ITUPublications
Recommendations

International Telecommunication Union
Standardization Sector

Recommendation

ITU-T F.748.17 (12/2022)

SERIES F: Non-telephone telecommunication services

Multimedia services

Technical specification for artificial intelligence cloud platform – Artificial intelligence model development

ITU-T F-SERIES RECOMMENDATIONS

NON-TELEPHONE TELECOMMUNICATION SERVICES

TELEGRAPH SERVICE	
Operating methods for the international public telegram service	F.1–F.19
The gentex network	F.20–F.29
Message switching	F.30–F.39
The international telemessage service	F.40–F.58
The international telex service	F.59–F.89
Statistics and publications on international telegraph services	F.90–F.99
Scheduled and leased communication services	F.100–F.104
Phototelegraph service	F.105–F.109
MOBILE SERVICE	
Mobile services and multidestination satellite services	F.110–F.159
TELEMATIC SERVICES	
Public facsimile service	F.160–F.199
Teletex service	F.200–F.299
Videotex service	F.300–F.349
General provisions for telematic services	F.350–F.399
MESSAGE HANDLING SERVICES	F.400–F.499
DIRECTORY SERVICES	F.500–F.549
DOCUMENT COMMUNICATION	
Document communication	F.550–F.579
Programming communication interfaces	F.580–F.599
DATA TRANSMISSION SERVICES	F.600–F.699
MULTIMEDIA SERVICES	**F.700–F.799**
ISDN SERVICES	F.800–F.849
UNIVERSAL PERSONAL TELECOMMUNICATION	F.850–F.899
ACCESSIBILITY AND HUMAN FACTORS	F.900–F.999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T F.748.17

Technical specification for artificial intelligence cloud platform – Artificial intelligence model development

Summary

Recommendation ITU-T F.748.17 provides a framework for the cloud-based development of artificial intelligence (AI) models. It covers the terminology, features, and reference design of an AI cloud platform to enable the development of AI models. It establishes the technical specifications of the platform's supporting functional modules, core functional modules, and auxiliary functional modules.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T F.748.17	2022-12-14	16	11.1002/1000/15194

Keywords

Artificial intelligence, cloud platform, model development.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T F.748.17

Technical specification for artificial intelligence cloud platform – AI model development

1 Scope

This Recommendation gives a reference architecture and functional requirements for artificial intelligence (AI) model development in an AI cloud platform.

In particular, the scope of this Recommendation includes:

- A reference architecture for the AI model development framework in an AI cloud platform;
- The descriptions of the modules that should be included in an AI cloud platform to facilitate AI model development; and
- The minimum and recommended functional requirements for AI model development that an AI cloud platform should satisfy.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.3531] Recommendation ITU-T Y.3531 (2020), *Cloud computing – Functional requirements for machine learning as a service*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 artificial intelligence (AI) [b-ISO/IEC 2382]: An interdisciplinary field, usually regarded as a branch of computer science, dealing with models and systems for the performance of functions generally associated with human intelligence, such as reasoning and learning.

3.1.2 algorithm [b-ISO/IEC 11557]: A set of rules for transforming the logical representation of data.

3.1.3 container [b-ITU-T Y.3535]: A set of software to provide isolation, resource control, and portability for the virtualization processing of an application.

NOTE 1 – A container runs on the kernel in a bare-metal machine or virtual machine.

NOTE 2 – "Application" implies business logic including a required library or binary to run in a container.

3.1.4 dataset [b-ISO 19115-1]: Identifiable collection of data.

3.1.5 model [b-ITU-T F.748.16]: An output created by an algorithm running on training data that can generate an inference or prediction, based on the input data.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 automated machine learning: The process of automating the time consuming, iterative tasks of machine learning model development. It allows developers to build machine learning models automatically with high scale, efficiency, and productivity all while sustaining model quality.

3.2.2 built-in: Indicating that an object is provided by the platform.

