Journal of Electronics and Information Science (2024)
Clausius Scientific Press, Canada

DOI: 10.23977/jeis.2024.090311
ISSN 2371-9524 Vol. 9 Num. 3

Design and Implementation of a Universal Authentication System for Web Services

Maoling Pen[1,a,*], Jiao He[1,b]

[1]Chongqing City Management College, Chongqing, 401331, China
[a]csxpml123@126.com, [b]san084@163.com
[*]Corresponding author

Keywords: Web Services; Unified Identity Authentication; Security; Access Control; Scalability

Abstract: With the rise of the Internet and information technology, various applications have emerged, complicating identity authentication across systems. To address this, we developed a secure unified identity authentication system for web services, integrating multiple encryption technologies to ensure user information's integrity and confidentiality. The system features a scalable and flexible architecture, including a front-end interface, back-end services, and database storage. User identity is secured via digital signatures, and multi-level protections like data encryption, access control, and log auditing enhance overall security. Single sign-on allows users to access all related systems with a single authentication, improving the user experience.

1. Introduction

1.1. Web Services

Web services are essential in SOA, providing cross-platform, standard, and language-neutral advantages. They enable scalable, interoperable communication using SOAP and XML, overcoming proprietary limitations [1]. These services, while individually simple, can be combined for complex business needs and can be found and connected online [2].

1.2. Overview of Unified Identity Authentication

With the growth of enterprise applications, unified identity authentication becomes vital for system management. Traditional username-password authentication is insecure and confusing. Consolidating authentication methods is essential for cross-system access, enabling single sign-on (SSO) and unified identity verification. SSO allows users to access multiple trusted systems with one login, ensuring secure and consistent access across applications.[3].

2. System Design

The system is a scalable, Java-based web application with a multi-layered structure, integrating various business systems for user convenience. It uses Vue3, TypeScript, and Element Plus on the

front end, and a back-end stack of Spring Boot, Spring Security, JWT, Redis, Mybatis-Plus, and PowerJob. It offers unified user management, CA authentication, and security audits, along with single sign-on and distributed hierarchical management. PKI components like CA, RA, and KM enhance enterprise security [4]. The framework is detailed in Figure 1.

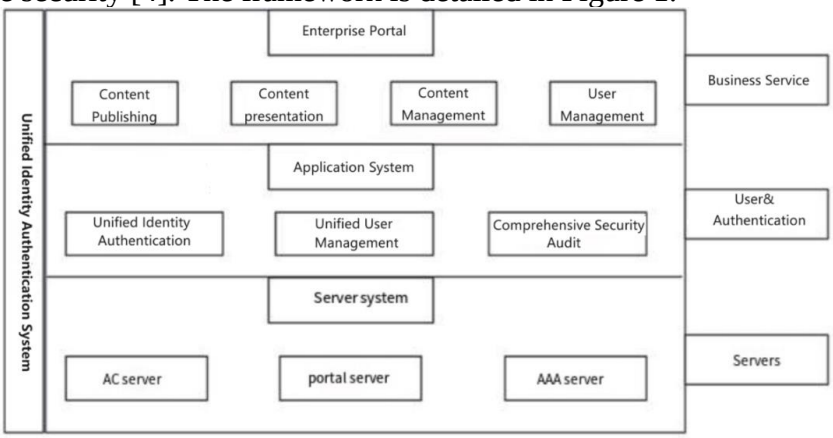

Figure 1: Overall technical framework

Users enter the service cluster through the gateway cluster, and the service cluster has three servers. After the front-end and back-end verification is successful, users can register and enter the monitoring center for task scheduling monitoring. The cache service is designed in a distributed manner, and a distributed lock is designed in the system. Documentation can be developed in the system, and file storage uses OSS services. Data interaction between the front-end and back-end is stored in MYSQL data tables. The overall design is shown in Figure 2.

