PROJECT DELIVERABLE REPORT

Grant Agreement Number: 101058732

Joint Industrial Data Exchange Platform

Type: other

Project's source control repository

Issuing partner	Brunel University London
Participating partners	All
Document name and revision	Project's source control repository
Author(s)	Dr. Florian Maurer (FHV) Dr Nithin Amirth Jayasree (BUL) Dr Faranak Bahrami (BUL) Miah Raihan Mahmud Arman (TVS)
Deliverable due date	
Actual submission date	06/11/2023

Project Coordinator	Vorarlberg University of Applied Sciences
Tel	+43 (0) 5572 792 7128
E-mail	florian.maurer@fhv.at
Project website address	www.jidep.eu

Dissemination Level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission services)	
CO	Confidential, only for members of the consortium (including the Commission services)	
SEN	Sensitive, limited under the conditions of the Grant Agreement	

Contents

Executive Summary

This report provides an overview of the JIDEP project's source control repository, emphasizing its pivotal role in ensuring developmental consistency and collaborative efficiency.

The JIDEP project utilizes GitHub[1] as its primary source control repository, taking advantage of the Github Team[2] feature to manage a JIDEP development team actively working on WP2, WP3 and WP4. Within this team, eight dedicated team members are actively collaborating across eleven code repositories from WP2, WP3 and WP4. A code repository stores and manages source codes and documentation for a software ecosystem. It is a central hub where developers can store, organize, and track changes to their codebase over time. Using a code repository facilitates collaboration version control and often includes tools for building, testing, and deploying software. This report provides an overview of the JIDEP project's source control repository organization's structure, team roles, repositories, collaboration practices, and security measures in place, ensuring smooth operations and safeguarding the JIDEP project's integrity.

[1] https://github.com
[2] https://github.com/team

1. Introduction

JIDEP offers EU industries a unique platform anchored by a decentralized blockchain framework. This design uniquely positions JIDEP to navigate challenges like trust deficiencies, non-repudiation concerns, and ensuring unequivocal identities and data ownership.

A pivotal element in JIDEP's European deliverable is the strategic handling of its online repository. Acknowledging the indispensable role of source control in preserving the project's integrity, facilitating its growth, and ensuring its adaptability, our methodology emphasizes:

- Transparency: Reflecting the decentralized nature of JIDEP and in alignment with European values of openness and accountability, our source control procedures ensure that every alteration, enhancement, or change is recorded with utmost transparency.

- Collaborative Facilitation: To serve the collaborative essence of the Horizon initiative and JIDEP's mission, our repository framework encourages concurrent developments, ensuring synergy without sacrificing quality or project coherence.

- Robust Security: In the face of rising cyber challenges, especially in a continental context, our source control infrastructure integrates advanced security protocols. This ensures robust protection against potential vulnerabilities, safeguarding both data and the codebase.

- Comprehensive Versioning: We maintain a detailed version history, enabling stakeholders across the EU to easily backtrack, refer, or build on any phase of the project, ensuring continuity and legacy preservation.

- Streamlined Integration & Deployment: Recognizing the evolving needs of European industries, our repository employs CI/CD methodologies for timely and accurate integrations, ensuring that the platform remains at the forefront of technological innovation.

In summary, JIDEP's commitment to its Horizon EU deliverable goes beyond code management. It's a dedication to embodying the ideals of the project, underlined by trust, collaboration, and data integrity, thereby contributing to the European vision for a cohesive and innovative industrial future.

As software development has evolved, so has the necessity for effective and efficient source control management. The JIDEP project has always emphasized the importance of a robust source control system. Such a system is integral not just for efficient version control but also for facilitating effective team collaboration. GitHub, in this context, has been our system of choice. GitHub, a leading platform in this domain, provides tools and features that enable projects like JIDEP to streamline software development workflow, collaborate seamlessly, and maintain source code security. This report provides an overview of the JIDEP project’s source control organization structure, software development team compositions and roles, repositories overview, collaboration practices, access control and security.

