

# Big Data Ecosystem - Architecture

Dr. Deepak Saxena, SME IIT Jodhpur

# Big Data Architecture

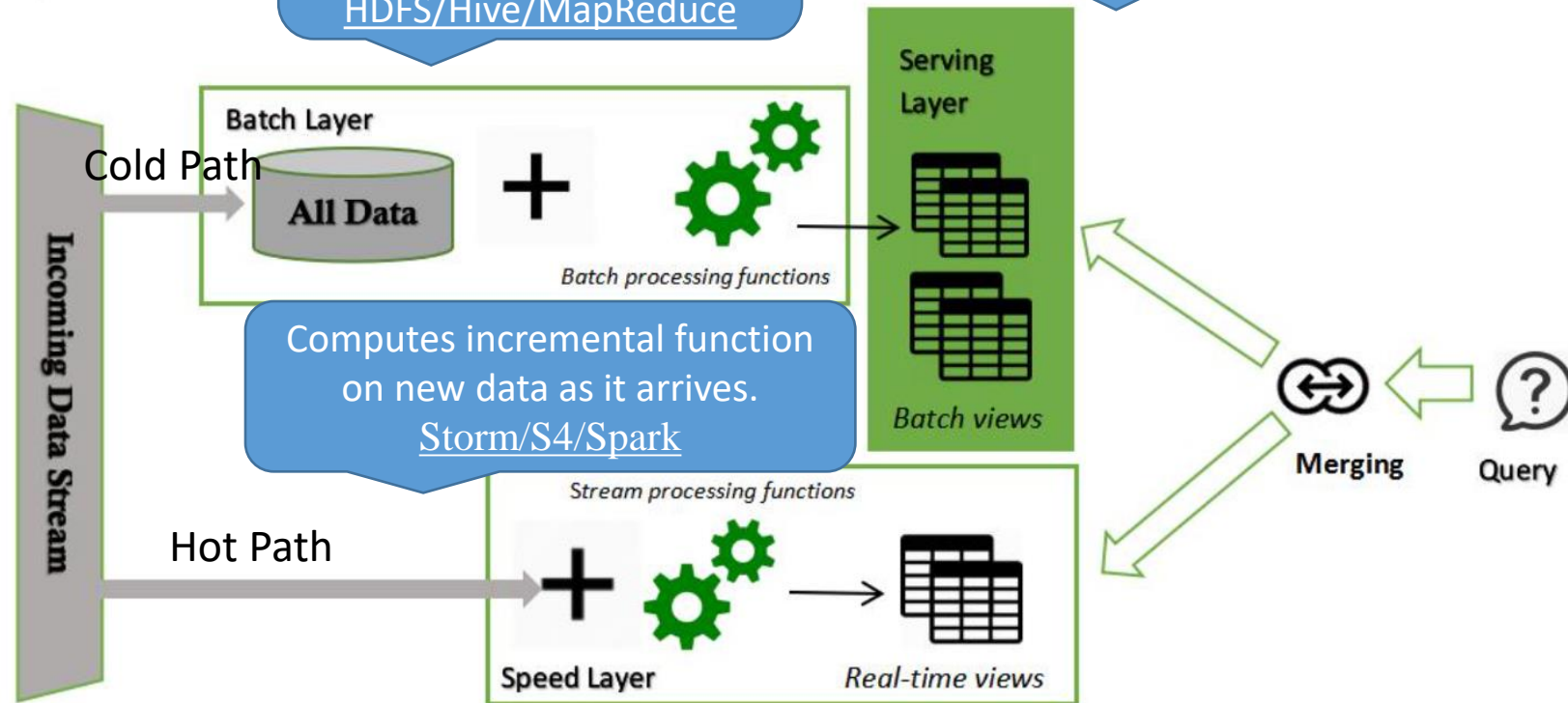
- Lambda
- Kappa
- Microservice
- Zeta
- IoT



# Lambda Architecture

A distributed file system  
which stores the entirety of  
the collected data.  
HDFS/Hive/MapReduce

Contains batch views to be  
interactively queried by the  
user  
Any NoSQL Database



