INSTITUTO DE INFORMÁTICA
Universidade Federal de Goiás

IoT
Internet das Coisas

Alunos:
Aline Ayumi - 202201680
Aline Lima - 202201681
Aline Nunes - 202206141
Artur Lapot - 201905524
Stephany Milhomem - 202201714

INSTITUTO DE INFORMÁTICA
UFG

UFG
UNIVERSIDADE FEDERAL DE GOIÁS

Conceito

A internet das coisas (IoT) refere-se a uma rede de dispositivos físicos, veículos, eletrodomésticos e outros objetos físicos incorporados com sensores, software e conectividade de rede, permitindo coletar e compartilhar dados entre si.

Dispositivos IoT, também conhecidos como "objetos inteligentes", podem variar de dispositivos simples de "casas inteligentes", como termostatos inteligentes, a dispositivos vestíveis, como smartwatches e roupas com RFID, até máquinas industriais complexas e sistemas de transporte.

Smart Decisions Game

"Smart Decisions" é um jogo que simula o processo de design de sistemas de software e promove o aprendizado sobre isso de maneira divertida.

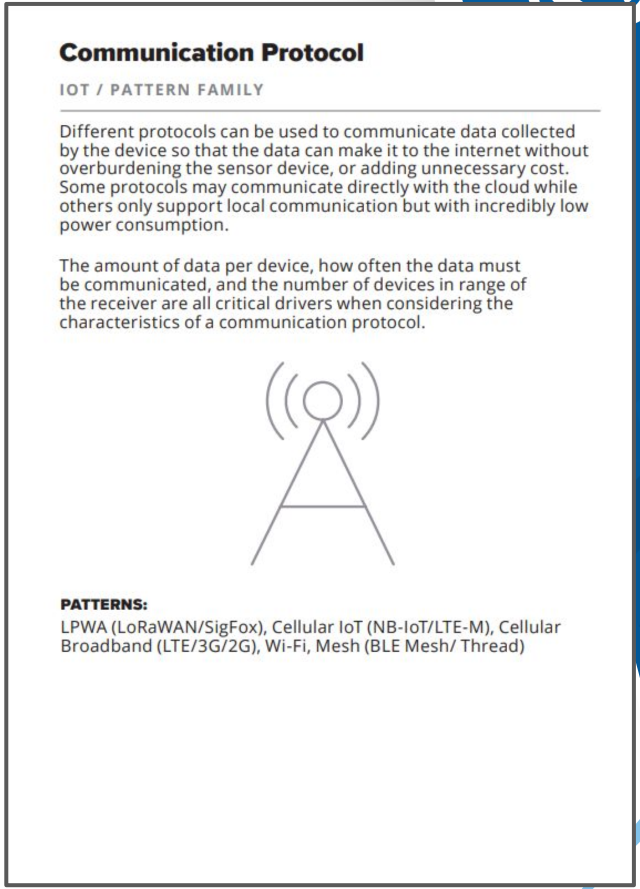
O jogo visa simular o processo de design de sistemas de IoT, onde os jogadores ou equipes competem para resolver desafios arquitetônicos. Ao final, a equipe que fizer as melhores escolhas arquitetônicas, balanceando os requisitos e atributos de qualidade, vence.

Como se joga?

1. Preparação:

- Cada jogador ou equipe começa revisando o cenário de jogo. Esse cenário define os objetivos do sistema IoT que será projetado, incluindo os atributos de qualidade (por exemplo, segurança, escalabilidade, consumo de energia, etc.).

- As cartas de design são distribuídas, representando diferentes componentes e opções de arquitetura.



Communication Protocol

IOT / PATTERN FAMILY

Different protocols can be used to communicate data collected by the device so that the data can make it to the internet without overburdening the sensor device, or adding unnecessary cost. Some protocols may communicate directly with the cloud while others only support local communication but with incredibly low power consumption.

The amount of data per device, how often the data must be communicated, and the number of devices in range of the receiver are all critical drivers when considering the characteristics of a communication protocol.

PATTERNS:
LPWA (LoRaWAN/SigFox), Cellular IoT (NB-IoT/LTE-M), Cellular Broadband (LTE/3G/2G), Wi-Fi, Mesh (BLE Mesh/ Thread)

Como se joga?

