



# GOLEMIO

P R A G U E C I T Y D A T A

## Datová platforma hl. m. Prahy Golemio – rutinní provoz

Analýza uživatelských požadavků, návrh architektury a implementace řešení

Verze 1.3, únor 2019

Operátor ICT a.s.

**O I** OPERÁTOR  
**C T** ICT

PRA  
PRA  
PRA  
PRA

HA  
GUE  
GA  
G

# Manažerské shrnutí

Od 1. 1. 2018 probíhá pilotní projekt Datová platforma hlavního města Prahy a od 1. 7. 2019 přechází v rutinní provoz. Cílem pilotní fáze projektu je otestování jak organizačních, tak technických možností napojení Datové platformy na datové zdroje různé povahy, jejich zpracování a nastavení dalšího rozvoje. Pilotní fáze pokračuje až do 30. 6. 2019, nicméně je nutné již v tuto chvíli určit další postup, a to hlavně s ohledem na pořízení softwarového a technologického zázemí.

Cílem této analýzy je definování funkčních požadavků na SW řešení a infrastrukturu Datové platformy v rozsahu nutném pro rutinní provoz, rozpadnutí výsledků funkční analýzy dále do konkrétních technologických požadavků a vybrání nevhodnější varianty implementace řešení. V návaznosti na vybranou variantu/technologie pak rozpracování do takového detailu, aby bylo možné vytvořit konkrétní zadání pro vývojářský tým, stanovit harmonogram, požadavky na infrastrukturu, lidské zdroje a rozpočet.

V obecnější rovině rozdělit způsob přístupu k řešení na formu využití nějakého existujícího komerčního nástroje (krabicového řešení), kompletní vlastní vývoj a kombinaci různých existujících opensource komponent, frameworků a vlastního vývoje. Již během analýzy se ukázalo, že „krabicové“ řešení není vhodné, jelikož žádné z existujících nesplňuje požadavky projektu zejména s ohledem na modularitu a variabilitu. Pro nejlepší určení konkrétních technologií a rozsahu vlastního rozvoje se ukázalo, že je nejdůležitější si ty klíčové nejdříve otestovat na konkrétních příkladech integrace a zpracování dat. Proto tým Datové platformy již během analýzy důsledně testoval různé možnosti a výsledkem tohoto testování je funkční prototyp, který je zároveň základem pro rozvoj uceleného řešení, které bude možné nasadit do produkce.

Výsledná varianta řešení je kombinací existujících opensource nástrojů (kromě vizualizační a analytické komponenty, které budou využívat také SW Microsoft PowerBI a Arcgis) a vlastního vývoje. Tato varianta umožní částečnou nahraditelnost jednotlivých částí systému, zamezí vendor lock-in a umožní vývoj a rozvoj jednotlivých částí systému bez závislosti na jednom dodavateli či jednom SW. Klíčovým požadavkem je, aby bylo možné Datovou platformu zároveň poskytnout jako opensource komukoliv volně k využití, a proto je postavena na co nejvíce otevřených, běžně dostupných a známých technologiích. Z toho poté vyplývá naprosto zásadní výhoda, že veškeré know-how zůstává v OICT.

**Primárním účelem dokumentu není jeho zveřejnění, ale poskytnutí podkladu pro další rozhodnutí vedení společnosti. Přesto se však domníváme, že obsahuje velmi cenné know-how, které vzniklo za veřejné peníze. Z toho důvodu dokument poskytujeme veřejnosti volně k využití. Může se ovšem zdát, že některé části jsou zjednodušené, nejsou zmíněny všechny možnosti nebo nejsou v dostatečném detailu. To je způsobeno právě tím, že se pro rozhodnutí o dalším postupu nejednalo o tolik zásadní informace.**

# Funkční požadavky na Datovou platformu vzhledem k uživatelům

Při stanovování plánu dalšího rozvoje (nebo zcela nového vývoje) jakéhokoliv systému je naprosto klíčové určení (kategorizace) koncových uživatelů a jejich funkčních požadavků na systém. Ty se pak promítají do podrobného rozpadu funkčních požadavků, které definuje správce systému. V případě Datové platformy se uživatelé rozdělují na dvě skupiny, na externí partnery, kterým budou data a vizualizace poskytovány pod speciálními uživatelskými přístupy a na data a vizualizace zprostředkována všem skrze veřejný webový portál Golemio.

Definice toho, co ve výsledku koncový uživatel uvidí a co mu Datová platforma bude umožňovat je širší a nepředpokládá se, že vše bude implementováno v první fázi vývoje, nicméně je klíčové, aby bylo do budoucna umožněno datovou platformu rozvíjet požadovaným směrem podle konkrétních projektů a požadavků partnerů. **Cílem analýzy není detailní popis produktu, ale identifikace požadavků, která umožní rozhodnutí o technologiích, stanovení harmonogramu implementace, strategie pořízení a odhad nákladů.**

## Externí partner (MHMP, MČ, firmy)

### Správa uživatelů a přístupy

Datovou platformou budou používat uživatelé z městských částí a nejrůznějších městských organizací (Magistrát hlavního města Prahy, Institut plánování a rozvoje, aj.) a to z odlišných oddělení. Někteří z uživatelů mohou vystupovat jako správci určitých datových sad, zatímco jiní budou data pouze zobrazovat, a to v jiném detailu a granularitě, než budou přístupná široké veřejnosti. Datová platforma musí proto nabízet různé možnosti správy uživatelských přístupů.

- DP umožní zobrazení vybraných dashboardů, grafů a reportů pro potřeby partnera
- DP umožní správu udělování přístupů k API (definování práv a typu operací), která musí být na úrovni jednotlivých datových sad a jednotlivých CRUD operací Správce DP (OICT)
- DP umožní definovat oprávnění přístupu k datům na základě podmínek datového obsahu, např. omezení zobrazení jenom na data vybraných městských částí

### Možnosti ovládání a náhledu pro partnery

Jednou z přidaných hodnot v používání platformy pro externí partnery OICT budou možnosti generování reportů nad ukládanými daty a přístup do analytických nástrojů platformy.

- DP umožní partnerovi generování nových reportů skrze GUI na předem definované DS
- DP umožní nastavení a zaslání notifikace partnerovi ohledně nestandardního chování senzorů (alerting)
- DP umožní partnerům definování a generování datových výstupů
- Platforma umožní přístup k analytickým nástrojům i externím uživatelům

## Open data

### Správa uživatelů a přístupy

Platforma musí poskytovat přehledné rozhraní obsahující seznam všech dostupných datových sad (pro potřeby OICT a jeho partnerů) a zároveň musí být možné vybrané datové sady označovat příznakem, že se jedná o otevřená data. Tento katalog obsahující pouze dostupná otevřená data musí být volně přístupný veřejnosti. Pro zajištění vyšší bezpečnosti a míry kontroly nad daty a infrastrukturou musí platforma umožňovat správu přístupů k API.

- DP umožní správu přidělování přístupů k API
- Platforma bude disponovat katalogem datových sad (může a nemusí být CKAN) včetně jejich umístění a oprávnění
- Platforma bude pro jednotlivé datové sady evidovat, zda se jedná o otevřený datový set a pak bude tyto informace zobrazovat pro veřejnost (katalogizovat)
- Platforma umožní zrcadlení odkazů na datové sady i do dalších datových katalogů (viz současný Portál pro otevřená data hl. m. Prahy)

### Možnosti sdílení dat a vizualizací

Uživatelé, kteří budou přistupovat k datům uloženým v Datové platformě, budou na různé technické úrovni. Platforma tedy musí umožnit jak přístup k veřejnému API a širokým možnostem exportu dat, tak i uživatelskou tvorbu nových reportů kombinujících vybrané datové sady.

- DP umožní vystavení veřejného API
- DP umožní vložení vytvořených vizualizací a tabulek do HTML stránek
- DP umožní automatické generování dat formátu CSV, XLS, JSON, geoJSON apod. do katalogu otevřených dat
- Platforma umožní uživatelům tvorbu vlastních jednorázových reportů nad vybranými datovými sadami (zvolit vizualizaci a kombinaci daných dimenzí, viz například <https://monitor.statnipokladna.cz/analyza/?query=4>)

# Funkční požadavky na Datovou platformu vzhledem ke správci DP (OICT)

Funkční požadavky na Datovou platformu z hlediska jejího provozovatele vycházejí především ze zkušeností s pilotním provozem datové platformy.

## Integrace dat

Integrační rozhraní vystavuje API, přijímá data, provádí kontroly a další zpracování. Kromě samotného uložení dat může realizovat i složitější workflow, např. na základě podmínek zasílat alarmy apod. DP bude umožňovat integraci dat pomocí uživatelského rozhraní pro definované typy a zdroje dat a v případě specifických zdrojů dat umožní toto rozhraní naprogramovat.

### Datové zdroje

V současnosti neexistuje ucelený závazný seznam budoucích datových zdrojů, protože jednotlivé projekty jsou vybírány s ohledem na strategické priority města a vzhledem k dynamičnosti prostředí, jakým Smart City je, se jejich specifikace také vyvíjejí v čase. Hlavním cílem je tedy modularita a možnosti přidávat nové datové zdroje.

- Data ze senzorické sítě
- Generická data – CSV, XLS, JSON, XML apod.
- Platforma musí umět ukládat výstup ze zpracování kamerového streamu (např. počet automobilů v daný okamžik)
- Platforma musí mít možnost ukládat surová binární data (např. obraz z kamery za určitý časový úsek), ale nebude přijímat kamerový stream ani nebude video zobrazovat

### Typy rozhraní

Aplikační rozhraní budou pro většinu uživatelů vstupní a výstupní brána do datové platformy. Za tímto účelem byly definovány následující typy aplikačních rozhraní, které musí platforma poskytovat:

- API gateway pro senzorická data
- Kvazistatická data
- Externí API
- Externí DTB

### Další požadavky na aplikační rozhraní

Vzhledem k měnícím se nárokům na aplikační rozhraní při provozu platformy je nezbytné, aby platforma podporovala jeho velkou modularitu a upravitelnost.

- Platforma musí umožňovat programátorskou úpravu API pro přístup k datům na straně OICT (např. úprava endpointů)
- Platforma nesmí mít technologický limit pro API k přístupu datům (export i import). Limit bude dále možné řídit pomocí parametrů pro jednotlivé role

- Platforma umožní správu udělování přístupů k API (definování práv a typu operací) musí být na úrovni jednotlivých datových sad a jednotlivých CRUD operací
- Platforma umožní správu přístupů k náhledům na data. Vytváření rolí (reporting, vizualizace, BI) musí být na straně správce DP (OICT) a musí být možná na úrovni jednotlivých DS
- Platforma umožní zpracovávat senzorická data, pokud integruje data od jednoho dodavatele, musí být možné integraci replikovat na straně správce DP (OICT)
- Platforma nebude komunikovat přímo se senzory, ale bude vždy komunikovat se systémem dodavatele, který zajistí komunikaci se senzorem.

## Ukládání dat

Z omezení použité technologie v pilotním provozu vyplývají požadavky na větší kontrolu nad datovým úložištěm. Jedná se především o scénáře vkládání historických případně jinak zpracovaných dat, exporty velkého množství dat apod.

