

SINGLE SIGN ON WITH DATA TRACKING

Mr. V. Nanda kumar^{*1}, Surya T^{*2}, Surya V^{*3}, Tharmarajan S^{*4}

^{*1}Professor, Department of CSE, PSNA CET, Dindigul, TamilNadu, India.

^{*2,3,4,5}Students, Department of CSE, PSNA CET, Dindigul, TamilNadu, India.

ABSTRACT

In today's digital era, users are increasingly accessing countless number applications every day. For accessing these services, the users first have to authenticate themselves and need to maintain a separate set of username and password for each application. This led to the development of Single Sign-On (SSO) is a session and user authentication service that permits a user to use one set of login credentials (e.g., username and password) to access multiple applications and also implement a data tracking which user can view what kind of data to be taken by client form SSO database. Existing methodology has solution for enterprises and organizations of all sizes, as it streamlines the management of various usernames and passwords, increases productivity, and minimizes administrative overhead only. Thus, our proposed methodology is shows a way to keep tracking the data which should take by oauth client. It will also keep a user's details safe and secure.

1. INTRODUCTION

Single sign on (SSO) is a concept that allows users to log in once and access multiple applications or services without having to enter credentials. With SSO, users log in only once and their authentication credentials are used to allow them to access other applications or services configured to use the SSO system. This approach simplifies the authentication process and increases security by reducing the number of passwords users must remember and providing centralized user access to applications and information. SSO is widely used in business environments, online services, and cloud applications to simplify the user experience and increase security.

2. PROBLEM STATEMENT

SSO addresses the need for users to have multiple credentials to access multiple applications and systems. In organizations where users need to access multiple applications or systems, each application usually requires a set of login credentials; this can be difficult to remember and can compromise security if multiple accounts use the same password. Additionally, managing user accounts and access rights across multiple systems can be time-consuming and resource intensive for the IT department. SSO solves these problems by allowing users to log in once and access multiple applications, while also simplifying the management of user accounts and access rights. However, using SSO brings its own challenges, such as integrating with existing systems and ensuring authentication and authorization.

3. EXISTING SYSTEM

Microsoft's Active Directory is a widely used service that includes SSO functionality. AD is widely used in Windows environments and provides users with a single set of credentials to access multiple applications. Lightweight Directory Access Protocol (LDAP) is an open, vendor-neutral protocol for accessing and managing distributed directory services. It can be used to provide SSO functionality in a variety of applications and systems. Security Assertion Markup Language (SAML) is an XML-based protocol for exchanging authentication and authorization information between two parties. It is widely used for SSO in web applications and is supported by many identity and management solutions. OAuth is an open authorization standard widely used in modern web applications. It can be used to provide SSO functionality and allows users to authorize applications to access their data without having to provide separate login credentials. OpenID Connect (ODC) is an authentication protocol built on OAuth 2.0 that provides SSO functionality.

It is often used in web applications to allow users to authenticate once and access multiple applications.

4. DRAWBACKS OF EXISTING SYSTEM

- **Single point of failure** -
If the SSO server or system fails, users will lose access to all applications that depend on it, which can significantly disrupt business operations.
- **Security Issues** -
SSO offers a key access point to multiple applications, increasing the risk of unauthorized access should the SSO system be compromised.

- **Implementation Complexity -**

Implementing SSO requires careful planning and coordination across multiple systems and applications. This can be difficult and time consuming, especially in large organizations with multiple assets.

- **Limited Compatibility -**

Not all applications or systems are compatible with SSO options, which can limit the benefits of using SSO.

- **Cost -**

Implementing an SSO system can require significant hardware, software and personnel investment to support and maintain the system, which can be prohibitive for some organizations.

5. PROPOSED SYSTEM

The proposed work for implementing a Single Sign-On system entails a careful evaluation of the organization's requirements, choosing a suitable vendor, designing and configuring the system, testing and validating it, deploying and educating the users and administrators, as well as routine monitoring and maintenance. The objective is to develop an SSO system that is effective, secure, and user-friendly and that satisfies the organization's unique objectives and specifications.

