Institute of Software Engineering
Software Quality and Architecture

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Master's Research Project Proposal

Reimplementation of OAuth 2.1 into Gropius

Nils Boike, Cara Heck, Johannes Zipfel

Course of Study: Software Engineering

Examiner: Prof. Dr.-Ing. Steffen Becker

Supervisor: Sandro Speth, M.Sc.

Abstract

This research is situated in the field of software engineering, specifically in authentication and authorization for users in web-based applications. The focus is on integrating OAuth authentication into Gropius, an issue management tool aimed at unifying workflows across multiple platforms like GitHub and Jira.
The current Gropius login service is stateless, but does not provide stateful sessions, requiring users to frequently re-authenticate and preventing seamless synchronization with external services. This results in a limited user experience and hinders integration capabilities.
The objective of this research is to design and implement an OAuth-based authentication system for Gropius that allows it to act both as an OAuth consumer (for external logins and data synchronization) and an OAuth provider (enabling Gropius accounts to be used in other applications while ensuring session-based access.
The research approach involves developing a new authentication service, incorporating session management and token-based synchronization for cross-service data handling. An appropriate database solution will be selected for secure token storage and session management, with consideration given to scalability, performance, and compatibility with the overall system architecture.
The implementation will enable session persistence, third-party authentication (e.g., GitHub and Jira), and token-based API synchronization for continuous data updates across platforms. It will enhance Gropius's interoperability and improve user experience by maintaining login states across sessions and devices.
Integrating OAuth into Gropius will significantly improve its authentication infrastructure, providing a more scalable, secure, and user-friendly system. This implementation will allow Gropius to better support cross-platform workflows, positioning it as a robust solution for unified issue management.

1 Introduction

Authentication and authorization are critical components of modern software engineering, particularly in web-based applications that handle sensitive user data and integrate with external services. OAuth [SC22], an open standard for access delegation, has become a widely adopted solution for enabling secure authentication and seamless integration between platforms. It allows applications to act as either consumers of third-party services (for features like social logins and data synchronization) or providers of authentication services for other systems. This research focuses on integrating OAuth into Gropius [SBB20], an issue management tool designed to unify workflows across platforms such as GitHub and Jira [24a; Atl].

The current Gropius login mechanism employs a stateless authentication service. As the current implementation does not use stateful sessions, the users have to frequently re-authenticate. [24b] This limitation impacts the user experience and creates barriers to integrating with external platforms, as there is no mechanism for persistent sessions. Addressing these challenges is essential to enhance Gropius's interoperability and provide a seamless experience for users who rely on multiple tools in their workflows.

To address these issues, this research aims to design and implement a new OAuth 2.1-based authentication system for Gropius. This system will allow Gropius to function both as an OAuth consumer—enabling third-party logins and synchronization with external platforms like GitHub—and as an OAuth provider, allowing other applications to authenticate users using their Gropius accounts. The solution will incorporate session-based [Gut01; HMP98] access to overcome the limitations of the existing stateless service, ensuring that users remain logged in across sessions and devices.

The research methodology involves developing a robust authentication service that combines session management with token-based synchronization for cross-service data handling. The implementation will consider scalability, security, and system compatibility, including the selection of a suitable database for managing session states and tokens. This approach will enable stateful user sessions, third-party authentication (e.g., via GitHub and Jira), and API-level data synchronization for real-time updates across integrated platforms.

The contribution of this research is the development of a comprehensive OAuth-based authentication system that enhances Gropius's ability to integrate with external platforms

and improves the overall user experience. By enabling session persistence, supporting third-party logins, and facilitating cross-platform data synchronization, this project positions Gropius as a more scalable, secure, and user-friendly tool for unified issue management. This advancement will strengthen Gropius's role as a central hub for managing workflows in diverse, interconnected environments.

Proposal Structure

Chapter 2 – Foundations and Related Work: Here, we provide the neccesary information about OAuth and the Gropius tool.

Chapter 3 – Objectives and Work Packages Here, we present the objectives and work packages of our research project.