- Platforma musí umožňovat přímý přístup do databáze uložených dat (pouze pro OICT nebo nástroje používané OICT)
- Platforma umožní manuální vložení dat do databáze místo integračního rozhraní (nahrání již zpracovaných dat, odpovídajících definované struktuře)
- Platforma umožní manuální vložení historických dat do databáze (i těch, ke kterým již existuje API)

## Dispečink

Protože platformu budou požívat uživatelé z různých městských organizací a částí, je nezbytné, aby obsahovala uživatelsky přívětivé rozhraní pro správu senzorů a vytváření pravidel (na základě času, překročení hodnot či splnění nějaké jiné podmínky) a odesílání upozornění. Níže uvedené požadavky není nutné plně realizovat v rámci prvotní implementace DP. Je možné s finalizací požadavků na uživatelské možnosti počkat až na zkušenosti z provozu.

- Platforma musí mít rozhraní pro odebrání senzoru / zapojení nového senzoru již používaného typu (management zařízení) - uživatelské rozhraní (GUI)
- Platforma umožní v rámci komponenty Dispečink volání API dodavatelů technologií (zdrojů dat)
- Platforma umožní pomocí nastavení pravidel, po jejichž nesplnění / splnění odešle alert (např. koncentrace CO<sub>2</sub> > definovaný limit, odeslání emailu)
- Platforma bude mít specializovaný dashboard na zobrazení alertů
- Platforma bude umožňovat nastavení uživatelských alertů (pravidla if - then na základě funkcí: <, >, =, &&, || a jejich kombinací)
- Platforma umožní nastavení pravidel, která budou spouštěna v závislosti na čase (automatizace určitých úkonů)
- Platforma umožní nastavení eventů – pravidel, která budou spouštěna v závislosti na jiné události



## Reporting v datové platformě

Jedním z cílů platformy je umožnit vizualizaci různých datových sad (např. časové řady a různá prostorová data) a tyto vizualizace kombinovat za účelem hledání souvislostí v datech.

- Platforma umožní zobrazení dat na časové řadě
- Platforma umožní zobrazení dat pomocí okna s jednou hodnotou (např. aktuální hodnota, denní průměr, ...)
- Platforma by měla být schopna zobrazit v dashboardech tzv. věkovou pyramidu (tento způsob zobrazení se v současnosti využívá v některých reportech v rámci hl. m. Prahy)
- Platforma umožní zobrazení real-time dat na mapě (např. poloha vozů PID a MHD)
- Platforma umožní zobrazení na mapě heatmapu
- Platforma umožní zobrazení na mapě předem definovaný GeoJSON
- Platforma umožní zobrazení na mapě předem definovaný GeoJSON Polygon
- Platforma umožní zobrazení na mapě kombinaci real-time dat, heatmap, GeoJSON dat a GeoJSON Polygonů
- Platforma by měla být schopna zobrazit mapové podklady (GIS data) IPR (např. jako podkladovou vrstvu)

## Analytické možnosti

Vzhledem k neustálému vývoji v oblasti datové analýzy je nezbytné, aby platforma uměla nad uloženými daty provádět datové analýzy a zároveň umožňovala napojení na externí analytické nástroje.

- Platforma umožní provádění ad-hoc datové analýzy (regrese, statistické zpracování)
- Platforma umožní generování pravidelných reportů
- Platforma umožní integraci jazyka R
- Platforma umožní provádění analýz v externím programu (R studio, AML studio, MATLAB, ...), který umí integrovat data z platformy a následně uložení výstupů analýz
- Platforma umožní na napojení analytických cloudových služeb (např. Azure machine learning studio – AML studio)

## Architektura a provoz

Požadavky na architekturu a provoz datové platformy vycházejí z poznatků pilotního provozu datové platformy a nejlepší praxe při návrhu architektury komplexních IT systémů. Jedná se především o motivaci zabránit přílišné závislosti na jednom dodavateli nebo technologii.

- Platforma bude modulární systém, tedy s možností rozvíjet jednotlivé komponenty separátně (např. oddělení databázové a analytické vrstvy, včetně popisu rozhraní mezi nimi)
- Platforma by měla být ve výsledku provozována v městském datovém centru (Kongresové centrum Praha), ale záleží na budoucím objemu dat a povaze dat a také na způsobu provozu ostatních systémů OICT (např. Multikanálový odbavovací systém). Je i možnost, že bude nakonec provozována v komerčním cloudu.

## Model jednání pro jednotlivé uživatelské role

V rámci uživatelské interakce s DP byly identifikovány následující uživatelské role:

- anonymní uživatel
- registrovaný uživatel
- datový analytik
- administrátor
- dispečer
- příjemce notifikací

### Anonymní uživatel

Jedná se o uživatele, kteří budou přistupovat k DP bez přihlášení a budou konzumovat otevřená data prostřednictvím dashboardů nebo exportů. Pro tyto uživatele budou nastavena jednotná pravidla přístupu k datům a řízení zátěže.

### API uživatel

Pro přístup k API bude nutná registrace, pro otevřená data ale bude postačovat zjednodušená bezplatná registrace, která nebude ověřována. Technicky se tedy bude jednat o registrovaného uživatele, s ověřením na úrovni anonymního uživatele.

### Registrovaný uživatel

Registrovaní uživatelé budou moci přistupovat i k datům, u kterých bude aplikováno omezení přístupu, a to v rozsahu jim přidělených přístupových rolí. Budou si moci generovat API klíče, které také umožní definovat specifické limity – vyšší počet záznamů v odpovědi, častější dotazy apod.

### Datový analytik

Datoví analytici budou interní uživatelé OICT nebo další oprávnění uživatelé, kteří budou mít rozšířené možnosti přístupu k datům. Tito uživatelé mohou také případně ukládat výstupy svých analýz zpět do DP.

### Administrátor

Administrátor DP bude spravovat uživatelská oprávnění a jednotlivé datové zdroje (např. připojovat nová zařízení).

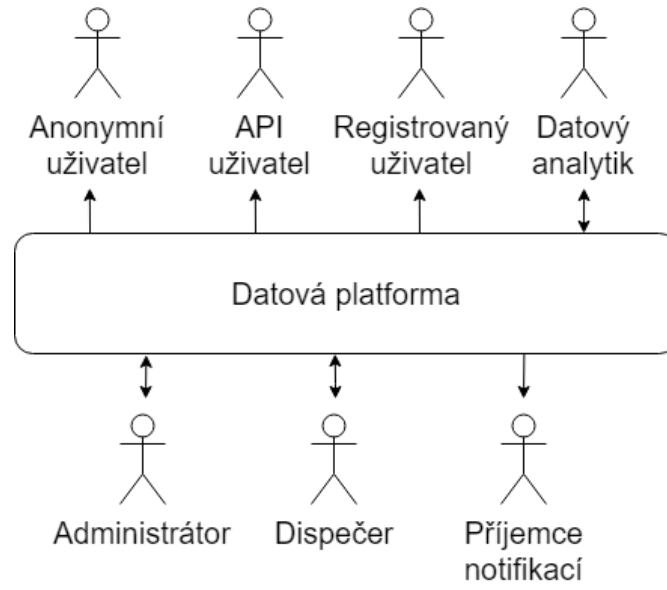
### Dispečer

Pro ovládání zařízení je určena role dispečera. Variantně může být do této role zahrnuta také správa zařízení.

### Příjemce notifikací

Role příjemce notifikací bude typicky kombinována s některou další rolí. Z pohledu interakce se jedná o samostatný případ užití.





# Zhodnocení variant

## Základní informace

Na základě požadavků uvedených v předchozí kapitole a dostupných řešení se nejvíce používá hotového „krabicového“ řešení jako vhodné. Požadavky směřují k vlastnímu vývoji Datové platformy hlavního města (DPHM). Zároveň však vývoj DPHM od úplných základů, bez využití alespoň některých předpřipravených komponent, není časově, ekonomicky ani manažersky zvládnutelný. Různé varianty popsané v této kapitole tak nabízejí menší nebo větší podporu pro tento proces – frameworky, předpřipravené komponenty apod.

OICT plánuje zdrojové kódy výsledné DP zveřejnit jako open source. Aby to dávalo smysl, měly by být, pokud je to možné, použité komponenty také volně dostupné. To umožní, aby kód DP využívali i jiní uživatelé, a případně jej také rozvíjeli.

Text v této kapitole obsahuje výběr některých technologií a možností pro implementaci DP. Nejedná se o analýzu a výčet všech možností, ale spíše výběr typických možností kombinací. Byl vytvořen na základě kombinace zkušeností s definicí a/nebo tvorbou datové platformy a námětů a požadavků OICT.

Kapitola neobsahuje finanční srovnání, které se sice nabízí, ale díky neznalosti dalšího vývoje není možné nebo by se jednalo o čisté věštění z křišťálové koule. Vzhledem k tomu, že se objem dat může v krátké době výrazně znásobit a mohou nárazově vzniknout požadavky na nové speciální služby, je naprosto zásadní, aby byl systém škálovatelný a snadno rozšiřitelný. Zároveň, aby OICT neplatil za služby, které nepotřebuje nebo nevyužívá.

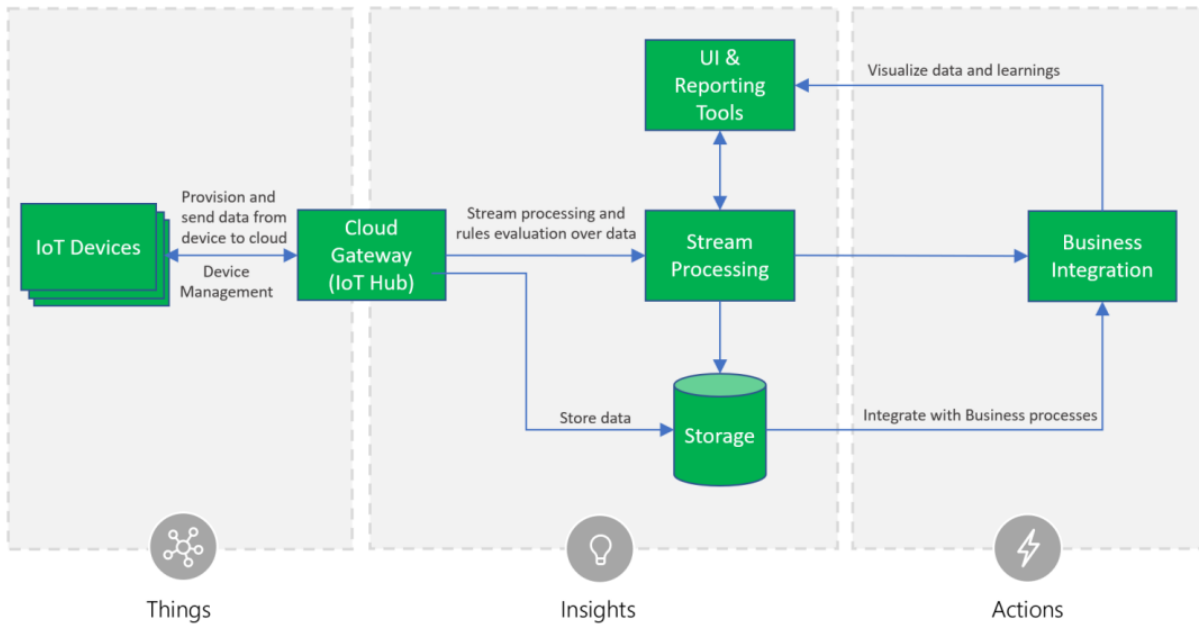
## Azure IoT

Microsoft Azure IoT je ucelená sada komponent, pomocí kterých je možné řešit jednotlivé části zpracování a prezentace IoT dat. Při tvorbě datové platformy je možné začít jenom s několika komponentami a postupně doplňovat další dle potřeby. Technologie je provozována v datových centrech Microsoftu po celém světě, v Evropě jsou v současnosti k dispozici dvě lokality (Dublin a Amsterdam) a další dvě (v Německu) se připravují.