# Lambda Architecture

## Advantages

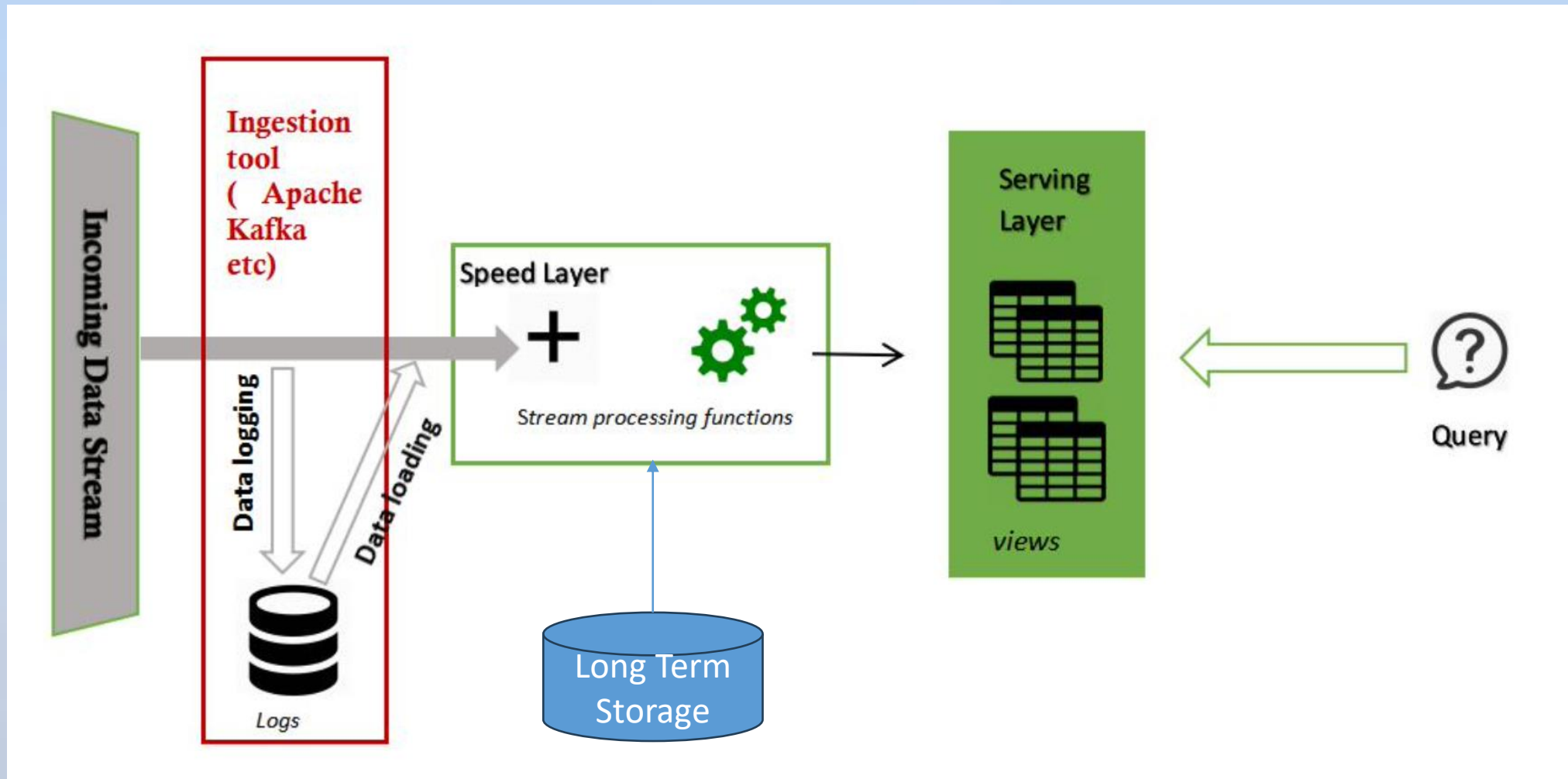
- Better accuracy
- Higher throughput
- Lower latency
- Resilient
- Fault tolerant

## Challenges

- Synchronization of batch and speed layer
- Need to maintain two separate codebase



