všechny kontejnery bez brány, a je určen pro finální nasazení,
- **database** profil spouští pouze databáze a kontejnery pro manipulaci s nimi.

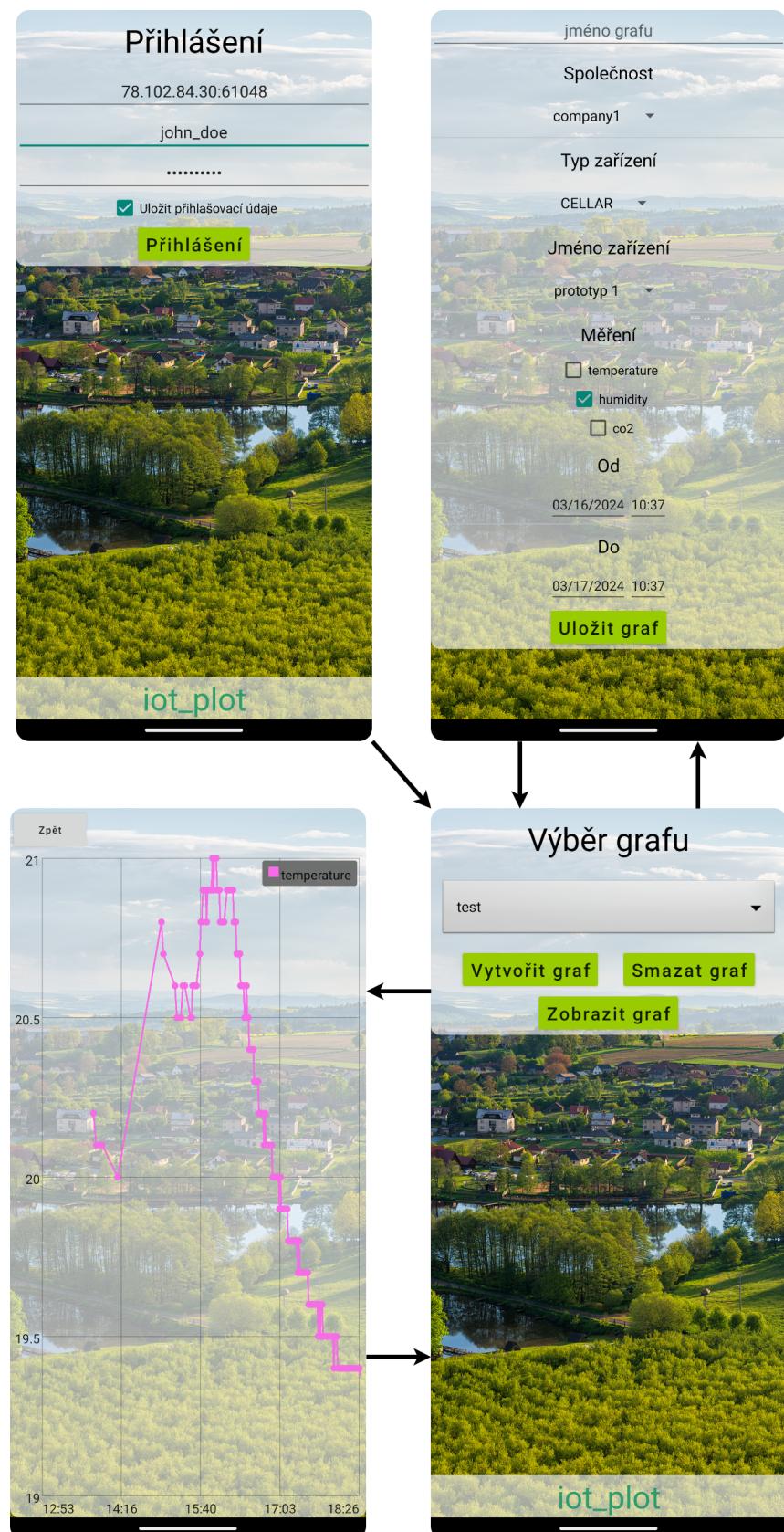
Součástí kontejnerizace kromě zařízení není ani mobilní aplikace z důvodu nedostatečné podpory virtualizace android zařízení pomocí nástroje Docker. Spouštění mobilní aplikace ve virtuálním prostředí je řešeno pomocí android emulátor.

5.1.5 Mobilní aplikace

Zdrojové kódy k mobilní aplikaci jsou umístěny ve složce *android_app*. Aplikace plně implementuje a rozšiřuje návrh z 3.9. Aplikace je psaná v jazyce Kotlin a je určená pro zařízení s operačním systémem Android. Aplikace pro správné fungování vyžaduje připojení k HTTP rozhraní implementovaném aplikací pro přístup k datům. Aplikace navíc oproti návrhu obsahuje možnost uložení předdefinovaných grafů pro pozdější zobrazení. Aplikace obsahuje několik obrazovek:

- obrazovku pro přihlášení vyžaduje po uživateli přihlašovací údaje pro přístup k datům, po přihlášení se uživatel dostane k výběru grafů,
- výběr grafů umožňuje uživateli vybrat již vytvořený graf a zobrazit si ho či přejít na obrazovku s vytvořením grafů,
- vytvoření grafů nabídne uživateli seznam všech zařízení podle společnosti, výběr hodnot a časového rozsahu pro zobrazení dat,
- zobrazení grafu vizualizuje data.

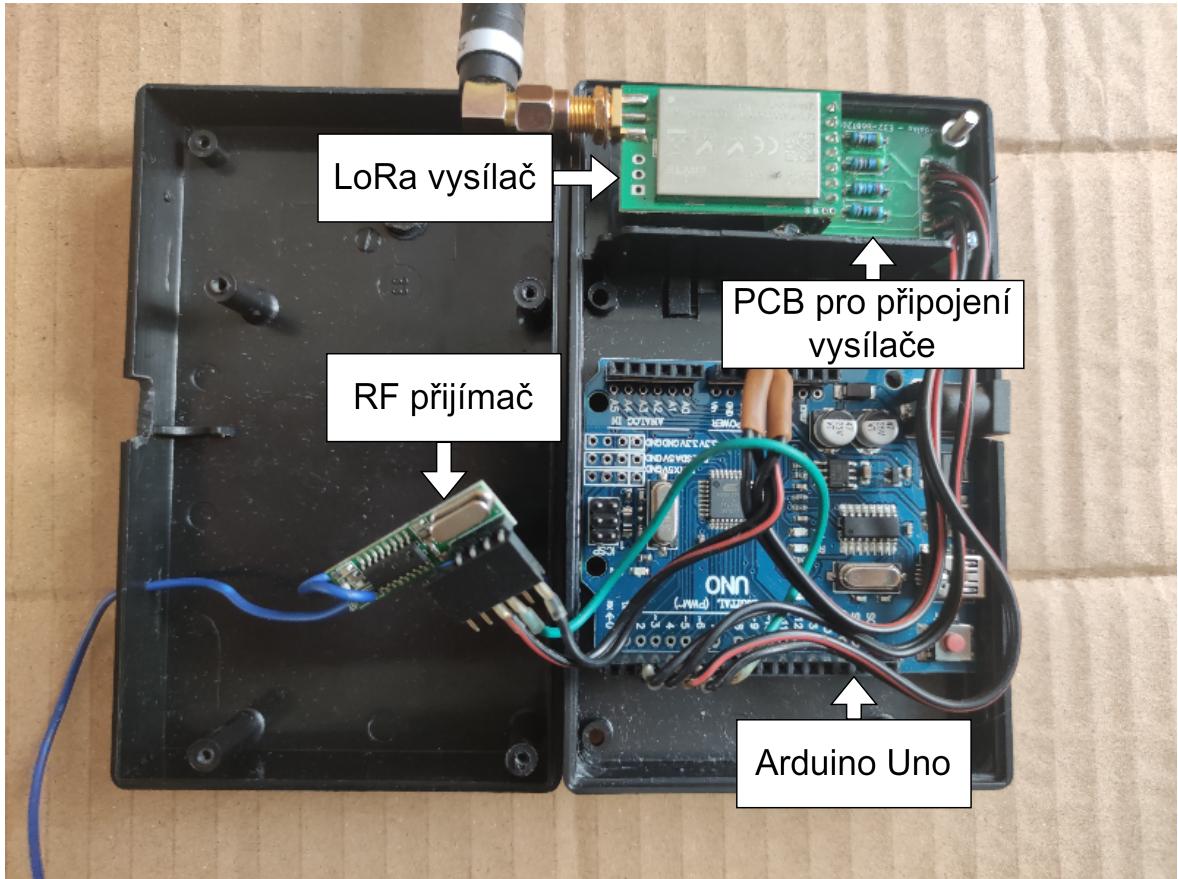
Jednotlivé obrazovky a směr přechodů mezi nimi je možné vidět na obrázku 5.1. V levém horním rohu se nachází obrazovka pro přihlášení uživatele. Z této obrazovky se provádí přechod na obrazovku výběru grafů umístěnou v pravém spodním rohu. Obrazovka pro vytváření grafů je zobrazena v pravém horním rohu a obrazovka se zobrazeným grafem teplot je umístěna v levém horním rohu. Obrázky jsou pořízeny s reálnými daty ve funkční aplikaci. Aplikace podporuje jak vertikální tak horizontální zobrazení pro lepší podporu vykreslování grafů. Uživatelské jméno, heslo a adresu serveru je možné uložit do perzistentního úložiště. Stejně tak je možné uložit předdefinované grafy pro specifické zařízení pro určený časový interval. Na perzistentní úložiště nejsou uloženy body grafu, ale pouze metadata nutná pro získání těchto bodů.



Obrázek 5.1 Obrazovky mobilní aplikace a přechody mezi nimi.

5.1.6 Zařízení pro vinaře

Vytvoření a implementace zařízení pro vinaře vychází z návrhu 4.3. Implementace se skládá z dvou fyzických zařízení, a to zařízení pro měření dat a zařízení pro přenos dat.

