1 Introduction

In order for a user to interact with a website, an interactive element such as an input text is typically provided. Many of these input text-elements work by having a predefined SQL-Query-Template, for which the users input gets concatenated. This query then gets passed to the Database-Management-System (DBMS), which executes it and returns the result. Although this approach seems simple at first, it has some pitfalls which need to be addressed, as it is prone to attacks like SQL-Injections. According to the Open Worldwide Application Security Project (OWASP), which is a non-profit organization that aims to improve the security of software projects, SQL-Injections are still in the Top 10 Security Risks of 2021 and it is likely that they will remain in the Top 10 of 2025, taking into account that they were the number one attack in the Top 10 of 2017 [**?**].

1.1 Generating SQL-Queries dynamically

Instead of relying on predefined SQL-Templates, we would like to propose the idea of generating SQL-Queries dynamically using Machine-Learning-Techniques.
By generating SQL-Queries dynamically, the users wishes provided in the input text can be taken into account by the AI automatically, providing more specific search results initially. Given that the AI performs well enough, this could lead to future websites dropping filtering mechanisms entirely, such as filtering by specific brands or other features, as the AI would reflect the users wishes automatically in the generated Query, further enhancing the Human-Computer Interaction. This new kind of interaction is highlighted in Fig. 1.

1.2 Why a privacy-preserving AI is necessary

Although the idea of generating SQL-Queries automatically sounds intriguing, there are some practical challenges that have to be taken into account, when deploying the AI into a software project. Under the assumption that the AI performs well and that it is unrestrained, nothing would prevent the user from requesting search results from the AI, that contain confidential information like passwords, email-addresses or other criteria, which poses a similar threat that are resembled by SQL-Injections. Arguably, this would make the threat of SQL like Injections even worse, as users with no technical background are now able to query for confidential data as well. To mitigate that issue, we enforce a privacy-preserving AI, with the goal to never output SQL-queries that could disclose confidential data.

1.3 Gathering Data

TODO