

Úvod

Tento dokument pojednává o architektuře navrhovaného řešení systému pro sledování stavu a rezervací parkovacích míst.

Dokument je rozdělen do následujících kapitol:

V kapitole **přehled architektury** jsou popsány moduly systému a jejich role.

Popis různých pohledů na architekturu systému v kapitole **různé pohledy na architekturu**. Tato kapitola pak obsahuje podkapitoly, které nahlíží na architekturu řešení z různých úhlů. (logický, procesní, deployment a implementační pohled na architekturu).

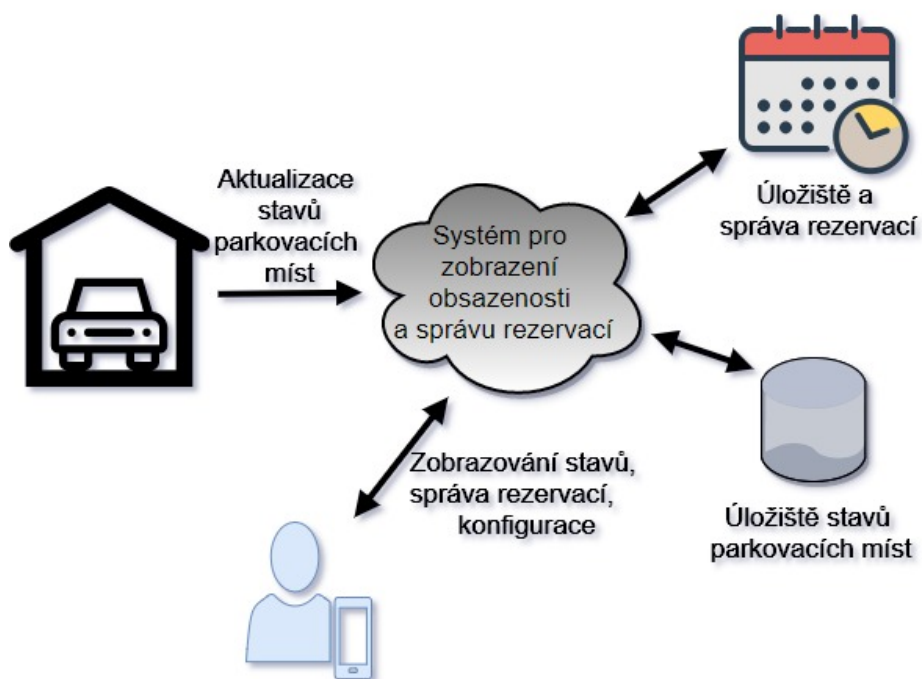
Poslední dvě kapitoly jsou věnovány velikosti, výkonu systému a jeho kvalitě.

Přehled Požadovaných funkcionalit

Z pohledu skupin funkcionalit při vyšší míře abstrakce vypadá přehled řešení následovně

1. parkovací místa budou hlásit svůj stav do systému (obsazenost, stav baterií, chybové stavy). Půjde o jednostrannou komunikaci.
2. uživatel bude mít možnost na dálku zjistit stav parkovacích míst,
3. uživatel bude mít možnost provést rezervaci parkovacího místa,
4. Stavy a rezervace parkovacích míst budou ukládány do sdíleného úložiště (aby data byla přístupná všem účastníkům)

Graficky jsou tyto závislosti znázorněny na následujícím diagramu.



Přehled Architektury/Architecture overview

Big Picture

Následující diagram zachycuje "Big picture" návrhu řešení. Z pohledu uživatele se bude jednat především o klientskou vrstvu, která obsahuje aplikaci Android, ve které uvidí stavy parkovacích míst a bude mít možnost spravovat rezervace. Ve webovém browseru

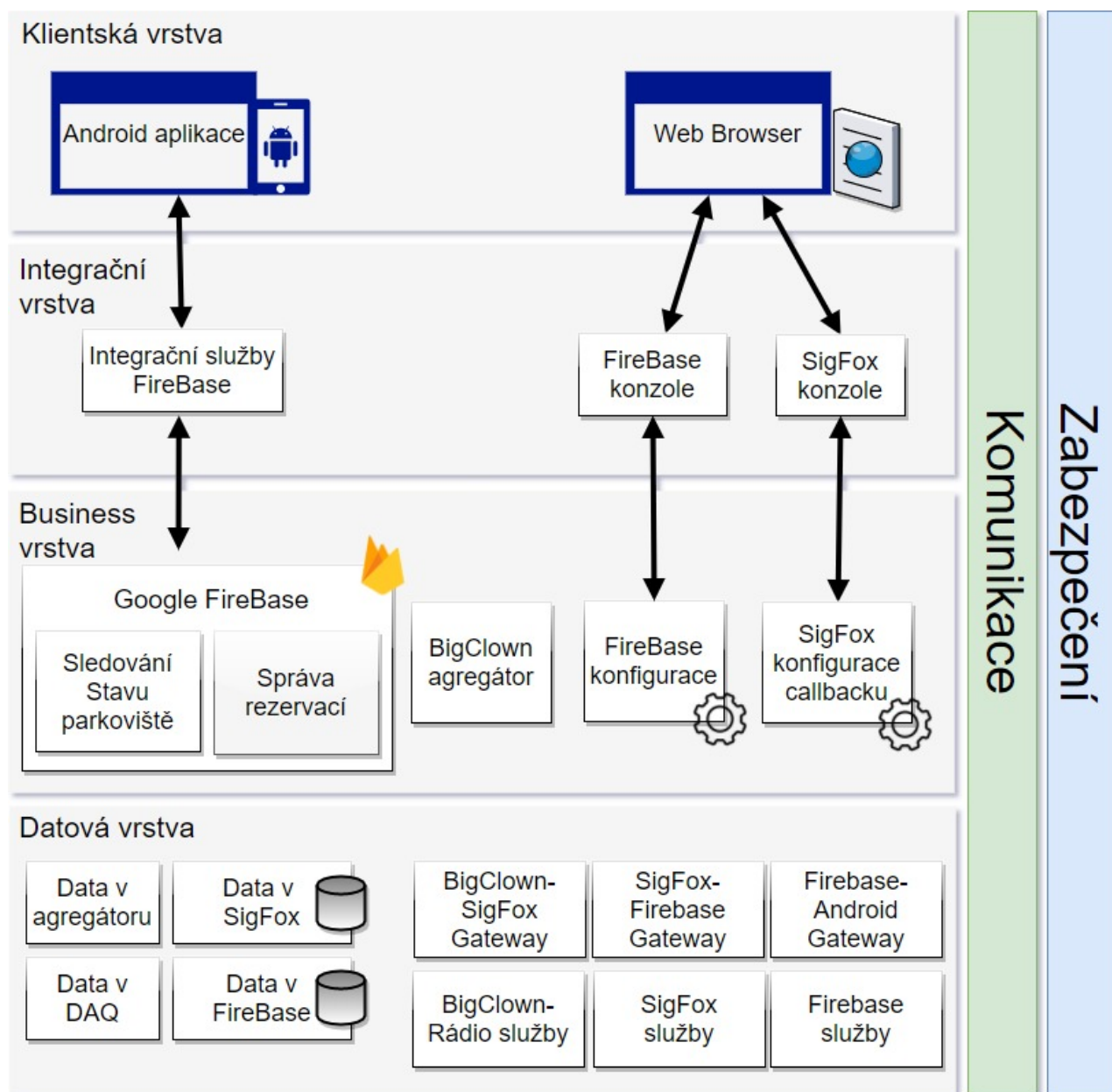
rovněž bude možné provádět konfigurace SigFox a GoogleFirebase nastavení. (např. adresu pro přeposílání zpráv, přiřazení parkovacích míst na pozice v synchronizační zprávě, logika zpracování příchozích zpráv, reporting, uživatelská práva a podobně)

Integrační vrstva obsahuje služby pro konektivitu mezi klientskou a business vrstvou. Jsou zde služby FireBase a konzole pro nastavení FireBase a SigFox.