3.2.3 customized: Indicating that an object is created or uploaded by the user.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AI Artificial Intelligence

AutoML Automated Machine Learning

CPU Central Processing Unit

GPU Graphics Processing Unit

HPO Hyper-Parameter Optimization

IDE Integrated Development Environment

NAS Neural Architecture Search

NPU Neural-Network Processing Unit

VM Virtual Machine

5 Conventions

In this Recommendation:

- The keywords "**is required to**" indicate a requirement that must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.
- The keywords "**is recommend to**" indicate a requirement that is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.
- The keywords "**optional**" indicate an optional requirement that is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 Overview

In essence, the artificial intelligence (AI) model is a function, and instead of relying on a logic inference or induction, the parameters of this function are obtained through training. The goal of the AI model development process is to fit the model structure and the parameters closer to the objective function by algorithm design and model training.

An AI cloud platform integrates the necessary modules for data management, algorithm development, model training and model deployment, to meet the full life cycle requirements of the AI model development and production in a single cloud platform. The AI cloud platform's deployment guidelines are detailed in Appendix I. Figure 6-1 illustrates a recommended AI model development framework in an AI cloud platform.

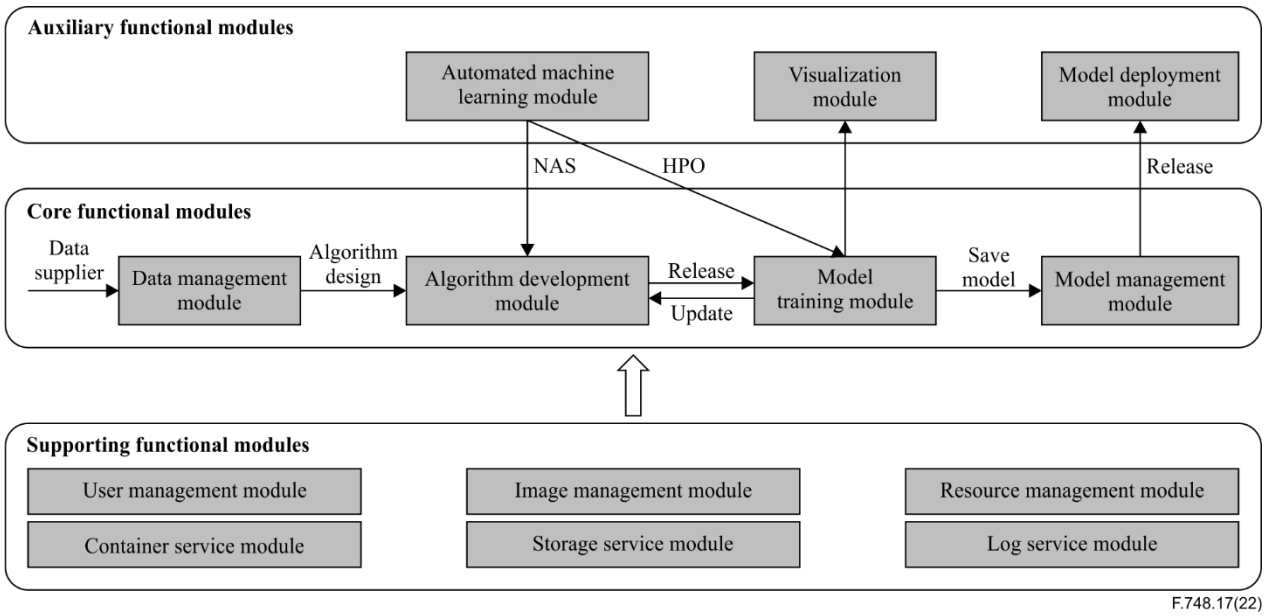

Figure 6-1 – AI model development framework

The framework given here is a description of the essential operations of an AI cloud platform for AI model development. The supporting functional modules on the bottom layer make up the AI cloud platform's infrastructure and enable the platform's fundamental operation, maintenance, resource management and scheduling, etc. The core functional modules depicted on the middle layer run through the entire AI model development process, including data management, algorithm development, model training and model deployment. The auxiliary functional modules are displayed on the top layer, which enhances the AI cloud platform's functionality for AI model development.