# Kappa Architecture



# Kappa Architecture

## Advantages

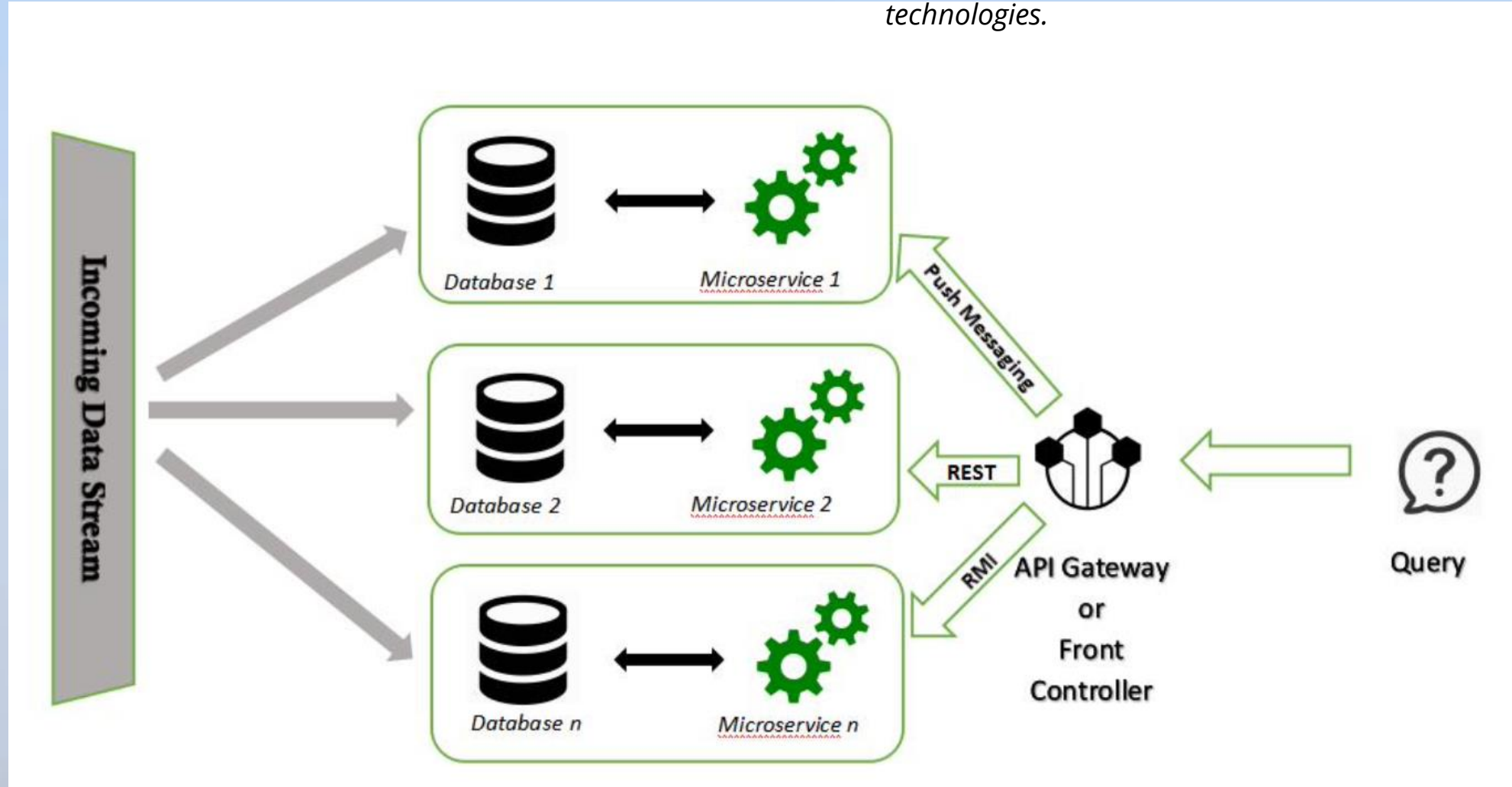
- Simplified architecture
- Single codebase

## Challenges

- Only analytical operations are possible not transactional ones.
- It is important to know that the data is not conserved for a long term; data is kept for a limited predefined period after which it is discarded.

# Microservice Architecture

The microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.



