

Návrh systémů IoT

1. Úvod do předmětu, internet věcí. Úvod do Pythonu.

Stanislav Vítek

Katedra radioelektroniky

České vysoké učení technické v Praze

Obsah přednášky

1. O předmětu
2. Internet věcí
3. Příklady IoT systémů
4. Příklad IoT zařízení
5. Architektura IoT systému
6. Programovací jazyk Python

O předmětu

<https://cw.fel.cvut.cz/wiki/courses/b0b37nsi>

Přednášející, cvičící

- Stanislav Vítek, viteks@fel.cvut.cz, <http://mmtg.fel.cvut.cz/personal/vitek/>
- Jan Hamacek, hamacjan@fel.cvut.cz
- Jakub Špicar, spicajak@fel.cvut.cz
- Lukáš Liebzet, liebzluk@fel.cvut.cz

Studijní výsledky, hodnocení

- **zápočet:** miniprojekty (max 60b + 10b bonus / min 30b)
- **zkouška:** semestrální práce (30b / 10b), prezentace (10b / 1b)

Co od předmětu čekat?

- Návrh jednoduchého IoT systému (from the scratch)
 - Webová aplikace, vizualizace dat
 - Ukládání dat do databáze (lokální i cloudové)
 - Zpracování dat (časové řady)
 - Připojení IoT zařízení pro sběr dat (drátově i bezdrátově)
- Nástroje
 - **SW:** Python, [MicroPython](#), [Flask](#), [pandas](#), [scikit-learn](#), git, docker
 - **HW:** RPi 4, RPi Pico W, periferie: DHT22, 10DOF, BLE

Struktura přednášek

- Tématické okruhy
- Časový plán: <https://cw.fel.cvut.cz/b242/courses/b0b37nsi/lectures/start>

1. Definice IoT, příklady. Úvod do Pythonu.

2. Pokračování v Pythonu. Technologie pro frontend.

- OOP v Pythonu
- CSS, HTML, MVC, templates, Jinja. Flask, FastAPI, Sanic, Tornado

3. Modely komunikace a komunikační rozhraní

- REST, GraphQL, gRPC, SOAP, MQTT, CoAP
- Serializace dat: JSON, XML, YAML, Protocol Buffers, XDR, MessagePack

4. Ukládání dat, databáze

- SQL, NoSQL, SQLite, SQLAlchemy

5. Zpracování časových řad

- Pandas

6. Git, CI/CD, Docker, deployment

7. Mikrokontroléry

- Obecný přehled, periferie, Raspberry Pi Pico, MicroPython, Wokwi
- Pokročilá práce, vícejádrový procesor, multitasking, komunikace

8. LPWAN sítě

- Bluetooth, Zigbee, Thread, Matter

9. Strojové učení

- Modelování, predikce, detekce anomálií, scikit

10. Cloud a virtualizace. Bezpečnost v IoT a distribuovaných systémech

Internet věcí

- Internet věcí (IoT) představuje propojení zařízení (věcí) / systémů s Internetem
- Klíčovou částí IoT jsou služby
- Propojení zařízení
 - by mělo být především bezdrátové
 - přináší nové možnosti vzájemné interakce nejen mezi jednotlivými systémy
 - umožňuje kontrolu, monitorování, zabezpečení a vytváření pokročilých služeb
- Termín vytvořil britský podnikatel Kevin Ashton (MIT Auto-ID Labs)
 - Označoval budoucí globální síť objektů propojených pomocí RFID
 - Úplná automatizace sběru dat
 - První článek o internetu věcí v roce 2004 z MIT nazvaný [Internet 0](#)

Související oblasti

- Vestavné (embedded) systémy
 - nejsou nezbytně připojeny k internetu
- Senzorové sítě
 - soubor senzorových zařízení propojených bezdrátovými kanály
- Kyberneticko-fyzikální systémy
 - interakce mezi fyzickými a kybernetickými systémy
- Systémy reálného času
 - důraz na časová omezení
- Pervasivní / všudypřítomná výpočetní technika

IoT systémy

- Propojení zařízení, systémů a služeb za účelem poskytnutí **dat**, které mohou být převedeny na **informace** a tyto informace na **znalosti**, které je možné užitečným způsobem využít
- Zařízení mohou
 - vyměňovat data s jiným připojeným zařízením nebo aplikací (přímo nebo nepřímo), nebo
 - sbírat data z jiných zařízení a zpracovávat je lokálně, nebo
 - posílat data do centrálních server nebo cloudových aplikací, nebo
 - realizovat některé úlohy lokálně a některé v rámci IoT architektury v závislosti na aktuálních podmínkách.

Cíle IoT systémů

Zjednodušení každodenního života

ClIoT, Consumer IoT - spotřebitelský internet věcí

- automatizace domácnosti, smart zařízení, nositelná elektronika (wearables)

Zefektivnění využití zdrojů

IIoT, Industrial IoT - průmyslový internet věcí

- vychází z M2M (machine to machine) komunikace, rozšíření o zpracování dat
- průmyslová automatizace, doprava, energetika, zdravotnictví
- snížení provozních nákladů, zvýšení produktivity a bezpečnosti pracovníků, předcházení výpadků pomocí monitoringu a včasné údržby

Příklady IoT systémů

Příklad 1: Smart City

- Chytré kontejnery
 - Senzory: ultrazvukový dálkoměr, váha
 - Odhad zaplnění kontejneru, optimalizace vyvážení
- Správa parkovišť
 - Senzory: ultrazvukový dálkoměr, UWB radar, světelné závory
 - Poskytování informací o možnostech parkování
- Monitorování dodržování hygienických norem
 - Senzory: PM_x, CO₂, CO, NO, NO₂, mikrofon, luxmetr
 - Dodržování limitů znečištění prachem, škodlivými látkami, hlukem, světlem