The supporting functional modules and the core functional modules are mandatory, the auxiliary functional modules are optional.

Typically, an AI cloud platform includes but is not limited to the following modules:

- **User management module**: This module is to manage the user information.
- **Image management module**: This module is to manage the images for development, training and inference.
- **Resource management module**: This module is to manage the computing resources.
- **Container service module**: This module is to manage the containers for providing the execution environment.
- **Storage service module**: This module is to manage the storage resources.
- **Log service module**: This module is to manage the log service.
- **Data management module**: This module is to process and manage the datasets.
- **Algorithm development module**: This module is to support the development of algorithms.
- **Model training module**: This module is to manage the training procedure.
- **Model management module**: This module is to support the model management and the version control functions.
- **Visualization module**: This module is to support the visualization of data, parameters and other useful information generated during training.
- **Model deployment module**: This module is to manage the deployment of the model serving on cloud or devices.

- **Automated machine learning (AutoML) module**: This module is to integrate the automated machine learning algorithms.

7 Supporting functional modules

7.1 User management module

The user management module allows a unified user identity authentication as well as a user authentication management.

The user refers to every authorized system user. Different users can manage resources with varying permissions. There are mainly three types of users an administrator, group user and a common user. In general, the administrators have access to all resources but common users only have access to a limited wealth of resources. The group users that belong to the same user group share the same permissions to the assigned resources.

Its basic functions include but are not limited to:

- **User registration and login**: It provides support for identity representation and authentication of users.
- **User authentication**: User authentication is the identification of user identities and permissions, which helps to avoid risk problems caused by a lack of permission management or inappropriate operation.

7.2 Image management module

The image management module is intended to manage the images for training and inference.

The platform is recommended to provide a wealth of frequently used built-in images that are compatible with popular AI frameworks. If these built-in images cannot meet the requirements of some users, the platform is recommended to allow the users to create or upload customized images.

The functional requirements of this module include but are not limited to:

- **Image packing**: It is recommended to support packing the environment of the container directly as an image.
- **Version control**: It is recommended to support pulling images from and pushing images to the image repository.
- **Image management**: The recommended management operations include but are not limited to creating, deleting, updating, retrieving, etc.

7.3 Resource management module

The resource management module facilitates the management and scheduling of computing resources, as well as provides basic support for a variety of task scenarios. The resource type typically contains the central processing unit (CPU), graphics processing unit (GPU), neural-network processing unit (NPU), etc. The functional requirements of this module include but are not limited to:

- **Resource specification**: It is recommended to support the customization of computing resources for various tasks. Users can configure the resource type, the number of nodes, and the node parameters. The number of nodes can only be set to a number between one and the cluster's maximum node count. The node specification specifies the number of CPU cores, GPU cards, and the memory available to customers for each node. A node's resource utilization cannot exceed the physical node's resource limit.

- **Resource monitoring**: It is recommended to support the real-time monitoring of resource usage during execution. Monitoring indicators include but are not limited to CPU usage, GPU usage, video memory usage, temperature, etc.
- **Resource management**: The recommended management operations include but are not limited to creating, deleting, updating, retrieving, etc.

7.4 Storage service module

The storage service module shall provide diverse configurations of data, as well as the management and scheduling of those resources.

There are mainly three types of storage services: block storage system, file storage system and object storage system.

- **Block storage**: It is suitable for client use. It is recommended to support containers, virtual machines (VM), remote mounts, disk storage allocation, log storage, and other common use cases.
- **Object storage**: It is ideal for maintaining data that does not alter frequently. It does not support directory structure or file manipulations. Image storage, video storage, files, software install packages, and archival data are all common usage cases.
- **File storage**: It is suitable for multi-client data with a directory structure.