2. Primeira Iteração:

- O jogador ou equipe escolhe suas decisões de design com base nas cartas disponíveis. Exemplos de decisões incluem:
    - **Fonte de Energia:** Baterias recarregáveis ou não recarregáveis, etc.
    - **Protocolo de Comunicação:** WLAN, Mesh, etc.
    - **Arquitetura de Borda:** Gateway, no Edge, etc.
    - **Banco de Dados:** Cloud SQL, Datastore, etc.
    - **Processamento de Dados:** Dataflow, Dataproc, etc.

Como se joga?

3. Registro na Folha de Pontuação:

Smart Decisions: Game Scorecard

Player name: ______________________________

	Iteration #1	Iteration #2	Iteration #3	Iteration #4	Iteration #5	
(a) Design Decisions *(Names of selected design concept(s))*						
(b) Driver selection points *(from cards)*						
(c) Instantiation points *(from dice)*						
(d) Analysis bonus points *(from review)*						**Final score:**
(e) Iteration total *(b + c + d)*						

Note: Help us improve by filling the questionnaire on the back of this sheet at the end of the session.

Como se joga?

3. Registro na Folha de Pontuação:

- **(a) Design Decisions:** O jogador anota as escolhas de design tomadas para a iteração.
- **(b) Driver Selection Points:** Pontos baseados nas cartas escolhidas. Esses pontos refletem como bem as decisões de design atendem aos atributos de qualidade do cenário.
- **(c) Instantiation Points:** O jogador rola os dados para ganhar pontos aleatórios, representando a sorte na implementação do design.
- **(d) Analysis Bonus Points:** Pontos bônus podem ser concedidos após uma análise crítica das escolhas de design.
- **(e) Iteration Total:** A pontuação final da iteração é calculada somando os pontos das seções **b**, **c** e **d**.

Como se joga?

4. Próximas Iterações:

- O jogo avança com os jogadores tomando novas decisões a cada iteração. Em cada iteração, uma necessidade específica do sistema é avaliada. As cinco necessidades do jogo são: Power Source, Communication Protocol, Edge Architecture, Database e Data Processing.

- Em cada iteração, os jogadores se concentram em resolver uma dessas necessidades específicas. As decisões e escolhas tecnológicas são registradas e pontuadas, com base em como cada decisão atende aos atributos de qualidade definidos no cenário.

- Os pontos de cada iteração são registrados na folha de pontuação, e o total de cada rodada é calculado.

