

# Development of Smart Distribution Transformer Terminal Based on Software Defined Technology

Hai Wu<sup>1,2,\*</sup>, Xianliang Teng<sup>1,2</sup>, Cheng Zhou<sup>1,2</sup>, Guo Hu<sup>1,2</sup>, Hualiang Zhou<sup>1,2</sup> and Shenjian Dai<sup>1,2</sup>

<sup>1</sup>NARI Group Corporation (State Grid Electric Power Research Institute), Nanjing 211106, China;

<sup>2</sup>NARI Technology Co., Ltd., Nanjing 211106, China

**Abstract.** In order to cope with the development trend of distribution network, the demand for terminal to have the ability of IoT has become increasingly prominent. This paper firstly expounds the development trend of low-voltage distribution network, and emphasizes the technical transformation of the core equipment of low-voltage distribution network-smart terminal; secondly, it puts forward the software and hardware architecture of smart terminal suitable for the IoT, and emphasizes the cooperative capability of the software and hardware platform; and then it introduces the key technologies of the smart terminal, integrates model standardization and virtualization technologies, realizes the plug and play of information flow, and supports application through engineering; finally, it analyzes the work plan of the next stage, and designs to improve the functions of the smart terminal.

## 1 Introduction

In order to realize the widespread sensing, data fusion, smart application and efficient operation and maintenance of the distribution network, as well as to support the optimization and improvement of operation and distribution and comprehensive energy services, the State Grid Corporation proposes to carry out the construction of the Internet of Things on the distribution equipment side and the Internet of Things on the customer side under the overall architecture of the smart Internet of Things system<sup>[1-4]</sup>. Located at the end of the whole power grid, the distribution area is directly connected with users to supply power to them, which is the junction of production and marketing and an important application scenario.

As the core and hub of the main station, distribution area and end-side system, the smart distribution transformer terminal not only completes the data collection, monitoring and control functions of distribution transformer, electricity meter, smart low-voltage circuit breaker, charging pile, temperature and humidity sensor and other equipment, but also realizes the regional autonomy in the distribution area<sup>[5]</sup>. At present, distribution transformer terminals and concentrators are deployed in the distribution area, which have the problems of function duplication and inability of data sharing. Moreover, standardized access of end equipment has not been fully realized<sup>[8-10]</sup>, and the widespread sensing capability is insufficient. The functions of the terminal are relatively fixed and the coupling between modules is high, which is difficult to dynamically expand new functions such as main equipment condition monitoring and evaluation, distributed power supply and flexible load control, and cannot meet the requirements of customized

functions and on-demand iteration of professional applications in various areas<sup>[6-7]</sup>. The on-site processing capacity of terminal data is insufficient, which relies too much on main distribution automation station and main electricity consumption information collection station, making it difficult to realize edge smart applications such as power quality management, low-voltage fault location and judgment<sup>[11-13]</sup>. In addition, the distribution area is large, the terminal remote upgrade and deployment and online monitoring capabilities are insufficient, and the operation and maintenance costs are high<sup>[14]</sup>.

In recent years, with the development of cloud computing and IoT technology, container technology and micro-service architecture have matured in cloud server system. As an efficient virtualization technology, the container technology has the characteristics of resource isolation, quota, measurability and rapid construction and deployment of user instances. Micro-service architecture decomposes functions into discrete services, thus reducing the coupling of the system and improving the flexible scalability of the system. With the gradual rise of edge computing technology, the above technologies have been gradually applied in the embedded system, and the conditions for their application in the power field are becoming increasingly mature. However, adaptive research and development should be carried out according to the secondary terminal performance cost requirements and service characteristics in the power field.

In this paper, the design idea of "hardware modularization and software App" has been adopted, and the implementation scheme of smart distribution transformer terminal with software defined characteristics has been proposed. A terminal platform based on lightweight container virtualization technology has been

\*Corresponding author's email: wuhai@sgepri.sgcc.com.cn

designed to ensure resource isolation, dynamic deployment of application functions and unified operation and maintenance after the integration of distribution transformer terminal and concentrator functions. On the one hand, converged communication resources and plug-and-play management technology have been proposed to ensure standardized access of end equipment and sharing of communication resources in distribution area and solve the problem of difficult operation and maintenance of terminals. On the other hand, with full consideration of service requirements of distribution applications, an open application development interface based on unified model and data management has been proposed to ensure the standardized development, dynamic loading and operation management of application and solve the problem that terminal functions are difficult to expand.

