

ObJack Software

Einführung in die ObJack Microservices Architektur.

Stand: Dezember 2025
Herausgeber: ObJack GmbH
Supportanfragen: office@objack.at
Copyright: © 2025 ObJack GmbH

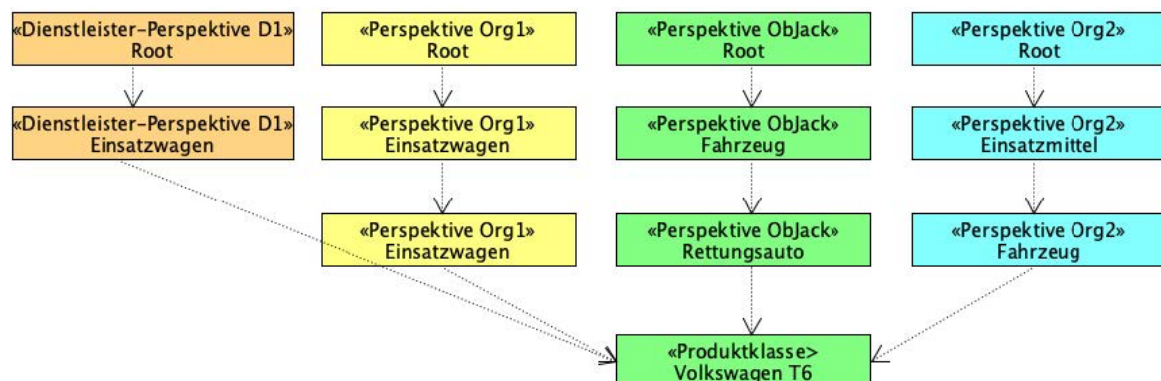
1. Einleitung

Die Architektur der ObJack-Plattform soll so weiterentwickelt werden, dass mehrere Instanzen parallel betrieben werden können und gleichzeitig ein Datenaustausch zwischen diesen Instanzen möglich ist. Das Datenmodell soll so flexibilisiert werden, dass Organisationen und Dienstleister eigene hierarchische Klassenstrukturen und eigene Klassen definieren und mit anderen teilen können.

2. Anforderungen

2.1 Flexibilisierung der Klassenstruktur (Klassen-Trees)

Die Hierarchie der Klassen soll so flexibilisiert werden, dass es nicht mehr nur eine, sondern beliebig viele Klassenhierarchien, also Kategorisierungen der Klassen, gibt. Dadurch ist es möglich, dass ein und dieselbe Produktklasse in unterschiedlichen Branchen oder Organisationen unterschiedlich in den Klassentree eingeordnet wird. Kunden (Organisationen) können fertige Klassentrees zur Verfügung gestellt werden, die sie dann auch selbst ändern können. Auch die Anlage vollkommen neuer Trees soll möglich sein. Organisationen sollten den Tree auch mit anderen teilen, also gleichsam veröffentlichen können. Klassen-Trees sollen auch ineinander integriert, also zusammengeführt werden können. Dabei kann der Benutzer einen Tree als Unterknoten in einen anderen Tree einfügen.



2.2 Organisationseigene Klassen

Zum gegenwärtigen Zeitpunkt können Klassen nur durch ObJack-Administratoren angelegt werden. Zukünftig sollen jedoch auch Organisationen (und damit auch Dienstleister) eigene Klassen anlegen können. Diese Klassen sollen (wie auch die Klassentrees) durch die Organisation global geteilt werden können. Geteilte Klassen oder Klassen-Trees müssen durch ObJack freigegeben werden, um für andere Organisationen sichtbar zu werden. Auch Dienstleister können Dienstleister-Klassen-Trees oder Klassenpriorisierungen definieren und teilen.

2.3 Organisationstrees

Wie im aktuellen Stand wird auch künftig zwischen Struktur- und Produktklassen unterschieden. Dabei können ausschließlich Objekte der Produktklassen, nicht jedoch der Strukturklassen, erzeugt werden.