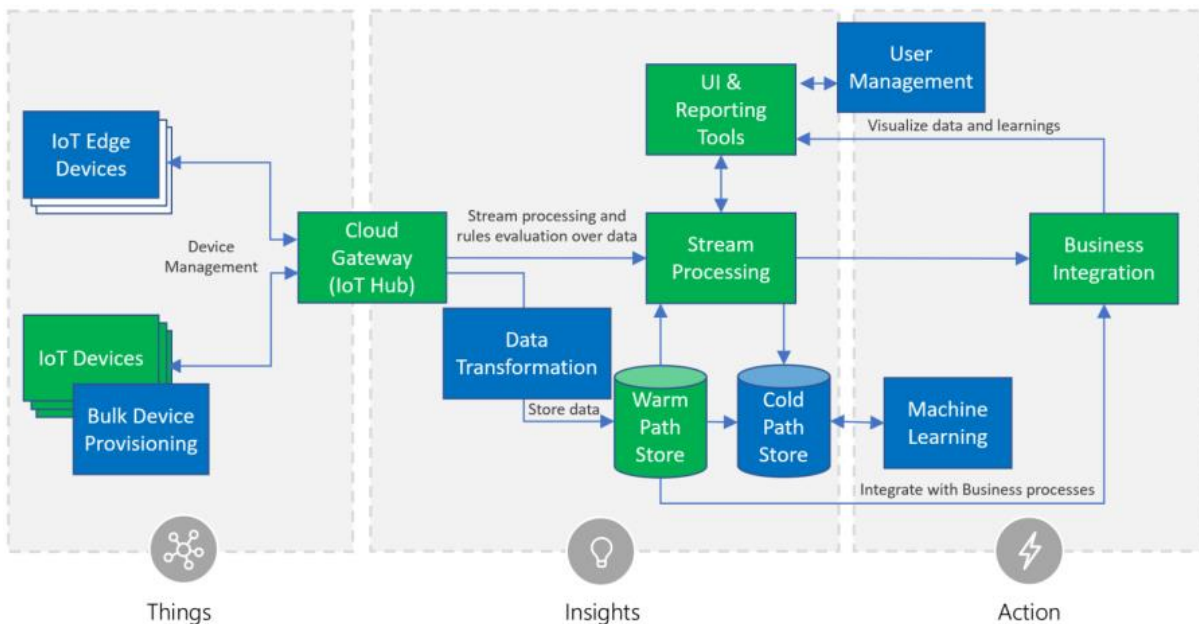
Základní procesní postup je následující:



Vzorová implementace pomocí základních komponent může pak vypadat následovně:



Postupně je možné doplňovat další komponenty (např. strojové učení) a architektura pak může vypadat následovně:



Zpracování dat, jejich uložení a následná prezentace by měla pokrývat požadavky na DPHM. V některých případech bude potřeba požadovanou vlastnost naprogramovat pomocí Azure Functions.

Komplikovanější je část příjmu dat ze zařízení a jejich ovládání. Azure IoT Hub vyžaduje specifický způsob komunikace, který současné ani plánované zdroje dat DPHM nepodporují. Aby bylo možné tyto zdroje dat připojit, je potřeba vytvořit gateway, která provede transformaci protokolů mezi zdrojem dat a IoT Hubem. V současnosti jsou obdobným způsobem připojeny některé zdroje dat k současnému řešení Datové platformy (Cisco Kinetics for Cities), přes aplikaci Moje Praha.

Jednotlivé komponenty mají různé modely účtování – podle objemu (počet zpráv, objem dat, počet uživatelů) a podle času (tj. typicky trvale aktivní) dedikovaného výkonu. Pro určení konkrétní ceny je potřeba znát množství dat z jednotlivých zdrojů dat a nechat zástupce MS vytvořit příslušnou cenovou kalkulaci.

#### Sumarizace

- ucelené řešení, nabízející zajímavé možnosti rozvoje
- vyřešení „režijních“ úkolů s možností soustředit se na obsah
- rychlejší nasazení a jednodušší provoz, dobře škálovatelné
- standardizované řešení, s možností podpory ze strany MS
- nutnost provozu v MS cloudu, cenová kalkulace je složitější a není moc nepřehledná
- není 100 % volnost, někdy je potřeba se přizpůsobit (obdobně jako v případě CKC)
- nepodporuje potřebné zdroje dat a potřeba psát konverzní moduly
- kapacity pro vývoj budou pravděpodobně dražší, jelikož jich není mnoho

## Jiné (více či méně) kompaktní řešení

Ostatní velcí poskytovatelé cloudových služeb nabízejí také skupinu IoT služeb. Jedná se například o:

- Amazon Web Services IoT
- IBM Bluemix
- Google Cloud IoT

Tyto služby typicky pokrývají menší část celého procesu a jsou méně rozšířené než Azure IoT. Implementace by tam byla složitější, zejména z důvodu horší dostupnosti kapacit a znalostí. Jelikož tyto možnosti se blíží dále popsanému heterogennímu systému a nepřinášejí další výhody, nejsou v tomto dokumentu dále rozepisovány.

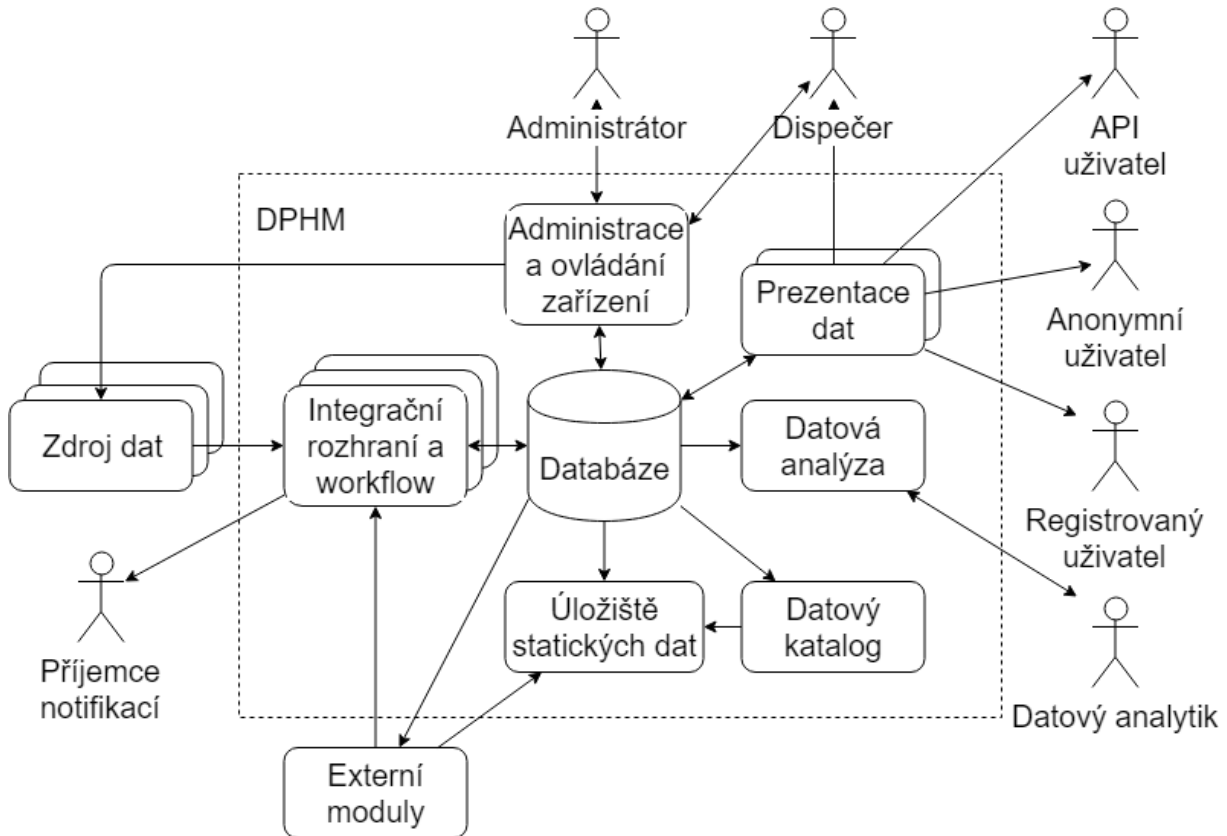
## Heterogenní systém

Následující text popisuje použití kombinace různých technologií, jejichž propojením může být DPHM vybudována. Až na výjimky je možné pro každou oblast (integrační rozhraní, prezentace dat apod.) zvolit samostatné řešení, jsou tak varianty popsány pro každou z nich samostatně.

Výhodou této varianty je, že většinu z nabízených technologií je možné provozovat jako on-premise, tak i ve vybraném hostingovém centru. Také nabízí volný výběr jednotlivých komponent a k dispozici jsou i open-source technologie.

Její nevýhodou je složitější architektura, nutnost většího a složitějšího testování integrace jednotlivých komponent a výrazně vyšší nároky na správu (administrátoři OS, databází, jednotlivých aplikačních technologií, sítí atd.).

Architektura heterogenní varianty DPHM je následující:



### Databázové úložiště

Úložiště dat je nejkritičtější část celé DPHM, nejhůře se škáluje a případné problémy se můžou projevit ve všech procesech platformy. Proto je při výběru technologií začít výběrem databáze a až následně dalších komponent.

Datová platforma může využít několik typů databází:

- relační
- time series
- dokumentové

Pro různé typy dat nebo práce s nimi jsou vhodné různé přístupy. Pro evidenci zařízení, uživatelů, parametrů je vhodná relační databáze. Ta může sloužit i pro ukládání samotných sensorických dat, ne vždy je to ale ten nejefektivnější způsob. Pro ně (a další data, která se mění v čase) bývá vhodnější varianta time series databáze. Ty rozdělují uložené informace na dimenze (id zařízení, čas, umístění apod.), které jsou indexované a na měřené hodnoty, které indexované nejsou. Přístup k datům je tak lépe optimalizovaný a databáze nabízí také služby jako retence, agregace apod. Objektové databáze slouží k ukládání obecných dat. Jako standardní úložiště je využívá například CKC nebo (obvykle) Azure IoT.