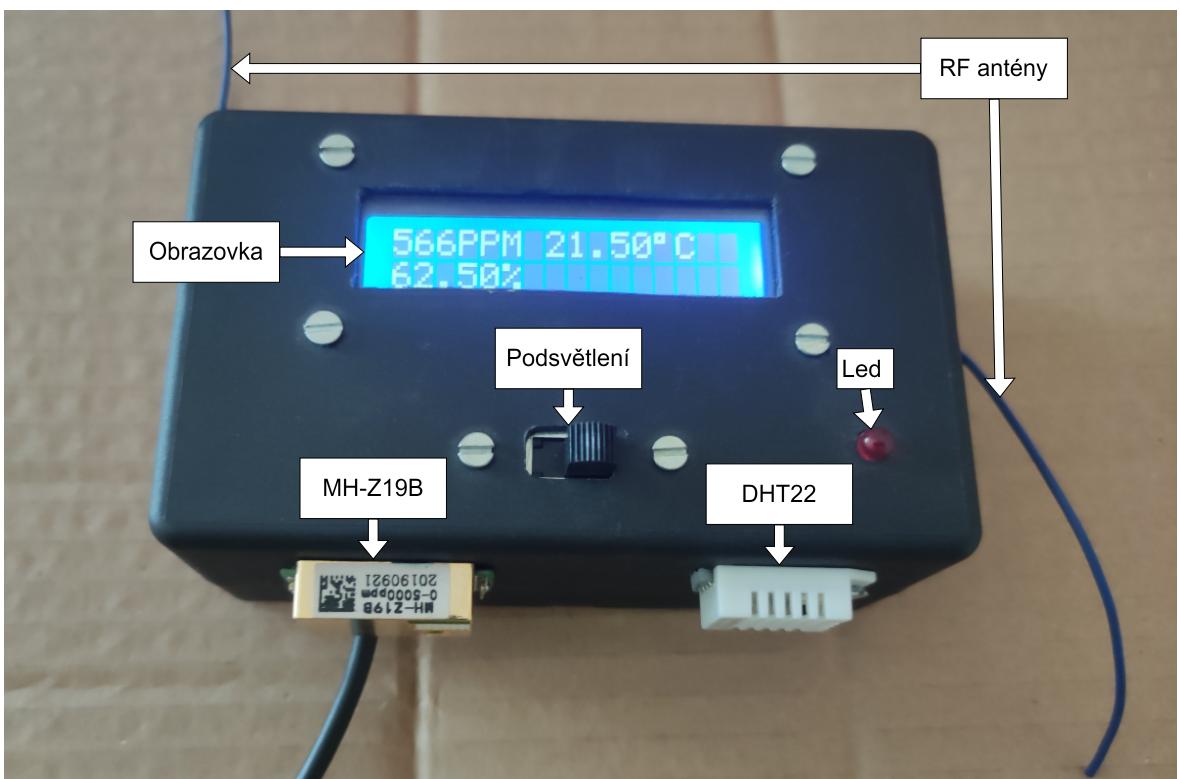


Obrázek 5.2 Zařízení pro přenos dat od zařízení pro vinaře.

Zařízení pro přenos dat je postaveno na jednočipovém počítači Arduino Uno a je osazené LoRa vysílačem E32_868T20D a RF přijímačem NiceRF 433MHz (wl101-341). Schéma zapojení zařízení je možné vidět v příloze 2. Obrázek vytvořeného zařízení je možné vidět na 5.2. Na obrázku jsou vyznačeny jednotlivé součástky. Pro usnadnění zapojení vznikl v rámci práce vlastní PCB adaptér pro připojení desky E32_868T20D. Adaptér je na obrázku také viditelný a vyznačený. Zdrojové soubory pro ovládání jednočipového počítače jsou umístěny v adresáři *device/wine_cellar/fm_lore_resender*. Knihovny potřebné pro sestavení projektu jsou umístěny ve složce *device/libs/arduino*. Zařízení po příjetí zprávy pomocí RF přijímače zkонтroluje čítač zprávy a kontrolní sumu, zprávu překonvertuje do formátu vstupního protokolu a pomocí LoRa vysílače zprávu odešle bráně. Veškeré parametry, jako je typ a číslo zařízení a parametry LoRa komunikace, lze změnit v zdrojových souborech. Hodnoty jsou definovány jako makra či globální proměnné a jsou vyznačeny komentářem. Jak bylo zmíněno v návrhu, zaří-

zení pro přeposílání kontroluje čítadlo a kontrolní sumu z RF komunikace. Při ztrátě zprávy při zaslání následující zprávy přes LoRa signalizuje ztrátu příznakovým bitem. Při poškození zprávy ji i tak přeposílá, ale opět je tato situace označena příznakovým bitem.

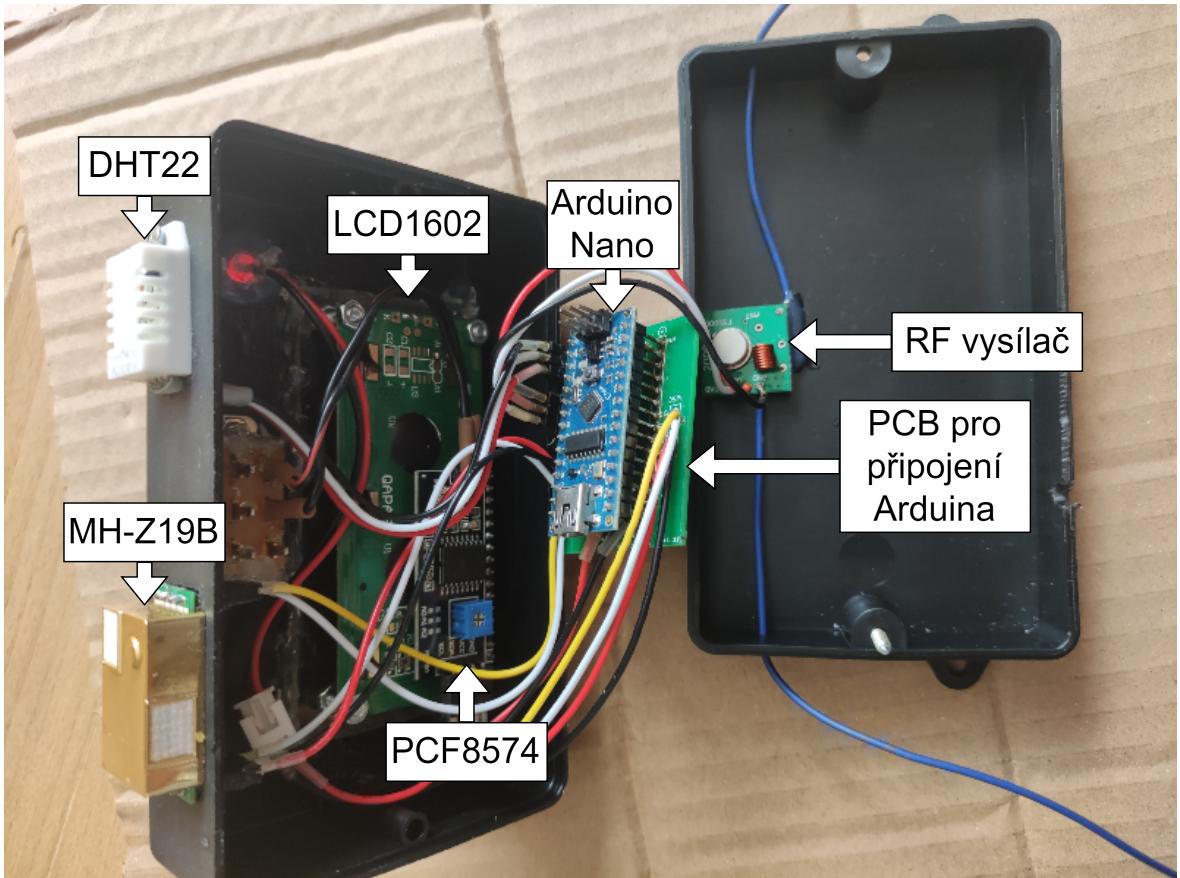
Zařízení pro měření je postaveno na jednočipovém počítači Arduino Nano. Zařízení je osazeno RF vysílačem NiceRF 433MHz, senzorem koncentrace oxidu uhličitého MH-Z19B, senzorem teploty a vlhkosti DHT22, LCD obrazovkou LCD1602 s převodníkem PCF8574, červenou led diodou pro upozornění na vysokou koncentraci oxidu uhličitého a spínačem pro zapínání podsvětlení obrazovky. Schéma zapojení zařízení je v příloze 2. Schémata připojení byla vytvořena ve volně dostupném programu Kicad. Fotografi vytvořeného zařízení včetně popisu jednotlivých částí je možné vidět na obrázku 5.3. Na zařízení jsou viditelně zobrazeny naměřené hodnoty. Pro usnadnění servisních zásahů na zařízení vznikla v rámci práce specializovaná PCB deska na které je jednočipový počítač umístěn a která obsahuje konektory pro ostatní komponenty a senzory na zařízení. Desku a vnitřní zapojení zařízení je možné vidět na obrázku 5.4. Každý komponent je připojen přes rozpojitelný konektor.



Obrázek 5.3 Měřící zařízení pro vinaře.

Zdrojové soubory pro měřící zařízení jsou umístěny ve složce *device/wine_cellar/device*. Knihovny nutné pro sestavení jsou umístěny ve složce *device/libs/arduino*. Veškeré parametry měření jsou konfigurovatelné pomocí globálních proměnných a maker, která jsou okomentovaná. Zařízení provádí měření v časových intervalech definovaných

proměnnou MEASUREMENT_DELAY_MS. Led světlo je zapnuto po překročení specifické hodnoty koncentrace oxidu uhličitého definované hodnotou CO2_THRESHOLD. Při nastavení frekvence zasílání dat je nutné brát omezení přenosové vrstvy LoRa vysvětleném v 2.2.1. Naměřené hodnoty se zobrazují na interním LCD displeji, jehož podsvícení lze aktivovat či deaktivovat posuvným spínačem.