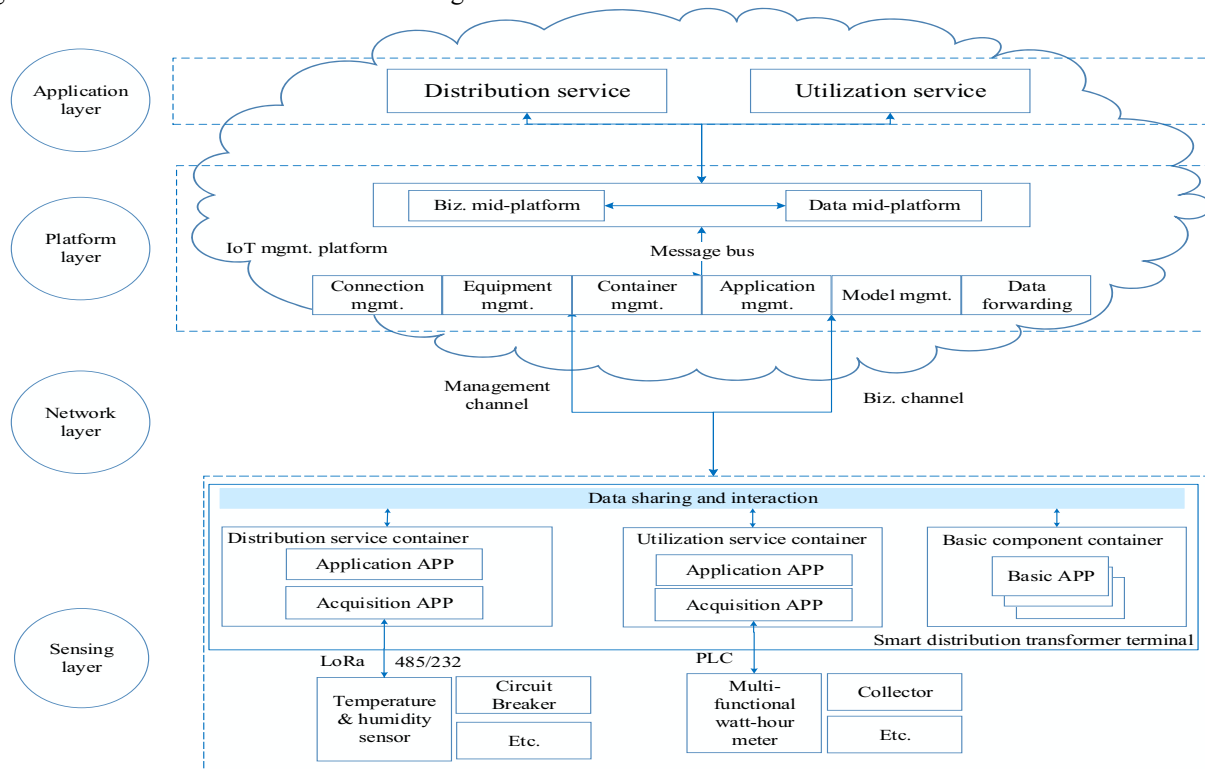
## 2 Overall architecture

### 2.1 System architecture

The smart IoT system consists of the sensing layer, network layer, platform layer and application layer. The sensing layer packet consists of two types of equipment, namely, edge and end. The platform layer consists of the IoT management platform, business mid-platform and data mid-platform. As an edge device for information aggregation and processing, the smart distribution transformer terminal forms a system following the architecture of the smart IoT, as shown in Figure 1. Two logical channels of service and management are

established between the terminal and the IoT management platform, and the two-way information flow interaction is formed through power optical fiber, wireless public network or private network, etc. The business channel carries service data between the platform layer and the terminal, including measurement data and on-site calculation result data. The management channel carries management data between the platform layer and the terminal, including the management information of containers, App and equipment. The terminal collects the data of electric meters, smart low-voltage circuit breakers and other end devices through HPLC, LoRa or RS485.

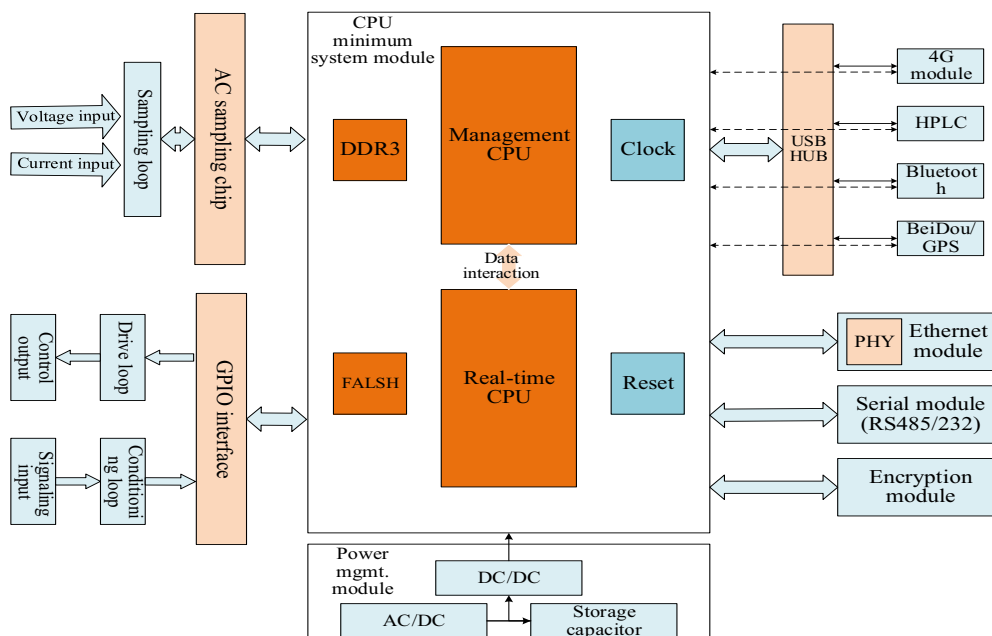
The hardware design of the terminal follows the principle of "platform and modularization". The modular design is adopted for the function units and communication units, which can be freely combined according to the requirements of application scenarios. The idea of App development and container deployment is adopted for the software design. In different application scenarios, the terminal function software definition can be realized by deploying different App and combining different hardware modules. The terminal integrates the functions of distribution transformer terminal and concentrator, integrates the concept of "edge computing", separates data collection from application computing, supports data interaction with the IoT management center, can also take into account the interface mode and communication protocol of existing equipment, and supports data interaction with the main distribution station and the main electricity consumption information collection station.