Business vrstva obsahuje samotnou logiku pro sledování stavu parkovacích míst a správu rezervací. Určitá část logiky probíhá i v agregátoru (detekce nefunkčních DAQ nodů, výroba synchronizačních zpráv, optimalizace zaslání zpráv - aby nedošlo k přečerpání povoleného limitu a následné blokaci služby)

V datové vrstvě jsou především služby pro přístup k datům, pro komunikaci a samotné datové úložiště. Data o obsazenosti budou na DAQ nodech, v agregátoru, v SigFox databázi a FireBase databázi. Informace o rezervacích budou pouze ve FireBase databázi.

Všemi úrovněmi prochází vertikální komunikační vrstva a vrstva zabezpečení, která reprezentuje skutečnost, že spolu prvky komunikují a že je komunikace zabezpečená.



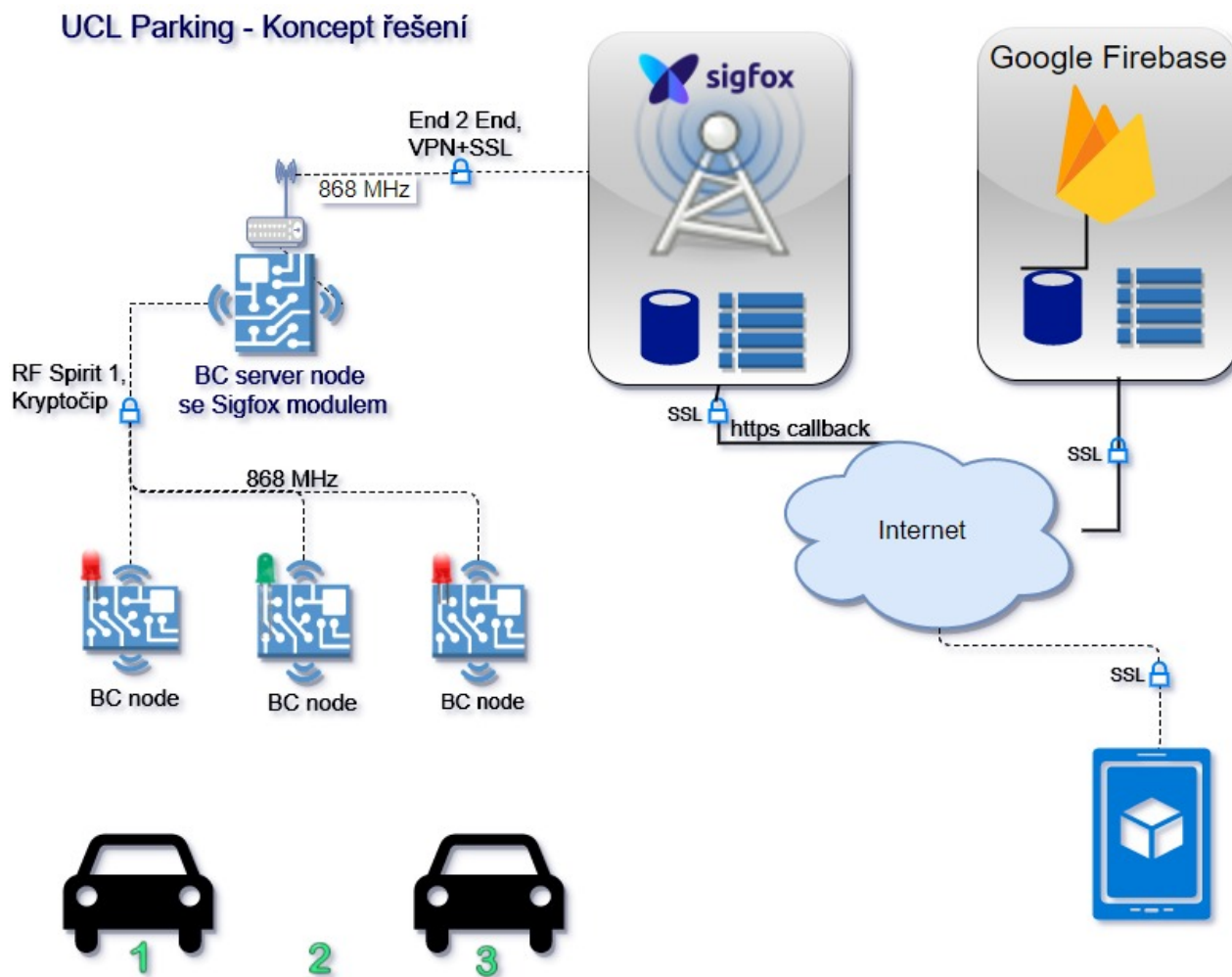
Koncept řešení

Základní koncept řešení je znázorněn na následujícím diagramu. jednotlivá čidla (DAQ uzly) zjišťují stavy parkovacích míst a zasílají je do agregačního prvku. Ten pak pomocí SigFox sítě data zprostředkuje do vzdáleného úložiště Google Firebase. Zde budou data

uchovávána a bude prováděna nastavená logika. Zde budou rovněž uloženy a zpracovávány rezervace. Klientská zařízení budou zobrazovat aktuální stavy parkovacích míst, a budou umožňovat správu rezervací.

Data budou po cestě šifrována pomocí kryptočipu a následně budou přenášena SigFox VPN sítí a dále bude komunikace zabezpečena šifrováním SSL.

Následující diagram zachycuje koncept řešení.



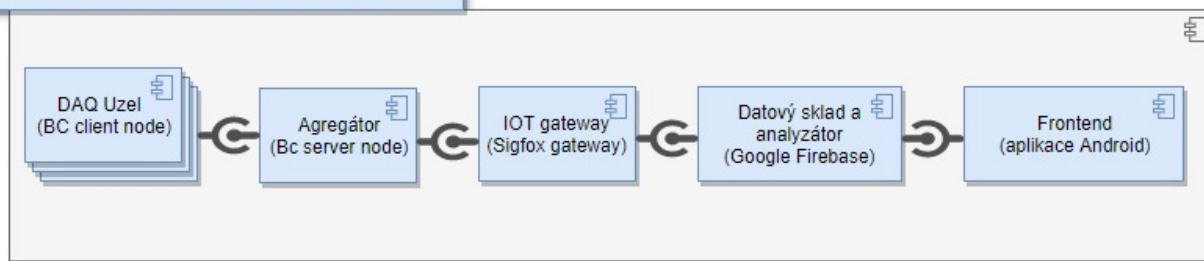
Přehled komponent řešení

Celé řešení je založeno na několika různých komponentech, které budou na různých místech a budou používat jiné technologie. Tyto komponenty spolu navzájem budou komunikovat.

V této kapitole jsou představeny jednotlivé komponenty, a jsou vysvětleny jejich role a vzájemné propojení.

Každá komponenta komunikuje přímo pouze se sousedními komponentami, jak je znázorněno na následujícím diagramu.