2 Foundations and Related Work

2.1 Foundations

OAuth

OAuth [JH12] is an open standard for access delegation, commonly used to grant websites or applications limited access to user information without exposing passwords. It allows users to authorize third-party applications to access their data on another service without sharing their credentials. After the resource owner grants permission, the client receives an access token from the authorization server. This token is then used to access the protected resources on the resource server. If the access token expires, the client can use a refresh token (if provided) to obtain a new access token without requiring the resource owner to log in again.

As an OAuth consumer, Gropius can request limited access rights to external accounts (e.g., GitHub), allowing the synchronization of issues without requiring direct password storage or management. As an OAuth provider, Gropius would enable users to log in to other applications using their Gropius credentials, thereby fostering greater flexibility and integration.

Gropius

Gropius is a state-of-the-art tool designed to address the challenges of managing issues in distributed, component-based systems. Conventional issue management systems are often constrained to a project-specific scope, which limits their effectiveness in handling cross-component dependencies. Gropius overcomes this limitation through its integrated issue management capabilities, allowing users to create and manage issues that span multiple components. This approach ensures that interdependencies between components are captured and addressed systematically, enhancing coordination and collaboration.

One of Gropius's key features is its cross-component visibility, which documents architectural dependencies and links issues across components. This capability provides a comprehensive view of how issues propagate throughout the system, enabling better communication and understanding among development teams. Additionally, Gropius offers a user-friendly interface with a modern web-based front-end, including modules for user management, project management, and a graphical cross-component issue modeller, ensuring ease of use for both technical and nontechnical users.

Furthermore, Gropius excels in seamless integration with existing systems by acting as a wrapper over multiple issue management platforms. This design enables users to interact with Gropius rather than dealing with the complexity of each integrated system, thereby simplifying workflows. Its real-time synchronization ensures that any changes made in a component's issue management system are instantly propagated through Gropius, keeping all stakeholders updated and aligned.

While Gropius excels in cross-component issue management, its current authentication mechanism is limited by a stateless design. Users must frequently re-authenticate, and the system lacks robust integration with external services for seamless workflows. Building on the foundational achievements of Gropius, this research focuses on enhancing its authentication infrastructure by integrating an OAuth-based system. This implementation will provide session persistence, third-party authentication (e.g., GitHub and Jira), and token-based data synchronization, addressing the identified limitations and enabling Gropius to leverage its cross-platform potential fully.

2.2 Related Work

Session management is a critical aspect of modern web applications. It ensures that user interactions are maintained across multiple requests without requiring repeated logins. Existing solutions often rely on cookies or token-based mechanisms to manage sessions. However, these approaches can have limitations in terms of security and scalability. By integrating OAuth with robust session management, Gropius can enhance user experience by maintaining login states across sessions, reducing the need for frequent re-authentication, and improving overall security through better token handling and storage mechanisms.

3 Objectives and Work Packages

3.1 Objectives

The primary objective of this project is to integrate an OAuth-based authentication and synchronization service into Gropius, making it possible for users to:

- Log in to Gropius using external providers (e.g., GitHub, Jira).
- Maintain session persistence across multiple sessions.
- Synchronize issue data from platforms like GitHub and Jira to Gropius using tokens with appropriate permissions.

Main Research Questions

How can OAuth be effectively integrated into Gropius to provide session-based authentication and enable seamless synchronization with external platforms like GitHub and Jira?

3.2 Work Packages

WP1 Analysis of Requirements and Current Architecture

Goals

Analyze the current login architecure and define the requirements for integrating OAuth.

Research Questions

RQ1.1 What are the limitations of the current stateless login service in Gropius, and how can they be addressed with OAuth-based session management?

Tasks

This work package is split into the following tasks:

T1.1 Analyze Gropius's current stateless login setup and its limitations.

T1.2 Identify requirements for OAuth integration, including user needs for external authentication and issue synchronization, as well as an interface to the OAuth service.

WP2 Design of OAuth Consumer and Provider for Gropius

Goals

Design the functionalities to enable user login and session management

Research Questions

RQ2.1 Does every needed OAuth service work with the current Infrastructure of Gropius?

