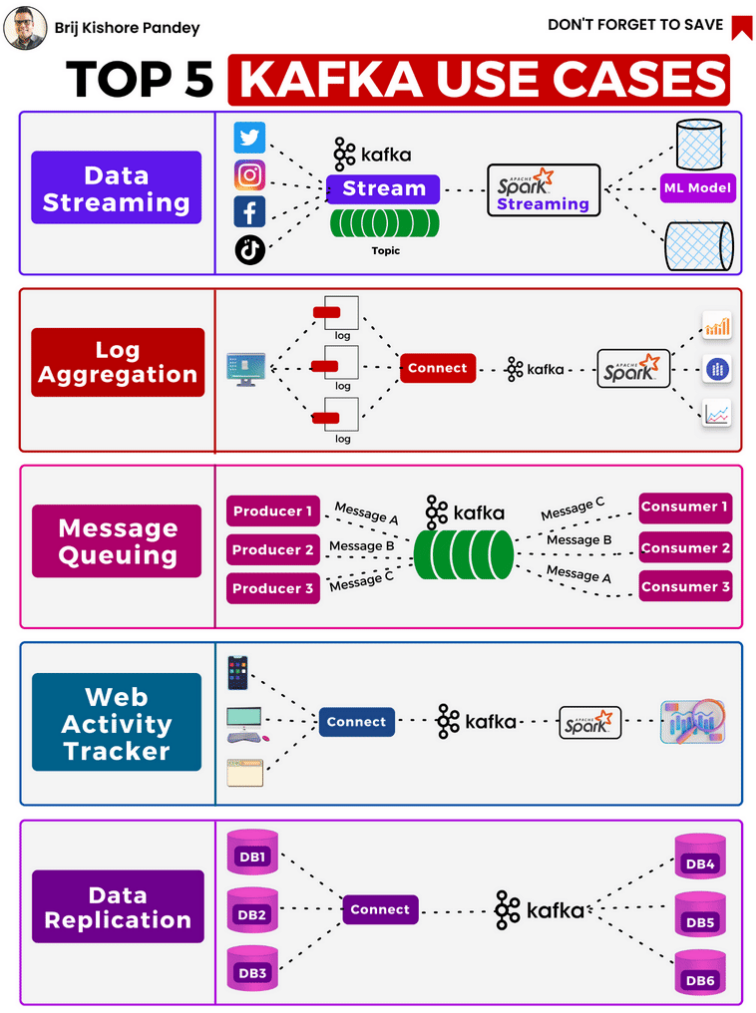
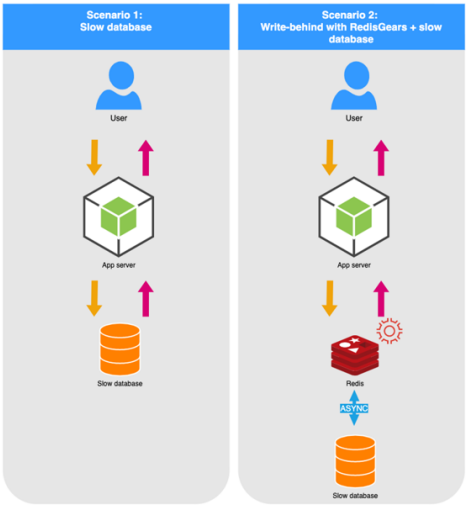
# Type of brokers

