

## Designpaper 6.1

---

### Übersicht

**Projekt:** Projekt Episko

*Inkrement:* 6

*Arbeitspaket:* 1

**Autor:** Simon Blum

**Datum:** 22.01.2025

**Zuletzt geändert:**

*von:*

*am:*

**Version:** 1

**Prüfer:** Max Rodler

**Letzte Freigabe:**

*durch:* Max Rodler

*am:* 27.01.2025

### Changelog

---

Datum	Verfasser	Kurzbeschreibung
21.01.2025	Simon Blum	Initiales Erstellen und Verfassen

---

### Distribution List

- Simon Blum [simon21.blum@gmail.com](mailto:simon21.blum@gmail.com)
  - Ben Oeckl [ben@oeckl.com](mailto:ben@oeckl.com)
  - Maximilian Rodler [maximilianreinerrodler@gmail.com](mailto:maximilianreinerrodler@gmail.com)
  - Paul Stöckle [paul.stoeckle@t-online.de](mailto:paul.stoeckle@t-online.de)
- 

## Designpaper

Im Rahmen des Arbeitspakets soll eine Schnittstelle entstehen welche es erlaubt Manifestdateien zu serialisieren und zu deserialisieren.

### API Design

#### Serialisieren

- Input: Datenstruktur mit Metadaten
- Output: Erfolgreich?
- SideEffects: Datei wird geschrieben/erstellt #### Deserialisieren

- Input: Pfad
- Output: Datenstruktur mit Metadaten
- SideEffects: Datei wird gelesen

### **Code Aufbau**

Für das Erstellen der Schnittstelle wird ein Modul in der “episko\_lib” erstellt, welches sowohl für die Cli als auch für das Gui Backend zur Verfügung steht.

### **Referenz Klassendiagramm**

Im Bezug auf das Klassendiagramm werden folgende Klassen als “structs” entstehen: - Metadata - Language - IDE - BuildSystem - MetadataController (mit begrenzter Funktionalität) - FileSystemHandler (mit begrenzter Funktionalität)

### **Referenz Sequenzdiagramme**

In diesem Arbeitspaket sollen Abschnitte aus folgenden den Sequenzdiagrammen behandelt werden: - zu U1.2 (Sequenz zwischen MetadataController und FileSystemHandler, mit Schnittstelle zu App)