



KARSTEN SILZ

9. NOVEMBER 2021



BETTER PROJECTS  
**FASTER**

[BPF.LI/S](#)

**WIE DATENVERSIONEN IN  
FRONT-ENDS VON JAVA-  
ANWENDUNGEN ZEIGEN?**



**FOLIEN,  
CODE &  
MEHR**

**[BPF.LI/SPA](https://bpf.li/spa)**

**SIE: JAVA-ENTWICKLER  
FÜR KUNDEN-VERWALTUNG**

**DAS SCHLIMMSTE, DASS  
IHNEN PASSIEREN KANN?**



**UMFRAGE**

**JAVASCRIPT LERNEN?**

**MIT NUTZERN REDEN?**

**MIT MARKETING REDEN?!**

**DAS SCHLIMMSTE?**

**NEIN!**



**ALLE DATEN WEG!**

**LÖSUNG: BACK-UP**



**LÖSUNG: BACK-UP**

**ALLES GUT!**

**NEIN!**

**JEDEN TAG:**  
**EIN PAAR DATEN WEG!**

**VERTIPPEN**

**LÖSCHEN**

**FALSCH BEDIENEN**

**NUTZER MACHEN FEHLER**

**ERRARUM HUMANUM EST**

**IRREN IST MENSCHLICH**

**ERRARUM HUMANUM EST**

EHE

VIELE NUTZER, **VIELE FEHLER**

DB-BACKUP 👎

NUTZER 😞 => ENTWICKLER

**NUTZER MACHEN FEHLER!**

# 2 BEISPIELE

**KUNDE: KEINE EMAIL MIT UNTERLAGEN**

**2X ANGERUFEN**

**JETZT OK**

**#1: FEHLENDE EMAIL**

# Personal

First name

Jon

Last name

Snow

Email

[jon.snow@nights-watch.com](mailto:jon.snow@nights-watch.com)







**ZEITMASCHINE!**



**Version 2 by Karsten**

**Sat, 26 September 2020, 14:38**



**Version 1 by Frank**

**Thu, 17 September 2020, 22:14**

2 changes, marked in red.

**First name**

Jon

**Last name**

Snow

**Email**

[jon.snow@nights-watch.com](mailto:jon.snow@nights-watch.com)

**First name**

John

**Last name**

Snow

**Email**

[john.snow@nights-watch.com](mailto:john.snow@nights-watch.com)



**FALSCHER EMAIL!**

1. **FALSCH** EMAIL ERLAUBT

2. **KEIN** EMAIL-VERSAND NACH KORREKTUR

**#1: URSACHEN**



**#1: FEHLENDE EMAIL**

300 KUNDEN: KEINE RECHNUNGEN

#2: KEINE RECHNUNGEN

# Personal

First name

MR Jon

Last name

Snow

Email

[jon.snow@nights-watch.com](mailto:jon.snow@nights-watch.com)







**ZEITMASCHINE!**

↔ Compare

with

Version 2 (previous) ▾

↺ Clear comparison



**Version 3 by Karl**  
Thu, 4 November 2021, 10:01



**Version 2 by Karsten**  
Sat, 26 September 2020, 14:38

1 change, marked in red.