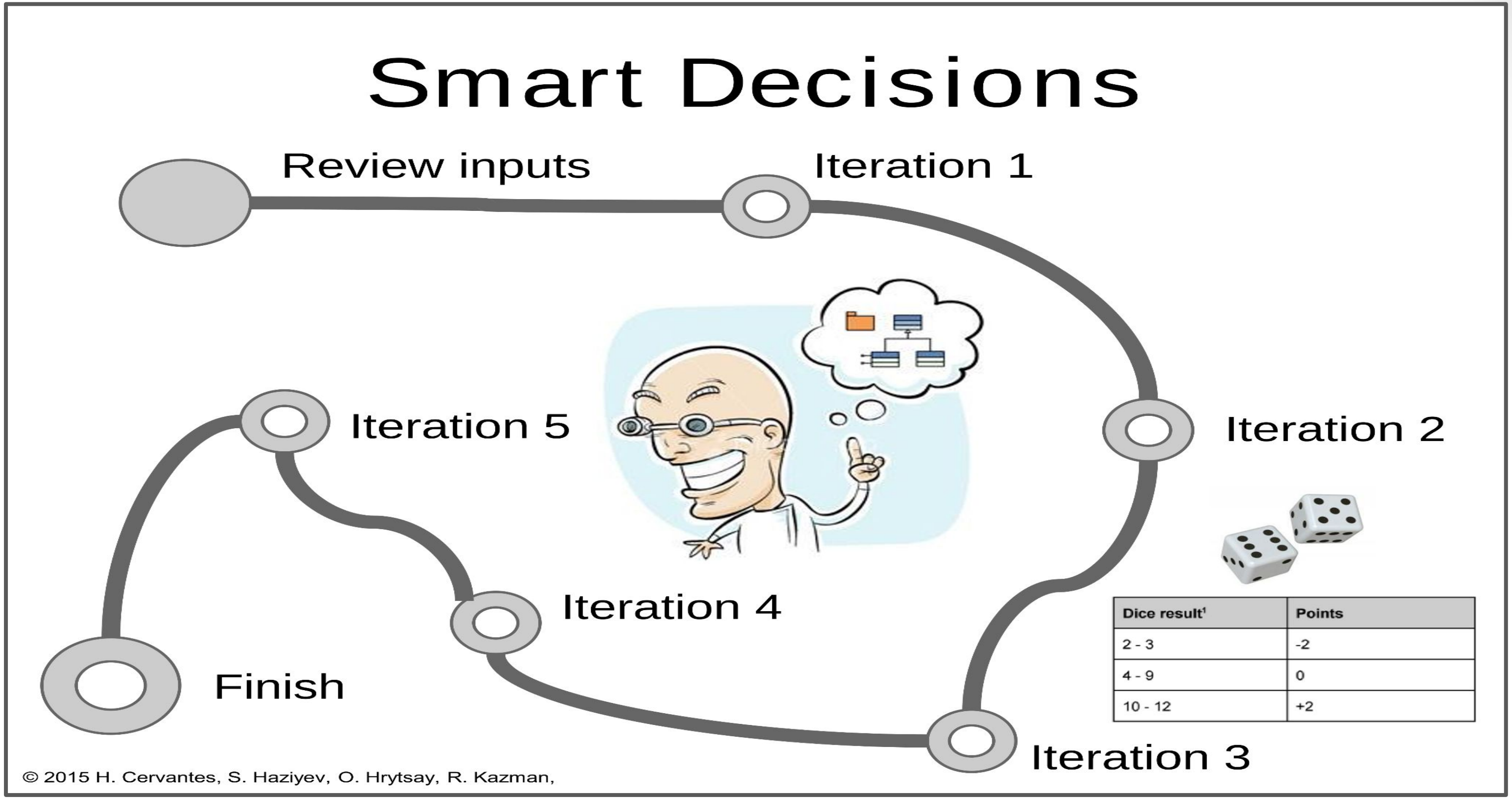
Como se joga?

5. Final do Jogo:

- Após todas as iterações (5), a pontuação final de cada jogador ou equipe é somada.

- O jogador ou equipe com a maior pontuação total é o vencedor.