2. Organization Structure

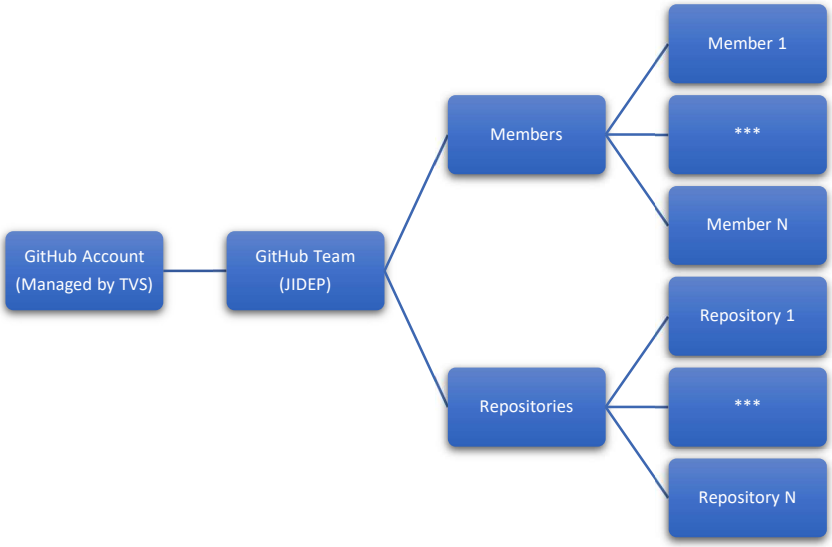

Figure 1 JIDEP source control organization structure

As illustrated in Figure 1, JIDEP's source control repository is managed by using the GitHub Team feature under the TVS's GitHub account. We havve established a team named "JIDEP", further showcased in Figure 2. In this JIDEP team, multiple repositories were created, and members from WP2, WP3 and WP4 were added to collaborate on repositories.

The GitHub Team feature provides the JIDEP project with a platform for streamlined repository management, efficient collaboration, and clear role definition. It also offers us centralization, access control, and dedicated spaces for discussions and comments on source codes, which enhance team productivity.

All our repositories, tools, and team members are under one umbrella. This not only aids in easier management but also ensures that every resource is just a few clicks away for any team member. Moreover, the hierarchical arrangement streamlines the decision-making processes, allowing for rapid yet informed choices.

Lastly, one of the significant benefits we've realized is the ease with which we can define and manage access levels. From administrators to contributors, every role is clearly defined, ensuring transparency and security.

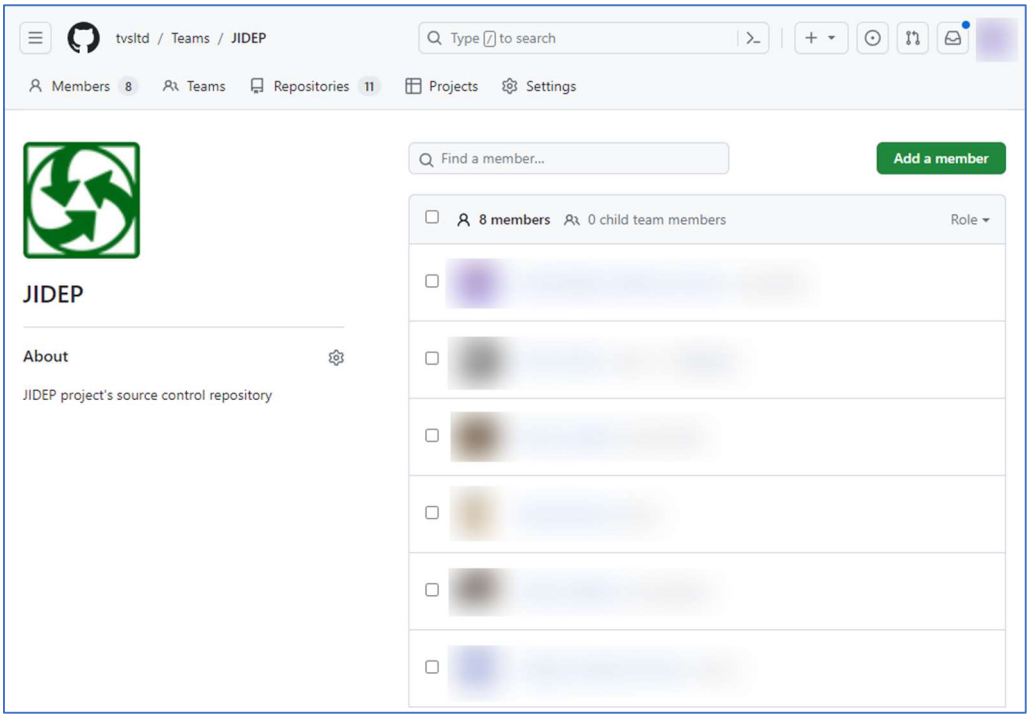

Figure 2 JIDEP source control repository team

3. Roles and Permissions

The JIDEP team, comprising diverse roles, collaboratively utilizes the GitHub repository to manage and streamline the development process.

The JIDEP team within TVS's GitHub account allow us to manage repository access and permissions in a more structured way. Regarding team member roles, GitHub offers a more streamlined set of roles. Here's a summary:

- **Maintainer:** Maintainers manage the team's details within a GitHub account. They can add or remove members from the team, manage the team's associated repositories, and edit the team's settings. They essentially oversee the team's functionality but don't have overarching control of the entire GitHub account like Owners do.

