

oneM2M IoT SL and AI/ML use

In Traditional Programming, the Input Data and a Program are fed into a Machine to generate Output. When it comes to Machine Learning (ML), Input Data and Expected Output are fed into the Machine during the Learning Phase, and it works out a Program for itself. To understand this better, refer to the Figure below:

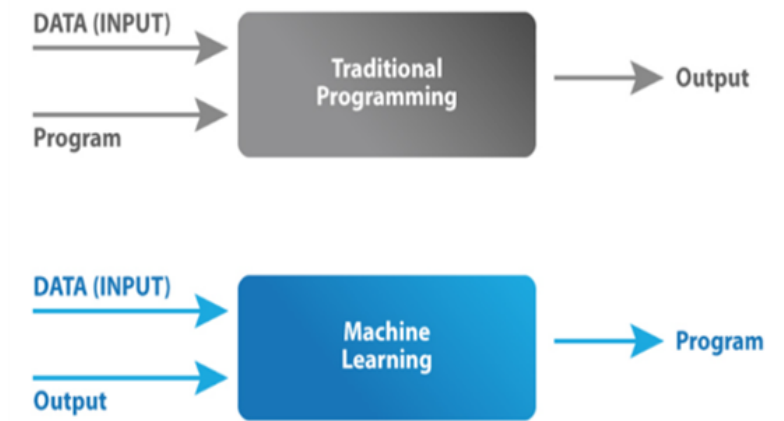


Figure: Traditional Programming vs Machine Learning

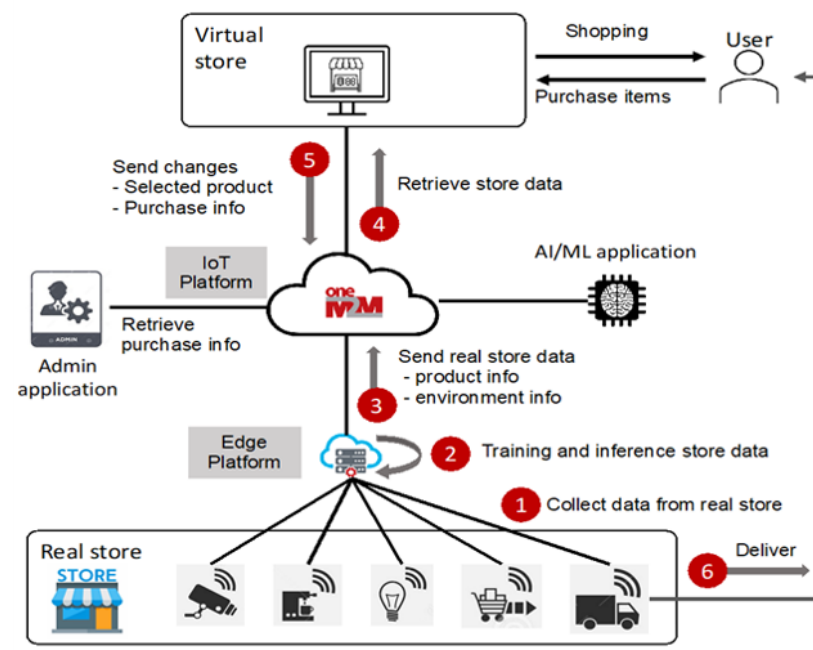


Figure: Conceptual Diagram of Metaverse Virtual Store in oneM2M IoT and Edge Platforms

The word "Metaverse" is made up of the prefix "Meta" (meaning "beyond") and the stem "verse" (a back-formation from "universe"); the term is typically used to describe the Concept of a Future Iteration of the Internet, made up of persistent, shared, 3D virtual spaces linked into a perceived virtual universe. The Metaverse in a broader concept refer to realize Virtual Worlds using IoT, AI and Augmented Reality(AR)/Virtual Reality(VR).

A Metaverse-based Online Store where Stores in the real world are created as Digital Twins in the Metaverse Virtual Space, and Users visit a Virtual Store in the Metaverse Space to purchase preferred products. For Real-Time Synchronization between the Real-World and the Virtual Stores in the Metaverse, various Smart Sensors are used to sense Real-World Products intelligently.

The Edge Node at the Real World store loads a trained AI/ML Model and infers Products' information. The retrieved product Data is then transferred to the IoT Platform for Real-Time Synchronization.

A User can now purchase Products from a Virtual Store in the Metaverse. The purchase info in the Metaverse is notified to the Administrator and the purchased product is delivered to the user.

The flow of Data has an important bearing on Architectural Components for IoT Solutions, that is part of the analysis of the impact of AI on IoT Architectures, in particular the oneM2M Service Layer. The typical focus is on Data that leads to some form of decision being taken and is classified as 'User-Plane' (UP) Data. The basic Model for IoT Solutions begins with sourcing Data from IoT Devices (illustrated in left-hand side of Figure 1 below). This Data then passes through a Signal Processing and Machine Learning (ML) Process to extract key features and to represent them as Knowledge-based Objects. The next stage of processing involves the Application of Rules-based AI in areas related to Reasoning, Decision making, Supervision and Explainable AI.