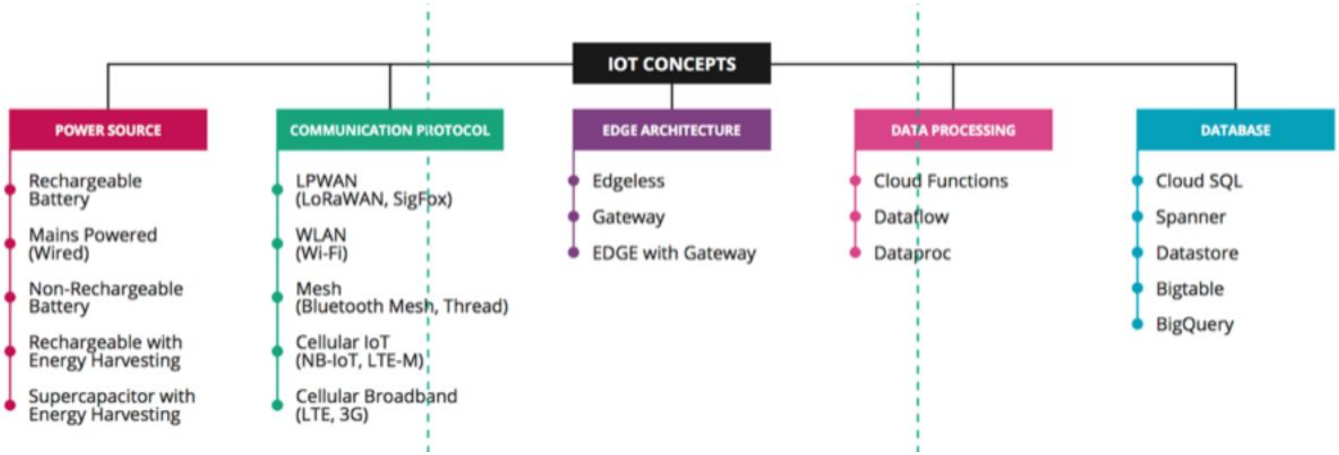
Tabuleiro do jogo



Dice result[1]	Points
2 - 3	-2
4 - 9	0
10 - 12	+2

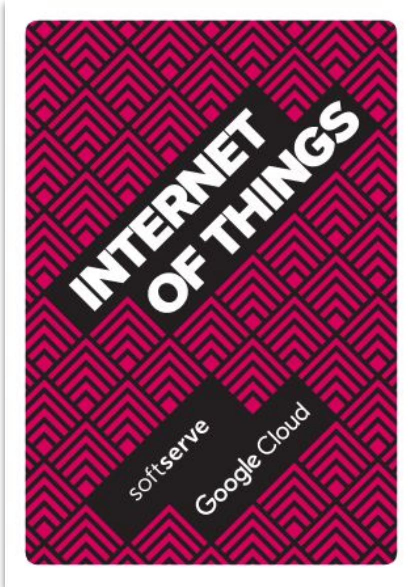
Cartas do jogo

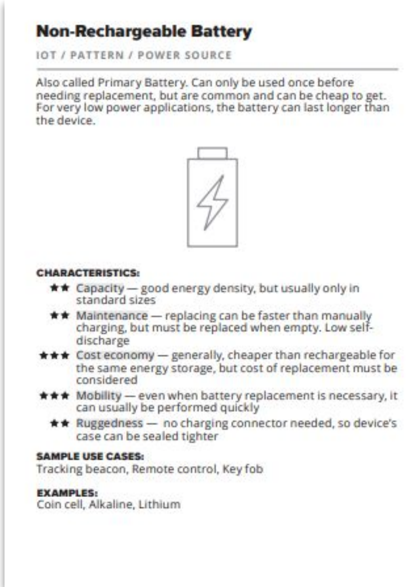
- **Cartas azul-escuras:** representam as famílias de cada padrão de design, sendo elas:



Cartas do jogo

- **Cartas vermelhas:** representam os padrões de design de fonte de energia (Power Source). Por exemplo:




Non-Rechargeable Battery

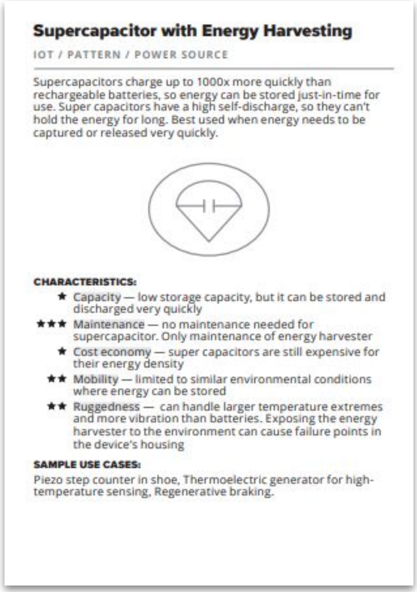
IOT / PATTERN / POWER SOURCE

Also called Primary Battery. Can only be used once before needing replacement, but are common and can be cheap to get. For very low power applications, the battery can last longer than the device.

CHARACTERISTICS:
- ★★ Capacity — good energy density, but usually only in standard sizes
- ★★ Maintenance — replacing can be faster than manually charging, but must be replaced when empty. Low self-discharge
- ★★★ Cost economy — generally, cheaper than rechargeable for the same energy storage, but cost of replacement must be considered
- ★★★ Mobility — even when battery replacement is necessary, it can usually be performed quickly
- ★★ Ruggedness — no charging connector needed, so device's case can be sealed tighter