- **Member:** Members are part of the team and have access rights to the team's repositories based on the permissions set by Owners or Maintainers. Typically, a member has read access by default, but this can be customized to grant them more specific permissions like write or admin rights to certain repositories.

Within the context of a team's repositories, these team members can also be granted varying permissions:

- **Read:** Members can view and clone the repository but cannot push changes.
- **Write:** Members can pull and push code to the repository, manage issues, and initiate pull requests.
- **Admin:** Members can manage the repository's settings, including adding collaborators, managing branches, and other administrative tasks.

These team member roles and permissions are specific to the JIDEP team within TVS's GitHub account. They provide granular control over repository access, ensuring the right

people have the necessary permissions, which aids in efficient collaboration and maintains the security and integrity of the projects.

4. Access Control and Security

Ensuring the security and preserving the integrity of our source code stands at the forefront of our priorities. We are using the following features from GitHub to ensure that:

.

- **User Roles and Permissions:** The GitHub Team's granular permission system allows us to assign roles like Maintainer and Member and permissions like Read, Write and Admin, ensuring everyone accesses only what they need.

- **Two-Factor Authentication (2FA):** Enforced for every team member, 2FA adds an extra layer of security, ensuring that access to our repositories is always in the right hands.

- **Audit Logs:** Regular monitoring of GitHub's audit logs provides insights into every activity. This helps in the early detection of any anomalies or unauthorized access.

5. Team Composition and Roles

Behind every successful project is a team of dedicated individuals, and the JIDEP project is no exception. Currently, The JIDEP source control team comprises eight members from WP2, WP3 and WP4. Each member brings a distinctive set of skills, ensuring a holistic development approach. Here's a summary:

- **Chief Technology Officer:**
  - **Role:** Maintainer
  - **Responsibilities:** As the project's technological visionary, the CTO might not directly interact with the codebase daily. However, they oversee the broader strategic direction. Using the repository, they can monitor the project's health, review critical commits, and ensure that the development aligns with its goals. Additionally, they might be involved in granting repository access permissions and managing top-level organizational settings on GitHub.
  - **Number:** 1

- **Lead Software Engineer:**
  - **Role:** Maintainer
  - **Responsibilities:** Acting as the bridge between vision and execution, the Lead Software Engineer frequently interacts with the repository. They're responsible for reviewing pull requests, ensuring code quality, and maintaining the main/master branch's integrity. Their role might also involve setting up branch policies, deciding on merging strategies and managing release tags.
  - **Number:** 1

- **Software Engineers and Researchers:**
  - **Role:** Member
  - **Responsibilities:** These are the primary users of the repository. Daily, they clone, branch, commit, and push their code. Software Engineers and Researchers work on feature branches, address issues raised in the repository, and submit pull requests for their code to be reviewed. They adhere to the collaboration practices set, such as following the specific Git branching model and contributing using conventional commits.
  - **Number:** 6

By assigning distinct roles and responsibilities in the repository context, the JIDEP team ensures a smooth, efficient, and organized development process. This structured approach promotes accountability, transparency, and effective collaboration in the software development lifecycle.

6. Repository Overview

Currently, the JIDEP project encompasses 11 diverse repositories depicted in Figure 3, each serving a specific purpose. These repositories, while distinct, are interlinked, representing the complex yet harmonious nature of the JIDEP project. These repositories are results of WP2, WP3 and WP4.

Each repository carries its history, challenges, and milestones. For instance, while one repository might be dedicated to user interface elements, another might be focused on ontology, and another might be focused on APIs/SDK. This compartmentalization ensures that experts in each domain can focus on what they do best.

However, despite their distinct nature, there is a unified guideline that all repositories adhere to. This includes coding standards, documentation practices, and review processes. Such standardization ensures consistency across the board.