Es gibt keinen globalen Klassen-Tree, sondern nur einen Standard-Tree, den die Organisation zusammen mit globalen Klassen, Datenfeldern und Modulen zur Verfügung gestellt bekommt. In diesem Fall werden Organisationsklassen und globale Klassen gemischt.

Damit die Information über die Art der globalen Klasse erhalten bleibt, muss der Branchen-Klassen-Tree-Path als Attribut in der globalen Klasse gespeichert werden. So kann man eine Klasse namens „A4“ beispielsweise als Fahrzeug und nicht als spezielles Gerät für den Audi A4 identifizieren. Alternativ kann diese Information als eindeutige Kategorie angegeben werden, um Verwirrung durch den Pfad zu vermeiden. Neu definierte Klassen müssen freigegeben werden, damit sie für die Organisation sichtbar werden.

Dienstleister Klassen und Trees: Der Dienstleister arbeitet mit den für die aktuell ausgewählte Anwenderorganisation priorisierten Klassen und Klassentrees, nicht mit eigenen. Die aktuell priorisierten Klassen und Objekte erhält der Dienstleister vom Dienstleister-Service.

2.4 Individuelle Objack Instanzen

Derzeit wird Objack als zentraler Server betrieben. Zukünftig sollen mehrere Server parallel betrieben werden können, wobei ein Datenaustausch aller Instanzen für zentrale Datenobjekte möglich sein muss. Zusätzlich sollen auch On-Premises-Instanzen unterstützt werden, die ebenfalls mit der zentralen Objack-Plattform integriert werden können, um einen Datenaustausch zu ermöglichen.

2.5 Admin Portal

Zur zentralen Administration der geteilten Daten soll ein Admin Portal zur Verfügung stehen. Dieses Portal soll die folgenden Daten/Aspekte zentral administrierbar machen:

- Instanzen (Kunden)
- Globalen Modulen & Datenfeldern
- Klassensammlungen (z. B. für Branchen wie RD, Immobilien, Gemeinden)
- Funktionssteuerung nach Lizenzmodell pro Instanz
- Onboarding-Prozess für neue Kunden:
 - Auswahl relevanter Klassen & Felder beim Start.
 - Vordefinierte Pakete (z. B. „Tankkarten“, „Medizinprodukte“)

2.6 Dienstleister Portal

Neben dem Admin-Portal soll es eine weitere zentrale Oberfläche für Dienstleister geben. Über dieses Portal können Dienstleister Dienstleistungen anbieten. Folgende Funktionalitäten sollen berücksichtigt werden:

- Das Dienstleister-Portal soll ein Marktplatz für Services innerhalb von ObJack werden.
- Dienstleister können Services (z. B. Wartungen, Reparaturen) anbieten.
- Es soll die Möglichkeit zur Bereitstellung von Reports und Benchmarks (z. B. Betriebskosten pro Fahrzeug) geben.

3. Technische Umsetzung

3.1 Umänderung der Primärschlüssel

Um verteilte Instanzen zu ermöglichen, die Daten untereinander austauschen können, muss eine Eindeutigkeit der Primärschlüssel über alle Instanzen hinweg gewährleistet sein. Aus diesem Grund muss eine Umstellung der Primärschlüssel auf UUIDs, die auf MariaDB UUIDs (Version 7) basieren, erfolgen.

3.2 Verteilung der Instanzen – ObJack Services

Die Instanzen behalten ihre Mandantenfähigkeit. Die Konfiguration der Instanzen wird über einen Spring-Config-Server zentralisiert. Dadurch können Konfigurationsänderungen zentral durchgeführt werden, ohne dass ein Redeployment der Instanzen notwendig ist. Die einzelnen Instanzen kennen unterschiedliche Zentral-Services, an die sie sich für die Abfrage/Speicherung globaler Daten wenden können. Dies sind aus heutiger Sicht:

- Zentraler Konfigurationsserver
- Zentraler Admin & Klassen-Service
- Zentraler Dienstleister Service

Dabei existieren die ObJack-Services, die die Daten einer oder mehrerer Organisationen beinhalten.