### Relační databáze

Pro uložení základních dat je vhodné použít relační databázi. Ta může být volitelně použita i pro uložení sensorických dat, čímž se (alespoň na začátku) může zrychlit vývoj a nasazení. Zvažované technologie:

- (Azure) Sql Server
  - využito na některých jiných OICT projektech
  - robustní řešení s podporou MS
  - nutnost platit licence
- MySQL/MariaDB
  - zkušenosti z jiných OICT projektů
  - široká dostupnost vývojářů a administrátorů
  - open source
- PostgreSQL
  - široká dostupnost vývojářů a administrátorů
  - open source
  - lepší než v MySQL/MariaDB v oblasti GIS dotazů

### *Time series databáze*

Time series databáze není nutná pro provoz DPHM. Může zjednodušit a zrychlit práci s daty, je to ale další komponenta, kterou bude potřeba v řešení integrovat a spravovat. V případě, že OICT zvolí využití i time series databáze, zvažované technologie jsou:

- InfluxDB
  - asi nejčastější volba pro IoT platformy
  - open source, některá rozšíření (např. cluster) jsou ale placená
- Prometheus
  - lepší dotazovací jazyk
  - plně open source
- OpenTSDB
  - open source
  - postavena na Hadoop

Každá z výše uvedených databází má trochu odlišný způsob práce s daty. Pokud není na straně OICT preference pro konkrétní z nich (např. existující znalosti), je možné zvolit InfluxDB, která je obecně nejčastější volbou. Pro zajištění vysoké dostupnosti je možné zvolit placenou variantu InfluxEnterprise, která nemá veřejný ceník, cena se ale pohybuje v desítkách tisíc dolarů ročně.

### *Dokumentová databáze*

Současné ani zvažované zdroje dat nevyžadují použití dokumentové databáze. Pokud by se v budoucnu pro ni vybírala technologie, tak se jeví jako vhodná varianta MongoDB. Tato databáze se již v rámci OICT využívá.

### *Provoz databází*

Datové úložiště se fyzicky složitěji škáluje, zejména z pohledu výkonu. Vzhledem k nejasným požadavkům na budoucí množství dat a zátěž může být zajímavé zvolit databázi jako službu. Tu cloudy obvykle poskytují, např. Amazon má na výběr několik databází (PostgreSQL, MySQL, MariaDB, Oracle, SQL Server) a také specifickou cloudovou databázi Aurora. Azure nabízí také několik možností (SQL Server, MySQL, MariaDB, PostgreSQL). Způsob provozu databáze bude závislý na možnostech provozovatele infrastruktury, tedy zda bude vůbec možnost používat databáze jako službu (SaaS).



## Integrační rozhraní a workflow

Integrační rozhraní vystavuje API, přijímá data, provádí kontroly a další zpracování. Kromě samotného uložení dat může realizovat i složitější workflow, např. na základě podmínek zasílat alarmy apod.

Jednotlivé varianty je navzájem možné kombinovat, a tedy pro různé zdroje dat nebo workflow použít různé způsoby implementace. Zvažované technologie:

- Node-RED
  - open source
  - vizuální definice workflow, vývoj v JavaScriptu
  - postaveno na Node.js, který je v OICT využíván
  - běžně použito pro IoT, jsou tedy k dispozici znalosti i vývojáři
  - velice rychlé nasazení
  - při nasazení v clusteru vyžaduje složitější nasazování změn
- Total.js Flow
  - inspirován Node-RED – vizuální workflow, JavaScript, Node.js
  - open source, vytváří několik vývojářů z ČR/SR
  - není zatím moc rozšířen, ale může být zajímavá alternativa
  - integrace s prezentací dat Total.js Dashboard
- Eclipse Kura
  - open source
  - částečně vizuální definice, vývoj v Javě
  - robustní řešení pro IoT integraci
  - spíše průmyslové nasazení, není příliš rozšířené
- kompletně vlastní vývoj
  - obdobně jako je teď Moje Praha
  - možnost integrace libovolného protokolu
  - možnost vývoje pomocí jakékoliv technologie/jazyka
  - náročnější na vývoj a provoz

## Prezentace dat

V této oblasti jsou dashboardy a jiné předpřipravené způsoby zobrazení, s možností uživatelských úprav – filtrování apod. Pro různé typy uživatelů a/nebo dat je možné zvolit různé varianty a vzájemně je kombinovat. Zvažované technologie:

- Power BI web
  - použito v OICT
  - běžná technologie, na trhu jsou dostupní vývojáři s potřebnou znalostí
  - podpora ze strany MS
  - umožňuje veřejné vystavení vybraných dashboardů
- Grafana
  - oblíbený nástroj pro vizualizaci IoT dat
  - rychlá tvorba dashboardů s pomocí předpřipravených komponent
  - open source, zdokumentováno rozhraní pro tvorby vlastních komponent
  - postavena na JavaScriptu, jsou tedy k dispozici znalosti i vývojáři

- pro přístup k datům vyžaduje přihlášení
- Redash
  - dobrá podpora pro tvorbu databázových dotazů
  - rychlá tvorba dashboardů s pomocí předpřipravených komponent
  - open source
  - postaven na Pythonu
  - méně rozšířený, malá znalostní báze
- Total.js Dashboard
  - jednoduchá a rychlá tvorba dashboardů
  - open source
  - dobré propojení s integračním rozhraním Total.js Flow
  - zatím neobsahuje některé složitější nebo méně obvyklé panely (widgety)
- thinger.io
  - jednoduchá datová platforma, která pokrývá integraci i prezentaci dat
  - open source licence, část kódu ale zatím není veřejně dostupný
- vlastní vývoj
  - HTML5 webová aplikace
  - pro frontend je možné použití framework jako základ (např. PatternFly)
  - backend může být použita libovolná technologie, dle dostupných kapacit (OICT běžně využívá PHP)
  - neomezené možnosti
  - časově náročnější

## Datová analýza

Do této oblasti patří různé ad-hoc dotazy a práce s daty mimo předdefinované výstupy. Zvažované technologie:

- Power BI Desktop
  - silný a prověřený nástroj
  - podporuje jazyk R a Python
  - existující znalost v rámci OICT
  - Microsoft dodává zdarma
- Externí moduly pro zpracování dat
  - přístup k datům přes API nebo (výjimečně) přímo z databáze
  - implementace specifických algoritmů
  - nutnost koordinace rozhraní (API, CSV struktura apod.) mezi DPHM a vývojem modulu
- vlastní uživatelské nástroje, pracující s exportem dat
  - postačuje nástroj, který dokáže pracovat s lokálními dat v souboru
  - bez technické závislosti na zbytku DPHM
  - každý uživatel může zvolit svoji technologii – R , Excel, různé uživatelské skripty

## Administrace a ovládání zařízení

Do této oblasti patří evidence zařízení v DPHM a zasílání příkazů k sensorům nebo obecně zdrojům dat. Vzhledem k tomu, že zatím nejsou známy zařízení/funkce, které by měla DPHM ovládat, je problematické

zvolit konkrétní řešení. Dle uživatelských požadavků by do této oblasti mělo spadat jenom uživatelské ovládání (např. zapnout/vypnout osvětlení). Technická konfigurace zařízení (různé interní parametry) by měla být řešena v systémech dodavatele zařízení.

Vhodné řešení závisí i na zvolené technologii pro Integrovaný rozhraní, jelikož její workflow je schopno pokrýt část potřebné funkcionality. To je pak možné doplnit o jednoduché formuláře (např. samostatné „statické“ webové stránky). Ty by se odesílaly do workflow v integrovaném rozhraní a z toho následně k zařízením. Například Node-RED platforma nabízí i tvorbu a distribuci jednoduchých formulářů a dashboardů, které by splňovaly požadavky na běžnou konfiguraci zařízení.

## Úložiště statických dat

Pro úložiště statických dat může být použit webový a/nebo FTP server, který bude přímo nabízet jednotlivé vygenerované soubory. Dle požadavků se nepočítá s žádnou přímou uživatelskou nadstavbou pro procházení uložených dat, tuto funkci zajistí Open data katalog, nebo komponenta nabízející tato uložená data.

Pokud nebude možné využít webový server některé s ostatních komponent, je možné zvolit z následujících možností:

- Apache
  - asi nejčastěji nasazovaný open source web server
- Nginx
  - modernější alternativa k Apache
- ProFTPD
  - FTP server, pokud by byl pro některé konzumenty vhodnější než web server

## Společné komponenty

Technologie v této kapitole jsou společné pro ucelené řešení (např. Azure IoT) i heterogenní systém. Jedná se o pokrytí specifických oblastí, které nejsou nabízeny v rámci IoT řešení.

### OLAP

Samostatnou oblastí datové analýzy je možnost práce s daty jako s OLAP kostkou (dle zadání „podobně jako Monitor Státní pokladny“). Pro varianty byly hledány open source technologie, aby byla zajištěna možnost úprav ze strany OICT. Komerčních OLAP řešení je několik, integrace by ale byla složitější a také by se muselo řešit licencování. Zvažované technologie:

- Saiku Analytics
  - využito pro Monitor Státní pokladny
  - open source
  - backend napsán v jazyce Java, frontend v JavaScriptu
  - jednoduché nasazení
- Apache Kylin
  - open source
  - backend napsán v jazyce Java, frontend v JavaScriptu
  - více možnosti vizualizace dat

## Open data katalog

Pro tuto oblast je potřeba data(set) management system (platform). Specifickým požadavkem je evidovat jak „privátní“ datové zdroje DPHM, tak i open data. Následně je nutná replikace dat do NKOD. Zvažované technologie:

- CKAN
  - využívá se v současnosti v rámci hl. m. Prahy
  - de-facto standard pro open data katalogy
  - open source, je možné jej upravit dle potřeb OICT
  - napsán v jazyce Python
  - již obsahuje potřebné integrace
  - pravděpodobně by mohl být zachován i režim provozu
- Dataverse
  - open source
  - napsán v jazyce Java, interně pracuje s databází PostgreSQL
  - dlouhodobě rozvíjen
  - nutnost vyvinout integraci na NKOD
  - spíše zaměřený na „Research Data Management“
- DKAN
  - kombinace CKAN funkcionality s CMS
  - open source
  - postaven na CMS Drupal, v jazyce PHP
  - nutnost vyvinout integraci na NKOD

## Provoz

### Režim provozu

Pro provoz DPHM je možné zvolit různé varianty (nebo jejich kombinace):

- on-premise/server hosting
- virtuální servery v hostingu (VPS nebo Docker/Kubernetes)
- cloud (SaaS – software as a service)

Na základě požadavků OICT je preferovaná varianta server hostingu v městském datovém centru v KCP. To zejména kvůli tomu, že se jedná o službu města a je tak logické, aby Datová platforma HMP byla provozována v datovém centru HMP. Nicméně vzhledem k nejasným požadavkům na budoucí množství dat a zatížení a potřebě průběžně reagovat na nové zdroje dat se zatím tato varianta jeví z pohledu DPHM jako nejméně vhodná. Kvůli době potřebné na pořízení nového HW v průběhu provozu bude mít DPHM poddimenzovaný (roste rychleji, než se předpokládá) nebo předimenzovaný (roste pomaleji, než se předpokládá) HW. Je možné, že do budoucna se ukáže jako výhodnější navázat provoz DPHM na některý z dalších systémů OICT (například Multikanálový odbavovací systém).