SAMPLE USE CASES:
Tracking beacon, Remote control, Key fob

EXAMPLES:
Coin cell, Alkaline, Lithium


Supercapacitor with Energy Harvesting

IOT / PATTERN / POWER SOURCE

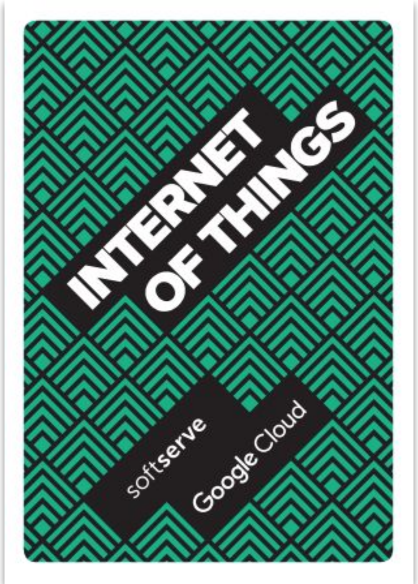
Supercapacitors charge up to 1000x more quickly than rechargeable batteries, so energy can be stored just-in-time for use. Super capacitors have a high self-discharge, so they can't hold the energy for long. Best used when energy needs to be captured or released very quickly.

CHARACTERISTICS:
- ★ Capacity — low storage capacity, but it can be stored and discharged very quickly
- ★★★ Maintenance — no maintenance needed for supercapacitor. Only maintenance of energy harvester
- ★ Cost economy — super capacitors are still expensive for their energy density
- ★★ Mobility — limited to similar environmental conditions where energy can be stored
- ★★ Ruggedness — can handle larger temperature extremes and more vibration than batteries. Exposing the energy harvester to the environment can cause failure points in the device's housing

SAMPLE USE CASES:
Piezo step counter in shoe, Thermoelectric generator for high-temperature sensing, Regenerative braking.

Cartas do jogo

- **Cartas verdes:** representam os padrões de design de protocolo de comunicação (Communication Protocol). Por exemplo:



WLAN (Wi-Fi)

IOT / PATTERN / COMMUNICATION PROTOCOL

High-bandwidth general purpose network protocol used by computers, phones, and other devices. Allows direct connection to the Internet and an internationally common standard.

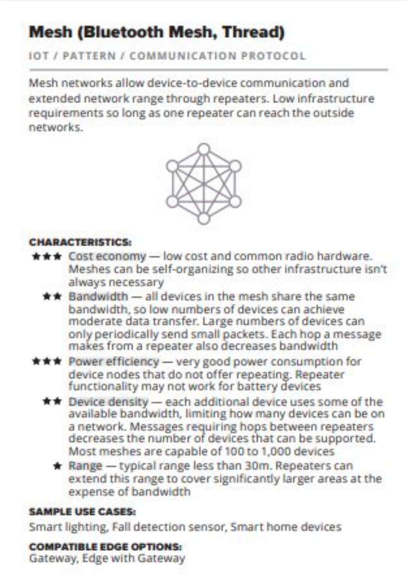
Wi Fi CERTIFIED

CHARACTERISTICS:

- ★★ Cost economy — even though Wi-Fi radios are common, their high-bandwidth support makes them a mid-level cost
- ★★★ Bandwidth — very high bandwidth and very low latency. Excellent for streaming or "constant-on" connections
- ★ Power efficiency — high bandwidth and low latency come at the expense of power consumption. IoT specific radios will allow for lower power consumption than general purpose versions
- ★★ Device density — tens to hundreds of devices per access point. Multiple access points can be used to cover a space
- ★★ Range — typical range can reach up to 150 feet (46 m) indoors and 300 feet (92 m) outdoors, bandwidth decreases with range. High power radios allow for decent range depending on obstacles

SAMPLE USE CASES:
Wired always-on sensors such as thermostats, Smart cameras, General purpose computing, Edge device backhaul

COMPATIBLE EDGE OPTIONS:
Edgeless, Gateway, Edge with Gateway



Mesh (Bluetooth Mesh, Thread)

IOT / PATTERN / COMMUNICATION PROTOCOL

Mesh networks allow device-to-device communication and extended network range through repeaters. Low infrastructure requirements so long as one repeater can reach the outside networks.

