Cross Business Integration Technology and Application Based on Mobile Portal

Xiaozhen Li*

State Grid Information and Communication Industry Group Co., LTD., Beijing, China

*Corresponding author: lixiaozhen1986@126.com

Abstract. This paper proposes a cross business data fusion technology solution, which effectively solves the data access of different systems through an asynchronous data aggregation method combining push and pull. Through multi-layer asynchronous method, it solves the consistency of user schedule display and improves office collaboration efficiency.

Keywords: component; schedule; cross business

1 Introduction

Many large enterprises have their own office systems[1] built to improve their office efficiency and process management through computer technology and related software[2]. This not only provides efficient collaborative work for users, but also provides fast and convenient approval methods for management personnel[3]. These systems that support office operations such as employee work meeting reservations and business travel approvals are closely related to business data and user schedules. Due to the independence of system data and operation, it is difficult to organically aggregate data between different systems, and there are certain limitations to work flexibility and convenience.

With the rise of mobile Internet technology, many enterprises have established a mobile portal on the basis of retaining the original functions of the Web side business system to achieve the unified convergence of business data on the mobile side. Cross system data integration is a common challenge[4]. How to effectively manage time, plan schedules, and improve work efficiency has become a challenge[5] .Traditional data aggregation techniques often use a single business push data method for aggregation, which is difficult to leverage when faced with stable business system operation and unwillingness to adapt interfaces. In addition, traditional technologies lack a joint synchronization strategy between the server and client, which cannot meet our need to ensure synchronization without significantly affecting server performance and user experience on mobile portals. In order to centrally display schedule data from different business systems of enterprises on mobile devices and provide users with efficient mobile office capabilities based on schedules, this paper adopts a cross business data consistency technology based on asynchronous timing strategy, achieving the collection

Z. Ahmad et al. (eds.), *Proceedings of the 2024 5th International Conference on Urban Construction and Management Engineering (ICUCME 2024)*, Advances in Engineering Research 242,
https://doi.org/10.2991/978-94-6463-516-4_38

and centralized display of data from different business systems according to schedule information.

2 Overall Design

2.1 Design of Cross Business Data Fusion Solution

This cross business business data fusion scheme uses Internet collaborative office technology as the main technology, takes computers as the core, and combines network communication technology, collaborative technology and other advanced technologies to build a resource sharing application that can achieve cross region, cross industry and cross department [6]. This solution is divided into two parts: the server and the client. The overall technical solution design is shown in Figure 1.

1) Server

Obtain data from various business systems through interface services and scheduled tasks, convert and process the obtained data with different structures, and store them in the schedule aggregation database according to a unified data structure[7]

2) Client

Using a mobile portal as the entry point, by querying the server's schedule data aggregation database, different types and formats of schedule data from different businesses are centrally displayed in a time series mannerl.

In this scheme, the server is responsible for collecting business system data and ensuring the consistency between business data and schedule data. The client is responsible for asynchronous page refreshing and updating schedule information, displaying the latest schedule information to users.

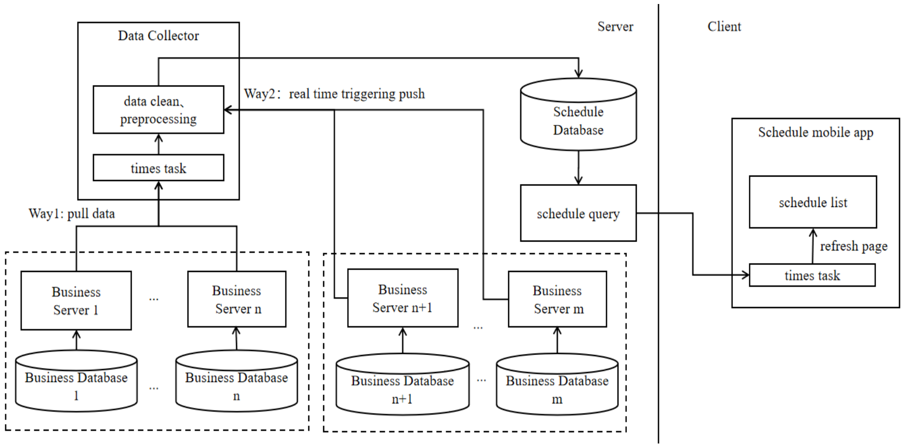

Fig. 1. Overall Architecture Design

2.2 Data Synchronization Mechanism

1) Server side business data collection

In order to address the issue of information silos in various business systems, integration is carried out among them to achieve data sharing and information integration[6].