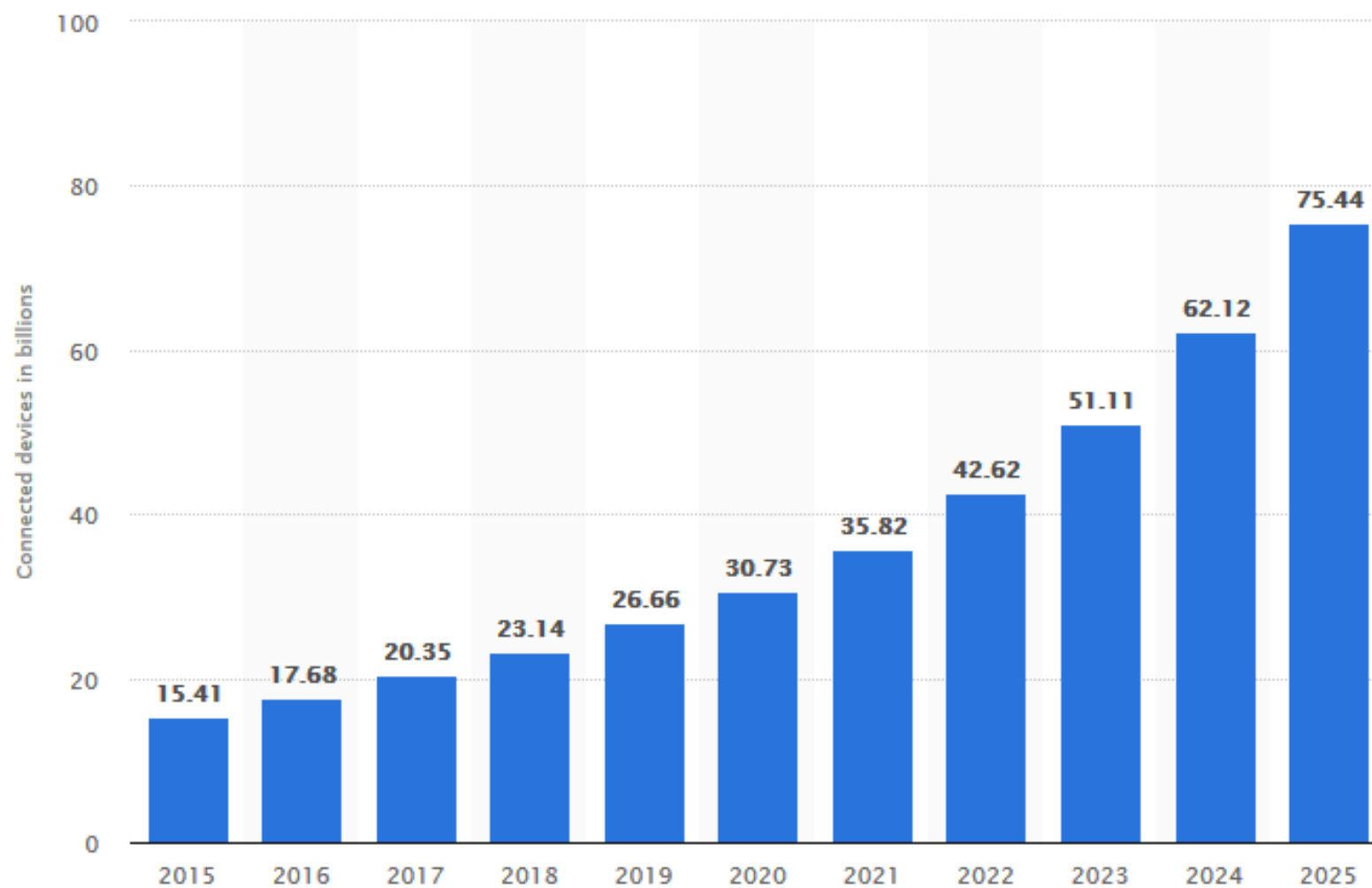
Příklad 2: Enviromentální monitoring

- Detekce lesních požárů
 - Senzory: teplota, vlhkost, úroveň osvětlení, kouř
 - Včasné varování před potenciálním lesním požárem, odhad rozsahu a intenzity
- Detekce říčních povodní
 - Senzory: ultrazvukový dálkoměr (vodní hladina), průtokoměr
 - Výstraha při zjištění rychlého nárůstu hladiny vody a rychlosti proudění

Příklad 3: Doprava, logistika

- Generování a plánování tras
- Sledování vozového parku
 - Sledování polohy vozidel v reálném čase
 - Upozornění na odchylky v plánovaných trasách
- Sledování zásilek
 - Monitorování podmínek uvnitř kontejnerů
 - Detekce kažení potravin
- Dálková diagnostika vozidel
 - Detekce závad na vozidle, upozornění na hrozící závady
 - Návrh nápravných opatření