CHARACTERISTICS:

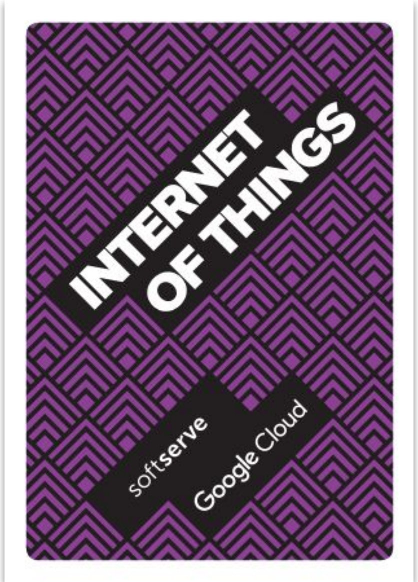
- ★★★ Cost economy — low cost and common radio hardware. Meshes can be self-organizing so other infrastructure isn't always necessary
- ★★ Bandwidth — all devices in the mesh share the same bandwidth, so low numbers of devices can achieve moderate data transfer. Large numbers of devices can only periodically send small packets. Each hop a message makes from a repeater also decreases bandwidth
- ★★★ Power efficiency — very good power consumption for device nodes that do not offer repeating. Repeater functionality may not work for battery devices
- ★★ Device density — each additional device uses some of the available bandwidth, limiting how many devices can be on a network. Messages requiring hops between repeaters decreases the number of devices that can be supported. Most meshes are capable of 100 to 1,000 devices
- ★ Range — typical range less than 30m. Repeaters can extend this range to cover significantly larger areas at the expense of bandwidth

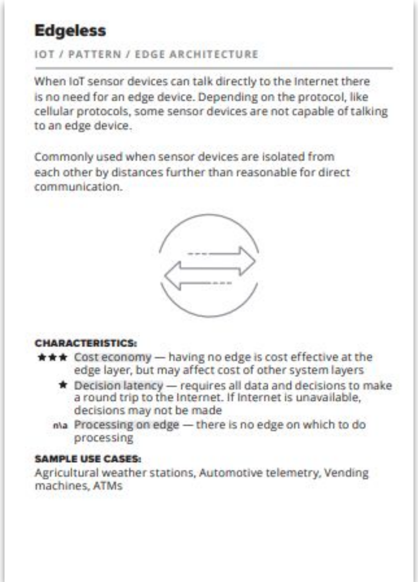
SAMPLE USE CASES:
Smart lighting, Fall detection sensor, Smart home devices

COMPATIBLE EDGE OPTIONS:
Gateway, Edge with Gateway

Cartas do jogo

- **Cartas roxas:** representam os padrões de design de arquitetura de borda (Edge Architecture). Por exemplo:





Edgeless

IOT / PATTERN / EDGE ARCHITECTURE

When IoT sensor devices can talk directly to the Internet there is no need for an edge device. Depending on the protocol, like cellular protocols, some sensor devices are not capable of talking to an edge device.

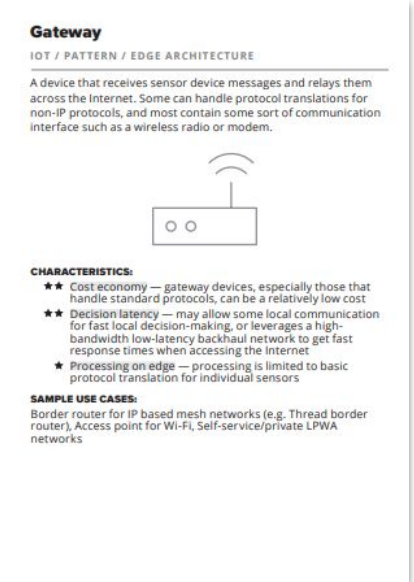
Commonly used when sensor devices are isolated from each other by distances further than reasonable for direct communication.

CHARACTERISTICS:

- ★★★ Cost economy — having no edge is cost effective at the edge layer, but may affect cost of other system layers
- ★ Decision latency — requires all data and decisions to make a round trip to the Internet. If Internet is unavailable, decisions may not be made
- n/a Processing on edge — there is no edge on which to do processing

SAMPLE USE CASES:

Agricultural weather stations, Automotive telemetry, Vending machines, ATMs



Gateway

IOT / PATTERN / EDGE ARCHITECTURE

A device that receives sensor device messages and relays them across the Internet. Some can handle protocol translations for non-IP protocols, and most contain some sort of communication interface such as a wireless radio or modem.