V případě volby server hostingu také hrozí zastarávání technologií. Průběžná obnova bude komplikovaná z důvodu nutné migrace jednotlivých komponent během provozu. To vše bude vytvářet zvýšené požadavky na lidské zdroje na straně OICT.

Pokud bude zvolena varianta Azure IoT nebo jiné ucelené cloudové řešení, pak je režim provozu určen. Pro heterogenní systém je nejvhodnější varianta VPS, která umožňuje operativně škálovat dostupný HW. Pokud by VPS/Docker služby nabízelo městské datové centrum, mohlo by se jednat dobrou alternativu.

### Další parametry

V případě server hostingu nebo VPS je potřeba určit další parametry. Výběr variant bude zejména záviset na dostupných kapacitách OICT.

#### Operační systém

- Linux
  - OICT běžně využívá Debian
  - zdarma
  - možnost provozování kontejnerů pro jednodušší nasazování – Docker/Kubernetes
- Windows Server
  - OICT využívá v rámci některých projektů
  - nutno platit licence

### Vysoká dostupnost

Pro jednotlivé komponenty bude potřeba zajistit vysokou dostupnost, typicky pomocí clusteru. V případě databáze to není úplně triviální, proto může být v tomto případě cloud/SaaS varianta výhodnější. Ta navíc nabízí zajištěné i tam zálohování apod. U aplikační vrstvy se typicky bude jednat o několik nezávislých instancí s load balancingem a vypracovaným postupem nasazování nových verzí (aby všechny instance měly stejný kód).

### Administrace

Různé varianty mají různé požadavky na administrátory, jak při instalaci, tak při provozu. Nejmenší jsou při cloudovém SaaS modelu, největší při použití server hostingu. Zejména v případě server hostingu bude potřeba zajistit 24x7 dohled, zálohování, síťovou bezpečnost (např. ochrana přes DDoS) a další aktivity, které by v případě ostatních variant nebyly nutné nebo by výrazně omezené. Tyto aktivity vyžadují dostatek lidských zdrojů na straně OICT.

# Technický popis zvoleného řešení

## Úvod

V rámci posouzení jednotlivých variant byl ze strany OICT jednoznačně zvolen heterogenní systém s tím, že bude preferován vlastní vývoj DP, s využitím jednotlivých samostatných komponent. Tento přístup nejlépe umožní splnění funkčních požadavků a zároveň bude dostatečně flexibilní při doplňování nových datových zdrojů a zpracování a zobrazování dat.

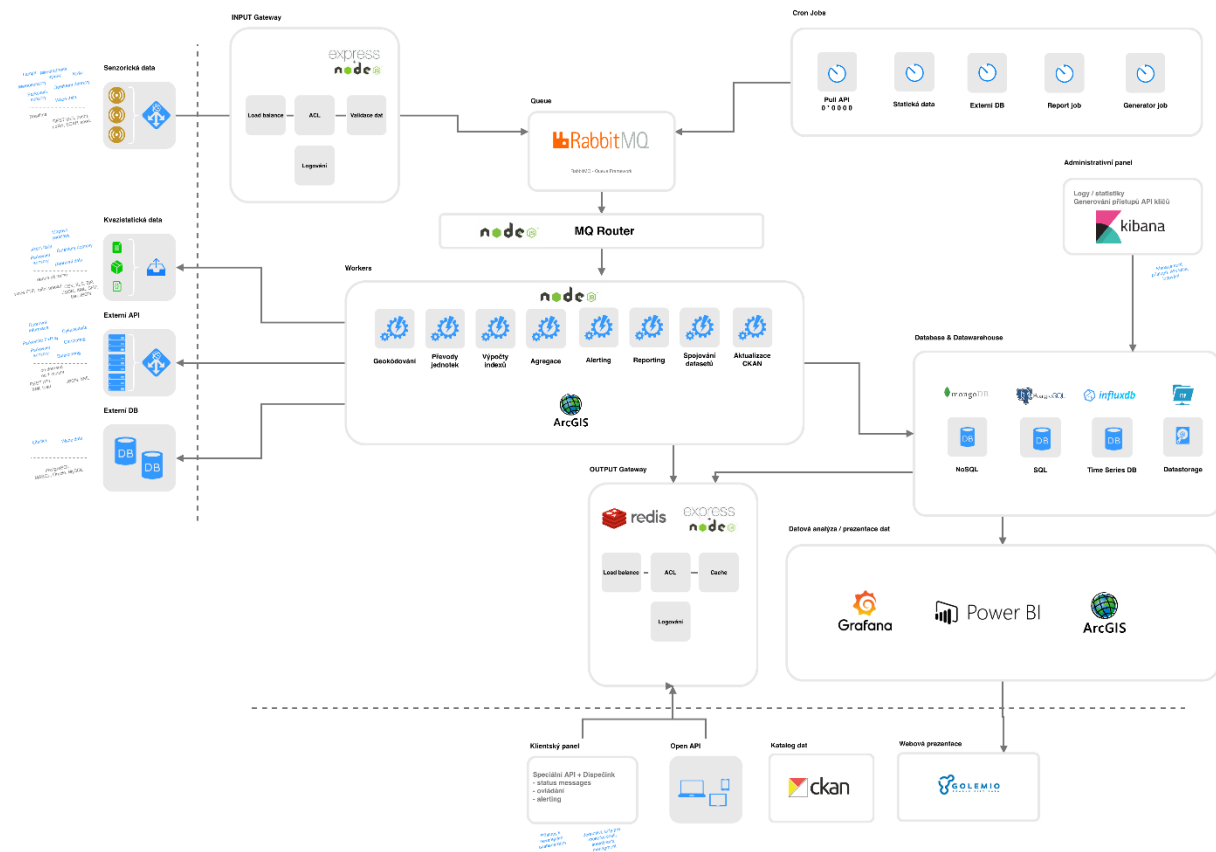
V některých případech zatím nebyla vybrána konkrétní technologie. V těchto případech se ale jedná o oblasti, které nejsou kritické z pohledu vývoje DP a nemají žádný zásadní dopad do harmonogramu ani náročnosti vývoje.

## Architektura a technologie

Pro jednotlivé komponenty byly v rámci posouzení zvoleny technologie a většina z nich byla následně ověřena implementací omezené sady funkčních požadavků a několika zdrojů dat. Zároveň došlo k rozdělení komponenty Integrovaní rozhraní a workflow na několik menších vzájemně provázaných komponent.

Vybrané technologie jsou typicky open source. V ostatních případech se jedná o technologie, které jsou k dispozici zdarma, ať již obecně nebo v rámci OICT nebo organizací podřízených MHMP.

Následující obrázek zachycuje jednotlivé komponenty a jejich použité technologie:





## Databázové úložiště

Pro uložení dat bylo zvoleno více technologií, které se vzájemně doplňují:

- MongoDB
- PostgreSQL
- InfluxDB

### Dokumentová databáze

MongoDB bude sloužit k uložení většiny senzorických dat a případně i dalších dat zpracovávaných v DP. Hlavním důvodem pro volbu této databáze je jednoduchá integrace s technologií pro Integrovaný rozhraní a workflow. Většina vstupních dat je ve formátu JSON a také jsou/budou typicky poskytovány jako JSON, což je i formát dokumentů uložených v MongoDB. Zpracování dat tak bude typicky sestávat z JSON transformací.

*Pro práci s daty v MongoDB z Node.js bude použita knihovna Mongoose.*

### Relační databáze

PostgreSQL bude jako relační databáze sloužit zejména pro uložení režijních dat DP, např. pro účely autentizace a autorizace nebo další data, u kterých bude vhodné pracovat s jejich relační strukturou, tj. vazbami mezi sebou. Tato databáze je běžně využívána v rámci open-source projektů, a tudíž je vhodnou variantou i pro relační databázi DP.

*Pro práci s daty v PostgreSQL z Node.js bude použita knihovna Sequelize.*

### Time series databáze

Pro vybraná senzorická data bude sloužit time series databáze InfluxDB. S jejím využitím se zatím, pro první zdroje dat, nepočítá, byla zvolena pro situace, kde z důvodu množství dat nebo požadavků na jejich zpracování a/nebo zpřístupnění bude vhodnější zvolit time series databázi. Time series databáze se zatím v rámci OICT nevyužívají, InfluxDB byla vybrána z toho důvodu, že je obvyklou volbou pro uložení časových řad ze senzorické sítě a existuje pro ni podpora v rámci ostatních vybraných technologií DP.

*Pro práci s daty v InfluxDB z Node.js bude použita knihovna node-influx.*

### Úložiště vstupních dat

Do budoucna se počítá také s ukládáním kopie vstupních dat do dočasného úložiště, bez dalšího zpracování. Toto úložiště může sloužit k případnému opětovnému zpracování starších vstupních dat, pokud by se zjistilo, že ve zpracování dat byla chyba a je nutné data zpracovat znova. Uložená data by se po určité době mazala, s ohledem na omezenou kapacitu tohoto úložiště.

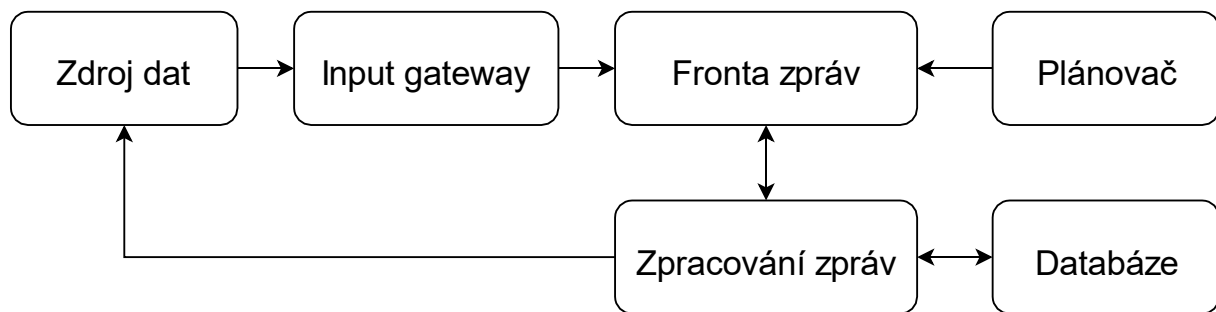
V rámci analýzy zatím nebyl určen způsob uložení, doba uložení ani požadavky na toto úložiště. Jednou z možností je ukládat zprávy do samostatných souborů, zejména pokud se bude jednat o JSON data.

## Integrační rozhraní a workflow

Proces v rámci integračního rozhraní a workflow byl rozdělen na několik částí, přičemž každá z nich bude implementována samostatně. Celý proces bude řízen pomocí zpráv uložených v několika frontách, které jsou průběžně plněné novými úkoly na straně jedné, a zpracovávány workery na straně druhé.

Komponenty, u kterých bude dále v rámci této kapitoly uvedeno, že budou vyvinuty v rámci OICT budou využívat technologii Node.js, s jazykem TypeScript. Tato technologie je již v rámci OICT využita v rámci jiných projektů a tým DP s ní má dobré zkušenosti.