- **Redundancy and failover –**

An SSO implementation should include redundant and failover servers to ensure that users can continue to access their applications in the event of a server failure.

- **Strong Authentication and Authorization -**

SSO applications must provide strong authentication and authorization to ensure that only authorized users can access the application. This may include multiple authentication, access controls, and capability checks.

- **Compatibility -**

SSO applications must be compatible with a variety of applications and systems to maximize the benefits of SSO. This will require the use of multiple SSO protocols and integration with legacy systems.

- **User Experience -**

The SSO implementation should provide users with a consistent and efficient experience, provide easy access to applications and simplify the login process.

- **Scalability and Performance -**

The SSO application must be scalable and able to manage many users and applications. It should also provide high performance and low latency to reduce user stress.

- **Cost Effective -** Implementing SSO must be cost effective with minimal hardware and software and efficient use of IT resources.

6. FEASIBILITY STUDY

Every system is feasible if they have unlimited resources and infinite time. The primary objective of the feasibility study remains testing the Technical, Operational and Economical feasibility for adding new modules. All systems are feasible if they have unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

6.1 TECHNICAL FEASIBILITY

Frequently asked questions during a feasibility study include:

- Are there appropriate tools for implementing the recommendations?
- Is the equipment under preparation capable of storing the information needed to implement the new system?
- Will the app adequately answer questions regardless of the number or location of users?
- Is it possible to change the system after creating it?
- Is there assurance of accuracy, reliability, availability and data security?

Early systems did not exist to meet "security infrastructure" requirements.

6.2 OPERATIONAL FEASIBILITY

A project plan is only useful if it can be turned into information. This will be done according to the work of the institution. The effective implementation of the project will be considered as an important part of the implementation of the project.

Some important questions to ask to evaluate the effectiveness of the program are:

- Are there enough users to support the administration?
- If it is built and used, will it be used and will it work properly?
- Will there be objections from users that will affect the results of the application?

The system is designed for the above problems. Administrative and user problems should be considered first. So there is no doubt that user protection will destroy a good app.

A well-planned program will ensure efficient use of computer resources and help increase performance.

6.3 ECONOMICAL FEASIBILITY

It is technically possible to create a system that will be a good investment for the organization if it is installed and use . In terms of economic potential, the development costs of building a system are judged by the best results of the new system. The financial benefits must equal or exceed the costs.

The system is economically viable. It does not require any additional hardware or software.

This system has some economic and financial benefits as it is interfaced using existing resources and technology that the NIC can use.