**Figure 1.** System Architecture

## 2.2 Terminal hardware architecture

Modular design has been adopted for the terminal



**Figure 2.** Hardware Architecture

In order to adapt to the IoT communication of various types of end equipment, there are many types of communication module modules, including wireless modules of 3G/4G/5G, LoRa and micro-power as well as wired modules of HPLC, RS485/232 and M\_BUS. Modules can be dynamically deployed in pluggable form. In addition, in order to smoothly expand the platform architecture to mid-voltage application scenarios and take into account hard real-time requirements, the core CPU module has been designed to support homogeneous dual-core asymmetric processing mode, with one core running the operating system to complete non-real-time functions such as terminal management, data statistics and analysis and the other core building a hard real-time running environment to complete the functions of on-site fault detection, isolation and control. The two cores interact with each other through internal DMA to ensure real time

hardware architecture, as shown in Figure 2, including core CPU, AC sampling, power management, communication module, GPRS clock, encryption, on-site indication and other modules.

and reliability.

## 2.3 Terminal software architecture

The terminal software architecture is based on container virtualization technology, for which a hierarchical micro-service architecture is adopted, consisting of physical hardware, operating system, resource management, universal component, data collection and business application layers, as shown in Figure 3. Through the unified design of device driver interface and data service interface between operating system, universal component and APP, hardware, operating system and application APP can be decoupled from each other, data collection and application calculation can be independent from each other, and service data sharing can be realized.

