



Projektarbeit: Ski-Service Auftragsverwaltung

Ausgangssituation

Firma Jetstream-Service

Problem: Relationale Datenbank
reicht nicht mehr aus

Lösung: Migration auf NoSQL für
Skalierbarkeit & Kosteneinsparungen

Technologien & Methoden

Datenbank: MongoDB
(NoSQL)

Backend: WebAPI mit C#,
Java oder Python

Testing: Postman

Versionierung: Git

Umsetzung nach IPERKA

Informieren

Planen

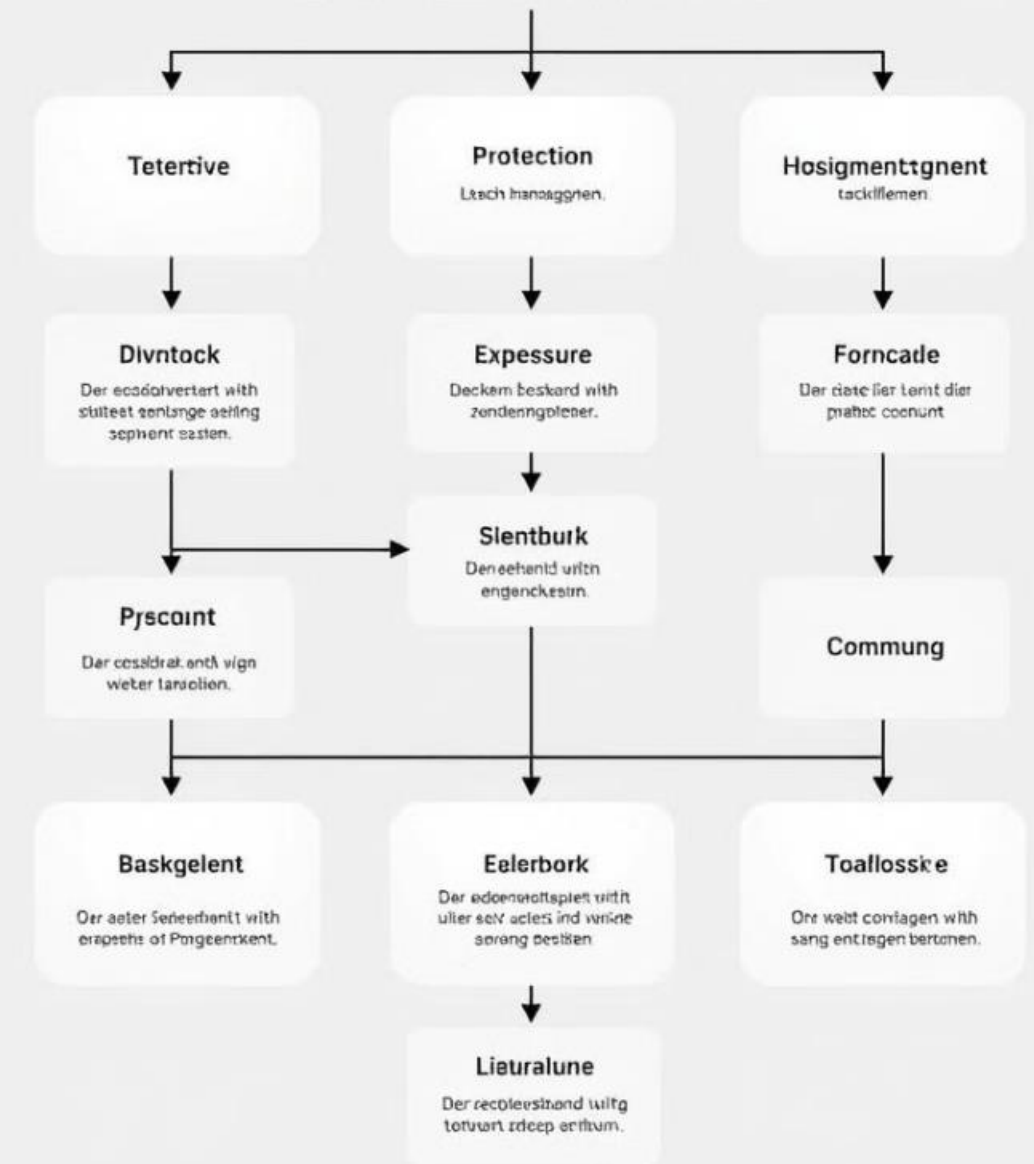
Entscheiden

Realisieren

Kontrollieren

Auswerten

IPERKA



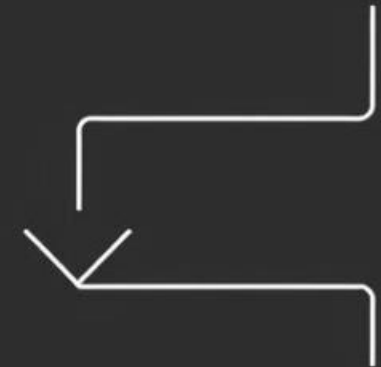
Migration der Datenbank (SQL → NoSQL)

- 1 Schritt 1: MongoDB.Driver package heruntergeladen
- 2 Schritt 2: ConnectionString angepasst
- 3 Schritt 3: MongoDBService.cs zum initialisieren
- 4 Schritt 4: SQL-Statements in MongoDB-Statements geändert

Tractoma		
Table		Lanil
1		en
2		on
4		on
5		on



Matlonn	
Str	Limi
Jr	ste
Plr	Unif en.jul
Sin	std
En.	ste



Metition	
E.	Jef kel phote
B	DOMllcgcall
B	Jef kel phores
JSOM	/hejeet
F	RayPleccpall
C	Sc-Pleccpart
D	Jef ket prieres

NoSQL.

Mongo DB

```
7  
7  
3  
4  
5 /Permals( top cate.);  
0 fire sales top  
0  
7 /fection{ cast:  
8   ails: top.autphiut.iol.dr.d);  
0   lise: Podfin CatalctnaB:  
8   lils: top.the spart for mal.dsl_pital);  
0 }  
6 feserguttl(oulen({ests/t/-d);  
0   erind = 1dj;  
0   erind (nclen({stlet,  
0 fepl { - {  
5   altnal(monogo(tsls/st ttop );  
5 }  
0 }
```

WebAPI-Integration mit NoSQL

Neue API-Funktionen

Login

Serviceaufträge abrufen

Serviceaufträge anlegen, mutieren, löschen

Anforderungen & Umsetzung

Migration vollständig umgesetzt?



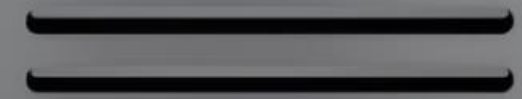
Benutzerkonzept implementiert?



Backup-Strategie?



Testprojekt mit Postman?



Live-Demo – API & Datenbank in Aktion



Login als Admin & User



Neuen Auftrag anlegen



Status ändern (Offen → In
Arbeit → Abgeschlossen)



Auftrag löschen

Herausforderungen & Lessons Learned

- Migration von SQL zu NoSQL
- Datenkonsistenz & Validierung
- Performance-Optimierung





Fazit

Projekt erfolgreich umgesetzt!

System ist jetzt skalierbar & effizient. Fragen?