CONTENT:

- Tiny ML
- Low power ML
- Power BI
- DevOps (life cycle, benefits and challenges)
- Distributed db using AI

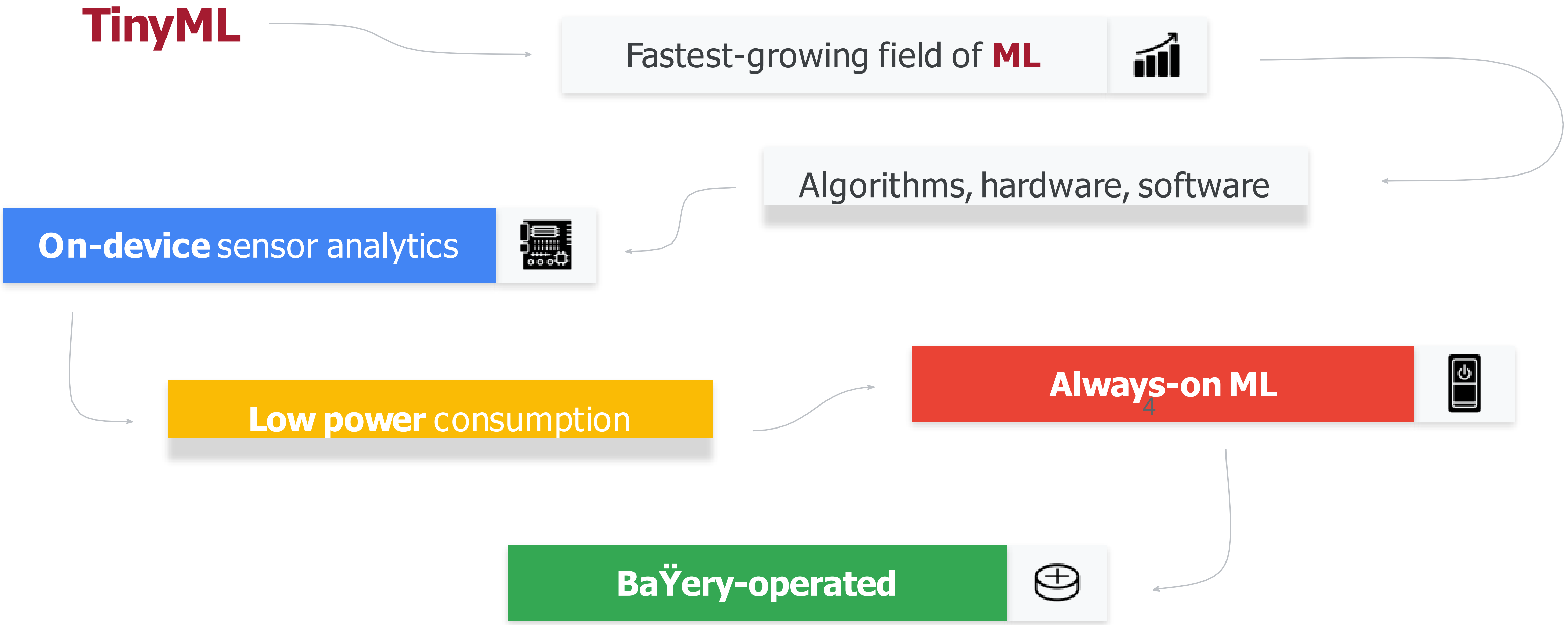
Tiny ML:

- Tiny machine learning (tiny-ML) is the intersection of machine learning and embedded internet of things (IoT) devices.
- The field is an emerging engineering discipline that has the potential to revolutionize many industries.
- Tiny-ML is a field of study in Machine Learning and Embedded Systems that explores the types of models you can run on small, low-powered devices like microcontrollers.
- It enables low-latency, low power and low bandwidth model inference at edge devices.

Tiny ML vs ML

- TinyML is a branch of machine learning that uses small form factor devices attached with sensors including bare metal or RTOS based microcontrollers to execute ML models and algorithms as software.
- Tiny in TinyML means that the ML models are optimized to run on very low-power and small footprint devices, such as various MCUs
- TinyML is capable of running machine-learning models on microcontrollers and embedded systems without relying on the cloud or other resources. This is because the models are highly optimized through techniques such as quantization, compression, and pruning, which reduces their size and complexity.
- Machine learning is best suited to applications that use lots of data or millions of examples, such as sound recordings, sensor logs, ATM transactions, and so forth

What is Tiny Machine Learning (TinyML)?



TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

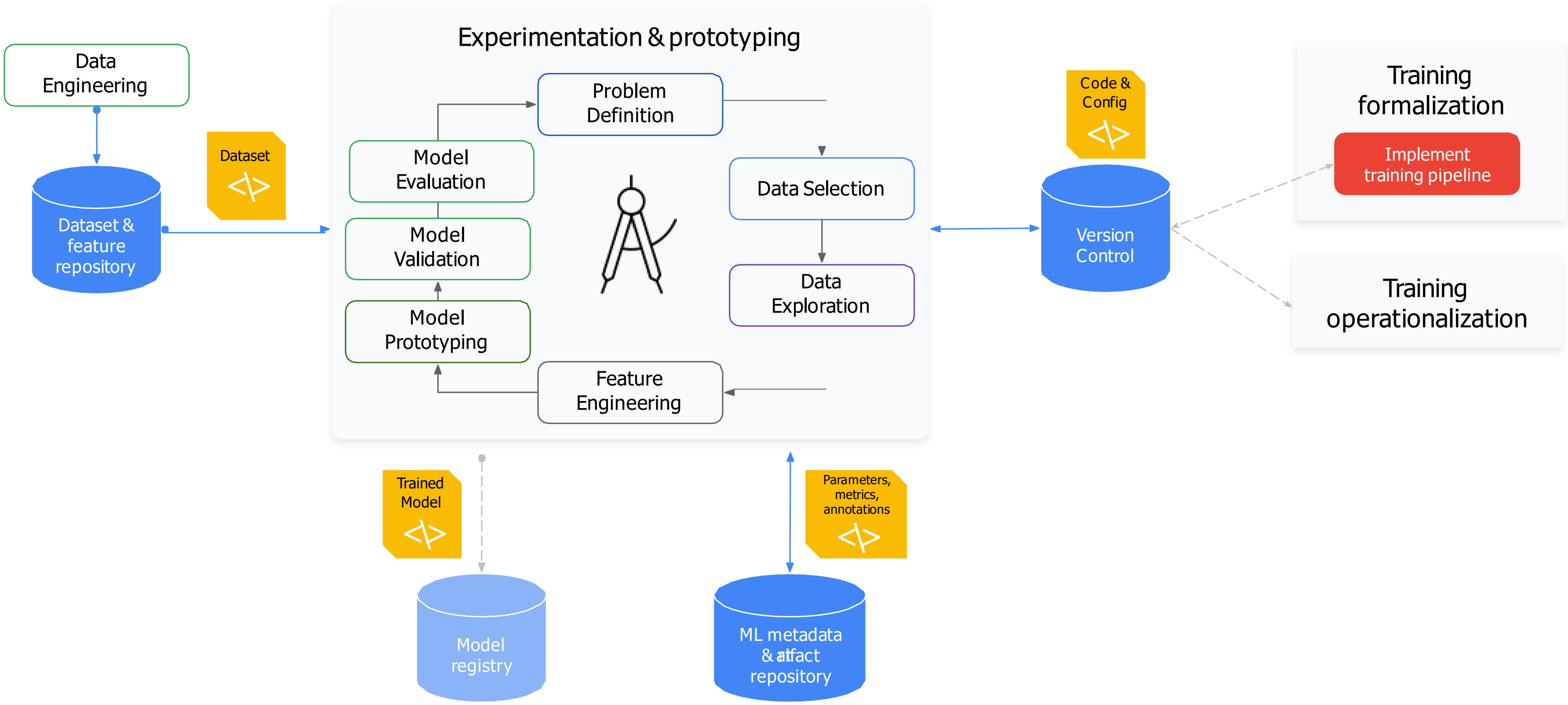
On-device sensor analytics

Low power consumption

Always-on ML

BaŸery-operated

ML Development



Data Engineering

Dataset & feature repository

Dataset

Experimentation & prototyping

Problem Definition

Data Selection

Data Exploration

Feature Engineering

Model Prototyping

Model Validation

Model Evaluation

Code & Config

Version Control

Training formalization

Implement training pipeline

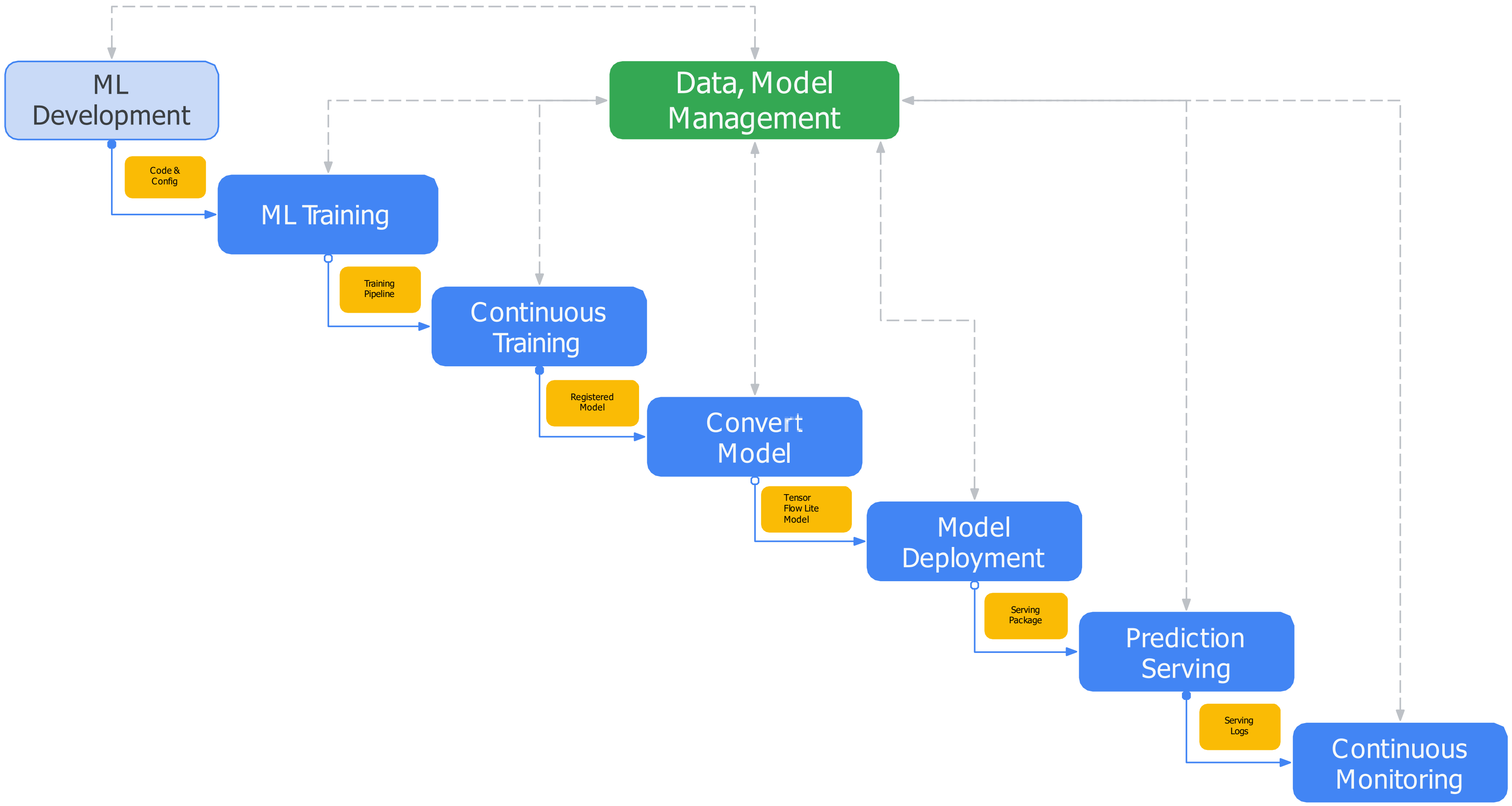
Training operationalization

Trained Model

Model registry

Parameters, metrics, annotations

ML metadata & aitfact repository



ML Development

Code & Config

ML Training

Training Pipeline

Continuous Training

Registered Model

Convert Model

Tensor Flow Lite Model

Model Deployment

Serving Package

Prediction Serving

Serving Logs

Continuous Monitoring

Data, Model Management

Advantages of Tiny-ML

1. **Low Latency:** Since the model runs on the edge, the data doesn't have to be sent to a server to run inference. This reduces the latency of the output.
2. **Low Power Consumption:** As we discussed before, microcontrollers consume very little power. This enables them to run without being charged for a really long time.
3. **Low Bandwidth:** As the data doesn’t have to be sent to the server constantly, less internet bandwidth is used.
4. **Privacy:** Since the model is running on the edge, your data is not stored in any servers.

Applications of Tiny-ML

By summarizing and analyzing data at the edge on low power devices, TinyML offers many unique solutions. Even though TinyML is an emerging field, it has been used in production for years.

- **Industrial Predictive Maintenance:** Machines are prone to fault. Using TinyML on low powered devices, it is possible to monitor the machine and predict faults ahead of time constantly.
- **Healthcare:** The Solar Scare Mosquito project uses TinyML to curb the spread of mosquito-borne diseases like Dengue, Malaria, Zika Virus, Chikungunya, etc.
- It works by detecting the mosquito breeding conditions and agitates the water to prevent mosquito breeding

Applications of Tiny-ML

- **Agriculture:** The Nuru app helps farmers detect diseases in plants just by taking a picture of it by running Machine Learning models on the device using TensorFlow Lite. Since it works on the device, there is no need for an internet connection.
- **Ocean Life Conservation:** Smart ML-powered devices are used to monitor whales in real-time in waterways around Seattle

Requirements for Low-Power Machine Learning Inference for IoT

- IoT edge devices that employ low-power machine learning inference applications typically perform different types of processing, as shown in Figure 1.