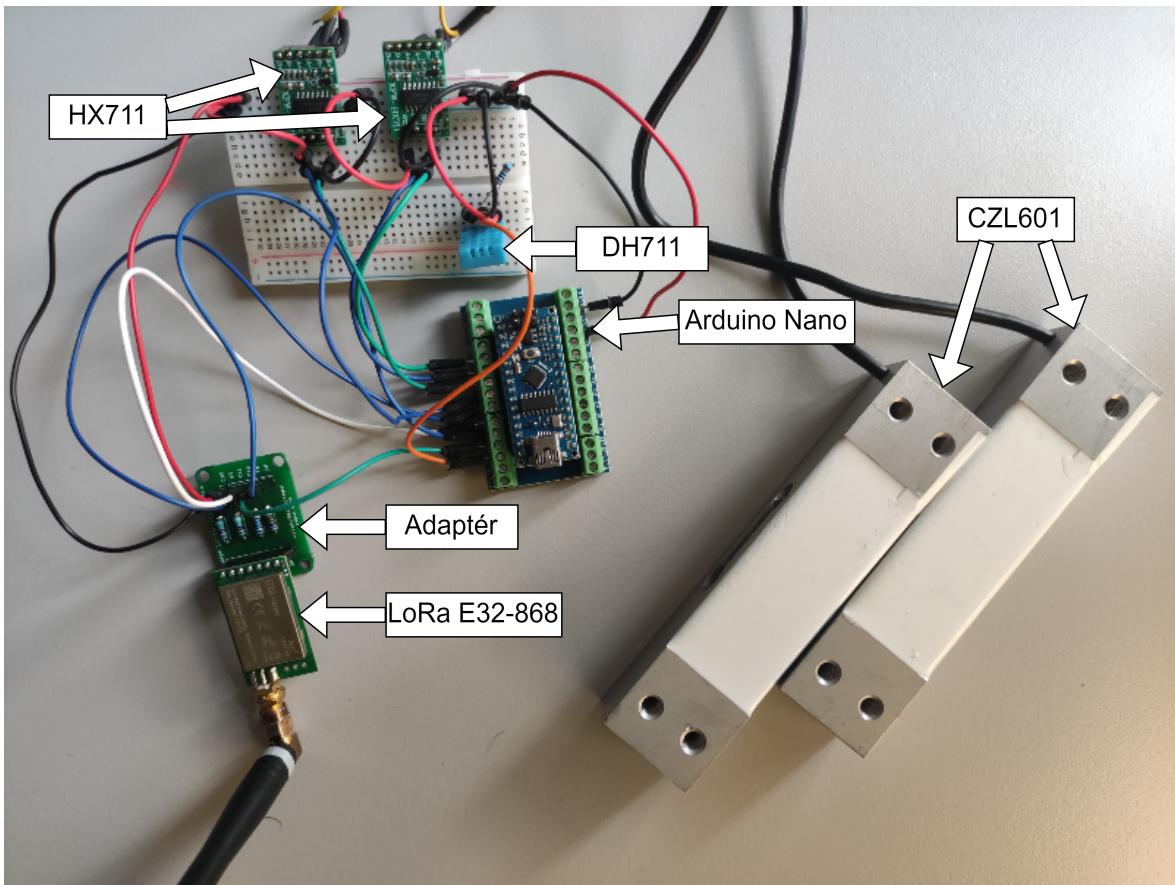
Obrázek 5.4 Zapojení měřícího zařízení pro vinaře.

5.1.7 Zařízení pro včelaře

Vytvoření a implementace zařízení pro včelaře vychází z návrhu 4.4. Implementace se skládá z jednoho fyzického zařízení, které provádí měření dat, přesněji hmotnosti včelích úlů, vzdušné vlhkosti a teploty.

Zařízení se skládá z:

- jednočipového počítače Arduino Nano,
- dvou tenzometrických snímačů zatížení CZL601 s kapacitou 100kg,
- dvou analogově-digitálních převodníků HX711,
- senzoru vlhkosti a teploty DHT11,



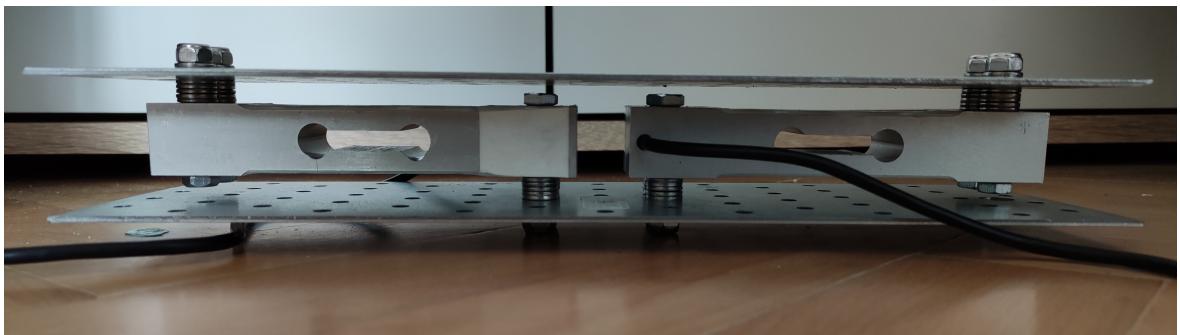
Obrázek 5.5 Zapojení měřícího zařízení pro včelaře.

- adaptéru vlastní výroby pro E32-868T20D a
- LoRa vysílače E32-868T20D.

Celé zařízení ovládá jednočipový počítač Arduino Nano. Na počítač jsou připojeny dva AD převodníky HX711 a ke každému z převodníků je připojen jeden tenzometrický snímač zatížení. Snímače zařízení mají každý kapacitu 100 kg a jsou určeny oba pod jeden úl, aby došlo k rozložení zatížení a byla zajištěna vyšší stabilita. Tyto dva senzory se starají o snímání hmotnosti včelího úlu. O snímání teploty a vlhkosti se stará senzor DHT11 přímo spojený s počítačem. Stejně jako zařízení pro přeposílání, které je součástí zařízení pro včelaře, využívají váhy pro komunikaci přes protokol LoRa modul E32-868T20D. Modul je k počítači připojen pomocí PCB adaptéru vlastního návrhu. Zapojení zařízení s popisem jednotlivých komponent je možné vidět na obrázku 5.5. Samotná platforma určená k vážení, která je umístěna pod včelí úly, je na obrázku 5.6. Schéma zapojení zařízení je umístěno v přílohách 2 a bylo vytvořeno ve volně dostupném programu KiCad.

Zdrojové soubory měřícího zařízení pro včelaře jsou umístěny ve složce *device/bee_scales*. Knihovny nutné pro sestavení programu jsou umístěny ve složce *device/libs*.

Program lze přeložit a nahrát pomocí oficiálního vývojového nástroje Arduino IDE. Veškeré parametry měření jsou konfigurovatelné pomocí globálních proměnných a makr, která jsou okomentovaná. Zařízení provádí měření v časových intervalech definovaných proměnnou **MEASUREMENT_DELAY_MS**. Naměřené hodnoty jsou pro účely ladění také posílány přes sériovou linku.



Obrázek 5.6 Platforma pro vážení úlu.

6 TESTOVÁNÍ

Tato kapitola se věnuje testování implementovaného systému pro sběr, analýzu a zobrazení dat v zemědělství. Testy jsou rozděleny do tří částí a to z:

- testů splnění požadavků na systém popsaných v 3.2,
- testů splnění požadavků na jednotlivá zařízení popsaných v 4.2,
- testu celého systému v reálném provozu.

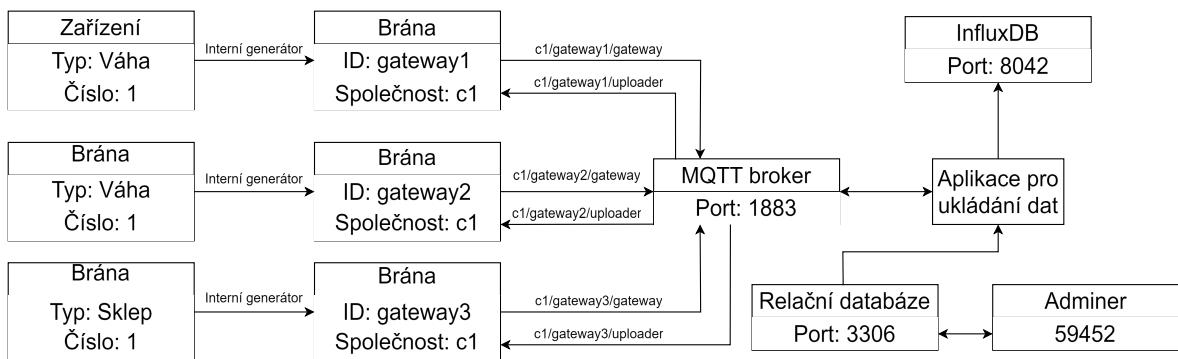
Každý test obsahuje popis provedení, který požadavek je testován a vyhodnocení splnění požadavku.

6.1 Testování požadavků na systém

Tato sekce se věnuje testování požadavků na systém popsaných v 3.2. Požadavky nejsou testovány samostatně, ale jednotlivé modelové případy jsou schopné otestovat více požadavků najednou.

6.1.1 Periodické měření více zařízení a ukládání dat

Tento modelový případ pokrývá požadavky 1, 2, 3. Cílem testu je ověřit schopnost systému ukládat periodická data z několika připojených zařízení. Zapojení jednotlivých komponent systému je možno vidět na diagramu 6.1. Testovaný systém obsahuje tři instance brány pro připojení zařízení a po jedné instanci MQTT broker, aplikace pro nahrávání dat, časové databáze a relační databáze a aplikace pro správu relační databáze Adminer. Test není prováděn s reálným zařízením, ale u bran je zvolen režim generátoru, cílem testu totiž není ověřit funkčnost zařízení jako takových, ale schopnost systému uchovat data obdržená branou.



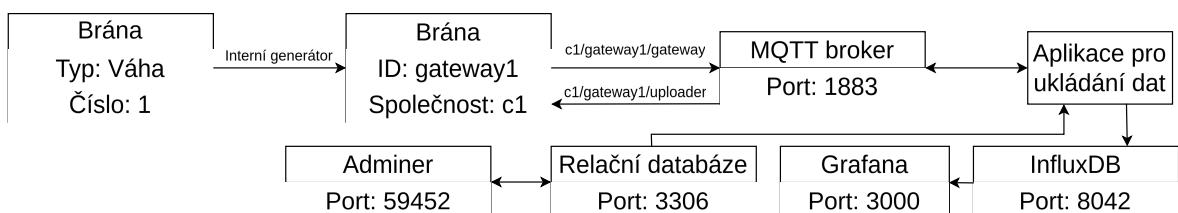
Obrázek 6.1 Diagram systému pro test periodického měření a ukládání dat.

Příprava a průběh testu Test je spuštěn pomocí scénáře nástrojem **docker compose**. Soubory nutné ke spuštění, včetně všech konfiguračních souborů jsou umístěny v repozitáři ve složce *testing/test_case_1* a celý scénář se spouští příkazem `docker compose -profile=test_case_1 up`. Veškeré aplikace a databáze jsou inicializovány z konfiguračních souborů a není nutné je nijak upravovat. Po spuštění kontejnerů dojde k postupnému propojení všech částí systému a všechna tři zařízení začnou úspěšně zapisovat do časové databáze.

Vyhodnocení testu Spuštění scénáře proběhlo úspěšně. Po spuštění začnou brány okamžitě generovat zprávy, ovšem platformní část systému ještě není zcela aktivní, aplikace pro ukládání dat není schopna spuštění bez inicializované časové a relační databáze. To ovšem není problém, protože brány ukládají nepotvrzené zprávy do lokálního úložiště. Inicializace všech komponent zabere zhruba 10-20 sekund, po inicializaci všech komponent jednotlivé brány díky aplikaci pro příjem dat pravidelně zapisují data do časové databáze. Tento test prokázal schopnost systému odbavovat více zařízení různých typů v jednu chvíli, schopnost periodicky ukládat data do časové databáze. Tedy splnění požadavků 1, 2, 3 bylo testem potvrzeno.