Funkční bloky řešení UCL parkoviště



Komponenty jsou blíže popsány v následující tabulce

Entita/Entity	Popis/Description
Komponenta „DAQ Uzel“	Sbírá informace z parkovacího místa a přeposílá je do „Agregátoru“. Tato komponent může mít více instancí, kde pro každé parkovací místo bude jedno takové zařízení. Pro účely této práce je komponenta založena na stavebnici BigClown. Každý takový prvek má na sobě připojeno čidlo pro měření obsazenosti parkovacího místa, čidlo pro zjišťování množství světla a tlačítko pro kalibraci. Kvůli úspoře baterií probíhá sbírání a odesílání dat vždy pouze za určený časový úsek. Tento časový úsek je kratší v případě, kdy jsou v garážích rozsvícená světla. Předpokládá se totiž, že se stav parkoviště bude měnit hlavně tehdy, pokud bude v garážích rozsvíceno.
Komponenta „Agregátor“	Sbírá data z jednotlivých „DAQ uzlů“. Dále data agreguje do jedné zprávy, kterou posílá přes IOT síť do cílového úložiště. V návrhu této realizace je agregace a komprimace dat do jedné zprávy je především z důvodu omezení velikosti Sigfox zpráv (12b) a jejich povolené množství za den (144ks). Tato komponenta existuje především z důvodu finanční úspory na celkových nákladech. V klasickém pojetí IOT by každé parkovací místo mělo svoje vlastní IOT gateway zařízení, tudíž by byly vyšší pořizovací náklady, ale i náklady za připojení těchto zařízení.
Komponenta „IOT gateway“	Je připojena přímo ke komponentě „Agregátor“ a umožňuje odesílání stavu připojených parkovacích míst na další zpracování a do datového úložiště. Při obdržení zprávy je vypublikovaná callback transakce pro nastavené posluchače, kterými mohou běžet na vlastních serverech, nebo to mohou být služby běžící v cloudu (jako jsou například Amazon AWS, Microsoft Azure, Google Firebase a podobně). V tomto řešení se zpráva bude přeposílat do úložiště Google Firebase.
Komponenta „Datový sklad“	Je úložiště dat. Pro tento projekt byl vybrán Google Firebase především proto, že je zdarma, podporuje šifrování informací, umožňuje relativně jednoduchou výrobu aplikací pro IOS i Android. Zároveň pro tuto platformu existuje relativně rozsáhlá dokumentace a mnoho video návodů.
Komponenta „Analyzátor“	načítá data z datového skladu a provádí s nimi potřebné analytické operace. Google Firebase má vestavěnou podporu pro Javascript. Nad každou získanou zprávou vždy provede rozpad na stavy jednotlivých parkovacích míst a jejich uložení do databáze.
Komponenta „Frontend“	je komponenta, která má na starosti interpretaci současných dat. V této realizaci byla zvolena cílová platforma Android a její nativní podpora Google Firebase.

Různé pohledy na architekturu

Tato kapitola obsahuje různé pohledy na architekturu navrhovaného řešení.

- **Logický pohled** - popis funkčních bloků rozdělený podle funkcionalit a podle propojení.
- **Procesní pohled** - popis procesů, které architektura pokrývá, jejich architektonicky důležitých prvků a vazeb mezi nimi.
- **Pohled nasazení** - popis hardwarové konfigurace a jednotlivých uzlů na kterých bude systém nasazen a jejich propojení.

Logický pohled

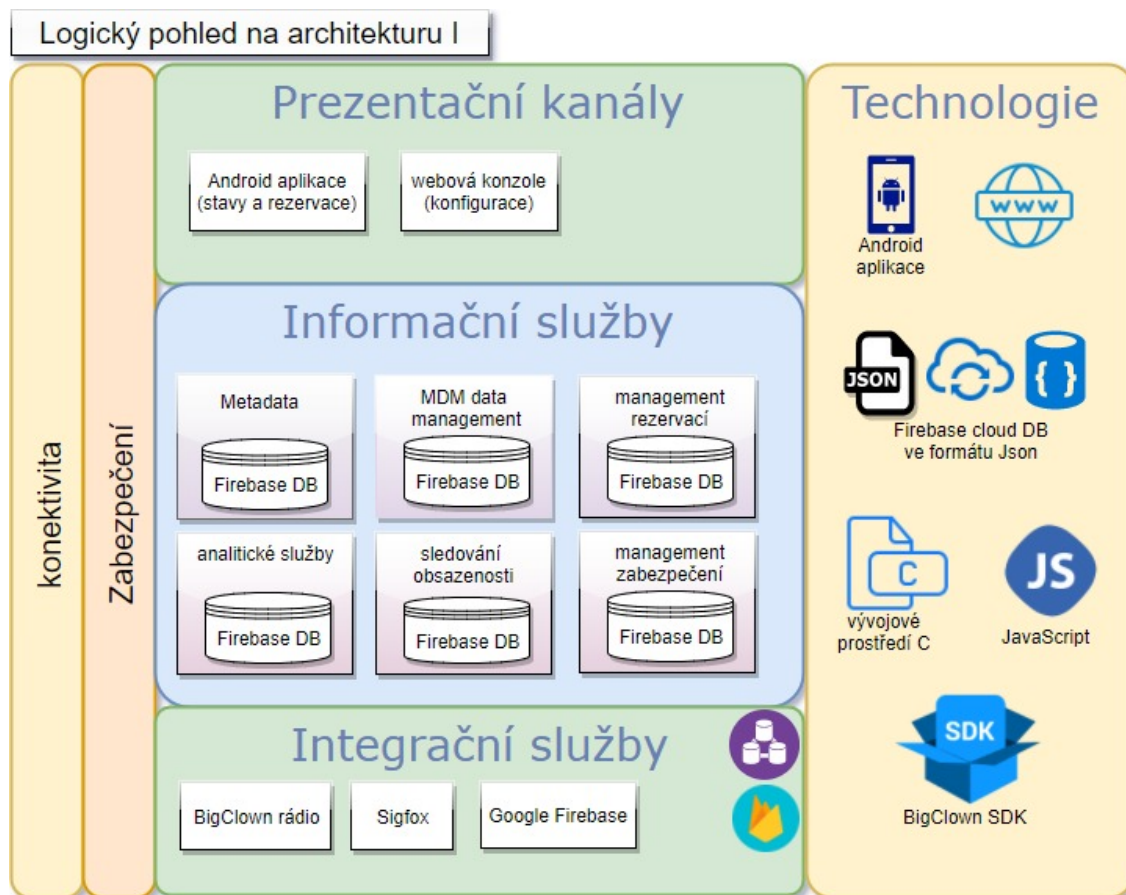
Systém se bude skládat z několika funkčních bloků, které je možné rozdělit do tří základních funkčních celků podle funkcionalit.

- 1. Prezentační kanály**, do kterých spadá aplikace pro platformu Android a webová konzole pro konfiguraci Sigfox a Firebase služeb
- 2. Informační služby**, které obsahují moduly pro provádění business logiky, ukládání dat a práci s daty.
- 3. Integrační služby**, které jsou zodpovědné za přenos informací mezi garáží, cloudovými službami a uživateli

Konektivita a zabezpečení se promítá do všech funkčních bloků, jak je vidět na následujícím diagramu.

Pro zajištění funkcionality je potřeba použít několik různých technologií a postupů a ty navzájem propojit.