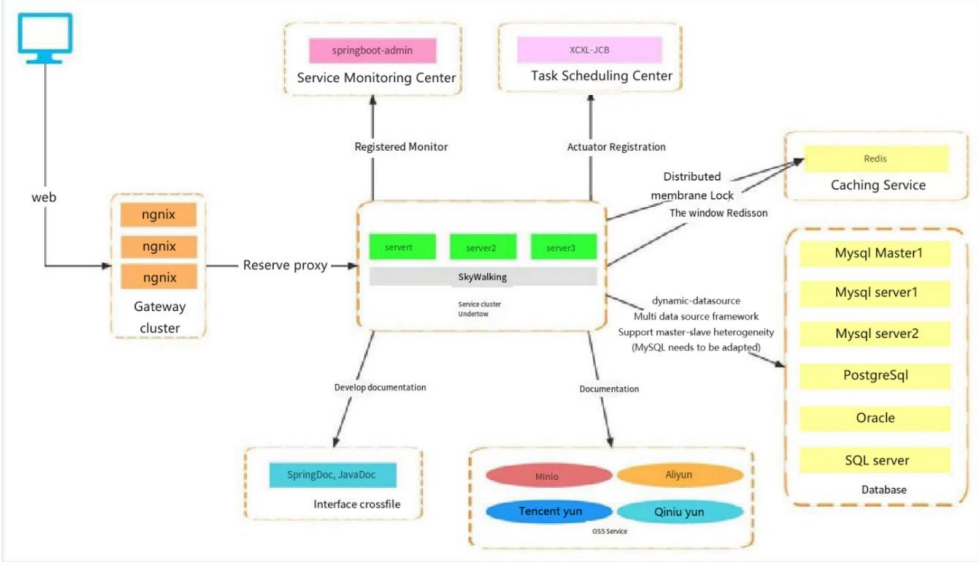

Figure 2: Overall system design

The unified identity authentication system comprises six core modules: User Management, offering self-services and announcements; System Management, covering user, role, menu, department, position, dictionary, parameter settings, announcements, logs, files, and client management; Tenant Management, handling tenants and package systems; System Monitoring, with features for user, cache, admin, and task monitoring; System Tools, for form construction and code generation; and Test Units, with test entities and tree tables [5].

3. System Implementation

3.1. Login and Registration

Identity verification and login are fundamental system functions. To counter brute force attacks and service misuse, a captcha is implemented alongside username and password authentication. Utilizing the WebSocket protocol, the client and server establish a TCP connection for real-time, two-way communication, with data stored in session storage and each user receiving a unique Session ID [6]. Web services manage authentication requests and sessions, tracking identities via session IDs to maintain user authentication and restrict access for unauthenticated users. The system comprises an AC server for authentication status determination and HTTP redirection, a Portal server offering login interfaces and password management, and an AAA server handling authentication, billing, and auditing [7].

3.2. System Management

The system encompasses user, role, menu, department, position, dictionary, and parameter management, along with notice announcements, log, file, and client management. Tenant management involves overseeing tenants and their package systems. Administrators can perform CRUD operations on user data, post announcements, and monitor logs to inform future system enhancements and updates.

3.3. Tenant Management

The tenant management part mainly involves the management of tenant information and tenant package management. Administrators can view the user's number, company department, tenant status, etc., and can also perform related add, delete, modify, and query operations.

3.4. System Monitoring

System monitoring can view information such as online users, cache status, and task scheduling, helping administrators understand the system's operating conditions, identify and resolve potential issues in a timely manner, and facilitate user management. In the event of a fault, the monitoring center issues an alarm in time to help administrators quickly locate and fix the problem. In addition, the monitoring center statistically analyzes system access requests to optimize system performance, improve access speed and response efficiency, and enhance the user experience [8].

3.5. System Tools

The system tools part is mainly an auxiliary work of the system, which can query data sources, table names, table descriptions, creation time, update time, etc.

4. System Testing

4.1. Testing Requirements

Functional testing requirements mainly involve testing the functions within the system to see if they meet the expected requirements. The functional testing requirements table is shown in Table 1.

Non-functional testing requirements mainly involve testing the system's compatibility, performance, and interface to see if the system can be used normally on the platform and whether

the performance meets market or company requirements. The requirements table is shown in Table 2.