Navrhovaný proces zpracování dat:



### Fronty zpráv

Pro implementaci front byla zvolena technologie RabbitMQ. Jedná se o ověřenou technologii pro fronty zpráv a integrace s ní je podporována ostatními vybranými technologiemi.

Každý typ konzumenta zpráv bude připojen k samostatné frontě. Producent zprávy bude určovat tzv. routing key, pomocí kterého bude následně zpráva distribuována do příslušných front a podle toho bude zvolena akce, která má být se zprávou provedena.

Pro komunikaci s RabbitMQ z Node.js bude použita knihovna amqplib.

Pro horizontální škálování nabízí RabbitMQ clustering, kdy jsou data replikována mezi více uzly. Všechny uzly jsou rovnocenné při distribuci zpráv, s výjimkou použití zrcadlení (viz níže).

Pro zajištění vysoké dostupnosti nabízí RabbitMQ zrcadlení front. V takovém případě je jeden uzel master a ostatní zrcadla, přičemž každá fronta má svůj vlastní master. Všechny operace s frontou jsou nejdříve aplikovány na master a následně propagovány na zrcadla. Klienti pracují vždy s masterem a v okamžiku jeho nedostupnosti převezme jeho funkci jedno ze zrcadel.

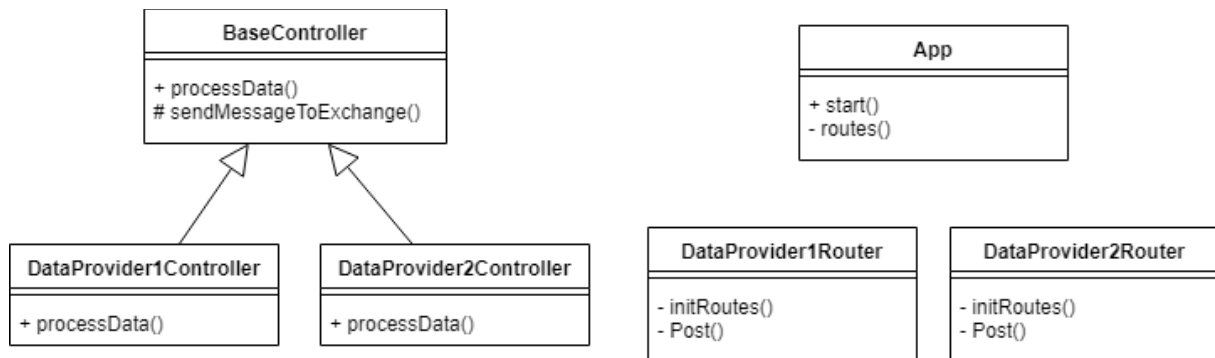
### Input gateway

Input gateway zajistí přijetí dat ze zdroje a po základní kontrole uloží data do fronty pro další zpracování. Jakmile fronta zprávu přijme jsou data považována za persistentně uložena v DP a potvrzena zdroji dat. Tato komponenta bude vyvinuta v rámci OICT.

Jelikož v rámci této komponenty bude provedena jenom základní kontrola a uložení do fronty, očekává se, že celá tato akce proběhne rychle a bude tak možné rychle potvrdit data jeho zdroji. Zároveň by tato akce neměla vyžadovat velký výpočetní výkon a je velice dobře horizontálně i vertikálně škálovatelná.

Jelikož je tato komponenta bezstavová, je možné její vysokou dostupnost zajistit pomocí více instancí, bez další synchronizace.

Navrhovaný diagram tříd (objektový model):



### Cron jobs

V některých případech se data nezpracovávají v okamžiku jejich přijetí ze zdroje dat, ale až později, typicky v nějakém pravidelném časovém intervalu. Pro tyto účely bude sloužit tato komponenta, která zajistí vložení řídicí zprávy do fronty pro další zpracování.

Tato komponenta bude vyvinuta v rámci OICT. Komponenta bude řízena konfiguračním souborem ve formátu cron souborů (minuta, hodina, den v měsíci, měsíc, den v týdnu, akce). Tento soubor bude po modifikaci následně zpracován Python skriptem, který vytvoří aktualizovaný konfigurační soubor pro systémový cron. Fakticky bude tedy systémový cron plánovač spouštět jednoduché předpřipravené akce pro zařazení zprávy do fronty.

### Workers

Konzumenti zpráv z jednotlivých front, budou tzv. workeri. Jedná se o funkce v Node.js, které budou přihlášeny k odběru zpráv ze svých front. V rámci škálování bude připraveno více instancí stejného konzumenta a RabbitMQ zajistí, že každá zpráva bude doručena právě jednomu workeru. Úkoly workerů mohou být různé, typicky se bude jednat o:

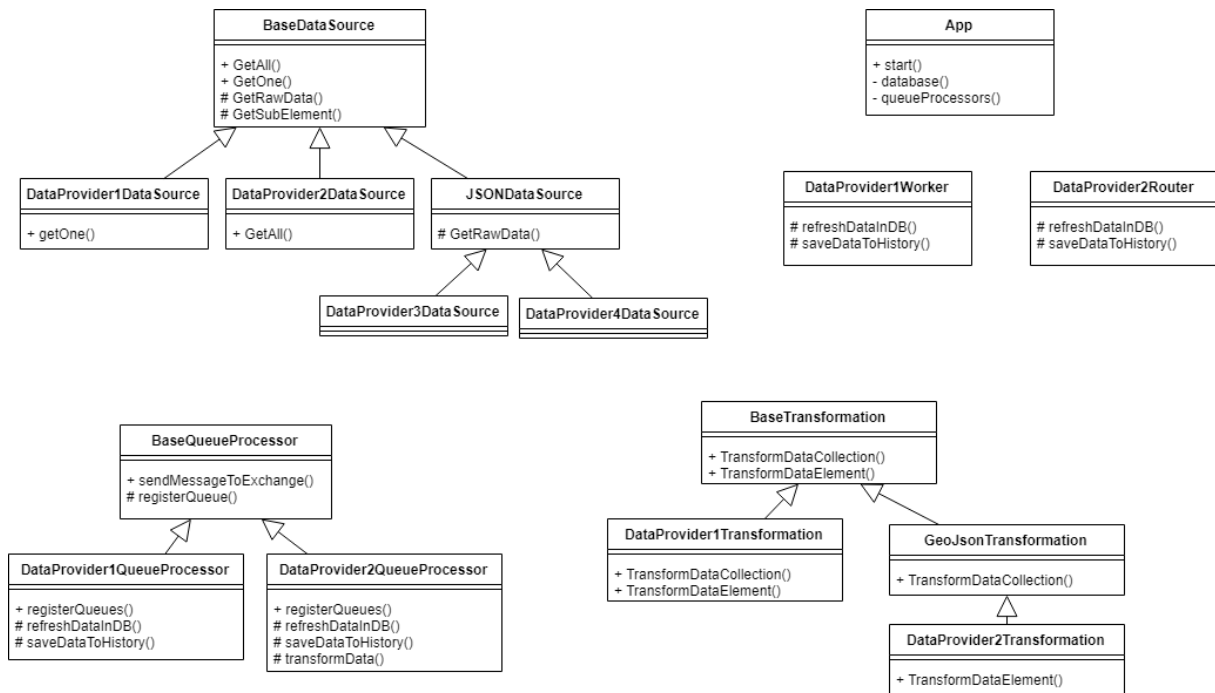
- zpracování dat ve zprávě a vytvoření nové zprávy s transformovaným obsahem
- uložení dat ze zprávy do databáze
- kontrola dat ve zprávě nebo v databázi a zaslání notifikace, pokud budou splněny podmínky

Celé zpracování příchozí zprávy tak bude rozděleno na několik kroků – prvotní kontrola, transformace, uložení do databáze, kontrola notifikací a případně další kroky – a každý krok bude realizován samostatným workerem na základě samostatné zprávy ve frontě. Cílem tohoto rozhodnutí bylo:

- možnost sdílení kódu (např. transformace) mezi více datovými zdroji
- lepší škálovatelnost celého procesu, zejména pokud jsou jednotlivé kroky různě časově a výpočetně náročné
- lepší diagnostika problémů ve zpracování, kde již přímo podle fronty bude vidět, která část DP má problémy

V případě, že zdroj dat bude do DP připojen v pull režimu, bude první worker v procesu získávat data voláním API zdroje dat.

Navrhovaný diagram tříd (objektový model):



## Notifikace

Pro realizaci notifikací zatím nebyla zvolena technologie. Prvotní vlastní implementace DP bude řešit vyhodnocování podmínek a následně zapsání případné notifikace do databáze. Pro převedení požadavku na notifikaci ve skutečné zaslání e-mailu, SMS, Slack zprávy apod. by bylo možné zvolit některé z existujících řešení nebo knihoven.

Vyhodnocování bude realizováno pomocí workerů. Pokud proces zpracování příchozích dat bude vyžadovat kontrolu dat a případné generování notifikace, vloží worker pro transformaci dat do fronty kromě požadavku na uložení do databáze také požadavek na provedení kontrol. Ty následně realizuje samostatný worker. Obdobně budou realizovány kontroly navázané na čas (např. využití klouzavého okna) – cron do fronty vloží požadavek na provedení kontrol, který zpracuje další worker.

Alternativou může být použití technologie Kapacitor, která je ale navázaná na použití InfluxDB.

## Prezentace dat

Data uložená v DP budou uživatelům zpřístupněna pomocí následujících cest:

- Klientský panel – rozcestník pro přístup k vizualizacím
- Vizualizace – dashboardy
- API – externí přístup k datům

Při zpřístupnění dat uživatelům je také potřeba vyřešit autorizaci k jednotlivým datovým sadám a také řízení zátěže.

## Klientský panel

Základním rozcestníkem při uživatelské práci s DP bude Klientský panel. Bude sloužit k autentizaci uživatelů a zobrazení odkazů na jednotlivé vizualizace, případně tyto vizualizace přímo zobrazovat (pomocí „vnoření“ obsahu z vizualizačního nástroje).

Pro implementaci byla zvolena technologie (JavaScript knihovny) React a Redux.

Zatím ještě nebyla vybrána technologie pro implementaci, bude se ale pravděpodobně jednat o některý JavaScriptový framework.

## Vizualizace dat

Pro vizualizaci dat bude použita kombinace nástrojů Grafana a PowerBI. První technologie byla vybrána pro jednoduchost a rychlost implementace dashboardů, ta druhá z důvodu rozšířených analytických nástrojů.

Prozatímni implementace dashboardů pracuje s přímým přístupem do databáze. Toto řešení je možné použít pro open data a data, kde je řízení přístupu řešeno na úrovni celé datové sady. Použité technologie ale neumožňují přístup na úrovni jednotlivých záznamů (řádků v databázi). V takovém případě bude potřeba upravit zdrojový kód Grafany pro možnost aplikovat filtrování dat na základě přihlášeného uživatele.

Z pohledu bezpečnosti dat ale bude do budoucna vhodné propojit Grafanu s výstupním API, které zajistí izolaci zobrazení od databáze a sjednotí řízení přístupu pro vizualizaci dat i výstupní API. Současná verze Grafany totiž zpřístupňuje SQL rozhraní k databázi a umožňuje případnému útočníkovi změnit (kromě jiného) filtrovací podmínky.