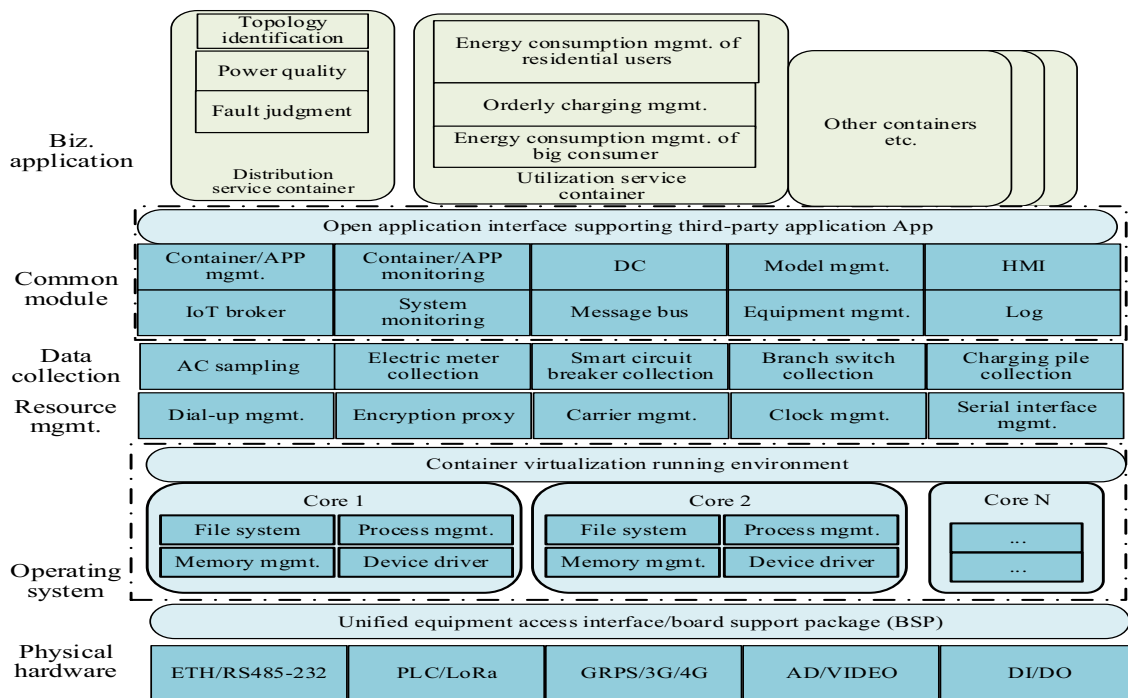


Figure 3. Software Architecture

In the hardware interface layer, peripheral resources can be virtualized, and data collection App and resource management components can be directly operated through a unified equipment interface. The operating system layer mainly includes container running environment and security reinforce operating system. For different service scenarios, it is necessary to support two modes of symmetrical multi-processing (SMP) and asymmetric multi-core processing (AMP). The resource management layer includes dial-up management, encryption broker, carrier management, clock management, serial interface management and other components, to take charge of sharing and managing hardware resources among APPs. The universal component layer includes components such as container/application management and monitoring, system monitoring and logging, data center, model management, message bus, IoT management platform broker, as well as equipment and plug-and-play management, to provide common system services. The data collection layer includes AC collection, electricity meter collection, smart circuit breaker collection, smart capacitor collection and other APPs, which can be dynamically expanded and deployed according to the type of external equipment, to complete external data collection. The business application layer includes distribution service apps, such as topology identification, low-voltage fault rapid judgment and power quality analysis in distribution area, as well as utilization service apps, such as household energy management, big customer energy management and orderly charging of electric vehicles, all of which are deployed in independent virtual containers.

### 3 Key technologies

#### 3.1 Light weight container and application management technology

Because the terminal integrates the functions of distribution transformer terminal and utilization information collector, in order to ensure the resource isolation and relative independence of the two services, the technology of hardware resource virtualization needs to be adopted. Docker engine is adopted for this terminal to realize virtualization and isolate resources such as process, internal memory and external memory. The container provides a complete running environment, which is the entity mirrored by the container. The container mirror contains all the dependencies and class libraries required by the application and is packaged with APP, which enables the application to be conveniently, quickly and dynamically deployed remotely.

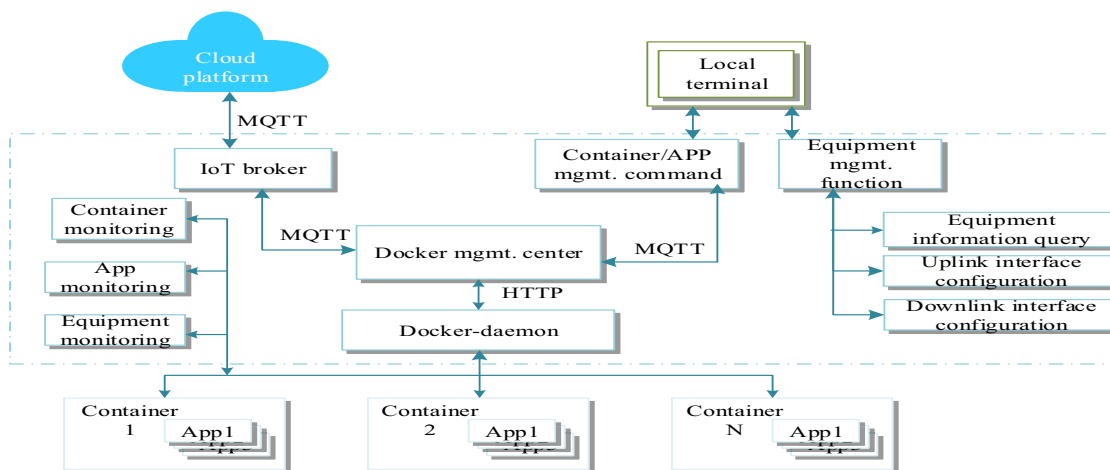
Docker was mainly used in server computing environment in the early days. Due to the abundant hardware resources of servers, the deployment mode of single container and single application was mostly adopted. However, as the terminal is an embedded system and the hardware resources are limited in consideration of cost factor, the container environment must be lightweight. The system is improved from two aspects of mirror clipping and application deployment. In the aspect of mirror clipping, the mirror usually needs to occupy about 100M of external memory space. Alpine Linux is selected for this system as the basic version, with a size of about 8M. On this basis, the universal components in the direction of power distribution and utilization in the distribution area and the resource management components related to APP are integrated, and the native

commands unrelated to the application in the distribution area are clipped. The final mirror size is about 15M, and the capacity is reduced by about 7 times. In the aspect of application deployment, a multi-container and multi-application deployment scheme is designed, and the basic mirror contains DPKG application package management component. The DPKG component can execute the execution script in the App package and manage it to realize the installation and uninstallation of App.