Tasks

This work package is split into the following tasks:

T2.1 Design the OAuth consumer functionality to allow users to log in using GitHub, Jira, and possibly other OAuth-compliant providers.

T2.2 Define the session management architecture, focusing on token storage, refresh capabilities, and multi-device access.

T2.3 Plan the OAuth provider implementation for Gropius to act as an authentication provider for external applications.

WP3 Implementation of OAuth Consumer and Provider Services

Goals

Implementing the previously designed methods

Research Questions

RQ3.1 How can token storage and session management be securely and efficiently implemented, and what role does the choice of database (e.g., PostgreSQL or MongoDB) play in this?

Tasks

This work package is split into the following tasks:

T3.1 Implement the OAuth consumer service using NestJS and PassportJS, enabling login via third-party providers and session-based access.

T3.2 Develop the synchronization API to handle token-based issue management, supporting GitHub and, potentially, Jira synchronization.

T3.3 Ensure compatibility with PostgreSQL for efficient token and session management, and optionally evaluate MongoDB for scalability.

WP4 Testing, Optimization, and Documentation

Goals

Include comprehensive tests, performance optimizations, and thorough documentation of all functions to ensure robustness, efficiency, and ease of maintenance.

Research Questions

RQ4.1 What are the best practices for documenting and maintaining OAuth-based authentication systems in the context of an open-source project like Gropius?

T4.1 Conduct thorough testing of all implemented OAuth flows and synchronization mechanisms, addressing potential security vulnerabilities.

T4.2 Optimize the architecture based on feedback from real-world testing scenarios.

T4.3 Document the code, providing comprehensive internal and external documentation for maintainability and future development.

WP5 Final Evaluation and Presentation

Goals

Evaluate the functionality and performance of the OAuth-based authentication system, ensuring it meets project objectives. The package will conclude with a presentation showcasing the implementation and its impact on Gropius.

T5.1 Evaluate the OAuth integration's performance and user experience within Gropius.

T5.2 Gather feedback.

T5.3 Prepare a presentation.