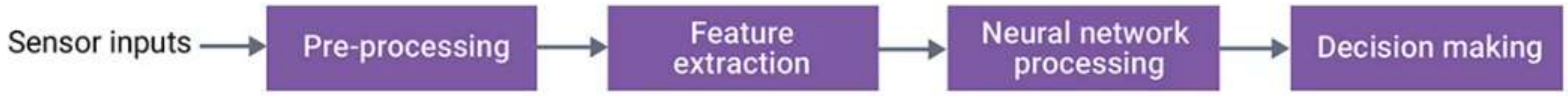


Figure 1. Different types of processing in machine learning inference applications

Requirements for Low-Power Machine Learning Inference for IoT

- These devices typically perform some pre-processing and feature extraction on the sensor input data before doing the actual neural network processing for the trained model.
- For example, a smart speaker with voice control capabilities may first pre-process the voice signal by performing acoustic echo cancellation and multi-microphone beam-forming.
- Then it may apply FFTs to extract the spectral features for use in the neural network processing, trained to recognize a vocabulary of voice commands.

Implementation Requirements

- Low/mid-end machine learning inference applications require the following types of processing:
- Various types of pre-processing and feature extraction, often with DSP intensive computations
- Neural Network Processing.
- Decision making, performed after the neural network processing, is more control oriented

POWER BI:

- Microsoft Power BI is a business intelligence (BI) platform that provides nontechnical business users with tools for aggregating, analyzing, visualizing and sharing data.
- Power BI's user interface is fairly intuitive for users familiar with excel, and its deep integration with other Microsoft products makes it a versatile self-service tool that requires little upfront training.

Common uses of Power BI

- Microsoft Power BI is used to find insights within an organization's data.
- Power BI can help connect disparate data sets, transform and clean the data into a data model and create charts or graphs to provide visuals of the data.
- All of this can be shared with other Power BI users within the organization.
- Power BI can also provide executive dashboards for administrators or managers, giving management more insight into how departments are doing.

Key features of Power BI

- **Artificial intelligence**. Users can access image recognition and text analytics in Power BI, create machine learning models using automated ML capabilities and integrate with Azure Machine Learning.
- **Hybrid deployment support**. This feature provides built-in connectors that allow Power BI tools to connect with a number of different data sources from Microsoft, Salesforce and other vendors.
- **Quick Insights.** This feature allows users to create subsets of data and automatically apply analytics to that information.

Key features of Power BI

- **Common data model support**. Power BI's support for the common data model allows the use of a standardized and extensible collection of data schemas (entities, attributes and relationships).
- **Cortana integration**. This feature, which is especially popular on mobile devices, allows users to verbally query data using natural language and access results using Cortana, Microsoft's digital assistant.
- **Customization**. This feature allows developers to change the appearance of default visualization and reporting tools and import new tools into the platform.
- **APIs for integration**. This feature provides developers with sample code and application program interfaces (APIs) for embedding the Power BI dashboard in other software products.

BI components

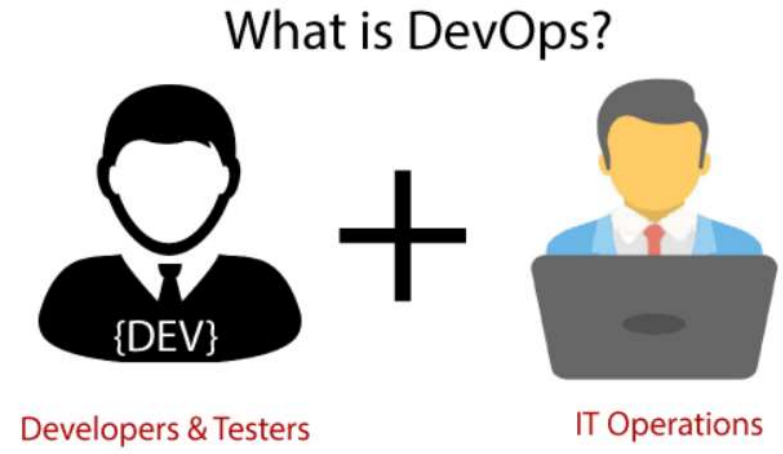
- Included within Power BI are several components that help users create and share data reports. Those are the following:
- **Power Query:** a data mashup and transformation tool
- **Power Pivot:** a memory tabular data modeling tool
- **Power View:** a data visualization tool
- **Power Map:** a 3D geospatial data visualization tool
- **Power Q&A:** a natural language question and answering engine

How to use Power BI

- To build a Power BI report, take the following steps:
- Connect the data sources.
- Query the data to create reports based on user needs.
- Publish the report to the Power BI service.
- Share the report, so cloud and mobile users can see and interact with it.
- Add permissions to give colleagues the ability to edit reports or create dashboards or limit their ability to edit.

What is DevOps?