In this system, the implementation principle of container and application management functions is shown in Figure 4. As Docker Client, the container management center requests Docker Daemon services through REST API; Docker Daemon is stationed in the background. As a server, it mainly accepts and processes requests sent by the container management center to complete the management of Docker container and mirror. Since the management function of the terminal can be issued through the main station, the container management center receives the command issued by the management channel, with the command described in JSON mode. The container management center converts the command into

the instruction code identified by Docker Daemon to complete the container installation, start-up and stop functions and complete the APP installation, enabling, start-up and stop functions. In order to facilitate engineering debugging, the operator can also complete the above management functions through the local command line by designing container and application management related commands.

In order to ensure the reliability of terminal operation, the running status of container and App in the terminal shall be monitored. The state of the container and APP is completed by the container and application monitoring module, which interacts with docker daemon regularly to query the resource state of the container and APP. When the processor time, internal memory or external memory resources used by the container exceed the threshold set for it, the operation of the container is stopped; when the usage of App resources in the container exceeds the threshold, the App is stopped, and an exception log file is generated and sent to the IoT management platform for exception analysis.



**Figure 4.** Container and Application Management

### 3.2 Model standardization and management technology

In order to realize the standardized access of end equipment and the sharing of collected data, the standardized modeling must be conducted to the information of low-voltage equipment connected to the terminal. The model mainly includes the equipment information model and the information interaction model between the terminal and the main station, and the terminal must manage the model uniformly.

In order to be compatible with the requirements of mid-voltage applications, the equipment information modeling method refers to IEC61850 object-oriented modeling idea. However, in order to reduce the complexity of the model and adapt to the low-voltage equipment IoT, the hierarchical structure and redundant information of the model need to be reduced. In addition, the principle of separation of data and services is followed,

the functions carried by various types of equipment and service App are analyzed, and the equipment is logically abstracted as equipment information and service capabilities. Equipment information includes basic information, measurement information, topology information and asset information; Device service includes service identification, command description and command parameters. The specific modeling methods are as follows:

Firstly, the basic data classes required by data interaction services are defined according to the equipment functions, with the main data classes divided into remote measurement, remote signaling, accumulation, parameter, command and event. Data classes mainly specify data name, storage type, quality, unit, maximum value, minimum value and step size. The data name, based on a unified standardized description, has definite semantics, which are closely related to service functions. The matching description of the main station can clearly define the signal content. The data class hierarchy is



simplified into two layers, and the data name is described in the form of LN (Logical Object) and DA (Data Attribute).

Then, on the basis of above data classes, for the equipment information model, a model template is defined for each type of equipment or service App (virtualized as a device). The basic information in the model includes equipment identification, equipment name, equipment type and manufacturer identification. The measurement information includes remote measurement quantity, such as voltage, current, power, harmonic, temperature and water level; remote signaling quantity, such as input status, action signal, off-limit alarm and equipment self-checking signal; accumulation quantity, such as active power, statistical time of voltage off-normal and number of faults; configuration parameters, such as fault trip limit and off-limit alarm threshold; remote control quantity, such as output relay control and reset; event quantity, such as fault information and freeze events. Topology information includes hierarchical information and topology connection information of low-voltage primary equipment. Asset information includes asset identification, hardware version, software version, manufacturer name, and date of production, installation and retirement.

For the information interaction model between the terminal and the main station, a subset of the terminal self-model, the edge computing App model and the collected and aggregated end equipment model is mainly included. According to the method of dataset (data set) in IEC61850, the information required to be sent to the main station can be selected from the above models to form a data list. At the same time, in order to be compatible with the old non-IoT protocols (such as IEC 60870-5-104), information related to protocol mapping, such as type identification, information object address, polarity, coefficient and offset, should be added to the model.

Finally, the description method of the model is chosen, and XML (Extensible Markup Language) or JSON (JavaScript Object Notation) and description language can be chosen. JSON, with key (name) and value (attribute value) information pairs as basic units, is simpler than XML, which shall be adopted preferentially.

In practical application, the instantiation of the model should be completed according to the type and quantity of service App and external equipment deployed in the terminal. The terminal model management component completes the storage and instantiation of the model, and can be merged and deleted according to the dynamic installation and uninstallation of APP. The instantiation of the model is divided into two parts, namely, the instantiation of data-to-equipment relationship. For data instantiation, according to the dynamic change process of the App of collection class, the instance object name is formed in the way of "Object Serial number", and the data name is expressed in the way of "instance object name.data name". For the instantiation of data-to-equipment relationship, PMS\_ID (the corresponding relationship between primary and secondary devices described in the engineering production management PMS system) is entered into the asset identification attribute in the asset information through the main station

distribution or equipment sending method. The model merging and deletion mainly involves the merging and deletion of different App information of collection class and edge computing class as well as data type templates.

### **3.3 Converged communication resources and plug-and-play management technology**

Because different data collection Apps will use the communication module of the terminal at the same time in some application scenarios. For example, both the electricity meter information collection App and the low-voltage branch monitoring unit collection App may interact with relevant equipment through HPLC (Broadband Carrier). At this time, there is a mutual exclusion problem. Therefore, the resource management layer is set up to manage hardware resources, and the sharing mechanism with priority should be designed.