7. SYSTEM ARCHITECTURE

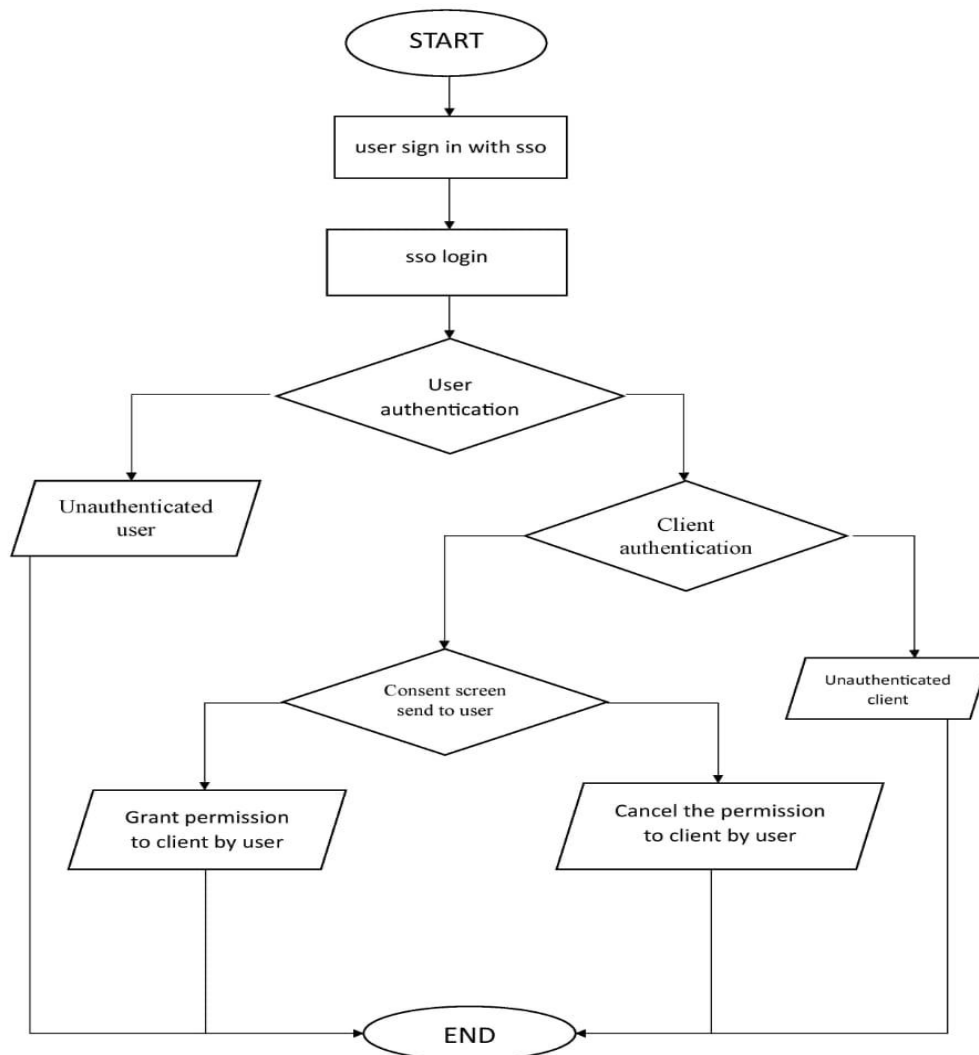


Figure 1: System architecture diagram

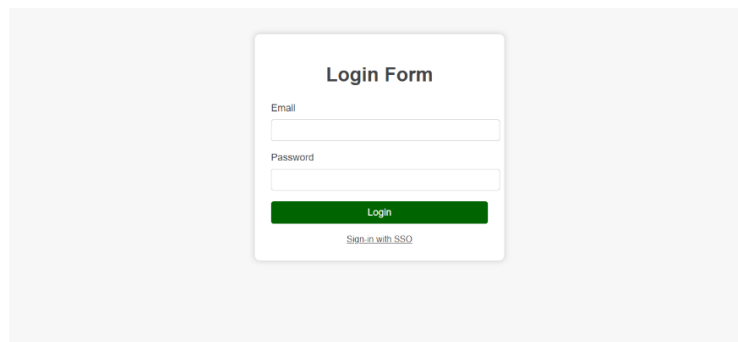
8. METHODOLOGY

- **Authentication Server** – The authentication server the basis of a signature in the system. It is responsible for authenticating users and providing access to various applications.

- **User Directory** – The user directory is a central database that stores user account information including usernames, passwords, and other related information.
- **SSO Protocol** -
The SSO protocol is a method used by SSO systems to authenticate users and provide information about applications and users.
- **Identity Provider (IDP)** -
The Identity Provider is responsible for issuing and managing user IDs and credentials. IDPs are often integrated with authentication servers.
- **Service Provider (SP)** -
A service provider is an application that the user wants to access. SP relies on the SSO system to authenticate users and share user information.
- **Token Service** -
Token Service creates and manages tokens that users enter into the application. These tokens are generally shorter and provide more security than traditional passwords.
- **Federation** - Federation refers to the process of sharing personal information between different users of SSO systems. This allows users to access applications in different organizations without creating separate accounts for each organization.

9. RESULTS AND DISCUSSION

9.1 Sign in with SSO



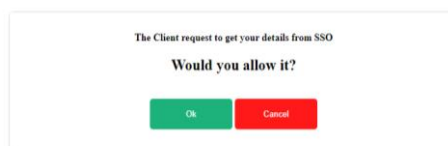
A screenshot of a 'Login Form' with a white background and a light gray border. It contains two input fields: 'Email' and 'Password'. Below the 'Password' field is a green 'Login' button. At the bottom of the form, there is a link that says 'Sign in with SSO'.