- The DevOps is a combination of two words, one is software Development, and second is Operations. This allows a single team to handle the entire application lifecycle, from development to **testing, deployment**, and **operations**.
- DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrat



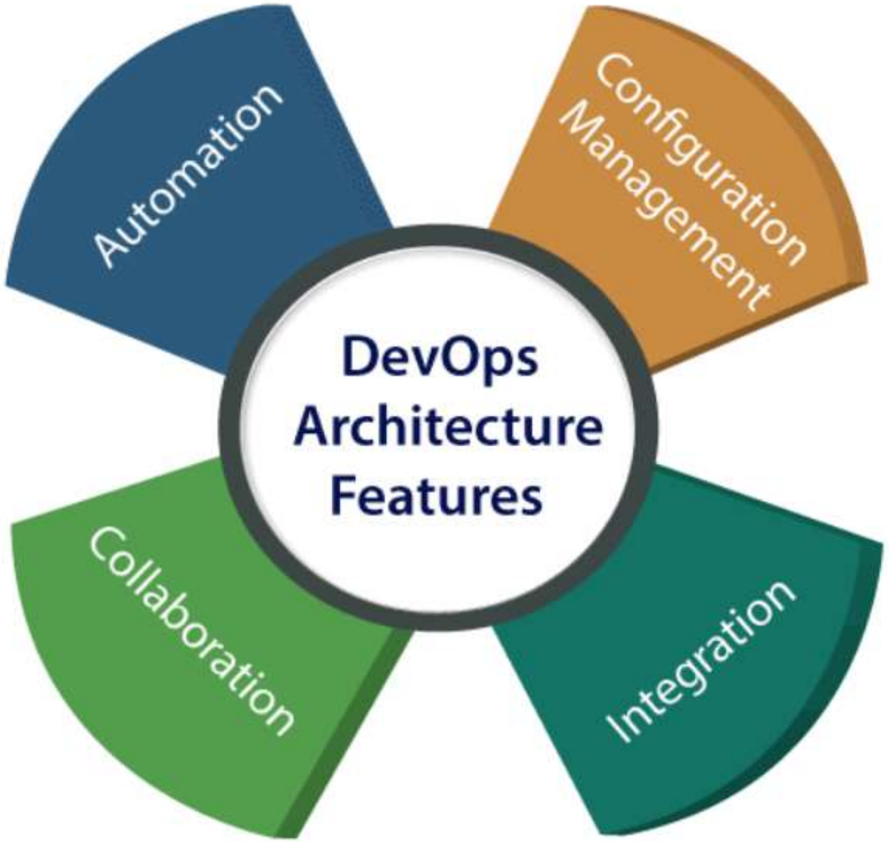
What is DevOps?

- DevOps promotes collaboration between Development and Operations team to deploy code to production faster in an automated & repeatable way.
- DevOps helps to increase organization speed to deliver applications and services. It also allows organizations to serve their customers better and compete more strongly in the market.
- DevOps can also be defined as a sequence of development and IT operations with better communication and collaboration.
- DevOps has become one of the most valuable business disciplines for enterprises or organizations. With the help of DevOps, **quality**, and **speed** of the application delivery has improved to a great extent.

Why DevOps?

- The operation and development team worked in complete isolation.
- After the design-build, the testing and deployment are performed respectively. That's why they consumed more time than actual build cycles.
- Without the use of DevOps, the team members are spending a large amount of time on designing, testing, and deploying instead of building the project.
- Manual code deployment leads to human errors in production.
- Coding and operation teams have their separate timelines and are not in synch, causing further delays.

DevOps Architecture Features



DevOps Architecture Features

1) Automation

- Automation can reduce time consumption, especially during the testing and deployment phase. The productivity increases, and releases are made quicker by automation. This will lead in catching bugs quickly so that it can be fixed easily.

2) Collaboration

- The Development and Operations team collaborates as a DevOps team, which improves the cultural model as the teams

3) Integration

- Applications need to be integrated with other components in the environment. The integration phase is where the existing code is combined with new functionality and then tested.

4) Configuration management

- It ensures the application to interact with only those resources that are concerned with the environment in which it runs.