Data collection adopts two methods: timed pull and business real time push.

a) Business real time push method

In order to improve timeliness, the method proposed in this article provides a schedule data collection service.

When a new schedule is generated by the business, the interface is actively called to add schedule information; When the business modifies a new schedule, actively call the interface to change the schedule information; When a new schedule is deleted by the business, the interface is actively called to delete the schedule information.

The production side push mode has high real-time performance, with data available for each push, avoiding resource consumption through empty polling and reducing service pressure compared to the consumer side pull; Meanwhile, in push mode, consumers only need to provide an interface to receive data without any additional expenses. The push mode is suitable for application scenarios where data is dynamic and has strong real-time performance.

b) Timed pull method

For some business systems that only maintain operations without making functional adjustments, this article designs a scheduled task that actively pulls data from the business system by designing a reasonable scheduled task strategy. The consumer pull model can modify data formats according to consumer needs, reduce data redundancy, and meet personalized needs; At the same time, there is no need to maintain failed push notifications on the production end. The pull mode is suitable for application scenarios with diverse data requirements and weak real-time performance.

Combine push mode and pull mode according to actual needs, and complement each other's advantages.

2) Client data synchronization mechanism

Mobile end design reference [8] task visualization design, complete calendar based schedule viewing.Client data synchronization includes two synchronization mechanisms: manual refresh synchronization and automatic synchronization.

- Under the automatic synchronization mechanism, users can obtain information in real time.
- Under the refresh synchronization mechanism, users obtain information by refreshing.

The automatic synchronization mechanism has strong timeliness, but the server-side pressure is high, which brings a significant burden to interface services, isolation devices, and databases. It is suitable for business systems with high real-time requirements but low concurrency; The refresh synchronization mechanism has low client pressure and allows users to obtain information by refreshing, making it suitable for most scenarios.

2.3 Create, Update, and Delete Schedule Processes

When the business system creates a schedule related business data, it triggers the server-side data collection process, as shown in Figure 2.

When a business system creates, updates, or deletes a schedule, the business database is first updated. For business systems that support transformation, the departure push interface is called in real-time to push schedule related data to the schedule. For business systems that do not support transformation, the schedule service initiates scheduled tasks, directly pulls data and puts it into the schedule database.

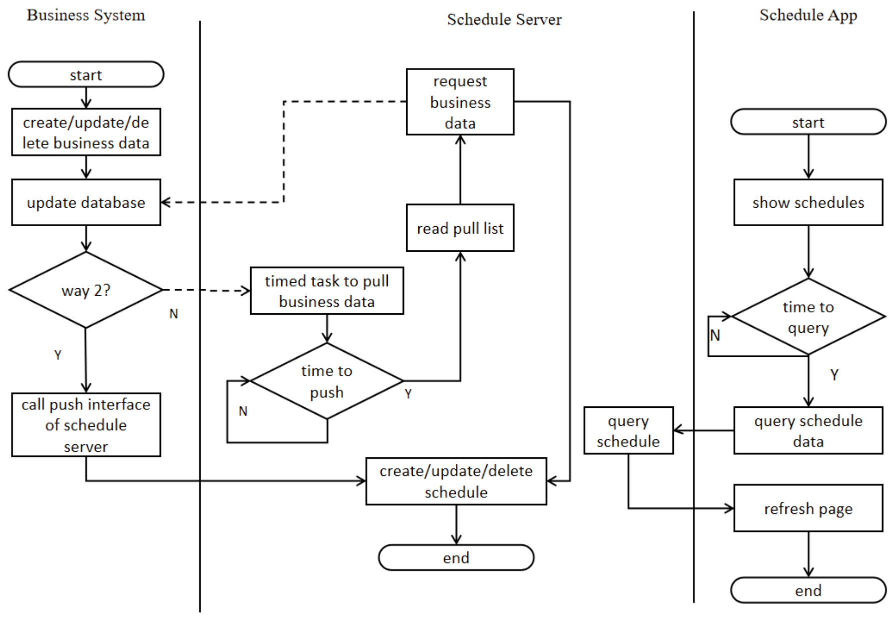

Fig. 2. Schedule business flow

When users view schedule related data on the mobile portal, if they have not left the schedule page, the schedule page will periodically request the schedule service to obtain the latest schedule data and complete the refresh display on the page.