6.1.2 Test upozornění uživatele a filtrace dat

Modelový příklad pokrývá požadavky 4, 5 a 7. Cílem testu je ověřit schopnost systému filtrovat data, detekovat odpojené zařízení a upozornit uživatele na překročení limitní hodnoty. Zapojení jednotlivých komponent systému je možno vidět na diagramu 6.2. Testovaný systém obsahuje jednu instanci aplikací: brány se simulátorem zařízení, MQTT broker, nahrávání dat, relační a časové databáze, Grafana, Adminer.

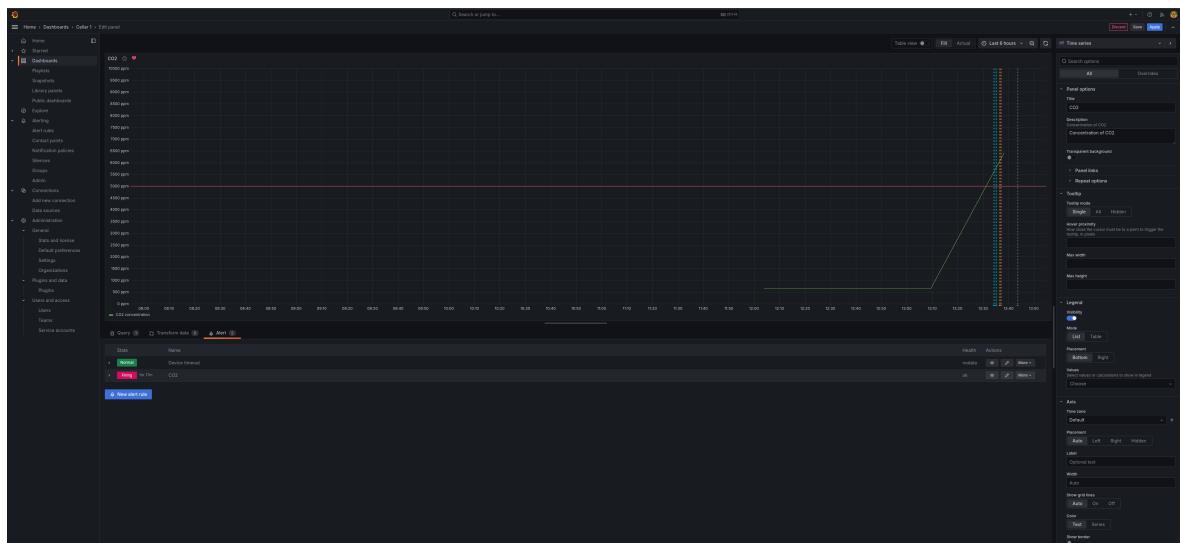


Obrázek 6.2 Diagram systému pro test notifikací.

Příprava testu Pro účely testu byl vytvořen skript pro nástroj **docker compose**. Soubory potřebné ke spuštění, včetně všech konfiguračních souborů jsou umístěny v repozitáři ve složce *testing/test_case_2* a spouští se příkazem `docker compose -profile=test_case_2 up`. Skript spouští pouze serverovou část, přesněji řečeno MQTT broker, aplikaci pro nahrávání, databáze, Adminer a aplikaci Grafana. Právě aplikace Grafana slouží k nastavení analýzy dat a zasílání upozornění uživateli. V aktuální

implementaci je pomocí simulátoru možné zasílat pouze konstantní hodnoty a z toho důvodu v průběhu testu není pouštěn pomocí kontejneru, ale ve vývojovém prostředí s hodnotami nastavenými pro testování limit. Zároveň je pomocí ručního spouštění možné ověřit detekci odpojení zařízení. Po spuštění serverové části je nutné nastavit analýzu dat v aplikaci Grafana. Nastavené analýzy jsou součástí perzistentních záznamů aplikace Grafana v repozitáři a při dalších spouštěních již není potřeba analýzy konfigurovat. Do aplikace Grafana byly přidány dvě jednoduché analýzy. První analýza provádí detekci překročení hodnoty koncentrace oxidu uhličitého přes hranici 5000 ppm a druhá analýza kontroluje periodický příjem zpráv a generuje upozornění při neaktivitě vyšší než 5 minut. Dále byl v Grafaně nastavena SMTP konfigurace pro účely zasílání emailových upozornění. Pro účely testu byl pozměněn generátor v bráně pro příjem dat, aby se s každou zprávou inkrementovaly hodnoty.

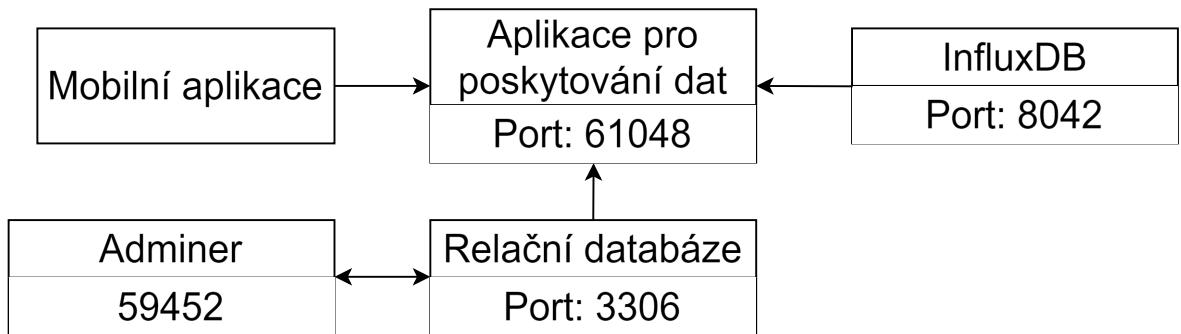
Průběh testu a výhodnocení Serverová část byla spuštěna a po její inicializaci byla ručně spuštěna upravená brána v režimu generátoru. Při každé zprávě z generátoru docházelo ke zvýšení hodnoty koncentrace o hodnotu 10. Po překročení hodnoty 5000 byl obdržen email, s upozorněním. Následně byl vypnuta brána. Po 5 minutách neaktivity byl obdržen další email tentokrát týkající se výpadku zařízení. Vizualizaci dat včetně alarmů z aplikace Grafana je možné vidět na obrázku 6.3. Filtrování hodnot bylo ověřeno zasíláním hodnot větších než než limit nastavený v aplikaci pro nahrávání dat. Při překročení limitních hodnot byla tato informace zalogována a zpráva byla zahozena. Tímto testem bylo ověřeno splnění požadavků 4, 5 a 7. Systém je schopný detektovat překročení limitů a odpojení zařízení. Systém je schopný na tuto situaci upozornit uživatele emailovým upozorněním. Systém je také schopný filtrovat a zahazovat nesmyslné hodnoty.



Obrázek 6.3 Data a upozornění v aplikaci Grafana.

6.1.3 Test vizualizace dat a autorizace uživatele

Modelový příklad pokrývá požadavky 6, 8, 10 a 11. Cílem testu je ověření možnosti autorizace a autentizace uživatele a zobrazení naměřených dat. Zapojení jednotlivých komponent systému je možno vidět na diagramu 6.4. Testovaný systém obsahuje jednu instanci aplikace pro poskytování dat, relační a časové databáze a Adminer.



Obrázek 6.4 Diagram systému pro test zobrazování dat a autorizace uživatele.

Příprava a průběh testu Test je spuštěn pomocí scénáře nástrojem **docker compose**. Soubory nutné ke spuštění, včetně všech konfiguračních souborů jsou umístěny v repozitáři ve složce *testing/test_case_3* a serverová část scénáře se spouští příkazem `docker compose -profile=test_case_3 up`. Veškeré aplikace a databáze jsou inicializovány z konfiguračních souborů a není nutné je nijak upravovat. Po spuštění kontejnerů dojde k postupnému propojení všech částí systému. Přes nástroj **docker compose** není možné spustit mobilní aplikaci pro zobrazování dat. Tuto aplikaci je nutné spustit buď pomocí emulátoru nebo na mobilním zařízení. Spuštění systému jako takového trvá zhruba 2-3 sekundy, po spuštění došlo k přihlášení do systému pomocí mobilní aplikace a kontrola přístupů k jednotlivým zařízením.

Vyhodnocení testu V databázi byli inicializováni dva uživatelé s přihlašovacím jménem *john_doe* a *steve_elan*. Tito uživatelé měli přiřazený různé kombinace společností a zařízení. Přihlášení pomocí zmíněných uživatelských údajů v mobilní aplikaci proběhlo úspěšně a uživateli byla korektně dostupná pouze přiřazená zařízení a bylo možné zobrazovat data z těchto zařízení. Přihlášení nevalidními kombinacemi přihlašovacích údajů nebylo úspěšné. Tento test prokázal splnění požadavku na vizualizaci dat uživatelem 6 a zamezení přístupu k datům neautorizovanému uživateli podle požadavku 8. Dále test potvrdil splnění požadavků 10 a 11, tedy možnost správy uživatelů a mechanismus pro zabránění neoprávněnému přístupu. Neúspěšné pokusy o připojení byly zalogovány. Při běhu testu byl objeven problém v mobilní aplikaci, kdy docházelo k pádu aplikace při načítání uživatele s pouze jedním typem zařízení. Tato chyba byla opravena.

6.1.4 Test nasazení systému

Analýza pokrývá požadavky 9, 12, 13 a 14. Cílem tohoto testu je ověřit jednoduchost nasazení, škálovatelnost, zálohování a logování systému. Požadavky byly částečně ověřeny již v předchozích testech a tento test pouze analyzuje implementovaný způsob nasazení systému.

Jednoduchost nasazení systému je dosažena využitím kontejnerizace jednotlivých komponent. Tuto vlastnost je možné ověřit v repozitáři ve složce *server/deployment* kde se nalézá skript pro nástroj **docker compose**, který umožnuje spuštění celého systému. Tímto je zajištěno splnění požadavku 9. V rámci kontejnerizace jsou záznamy logů a dat aplikací ukládány do perzistentního úložiště a tím je zajištěno splnění požadavků 12 a 13. Splnění požadavku 14 je docílen samotným návrhem systému, který umožňuje podporu více uživatelů a společnosti.