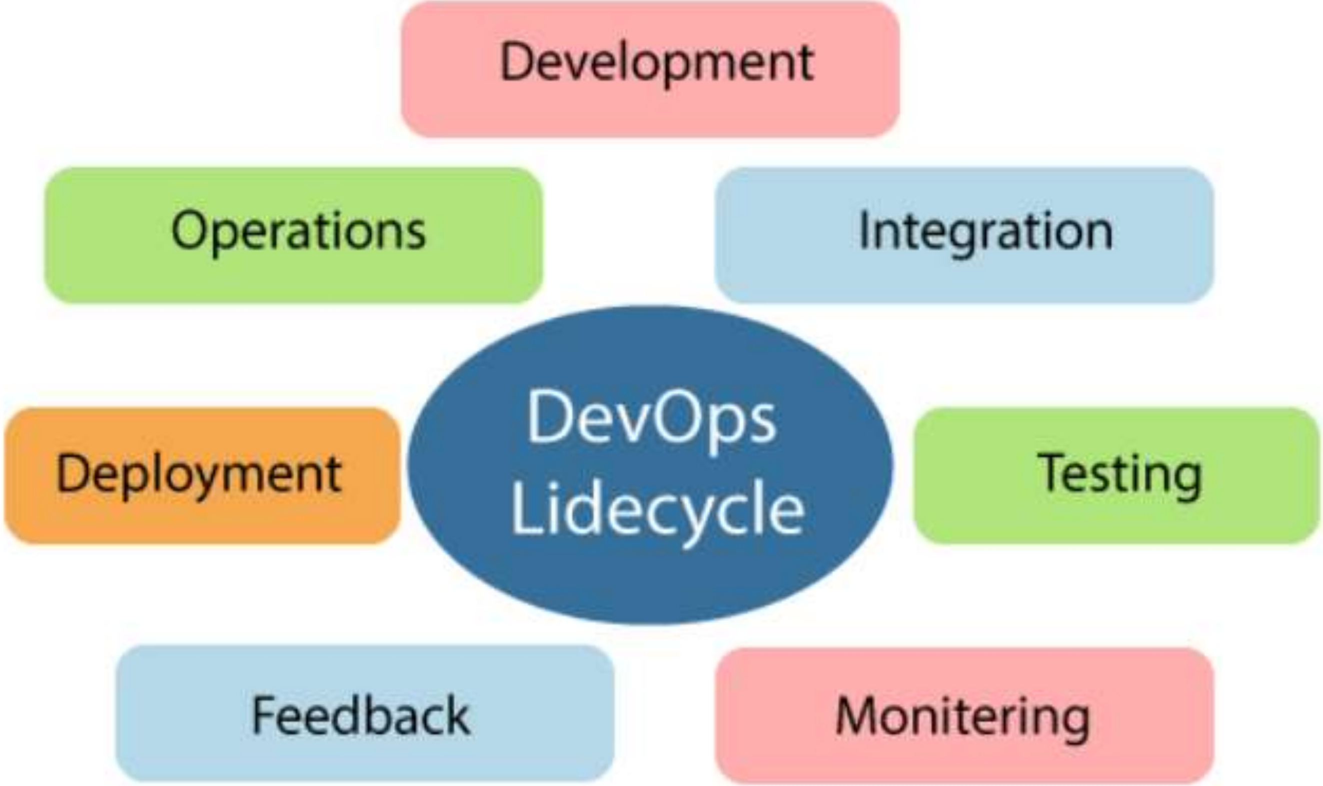
Advantages

- DevOps is an excellent approach for quick development and deployment of applications.
- It responds faster to the market changes to improve business growth.
- DevOps escalate business profit by decreasing software delivery time and transportation costs.
- DevOps clears the descriptive process, which gives clarity on product development and delivery.
- It improves customer experience and satisfaction.
- DevOps simplifies collaboration and places all tools in the cloud for customers to access.

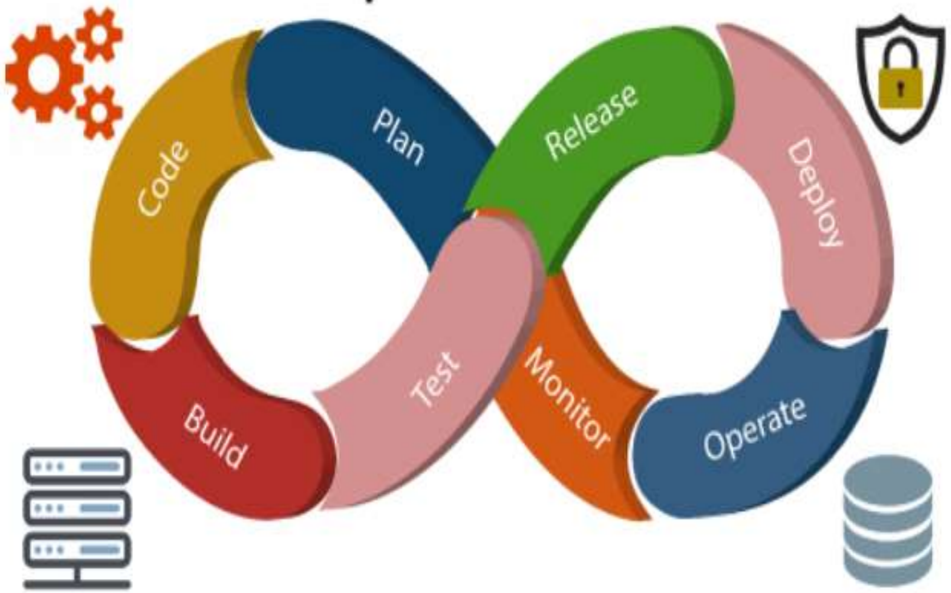
Disadvantages

- DevOps professional or expert's developers are less available.
- Developing with DevOps is so expensive.
- Adopting new DevOps technology into the industries is hard to manage in short time.
- Lack of DevOps knowledge can be a problem in the continuous integration of automation projects.