3.3 Zentraler Konfigurationsservice

Dieser Service ist die einzige Instanz, die ein ObJack-Service kennt. Er kontaktiert sie, um die IP-/DNS-Namen anderer zentraler Services zu erfahren. Jeder ObJack-Service verbindet sich mit den übrigen zentralen Services über eine Websocket-Verbindung. Damit kann auch mit den externen, On-Premises laufenden ObJack-Services bidirektional kommuniziert werden.

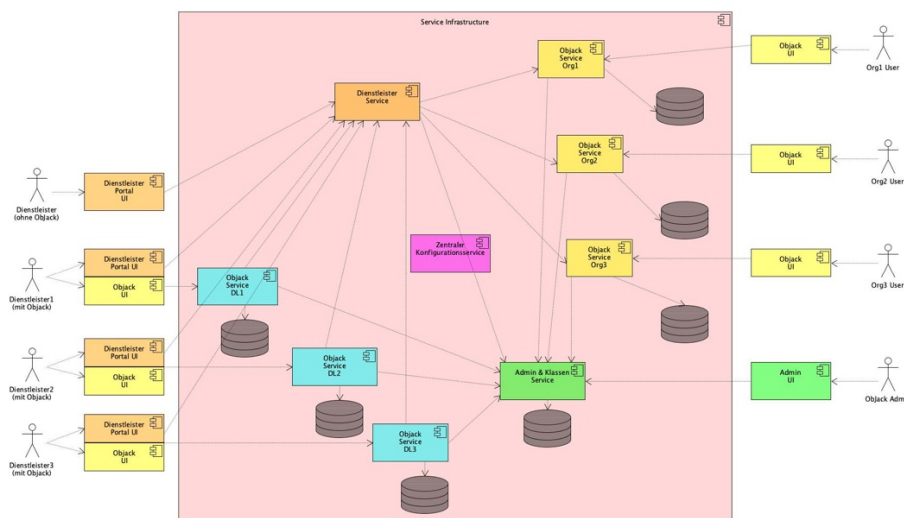
3.4 Zentraler Admin- und Klassen-Service

Der zentrale „Admin & Klassen“-Service ist für die Speicherung aller allgemein verfügbaren Produktklassen und Kategorisierungen (Klassen-Trees) verantwortlich. Jede Organisations-Instanz bezieht die Produktklassen von diesem Service und kann über diesen Service das Veröffentlichen eigener Produktklassen und Klassen-Trees beantragen. Zusätzlich kennt der „Admin & Klassen“-Service alle Organisationen und ermöglicht deren zentrale Verwaltung. Jede Organisation muss eindeutig einer Objack-Instanz zugeordnet sein. Jede Objack-Instanz kennt die Organisationen, für die sie selbst verantwortlich ist, aufgrund der in der Datenbank abgelegten Stammdaten der jeweiligen Organisation. Die Objack-Instanz gibt der zentralen Admin-Instanz über eine REST-Schnittstelle Auskunft, für welche Organisationen sie verantwortlich ist. Jeder Objack-Service baut mit dem Admin-&Klassen-Service eine Websocket-Verbindung auf. Basierend auf dieser Verbindung kennt der Admin-Service alle Objack-Services, die derzeit online sind, und weiß, für welche Organisationen sie jeweils verantwortlich sind. Über die WebSocket-Verbindung ist eine bidirektionale Kommunikation zwischen den Services möglich, auch wenn der Objack-Service in einer On-Premises-Infrastruktur des Kunden läuft.