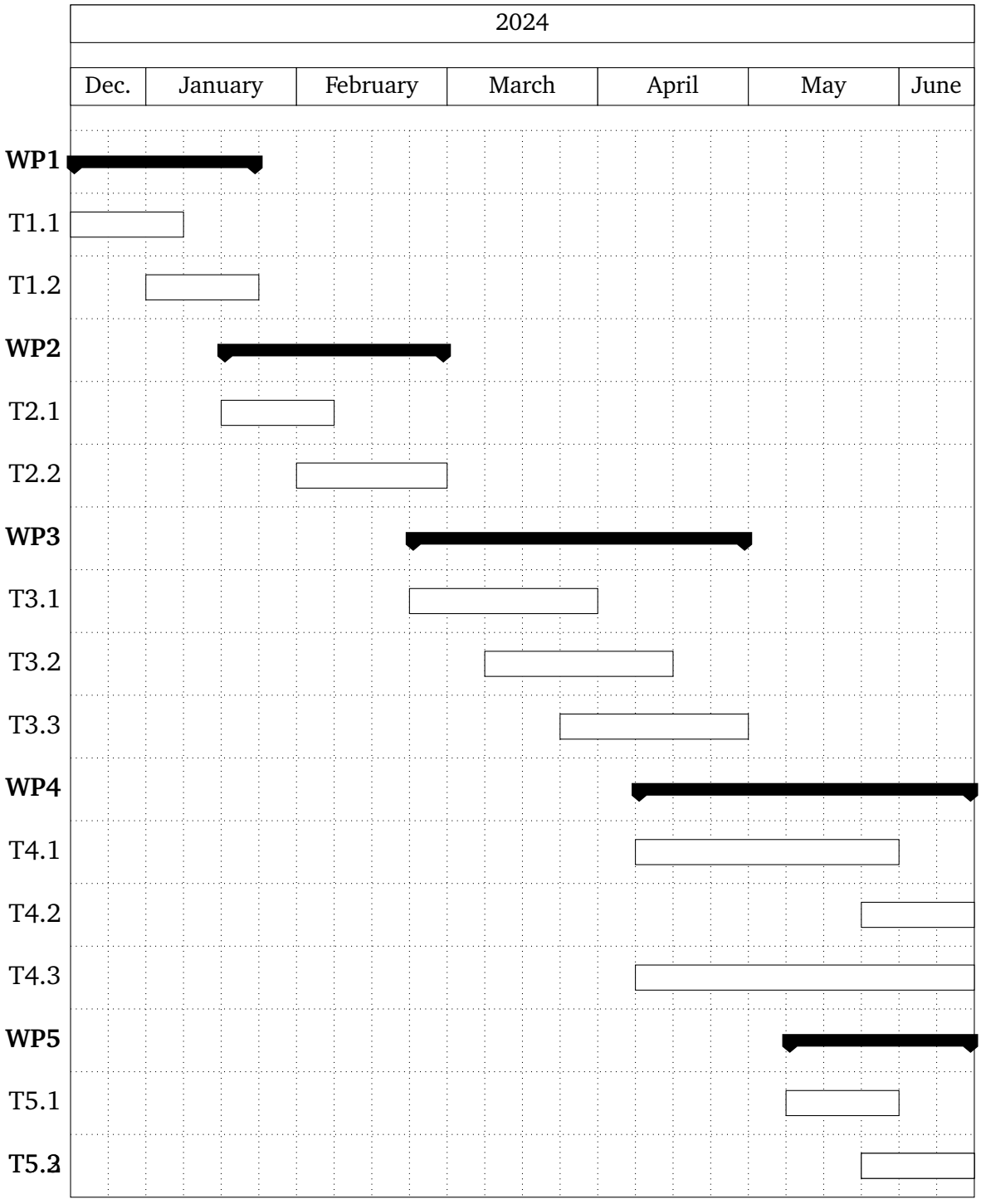

Figure 3.1: Gantt Chart for Integrating OAuth into Gropius

Bibliography

[24a] 2024. URL: https://github.com/ (cit. on p. 1).

[24b] *Gropius: a cross-component issue management system for component-based architectures*. 2024. URL: https://github.com/ccims (cit. on p. 1).

[Atl] Atlassian. *Überzeugende Ergebnisse Beginnen mit jira*. URL: https://www.atlassian.com/de/software/jira (cit. on p. 1).

[Gut01] K. Gutzmann. "Access control and session management in the HTTP environment." In: *IEEE Internet Computing* 5.1 (2001), pp. 26–35. DOI: 10.1109/4236.895139 (cit. on p. 1).

[HMP98] S. Hadjiefthymiades, D. Martakos, C. Petrou. "State management in WWW database applications." In: *Proceedings. The Twenty-Second Annual International Computer Software and Applications Conference (Compsac '98) (Cat. No.98CB 36241)*. 1998, pp. 442–448. DOI: 10.1109/CMPSAC.1998.716693 (cit. on p. 1).

[JH12] M. Jones, D. Hardt. *The oauth 2.0 authorization framework: Bearer token usage*. Tech. rep. 2012 (cit. on p. 3).

[SBB20] S. Speth, U. Breitenbücher, S. Becker. "Gropius — A Tool for Managing Cross-component Issues." In: *Software Architecture*. Ed. by H. Muccini, P. Avgeriou, B. Buhnova, J. Camara, M. Caporuscio, M. Franzago, A. Koziolek, P. Scandurra, C. Trubiani, D. Weyns, U. Zdun. Cham: Springer International Publishing, 2020, pp. 82–94. ISBN: 978-3-030-59155-7 (cit. on p. 1).

[SC22] J. Singh, N. K. Chaudhary. "OAuth 2.0 : Architectural design augmentation for mitigation of common security vulnerabilities." In: *Journal of Information Security and Applications* 65 (2022), p. 103091. ISSN: 2214-2126. DOI: https://doi.org/10.1016/j.jisa.2021.103091. URL: https://www.sciencedirect.com/science/article/pii/S2214212621002684 (cit. on p. 1).

All links were last checked on December 9, 2024.