6.2 Testování požadavků na zařízení

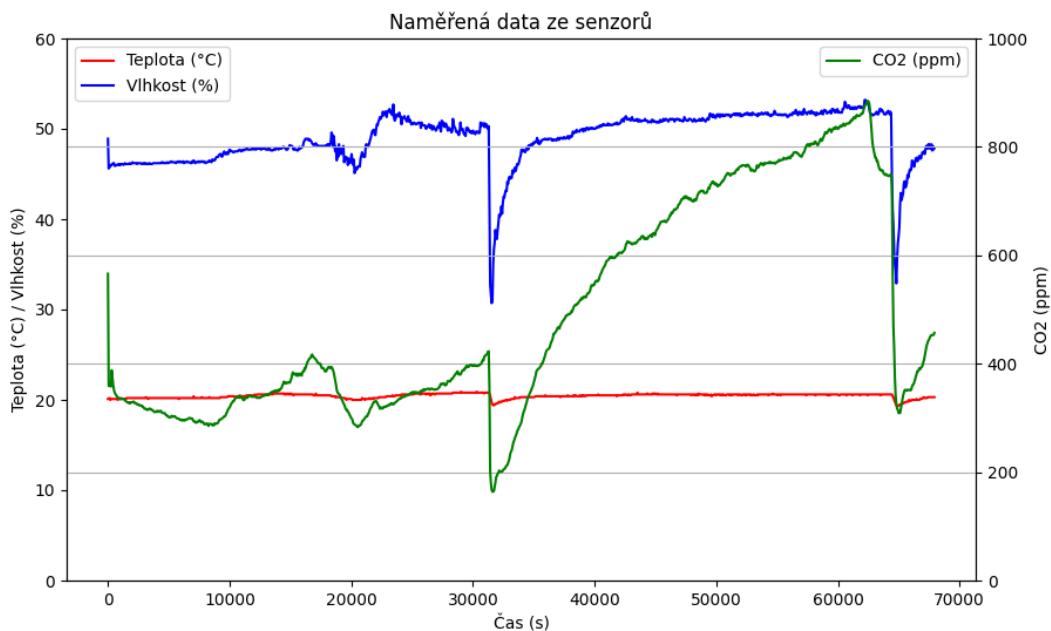
Tato sekce se věnuje testování požadavků na jednotlivá zařízení popsaných v 4.2. Testována je zde pouze základní funkcionality zařízení, dlouhodobé testy jsou prováděny v rámci testů v reálném provozu popsaných v 6.3.

6.2.1 Testování zařízení pro vinaře

Testování zařízení je rozděleno do dvou testů, které pokrývají veškeré požadavky. V prvním testu je ověřena schopnost měření koncentrace oxidu uhličitého, teploty, vlhkosti, dále schopnost upozornit uživatele na vysoké koncentrace oxidu uhličitého a možnost výčtu hodnot přímo ze zařízení, tedy požadavky 2, 3 a 4. Test byl proveden s upravenou branou pro připojení zařízení. Brána hodnoty pouze zapisovala do textového souboru, z důvodu zabránění ovlivnění testu možnými chybami na jiných částech systému. Test měření probíhal v prostorách domu při pokojové teplotě. Při testu bylo zařízení pro měření, zařízení pro přenosilání a brána zapnuty a ponechány několik hodin. Během měření byla manuálně pomocí spínače pravidelně aktivována obrazovka, která zobrazovala aktuální naměřené hodnoty a tím bylo potvrzeno splnění požadavku 3. Pro účely testu byla změněna hodnota koncentrace oxidu uhličitého při které dochází k upozornění uživatele z hodnoty 5000 ppm na 500 ppm. Upozornění bylo dosaženo přítomností osob v uzavřené místnosti. Po překročení hodnoty 500 ppm došlo k rozsvícení červeného led světla. Po vyvětrání místnosti a snížení koncentrace bylo upozornění přerušeno a tím byl ověřen požadavek 4. Po ukončení testu došlo ke kontrole naměřených hodnot. Vytvořené soubory jsou uloženy v repozitáři ve složce *testing/cellar_device_measurements*. Graf naměřených hodnot je možné vidět na obrázku 6.5. Měření probíhalo téměř 20 hodin. Vlhkost se pohybovala v rozmezí 20 až 30 procent, teplota kolem 20 stupňů Cel-

sia a koncentrace oxidu uhličitého mezi 200 až 900 ppm. Na grafu je patrné, že zhruba v půlce měření a ke konci měření došlo k výraznému poklesu hodnot, způsobeného vyvětráním místnosti. Tímto měřením bylo potvrzeno splnění požadavku 2.

Druhý provedený test ověřuje splnění schopnosti zařízení odolat vlhkosti a schopnost vysílat data z podzemních prostor, tedy požadavků 1 a 5. Při provádění testu bylo zařízení pro měření umístěno do sklepních prostor zhruba 3 metry pod zemí. Zařízení pro přenos dat bylo umístěno na povrch. Brána pro příjem dat byla umístěna do obytných prostor ve zhruba 200 metrové vzdálenosti od sklepa. V době průběhu testu nebyla ukládána žádná data, ale byla kontrolována položka čítače ve vstupním protokolu. Zprávy byly generovány každých deset minut. Test probíhal zhruba dvacet hodin. Po dobu průběhu testu nedošlo ke ztrátě žádné zprávy. Pokud by došlo ke ztrátě, výpadek by byl detekován pomocí přeskočení hodnoty čítače. V podzemních prostorách byla naměřena stoprocentní vlhkost vzduchu. Tento test ověřil schopnost zařízení odesílat data z podzemních prostor a odolnost proti vlhkosti. Odolnost proti vlhkosti bude dlouhodoběji testována při reálném nasazení systému.



Obrázek 6.5 Data naměřená zařízením do sklepa.

6.2.2 Testování zařízení pro včelaře

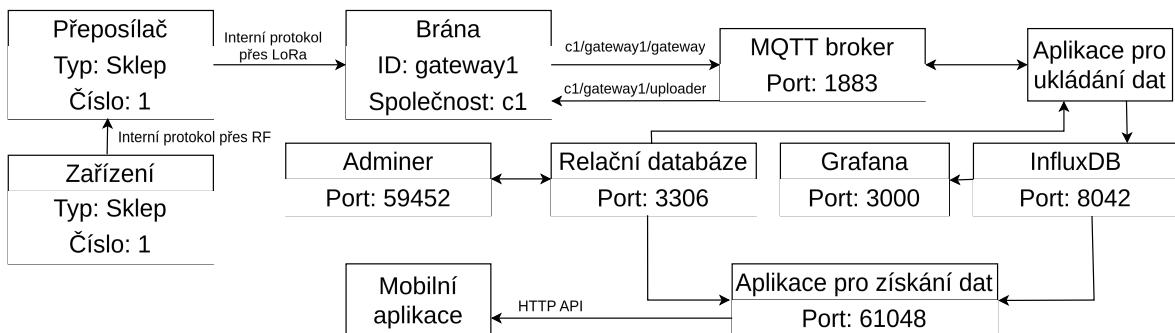
Testování zařízení je opět rozděleno do dvou testů. Cílem prvního testu je ověření schopnosti zařízení měřit korektně váhu, tedy požadavek 2. Test byl proveden vážením dvacetikilového závaží na provizorně vyrobeném držáku tenzometrických senzorů. První testy proběhly neúspěšně z důvodu chybné kalibrace senzorů. Po zkalirovaní obou

tenzometrických senzorů reportovalo zařízení korektní hodnoty.

Druhý test prověřoval splnění požadavků na schopnost operovat na baterie 3, schopnost bezdrátové komunikace na velké vzdálenosti 1 a odolnost proti počasí 4. Test byl proveden umístěním zařízení do volného prostoru s bateriovým napájením. Brána pro připojení zařízení byla umístěna ve vzdálenosti zhruba jednoho kilometru od měřícího zařízení. Test probíhal zhruba dvacet hodin. Zprávy byly pravidelně zasílány každých deset minut. Po dobu běhu testu nedošlo ke ztrátě žádné zprávy. Detekce ztráty zprávy probíhal pomocí kontroly čítače zpráv. Tento test potvrdil splnění požadavků 1, 3 a 4.

6.3 Testování v reálných podmínkách

Testování v reálných podmínkách prověřilo schopnost systému jako celku fungovat bez přerušení po delší časové období v reálných podmínkách a zároveň pokrývá většinu požadavků. Test probíhal v obci Uherčice a měření probíhala pomocí jednoho zařízení umístěného ve vinařském sklepě. Diagram systému je možné vidět na obrázku 6.6.



Obrázek 6.6 Diagram systému pro test v reálných podmínkách.

Systém obsahoval veškeré komponenty vytvořené v práci. Zařízení bylo umístěno zhruba 4 metry pod zemí a pomocí RF vysílače přenáší data do zařízení pro přenos dat umístěného na povrchu. Brána byla umístěna zhruba půl kilometru od zařízení pro přenos dat. Serverová část byla nasazena na serveru mimo oblast měření.

Příprava a průběh testu Při přípravě byla nasazena serverová část systému pomocí nástroje Docker. Po spuštění byla nastavena relační databáze, aby obsahovala dané zařízení a uživatele s přístupem k zařízení. Následně došlo k umístění zařízení pro sběr dat do sklepa a zařízení pro přenos dat na povrch. Poslední částí přípravy byla aktivace brány. Po aktivaci brány započalo nahrávání dat. Celý systém byl aktivován 6. 4. 2024 zhruba ve 12:30 hodin. Systém korektně sbíral data zhruba 8 hodin, kdy došlo k první poruše - serverová část přestala přijímat nová data. Při hledání důvodu poruchy bylo zjištěno, že na straně brány byla chyba při komunikaci s přijímačem LoRa. Po opravení chyby byl systém opět aktivován 8. 4. 2024 zhruba v 17:30 hodin. K dalšímu výpadku

nedošlo až do doby vyhodnocení testu, které proběhlo 13. 4. 2024. Systém tedy byl schopný po opravě fungovat více než 5 dnů bez výpadku.



Obrázek 6.7 Data naměřená zařízením do sklepa

Vyhodnocení testu Test objevil jeden problém v bráně pro příjem dat, popsaný v předchozí sekci, jinak byl test úspěšný. Při vyhodnocování dat byl použit nástroj Grafana. Veškerá data naměřená za celou dobu testu je možné vidět na obrázku 6.7. Data jsou rozdělena do tří grafů. První graf vizualizuje koncentraci oxidu uhličitého, druhý teplotu a třetí graf vlhkost. Na grafech je patrný výpadek dat mezi 6. a 8. dubnem. Data jsou spojitá, v grafu se tedy výpadek projevuje přímým spojením okrajových bodů a nejlépe je vidět na grafu s teplotou. Je vhodné podotknout, že zařízení jako takové vyžaduje určitou dobu pro ustálení hodnot, pokud je například z teplého prostředí přemístěno do chladného. Tato vlastnost je viditelná na počátečním postupném snižování teploty či zvyšování hodnot vlhkosti. Po ustálení jsou hodnoty stabilní a ani jedna z veličin se při měření příliš nelišila. Hodnota teploty se ustálila na $9,5^{\circ}\text{C}$, vlhkosti na 97 % a koncentrace oxidu uhličitého na 160 ppm.