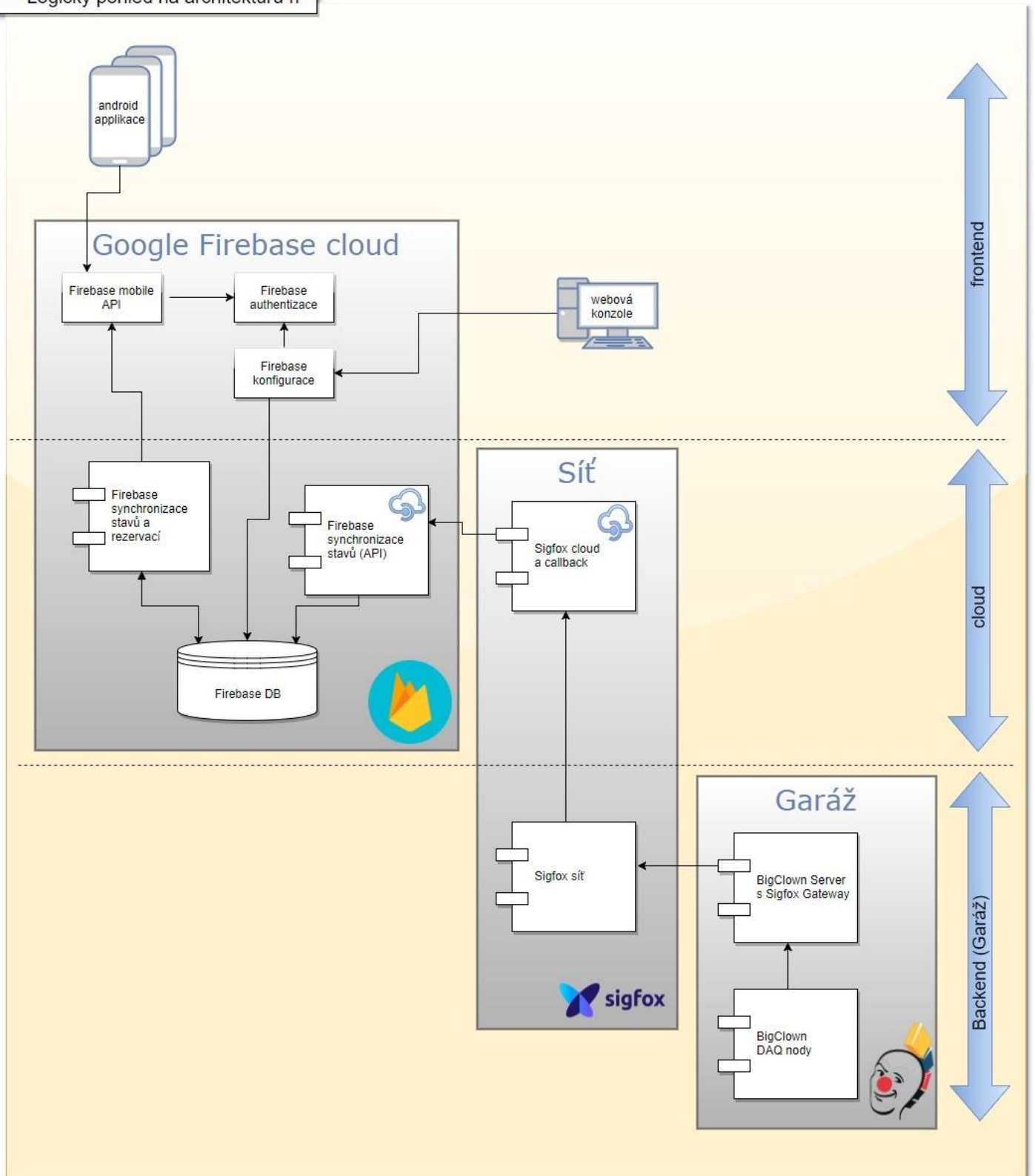
Hlavní zástupci použitých technologií jsou zobrazeny v následujícím diagramu ve sloupci Technologie.



Na stejné bloky systému lze nahlížet také z pohledu propojení a přenosu informací.

Na diagramu níže je cílový systém zobrazen při rozčlenění do tří vrstev Frontend, cloud, backend a jejich propojení.

Propojení těchto komponent je vyobrazeno na následujícím diagramu.



1. Backend : v garáži budou nainstalovány dva druhy zařízení. DAQ nody a Server node se Sigfox gateway. Ta bude propojena se sítí Sigfox.

2. Cloud : Cloudové služby Sigfox a Firebase umožní ukládání změn stavů parkovacích míst, stavy baterií a rezervací.

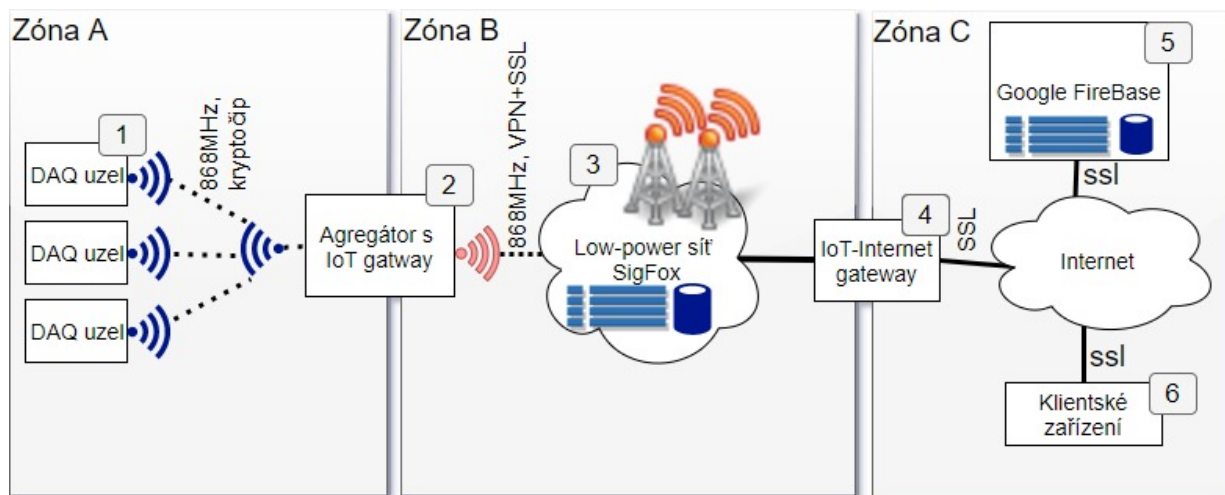
3. Frontend : Webová konzole umožní nastavení Firebase. Android aplikace umožní sledování stavu parkovacích míst a správu rezervací.

Procesní pohled - tok dat

Informace o obsazenosti parkovacích míst se bude zjišťovat DAQ uzlech. Zde se bude binárně kódovat a posílat do Agregátoru pomocí šifrované komunikace mezi zařízeními BigClown. V Agregátoru se budou přichozí data analyzovat a bude se vyhodnocovat nutnost zaslání synchronizační zprávy na vzdálené úložiště. Pro zaslání zpráv bude využita síť SigFox, která rovněž pracuje na frekvenci 868 MHz. Zpráva zůstane uložena u operátora a bude přeposlána přes callback a IoT gateway do Google Firebase. Zde bude rozkódována a uložena do určité struktury. Následně budou o změně stavu notifikovaná připojená zařízení. Graficky je tento proces zachycen na následujícím obrázku.

Informace o rezervacích budou rovněž ukládány a spravovány v Google Firebase.