Práce s PowerBI zatím nebyla v rámci pilotní implementace ověřena.

Na mapové vizualizace bude použit ArcGIS, zejména díky možnosti použití licencí, které vlastní MHMP.

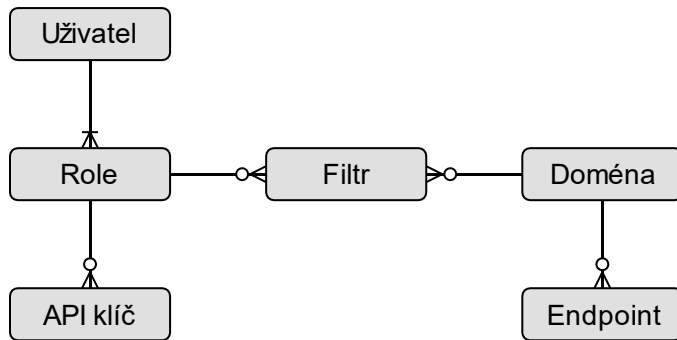
## Výstupní API

Zpřístupnění dat pro externí systému bude realizováno pomocí výstupního API. Typicky se bude jednat o open API, pro některé datové sady a uživatele se ale může jednat i o API s omezeným přístupem. Řízení přístupu ale nebude součástí této komponenty, ale bude vyčleněno do samostatné komponenty. Výstupní API bude vyvinuto v rámci OICT pomocí Node.js.

## Řízení přístupu k datům

Pro řízení přístupu k datům bude implementována tzv. permission proxy vrstva. API dotazy uživatelů nebudou směrovány přímo na výstupní API, ale budou zpracovány pomocí proxy vrstvy. Ta na základě rolí přidělených uživateli doplní filtrovací podmínky do zaslaného dotazu tak, aby výstupní API zpřístupnilo jenom tu část dat, ke které má daná role oprávnění. Uživatelské dotazy budou identifikovány pomocí API klíčů, které si budou uživatelé sami generovat v DP.

Definice oprávnění bude na základě kombinace role a business domény, do které příslušný endpoint spadá, viz následující obrázek:



Kromě kontroly oprávnění bude tato komponenta také vytvářet podklady pro statistiky využití jednotlivých služeb.

V rámci výběru technologií byly zvažovány existující technologie pro řízení přístupu. Jelikož ale nebyla nalezena žádná, která by splňovala (specifické) funkční požadavky, bylo rozhodnuto o vývoji v rámci OICT. Použita bude technologie Node.js pro proxy vrstvu a React pro GUI frontend.

### Řízení zátěže

Jednotlivé API klíče budou mít kromě oprávnění také definovány limity na počet dotazů v čase a na počet záznamů v rámci odpovědi na jeden dotaz.

Pro řízení zátěže jsou zvažovány tyto možnosti:

- Tyk
- Kong
- Express Gateway
- vlastní vývoj

Výše zmíněná existující řešení nabízejí také řízení přístupu, vzhledem k funkčním požadavkům je ale není možné pro tento účel využít.

Pro prvotní implementaci bude použita technologie Express Gateway. Pokud budoucnosti bude potřeba vyšší výkon, bude pravděpodobně použita technologie Kong. V případě vlastního vývoje bude také potřeba zvážit, jestli je Node.js (jinak použitý pro většinu vlastního vývoje DP) dostatečně výkonná technologie pro tento účel.

Jelikož přes tuto komponentu budou řešeny externí přístupy k datům, je možné ji použít i jako SW load balancer.

Součástí řízení zátěže by měla být také ochrana proti DDoS útokům. Jejich detekce a eliminace by se měla realizovat ve spolupráci s provozovatelem datového centra DP.

## Datová analýza

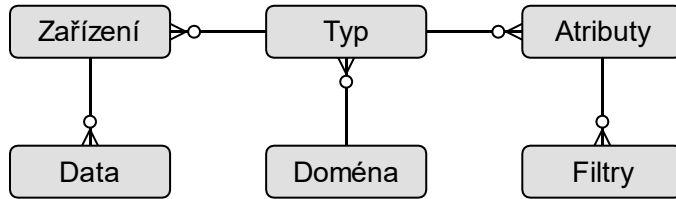
Pro implementaci datové analýzy zatím nebyla vybrána konkrétní technologie. Bude ale použit způsob, který zajistí připojení externích aplikací, ať již přímo k databázi nebo pomocí exportů dat. Tento způsob umožní použití nástroje R a mnoha dalších.



## Administrace a ovládání zařízení

Administrace a ovládání zařízení bude součástí Klientského panelu, budou tedy použity stejné technologie.

Administrace zařízení bude realizována v relační databázi. Návrh datové struktury je následující:



DP nebude udržovat aktuální nastavení zařízení. Pokud formulář pro ovládání zařízení bude vyžadovat zobrazení aktuálně nastavených parametrů, provede DP volání API zdroje dat a tyto informace získá v rámci zobrazení formuláře. Alternativně je možné použít aktuální data ze zařízení. Po odeslání formuláře provede DP volání API zdroje dat.

## Úložiště statických dat

Pro uložení statických dat, které vzniknou jako exporty z DP bude použit ProFTPD server. Exporty budou vznikat na základě požadavku, který vytvoří cron a následně zpracuje worker. Ten výsledný soubor uloží na sdílený souborový systém, kde bude přístupný přes FTP. Worker zároveň odešle do open data katalogu požadavek na vytvoření nového záznamu s odkazem na vytvořený soubor.

## OLAP

Pro práci s OLAP kostkou byla zvolena technologie Saiku Analytics. Do uživatelského rozhraní bude zakomponována jako jedna z částí Klientského panelu. V rámci implementace bude potřeba zajistit provoz samostatné backend aplikace napsané v jazyce Java. Součástí implementace bude i analýza dat a definice OLAP kostky.

## Open data katalog

Pro provoz open data katalogu bude použita technologie CKAN, tj. rozvoj stávající implementace katalogu. CKAN je implementován v jazyce Python. V rámci funkčních požadavků nejsou na jeho rozvoj kladeny velké požadavky, nemělo by se tedy jednat o významnější část implementace DP.

# Návrh infrastruktury

## Základní popis a rozdělení

Cílově bude DP provozována v datovém centru na několika VPS s operačním systémem Linux. Jednotlivé komponenty budou nasazovány pomocí technologie Docker. Zvolené řešení umožňuje operativně přesouvat jednotlivé komponenty mezi jednotlivými VPS a zajistí tak jednu z forem horizontálního škálování. Zároveň platí, že zvolené řešení umožňuje vertikální škálování a pro většinu komponent také horizontální škálování.

V rámci vývoje a testování je možné všechny Docker kontejnery provozovat na jednom VPS a následně je podle možností a potřeb rozdělovat mezi samostatné VPS.

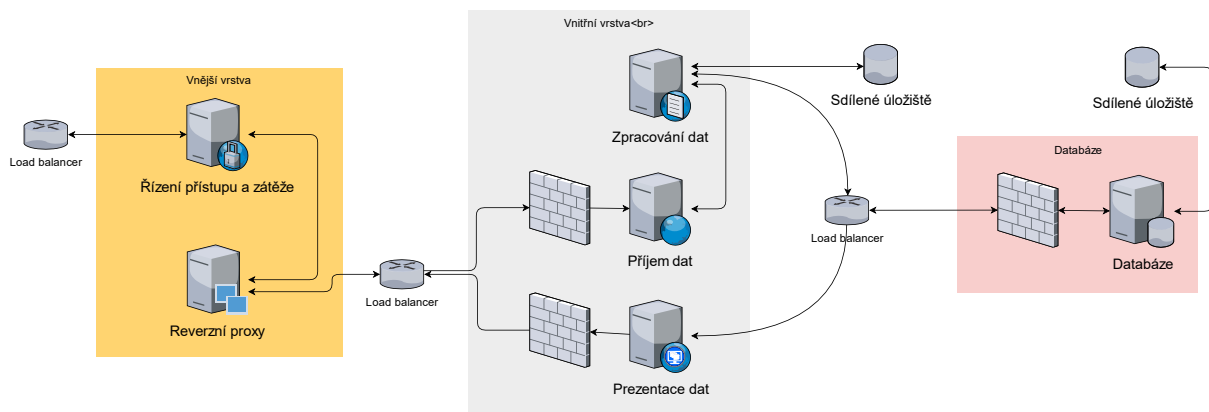
Z pohledu bezpečnosti a omezení případných dopadů výkonnostních problémů je vhodné jednotlivé komponenty rozdělit alespoň mezi tyto VPS:

- vnější vrstva
  - řízení přístupu a zátěže
  - reverzní proxy pro vnitřní vrstvu
- vnitřní vrstva
  - příjem dat – input gateway a fronty zpráv
  - zpracování dat – workers, cron jobs, notifikace
  - prezentace dat – výstupní API, clientský panel, vizualizace dat, OLAP, datová analýza, administrace a ovládání zařízení
- databáze

Výše uvedené rozdělení je seřazeno podle důležitosti provozu – příjem dat je více kritická část DP než prezentace dat. Také jsou v řazení zohledněny závislosti – pro plnohodnotnou funkčnost daného VPS musí být k dispozici předchozí VPS, nejsou ale nutné VPS následující.

Z důvodu zajištění vysoké dostupnosti by měly být pro každý typ VPS použity alespoň dvě instance. Mezi jednotlivými VPS bude použit samostatný load balancer (přístup k vnější vrstvě) nebo bude tato funkcionality zajištěna aplikačně (vnitřní komunikace DP).

Na následujícím obrázku je zachycen návrh infrastruktury:



## Požadavky na zdroje

### Disková kapacita

Současná DP obsahuje dle údajů OICT cca. 21 GB dat, s denním přírůstkem cca. 150 MB nových dat přijatých v rámci cca. 21.000 požadavků.

Při zachování současného množství nových dat je tedy odhadovaný roční přírůstek cca. 50 GB nových dat. Vzhledem k plánu zapojovat do DP nové datové zdroje bude dále v textu počítáno s dvojnásobným přírůstkem, tj. 100 GB nových dat ročně. Toto číslo je jenom odhad, navíc bez znalosti požadavků nových datových zdrojů a může se rapidně změnit. Například díky vzniku nového projektu může být přírůstek dat 100násobně vyšší, než je v současnou chvíli.

Dále bude potřeba uchovávat nezpracovaná vstupní data, kde v případě uložení po dobu 6 měsíců bude potřeba dalších 150 GB.

Některé datové zdroje zpracovávají výrazně větší množství dat, než které se následně uloží. Např. v případě poloh autobusů se zpracovává 3 GB denně, tato data se ale během následujících několika dnů smažou. Z pohledu databáze se tedy nejedná o velké množství dat, dočasné úložiště vstupních dat ale bude vyžadovat dalších 700 GB navíc.

Pro exporty dat, na které bude odkazovat open data katalog bude potřeba 100 GB.

Objem režijních dat (seznamy zařízení, uživatelů, konfigurace apod.) bude v porovnání s výše uvedenými objemy dat zanedbatelný. Obdobně i požadavky systémové úložiště (operační systém, aplikace) není v porovnání s objemem uložených dat významný.