First name

MR Jon

First name

Jon



**NAME GEÄNDERT!**

**MARKETING: PROMO FÜR VIELFLIEGER**

**NAMEN ÜBERSCHRIEBEN**

**#2: URSACHEN**

MANUELL NAMEN KORRIGIEREN

KORREKTE NAMEN IN ZEITMASCHINE

#2: LÖSUNG



**#2: KEINE RECHNUNGEN**

**ZEITMASCHINE: NUTZER  
LÖSEN BEIDE PROBLEME -  
OHNE UNS ENTWICKLER!**

**ZEITMASCHINE KOSTENLOS:  
EINBAUEN?**



**UMFRAGE**

**MEHRHEIT: JA?**

**ZEITMASCHINE VERDOPPELT**

**ENTWICKLUNGSZEIT:**

**EINBAUEN?**



**UMFRAGE**

**MEHRHEIT: NEIN?**

**ZEITMASCHINE:**

**KOSTENLOS JA,**

**TEUER NICHT**

**AUFGABE: PREISWERTE  
ZEITMASCHINE FÜR JAVA!**

**ANFORDERUNGEN FÜR ZEITMASCHINE**

**KANDIDATEN FÜR ZEITMASCHINE**

**KANDIDATEN-BEWERTUNG**

**ERFAHRUNGEN MIT ZEITMASCHINE**

**WARUM MIR ZUHÖREN –  
UND GLAUBEN?**



**22 JAHRE JAVA**

**JAVA-REPORTER @ INFOQ**

**6 KRITERIEN**

**VERKAUFE NIX**

**NUTZE JAVERS, SPENDE \$10/MONAT**

**SIE ENTSCHEIDEN!**

**ANFORDERUNGEN FÜR ZEITMASCHINE**

**KANDIDATEN FÜR ZEITMASCHINE**

**KANDIDATEN-BEWERTUNG**

**ERFAHRUNGEN MIT ZEITMASCHINE**

**ANFORDERUNGEN FÜR  
ZEITMASCHINE**

**ERZEUGE VERSION**

**ZEIGE ALLE VERSIONEN**

**FRAGE ALLE VERSIONEN AB**

**ZEIGE EINE VERSION**

**FRAGE EINE VERSION AB**

**ZEIGE UNTERSCHIEDE  
ZWEIER VERSIONEN**

**VERGLEICHE ZWEI VERSIONEN**

**FRONT-END**

**BACK-END**

**WARUM FRONT-END?**

**WENN WIR ENTWICKLER**

**KEIN FRONT-END**

**BAUEN, DANN SIND**

**WIR DAS FRONT-END!**

**FRONT-END: DAMIT WIR  
NICHT DAUERND IN DATEN  
WÜHLEN MÜSSEN!**

**ARCHITEKTUR: MICROSERVICE & MONOLITH**

**DB: RELATIONAL & NOSQL**

**MINIMAL-INVASIV**

**PREISWERT**

**MEHR ANFORDERUNGEN**

~~ANFORDERUNGEN FÜR ZEITMASCHINE~~

KANDIDATEN FÜR ZEITMASCHINE

KANDIDATEN-BEWERTUNG

ERFAHRUNGEN MIT ZEITMASCHINE

**KANDIDATEN FÜR  
ZEITMASCHINE**

**EIGEN-ENTWICKLUNG**

**HIBERNATE ENVERS**

**JAVERS**

**XTDB & DATOMIC**

**KANDIDATEN**

**SPEICHERN: JAVA SERIALIZATION**

**1 TABELLE**

**ANNOTATIONEN FÜR KLASSEN-ATTRIBUTE**

**EIGEN-ENTWICKLUNG**

WIE OBJEKTE VERGLEICHEN?

NUTZUNG IN ANDEREN PROJEKTEN?

EIGEN-ENTWICKLUNG?

**EIGEN-ENTWICKLUNG**

**HIBERNATE ENVERS**

**JAVERS**

**XTDB & DATOMIC**

**KANDIDATEN**

**HIBERNATE-ERWEITERUNG**

**1 GLOBALE TABELLE + 1 TABELLE PRO ENTITY**

**NEUE "AUDIT ENTITIES"**

**@AUDITED FÜR KLASSEN-ATTRIBUTE**

**HIBERNATE ENVERS**

WIE OBJEKTE VERGLEICHEN?

WIE RELATIONAL OHNE HIBERNATE?

WIE NOSQL?

HIBERNATE ENVERS?