SSO Login



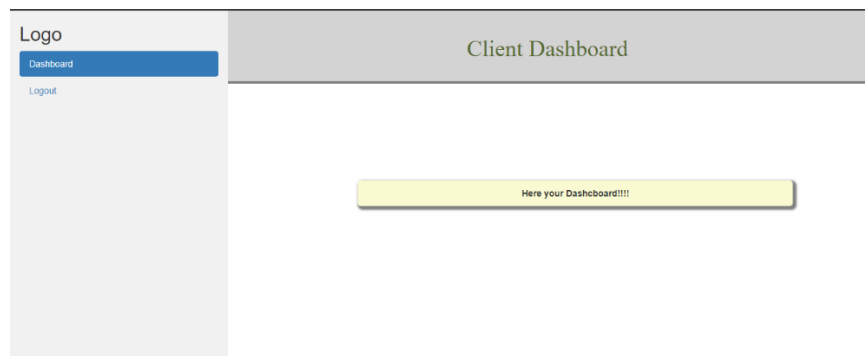
A screenshot of an 'SSO-Login Form' with a light gray background. It contains two input fields: 'Username' and 'Password'. Below the 'Password' field is a green 'Login' button.

Consent Screen



A screenshot of a 'Consent Screen' with a white background and a light gray border. It contains the text 'The Client request to get your details from SSO' and 'Would you allow it?'. Below the text are two buttons: a green 'Ok' button and a red 'Cancel' button.

After Successful Sign in Dashboard



10. CONCLUSION

A single set of login credentials, various apps or services can be accessed via a single sign-on (SSO). SSO has a number of advantages, including improved security, a better user experience, and easier identity managements limits the amount of passwords users must manage, which lowers the risk of security breaches connected to passwords. Users only need to remember one set of login credentials. SSO can also simplify the login process, improving both its effectiveness and usability. SSO can, from an administrative perspective, lighten the load on IT departments by streamlining identity management and lowering the volume of password reset requests. By centralizing access control and enabling finer-grained permission management, it can also increase security. SSO is a useful solution that can streamline the login process, improve security, and lessen the workload on IT departments. Even while it can need some initial setup and integration work, the advantages it offers make the cost well worth it.

11. REFERENCES

- [1] Wijayarathna C & Arachchilage N A 2021 An Empirical Usability Analysis of the Google Authentication API. In proceedings of the Evaluation and Assessment on Software Engineering 268 ACM.
- [2] Sciarretta G, Armando A, Carbone R, & Ranise S 2021 Security of Mobile Single Sign-On: A Rational Reconstruction of Facebook Login Solution. In SECURE 147.
- [3] Wang, Rui; Chen, Shuo; Wang, XiaoFeng (2021). Signing Me onto Your Accounts through Facebook and Google: A Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services.
- [4] Markov Sommer (September 4, 2021) use an single sign – on in a SAP for identity access management.
- [5] Aji Purwinarko, W Hardyanto, M A Adhi (June 2021) : Implementation of google single sign on(sso) in the library management system, Journal of Physics Conference Series.
- [6] Ramamoorthi L & Sarkar D 2022 Single Sign-On Implementation: Leveraging Browser Storage for Handling Tabbed Browsing Sign-outs. In Developments and Advances in Defense and Security 15 Springer, Singapore
- [7] Michele Benolli, Seyed Ali Mirheidari, Elham Arshad, and Bruno Crispo 2022. The Full Gamut of and Attack: An empirical Analysis of OAUTH CSRF in the Wild. In proceedings of the 18th International Conference on Detection of Intrusions and Malware and Vulnerability Assessment, Berlin, Heidelberg, 21-41.
- [8] OAUTH 2.0: Architectural design augmentation for mitigation of common security vulnerabilities on 7 January 2022, Journal of Information Security and Applications.
- [9] Marcia Costa march 17, 2022 Single Sign-On: International Conference on Surgical Cancer Care, Volume 23, Issue 4, E163, April 2022.
- [10] "Single Sign-On in Cloud Computing: A Review" by Sunil Kumar, published in the International Journal of Computer Applications in 2021.