The functional requirements of this module include but are not limited to:

- **Storage allocation**: It is recommended to allocate storage space for all programs and data via different tasks.
- **Storage sharing**: It is recommended to tackle the requirement where numerous processes share the same storage space.
- **Storage management**: The recommended management operations include but are not limited to creating, deleting, updating, retrieving, etc.

7.5 Container service module

The container service module is required to provide an execution environment for each task.

The functional requirements of this module include but are not limited to:

- **Container orchestration**: The module is required to automate the deployment, management, scaling, and networking of containers.
- **Container lifecycle management**: The module is required to manage the lifecycle of containers.
- **Container management**: The recommended management operations include but are not limited to creating, deleting, updating, retrieving, etc.

7.6 Log service module

The log service module keeps track of the actions of each platform's essential task, which helps with timely detection, risk mitigation, and operational accident avoidance.

The log types typically contain system log, task log, user log, etc.

The functional requirements of this module include but are not limited to:

- **Log collection and storage**: The module is recommended to support log collection and storage for a variety of tasks, such as data management, algorithm development, model training, model deployment, etc.
- **Log management**: The module is recommended to support basic operations of the log, including but not limited to log querying, filtering, downloading, backing up, etc.

8 Core functional modules

8.1 Data management module

Data is the basis of an AI model development. The data management module is intended to transform the input data into a format that is suitable for AI model development. Figure 8-1 illustrates a recommended data management workflow. The dataset initialization, data import, and dataset release are required modules, and the data pre-processing and data labelling are optional modules.

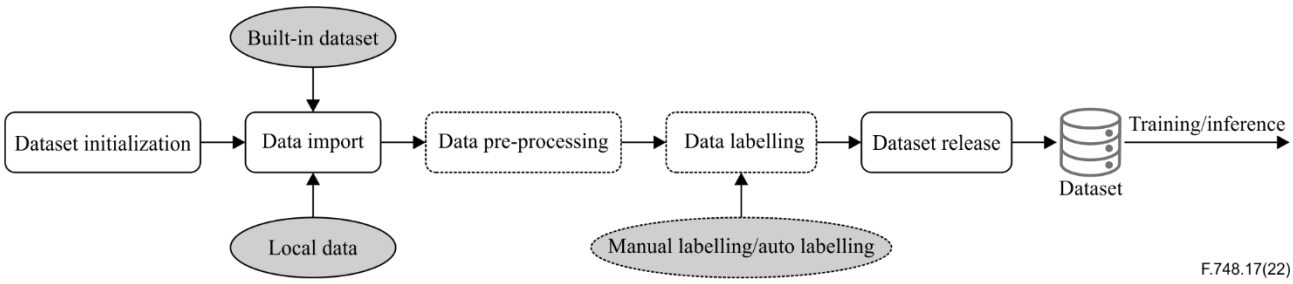

Figure 8-1 – Data management framework

8.1.1 Dataset initialization

Dataset initialization is the foundation of data management and processing. The platform is recommended to enable the user to initialize a new dataset by specifying the metadata such as the dataset name, data type, labelling type, etc. The users can conduct management operations such as deleting, updating and retrieving the newly created dataset.

8.1.2 Data import

The platform is recommended to support the import of massive structured, semi-structured, and unstructured data into a certain dataset.

The data types include local data and built-in datasets. These data can be labelled or unlabelled.

Import modes include single upload and batch upload. The platform is optional to provide an interface to monitor the progress of the data import process.

8.1.3 Data pre-processing

The platform is optional to provide data quality enhancement services such as data cleaning, data verification, and data filtering.

For labelled data, the platform is optional to provide data augmentation algorithms to increase the effective data size and promote the diversity of training samples. The algorithms include but are not limited to cropping, flipping, scaling, rotation, histogram equalization, contrast enhancement, and so on.