**EIGEN-ENTWICKLUNG**

**HIBERNATE ENVERS**

**JAVERS**

**XTDB & DATOMIC**

**KANDIDATEN**

# "OBJECT **AUDIT** AND **DIFF** IN JAVA"

1.0: DEZEMBER 2014

JSON

4 TABELLEN

# JAVERS

**ANNOTATIONEN FÜR KLASSEN-ATTRIBUTE**

**H2, POSTGRESQL, MYSQL, MARIADB, ORACLE,  
SQL SERVER, MONGODB**

**SPRING BOOT STARTER**

**JAVERS**

**NUR 1 MAIN COMMITTER**

**WIE ANDERE DB?**

**JAVERS?**

**EIGEN-ENTWICKLUNG**

**HIBERNATE ENVERS**

**JAVERS**

**XTDB & DATOMIC**

**KANDIDATEN**

**"IMMUTABLE RECORDS"**

**SEPARATE NOSQL DB (XTDB: KAFKA)**

**EIGENE ABFRAGE-SPRACHEN UND APIS**

**XTDB & DATOMIC**

**DATOMIC: ON-PREMISES (UPDATES NUR 1 JAHR) & CLOUD**

**XTDB: APACHE-KAFKA-AUFSATZ**

**XTDB & DATOMIC**

**EIGEN-ENTWICKLUNG**

**HIBERNATE ENVERS**

**JAVERS**

**XTDB & DATOMIC**

**KANDIDATEN**

~~ANFORDERUNGEN FÜR ZEITMASCHINE~~

~~KANDIDATEN FÜR ZEITMASCHINE~~

KANDIDATEN-BEWERTUNG

ERFAHRUNGEN MIT ZEITMASCHINE

# KANDIDATEN-BEWERTUNG

**AUFWAND: INITIAL**

**DB-ÄNDERUNGEN**

**AUFWAND: WARTUNG**

**ÄNDERUNGS-SICHERHEIT**

**DB-SPEICHERPLATZ**

**UNIVERSALITÄT**

**KRITERIEN**

1 STERN: SCHLECHT



2 STERNE: AUSREICHEND



3 STERNE: GUT



BEWERTUNG

**AUFWAND: INITIAL**



**DB-ÄNDERUNGEN**



**AUFWAND: WARTUNG**



**ÄNDERUNGS-SICHERHEIT**



**DB-SPEICHERPLATZ**



**UNIVERSALITÄT**



**EIGEN-ENTWICKLUNG**

EIGEN-ENTWICKLUNG 12 ★

HIBERNATE ENVERS

JAVERS

XTDB & DATOMIC

BEWERTUNG

**AUFWAND: INITIAL**



**AUFWAND: WARTUNG**



**DB-SPEICHERPLATZ**



**DB-ÄNDERUNGEN**



**ÄNDERUNGS-SICHERHEIT**



**UNIVERSALITÄT**



**HIBERNATE ENVERS**

EIGEN-ENTWICKLUNG 12 ★

HIBERNATE ENVERS 12 ★

JAVERS

XTDB & DATOMIC

BEWERTUNG

**AUFWAND: INITIAL**



**AUFWAND: WARTUNG**



**DB-SPEICHERPLATZ**



**DB-ÄNDERUNGEN**



**ÄNDERUNGS-SICHERHEIT**



**UNIVERSALITÄT**



**JAVERS**

EIGEN-ENTWICKLUNG 12 ★

HIBERNATE ENVERS 12 ★

JAVERS 13 ★

XTDB & DATOMIC

BEWERTUNG

**AUFWAND: INITIAL**



**AUFWAND: WARTUNG**



**DB-SPEICHERPLATZ**



**DB-ÄNDERUNGEN**



**ÄNDERUNGS-SICHERHEIT**



**UNIVERSALITÄT**



**XTDB & DATOMIC**

EIGEN-ENTWICKLUNG 12 ★

HIBERNATE ENVERS 12 ★

JAVERS 13 ★

XTDB & DATOMIC 13 ★

# BEWERTUNG

**ÄNDERUNGS-SICHERHEIT: XTDB ODER DATOMIC**

**DB NICHT IN JAVERS: JAVERS + MYSQL/POSTGRES/  
MONGODB... SEPARAT (MICROSERVICE?)**

**SONST: JAVERS**

**EMPFEHLUNG**

~~ANFORDERUNGEN FÜR ZEITMASCHINE~~

~~KANDIDATEN FÜR ZEITMASCHINE~~

~~KANDIDATEN BEWERTUNG~~

ERFAHRUNGEN MIT ZEITMASCHINE

# ERFAHRUNGEN MIT ZEITMASCHINE

"YOUR HOME IN GOOD HANDS"

SAAS FÜR PROFESSIONELLE **CAT-SITTERS** IN UK

SEIT **MAI 2021** IN CLOSED BETA

**MEIN** PROJEKT

**CAT-  
SITTER  
FRONT-  
END**

**FLUTTER**

**IOS & ANDROID**

**REST JSON**



**SPRING BOOT**



**DOCKER  
COMPOSE**

**REST JSON**



**MANAGER  
FRONT-  
END**

**ANGULAR**

**BROWSER**

**BACK-END**

**JAVERS** FÜR KUNDEN- & TEAM-DATEN

**ICH:** GANZER CODE + BETRIEB

**MEIN** PROJEKT



**HAT JAVERS FUNKTIONIERT?**



**JAVERS NOCH MAL NEHMEN?**



**JAVERS EMPFEEHLEN?**

**ERZEUGE VERSION**

**ZEIGE ALLE VERSIONEN**

**FRAGE ALLE VERSIONEN AB**

**ZEIGE EINE VERSION**

**FRAGE EINE VERSION AB**

**ZEIGE UNTERSCHIEDE  
ZWEIER VERSIONEN**

**VERGLEICHE ZWEI VERSIONEN**

**FRONT-END**

**BACK-END**

ZEIGE **ALLE** VERSIONEN

ZEIGE **EINE** VERSION

ZEIGE UNTERSCHIEDE  
**ZWEIER** VERSIONEN

**FRONT**-END

ERZEUGE VERSION

FRAGE **ALLE** VERSIONEN AB

FRAGE **EINE** VERSION AB

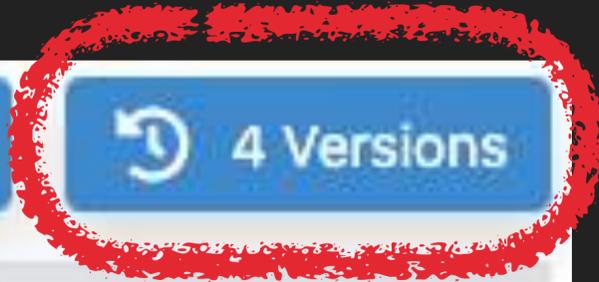
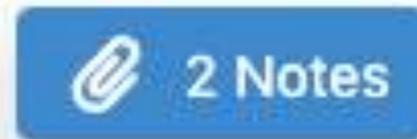
VERGLEICHE **ZWEI** VERSIONEN

**BACK**-END

**GENERISCHE UI**

**ZEIGE ALLE VERSIONEN**

# Personal



First name

Jon

Last name

Snow

Email

[jon.snow@nights-watch.com](mailto:jon.snow@nights-watch.com)