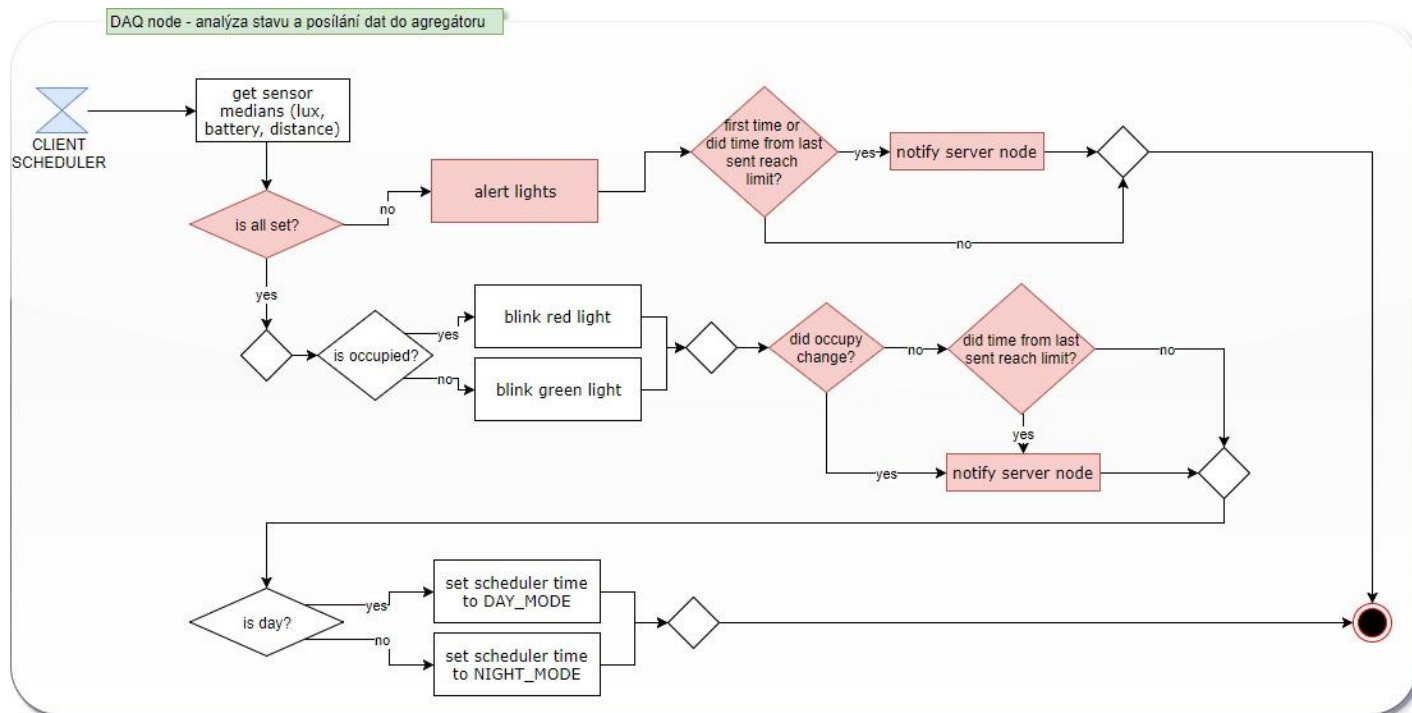
Následující diagram je rozdělen do tří zón, kde zóna A je v garážích, zóna B je Sigfox operátor, který bude zodpovědný za přenos zpráv, a zóna C je internet a služby internetu.



Procesní pohled - vnitřní logika komponent

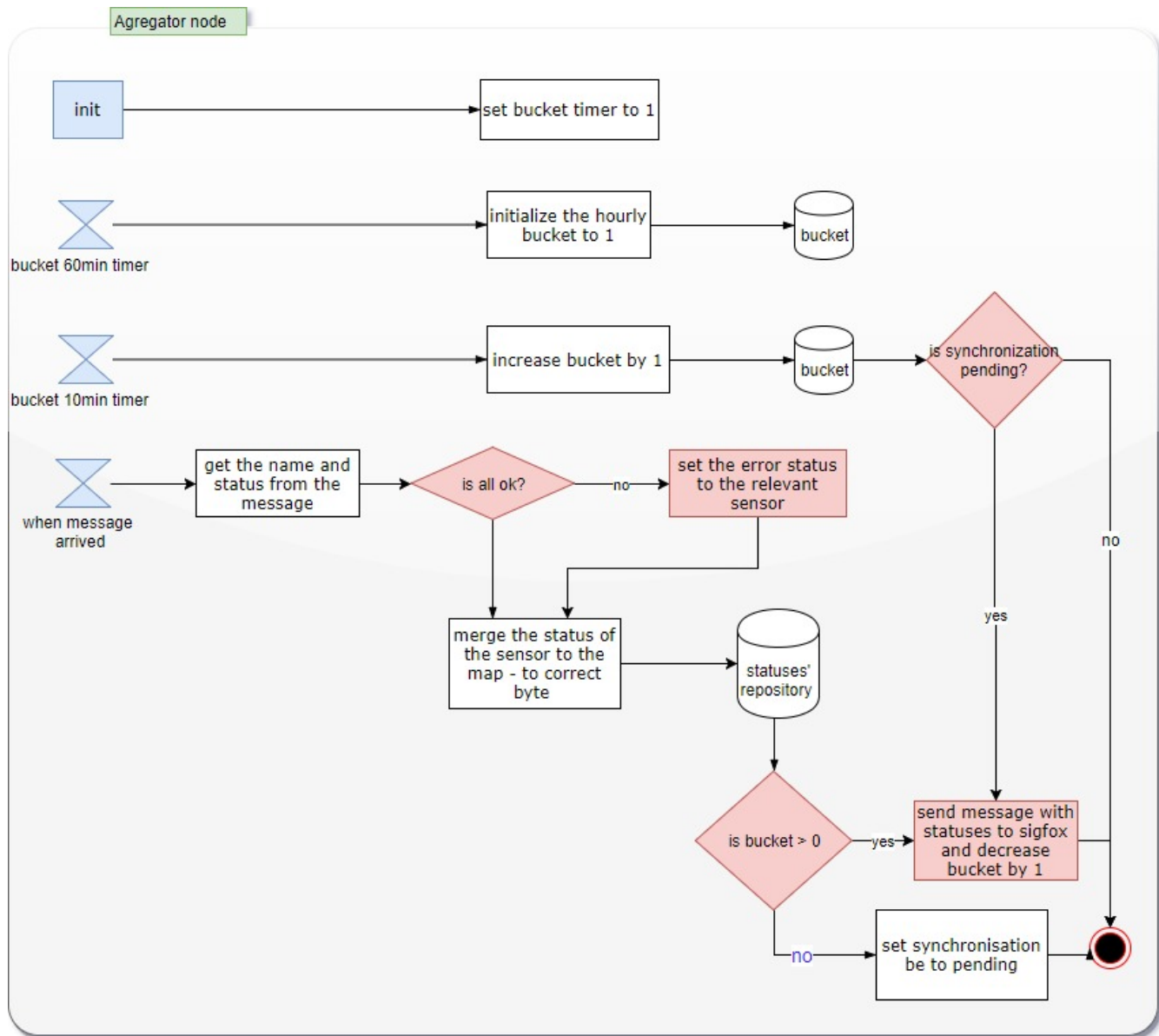
Systém se bude skládat z několika částí, a v každé části bude ve smyčce probíhat určitá logika.

1. v DAQ uzlu se bude snímat stav parkovacího místa, snímání bude probíhat v určitých časových intervalech, kde tyto intervaly budou prodlužovány v případě, že bude v garáži tma. Jde o úspornou funkci řešení. Notifikace agregátoru proběhne vždy, když bude změna. V případě že se změna nestala za poslední hodinu bude poslána synchronizační zpráva. Navrhovaný postup je vyobrazen v diagramu níže.