CHARACTERISTICS:

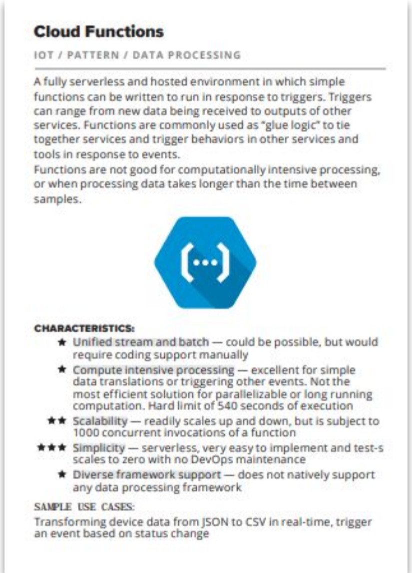
- ★★ Cost economy — gateway devices, especially those that handle standard protocols, can be a relatively low cost
- ★★ Decision latency — may allow some local communication for fast local decision-making, or leverages a high-bandwidth low-latency backhaul network to get fast response times when accessing the Internet
- ★ Processing on edge — processing is limited to basic protocol translation for individual sensors

SAMPLE USE CASES:

Border router for IP based mesh networks (e.g. Thread border router), Access point for Wi-Fi, Self-service/private LPWA networks

Cartas do jogo

- **Cartas rosas:** representam os padrões de design de processamento de dados (Data Processing). Por exemplo:



Cloud Functions

IOT / PATTERN / DATA PROCESSING

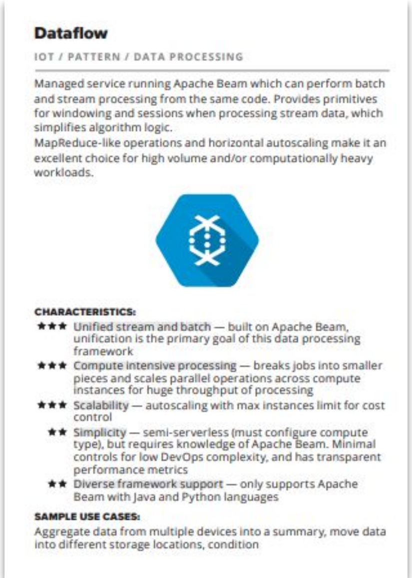
A fully serverless and hosted environment in which simple functions can be written to run in response to triggers. Triggers can range from new data being received to outputs of other services. Functions are commonly used as "glue logic" to tie together services and trigger behaviors in other services and tools in response to events.
Functions are not good for computationally intensive processing, or when processing data takes longer than the time between samples.

CHARACTERISTICS:

- ★ Unified stream and batch — could be possible, but would require coding support manually
- ★ Compute intensive processing — excellent for simple data translations or triggering other events. Not the most efficient solution for parallelizable or long running computation. Hard limit of 540 seconds of execution
- ★★ Scalability — readily scales up and down, but is subject to 1000 concurrent invocations of a function
- ★★★ Simplicity — serverless, very easy to implement and test-s scales to zero with no DevOps maintenance
- ★ Diverse framework support — does not natively support any data processing framework

SAMPLE USE CASES:

Transforming device data from JSON to CSV in real-time, trigger an event based on status change



Dataflow

IOT / PATTERN / DATA PROCESSING

Managed service running Apache Beam which can perform batch and stream processing from the same code. Provides primitives for windowing and sessions when processing stream data, which simplifies algorithm logic.
MapReduce-like operations and horizontal autoscaling make it an excellent choice for high volume and/or computationally heavy workloads.

CHARACTERISTICS:

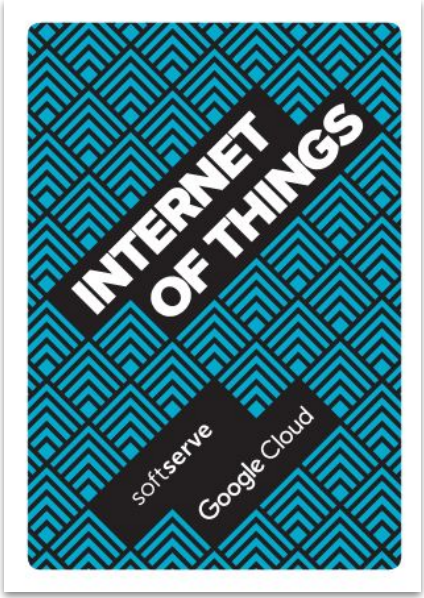
- ★★★ Unified stream and batch — built on Apache Beam, unification is the primary goal of this data processing framework
- ★★★ Compute intensive processing — breaks jobs into smaller pieces and scales parallel operations across compute instances for huge throughput of processing
- ★★★ Scalability — autoscaling with max instances limit for cost control
- ★★ Simplicity — semi-serverless (must configure compute type), but requires knowledge of Apache Beam. Minimal controls for low DevOps complexity, and has transparent performance metrics
- ★★ Diverse framework support — only supports Apache Beam with Java and Python languages