3.5 Zentraler Dienstleister-Service

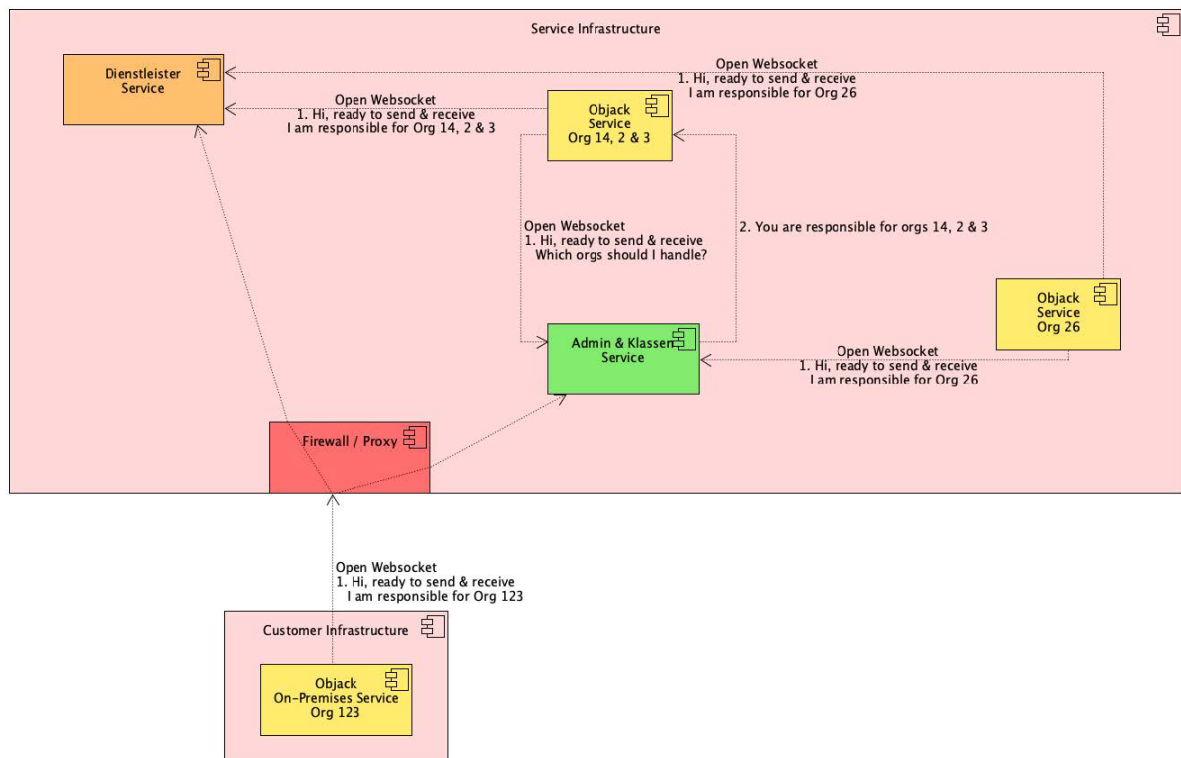
Der zentrale Dienstleister-Service kennt über den zentralen Admin-Service alle Organisations-Services. Er kennt alle Dienstleister, d. h. sowohl diejenigen, die selbst Objack verwenden, als auch diejenigen, die über Objack Dienstleistungen anbieten. Der Dienstleister-Service weiß, welchem Dienstleister welche Kundenorganisationen zugeordnet sind. Eine REST-Anfrage eines Dienstleisters könnte beispielsweise lauten: „Gib mir alle anstehenden Wartungen mir zugeordneter Organisationen.“ Der Dienstleister-Service kontaktiert daraufhin alle relevanten Organisationen und sammelt die Informationen für den Dienstleister ein.

