

# Dokumentation der Designentscheidungen nach dem ADR Modell

---

Im Kontext von der Entwicklung einer Anwendung welche eine Dokumentendruckschnittstelle sowie API-Schnittstellen "Fahrzeugentwicklung, Logistik" und eine selbst entwickelte Verwaltungsschnittstelle enthält, sind wir verschiedenen Problemen begegnet. Die Entwicklung einer solchen Anwendung erfordert eine sorgfältige Planung und Implementierung, um sicherzustellen, dass alle Komponenten nahtlos miteinander interagieren und die gewünschten Ergebnisse liefern. Während des Entwicklungsprozesses können jedoch verschiedene Probleme auftreten, die die erfolgreiche Umsetzung der Anwendung gefährden können. Einige dieser Probleme können technischer Natur sein, wie beispielsweise Konflikte bei der Integration von APIs oder Probleme bei der Datenübertragung zwischen den verschiedenen Komponenten der Anwendung. Andere Herausforderungen können jedoch auch organisatorischer oder prozessualer Natur sein, wie z.B. Probleme bei der Zusammenarbeit zwischen verschiedenen Teams oder Schwierigkeiten bei der Erstellung und Aktualisierung der Dokumentation für die Anwendung. Um diese Herausforderungen zu bewältigen und eine erfolgreiche Implementierung zu gewährleisten, ist es wichtig, geeignete Prozesse und Tools zu implementieren, die eine effektive Zusammenarbeit ermöglichen und die Qualität und Zuverlässigkeit der Anwendung sicherstellen.

## Inhaltsverzeichnis

---

- 1. [Entscheidung 1.0](#)
  - 1.1. [Threading zur Performanceverbesserung / Designentscheidung Johannes Wittwer](#)
  - 1.2. [Einheitliche & Standardisierte Struktur des Projektes durch die Verwendung von Maven / Christopher Caspari](#)
  - 1.3. [Verwendung von Docker Containern / Lukas Gabriel](#)

## 1. Entscheidung 1.0

### 1.1. Threading zur Performanceverbesserung / Designentscheidung Johannes Wittwer

Im Kontext der Nutzung der verschiedenen Schnittstellen, sind wir dem Problem der langsamen Antwortzeit der Schnittstellen begegnet. Wir haben uns für Threading entschieden um dadurch alle Schnittstellen gleichzeitig abzufragen und die Antwortzeiten zu verbessern. Dabei haben wir die Downside akzeptiert, dass die Programmierung etwas aufwendiger ist. Unsere Entscheidung ist auf die Qualität der Performance ausgelegt, wie auch definiert in unseren Hauptzielen.

### 1.2. Einheitliche & Standardisierte Struktur des Projektes durch die Verwendung von Maven / Christopher Caspari

Durch die Verwendung von Maven wird die Projektstruktur automatisch erstellt und die Verzeichnisse werden entsprechend der Standard-Konventionen benannt. Dadurch kann eine bessere Lesbarkeit und Wartbarkeit des Codes und die Zusammenarbeit mit anderen Entwicklern erleichtert werden. Maven kann darüber hinaus auch genutzt werden, um Abhängigkeiten zu verwalten anhand einer "pom.xml"-Datei welche die einzelnen Abhängigkeiten gliedert, Build-Prozesse automatisieren und ausführbare Artefakte erstellen. Das kann dazu beitragen, die Effizienz und Qualität der Entwicklung zu verbessern.

### 1.3. Verwendung von Docker Containern / Lukas Gabriel

Die Architektur soll in der Lage sein, eine steigende Anzahl von Anfragen zu bewältigen. Um dies zu erreichen, ist es sinnvoll, die Architektur durch Verwendung von Docker-Containern zu skalieren. Docker ist eine Plattform zur Entwicklung, Bereitstellung und Ausführung von Anwendungen in Containern, die eine schnelle Bereitstellung und Skalierung von Anwendungen ermöglicht. Durch die Verwendung von Docker-Containern kann die Architektur schnell und einfach auf verschiedene Umgebungen und Plattformen portiert werden. Zudem kann durch die Verwendung von Containern eine höhere Ausfallsicherheit erreicht werden, da bei einem Ausfall eines Containers nur dieser betroffen ist und nicht die gesamte Anwendung.