REST:  
Representational  
State Transfer

RMI: Remote  
Method Invocation

# Microservice Architecture

## Advantages

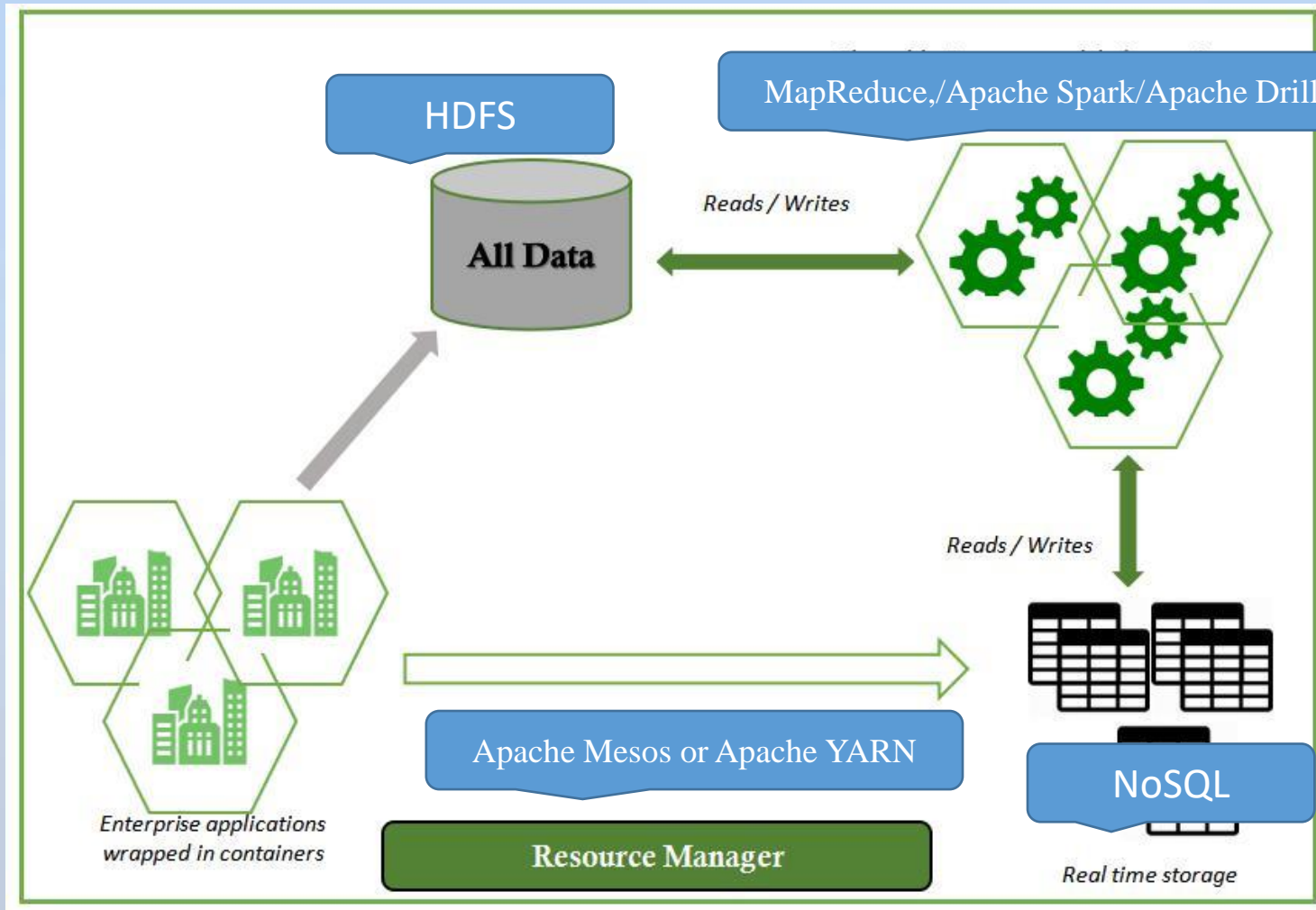
- Faster development, testing, and deployment
- Fault tolerant
- Limited vendor/technology lock-in
- Easy onboarding and maintenance

## Challenges

- An inter-service communication mechanism is required, and its development is quite complex.
- Though the deployments are faster, they are more complex to setup



# Zeta Architecture



The goal of the container is to guarantee a single, standardized method of deployment. It also implies that deployed resources are isolated containers that don't concern about any environment changing, i.e. they deploy in the same manner in local environment as in prod environment. Thanks to this isolation, the containers can be freely moved between machines with the guarantee of repeatability (results on local server will be the same as on prod's one). A famous example of isolated containers is Docker, but can also be used Kubernetes or Mesos.