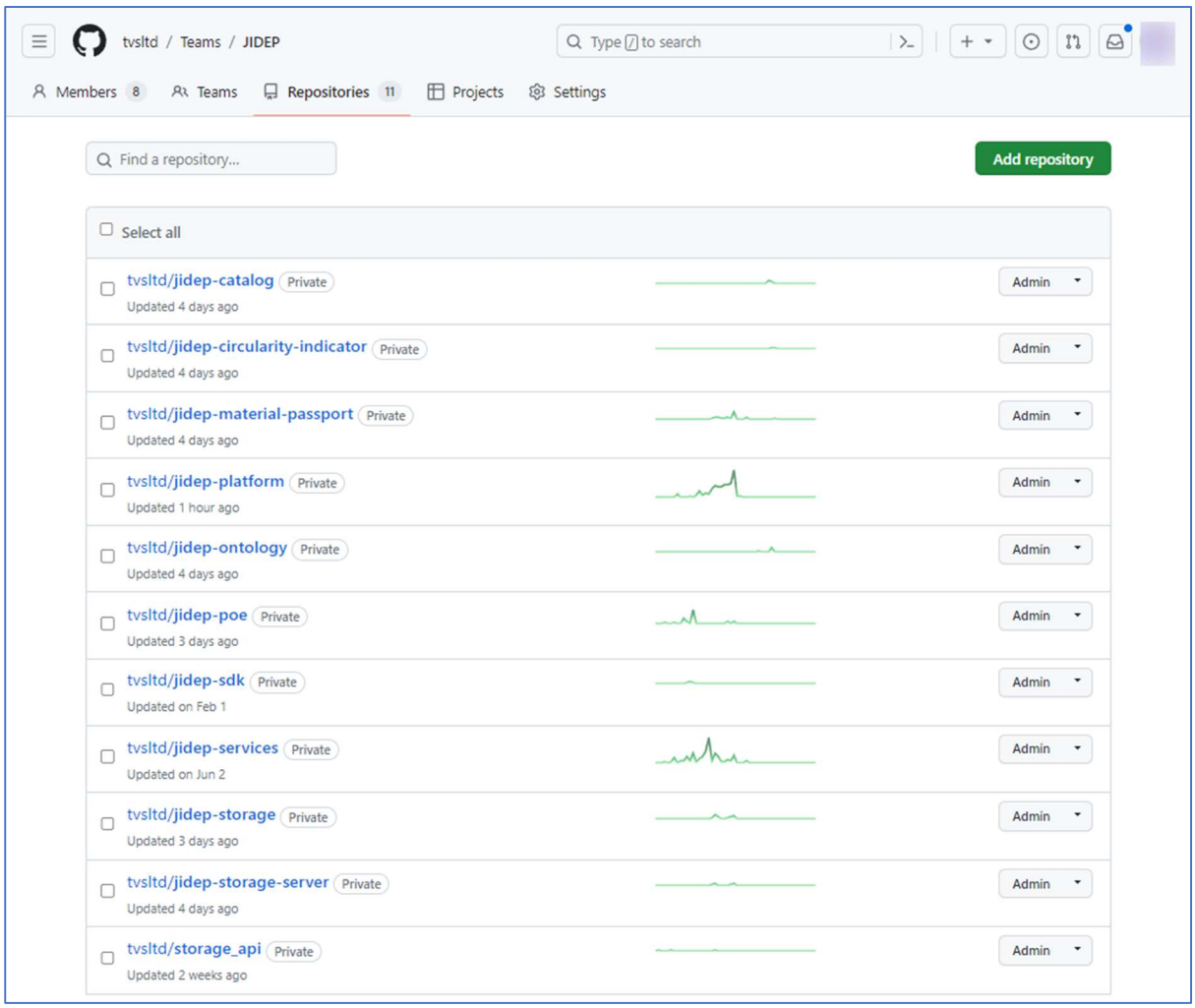

Figure 3 List of repositories under the JIDEP team

7. Collaboration Practices

Collaboration practices play a crucial role in the development process of any project. That is why the JIDEP project has adopted the Gitflow[3] branching model for the source control repositories. For each repository, this structured approach forces the creation of two branches: the Master Branch, which is for production code, and the Develop Branch, which is for upcoming developments and changes. Additionally, the branching model incorporates Feature Branches for new functionalities, Release Branches for finalizing new versions, and Hotfix Branches for immediate production fixes. This systematic design ensures a streamlined and efficient development process, tailored for team projects with defined release cycles.

In addition to the branching model, every code submission undergoes strict peer review. This practice ensures code quality and facilitates knowledge sharing and mentoring.

Furthermore, GitHub Actions[4] are integrated into our repositories, enabling continuous integration and continuous deployment (CI/CD). This means that our repositories enjoy automated testing and deployment, ensuring rapid delivery cycles without compromising code quality. We will discuss the branching model, code review, and CI/CD in more detail in the D4.2 Integration report.

8. Conclusion

In summary, JIDEP's commitment to its Horizon EU deliverable goes beyond code management. It's a dedication to embodying the ideals of the project, underlined by trust, collaboration, and data integrity, thereby contributing to the European vision for a cohesive and innovative industrial future.

This document offers insights into the structure and features of the JIDEP source control repository. Currently, The JIDEP source control repository is private and can only be accessible by authorised consortium members. For more in-depth information, the specific details of the platform will be elucidated in the D4.1 JIDEP tools and operation manual, set for release on M20 and M24.

The JIDEP project's source control and collaboration practices have not only streamlined our development process but have also fostered a culture of continuous learning and improvement. As we march forward, our commitment remains to refine these practices further, ensuring the project's ongoing success.

[3] https://nvie.com/posts/a-successful-git-branching-model
[4] https://github.com/features/actions