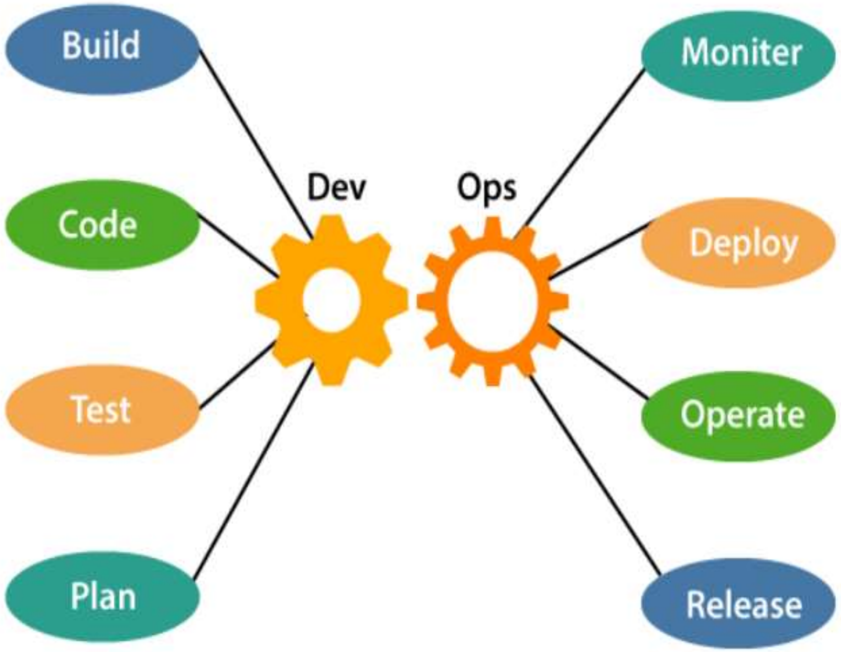
DevOps Lifecycle



DevOps Architecture



DevOps Components



DevOps Principles

1. **End to End Responsibility:** DevOps team need to provide performance support until they become the end of life. It enhances the responsibility and the quality of the products engineered.
2. **Continuous Improvement:** DevOps culture focuses on continuous improvement to minimize waste. It continuously speeds up the growth of products or services offered.
3. **Automate Everything:** Automation is an essential principle of the DevOps process. This is for software development and also for the entire infrastructure landscape.
4. **Custom Centric Action:** DevOps team must take customer-centric for that they should continuously invest in products and services.
5. **Monitor and test everything:** The DevOps team needs to have robust monitoring and testing procedures.
6. **Work as one team:** In the DevOps culture role of the designers, developers, and testers are already defined. All they needed to do is work as one team with complete collaboration.

DevOps challenges

1. Manual testing
2. No DevOps center of excellence
3. Test data
4. Manual deployments
5. Planning in a DevOps environment
6. DevOps and suppliers
7. DevOps and governance
8. No integrated tools architecture
9. Limited transparency
10. Manual processes
11. Collaboration between development and operations

Distributed AI

- Distributed AI is a computing paradigm that bypasses the need to move vast amounts of data and provides the ability to analyze data at the source.
- Gartner, a global provider of business insights, estimates that by 2025, 75 percent of data will be created and processed outside the traditional data center or cloud.
- This explosion of data being generated by people and machines from mobile devices, Internet of Things (IoTs), and machine data from production floors makes us rethink where computing needs to be performed.

Distributed AI

- Distributed computing solves many of these challenges by bringing computation, storage, network, communication, power, and application functions closer to data sources at the point of action.
- Additionally, it enables businesses to tighten data security with capabilities to control and comply with privacy regulations.