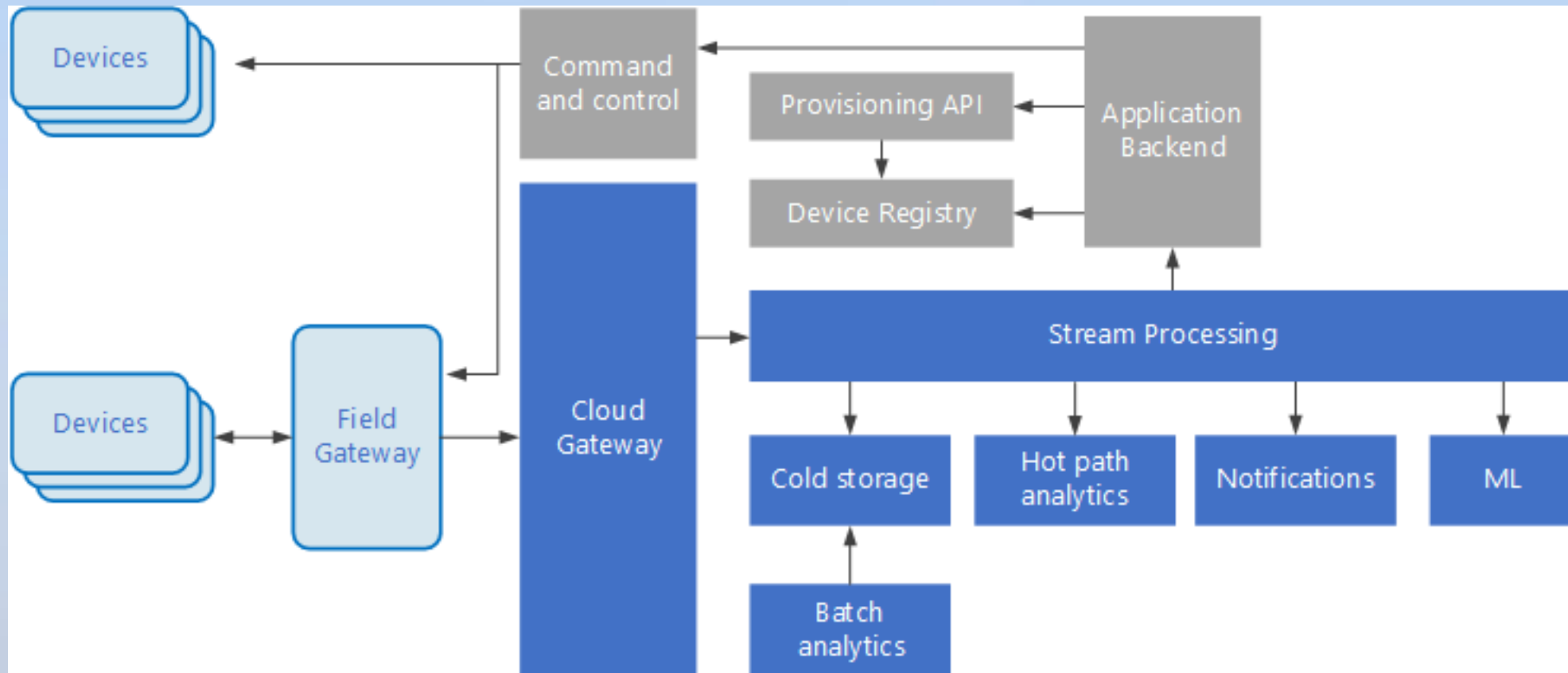
# Zeta Architecture

## **Advantages**

- Better utilization of hardware
- Near real-time backup



# IoT Architecture



| Architectures<br>Features | Lambda  | Kappa  | lot-a  | Microservice<br>s                          | Zeta  |
|---------------------------|---|--|--|--|---|
| Analysis type             | Batch/Real-time                                 | Real-time                                      | Batch / Real-time                                    | Batch/ Real-time                           | Batch/ Real-time                            |
| Processing methodology    | Query and reporting                             | Query and reporting                            | Query and reporting/ Analytical/ Predictive analysis | Query and reporting/ Analytical            | Query and reporting                         |
| Data frequency            | Real-time feeds                                 | Continuous feeds                               | On-demand feeds                                      | On-demand feeds                            | On- demand feeds                            |
| Data type                 | Master data                                     | Transactional                                  | Master data  | Transactional data                         | Transactional data                          |
| Content format            | Structured , Semi-structured & Unstructured     | Structured, Semi-structured & Unstructured     | Structured, Semi-structured & Unstructured           | Structured, Semi-structured & Unstructured | Structured, Semi-structured & Unstructured  |
| Data sources              | Human & Machine generated , web or social media | Machine & Human generated, Web or social media | Machine generated                                    | Internal data sources, machine generated   | Web and social media, Internal Data sources |
| Data consumers            | Human   | Human  | Human/ Other data repositories                       | Business process                           | Enterprise applications                     |