### CPU a RAM

Odhad potřebného výkonu DP není možné odvodit od zkušeností ze současné DP. Neočekává se, že by DP zpracovávala data real-time, tj. že by uživatelé očekávali okamžité zpracování a zobrazení nově příchozích dat. Proto je pro základ dimenzování zátěže možné použít průměrné zatížení. Většina sensorických dat je také časově rovnoměrně rozložena.

Na základě této úvahy je tedy možné počítat s několika desítkami požadavků za minutu, každý z nich v průběhu o velikosti několika desítek kB. Pro příjem a zpracování takového objemu dat by mělo postačovat několik jader běžného procesoru a jednotky (resp. malé desítky po započtení režie) GB paměti.

### Datový tok

Pokud datová centra účtují za datový tok, tak typicky jenom datový tok z/do datového centra, ale ne datový tok mezi jednotlivými VPS. V dalším textu tak bude provedena kalkulace jenom těchto dvou datových toků. Některá datová centra navíc jeden nebo i oba směry účtují paušálně, podle kapacity linky, a ne podle reálně přenesených dat.

Pro příchozí datový tok je použít objem zpracovaných dat. Požadavky uživatelů na prezentaci dat budou tvořit jenom malou část příchozího datového toku. Pokud vyjdeme z předpokladu, že velikost příchozí zprávy není významně vyšší než velikost uložených dat po jejím zpracování, měl by být příchozí datový tok do 10 GB denně.

Odchozí datový tok je závislý zejména na počtu uživatelů, intenzitě jejich práce a množství zobrazených dat. Tady zatím není k dispozici dostatek informací pro stanovení odhadu. Pro zatím zvažovaná datacentra to ale není důležité, jelikož mají paušální platbu za připojení.

## Odhad finančních nákladů

Tabulka níže obsahuje vzorový odhad měsíčních nákladů po 1 roku provozu DP, s využitím veřejných cen vybraného datového centra v ČR. Reálné náklady se budou v čase měnit podle toho, jak se bude zvyšovat objem uložených dat a jak bude vzrůstat počet datových zdrojů a uživatelů. Počet VPS, množství přidělené RAM a CPU je možné typicky v datových centrech operativně měnit. Velikost přiděleného diskového prostoru se někdy mění obtížněji, tato operace může někdy vyžadovat delší odstávku (tj. delší než jednotky minut).

Pro vzorovou kalkulaci byla zvolena služba Forpsi Cloud Pro. Nabízí datové centrum v ČR s možností jednoduché správy připojených VPS a také nabízí služby virtuální privátní sítě, load balancingu, sdíleného datového úložiště a 1 Gbit síťové adaptéry pro VPS.

Položka	CPU dedikovaných jader	RAM	disková kapacita	Cena	Počet	Celkem
<b>vnější vrstva</b>	2	8 GB	30 GB	1 404 Kč	2	2 808 Kč
<b>vnitřní vrstva</b>	3	8 GB	30 GB	1 692 Kč	3	5 076 Kč
<b>databáze</b>	4	12 GB	30 GB	2 340 Kč	2	4 680 Kč
<b>sdílené datové úložiště</b>			200 GB	3 125 Kč	1	3 125 Kč
<b>sdílené souborové úložiště</b>			1 TB	725 Kč	1	725 Kč
<b>virtuální síť</b>				180 Kč	2	360 Kč
<b>load balancer</b>				188 Kč	1	188 Kč
<b>připojení 1 Gbit</b>				v ceně		
				<b>Celkem za měsíc</b>		<b>16 962 Kč</b>

Všechny uvedené ceny jsou bez DPH. Jedná se o vzorovou kalkulaci, některé služby nejsou dostupné ve všech datových centrech poskytovatele.

# Organizační zajištění

## Specifikace požadavků na role

V následujícím textu jsou popsány jednotlivé projektové role potřebné pro implementaci DP.

### Analytik

Úlohou analytika bude:

- sběr uživatelských požadavků na zpracování a zobrazení jednotlivých datových sad
- analyzovat datové zdroje a definovat datové struktury DP a transformace
- definovat formát a struktury výstupních API
- práce s nástroji pro vizualizaci dat (Grafana, PowerBI apod.)

Jedná se tedy o kombinaci business a datového analytika. Dle dostupných kapacit je možné (a vhodné) oba tyto pohledy pokrýt jednou osobou. Rozdělení práce mezi více analytiků je vhodnější skrz různé datové oblasti – doprava, životní prostředí atd.

### Databázový specialista

Úlohou databázového specialistu bude:

- definice a údržba databázových modelů
- optimalizace dotazů

Tato role může být kombinována s rolí Vývojář backend.

### Vývojář backend

Úlohou backend vývojáře bude:

- implementovat jednotlivé části procesu
  - vstupní API
  - zpracování úkolů z vnitřních front DP
  - výstupní API
  - datová analýza
- implementovat společné části DP
  - správa front
  - řízení přístupu
  - plánování

Vzhledem k použitým technologiím je potřebná znalost:

- Node.js
- Typescript
- Mongoose

### Vývojář frontend

Úlohou frontend vývojáře bude:

- implementovat dashboardy
- implementovat „režijní“ GUI
  - správa přístupových oprávnění
  - ovládání zařízení a DP jako celku

Vzhledem k použitým technologiím je potřebná znalost:

- JavaScript
- React
- případně další technologie, které nebyly zatím ještě rozhodnuty a budou při vývoji použity

## Tester

Úlohou testera bude:

- spolupracovat s analytikem na zadání pro vývojáře
- testovat zpracování dat a transformace
- testovat splnění uživatelských požadavků na zpřístupnění/zobrazení dat

## Konfigurační řízení

Konfigurační řízení v rámci projektu bude zajišťovat:

- údržbu zdrojových kódů a dalších výstupů projektu
- údržbu prostředí – vývojové, testovací, produkční
- zajištění konzistentního obsahu jednotlivých nasazovaných verzí DP

Použité nástroje a projektové postupy nebyly zatím definovány. Je ale vhodné, aby tento člen týmu znal použité technologie a uměl je provozovat.

## Projektové řízení

Úlohou projektového manažera bude řídit projekt a sledovat jednotlivé parametry – čas, náklady a kvalitu.

## Odhad kapacit

V současnosti již existuje částečná implementace nové DP, a to zpracování některých zdrojů dat, včetně jejich zobrazení a poskytování přes výstupní rozhraní. Dále je již implementována významná část jádra DP (plánování, řízení přístupu apod.). Tento stávající stav můžeme označit jako 0. etapu. V rámci následné 1. etapy bude dokončena implementace jádra DP a následně zapojeny další datové zdroje – vybraných 45 zdrojů, které jsou již zapojeny do CKC a dále také cca. 30 nových datových zdrojů.

Následující tabulka obsahuje odhadované kapacity potřebné pro jednotlivé role pro 1. etapu.

Role	Počet čd
Analytik	90
Databázový specialista	30
Vývojář backend	210
Vývojář frontend	130
Tester	70

Konfigurační řízení	40
Projektové řízení	60
<b>Celkem</b>	<b>630</b>

Jelikož je cílem realizovat 1. etapu během 6 měsíců, je potřeba pro tuto dobu celkem cca. 5,5 FTE. Navíc je pro obě vývojářské role potřeba alespoň dvě osoby na každou z rolí.

## Harmonogram implementace

Následující tabulka obsahuje zjednodušený harmonogram. Při plánování projektu je v prvním kroku potřeba rozdělit kapacity vývojářů a k nim pak naplánovat ostatní projektové role.

Pořadí	Úkol	Začátek	Konec
<b>1.</b>	Jádro – dokončení	Týden 1	Týden 12
<b>1.a.</b>	Řízení přístupu a zátěže	Týden 1	Týden 6
<b>1.b.</b>	Kontrola notifikací	Týden 3	Týden 4
<b>1.c.</b>	Údržba front, ošetření chybových stavů	Týden 7	Týden 8
<b>1.d.</b>	Generalizace transformace dat	Týden 7	Týden 12
<b>1.e.</b>	CKAN integrace, experty dat	Týden 7	Týden 10
<b>1.f.</b>	Logování a sledování stavu DP	Týden 10	Týden 12
<b>2.</b>	Datové zdroje – existující	Týden 3	Týden 14
	<b>opakování následujících kroků pro každý datový zdroj</b>		
<b>2.a.</b>	Analýza datového zdroje	Den 1	Den 1
<b>2.b.</b>	Implementace zpracování dat	Den 2	Den 3
<b>2.c.</b>	Implementace prezentace dat	Den 2	Den 3
<b>2.d.</b>	Testování	Den 4	Den 4
<b>2.e.</b>	Nasazení	Den 5	Den 5
<b>3.</b>	Datové zdroje – nové	Týden 15	Týden 26
	<b>opakování následujících kroků pro každý datový zdroj</b>		
<b>3.a.</b>	Analýza datového zdroje	Den 1	Den 2
<b>3.b.</b>	Implementace zpracování dat	Den 3	Den 7
<b>3.c.</b>	Implementace prezentace dat	Den 3	Den 5
<b>3.d.</b>	Testování	Den 6	Den 8
<b>3.e.</b>	Nasazení	Den 9	Den 10
<b>4.</b>	Migrace dat	Týden 22	Týden 26

Poznámky:

- V 1. a 2. úkolu je rolí analytika zejména pokrýt rozdíly proti současnému stavu (CKC) a to zejména z uživatelského pohledu. V rámci 3. úkolu se bude jednat o standardní analýzu datových zdrojů stejně, jako při jejich zapojování do CKC.

- Jednotlivé kroky v rámci úkolů 2 a 3 jsou časově kratší (typicky 1-3 čd), postupně se ale opakují. Jakmile je daná oblast datového zdroje realizována, je možné pokračovat následujícím datovým zdrojem a kapacity jednotlivých rolí tak alokovat trvale.

## Odhad nákladů

Na základě dostupnosti kapacit OICT bylo odhadnuto množství kapacit, které bude potřeba pořídit externě. Jelikož implementace DP neobsahuje větší ucelené celky, které by bylo možné poptat na základě smlouvy o dílo, je nejvhodnější použití režimu body shop. Externí kapacity je tak možné operativně využívat na implementaci jednotlivých dílčích celků a zároveň přitom zajistit kontrolu nad celým procesem a vývojem. Tento postup také umožňuje jednodušší budoucí údržbu, protože kmenoví zaměstnanci OICT budou zapojeni do většího detailu implementace jednotlivých komponent DP.

Následující tabulka ukazuje předpokládaný počet ČD, které budou pokryty interními kapacitami OICT a kolik externími kapacitami. Výsledná cena se pak bude odvíjet zejména od vysoutěžené ceny za body shop.

Role	Počet čd	Kapacit interně	Kapacit externě
Analytik	90	90	0
Databázový specialista	30	20	10
Vývojář backend	240	100	140
Vývojář frontend	130	50	80
Tester	70	50	20
Konfigurační řízení	40	40	0
Projektové řízení	60	0	60
<b>Celkem</b>	<b>660</b>	<b>350</b>	<b>310</b>