Počet IoT zařízení



Počet IoT zařízení v kontextu ostatních zařízení

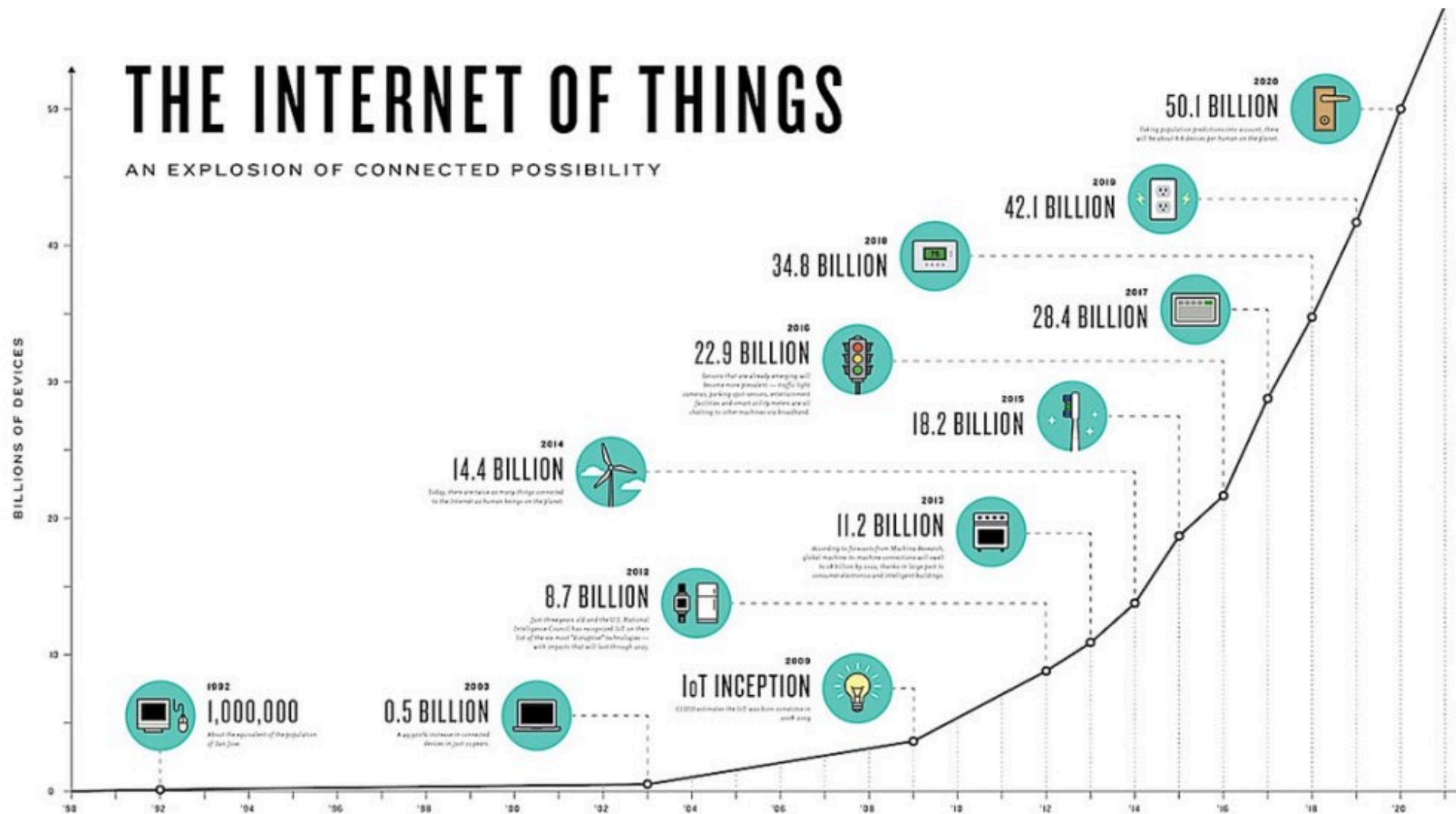
THE INTERNET OF THINGS

Connected devices (billions)



	15 billion	28 billion	CAGR 2015–2021
Cellular IoT	0.4	1.5	27%
Non-cellular IoT	4.2	14.2	22%
PC/laptop/tablet	1.7	1.8	1%
Mobile phones	7.1	8.6	3%
Fixed phones	1.3	1.4	0%

Počet všech připojených zařízení



Příklad IoT zařízení

Chytrá lednička

uživatel

- odchází z domova

zařízení

- ví, že došlo mléko

processing

- pomáhá rozhodovat

notifikace

- upozorňuje uživatele



Rozbor dílčích částí 1/2

Uživatel

- Rozpoznání člena domácnosti (rodič vs. dítě)
- Identifikace důvodu, proč uživatel odchází
- Identifikace kontextu (např. otevírací hodina obchodu)

Lednička

- Jak pozná, že mléko je skutečně potřeba
- Mléko zcela došlo, nebo ho jen málo zbývá (predikce)

Rozbor dílčích částí 2/2

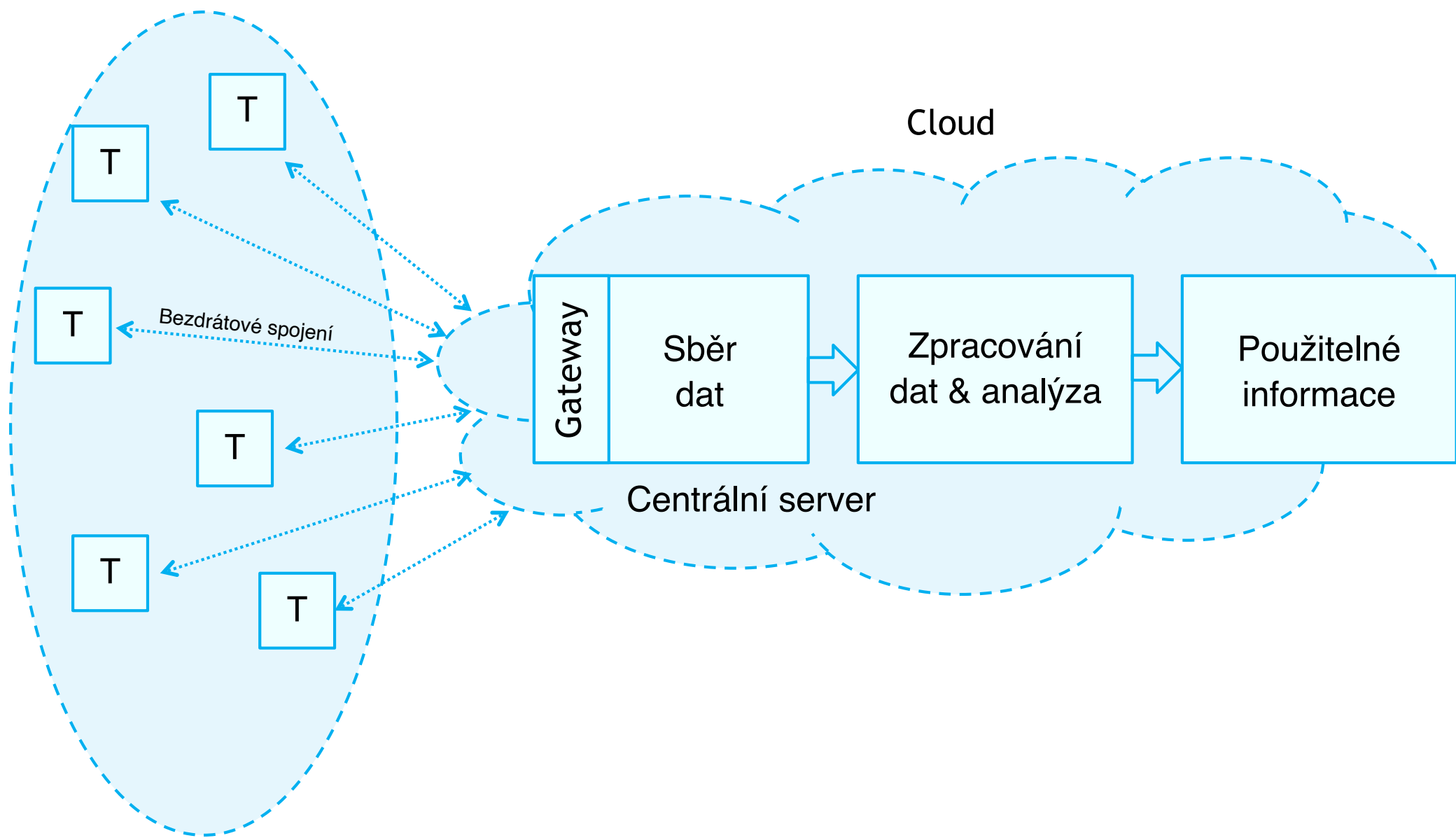
Processing

- Jaká jsou pravidla?
- Jsou pravidla statická nebo dynamická (učení)

Notifikace

- Soukromí?
- Detailnost informace?
- Přemíra informací?