# BROWSER

# IOS & ANDROID

**Jon Snow**

Birthday 2014

Gender Male

**CUSTOMER: BUSINESS**

Data 0 Notes 1 Version

Customer of KA JCL

Visit frequency Once a day

Got magnet No

**CUSTOMER: PERSONAL**

Data 2 Notes 4 Versions

First name Jon

Last name Snow

Email jon.snow@nights-watch.com

Tasks Team My Data

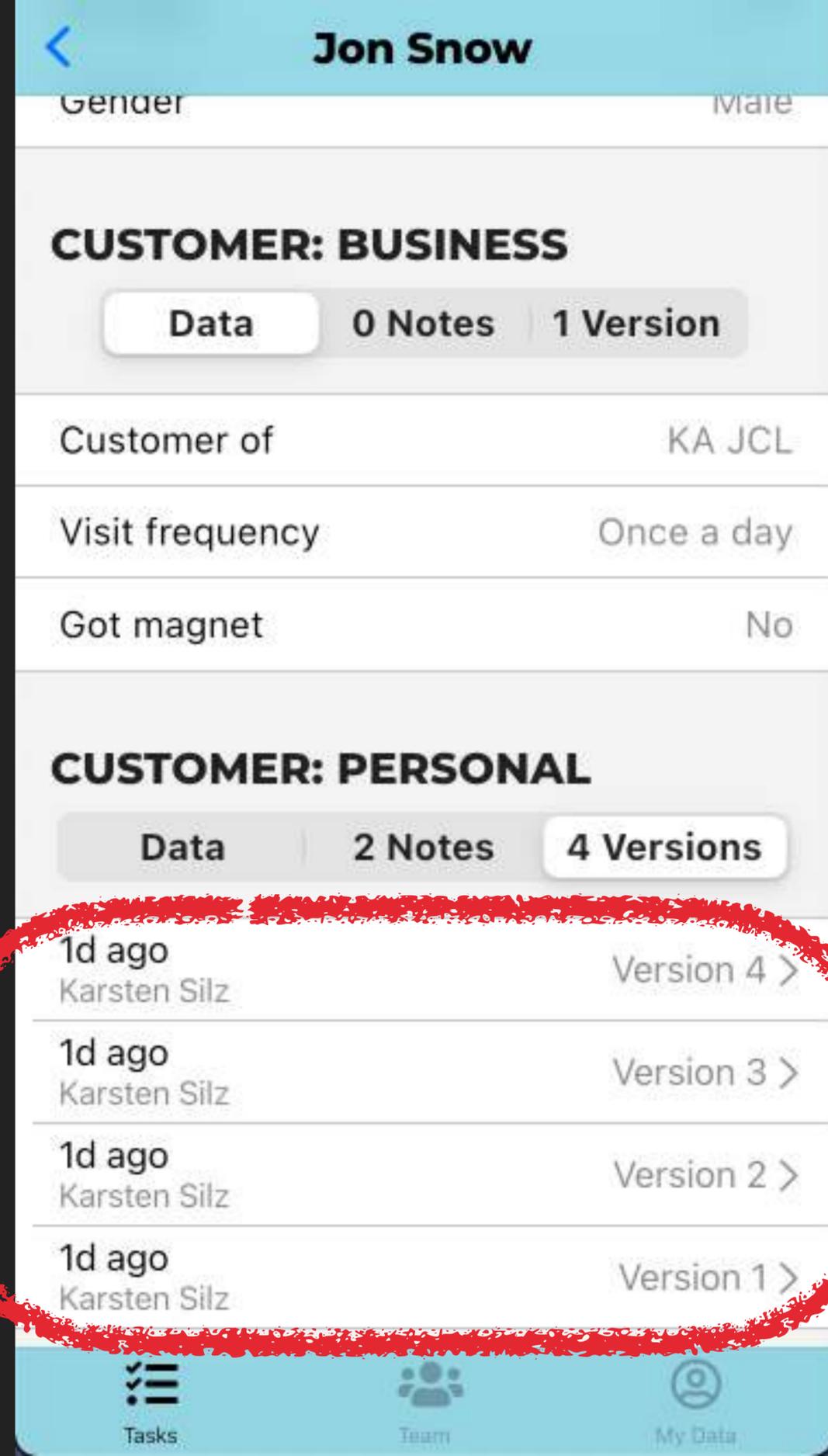
# Versions

Total number of versions: 4

Timestamp 	Relative time 	Version 	Author 
<a href="#">Thu, 4 November 2021, 10:27 PM</a>	A day ago	4	 Karsten
<a href="#">Thu, 4 November 2021, 10:01 PM</a>	A day ago	3	 Karsten
<a href="#">Thu, 4 November 2021, 10:01 PM</a>	A day ago	2	 Karsten
<a href="#">Thu, 4 November 2021, 10:00 PM</a>	A day ago	1	 Karsten

# BROWSER

# IOS & ANDROID



**SPEZIFISCHE UI IN GENERISCHER KOMPONENTE**

**ZEIGE EINE VERSION**



# Version Details

← Back

↔ Compare

with



Clear comparison



## Version 2 by Karsten

A day ago (Thu, 4 November 2021, 10:01 PM)

First name

Jon

Last name

Snow

Email

[jon.snow@nights-watch.com](mailto:jon.snow@nights-watch.com)

**GENERISCH**

**BROWSER**



# Version Details

← Back

↔ Compare

with



Clear comparison



## Version 2 by Karsten

A day ago (Thu, 4 November 2021, 10:01 PM)