3 Implementation and Application

3.1 Office Schedule Data Model

In order to achieve unified aggregation of office data, this article extracts the common characteristics of schedules in different businesses, mainly including three key information: schedule type, start and end time, and location, to complete the design of a unified data model which shows in table 1 for office schedules.

Table 1. Schedule Data Module

Filed Name	Data Type	Data Length
schedule_id	varchar	20
user_id	varchar	32
schedule_type	varchar	32
start_time	bigint	20
end_time	bigint	20
location	varchar	255

3.2 Data Collection Security

[9] A remote data collection system design method based on Java language was proposed, which uses MVC architecture and JDBC technology to complete the data collection function.

This solution is based on the Spring Boot framework and adopts a microservices architecture to achieve the functionality of data collection services.

In order to ensure the security and reliability of data collection, this system issues a unique key to third-party systems. Each application uses its own key acquisition interface to request a consumption token, thereby securely accessing the interface to push data. All data is encrypted before transmission, and the receiving side is responsible for decrypting the application. The Data security collection steps are show in figure 3.

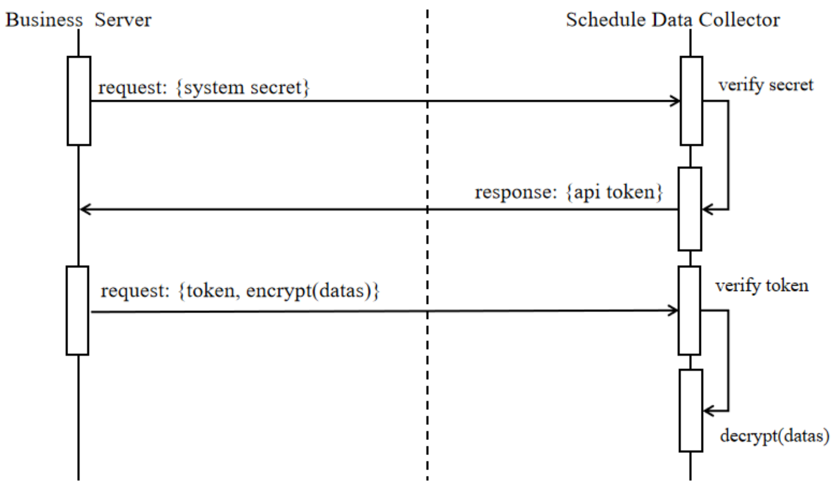

Fig. 3. Data security collection steps

3.3 Cycle Schedule Strategy

In order to fully utilize the periodic reminder of schedule reminders, this article designs and implements a schedule reminder strategy, including three methods: interval days, weekly repetition, and monthly repetition. Through flexible schedule reminder strategies, it meets the schedule needs of different scenarios.

- Interval Days Method: Set the schedule interval days to n, and if the current schedule date is “d”, the next reminder schedule date will be “d+n”, “d+2n”, “d+3n”, and so on.
- Weekly repetition method: Set the repetition date w for each Sunday schedule, so the next reminder schedule date is on the w of each week.
- Monthly recurrence method: Set the monthly schedule recurrence date m, so the next reminder date is m of each month.

3.4 Synchronization Strategy Display

In order to achieve timely and effective schedule updates while balancing the efficiency of business systems and schedule services, the backend pulls the scheduled service and sets the timing strategy to 5 minutes, while the frontend page queries and requests the scheduled task refresh strategy to 1 minute.

3.5 High Availability and High-Performance Implementation

In order to cope with the pressure of single point concurrency of schedule services in the case of excessive business data, this solution adopts a cluster deployment approach. A cluster refers to a distributed processing system or parallel processing system that is constructed by multiple computers in a high-speed network interconnection state, but is like a separate integrated computing resource to collaborate and achieve predetermined goals. Clustering can simplify system management and enhance system performance[10].This solution can achieve a schedule data pulling speed of 200 entries/second by deploying 6 node schedule pulling services and 3 node query services, meeting the concurrent request requirements of 100tps mobile schedule viewing.

3.6 Schedule Real Time Triggering Interface

1) Create Schedule Interface
  a) Request URI: /schedule/add.
  b) Request Method: post.
  c) Request Params

- userid: int, required.
- startTime: int, required.
- endTime: int, required.
- location: string, required.
- businessSys: string, required.
- businessId: string, required.

  d) Response Define

- code: string, required.
- message string, required.
- data: string, required.