Obrázek 6.8 Událost naměřená zařízením do sklepa.

Ke změně hodnot došlo 12. 4. po osmé hodině večerní. Tato změna je vyobrazena na grafech 6.8. Horní graf obsahuje hodnoty koncentrace oxidu uhličitého, prostřední graf teplotu a spodní graf vlhkosti. Po osmé hodině večerní došlo k nárůstu teploty ze stabilních $9,5^{\circ}\text{C}$ na 11°C , koncentrace oxidu uhličitého z 160 ppm až na 260 ppm a snížení vlhkosti z 97% na 94% . Po deváté hodině večer se hodnoty vrátili zpět na původní. Tato změna byla způsobena hodinovou degustací vína za přítomnosti 12 osob. Díky této události je jasně patrný vliv přítomnosti většího množství lidí na prostředí ve vinném sklepě. Zároveň lze prohlásit, že tato změna nemá dlouhodobý vliv a prostředí se vrátí velmi rychle k původním hodnotám.

Během testu bylo k naměřeným datům pravidelně přistupováno přes mobilní aplikaci. Na serverové části ani na samotném zařízení pro vinaře během testu nedošlo k jediné chybě. Data naměřená při testu ve formě zálohy databáze InfluxDB jsou v repositáři uloženy ve složce *testing/test_case_real*.

ZÁVĚR

Cílem této práce bylo navrhnut a implementovat systém pro sběr, analýzu a vizualizaci dat v zemědělství. Výsledný systém popsaný v 3 obsahuje serverovou část, a uživatelskou část. Serverová část se skládá z aplikace pro příjem dat, časové a relační databáze, aplikace pro poskytování dat a nástrojů pro správu databází Grafana a Adminer. Uživatelská část se skládá ze zařízení pro sběr dat, brány pro připojení zařízení a mobilní aplikace pro vizualizaci dat, protokolů a rozhraní dostupných uživateli pro případnou integraci s jinými systémy.

Serverová část byla navržena tak, aby umožňovala připojení desítek zařízení najednou, a to i různých uživatelů, a její nasazení bylo jednoduché a rychlé. Jednoduchosti a rychlosti nasazení bylo dosaženo kontejnerizací jednotlivých komponent. Serverová část je implementovaná v jazyce Python a je kontejnerizována nástrojem Docker. Pro správu metadat byla zvolena relační databáze MariaDB a pro ukládání měření byla zvolena časová databáze InfluxDB. Pro analýzu dat a notifikaci uživatelů byl zvolen již existující nástroj Grafana.

Uživatelská část se skládá ze zařízení pro sběr dat, která komunikují s bránou pomocí protokolu navrženém v 3.4.1. Protokol je navržen a zdokumentován tak, aby bylo možné přidávat různá nová zařízení poskytující data a tím pádem přidávání podpory nových zařízení není vázáno na jednu osobu, ale na komunitu využívající komponenty vzniklé v rámci této práce. Na stejném principu je navržena brána pro připojení zařízení. Protokol používaný pro nahrávání dat do serverové části není pevnou součástí systému a může být nahrazen pro integraci do jiných systémů. Tím pádem zařízení i bránu vytvořené v rámci této práce lze integrovat i do již existujících systémů. Dalšími komponenty uživatelské části je mobilní aplikace a rozhraní pro získání dat. Mobilní aplikace vytvořená v rámci této práce umožňuje uživateli po přihlášení zobrazení grafů s naměřenými hodnotami pro jednotlivá zařízení. Rozhraní pro získání dat popsané v 3.8 umožňuje integraci systému s jinými vizualizačními nástroji.

Při implementaci práce vznikla dvě zařízení jakožto příklad využití. Zařízení vznikla na popud samotných zemědělců. První zařízení je určeno do vinného sklepa určeného pro monitorování teploty, vlhkosti a koncentrace oxidu uhličitého. Druhé zařízení je určeno pro monitorování hmotnosti včelích úlů za účelem odhadu doby stáčení medu a dokrmování včelstev. Systém se neomezuje pouze na tyto dva příklady využití a je možné ho rozšiřovat o jiná zařízení.

Výsledek práce jako takový si lze snadno vyzkoušet pomocí kontejnerizace. Zároveň není nutné vlastnit fyzické zařízení osazené senzory díky schopnosti brány vytvářet virtuální zařízení a tím simulovat sběr dat. Spuštěním jediného programu se spustí veškeré části systému a k této instanci je následně se uživatel již jen připojí mobilní

aplikací a celý systém si během pár minut vyzkoušet bez nutnosti jakékoliv investice. Jednotlivé celky jsou snadno integrovatelné do jiných již existujících systémů a zároveň je celek možné integrovat pomocí přesně definovaného rozhraní do existujících řešení.

Škálovatelnost je dosažena na straně uživatelské části možností zvolit si s jakou instancí serverové částí bude komunikovat, tím pádem různí zemědělci nemusí sdílet stejné serverové řešení. Při případném sdílení serverových zdrojů je možné nasadit více instancí aplikací pro přístup k datům a distribuovat časovou databázi na více strojů. V rámci návrhu byly určeny požadavky na systém 3.2 a na obě zařízení 4.2. Splnění těchto požadavků bylo ověřeno pomocí testování. Finální test celého nasazeného systému byl vyhodnocen po týdnu fungování a po vyhodnocení byl systém ponechán běžet v reálném provozu. Už samotné testy prokázaly přínos pro vinaře, kdy byla detekována změna podmínek ve vinném sklepě při přítomnosti většího množství lidí, což může mít neblahý vliv na uskladněné víno. Vinař může být i upozorněn emailem při překročení předdefinovaných hodnot. Podobných využití může být desítky. Podobný přínos přinese nasazení zařízení pro monitorování váhy vcelích úlů plánované na začátek léta. Přínos potvrzuje i zájem o vyzkoušení výsledku této práce nejen jednotlivců, ale i spolků.

SEZNAM POUŽITÉ LITERATURY

- [1] Osinga, S. A.; Paudel, D.; Mouzakitis, S. A.; aj.: Big data in agriculture: Between opportunity and solution. *Agricultural Systems*, ročník 195, 2022: str. 103298.
- [2] Jeedigunta, S.; Goel, A.: Here's how the agricultural sector can solve its data problem. [online]. <https://www.weforum.org/agenda/2023/01/here-s-how-agricultural-sector-data-problem-davos2023/>, 2023, [cit. 2023-12-02].
- [3] Osinga, S. A.; Paudel, D.; Mouzakitis, S. A.; aj.: *Big data in agriculture: Between opportunity and solution*. Agricultural Systems, páté vydání, 2016, ISBN 978-0128053942.
- [4] Shibusawa, S.; Haché, C.: Data collection and analysis methods for data from field experiments. *System Analysis and Modeling in Food Agriculture*, ročník 12, 2009: str. 311.
- [5] Yang, C.; Everitt, J. H.; Du, Q.; aj.: Using High-Resolution Airborne and Satellite Imagery to Assess Crop Growth and Yield Variability for Precision Agriculture. *Proceedings of the IEEE*, ročník 101, č. 3, 2013: s. 582–592, doi:10.1109/JPROC.2012.2196249.
- [6] Li, S.; Simonian, A.; Chin, B. A.: Sensors for agriculture and the food industry. *The Electrochemical Society Interface*, ročník 19, č. 4, 2010: str. 41.
- [7] SwiftSensors: Agriculture & Farming Monitoring System. [online]. <https://www.swiftsensors.com/industry/agriculture-monitoring-system/>, 2023, [cit. 2023-12-03].
- [8] Teja, R.: What is a Sensor? Different Types of Sensors and their Applications. [online]. <https://www.electronicshub.org/different-types-sensors>, 2021, [cit. 2023-12-09].
- [9] Xu, J.; Gu, B.; Tian, G.: Review of agricultural IoT technology. *Artificial Intelligence in Agriculture*, ročník 6, 2022: s. 10–22.
- [10] Planet: Data Driven Precision Agriculture with Planet. [online]. <https://www.planet.com/markets/monitoring-for-precision-agriculture/>, 2023, [cit. 2023-12-03].
- [11] BeeHero: Beekeepers. [online]. <https://www.beehero.io/beekeepers>, 2023, [cit. 2023-12-03].

- [12] JohnDeere: Precision AG Technology. [online]. <https://www.deere.com/en/technology-products/precision-ag-technology/>, 2023, [cit. 2023-12-03].
- [13] Buyya, R.; Dastjerdi, A. V.: *Internet of Things: Principles and paradigms*. Elsevier, 2016.
- [14] Lavric, A.; Petrarau, A. I.: LoRaWAN communication protocol: The new era of IoT. In *2018 International Conference on Development and Application Systems (DAS)*, IEEE, 2018, s. 74–77.
- [15] Devalal, S.; Karthikeyan, A.: LoRa technology-an overview. In *2018 second international conference on electronics, communication and aerospace technology (ICECA)*, IEEE, 2018, s. 284–290.
- [16] LoRaAllianceTechnicalCommittee: LoRaWAN® Regional Parameters [online]. https://lora-alliance.org/wp-content/uploads/2020/11/RP_2-1.0.2.pdf, 2020, [cit. 2023-12-17].
- [17] rfwireless world: LoRaWAN Airtime calculator, LoRa Packet duration calculations. [online]. <https://www.rfwireless-world.com/calculators/LoRaWAN-Airtime-calculator.html>, 2020, [cit. 2023-12-17].
- [18] Microchip: Introduction to Bluetooth® Low Energy. [online]. <https://microchipdeveloper.com/xwiki/bin/view/applications/ble/>, 2023, [cit. 2023-12-29].
- [19] StandardOASIS: MQTT Version 5.0. ročník 22, 2019: str. 2020.
- [20] Çorak, B. H.; Okay, F. Y.; Güzel, M.; aj.: Comparative analysis of IoT communication protocols. In *2018 International symposium on networks, computers and communications (ISNCC)*, IEEE, 2018, s. 1–6.
- [21] TechTarget: What is Narrowband IoT (NB-IoT)? [online]. <https://www.techtarget.com/whatis/definition/narrowband-IoT-NB-IoT>, 2021, [cit. 2024-01-13].
- [22] Michal Lom, O. P.: Sítě pro internet věcí v České republice. [online]. <https://elektro.tzb-info.cz/informacni-a-telekomunikacni-technologie/16519-site-pro-internet-veci-v-ceske-republice>, 2017, [cit. 2024-01-13].
- [23] Buyya, R.; Calheiros, R. N.; Dastjerdi, A. V.: *Big data: principles and paradigms*. Morgan Kaufmann, 2016.