Architektura IoT systému



Požadavky na architekturu

- Sběr, uložení, analýza a sdílení dat / informací / znalostí
- Zpracování velkého objemu dat - **big data**
- Dynamika, adaptace, automatická konfigurace
- Interoperabilní a efektivní přenos a sdílení dat
 - volba vhodného přenosového standardu
- Integrovatelnost do informačních systémů
- Bezpečnost

Základní stavební bloky IoT systému

Hardware

- zařízení generující data, komunikační a výpočetní infrastruktura, datová úložiště

Middleware

- programové vybavení překrývající heterogenitu aplikací, operačních systémů a hardware tím, že poskytuje jednotné rozhraní; umožňuje sběr, uložení a sdílení dat

Software

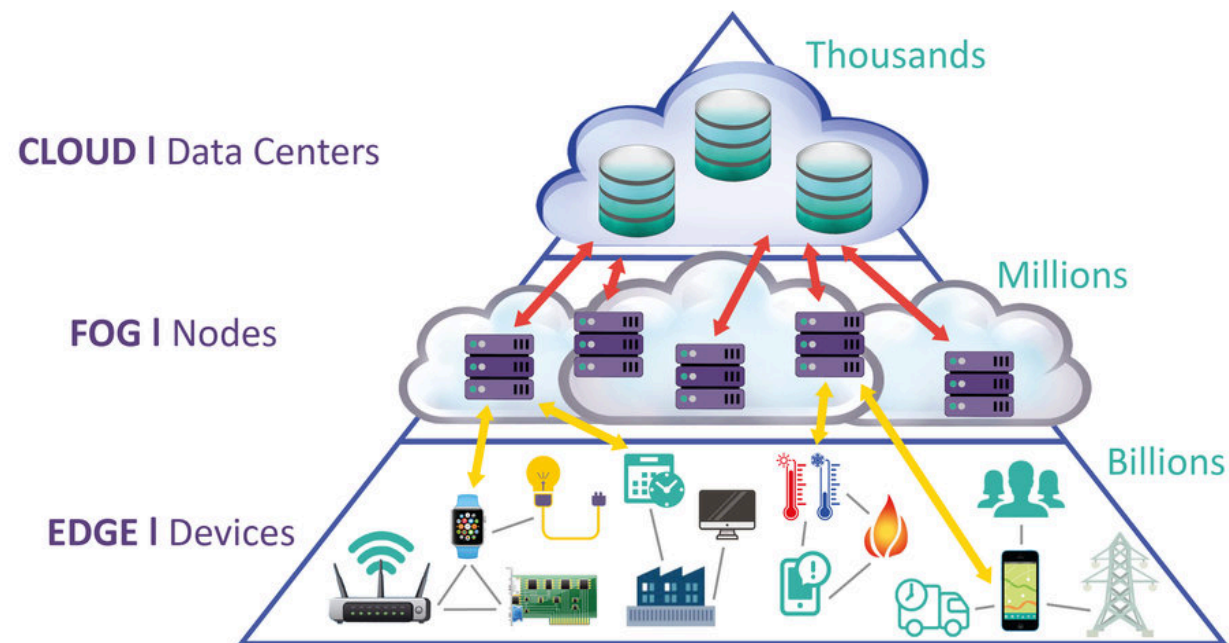
- analýza a kombinace velkého množství heterogenních dat a vytváření informací a znalostí (datová fúze)
- výsledky fúze mohou vést k ovlivnění podmínek nebo k jako podpora rozhodovacích a organizačních procesů

Způsoby propojení v rámci IoT architektury

Propojení mezi prvky IoT bude vždy závislé na určení systému

Komunikace většinou probíhá

- Mezi zařízeními
- Mezi zařízeními a cloudem
- Mezi cloudy



Způsoby propojení mezi zařízeními

Komunikaci mezi zařízeními (**Edge devices**) bude použita v systémech, kde využití cloudu pro ukládání, vyhodnocování a sdílení dat není pro dané řešení vyhovující

- Nedostatečná (nebo dokonce žádná) kapacita linky pro zaslání veškerých dat
- V RT systémech je doba nutná pro poslání dat do cloudu a zpět nepřijatelná

Pro některé systémy (Industrial IoT) je lepší využívat decentralizovaný **Fog computing**

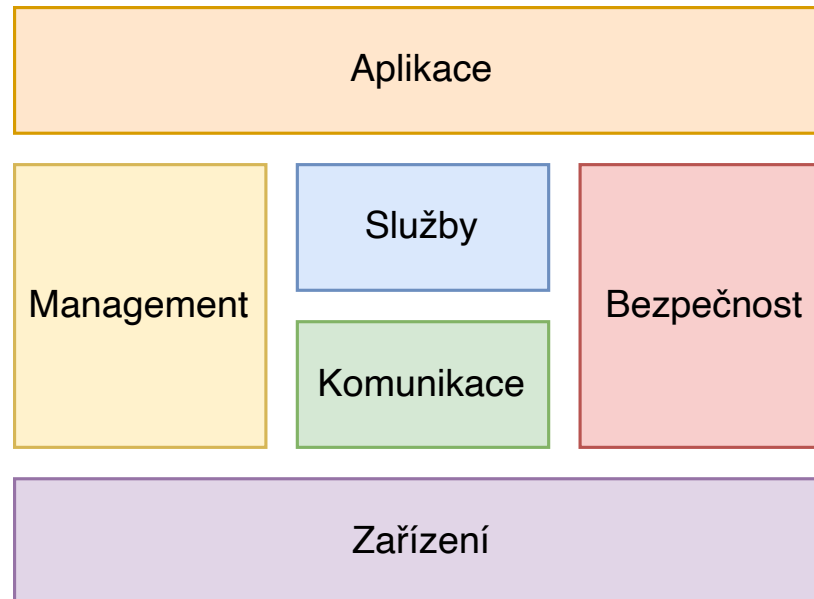
- Zpracování dat blíže zařízením (předzpracování), Peer-to-peer komunikace
- Lepší škálovatelnost, spolehlivost, rychlejší odezva, snížení nákladů

Komunikace ze **zařízení do cloudu** je obvyklá v Consumer IoT.

Komunikace **mezi cloudy** se používá při sdílení dat mezi doménami, např. mezi privátním a veřejným cloudem

Logický design IoT systému

- Logický návrh systému IoT se týká abstraktní reprezentace entit a procesů, bez ohledu na nízkourovňová specifika implementace.
- Systém IoT se skládá z řady funkčních bloků které systému poskytují možnosti identifikace, snímání, ovládání, komunikaci a řízení.