2) Update Schedule Interface
a) Request URI: /schedule/update.
b) Request Method: post.
c) Request Params

- scheduleId: int, required.
- userid: int, required.
- startTime: int, required.
- endTime: int, required.
- location: string, required.
- businessSys: string, required.
- businessId: string, required.

d) Response Define

- code: string, required.
- message string, required.
- data: string, required.

3) Delete Schedule Interface
a) Request URI: /schedule/delete.
b) Request Method: post.
c) Request Params

- scheduleId: int, required.
- *d) Response Define*
- code: string, required.
- message string, required.
- data: string, required.

3.7 Application Effect

The results of this article were applied to a medium-sized enterprise with a population of ten thousand. After sorting out, the schedule data for travel and conference business was extracted, forming an office schedule application. We use data analysis methods to statistically analyze the results[7]. The monthly schedule data generated is shown in Table 2.

Table 2. Schedule Total Quantity

Calculate Classification	Business Travel Schedule	Offline Meeting Schedule	Online Meeting Schedule
Seven Day Cumulative Schedule Quantity	3	1	4087

Month Cumulative Schedule Quantity	14	16	19822
Total Cumulative Schedule Quantity	1876	1233	389251

4 Conclusions

This article implements a mobile unified schedule integration technology solution that integrates travel and conference related schedules. Due to the performance bottleneck of the server, the synchronization of page information is not real-time refreshing. In order to alleviate the performance pressure on the server, this article ensures the best experience while sacrificing some real-time performance. Through project practice, setting a pull time synchronization of 5 minutes and a page visualization automatic refresh of 1 minute can achieve the best user update experience and server performance. When facing larger enterprises with more complex data environments in the future, this system adopts a combination of scheduled collection and proactive business push to better adapt to the complexity of multiple business systems in the enterprise. By deploying the services of this system in a distributed and cross network manner, it can better leverage the system's ability to aggregate data for Daxing Enterprises..

Although this article solves the problem of data aggregation technology for large enterprises, the proposed solution lacks sufficient recognition of schedules in other office scenarios. Although the cycle strategy is flexible, it is not convenient for application operations, and the overall convenience still needs to be further improved. In the future, the combination of artificial intelligence technology and the ability of voice assistants combined with public applications to awaken business can be further utilized to enable users to carry out mobile work more efficiently and intelligently.

References

1. X. N. Wan, X. Z. Lin, G. S. Dong, J. Sun, Design and Application of Big Data, Cloud Computing and Middle Platform Technology in Templating Energy Enterprise Portal Projects, Contemporary Chemical Industry, 2024,53(01), 242-246, 10.13840/j.cnki.cn21-1457/tq.2024.01.044.
2. Chandra S. Amaravadi, Office Information Systems: A Retrospective and a Call to Arms, Journal of Software Engineering and Applications,, 2014 7(7) 700-712, 10.4236/jsea.2014.78065.
3. H. Y. Wu, P. Q. Lv, Design and Implementation of Company Office Automation System, Journal of Fujian Computer, 2023,39(08), 59-64, 10.16707/j.cnki.fjpc.2023.08.014.
4. Eli Rohn, CAS-Based Approach for Automatic Data Integration,American Journal of Operations Research, 2013(3), 181-186, 10.4236/ajor.2013.31A017.
5. J. J. Li, Z. F. Wu, M. D. He, J. C. He, J. Li, Design and Implementation of Intelligent Schedule Management Software, Science and Technology & Innovation, 2024(09), 49-51, 10.15913/j.cnki.kjycx.2024.09.013.
6. Y. Niu, The Application of Computer Technology in Office Automation, Technology Innovation and Application, 2024,14(08), 187-190, 10.19981/j.CN23-1581/G3.2024.08.043.
7. L. Rosaria, Valle. Ermelinda Della, Data Mining and Exploratory Data Analysis for the Evaluation of Job Satisfaction, iBussiness, 2011 3(4), 372-382, 10.4236/ib.2011.34050.

8. L. Wang, Design and Implementation of a Visual Personal Task Management System, 41-43, 2023,24(08), XINXI JILU CAILIAO,10.16009/j.cnki.cn13-1295/tq.2023.08.010.
9. G. X. Tan, J. Gang, Q. C. Qi, Design and Implementation of a Remote Data Collection System Based on Java Language,Science and Technology & Innovation, 2024(01), 19-22, 10.15913/j.cnki.kjycx.2024.01.005.
10. H. Wang, Research on High Availability of Data Cluster in OA Collaborative Office System, 2023,7(16), 143-146, 10.19850/j.cnki.2096-4706.2023.16.031.