3.6 Service Übersicht



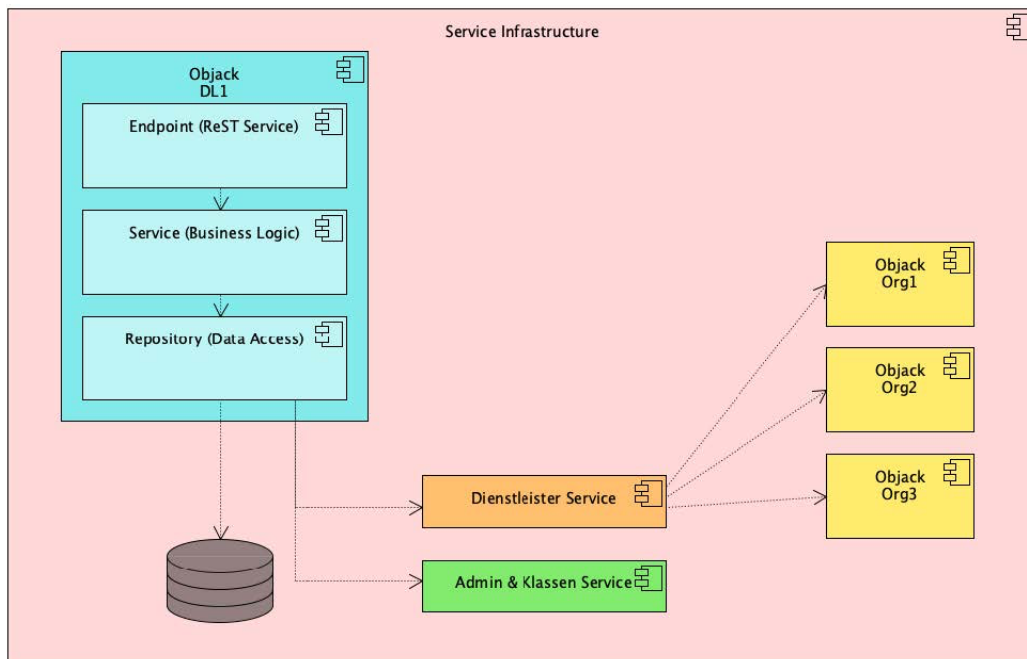
3.7 Einbindung von On-Premises Instanzen

Eine wesentliche Anforderung an die neue Infrastruktur ist die Einbindung von Installationen, die in der Infrastruktur des Kunden betrieben werden. Da eine Öffnung dieser Infrastrukturen oftmals mit großen organisatorischen und technischen Hürden auf Kundenseite verbunden ist, wird die Kommunikation zwischen allen zentralen Services über bidirektionale WebSocket-Verbindungen abgewickelt. Die einzige technische Voraussetzung in der Infrastruktur des Kunden ist, dass die Objack-Instanz über HTTPS mit den zentralen Services kommunizieren darf. Jeder hochgefahrenen Objack-Service öffnet von sich aus die Verbindung zu allen notwendigen zentralen Services, macht sich dort bekannt und ist dann bereit, Daten zu senden und zu empfangen. Die zentralen Services können bei Bedarf Daten der Objack-Services lokal cachen, um die Performance der Datenzugriffe zu erhöhen.



3.8 Kapselung der Kommunikation zwischen Services

Um die Kommunikation zwischen den Services zu kapseln, werden diese Zugriffe auf Ebene der Repositories, also der Datenzugriffsschicht der Anwendung, integriert. Das Ideal ist dabei, dass keine oder nur wenige Änderungen der Service- oder RestController-Klassen erforderlich sind.



