Research on SaaS Layer Application Architecture for DCCP Considering Ubiquitous Internet of Things

Hongjun Liu¹, Ti Guan¹, Yujie Geng¹, Qiang Ma¹, Dawei Tian¹, Yong Liu¹, Lin Lin¹ and Sheng Li²

¹Power Dispatching and Control Center, State Grid Shandong Electric Power Company, Jinan, Shandong 250001, China
²TELLHOW SCI-TECH CO., LTD, Nanchang, Jiangxi 330000, China

Email (Yujie Geng): 278893104@qq.com

Abstract. In 2019, State Grid Corporation of China officially proposed the overall planning of building SG-eIoT (State Grid-electric Internet of Things). DCCP (Dispatching and Control Cloud Platform) is an essential foundation and critical part of SG-eIoT, it aims at achieving cross-departmental resource sharing through resource virtualization, data standardization, and application orientation, so as to break a new way to improve operation level of power system, and operation efficiency of State Grid. At present, infrastructure layer and public service layer of DCCP have been formed, but with some inadequacies in application layer architecture. In consideration of practical operation of State Grid, this paper proposes an application architecture what is not only in line with operation specifications, but also convenient for unified management. This design helps to wipe off obstacles for data communication among different applications, and realize SG-eIoT. In addition, competition mechanism is introduced in application architecture to enhance practicability of applications.

1. Introduction
Facing three major challenges, reformation of State Grid operation mode, bottlenecks in cooperation development, and changes in social economic formation, State Grid Corporation of China officially proposed the overall planning of building SG-electric Internet of Things in 2019. DCCP is an essential foundation and component, it aims at achieving cross-departmental resource sharing through resource virtualization, data standardization, and application orientation, so as to break a new way to improve operation level of power system, and the operation efficiency of State Grid. In addition, the DCCP can further expand user scope, and explore the blue ocean market of digital economy[1].

2. Current situation of DCCP
DCCP architecture is designed to typical bottom-up layering of cloud computing, including IaaS layer, PaaS layer and SaaS layer, as well as the functional module for cloud security[2], shown in Figure 1
IaaS layer is to realize virtualization of resource, build computing resource pool, storage resource pool and network resource pool. Computing resource pool provides computing resources by virtualizing server; Storage resource pool includes devices like servers and centralized storage, it can improve resource utilization by virtualizing storage; Network resource pool contains devices such as routers and switches, which can enhance the forwarding and control capacity of network flow through virtualizing network.

PaaS layer integrates the core components of DCCP, bases on the "Structured Design of Power Dispatching General Data Objects", provides common components of DCCP, model data centre, real-time data centre, big data platform, model maintenance tools, vertical and horizontal data synchronization tools, and it supports SaaS layer application development.

SaaS layer packages applications into services. Typical DCCP has applications such as query, analysis, decision-making, and simulation. And applications can be extended as required.

![DCCP Layered architecture diagram](image-url)

**Figure 1.** DCCP Layered architecture diagram

3. Summary of problems in application architecture of DCCP
After analyzing the achievements and future planning of DCCP, it's found that State Grid Corporation has issued strict specifications for overall architecture of IaaS layer and PaaS layer, and provincial grid companies have been established. In terms of application architecture in the SaaS layer, State Grid Corporation gives only guidance of direction. Power grid companies at different layers can launch projects independently according to local needs. Currently, they have been exploring and developing applications in SaaS layer, and have put some applications into operation. Some problems of architecture are exposed during development and using. Firstly, application architectures are not standardized, unified data model is not adopted, which not conform to overall planning of DCCP. Secondly, applications are independently packaged, without public interfaces or service, which causes information island in the platform. Finally, there isn't competitive evaluation mechanism for online applications, no competitive environment for the survival of the fittest[2-4].

4. Application architecture design based on standard data structure
A complete SaaS layer application should include three layers, UI, logic, and data model. Figure 2
gives a detailed description. The UI layer includes menu, form layout, portal and other user interaction designs; the logic layer executes processes, triggers events, controls user permissions; the data model layer is responsible for definition, organization and association of data model, which is the basis to realize functions of applications.

Figure 2. Application architecture of SaaS layer

Data model layer is the basis of DCCP. The essence of the SaaS application layer is actual information flow between people, devices, and between people and devices. To ensure smooth flow of information, avoid information silos, all application underlying design must be based on standard data model of DCCP. Take creating a dispatching events management system as an example, data generated by the software system must be based on the standardized mode of personnel, equipment, and organization in PaaS layer, related business data identification should be unified. And then data fields are expanded to record related data. When data models of all the applications are built on the basis of PaaS layer specifications, data generated by all applications on the cloud platform will establish a natural bloodline correlation, ultimately achieve SG-eIoT.

5. Implementation of application architecture based on standard data structure
DCCP is open, offering a advantaged platform to grid companies in different layers to develop applications. It’s also a highly constrained platform due to corporate environment. To realize unified models and application architecture, and to support data interaction among applications, the UI layer, logic layer and data mode layer are independently packaged to respectively UI engine, processing logic engine, data modeling and access engine, as shown in Figure 3[5-6].

Data modeling and access engine provides a standard access portal for the SaaS layer data model, and supports access to standard models, additions, deletions, and changes to sub-table data.

Logic processing engine includes a standardized process parser and an instantiation tool for SaaS layer applications. All applications developed based on this engine can be associated with arithmetic logic.

UI engine provides standard front-end design architecture for all SaaS layer applications. Applications developed based on this engine ensure consistent application style and support cross-application page access.

Maintainers of DCCP can upgrade the three engines, while application developers can only realize functions based on these three engines.
Figure 3. Three-tier engine design for application

Applications based on the above technical specifications, the architecture is shown in Figure 4.