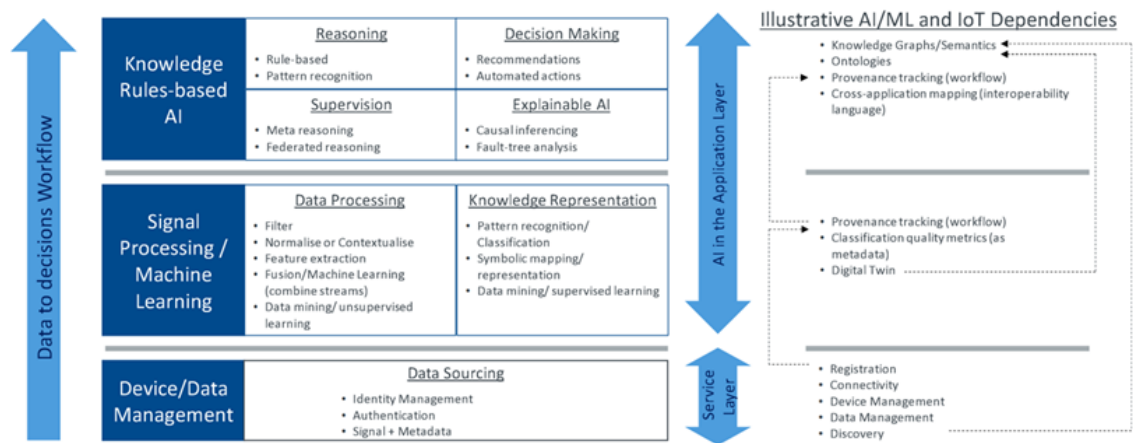


Figure 1: IoT Data to Decisions Workflow

This workflow illustrates some of the Generic and Commonly used Data Sourcing and AI/ML Capabilities involved in supporting End-to-End (E2E) IoT Solutions. It also shows how AI/ML Capabilities depend on Service Layer Capabilities. An example (illustrated in right-hand side of Fig. 1) is the relationship between a **Registration Capability that manages the Identify of a Device and its Value in providing information about the Provenance of Data used in Pattern Recognition or Causal Inferencing Functions**. In this example, the act of tracking data provenance depends on a 'Registration' Service Capability. Provenance tracking can improve the quality and dependability of an AI/ML system and occurs in the background to 'User Plane' (UP) activity. In Architectural terms, such background processes and use of Data that improve the Quality of AI/ML Applications occur in what is referred to as the 'Control Plane' (CP).

A traditional approach (illustrated in Fig 2 below) relies on an Application Ingesting and Storing Data for Processing. The SW implementation concentrates AI/ML and Service Layer Capabilities in the Application Layer. This places a burden on Solution Developers to master Application, AI and Service Layer disciplines.

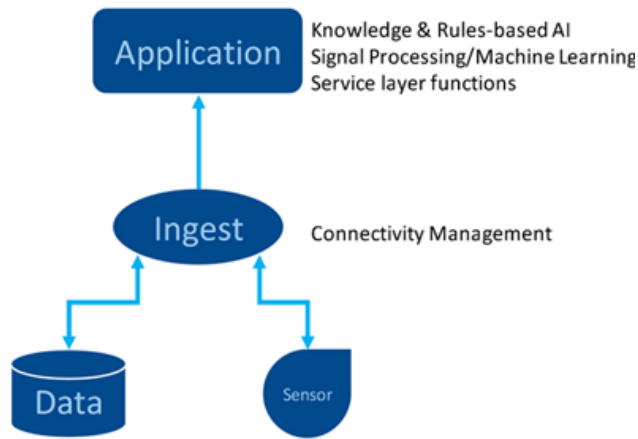


Figure 2: Traditional approach for IoT solutions

The alternative is a Developer friendly Approach (illustrated in Fig 3 below). This approach provides developers with an **Abstraction Layer - i.e. a Common Service Layer** - that makes AI and the more usual IoT-enabling Services accessible through a Standardized API. This arrangement means that the IoT Application can rely on notifications from the Common Services Layer to trigger its Functions when notified of changes in IoT Data. **It can also draw on a Library of AI-enabled Services provided within the Common Services Layer.**

A Three-Tier Framework to organize the logical aspects of AI in IoT maps AI Applications into a 'User' Plane (UP).

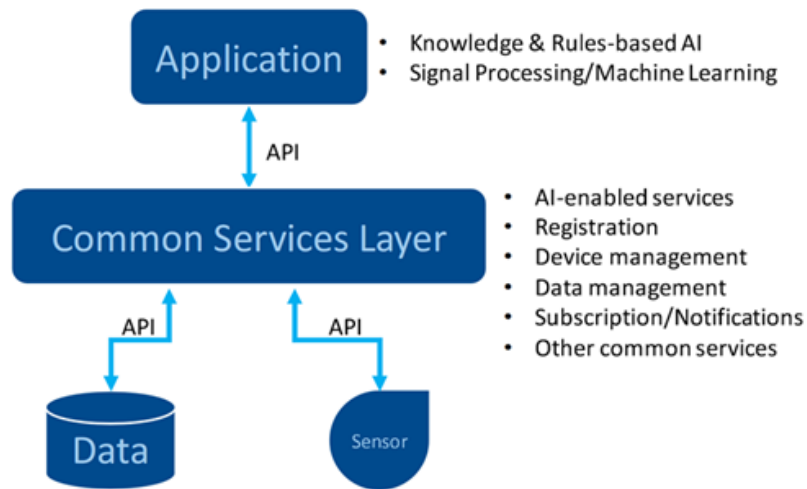


Figure 3: Scalable and developer-friendly approach for IoT solutions