The serial interface, Bluetooth, wireless communication module, local carrier module, encryption chip and other resources of distribution transformer terminal are uniformly operated by the corresponding resource management module. App interacts with the resource management module through socket, with the interaction message containing App identification, operation type (read, write or set), operation-related data and priority. The resource management module performs hardware access tasks according to priority, supporting synchronous and asynchronous modes. Priority is sorted according to service type, with the priority given to regular meter reading task of utilization class. In order to support asynchronous operation, the access message and the response message also contain message sequence numbers, which are used by APP to match the sending and receiving information.

Due to the large variety and quantity of low-voltage equipment, manual configuration and debugging are required when the equipment is connected to the system, for which there are the problems of heavy workload and low efficiency, and it is difficult to ensure the accuracy and consistency of configuration information input. In this paper, a device access plug-and-play mechanism is designed, which is based on equipment model standardization, including terminal and end equipment discovery process, terminal internal model instantiation process and IoT management platform and terminal plug-and-play process.

Terminal and end equipment discovery processes are divided into two types. For the non-networked end equipment, APP in the terminal obtains the equipment access status through polling; for the networked end equipment, the equipment actively sends the online status to the terminal. The terminal model management module completes the instantiation operation to the model according to the online and offline status of the equipment. The IoT management platform obtains the equipment number, equipment type, manufacturer name and other information of low-voltage end equipment and terminal through handheld equipment, generates an equipment description file, and finds the corresponding model information from the model base. At the same time, the

model information is matched and checked with the information sent by the terminal, so as to complete the update of the online and offline status of the low-voltage end equipment and the terminal. The IoT management platform then generates the information interaction model between the terminal and the main station according to the template of the data required by the main station application, and sends it to the terminal. The terminal sends data according to the interaction model to realize plug and play of the data flow between the terminal and the IoT management platform.

### 3.4 Open application development interface and data management technology

In order to solve the problems of data non-sharing, high coupling between applications and difficulty in expansion, the terminal follows the principle of separation of data and

applications, provides the open interface for third-party application development, and enables device capabilities. Interface mainly involves three aspects: data center, message bus and interface design. As shown in Figure 5.

Data center includes message parsing and encapsulation, database management, real-time database and historical database. The structure of multi-dimensional relational table is adopted for the real-time database, and the character hash is adopted for the data index, which are organized according to balanced binary tree and reside in memory, mainly storing remote measurement, remote communication and real-time electricity data. Lightweight database SQLite is adopted for the historical database, which is stored in external memory, mainly storing SOE, freeze events, operation records and parameters, etc. The data identification and information body are completely expressed according to the description of the model file.

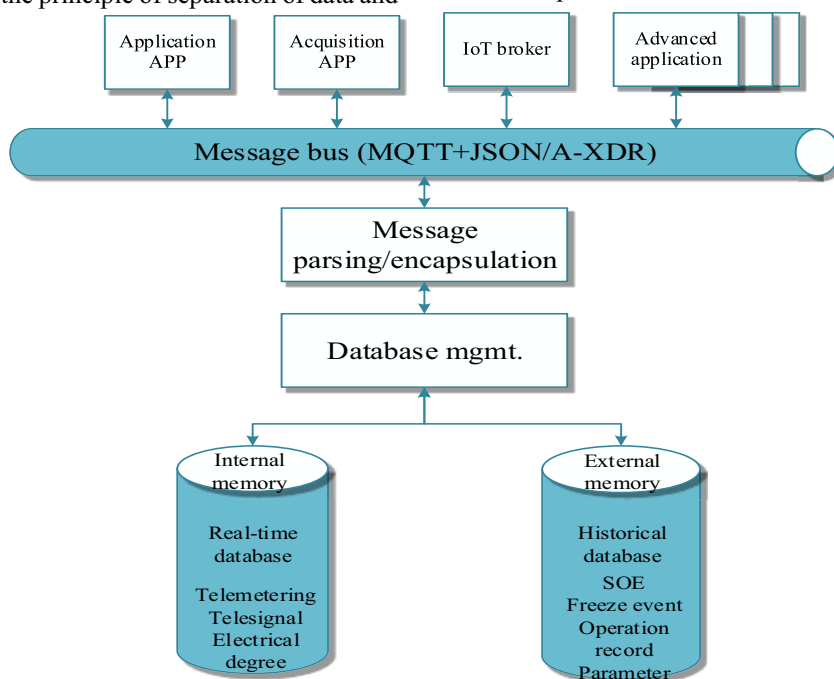


Figure 5. Data Center

All kinds of Apps and data center in the terminal may be deployed in different containers. Since the containers isolate the memory and data cannot be shared between App and data center by sharing memory, the MQTT protocol is selected as the message bus. Data interacts through subscription and publishing mechanisms, with the interaction channel of network channel. The container network mode is set to bridge mode among the four modes (host, container, none and bridge). After the container is started, a virtual bridge will be established, through which Apps in the container attached to the virtual bridge can send and receive messages. All kinds of Apps interact with the data center through MQTT Broker, obtain data on demand, and complete advanced application functions. To prevent system exceptions caused by exceptions from a single broker (message broker), the broker is deployed inside each container, which connects with the main broker outside the container, and the slave broker is deployed outside the container at the same time. When the

main broker is abnormal, the slave broker automatically switches to the working mode to achieve the goal of disaster prevention and fault tolerance.