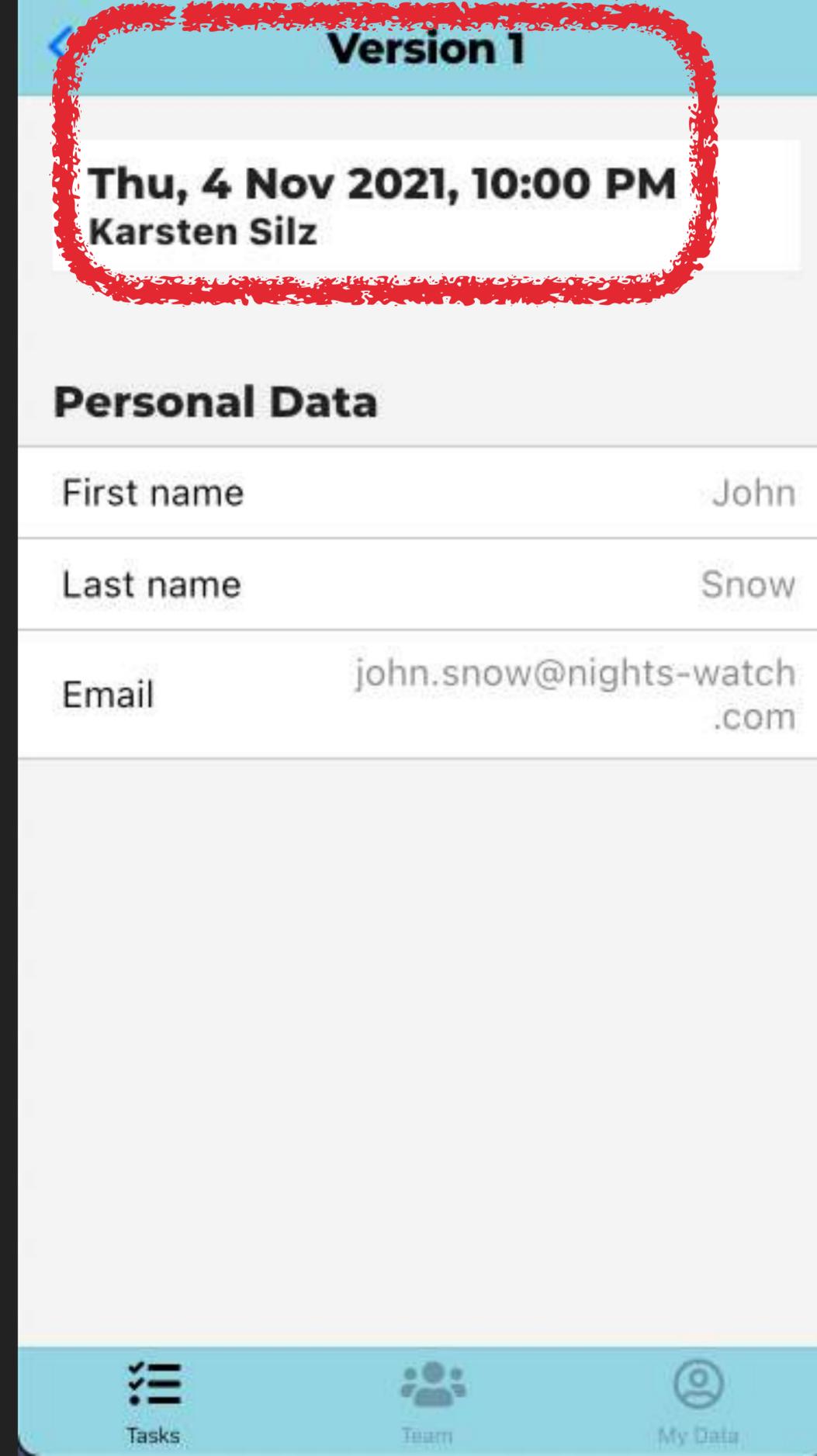
First name	Jon
Last name	Snow
Email	<a href="mailto:jon.snow@nights-watch.com">jon.snow@nights-watch.com</a>