2. v Agregátor uzlu - se budou sbírat data z jednotlivých DAQ nodů a budou se agregovat do jedné zprávy. V případě, že se neobdrží zpráva od některého z čidel za poslední hodinu, označí se čidlo ve zprávě jako chybné. Tento uzel komunikuje se Sigfox sítí a je potřeba zohlednit v logice limity této sítě. Je možné odeslat šest 12B zpráv za hodinu. V navrhované logice bude existovat pomyslné úložiště neposlaných zpráv, které se bude navyšovat každých 10 minut o 1, a zároveň se bude resetovat každou hodinu. Tímto způsobem je možné zaručit nepřekročení limitu a zároveň synchronizovat stav parkoviště ve chvíli, kdy ke změně dojde.

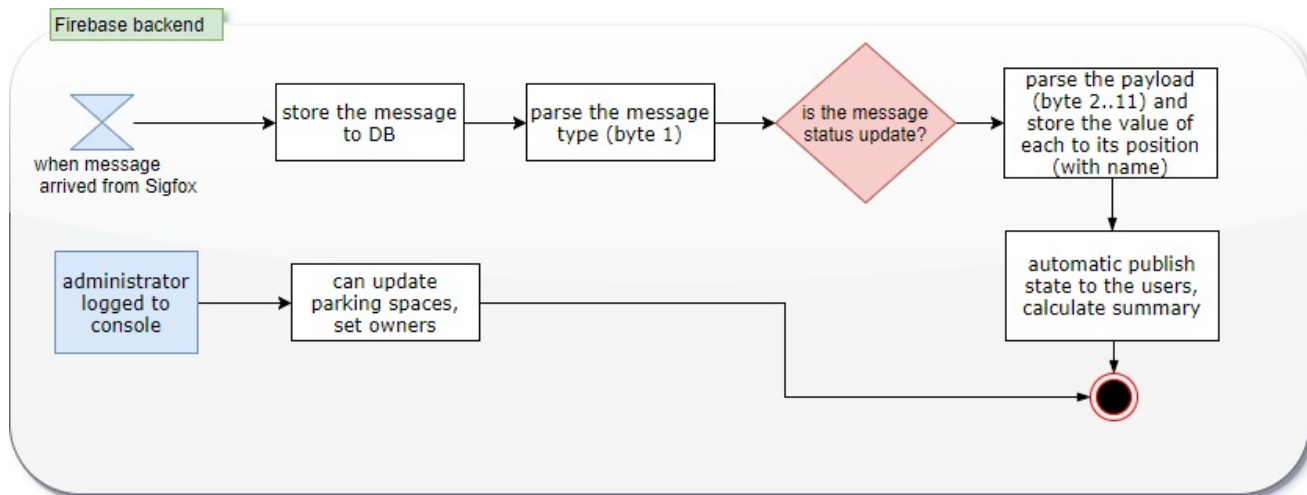
Tato logika je znázorněna v následujícím diagramu:



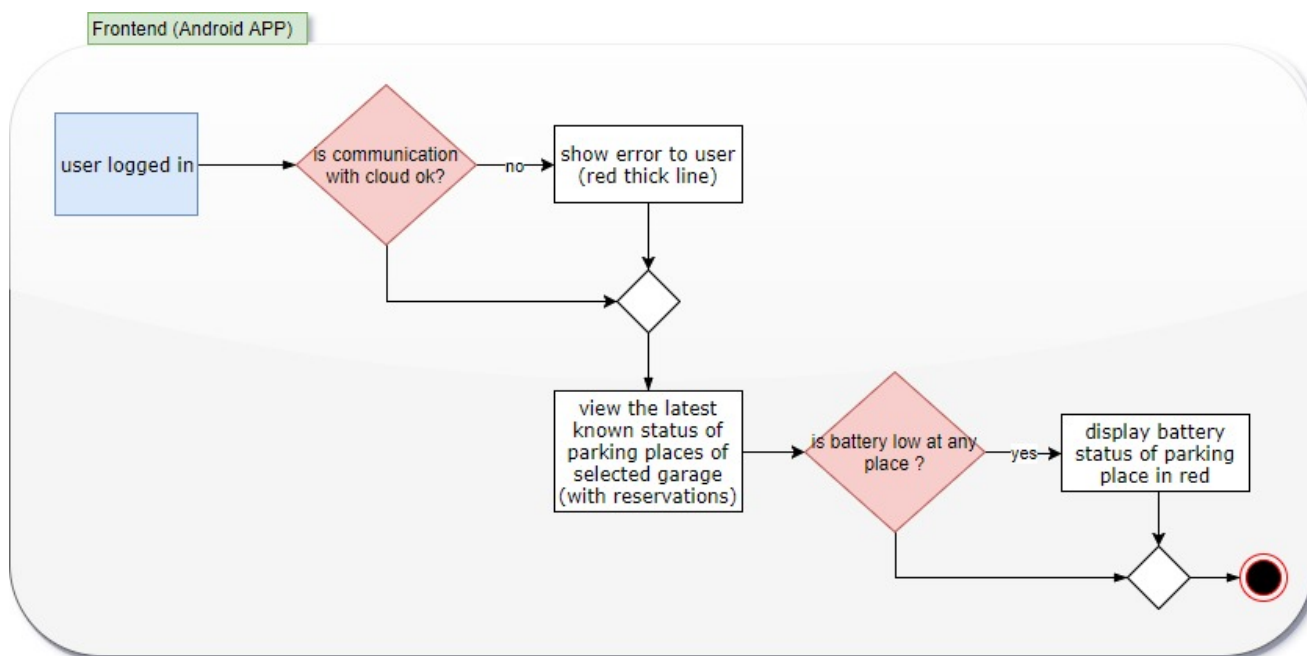
3. v Sigfox backendu (IOT gateway) - se přijatá zpráva pouze přepoše do Firebase úložiště, jak je znázorněno v následujícím diagramu.



4. v Google Firebase (Datový sklad, analyzátor dat) - se po každé příchozí zprávě ze Sigfoxu provede logika, která přečte příchozí zprávu, zjistí z ní stavy parkovacích míst a napáruje je do json DB tak, aby informace byly čitelné pro připojené aplikace. Logika je znázorněna v následujícím diagramu.



4. v Android aplikaci - se bude kontrolovat stav připojení, a v případě chyby se stav zobrazí uživateli. Rovněž se v případě kritického stavu baterií zobrazí varování.