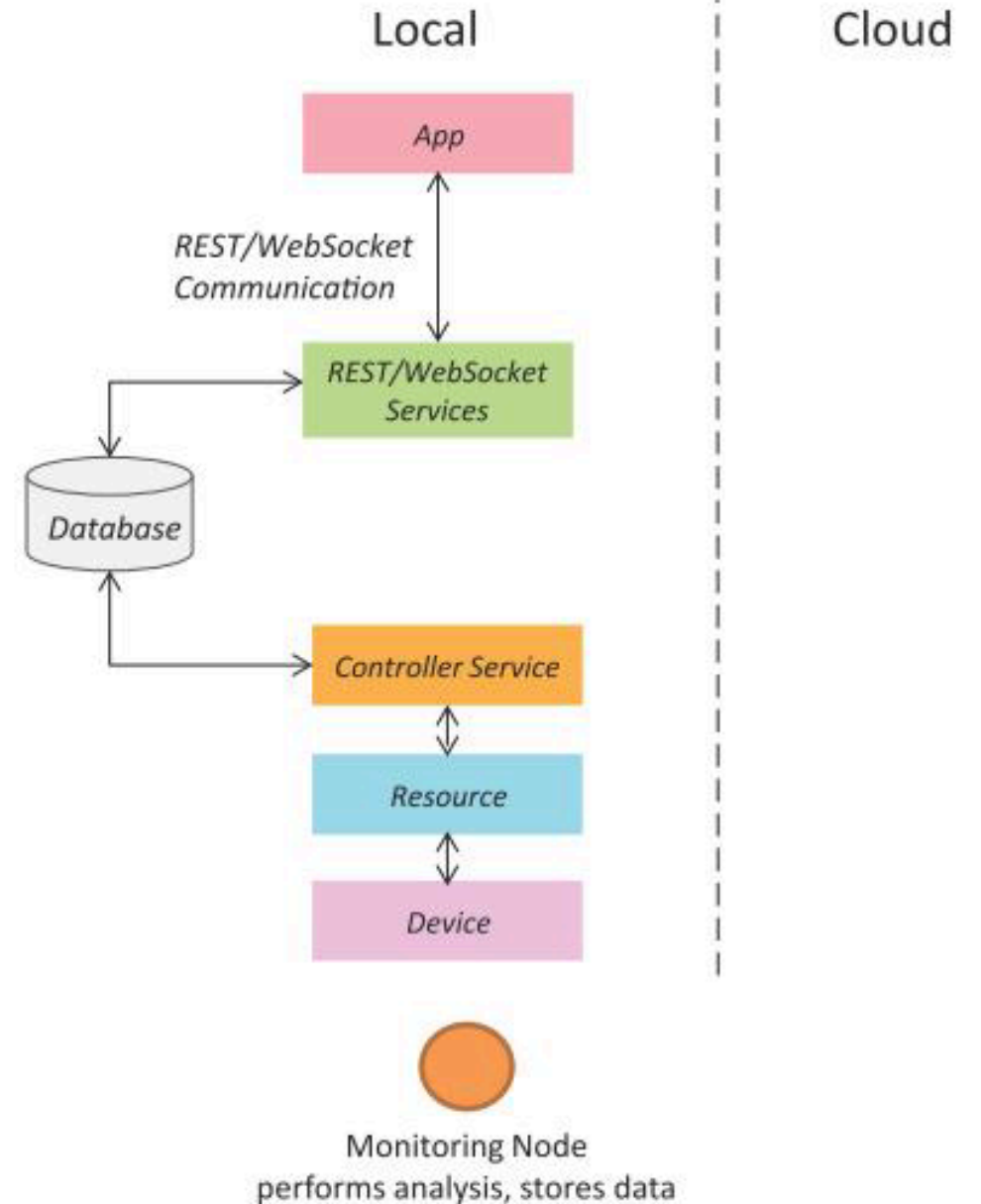


Komponenty IoT systémů

- **Zařízení:** umožňuje identifikaci, vzdálené snímání, ovládání a monitorování.
- **Zdroje:** softwarové komponenty v zařízení IoT pro přístup k informacím ze senzorů nebo pro ovládání akčních členů. Zdroje zahrnují také softwarové komponenty, které umožňují přístup zařízení k síti.
- **Služba řídicí jednotky:** služba, která běží v zařízení a komunikuje s webovými službami - odesílá data ze zařízení a přijímá příkazy z aplikace pro ovládání zařízení.
- **Databáze:** lokální, nebo v cloudu, ukládá data generovaná zařízením.
- **Webová služba:** propojuje zařízení, aplikace, databázi a analytické komponenty
- **Analytická složka:** analýza a generování výsledků srozumitelné a dostupné formě.
- **Aplikace:** poskytují rozhraní, které mohou uživatelé používat k ovládání a sledování IoT systému a prohlížení zpracovaných dat.