Table 1: Functional test requirements table

Testing Requirement	Brief Introduction of the Testing Requirement	Priority
User Login	Users can log in to the system through account and password, or third-party applications.	2
Captcha Verification	The user login captcha refreshes and inputs normally.	1
System Management	Manage and view operations for users, roles, menus, departments, positions, dictionaries, parameters, announcements, logs, files, and clients.	1
Tenant Management	Manage and view operations for tenant management and tenant package management.	1
System Monitoring	Platform administrators can manage and view online users, cache monitoring, Admin monitoring, and task scheduling center.	1
System Tools	Manage and view operations for form construction and code generation.	1
Test Menu	Manage and view operations for form testing and test tree tables.	1
Logout	Logout is an operation where users choose to exit after completing tasks in the system.	3

Table 2: Table of non-functional test requirements

Test Requirement Item	Condition	Metrics
Performance	Query Response Time	Response time for query results should be within 3 seconds.
	Page Response Time	Page response time should be within 1 second
Interface Requirements	Use the universal graphical interface of Windows and must support mouse and keyboard.	User-friendly and can run normally
Compatibility	Support for Windows XP/7/8/10/11 systems, IE11 and above versions, 360, Chrome, Firefox, and other mainstream browsers.	The interface should display normally, and functions should run normally.

4.2. Testing Modules

(1)Registration Module

Tests user account creation with department selection and password entry, requiring password confirmation to prevent registration errors.

(2)Login Module

Assesses login functionality with correct credentials and captcha, ensuring failure with incorrect information.

(3)System Management Module

Verifies normal operation and correct viewing of system functions post-login, including user, role, and department management.

(4)Tenant Management Module

Manages tenant data, testing access to user management and tenant information post-login for administrative tasks.

(5)System Monitoring Management Module

Monitors user status, task processing, and file cache, ensuring operational visibility.

(6)System Tools Management Module

Tests access to auxiliary system functions post-login.

(7)Test Unit Management Module

Facilitates querying and operation of user and department data, testing function accessibility.

(8)Personal Information Viewing Module

Ensures users can view personal details post-login, with administrative access to tenant information.

(9)Information Add, Delete, Modify, and Query Module

Tests CRUD operations across the system for administrative personnel post-login.

5. Conclusions

The system utilizes unified identity authentication with encryption for secure user data management. It employs tokens as secure identity markers, granting access to services post-authentication to ensure user information's authenticity and confidentiality. The architecture is designed for scalability and maintainability, with efficient front-end, back-end, and database models, simplifying future management. Security is bolstered through encryption, access controls, logging, and audits to safeguard user data and minimize authentication response times, enhancing user experience.

Acknowledgements

This work is supported by the Natural Science Foundation of Chongqing (cstc2020jcyj-msxmX0876), the Science and Technology Research Project of Chongqing Education Commission (KJQN202203306)

References

[1] S Ramya, M Doraipandian, K Krithivasan, P Madhubala. Design and Implementation of Authentication for Remote Sensing Images in Traffic Congestion Control System using Hash Function in Graph Theory." 2023 2nd International Conference on Vision towards Emerging Trends in Communication and Networking Technologies (ViTECoN) (2023):1-5.

[2] Huang Hua. Design and Implementation of Unified Identity Authentication System for Web Applications [D]. University of Electronic Science and Technology, 2012.

[3] Jin Bin. Design and Implementation of a Unified Identity Authentication and Access Control Single Sign-On System [D]. Shanghai Jiao Tong University, 2007.

[4] Tong Min, Zhang Lina, Liang Wuqi. Design and Implementation of a JWT-based Distributed System Authentication and Authorization Mechanism [J]. Journal of Hefei Normal University, 2022, 40(03): 7-10.

[5] Wu Fan, Bian Jianling, Song Zhenqian, et al. Design Patterns and Applications of Microservices Software Architecture [J]. Digital Communication World, 2024, (01): 102-104.

[6] Yin Li, Du Xiaonan. Design and Implementation of a Campus Points System Based on SpringBoot and Vue [J]. Integrated Circuit Applications, 2023, 40(07): 414-415. DOI: 10.19339/j.issn.1674-2583.2023.07.187.

[7] Zheng Dongxi, Tang Shaohua, Li Shaofa. Web Service Unified Identity Authentication Protocol [J]. Journal of South China University of Technology (Natural Science Edition), 2005, (02): 65-69.

[8] Li Y J ,Shi H ,Deng Q Y , et al.An Improved Scheme of One-Time Password Identity Authentication Based on the S/KEY System[J].Applied Mechanics and Materials,2014,3468(644-650):2763-2767.