5. Funkcionality v Android aplikaci

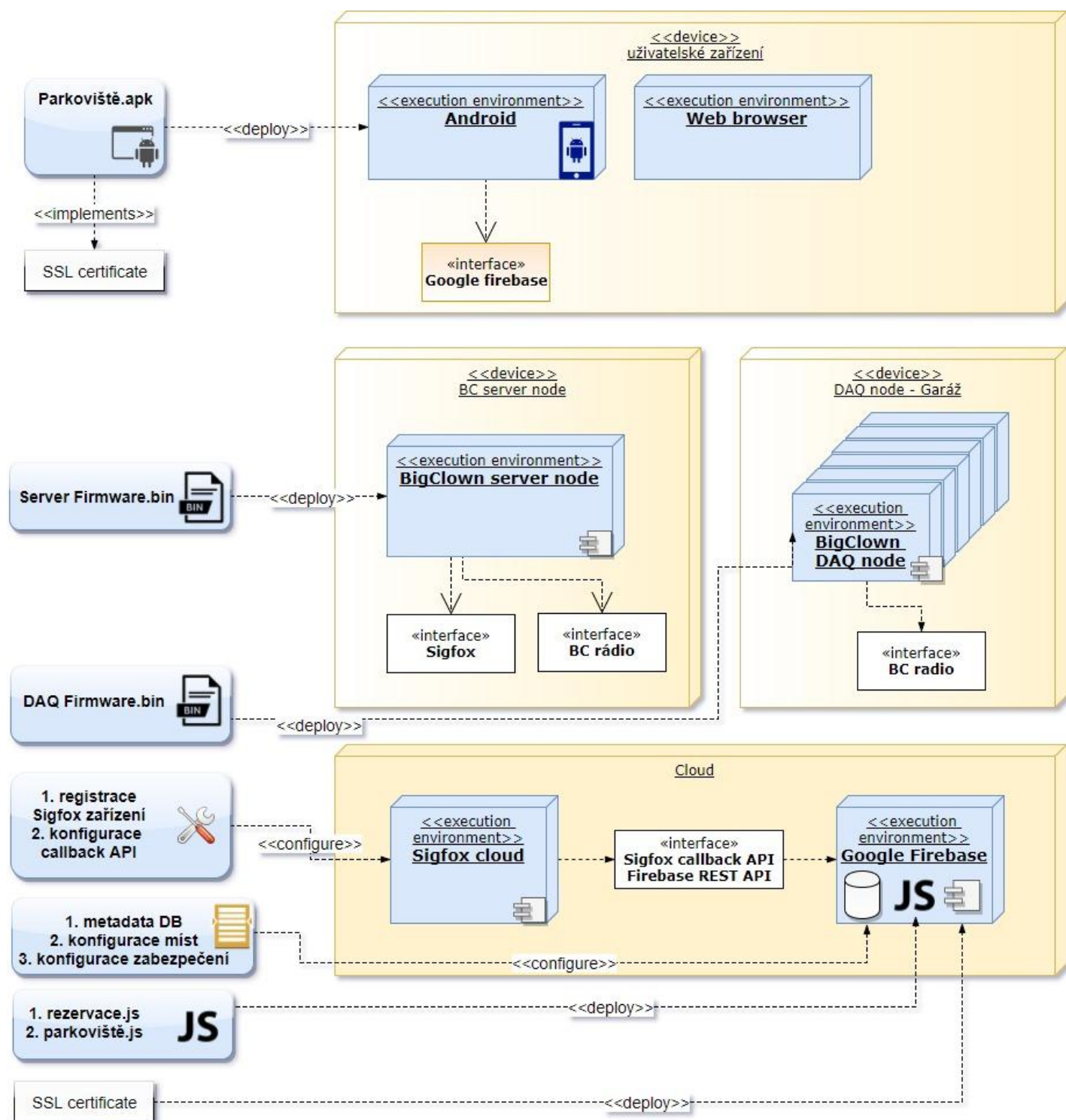
Přihlašování, zjišťování stavů, a práce s rezervacemi v aplikaci Android je detailně popsána v sekci [GUI](#)

Nasazení

Systém se bude skládat z několika funkčních celků, které budou nasazeny na různých místech.

- 1. uživatelská zařízení** - na mobilní zařízení s platformou Android bude nasazena aplikace UCL Parkoviště.
- 2. v garáži** budou nasazeny DAQ nody a ServerNode. Nasazení bude spočívat v instalaci zařízení, ve kterých bude nahrán konkrétní firmware. Při instalaci bude potřeba zařízení zkalibrovat, ověřit konektivitu mezi DAQ nody a ServerNodem. Zároveň je potřeba ověřit sílu signálu ServerNodu k Sigfox síti.
- 3. Cloud** - Sigfox bude nakonfigurován na přeposílání zpráv (obsazenosti a baterií) do Firebase. S pomocí Firebase konzole bude nasazena struktura databáze, budou vyplněna jména vlastníků pro parkovací místa. Zároveň budou nasazeny skripty pro provádění business logiky (dekompozice zpráv, správa rezervací)

Deployment diagram je znázorněn níže.



Implementace/Implementation

V následujících kapitolách je popsáno řešení z pohledu implementace, kde se funkcionality běžně dělí podle vrstev na prezentační, aplikační, persistentní. V této implementaci je přidána jedna vrstva navíc a tu zde nazýváme IOT vrstva.

Tato vrstva je zodpovědná za sbírání dat a přenos informace mezi zařízeními a je jí věnována jedna samostatná podkapitola.

Prezentační vrstva

Prezentační vrstva bude zahrnovat Android aplikaci pro smartphony a webové rozhraní Firebase. V rámci Android aplikace bude možné zobrazovat obsazenost parkoviště, vytvářet, editovat a mazat rezervace pod daným uživatelským účtem či v zastoupení (recepce).

Webové rozhraní je součástí Google Firebase a bude sloužit správci parkoviště pro přiřazování vlastníků parkovacích míst.

Aplikační vrstva

Významná část aplikační logiky bude realizována v rámci Google Firebase - zejména zpracování zpráv ze sítě SigFox a zpracování rezervací iniciované z Android aplikace.

Persistentní vrstva

Data o rezervacích a vlastnicích parkovacích míst budou spravována v rámci NoSQL databáze Google Firebase (cloud). Stejně tak se do této databáze bude prostřednictvím sítě Sigfox přenášet aktuální obsazenost parkovacích míst a informace o stavu baterií jednotlivých DAQ uzlů.

Vrstva IoT

IOT vrstva bude reprezentována jednotlivými DAQ uzly, které budou odpovědné za dané parkovací místo - snímání stavu. Fyzicky bude každý DAQ uzel představovat mikrokontrolér BigClown vybavený senzorem vzdálenosti pro vyhodnocení, zda parkovací místo je volné či obsazené. DAQ uzly budou předávat data řídicímu uzlu - tzv. Server node, který bude získaná data dále předávat pomocí sítě Sigfox do cloudového úložiště - Firebase databáze.

Velikost a omezení systému

Tato kapitola obsahuje popis hlavních dimenzionálních vlastností systému, které mají dopad na architekturu, a omezení, která mají dopad na výkon systému.

Omezení řešení

1. Množství DAQ nodů - Do systému bude zapojeno šest parkovacích míst, které využívá škola Unicorn College na Parukářce. Bude tedy vyžadovat 6ks DAQ nodů.

2. Množství spárovaných DAQ nodů pro k jednomu server nodu - teoretický limit udávaný výrobcem je 32 spárovaných zařízení pro jeden ServerNode v zabezpečeném módu. Je to z důvodu omezeného úložiště pro ukládání klíčů zařízení.

Systém je rozšiřitelný. Pro větší množství parkovacích míst je možné zavést více server nodů a více Sigfox zařízení a více předplatných pro Sigfox.

Výrobce rovněž udává, že je možné propojit bezdrátově až 256 zařízení, ale za cenu vyřazení kryptování informací.

3. Sigfox omezení - je omezení v množství (6/h) a ve velikost (12B) zasílaných zpráv v síti Sigfox. Nejedná se o technologické omezení, ale o nařízení regulačního úřadu. Překračování těchto limitů může mít za následek zakázání služby.

Pro současných šest DAQ nodů je dostačující jeden agregační Server node s připojením do sítě Sigfox. Zpráva bude obsahovat hlavičku a 11Bytů pro uložení stavů 11 parkovacích míst a detailním stavem jejich baterií.