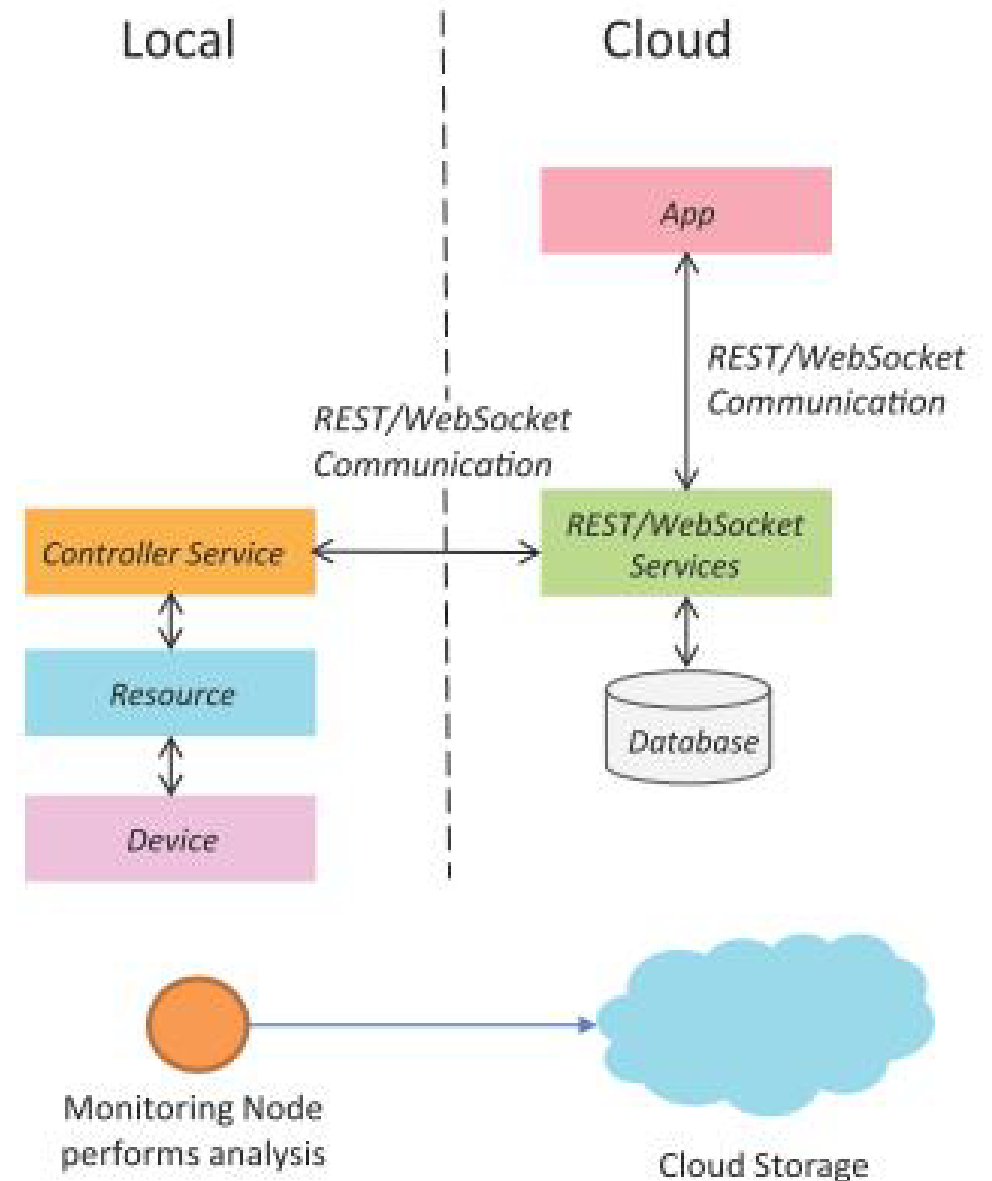
IoT - úroveň 1

- má jeden uzel/zařízení, který provádí snímání a/nebo provádí činnost pomocí aktuátoru a ukládá data, provádí analýzu a je hostitelem aplikaci
- vhodné pro modelování low-cost řešení s nízkou složitostí, kde jsou data nejsou nijak velká a požadavky na analýzu nejsou výpočetně náročné



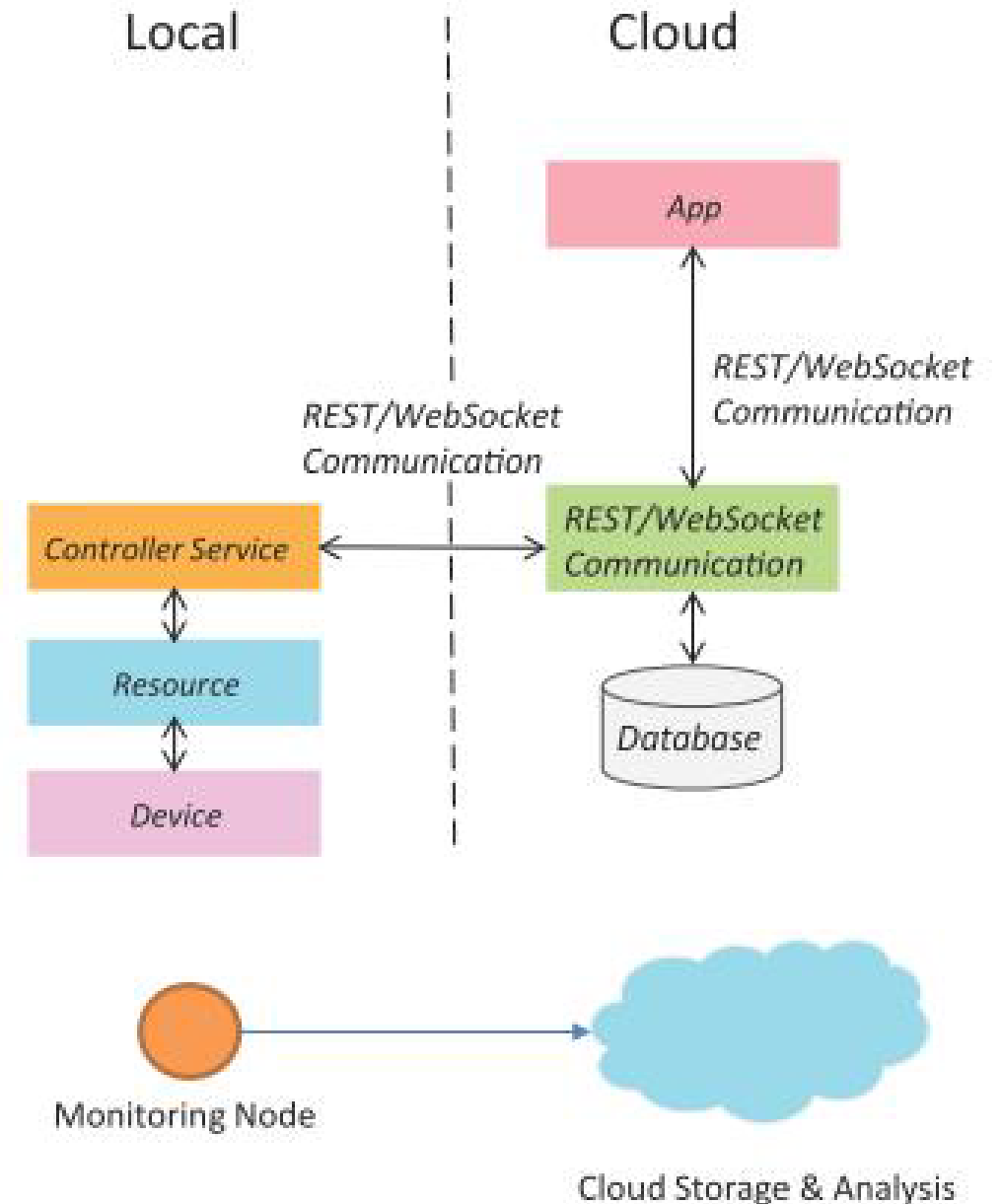
IoT - úroveň 2

- Má jeden uzel, který provádí snímání a/nebo ovládání a lokální analýzu.
- Data jsou uložena v cloudu a aplikace je obvykle založena na cloudu.
- Vhodné pro řešení, kde se jedná o velký objem dat, nicméně primární analýza není výpočetně náročná a lze ji provést lokálně.