8.1.4 Data labelling

The quality of labelling significantly affects the model performance. The platform is optional to provide labelling tools to enable the users to label the imported data.

The platform is recommended to enable the users to perform labelling tasks for multiple purposes, including but not limited to the following:

- **Image**: image classification, object detection, image segmentation.
- **Video**: target tracking, semantic segmentation, video classification.
- **Audio**: sound classification, speech recognition, voiceprint recognition.
- **Text**: text classification, named entity recognition, sequence to sequence, word segmentation.

The specific functional requirements include but are not limited to:

- **Manual labelling**: The platform is recommended to enable the users to manually label data. Besides, the platform is optional to support team labelling to maximize labelling efficiency. The users can perform crowdsourcing labelling for a large amount of data.
- **Automatic labelling**: Automatic labelling is intended to reduce the cost of manual labelling. The platform is recommended to enable the users to use the built-in algorithms to automatically generate labels or train a specific model to label the dataset.
- **Labels management**: The platform is recommended to enable the users to manage the annotations, such as viewing, confirming, modifying, removing, relabelling, etc.

8.1.5 Dataset release

Prior to a dataset being used for training or inference, it is required to be released. The platform is required to support the release of datasets and the management of dataset versions.

The platform is recommended to support the user in releasing various versions of the same dataset labelled at different times. The supported version management operations include but are not limited to viewing versions' detailed information, adjusting the current version and deleting versions.

8.2 Algorithm development module

This module is intended to develop AI algorithms based on the prepared dataset, image and resource. In an AI cloud platform, a user develops and debugs an algorithm in a docker.

The platform is recommended to support customized algorithms and built-in algorithms:

- **Customized algorithms**: The platform is required to enable the users to design and customize algorithms with the development tools.
- **Built-in algorithms**: The platform is recommended to provide a repository of built-in algorithms for various use cases, such as computer vision, natural language processing, recommendation, reinforcement learning, etc. These built-in algorithms can be edited or directly used for training to improve the efficiency of the algorithm development.

The functional requirements for this module are outlined below.

8.2.1 Development tools

The platform is recommended to provide users with online interactive development and debugging environments such as notebooks and drag-and-drop tools. Meanwhile, it is recommended to support the usage of local integrated development environment (IDE) for remote algorithm development. The users can edit, compile, and debug the code with a variety of development tools that support remote development.

8.2.2 Algorithm management

The platform is recommended to support algorithm management operations, including but not limited to creating, deleting, updating, retrieving, downloading, forking, and releasing for training.

8.3 Model training module

This module is intended to support the training procedure of AI algorithms with the prepared dataset, image, resource and algorithm. The functional requirements for this module are outlined below, which can be referred from [ITU-T Y.3531].

8.3.1 Parameters customization

The platform is required to allow the users to customize the meta parameters, model parameters, and resource parameters when creating a new training task. The parameters include but are not limited to:

- **Meta parameters**: task name, algorithms type, target algorithm, image, dataset, output path, etc.
- **Model parameters**: learning rate, iteration, epoch, batch-size, etc.
- **Resource parameters**: number of nodes, resource type, node specification, etc.

8.3.2 Model training

The platform is recommended to support distributed training. The users can train the models on multiple cores (CPU / GPU / NPU / etc.) or multiple devices within a cluster or across multiple clusters.

The platform is recommended to allow the users to restart training tasks from a certain checkpoint.

The platform is recommended to support resource monitoring and log management, which refers to clause 7.

8.3.3 Model saving

The platform is required to allow the users to save the output models to the model management module.

8.3.4 Task management

The platform is recommended to support basic task management operations, including but not limited to creating, deleting, updating, retrieving, visualization, starting, stopping, and viewing detailed information.