The format of the message topic is designed as: source App name/operation type/transmission reason/destination App/information type. The operation type can be divided into reading, setting, selecting, confirming, etc.; Transmission reason can be divided into request, response, cycle, burst, etc. Information type can be divided into real-time data, historical data, parameter, model, etc. The message body is mainly composed of information identification and specific contents, which is related to the equipment model and described by JSON. However, in consideration of the compatibility with object-oriented utilization information data exchange DLT698.45 protocol of marketing service, A-XDR code can also be adopted. In addition, in order to improve the efficiency of data access and meet the application scenario of APP high-speed data processing, a set of third-party application

development database interface suitable for electricity distribution and utilization is also designed, with the interface contents shown in Table 1.

**Table 1** Interface Contents

SN	Classification	Purpose
1	Equipment access	To realize the access of serial interface, SPI, USB and other peripheral devices
2	Real-time data	To realize the reading and writing of real-time data
3	Event	To realize the pushing, reading and writing of event data
4	Parameter	To realize the reading and writing of parameter data
5	Control	To realize the remote communication control of outgoing node output and App
6	Historical data	To realize the storage and reading of historical cycle data
7	Equipment management	To realize the equipment registration, deletion and App association management
8	Model	To realize the dynamic addition and deletion of models
9	Topology	To realize the reading and writing of topology data
10	Log	To realize the reading and writing of log function

## 4 Engineering application

The use of software defined technology enables terminal functions to be customized according to user requirements. The installation and use of smart terminal bring the fault risk early warning means to the low-voltage equipment in the distribution area. According to the structure of the low-voltage distribution network and the principle of loop impedance measurement, the loop impedance can be determined by measuring the voltage and current change rate of each user at two moments. The statistical results of loop impedance measurement can be used to predict the potential fault risks of users, locate the aging or potential fault points of lines, guide the operation and maintenance personnel to actively and timely arrange on-site maintenance, eliminate faults in the bud, and reduce the failure rate of equipment in the distribution area. Long-period low-voltage loop impedance data are collected and stored by distribution automation main station. In a certain period, the impedance information of low-voltage distribution network in each region is analyzed and compared with big data, and checked with the results of low-voltage fault prediction of each smart distribution transformer terminal, so that the fault occurrence of low-voltage distribution network can be predicted more accurately and the success rate of fault prediction can be provided.

In Dongguan low-voltage visualization project, the distribution transformer terminal cooperates with low-voltage fault indicator and concentrator to complete low-voltage topology identification and fault location by deploying low-voltage fault indicator collection, topology identification and utilization data collection as well as

low-voltage monitoring and fault location. In the Guangzhou Operation and Distribution 2.0 pilot project, distribution service containers and utilization service containers were deployed at the distribution transformer terminals respectively. The distribution service container consists of APPs of distribution transformer monitoring, branch switch monitoring, low-voltage fault location and distribution remote communication, while the utilization service container consists of APPs of carrier micro-power wireless dual-mode meter reading, AC sampling and metering remote communication. By means of container software isolation, the integration of distribution monitoring and analysis and utilization information collection functions is realized in the same terminal, and the data collection and fault sensing of low-voltage equipment are realized. Remote deployment and plug-and-play mechanism is adopted for all above projects, and the debugging workload in newly-built distribution area is reduced from 8 hours to 3 hours.

With the active sensing and fault location functions of the distribution transformer terminal low-voltage power outage equipment, in Shandong Laiwu pilot, more than 270 active repair task orders have been issued, about 280 repair people have been dispatched, the low-voltage repair time has been shorten by about 30 minutes on average, the distribution network operation and maintenance management and power supply service level have been further improved, the repair efficiency has been significantly improved, and significant progress has been made in active repair. With the low-voltage monitoring and collection functions of the distribution transformer terminal, there are 703 pole transformers and 1523 outgoing lines in Suzhou Tongli Demonstration Project, with a total of about 25,000 users. There are about 2158 cable branch boxes, with about 2058 new branch boxes and about 100 old branch boxes. The average estimated time to warranty site for emergency repair in the whole Tongli core area has been reduced from 30 minutes to 15 minutes. At the same time, with the full application of smart sensing equipment, 80% of hidden dangers of neutral point faults and low-voltage circuit disconnection faults can be discovered and eliminated in advance by the smart sensing function of power supply service command, and the average power outage time per household can be reduced from 82 minutes to 55 minutes per year. Real-time monitoring of power quality has been realized, and abnormal voltage conditions have been found and treated in advance. At the same time, the harmonic over-limit situation has been analyzed, the unbalance degree of three-phase load has been comprehensively considered, and measures, such as distribution transformer shift adjustment and three-phase load adjustment, have been taken. A total of more than 80 alarms have occurred, more than 60 load adjustments have been carried out, more than 20 times of distribution transformer shift adjustment have been carried out, and the qualified rate of comprehensive voltage of urban network has reached 99.996%.