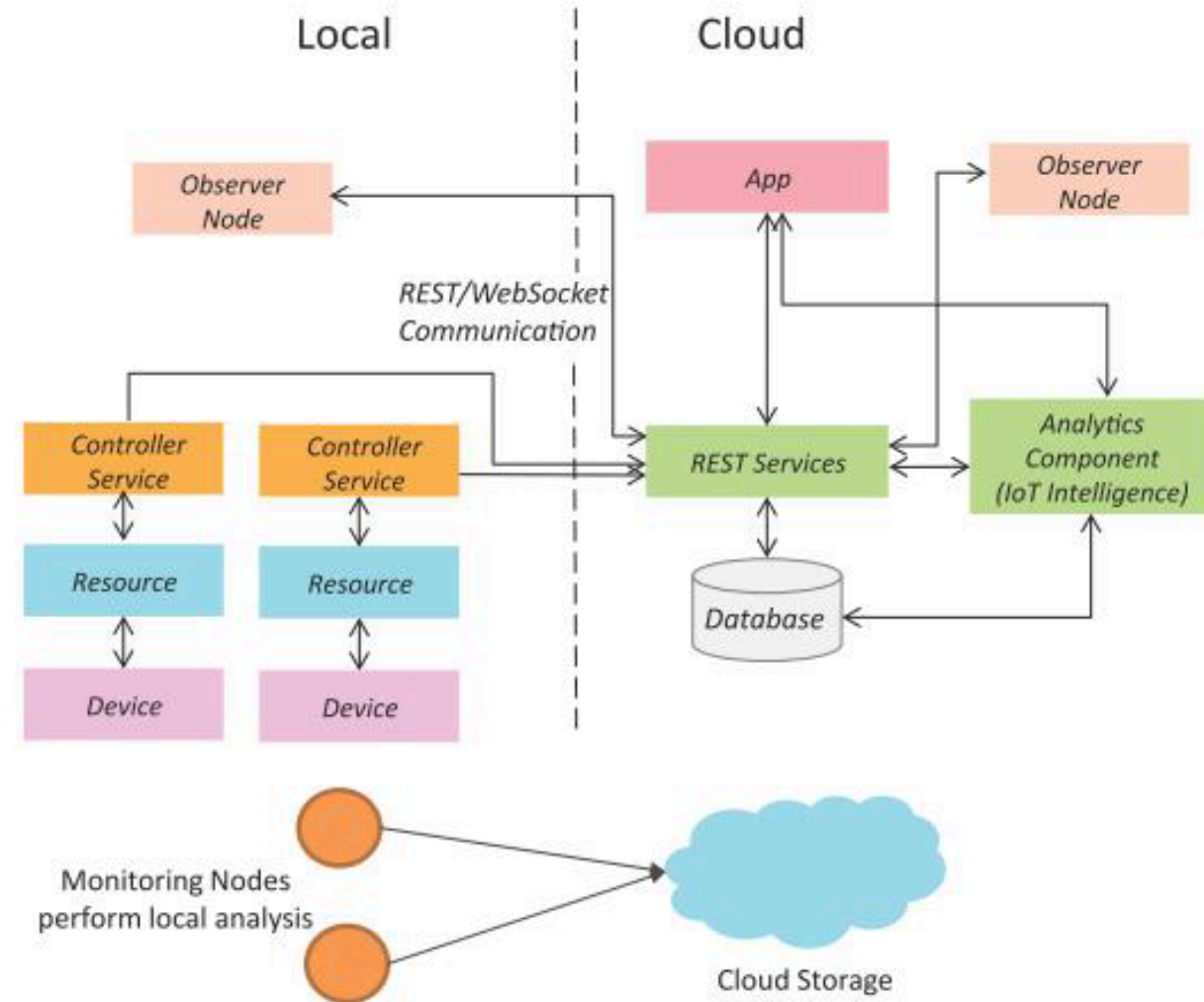
IoT - úroveň 3

- Má jeden uzel.
- Data jsou uložena a analyzována v cloudu a aplikace je založena na cloudu.
- Vhodné pro řešení v aplikacích, kde senzor produkuje velká data a analýza je výpočetně náročná.



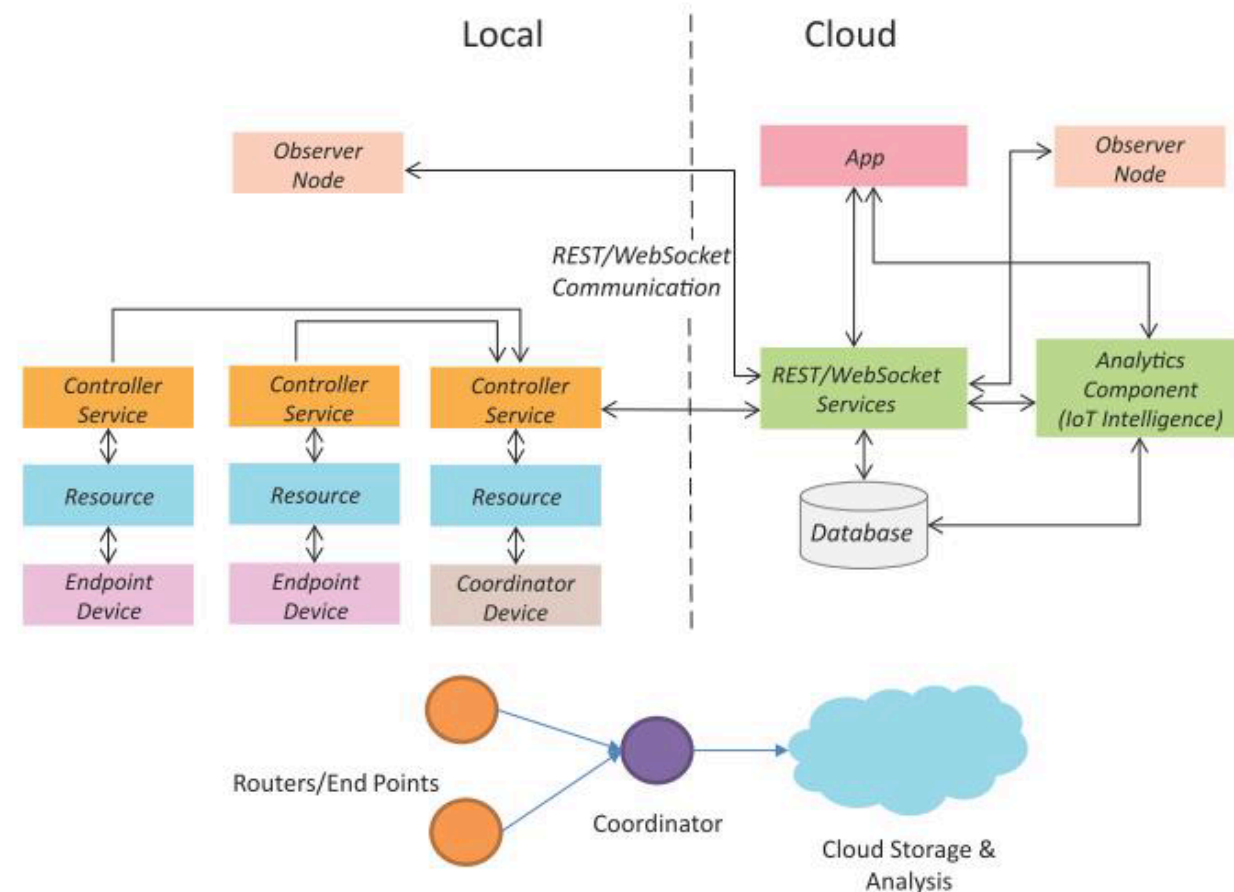
IoT - úroveň 4

- Má více uzlů, které provádějí místní analýzu.
- Data jsou uložena v cloudu a aplikace je založena na cloudu.
- Obsahuje místní a cloudové pozorovací uzly, které mohou přihlásit k odběru a přijímat informace shromážděné v cloudu ze zařízení internetu věcí.
- Řešení s více uzly, velkými daty a výpočetně náročnou analýzou.