Distributed AI

AI can be a game changer in addressing this, but there are a few major problems:

1. Too much data that needs to be collected, which overburdens networks and generates latency issues
2. Too little data available to drive meaningful insights because the assets are not connected and monitored

Distributed AI

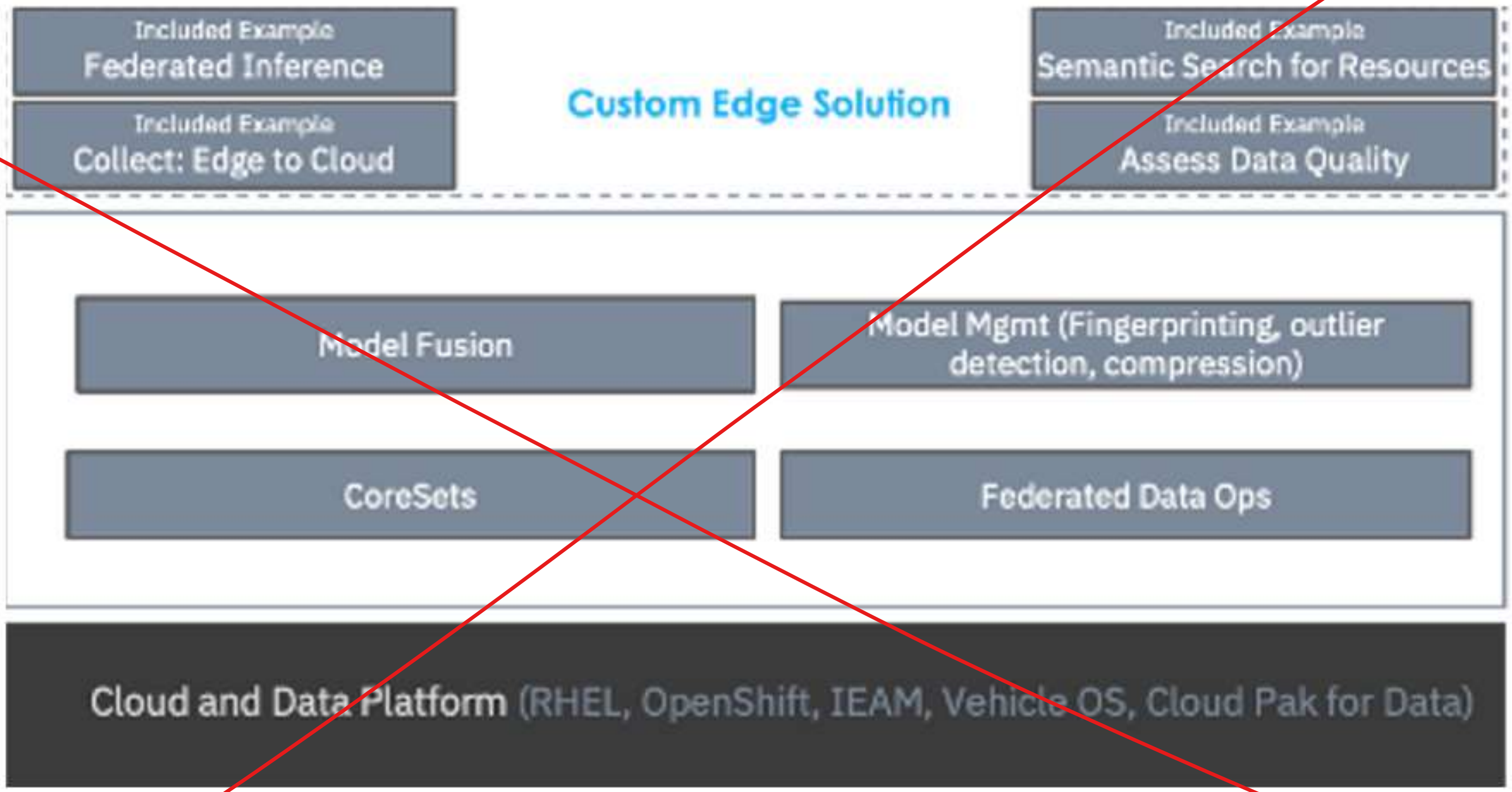
- Distributed AI, a computing paradigm that bypasses the need to move vast amounts of data and provides the ability to analyze data at the source.
- Distributed computing solves many challenges by bringing computation, storage, network, communication, power, and application functions closer to data sources at the point of action.
- Additionally, it enables businesses to tighten data security with capabilities to control and comply with privacy regulations.

~~High-level architecture~~

~~The Distrubuted AI APIs contain the following APIs:~~

- ~~CoreSets API: Algorithms to intelligently sub-sample data before transporting over the network~~
- ~~Federated DataOps API: Algorithms to assess the data quality and repair labels and missing values~~
- ~~Model Fusion API: Algorithms to train the models while data is resident across many sites~~
- ~~Model Management API: Algorithms to maintain the most suitable models for each site~~
- ~~Model Compression API: Algorithms to compress models based on a resource constrained site~~

High-level architecture



Included Example
Federated Inference

Included Example
Collect: Edge to Cloud

Custom Edge Solution

Included Example
Semantic Search for Resources

Included Example
Assess Data Quality

Model Fusion

Model Mgmt (Fingerprinting, outlier detection, compression)

CoreSets

Federated Data Ops

Cloud and Data Platform (RHEL, OpenShift, IEAM, Vehicle OS, Cloud Pak for Data)