The platform is recommended to support training task version management. The users can tune the parameters based on the training result to generate a new version of the task. By comparing the performance of the output models, the most satisfactory model can be determined.

8.4 Model management module

This module is intended to manage the trained models in a unified manner. The functional requirements for this module are outlined below.

8.4.1 Model import

The platform is required to allow the users to import models from the training module or upload local models.

8.4.2 Format management

The platform is recommended to support common standard formats such as, .onnx, .pb, .pt, etc.

The platform is recommended to support converting the models to the required formats to deploy them on different devices.

8.4.3 Model version management

The platform is recommended to support model version management. In multiple training tasks, an algorithm generates numerous versions of a model. The supported model version management operations include but are not limited to creating, deleting, updating, retrieving, downloading, etc.

9 Auxiliary functional modules

9.1 Visualization module

This optional module is intended to support the visualization of model, data, and parameters throughout the training procedure and provide an intuitive reference for AI model training and tuning.

The platform is optional to support various visualization contents, including but not limited to:

- **Model structure**: an interface to visualize the computation graph and model structure graph.
- **Scalar data**: an interface to visualize the scalar data generated throughout the training procedure.
- **Media data**: an interface to visualize media data, such as images, text, video, audio, etc.
- **Hyper-parameter**: an interface to visualize the hyper-parameter of the models.
- **Customized data**: an interface to visualize a customized combination of data.
- **Data statistical analysis**: an interface to visualize the statistical information of data generated throughout the training procedure.
- **Data embedding projection**: an interface to visualize the embedding projection of high dimensional data.
- **Data exception detection**: an interface to visualize the exception point of data.

9.2 Model deployment module

This optional module is intended to deploy the models on cloud, server, edge, and end devices for model inference. The users need to prepare the model, dataset, image and inference script, and convert the model to the required format before deploying a model.

9.2.1 Cloud deployment

For deployment on cloud, the platform is recommended to support both online serving and batch inference. The term online serving refers to the process of packing the model inference as a web service that can be called via HTTP, gRPC, etc. The term batch inference refers to the process of performing inference on a batch of data.

Similar to the model training module, the platform is recommended to support task management, the users can create, delete, update, retrieve, start, terminate deployment tasks, and view task detailed information.

The platform is optional to support resource monitoring and logs generating / presenting / downloading, which refers to clause 7.

9.2.2 End device deployment

For deployment on end devices, the platform is optional to support encapsulating models as SDK programs that can run offline on end devices.

9.3 Automated machine learning module

This optional module is intended to automate critical model development activities such as model design, hyper-parameter tuning and model training.

Hyper-parameter optimization (HPO) and neural architecture search (NAS) are recommended components of an automated machine learning (AutoML).

- **HPO**: a module uses certain strategies to optimize the hyper-parameters of AI models automatically, such as learning rate, batch size and weight decay.
- **NAS**: a module automatically designs the neural network architecture of the AI model by searching and optimizing the structure from a certain search space.

The platform is optional to provide a repository of built-in search strategies and support uploading customized search strategies.

The platform is optional to support the users to perform management operations on AutoML tasks, including but not limited to creating, stopping, deleting, modifying, retrieving, etc.

Appendix I

Deployment of the AI cloud platform

(This appendix does not form an integral part of this Recommendation.)

Platform deployment includes all the actions that make the AI cloud platform available for AI model development. The correct deployment of the AI cloud platform is required to ensure the platform's normal functionality. The deployment process consists of a set of interconnected actions that are tailored to the demands or features of each individual.

In this appendix, a guide for the AI cloud platform's deployment is introduced.

I.1 Deployment architecture overview

Majority of cloud platforms in literature employ microservice architecture, which is an architectural approach that structures an application as a collection of loosely connected and independently deployable services. The microservice design provides high flexibility, scalability and availability.

Figure I.1 depicts a recommended schematic deployment design for an AI cloud platform based on microservices.