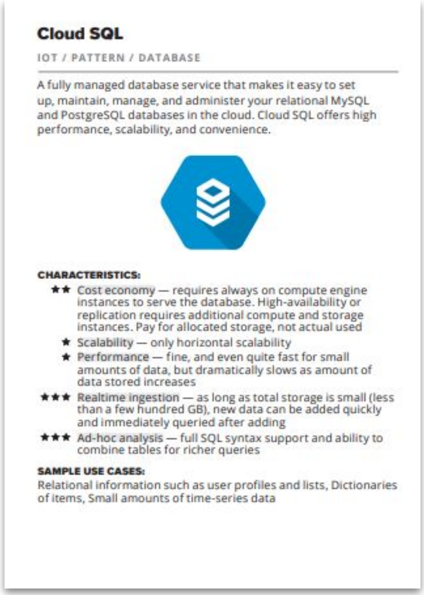
SAMPLE USE CASES:

Aggregate data from multiple devices into a summary, move data into different storage locations, condition

Cartas do jogo

- **Cartas azul-claras:** representam os padrões de design de base de dados (Database). Por exemplo:





Cloud SQL

IOT / PATTERN / DATABASE

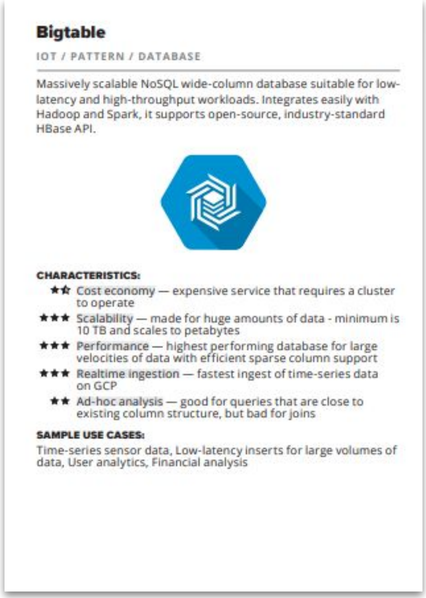
A fully managed database service that makes it easy to set up, maintain, manage, and administer your relational MySQL and PostgreSQL databases in the cloud. Cloud SQL offers high performance, scalability, and convenience.

CHARACTERISTICS:

- ★★ Cost economy — requires always on compute engine instances to serve the database. High-availability or replication requires additional compute and storage instances. Pay for allocated storage, not actual used
- ★ Scalability — only horizontal scalability
- ★ Performance — fine, and even quite fast for small amounts of data, but dramatically slows as amount of data stored increases
- ★★★ Realtime ingestion — as long as total storage is small (less than a few hundred GB), new data can be added quickly and immediately queried after adding
- ★★★ Ad-hoc analysis — full SQL syntax support and ability to combine tables for richer queries

SAMPLE USE CASES:

Relational information such as user profiles and lists, Dictionaries of items, Small amounts of time-series data



Bigtable

IOT / PATTERN / DATABASE

Massively scalable NoSQL wide-column database suitable for low-latency and high-throughput workloads. Integrates easily with Hadoop and Spark, it supports open-source, industry-standard HBase API.

CHARACTERISTICS:

- ★★ Cost economy — expensive service that requires a cluster to operate
- ★★★ Scalability — made for huge amounts of data - minimum is 10 TB and scales to petabytes
- ★★★ Performance — highest performing database for large velocities of data with efficient sparse column support
- ★★★ Realtime ingestion — fastest ingest of time-series data on GCP
- ★★ Ad-hoc analysis — good for queries that are close to existing column structure, but bad for joins

SAMPLE USE CASES:

Time-series sensor data, Low-latency inserts for large volumes of data, User analytics, Financial analysis