- [24] Rahm, E.; Do, H. H.; aj.: Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, ročník 23, č. 4, 2000: s. 3–13.
- [25] Tripathi, A.: A Beginners Guide to Data Analytics in IoT – Steps to Implement IoT Analytics and More [online]. <https://www.analytixlabs.co.in/blog/analytics-in-iot/>, 2023, [cit. 2024-03-09].
- [26] Ted, D.; Ellen, F.: Time Series Databases: New Ways to Store and Access Data. URL: https://www.academia.edu/29891282/Time_Series_Databases_New_Ways_to_Store_and_Access_Data.
- [27] Naqvi, S. N. Z.; Yfantidou, S.; Zimányi, E.: Time series databases and influxdb. *Studienarbeit, Université Libre de Bruxelles*, ročník 12, 2017.
- [28] OneSoil: Agricultural OneSoil Map with AI detected fields and crops. [online]. <https://map.onesoil.ai/>, 2024, [cit. 2024-06-01].
- [29] GrafanaLabs: Grafana documentation. [online]. <https://grafana.com/docs/>, 2024, [cit. 2024-01-07].
- [30] Onugha, D.: Comparing Cloud Hosting vs. Self Hosting. [online]. <https://medium.com/@danielonugha0/comparing-cloud-hosting-vs-self-hosting-5e099dd6c0de>, 2023, [cit. 2024-01-14].
- [31] IBM: What is containerization?. [online]. <https://www.ibm.com/topics/containerization>, 2023, [cit. 2024-01-14].

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

GPS	Global Positioning System
GIS	Geographic Information System
VNA	Vector Network Analyser
RFID	Radio Frequency Identification
IoT	Internet of Things
IOS/OSI	International Organization for Standardization/Open Systems Interconnection
LPWAN	Low Power Wide Area Network
LoRa	Long Range
STK	Short Term Key
LTK	Long Term Key
BLE	Bluetooth Low Energy
TLS	Transport Layer Security
AES	Advanced Encryption Standard
QoS	Quality of Service
JSON	JavaScript Object Notation
CSV	Comma Separated Value
AGPL	Affero General Public License
NB IoT	Narrow Band Internet of Things
GSM	Global System for Mobile communication
LTE-M	Long-Term Evolution for Machines
SSL	Secure Socket Layers
MQTT	Message Queuing Telemetry Transport
LCD	Liquid Crystal Display
PCB	Printed Circuit Board
SMTP	Simple Mail Transfer Protocol

SEZNAM OBRÁZKŮ

Obr. 1.1.	Infračervené snímky pole při prvním růstu (a) a při zrání (b) s odhadem výnosu získaným regresní rovnicí z původních snímků (c, d) [5].	14
Obr. 1.2.	Příklad vnějšího pouzdra senzoru používaného v zemědělství [7].	15
Obr. 1.3.	BeeHero senzor umístěný v úlu [11].....	19
Obr. 2.1.	Vrstvy IoT systémy v zemědělství [9].	22
Obr. 2.2.	Příklad jednoduchého systému s různými komunikačními protokoly....	23
Obr. 2.3.	Architektura protokolu BLE [18].....	28
Obr. 2.4.	Příklad jednoduchého zapojení zařízení do sítě MQTT.	29
Obr. 2.5.	Porovnání LPWAN protokolů s 4G [21].	31
Obr. 2.6.	Interaktivní mapa zemědělských plodin [28].	38
Obr. 2.7.	Dostupné zdroje dat v základní instanci Grafany.	39
Obr. 2.8.	Příklad nasazení systému přes službu Docker.....	44
Obr. 3.1.	Diagram případu užití platformy.....	47
Obr. 3.2.	Diagram komponent a jejich vztahu k zařízení.....	49
Obr. 3.3.	Diagram komponent brány pro zařízení.	50
Obr. 3.4.	Stavový diagram brány pro zařízení.....	51
Obr. 3.5.	Formát zprávy vstupního protokolu.....	52
Obr. 3.6.	Sekvenční diagram výstupního protokolu.	56
Obr. 3.7.	Diagram komponent aplikace pro ukládání a filtrování dat.	58
Obr. 3.8.	Stavový diagram aplikace pro ukládání a filtrování dat.	59
Obr. 3.9.	Databázové schéma relační databáze.	59
Obr. 3.10.	Schéma a příklad časové databáze vůči struktuře.	60
Obr. 3.11.	Návrh rozložení elementů pro přihlášení a volbu parametrů v mobilní aplikaci.	63
Obr. 3.12.	Návrh rozložení elementů pro okno zobrazení grafu.....	63
Obr. 4.1.	Rozložení datové části a příznakových znaků vstupního protokolu zařízení pro vinaře.....	67
Obr. 4.2.	Komponenty zařízení pro vinaře a jejich komunikace.	68
Obr. 4.3.	Rozložení datové části a příznakových znaků vstupního protokolu zařízení pro včelaře.....	69
Obr. 4.4.	Komponenty zařízení pro včelaře a jejich komunikace.	70
Obr. 5.1.	Obrazovky mobilní aplikace a přechody mezi nimi.	77
Obr. 5.2.	Zařízení pro přeposílání dat od zařízení pro vinaře.	78
Obr. 5.3.	Měřící zařízení pro vinaře.....	79
Obr. 5.4.	Zapojení měřícího zařízení pro vinaře.	80

Obr. 5.5.	Zapojení měřícího zařízení pro včelaře.	81
Obr. 5.6.	Platforma pro vážení úlu.....	82
Obr. 6.1.	Diagram systému pro test periodického měření a ukládání dat.	83
Obr. 6.2.	Diagram systému pro test notifikací.	84
Obr. 6.3.	Data a upozornění v aplikaci Grafana.	85
Obr. 6.4.	Diagram systému pro test zobrazování dat a autorizace uživatele.	86
Obr. 6.5.	Data naměřená zařízením do sklepa.	88
Obr. 6.6.	Diagram systému pro test v reálných podmínkách.	89
Obr. 6.7.	Data naměřená zařízením do sklepa	90
Obr. 6.8.	Událost naměřená zařízením do sklepa.....	91

SEZNAM TABULEK

Tab. 2.1.	Protokoly používané v IoT na modelu ISO/OSI [13].....	23
Tab. 2.2.	Frekvenční pásma LoRa s povolenými pracovními cykly [16].	25
Tab. 2.3.	Příklad nevyčištěného datasetu.....	33
Tab. 2.4.	Příklad datasetu po vyčištění.	34
Tab. 3.1.	Příklad dvou záznamů ve výstupním souboru.	54

SEZNAM PŘÍLOH

- P I. Konfigurační soubory
- P II. Zapojení měřících zařízení

PŘÍLOHA P I. KONFIGURAČNÍ SOUBORY

Konfigurační soubor pro bránu:

```
{  
    "generic_settings": {  
        "verbose": true,  
        "log_path": "./log",  
        "gateway_id": "gateway1",  
        "company": "company"  
    },  
    "device_settings": {  
        "supported_devices": [  
            {  
                "device_type": 1,  
                "device_number": 1  
            },  
            {  
                "device_type": 2,  
                "device_number": 1  
            }  
        ],  
        "device_communication_type": "LORA",  
        "generator_settings": {  
            "device_type": 1,  
            "device_number": 1  
        },  
        "lora_settings": {  
            "uart_device_path": "/dev/ttyS0",  
            "lora_channel": 2,  
            "m0_pin": 4,  
            "m1_pin": 5,  
            "uart_baudrate": 9600,  
            "lora_address": 5156  
        }  
    },  
    "output_settings": {  
        "output_type": "MQTT",  
        "csv_path": "/home/nothrax/data",  
        "mqtt_settings": {  
            "hostname": "localhost",  
            "port": 1883,  
            "username": "test1",  
            "password": "test2",  
            "ssl": false,  
            "ca_file": "",  
            "client_cert": "",  
            "client_key": ""  
        }  
    }  
}
```

Konfigurační soubor pro aplikaci pro nahrávání dat:

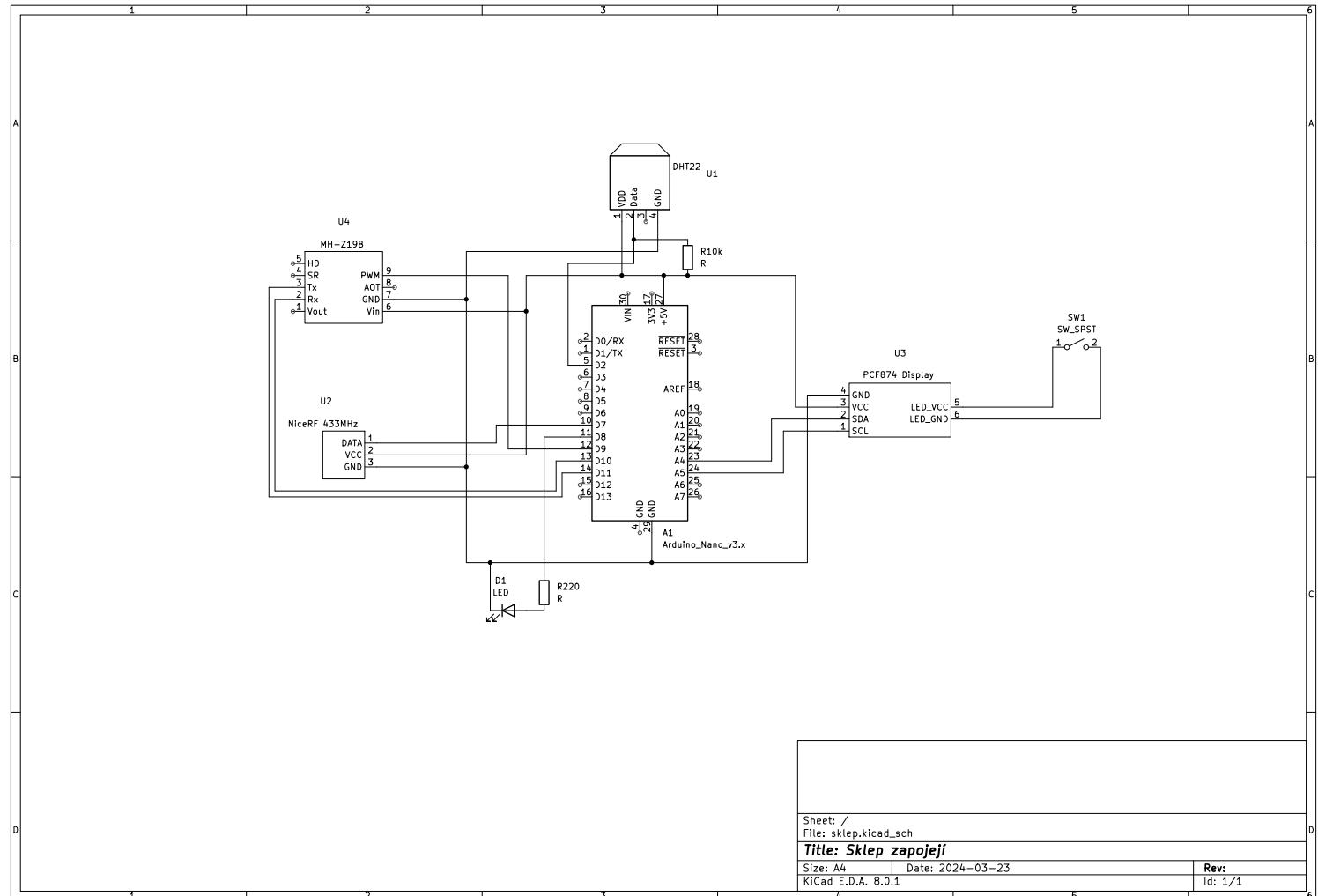
```
{  
    "mqtt_settings": {  
        "address": "127.0.0.1",  
        "port": 1884,  
        "ssl": false,  
        "ca_cert": "./ca.pem",  
        "cert_file": "./cert.pem",  
        "keyfile": "key.pem"  
    },  
    "timeseries_settings": {  
        "address": "127.0.0.1:8042",  
        "token": "randomTokenValue"  
    },  
    "relation_settings": {  
        "address": "127.0.0.1",  
        "port": 8042  
    }  
}
```

```
        "user": "root",
        "password": "example"
    }
```