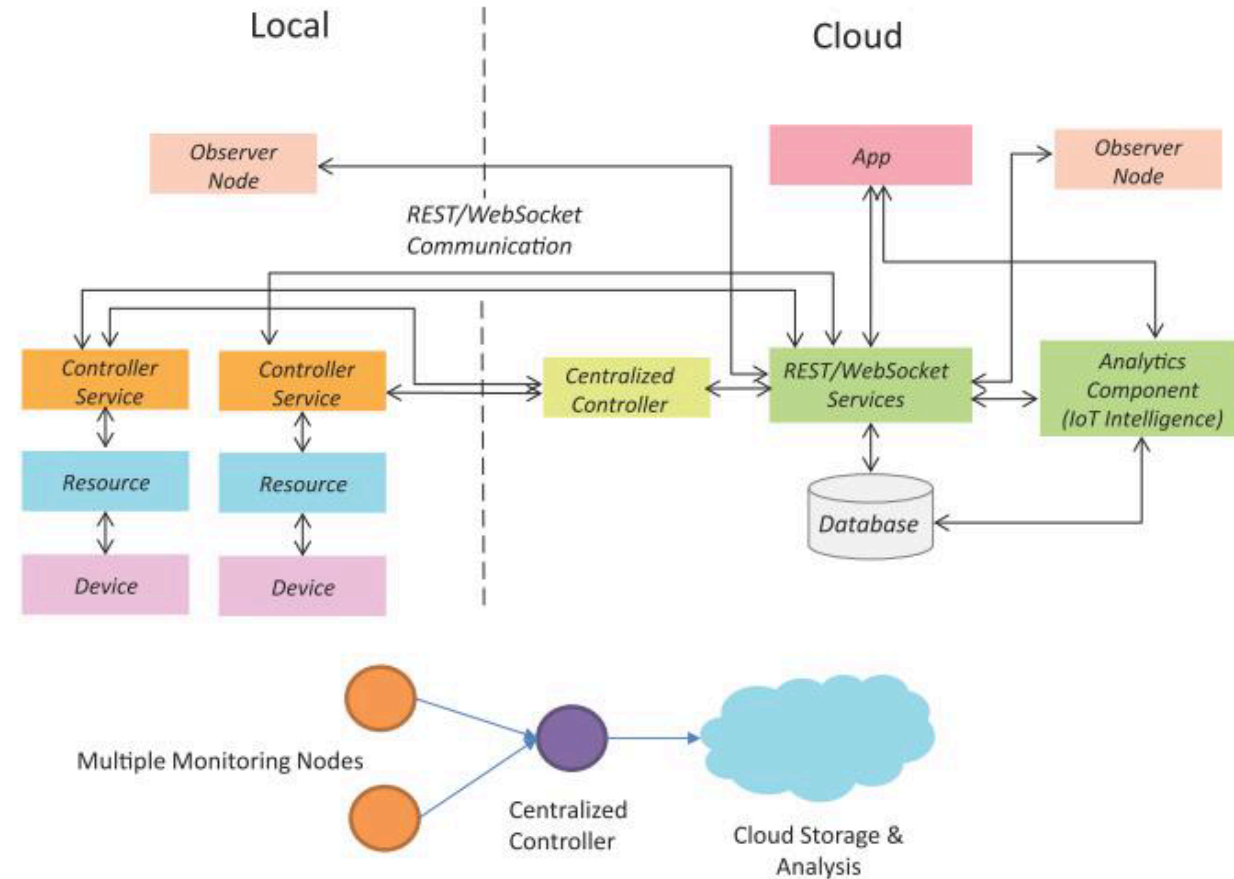
IoT - úroveň 5

- Má více koncových zařízení a jeden koordinační uzel.
- Koordinační uzel shromažďuje data z koncových zařízení a odesílá je do cloudu.
- Data se ukládají a analyzují v cloudu
- Typicky bezdrátové senzorové sítě, kde jsou velká data a analýza je výpočetně náročná.



IoT - úroveň 6

- Má více nezávislých koncových uzlů, které odesílají data do cloudu.
- Analytická komponenta ukládá výsledky do cloudové databáze.
- Výsledky jsou vizualizovány pomocí cloudové aplikace.
- Centralizovaná řídicí jednotka zná stav koncových uzlů a posílá řídicí příkazy.



Programovací jazyk Python

Jak začít?

- Pokud máme Python nainstalovaný (a systémová cesta **PATH** odkazuje do adresáře, kde je umístěn binární soubor), stačí ho spustit příkazem **python**
- Pokud Python nainstalovaný nemáme, tak si ho nainstalujeme :-)
 - <https://www.python.org/downloads>
 - včetně editoru IDLE, instalační nástroj **pip**
 - <https://www.anaconda.com/>
 - IDE Spyder, instalační nástroj **conda**, správa pomocí Anaconda Navigator
 - <https://jupyter.org/>
 - interaktivní prostředí, běží na lokálním počítači, editor v prohlížeči
- Pokud si Python (zatím) instalovat nechceme
 - <https://colab.research.google.com>
 - online interaktivní prostředí, velké množství předinstalovaných modulů

Datové typy v Pythonu

- Primitivní:
 - numerické **int**, **float**, **complex**,
 - pravdivostní hodnota **bool** (True/False), podmnožina **int**,
- Strukturované:
 - sekvenční seznam **list**, n-tice **tuple**, rozsah **range**,
 - sekvenční textový řetězec **string**,
 - binární sekvenční **bytes**, **bytearray**, **memoryview**,
 - mapující slovník **dict**,
 - množina **set**, **frozenset**
- (abstraktní) datový typ je v Pythonu automaticky určený a implicitní
- je možné provést explicitní konverzi (pokud existuje jednoznačná konverze)