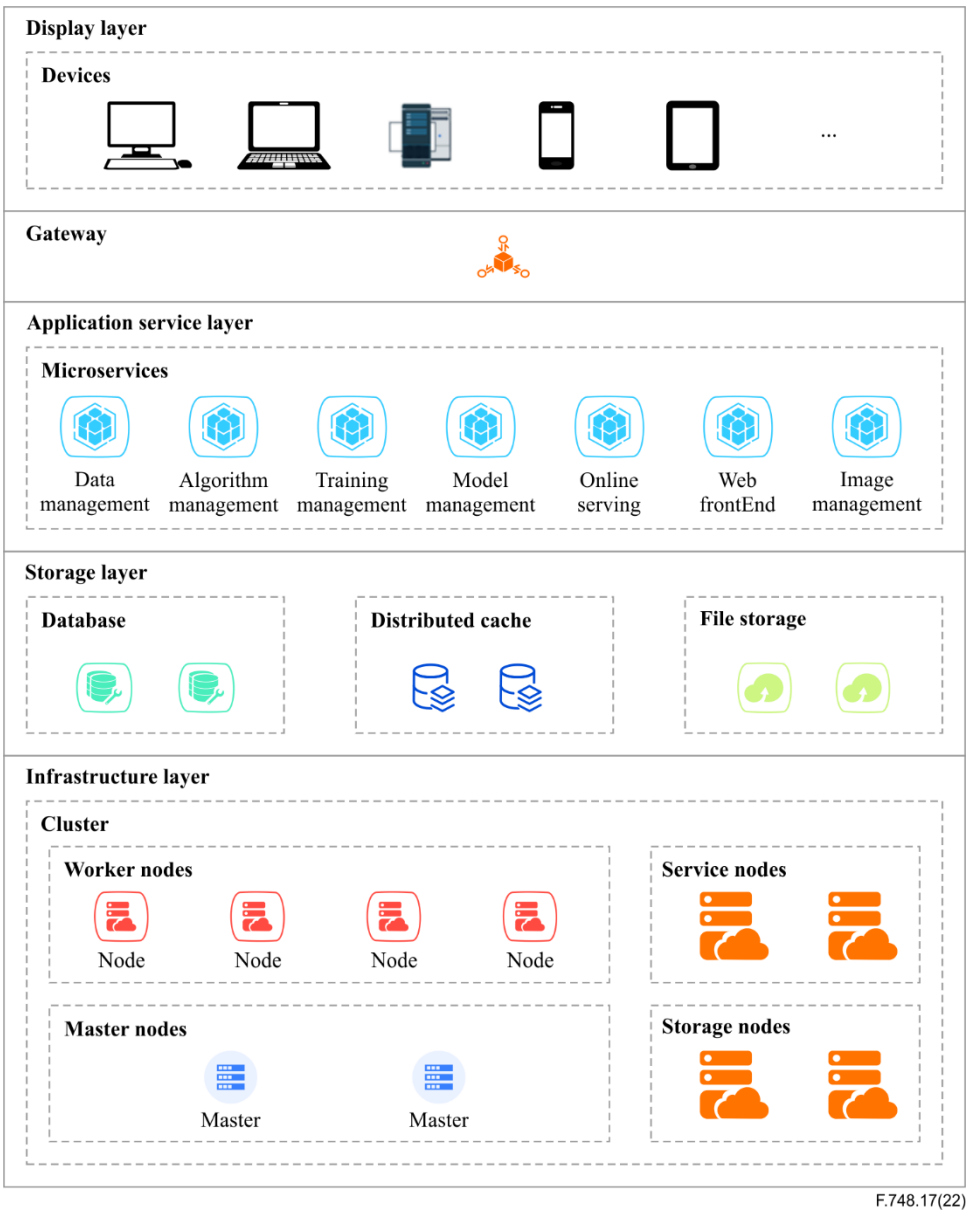

Figure I.1 – Deployment architecture

I.2 Prerequisites and dependencies

For the correct deployment of the AI cloud platform, certain dependencies shall be prepared as follows on demand. There are a variety of dependencies, including but not limited to:

- **Container service**: The container service gives applications a uniform packaging environment in which to run.
- **Image registry service**: The image registry service provides image life cycle services.
- **Cluster management tool**: The cluster management tool is used to manage containerized applications across several hosts in the cloud platform and provide a framework for application deployment, planning, update and maintenance.
- **Database management system**: The database management system provides a data warehouse, which stores data in a specific format. The warehouse allows users to add, query, update and delete data.
- **Storage system**: The storage system is used to store user data, share files, backup data, etc.
- **Version control system**: The version control system is used for data backup and version control.

- **Front-end development kit**: The front-end development kit ensures the normal operation of front-end services.
- **Back-end development kit**: The back-end development kit ensures the normal operation of back-end services.
- **Web and reverse proxy servers**: The web and reverse proxy servers provide routing, reverse proxy, logging, permission management, and other functions.

I.3 Cluster deployment

Generally, a cluster consists of four types of nodes:

- **Master nodes**: The master nodes form a control plane that coordinates the cluster.
- **Worker nodes**: The worker nodes undertake workloads.
- **Storage nodes**: The storage nodes provide a distributed file storage system.
- **Service nodes**: One or multiple service nodes run applications as microservices.

It meets the following conditions:

- Every machine meets the minimum requirements for the assigned type of node.
- Full network connectivity is available between all the machines in the cluster (public or private network).

To use GPUs in this cluster, follow the official manual to install the necessary plugins and drivers.

I.4 Microservices deployment

Before deployment, the microservice architectural pattern should be used to structure the system as a collection of services. Each module, including basic fundamental modules, functional modules, and optional modules, should be separated into a microservice.

The following are a few different microservice deployment patterns:

- **Multiple service instances per host**: It is a pattern in which a single host runs several instances of different services (physical or virtual machine).
- **Single service instance per host**: It is a pattern in which each single service instance is deployed on its host.
- **Service instance per virtual machine**: It is a pattern in which the service is packaged as a virtual machine image and deployed as a separate virtual machine.
- **Service instance per container**: It is a pattern in which the service is packaged as a container image and deployed as a container.
- **Serverless deployment**: It is a pattern in which physical or virtual hosts, or containers, are deployed using a deployment infrastructure that hides any concept of servers.

I.5 System verification

The user may verify the deployment of the system through a simple workflow.

- **Verification of cluster status**: This step is to verify that the cluster is successfully deployed by launching an application to check the state of the cluster.
- **Verification of microservices status**: This step is to check that each microservice is installed and configured correctly.
- **Verification of module functionality**: This step is to ensure each module is available. The user needs to select a minimum collection of test data, including an image, a dataset, and an algorithm to train the model on the platform.

Bibliography

[b-ITU-T F.748.16] Recommendation ITU-T F.748.16 (2022), *Requirements for applications and services in smart manufacturing based on machine vision*.

[b-ITU-T Y.3535] Recommendation ITU-T Y.3535 (2022), *Cloud computing - Functional requirements for a container*.

[b-ISO/IEC 2382] ISO/IEC 2382:2015, *Information technology – Vocabulary*.
<https://www.iso.org/standard/63598.html>

[b-ISO/IEC 11557] ISO/IEC 11557:1992, *Information technology − 3,81 mm wide magnetic tape cartridge for information interchange − Helical scan recording − DDS-DC format using 60 m and 90 m length tapes*.
<https://www.iso.org/standard/19511.html>

[b-ISO 19115-1] ISO 19115-1:2014, *Geographic information − Metadata − Part 1: Fundamentals*.
<https://www.iso.org/standard/53798.html>

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	**Non-telephone telecommunication services**
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Published in Switzerland
Geneva, 2023