## 5 Conclusion

In this paper, the smart distribution transformer terminal



is developed according to the idea of software definition. Through the research on virtualization technology, the function of dynamic deployment application has been realized without changing the terminal hardware. Through the research on the IEC 61850 modeling idea, the self-description of equipment or APP capability has been realized, laying a foundation for "plug and play" and providing standardized data guarantee for enterprises and various business applications. Through the design of software and hardware architecture in the terminal, the data bridge has been constructed, and the data interaction between containers and CPUs has been realized. Through the analysis of distribution and utilization service data, the open and unified application development interface has been realized, providing support for the third party to develop App.

With the gradual application of terminals on the spot, some problems have been also highlighted. Because wireless data transmission is generally adopted for the on-site terminals and the network is unstable, the online rate of terminals is not high. And after there are problems with the terminal, the manual on-site location is generally adopted for the fault location method, with single processing method and huge consumption of human resources. Based on the above problems, the terminal functions need to be subsequently improved in terms of the extensive interconnection, on-site decision-making and convenient and efficient operation and maintenance of the equipment. Firstly, the stability of terminal network communication and the fault-tolerant scheme of software shall be studied under complex network conditions. Secondly, through cloud-side cooperation, the remote controllable safe operation of terminal operation and maintenance shall be realized, and the means of fault location shall be enriched. Finally, the local decision-making processing capability of the terminal shall be further developed by improving the equipment condition monitoring and evaluation capability. In terms of convenient and efficient terminal operation and maintenance, the requirements and related measures of cross-disciplinary management after integration have been studied.

## References

1. Lu, J., Luan, W., Liu, R., et al. (2018) Architecture of distribution Internet of Things based on widespread sensing & software defined technology. *J. Power System Technology*, 42(10):3108-3115.
2. Lu, J., Sheng, W., Liu, R., et al. (2019) Design and application of power distribution Internet of Things. *J. High Voltage Engineering*, 45(6):1681-1688.
3. Wang, C., Wang, D., Zhou, Y. (2015) Analysis of Smart Distribution System Architecture and Technical Challenges. *J. Automation of Electric Power Systems*, 39(09):2-9.
4. Liu, J., Cheng, H., Zhang, Z. (2013) Planning the number of distribution terminal configuration in distribution automation system. *J. Automation of Electric Power Systems*, 37(12):44-50.
5. Tian, C., Yin, Y., Guan, C., et al. (2018) Comprehensive evaluation strategy of medium and high voltage distribution network based on power supply division. *J. Power System Protection and Control*, 46(21):152-159.
6. Han, Z., Xu, B., Suo, N., Chen Y. (2012) Implementation of Automatic Discovery Technology of Distribution Terminals. *J. Automation of Electric Power Systems*, 36(18):82-85.
7. Liu, D., Wang, Y., Lu, Y. (2016) Analysis of the difference between the measurement models of IEC 61968 and IEC 61850. *J. Distribution & Utilization*, 33(10):19-24.
8. Chen, R., Lu, J., Liu, M., et al. (2019) Topology model generation method of distribution network for distributed feeder automation. *J. Southern Power System Technology*, 13(1):60-65.
9. Wang, Y., Xu, X., Mei, J., et al. (2015) Automatic recognition technology of distribution terminal based on IEC 61850. *J. Electronic Measurement & Instrumentation*, 53(6):32-36.
10. Liu, J., Lin, T., Zhao, J., Wang, P., Su, B., Fan, X. (2014) Research on Distribution Automation System Planning for Power Supply Reliability. *J. Power System Protection and Control*, 42(11):52-60.
11. An, Z., Chen, H., Chen, H., Hou, Z., Chang, S. (2016) Design of Smart Distribution Transformer Terminals for New Rural Networks. *J. Power System Protection and Control*, 44(09):140-144.
12. Zhang, Z., Zhou, J., Cao, Y., et al. (2017) Distribution automation terminal based on modular configuration design. *J. Automation of Electric Power Systems*, 41(13):106-110.
13. Huang, K., Zhao, K., Yang, L., Yan, Z., Peng, H., Zhao, M. (2019) Lightweight human-computer interaction architecture and key technologies of power grid control system. *J. Automation of Electric Power Systems*, 43(07):159-166.
14. Shan, X., Lu, X., Zhao, M., Gao, Z., Xu, C., Teng, X., Wang, B. (2019) Analysis of the key technologies of artificial intelligence applied to grid control. *J. Automation of Electric Power Systems*, 43(01):49-57.