**SPEZIFISCH**

**BROWSER**

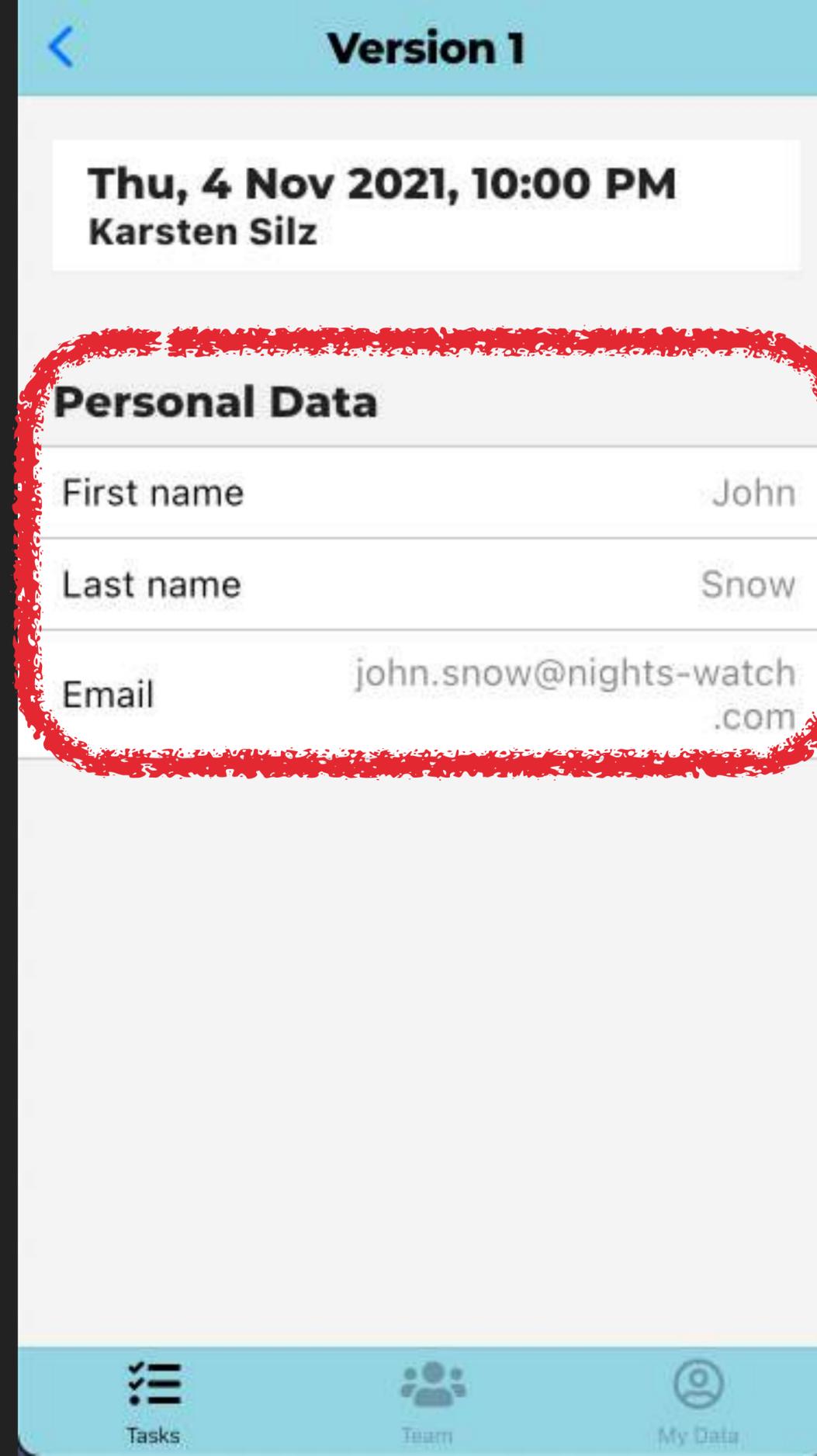
GENERISCH

IOS &  
ANDROID



# IOS & ANDROID

SPEZIFISCH



**SPEZIFISCHE UI IN GENERISCHER KOMPONENTE**

**SPEZIFISCHE UI: ZEIGT UNTERSCHIEDE**

**ZEIGE UNTERSCHIEDE  
ZWEIER VERSIONEN**

# Version Details

← Back

↔ Compare

with

Version 1 (previous) ▾

↺ Clear comparison



## Version 2 by Karsten

A day ago (Thu, 4 November 2021, 10:01 PM)



## Version 1 by Karsten

A day ago (Thu, 4 November 2021, 10:00 PM)

2 changes, marked in red.

First name

Jon

**GENERISCH**

First name

John

Last name

Snow

Last name

Snow

Email

[jon.snow@nights-watch.com](mailto:jon.snow@nights-watch.com)

Email

[john.snow@nights-watch.com](mailto:john.snow@nights-watch.com)

# BROWSER

# Version Details

← Back

↔ Compare

with

Version 1 (previous) ▾

↺ Clear comparison



## Version 2 by Karsten

A day ago (Thu, 4 November 2021, 10:01 PM)



## Version 1 by Karsten

A day ago (Thu, 4 November 2021, 10:00 PM)

2 changes, marked in red.

**SPEZIFISCH**

First name

Jon

First name

John

Last name

Snow

Last name

Snow

Email

[jon.snow@nights-watch.com](mailto:jon.snow@nights-watch.com)

Email

[john.snow@nights-watch.com](mailto:john.snow@nights-watch.com)