Figure 4. Application architecture based on three-tier engine

Developers who are weak in programming but with strong industry background and capabilities of process optimization can design specific applications based on SaaS platform for electric power industry, through packaging three-tier architectures of applications independently. It provides a standard and open stage for dispatchers to demonstrate their talents and to improve grid management.
level.

5.1 Structure design of data modeling and access engine

Figure 5 depicts the data modeling and structure of the Access Engine. The core technology is encapsulating the middleware Shark into data modelling and access engine[7], realizing cross-database and cross-table data reading and writing, isolating database access requests from databases, and opening standard access interfaces to all applications.

**Figure 5. Structure of data modeling and Access Engine**

Main features of the engine:
(a) All applications are not directly connected to the database, so the configuration is very simple.
(b) Support spring JDBC framework, and SQL parsing based on druid.
(c) Provide API to solve the problem of multi-machine sequenceid.
(d) Support resource allocation based on zookeeper and redis3.x cluster.
(e) Support rendering content based on velocity template engine, support SQL statement configuration independently, dynamic stitching, and ultimately achieve logic decoupling.
(f) Provide built-in validation pages to facilitate development, testing and maintenance.

5.2 Structure design of logic processing engine

**Figure 6. Structure of logic processing engine**
Figure 6 depicts the structure of logic processing engine. It includes F2 database access layer, workflow engine kernel and Web peripheral facilities developing layer[8]. Workflow engine provides standard interfaces according to WFMC and BPMN specifications, including defining interfaces, management interfaces, application interfaces, remote interaction interfaces, database interfaces and so on. On basis of these interfaces provided by workflow engine, the product realizes integration of design tools, management tools, application tools, and adapters with other systems.

Workflow engine kernel architecture is designed as figure 7.

Development of workflow engine kernel is based on J2EE/Java. The engine can be applied to software integration, application system development, process service centre and embedded system integration. Workflow engine supports online Web process design and online Web form design. Developers can complete process design by pulling and dragging the mouse without programming. The workflow engine can directly access the data engine, and then manipulate the database.

5.3 Structure design of UI engine

Figure 8 depicts the structure of UI engine. Its main function is to encapsulate the popular web framework VUE[9]. This engine can analyse source codes of running applications on the platform, extract common programs, make them into components, and finally generate a set of component library. These components cover common page elements, common pages, general parsers, etc. Applications based on component library guarantee the unification of technology, style and architecture, and provide the technical basis for mutual access between applications. The UI engine structure as follows[10].
6. Design of application evaluation mechanism

With introduction of application architecture proposed by this paper, technical difficulties in application development will be diminished, DCCP will become a blooming application platform. To realize the same function, it may happen that various applications will be developed by different individuals or teams. Therefore, a mechanism of evaluation and elimination must be built to ensure the quality of applications. Design of the whole mechanism as follows.

Initially, grouping function is established on cloud platform to classify the same operations to the same group for comparative analysis. Secondly, all online applications are registered to DCCP's service bus in order to support statistics of application access times, access duration and response time. And functions to statistic visits and favorites, to score services are developed in parallel. Finally, valid period for scoring is set, the platform will automatically save the highest score and uninstall other applications when the scoring operation becomes due. In Figure 9, we can see the ranking of application click count. In Figure 10, we can see the response time curve for all the public services.

![Figure 8. Structure of UI engine](image_url)

![Figure 9. Application access statistics table](image_url)
7. Conclusion
Standardized application architecture based on UI engine, logic processing engine, data modeling and access engine has been in trial use in DCCP of Shandong, and already proved highly efficient, and very universal. Applications use this architecture are unified in architecture, convenient and fast for cross-system data access. This architecture lays a solid foundation of data platform for SD-eIoT. Furthermore, realization of this application architecture gives guidance and reference to other application designs and platform upgrading.

References
[1] Jing Ming, Xu Hongqiang, Liang Chenghui. Application of information standardization of smart grid in dispatching center[J].Power System Technology,2010,34(10):15-19.
[2] IEC 61970 Energy management system application program interface(EMS-API):Part 552 CIM XML model exchange format[S].2009.
[3] Xin Yaozhong. Research on international standard architecture for smart grid dispatching and control technology[J].Power System Technology,2015,39(1):1-10(in Chinese).
[4] Cao Junwei, Wan Yuxin, Tu Guoyu, et al. Information system architecture for smart grids[J]. Chinese Journal of Computers,2013,36(1):143-167(in Chinese).
[5] Cui Wei, Shi Yong, Sun Bing. The construction and integration of grid model based on IEC 61970/61968[J]. Power System Protection and Control, 2011,39(17):60-63(in Chinese).
[6] Li Dapeng, Chen Zhenyu, Deng Zhaoyun, et al. A wide area service oriented architecture design for plug and play of power grid equipment[J]. Procedia Computer Science, 2018, 129:353-357.
[7] Chen Zhenyu, Li Dapeng, Deng Zhaoyun, et al. The application of power grid equipment plug and play based on wide area SOA[C]. Proceedings of 2nd IEEE International Conference on Energy Internet. Beijing: IEEE, 2018:19-23.
[8] International Electrotechnical Commission. 2013 energy management system application program interface(EMS-API):part 552 CIM XML model exchange format: IEC 61970-552[S].2013.
[9] Xin Yaozhong, Mi Weimin, Li Wei. CIM-based grid model and graphics exchange format[J]. China Standardization, 2011,47(4):72-74.
[10] Xu Hongqiang. The architecture for dispatching and control cloud and its application prospect[J]. Power System Technology, 2017,41(10):3104-3111(in Chinese).