

# Abnahmeprotokoll

## AP2-Projektarbeit: Containerisiertes Framework zur automatisierten Kommunikation zwischen multiplen Clients mit integriertem Lasttestmodul

**Anwesende:**

**Datum:** 02.12.2024

- Alexander Knüppel (Projektleiter)
- Sven Lieckfeldt (Softwarebetreuer)
- Bumin Hatiboglu (Projektbetreuer)

### Vorstellung der Lösung:

Umsetzung eines Projektarbeit im Rahmen der AP2. Umgesetzt wurde ein Framework welches Kommunikation mittels REST-API und einer containerisierten Datenbank ermöglicht, um eine Umsetzbarkeit als Kommunikationsstruktur für eine Backup-Server-Architektur zu darzustellen und zu erleichtern.

### Bewertung der Umsetzung:

#### 1. Kompatibilität zum Registry-Server:

Die Anwendung gewährleistet eine erfolgreiche Kommunikation mittels REST-API, um die Kompatibilität mit der Schnittstelle des Registry-Servers zu demonstrieren. Da der Registry-Server zum Abnahmezeitpunkt nicht verfügbar ist, wurde die erfolgreiche Kommunikation zwischen dem Lasttestmodul und dem Kommunikations-Server als Erfolgskriterium herangezogen.

**Gewichtung: hoch**

**Entscheidung:**

☒ erfüllt      ☐ nicht erfüllt

#### 2. Anpassbarkeit und Erweiterbarkeit:

Die Anwendung wurde so gestaltet, dass sie anpassbar ist und zukünftige Erweiterungen problemlos integriert werden können.

**Gewichtung: hoch**

**Entscheidung:**

☒ erfüllt      ☐ nicht erfüllt

# Abnahmeprotokoll

## 3. Einhaltung der Projektzeit:

Das Projekt wurde innerhalb des festgelegten Zeitrahmens von maximal 80 Stunden abgeschlossen.

**Gewichtung: niedrig**

**Entscheidung:**

☒ erfüllt      ☐ nicht erfüllt

## 4. Verwendung von Open-Source-Lösungen:

Es wurden Open-Source-Lösungen verwendet, um Kosteneffizienz sowie Überprüf- und Wartbarkeit zu gewährleisten.

**Gewichtung: mittel**

**Entscheidung:**

☒ erfüllt      ☐ nicht erfüllt

## 5. Sicherheitskonzept:

Die Anwendung bietet grundlegende Sicherheitsmechanismen und ist anpassbar für zukünftige Sicherheitsimplementierungen, auch wenn ein umfassendes Sicherheitskonzept nicht vollständig implementiert sein muss.

**Gewichtung: mittel**

**Entscheidung:**

☒ erfüllt      ☐ nicht erfüllt

## 6. Dokumentation:

Alle erforderlichen Dokumentationen (Diagramme, Dokumentationen) wurden erstellt und bereitgestellt.

**Gewichtung: hoch**

**Entscheidung:**

☒ erfüllt      ☐ nicht erfüllt

**Unterschrift:** \_\_\_\_\_

# Abnahmeprotokoll

## Definitionen

### Zur Gewichtung:

**niedrig:** Relevanz für den Erfolg des Projekts niedrig, da Kernfunktionen und Erwartungen trotzdem erfüllt werden

**mittel:** Relevanz für den Erfolg des Projekts mittel, da Kernfunktionen und Erwartungen vermutlich erfüllt werden können. Muss vom Prüfer beurteilt werden.

**hoch:** Relevanz für den Erfolg des Projekts, da Kern-Funktionalität oder Hauptziel. Elementar wichtig. Muss vom Prüfer beurteilt werden.

### Begründung für die vergebenen Gewichtungen:

Die Gewichtungen wurden Anhand der im Lastenheft angegebenen Priorisierungen der gewünschten Ergebnisse und Inhalte des Projektes, bzw. des Projektzieles erstellt und müssen vom jeweiligen Prüfer nach bestem Wissen und Gewissen überprüft und beurteilt werden.

### Begründungen im Einzelnen:

1. Kernfunktionalität, muss zu den Schnittstellen des Registry-Servers kompatibel sein. Ausgewählte Schnittstelle des R.-Servers ist REST, Datentyp im Austausch ist JSON.
2. Kernfunktionalität, notwendige Flexibilitätsanforderung aufgrund der frühen Planungsphase der Architektur des H2Go-AAS-Netzwerkes.
3. Sekundär relevant da es sich eher um eine optionale Komponente und eher eine Machbarkeitsstudie handelt.
4. Sekundär relevant, da die Vorgabe als optionales „Kann“- , kein „Muss“-Ziel formuliert wurde.
5. Sekundär, da wie im Lastenheft formuliert Kompatibilität und Flexibilität wichtiger als eine fertig formuliertes Sicherheitskonzepts sind.
6. Kernziel, wichtig um Nachvollziehbarkeit und somit Wart- und Anpassbarkeit zu gewährleisten.