**BROWSER**

~~ZEIGE ALLE VERSIONEN~~

~~ZEIGE EINE VERSION~~

~~ZEIGE UNTERSCHIEDE~~

~~ZWEIER VERSIONEN~~

FRONT-END

ERZEUGE VERSION

FRAGE ALLE VERSIONEN AB

FRAGE EINE VERSION AB

VERGLEICHE ZWEI VERSIONEN

BACK-END

**ERZEUGE VERSION**

**FRAGE ALLE VERSIONEN AB**

**FRAGE EINE VERSION AB**

**VERGLEICHE ZWEI VERSIONEN**

**BACK-END**

~~ZEIGE ALLE VERSIONEN~~

~~ZEIGE EINE VERSION~~

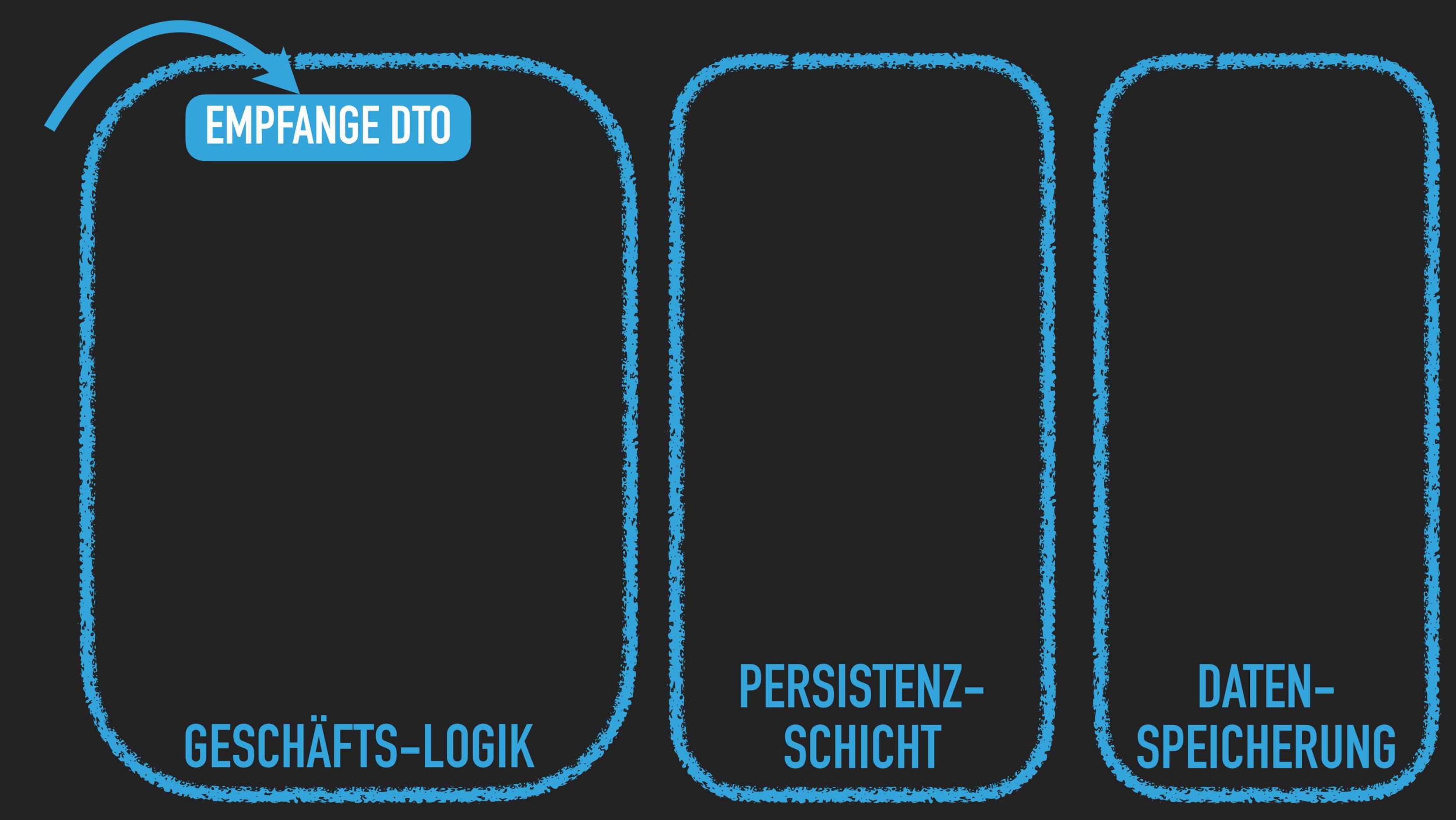
~~ZEIGE UNTERSCHIEDE~~

~~ZWEIER VERSIONEN~~

**FRONT-END**

WIE PASST DAS IN DIE ARCHITEKTUR?

ERZEUGE VERSION



The diagram illustrates a three-layer architecture. It consists of three vertical, rounded rectangular boxes arranged horizontally. The leftmost box is labeled 'GESCHÄFTS-LOGIK' at the bottom. The middle box is labeled 'PERSISTENZ-SCHICHT' at the bottom. The rightmost box is labeled 'DATEN-SPEICHERUNG' at the bottom. A blue arrow originates from the top left, curves over the top of the first box, and points to a blue rounded rectangle labeled 'EMPFANGE DTO' which is positioned at the top of the first box.