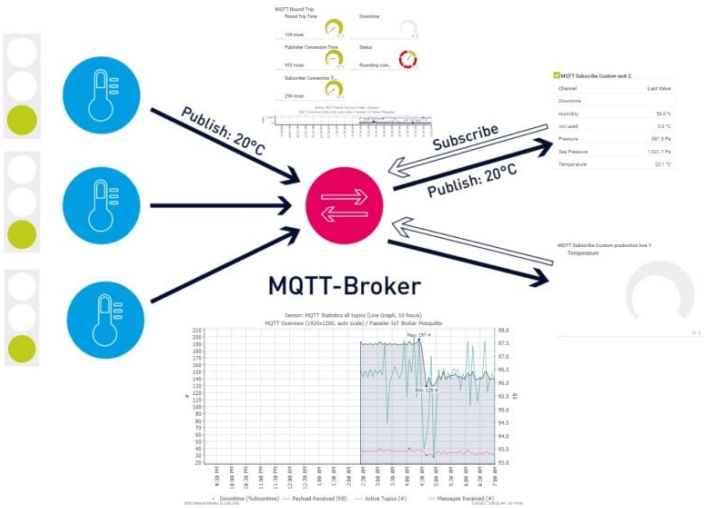
**Apache Kafka** is an open-source, distributed event streaming platform that handles real-time data feeds, acting as a central nervous system for applications by collecting, processing, and storing streams of events (data) at scale with high throughput and low latency. It's used to build reliable data pipelines, stream processing applications, and data integration systems, enabling features like activity tracking, real-time analytics, and decoupling microservices across various industries like finance, IoT, and e-commerce.



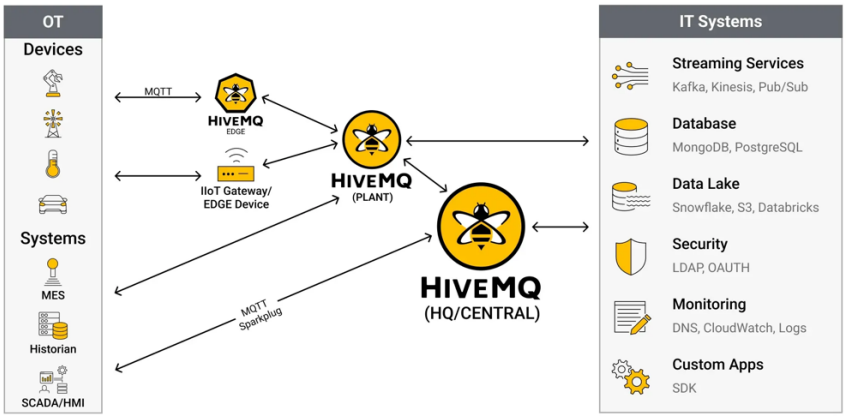
**Redis** (Remote Dictionary Server) is a high-performance, open-source, in-memory data store used as a cache, database, and message broker, known for its speed in handling complex data structures like strings, lists, hashes, and sets. It stores data primarily in RAM for incredibly fast reads and writes, making it ideal for low-latency applications, real-time analytics, session storage, and as a caching layer to offload primary databases. Redis offers features like built-in replication, persistence to disk for durability, and support for modules for JSON, search, and streams, extending its use beyond simple key-value storage.




**MQTT** (Message Queuing Telemetry Transport) is a lightweight, publish-subscribe messaging protocol crucial for the Internet of Things (IoT) and machine-to-machine (M2M) communication, ideal for resource-constrained devices with low bandwidth, enabling efficient, decoupled communication between devices via a central broker and topics. It works by publishers sending messages to a broker, which then routes them to subscribers interested in specific topics, ensuring efficient data flow for remote sensing and control.



**HiveMQ** is a specialized data streaming platform designed for the Internet of Things (IoT) and Industrial AI. As of January 2026, it is positioned as an **Industrial AI Platform** that enables businesses to connect, contextualize, and analyze real-time operational data from the edge to the cloud.




**Apache ActiveMQ** is a popular open-source Java-based message broker that acts as middleware, allowing different applications to communicate reliably by sending and receiving messages through queues and topics, supporting various protocols like JMS, MQTT, and AMQP, and offering features like persistence, clustering, and multi-language client support for decoupled, scalable distributed systems. It decouples senders and receivers, so they don't need to be available simultaneously, making systems more resilient and building asynchronous communication