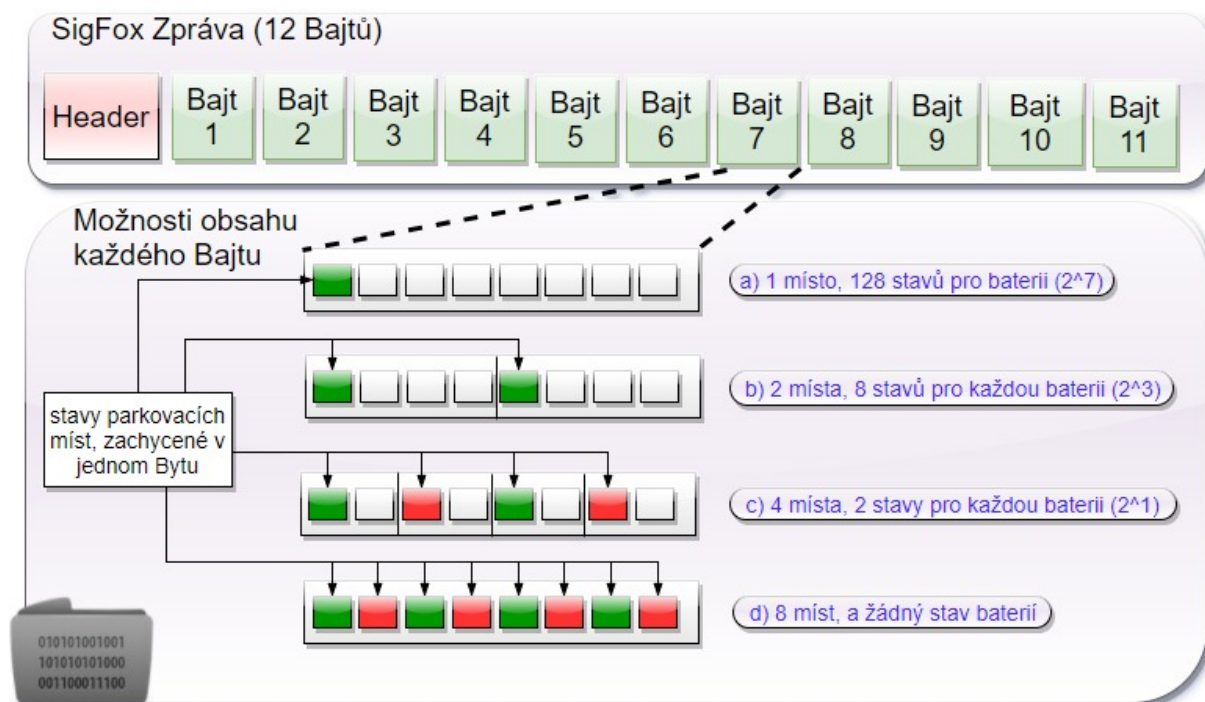
V případě ukládání stavu jednoho místa (**varianta "a" na obrázku níže**) do jednoho Bytu je možné v jedné agregované zprávě zaslat 11 stavů parkovacích míst se stavem obsazenosti a s relativně detailním stavem baterie.

V případě potřeby uložení stavu dvou čidel do každého jednoho Bytu (**varianta "b"**) zbyde pro kontrolu baterie 8 stavů a v jedné agregované zprávě je možné zaslat stav 22 čidel.

Je možné poslat v v jednom Bytu stav 4 míst (**varianta "c"**), kde u každého budou pouze dva stavy baterie (např 0=kritický, 1=OK). Tím bychom mohli v jedné zprávě agregovat až 44 parkovacích míst. Zde by se projevil limit na množství párovaných čidel k jednomu agregačnímu nodu.

Pro úplnost diagramu je zde ještě hypotetická **varianta "d"**, kde se v rámci zprávy neposílají stavy baterií. V takovéto konfiguraci by teoreticky mohla zpráva obsahovat stav obsazenosti 88 parkovacích míst. Synchronizace stavu baterií by se v tomto případě mohla provádět ve zprávě s jinou hlavičkou. Takovéto zprávy by se posílaly méně často - například 1x denně.

Ukázka informace uložené v jednom Bytu pro varianty a až d je znázorněna graficky níže:



3. Google Firebase - pro účely tohoto projektu je dostačující program Spark Plan, který je zdarma. V tomto programu je omezení pro maximálně 100 souběžných připojení, úložiště nemá být větší než 1GB a za měsíc nebude staženo více než 10GB dat. Při větším objemu je možné přejít na program Flame plan za 25USD/měsíc.

4. Android aplikace vyžaduje systém Android 4.4, nebo vyšší

Přehled důležitých architektonických rozhodnutí a omezení

Číslo	Téma	Architektonické rozhodnutí
1	Mikrokontrolér	Bude použit mikrokontrolér BigClown, s bateriovým modulem, s rádiem a kryptočipem
2	Čidlo na měření světla	Bude použit LUX modul od společnosti BigClown
3	Čidlo na měření vzdálenosti	Bude použito ultrazvukové čidlo HC-SR04P
4	Protokol pro komunikaci	Bude použit rozšířený MQTT protokol pro komunikaci mezi DAQ nody a Agregátor nodem
5	Síť pro komunikaci z agregátoru do internetu	Bude použita síť SigFox
6	Řešení omezení Sigfox sítě. 1. velikost zprávy=12B 2. maximálně 6 zpráv/h	<p>1. Každá SigFox zpráva (o velikosti 12Bajtů) bude obsahovat stav až 11 parkovacích míst. V řídicím Bajtu budou informace o agregátoru, v dalších 11 Bajtech bude uložen stav 11 parkovacích míst. Pořadí každého parkovacího místa ve zprávě bude odpovídat pořadí spárování. Logická reprezentace bude pak zajištěna v cílovém systému.</p> <p>2. Na začátku každé hodiny se zašle tzv. heartbeat zpráva. Teoreticky by mohla synchronizace probíhat každých 10 minut. Pro zrychlení synchronizace při změně stavu se nebudou odesílat zprávy každých 10 minut, ale budou se posílat pouze tehdy, pokud došlo ke změně a pouze pokud nebyl vyčerpán limit zpráv.</p>

7	Zobrazení dat uživateli	Pro zobrazení dat, systém rezervací a autorizace bude použito cloudové řešení Google Firebase. To bude získávat data ze SigFox sítě a ukládat do DB. Následně bude jednoduchá Android aplikace načítat tato data a bude zobrazovat stavy parkovacích míst uživatelům a bude umožňovat provádět rezervace.
---	-------------------------	---

Kvalita systému

Nový systém bude splňovat následující kvalitativní požadavky:

Rozšiřitelnost -

1. systém bude škálovatelný. Pro jeden Server node může být při současném nastavení synchronizačních zpráv až 11 DAQ nodů (parkovacích stání). Při snížení počtu stavů baterií je možné pro navýšit počet DAQ nodů na 32. Pro větší implementace je pak možné využít i více Server nodů.
2. systém bude modulární. Moduly bude možné zaměňovat za jiné, pokud bude zachován stejný interface. Bude možné například použít backend (DAQ nody, server node, sigfox) a navázat na podnikový informační systém.

Spolehlivost

1. spolehlivost zjišťování stavu parkovacích míst bude zvýšena opakováním čtením hodnot a zjišťováním jejich mediánu.
2. předpokládme, že Sigfox a Google Firebase jsou spolehliví poskytovatelé.

Bezpečnost

1. mezi DAQ nody a Server nodem bude zabezpečená komunikace s pomocí kryptočipu
2. bezdrátová komunikace Sigfox bude zabezpečena pomocí SSL certifikátu
3. mezi Sigfox a Google firebase bude komunikace zabezpečena s pomocí SSL certifikátu
4. mezi Google firebase a android aplikací bude zabezpečena pomocí SSL certifikátu