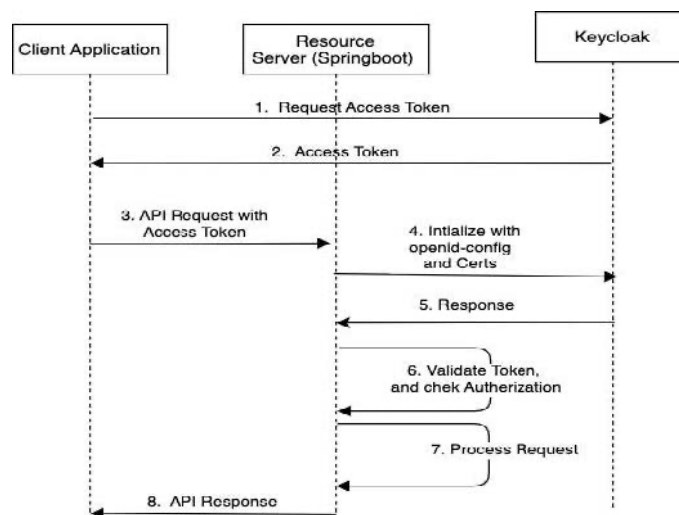
4. Security

4.1 Kommunikation zwischen den Services

Die Kommunikation zwischen dem Zentralen Konfigurationsservice, dem Dienstleister-Service, dem Admin- und Klassen-Service sowie den Objack-Services der einzelnen Organisationen wird durch SSL (HTTPS) gesichert und durch beidseitige Zertifikate authentifiziert. Wenn eine neue Organisation in die Objack-Service-Infrastruktur hinzugefügt wird, wird zuerst das Zertifikat für den Instance-Server generiert und im Konfigurationsservice abgelegt. Anschließend werden der Dienstleister und der Admin-Service über die neue Konfiguration informiert. Nachdem alle Server-Instanzen das Zertifikat der neuen Instanz kennen, kann sich diese gegenüber dem Dienstleister- und Administrationsservice authentifizieren.

4.2 Kommunikation zwischen den Benutzern und den Services

Derzeit ist die Authentifizierung mit Benutzername und Passwort sowohl lokal in der Instanzdatenbank als auch über SAML2 implementiert. Da zukünftig unterschiedliche Authentifizierungsmechanismen wie OAuth, Google oder Microsoft gewünscht sein könnten, wäre ein separater Authentifizierungsdienst wie Keycloak von Vorteil. Mit einem solchen Authentifizierungsservice können die Authentifizierungsmechanismen dynamisch angepasst und geändert werden. Außerdem kann der Zugang zum Service für User oder Instanzen einfach im UI aktiviert/deaktiviert werden oder User können dazu gezwungen werden, sich neu anzumelden.



5. Grobe Aufwände und Zeitplanung

Diese Abschätzung dient lediglich der groben Zeitangabe für die Projektplanung. Da die Anforderungen noch nicht genau definiert sind, kann sie nie akkurat sein.

In der aktuellen Schätzung werden ausschließlich die Implementierung und der Test berücksichtigt. Bevor die einzelnen Module implementiert werden können, muss zunächst eine kurze User Story erstellt werden. Dadurch sollen Missverständnisse bei der Umsetzung vermieden werden. Die Implementierung erfordert daher Vorlaufzeit, in der unter anderem Review-Runden mit Objack stattfinden.

Die UI-Funktionalität des Admin- und Dienstleisterportals wurde noch nicht geschätzt. Die Funktionalität und Implementierung basieren auf der neuen Architektur. Aufgrund der vielen offenen Fragen bezüglich der Aufteilung der Funktionalität können wir im Moment keine belastbare Schätzung diesbezüglich liefern.

	Aufwand in Tagen für 1 MitarbeiterIn	Durchlaufzeit in Tagen wenn 2 Entwickler daran arbeiten
Flexibilisierung der Klassenstruktur	14	9
Teilen von Organisationsklassen und Baums	13	9
Individuelle Objack Instanzen	8	5
Objack Services	10	8
Zentraler Konfigurationsservice	5	5
Admin & Klassenservice	15	9
Dienstleister Service	15	9
Absicherung der Kommunikation / Zertifikate	5	5
Extrahieren der Superadmin-Funktionalität in eine separate Applikation - erste Version von Admin Portal	8	6
Admin Portal	muss genauer spezifiziert werden	
Dienstleister Portal	muss genauer spezifiziert werden	
Umbau der Authentifizierung auf Keycloak	8	8
Summe	101	ca. 15 Wochen