



# «Danger Ranger» «Nachhaltigkeit in Gemeinden (PLANVAL AG)»

---

Technische Informationen für die Jury



## Technische Informationen für die Jury

### Aktueller Stand des Sourcecodes

- Link zu Github Repository: <https://github.com/magasser/bernhaeckt-2023>

### Ausgangslage

- Worauf habt ihr euch fokussiert?  
Wir haben uns darauf konzentriert die Daten der Planval AG bestmöglich in eine Datenbank auszulagern. Wir haben uns zuerst überlegt wie wir die 50 verschiedenen Metriken, welche zum Teil sehr unterschiedlich sind, in ein Passendes Schema bringen.  
Wir haben uns vorgenommen 3 verschiedenen Module zu erstellen. Das erste Modul wäre ein Admin interface, wo man die Daten der Planval AG verwalten kann. Hier haben wir uns auch überlegt, wie wir die Datenerhebung automatisieren können. Dafür haben wir in der Datenbank eine Source Tabelle erstellt in der API URIs hinterlegt sind. Diese Sources können dann beim Erfassen neuer Daten ausgewählt werden, um automatisch die Daten auszulesen. Man kann ausserdem auch direkt Werte eingeben, welche abgespeichert werden, falls es keine Passende Automatische Source gibt. Dieses Vorgehen ermöglicht es für den gleichen Datenindikator pro Gemeinde unterschiedliche Quellen zu verwenden.  
In der Datenbank sind neben den Werten pro Indikator und Gemeinde auch regeln hinterlegt. Mithilfe einer SQL-View werden die Regeln auf die gesetzten Daten angewendet um die Bewertung von 0-10 zu errechnen.  
Das Zweite Modul ist das Dashboard. Auf dem Dashboard können die verschiedenen Werte dargestellt und analysiert werden. Damit das Dashboard je nach Vorlieben angepasst werden kann, ist es komplett customizable. Dafür wird ein JSON definiert, indem die Config des Dashboards festgelegt ist.  
Das Dritte Modul wäre, um Umfragen zu erstellen und zu verwalten. Für dieses Modul wurde die Datenbank vorbereitet. Für ein UI hatten wir leider keine Zeit.
- Welche technischen Grundsatzentscheide habt ihr gefällt?  
Wir haben uns entschieden das Ganze Projekt mithilfe von Docker zu verwalten. Der Grund dafür ist, dass wir damit einfach das ganze deployen können.  
Wir haben uns entschieden den Ganzen Entwicklungsprozess mithilfe von Kanban durchzuführen. Wir haben uns auch für einen Monolithen gegenüber einer Microservice Architektur entschieden, um bei der Umsetzung schneller voranzukommen. Bei der Entwicklung haben wir zuerst das Datenbank Schema definiert, bevor wir mit dem Entwickeln des BE/FE angefangen haben. Wir haben im BE MVC angewendet.

### Technischer Aufbau

- Welche Komponenten und Frameworks habt ihr verwendet?  
Wir haben folgende Frameworks verwendet:  
Spring Boot, Java, Flyway, Swagger, Git, Angular, Material Design, Postgres, Docker, Docker-Compose, ChartJs, PrimeNg, Lombok, Maven, NPM
- Wozu und wie werden diese eingesetzt?  
Spring Boot in Kombination mit Spring Security und Java wurde verwendet, um die API zu definieren. JPA wurde verwendet um die Datenbank anzubinden. Postgres wurde verwendet, um die Daten zu verwalten. Flyway wurde verwendet, um die Datenbank Änderungen zu migrieren. Swagger wird verwendet um die API zu Dokumentieren und damit man sie auch ohne FE ausprobieren kann. Git in Kombination mit GitHub wurde verwendet, um den Source Code zu verwalten. Angular wird verwendet, um das Frontend aufzusetzen. Material Design zusammen mit SCSS wurde verwendet, um das Design der Webseiten festzulegen. Docker und Docker-Compose werden verwendet, um alle Container zu definieren und zu starten. ChartJs und PrimeNg werden verwendet, um Diagramme zu erstellen. Lombok wird verwendet, um den Boilerplate code im BE zu reduzieren.  
Maven und NPM werden verwendet um die Dependencies im FE / BE zu verwalten.

#### Implementation

- Gibt es etwas Spezielles, was ihr zur Implementation erwähnen wollt?  
Wir haben einen grossen Wert auf ein Modernes UI gelegt und auf eine klare Datenbank Struktur.
- Was ist aus technischer Sicht besonders cool an eurer Lösung?  
Besonders cool ist, dass man das Dashboard komplett customizen kann.

#### Abgrenzung / Offene Punkte

- Welche Abgrenzungen habt ihr bewusst vorgenommen und damit nicht implementiert? Weshalb?  
Wir haben uns Entschieden nicht alle Daten in die DB zu übernehmen. Grund dafür war, dass dies zu Zeitaufwändig ist, stattdessen haben wir uns beschränkt eine Handvoll Testdaten zu erstellen.  
Wir haben bei den Sources keine Echte APIs angebunden, sondern diesen Part wird momentan gemockt, auch hier ist der Grund, dass wir dies aus Zeitlichen Gründen gestrichen haben.  
Wir haben das Modul zum Erstellen und Verwalten von Umfragen nicht umgesetzt. Grund dafür ist auch hier, weil wir nicht genügend Zeit haben.