Konfigurační soubor pro aplikaci pro získávání dat:

```
{
    "flask_settings": {
        "host": "127.0.0.1",
        "port": 5000
    },
    "timeseries_settings": {
        "address": "127.0.0.1:8042",
        "token": "randomTokenValue"
    },
    "relation_settings": {
        "address": "127.0.0.1",
        "user": "root",
        "password": "example"
    }
}
```

PŘÍLOHA P II. ZAPOJENÍ MĚŘICÍCH ZAŘÍZENÍ



Sheet: /
File: sklep.kicad_sch

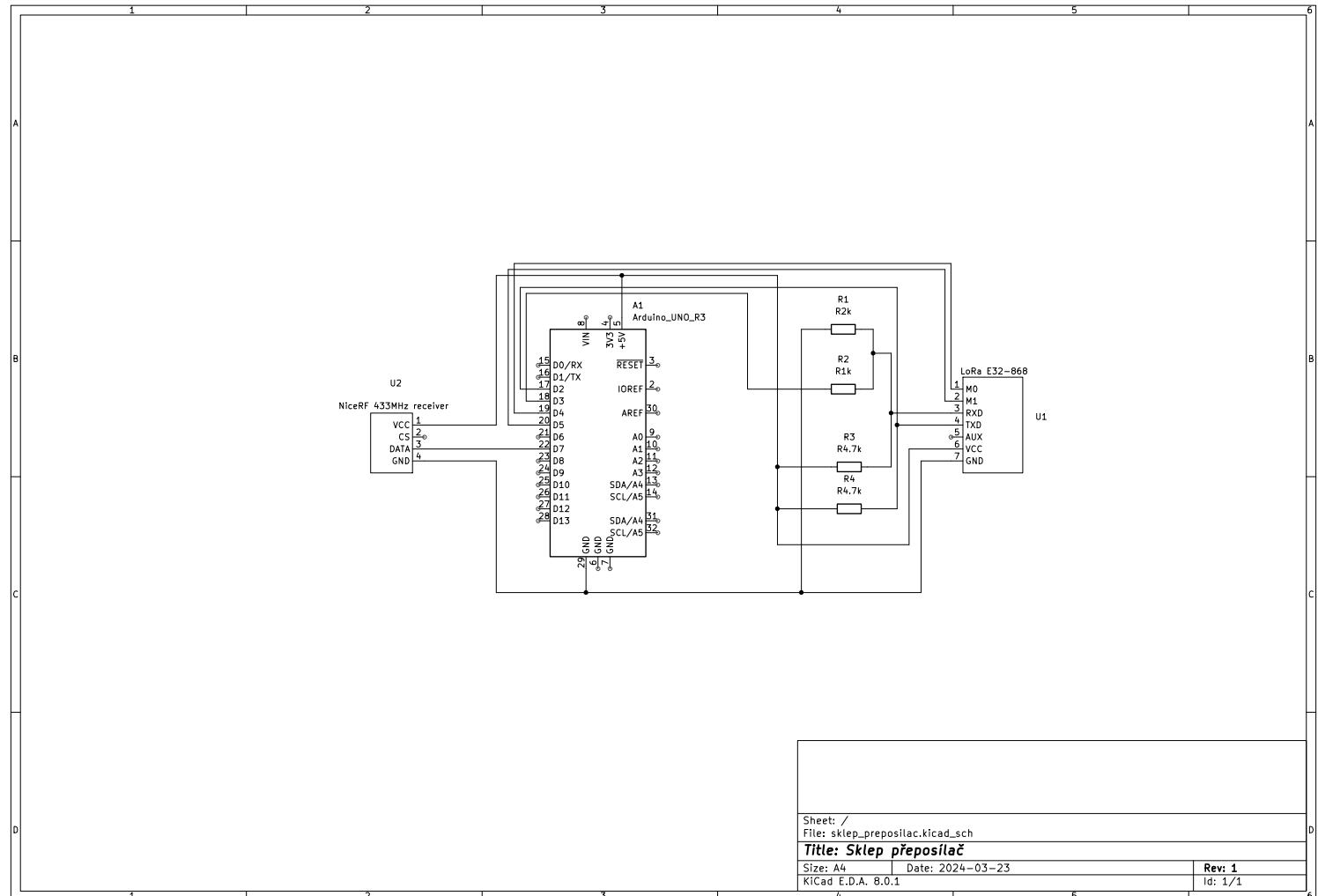
Title: Sklep zapojení

Size: A4 Date: 2024-03-23

KiCad E.D.A. 8.0.1

Rev:

Id: 1/1



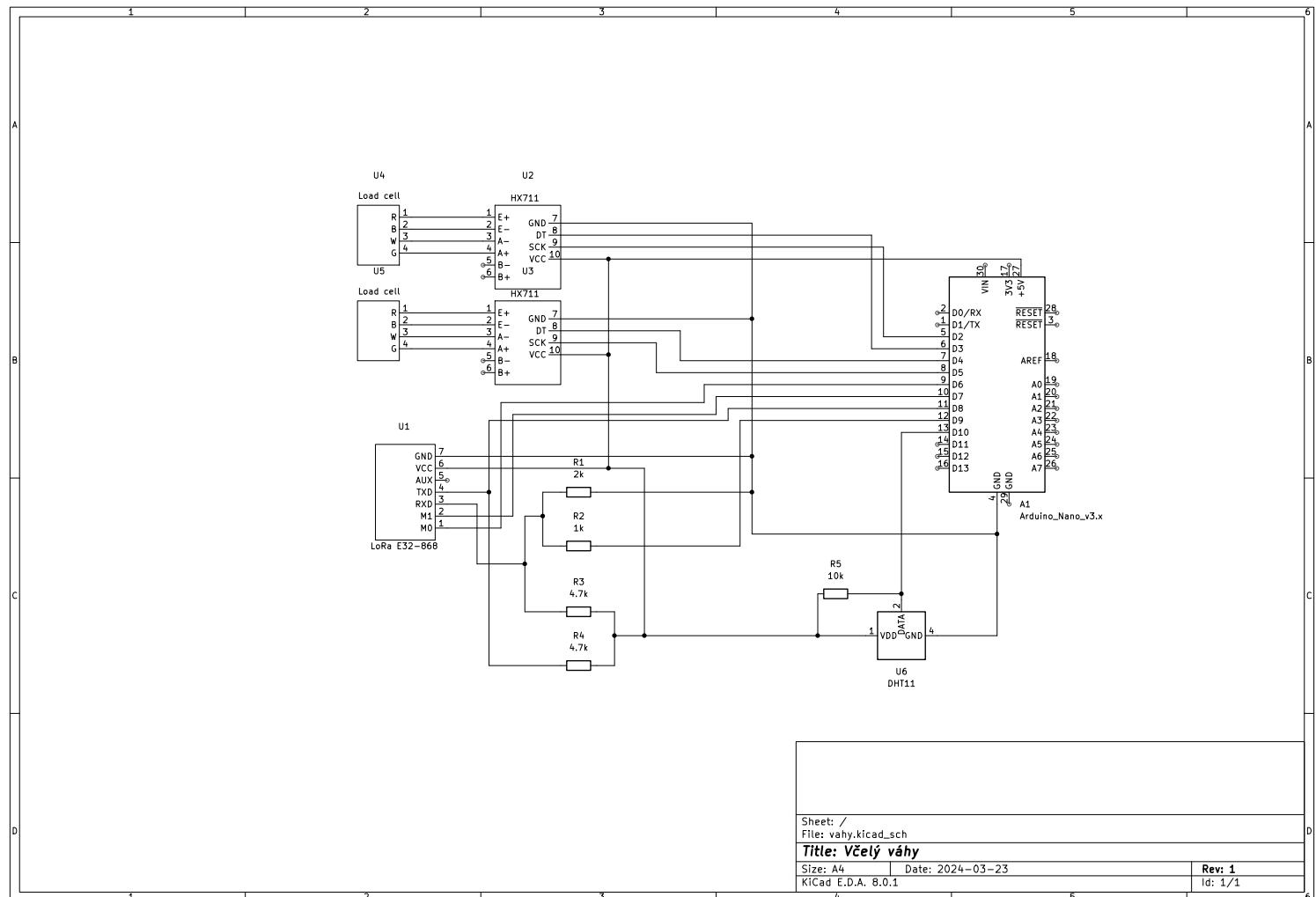
Sheet: /
File: sklep_přepořílač.kicad_sch

Title: Sklep přepořílač

Size: A4 Date: 2024-03-23

KiCad E.D.A. 8.0.1

Rev: 1
Id: 1/1



Sheet: / File: vahy.kicad_sch

Title: Včelý váhy

Size: A4 Date: 2024-03-23

Rev: 1

KiCad E.D.A. 8.0.1

Id: 1/1