**EMPFANGE DTO**

**GESCHÄFTS-LOGIK**

**PERSISTENZ-  
SCHICHT**

**DATEN-  
SPEICHERUNG**

EMPFANGE DTO

VALIDIERE DTO

GESCHÄFTS-LOGIK

PERSISTENZ-  
SCHICHT

DATEN-  
SPEICHERUNG

EMPFANGE DTO

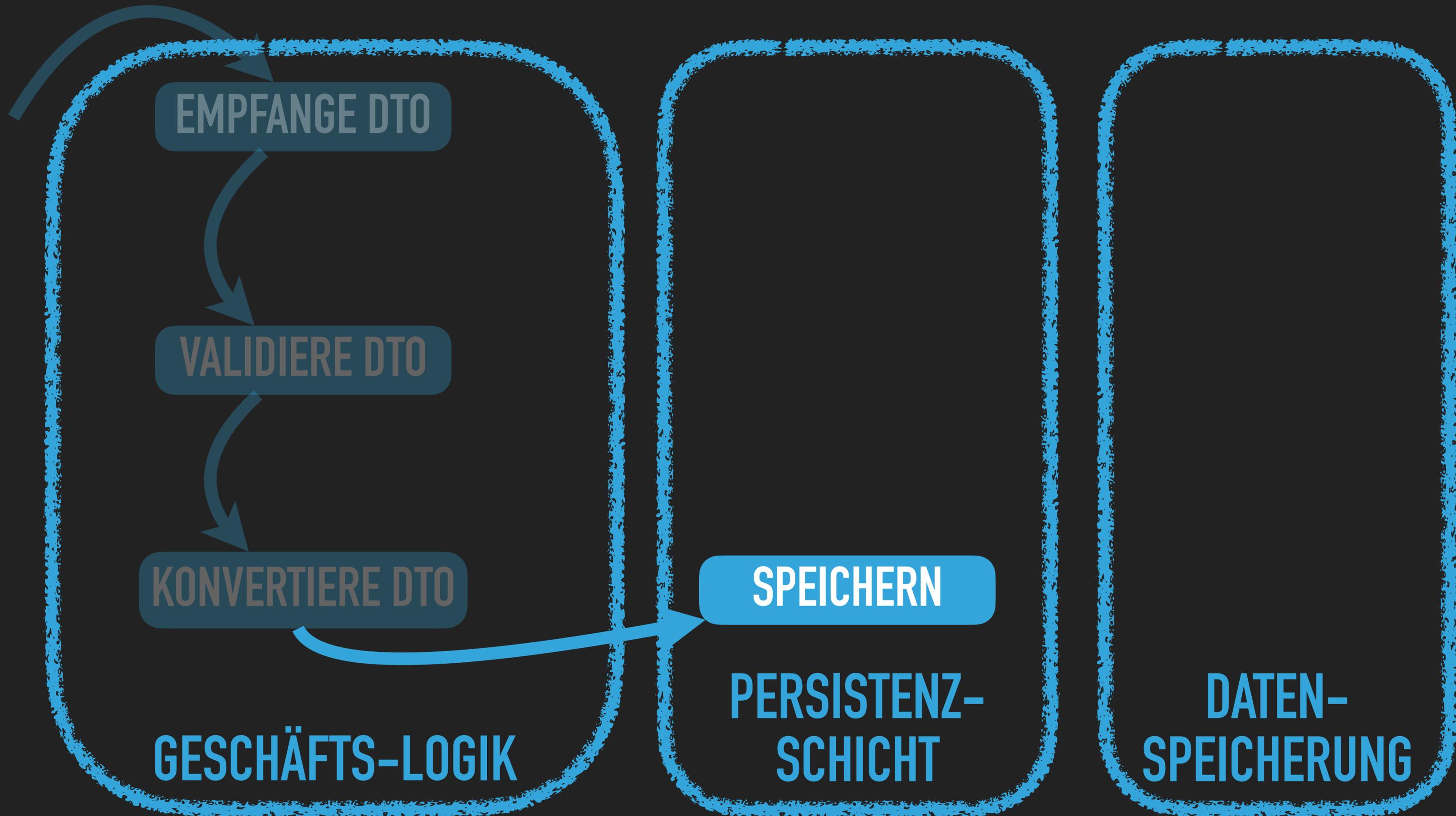
VALIDIERE DTO

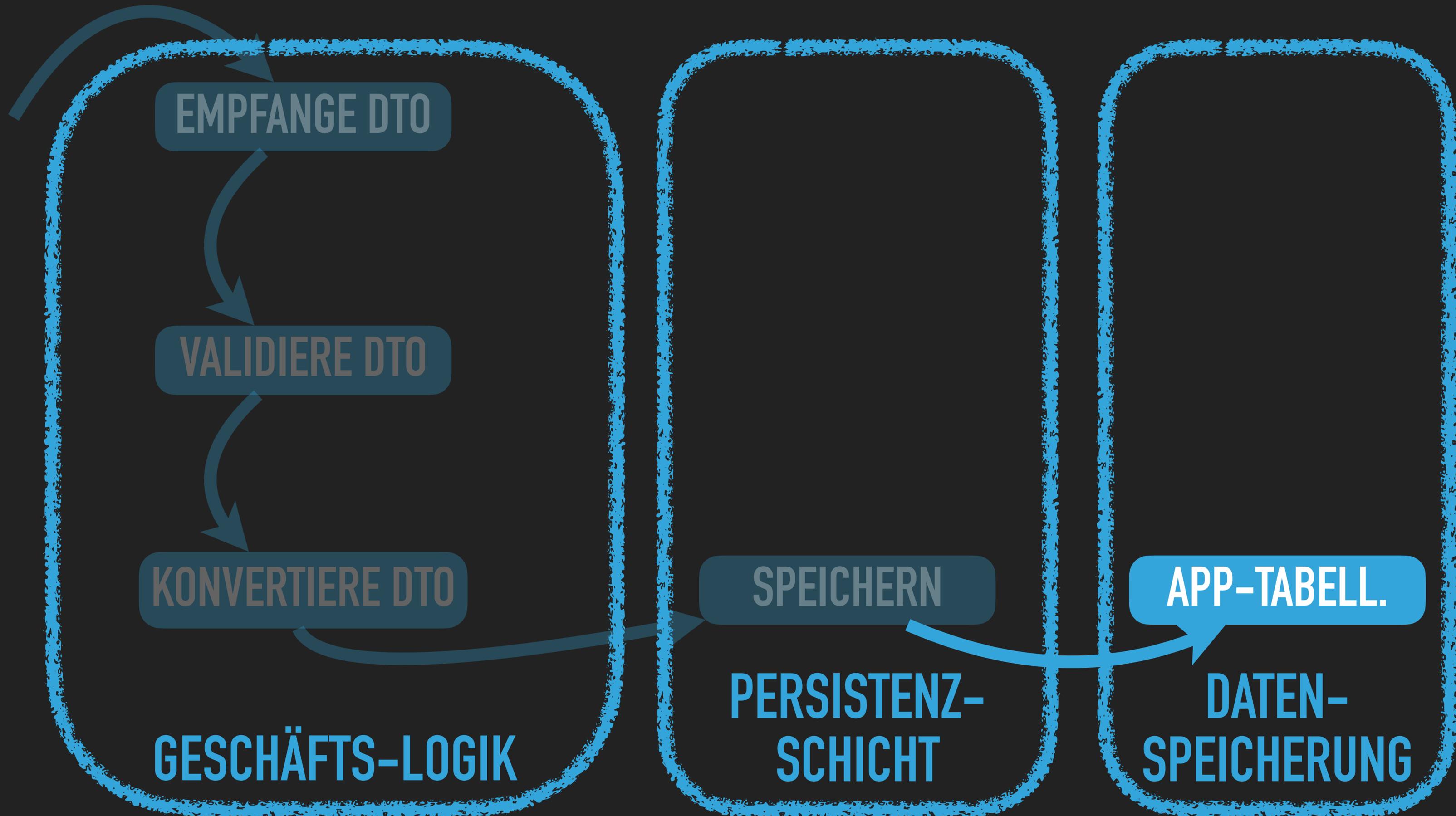
KONVERTIERE DTO

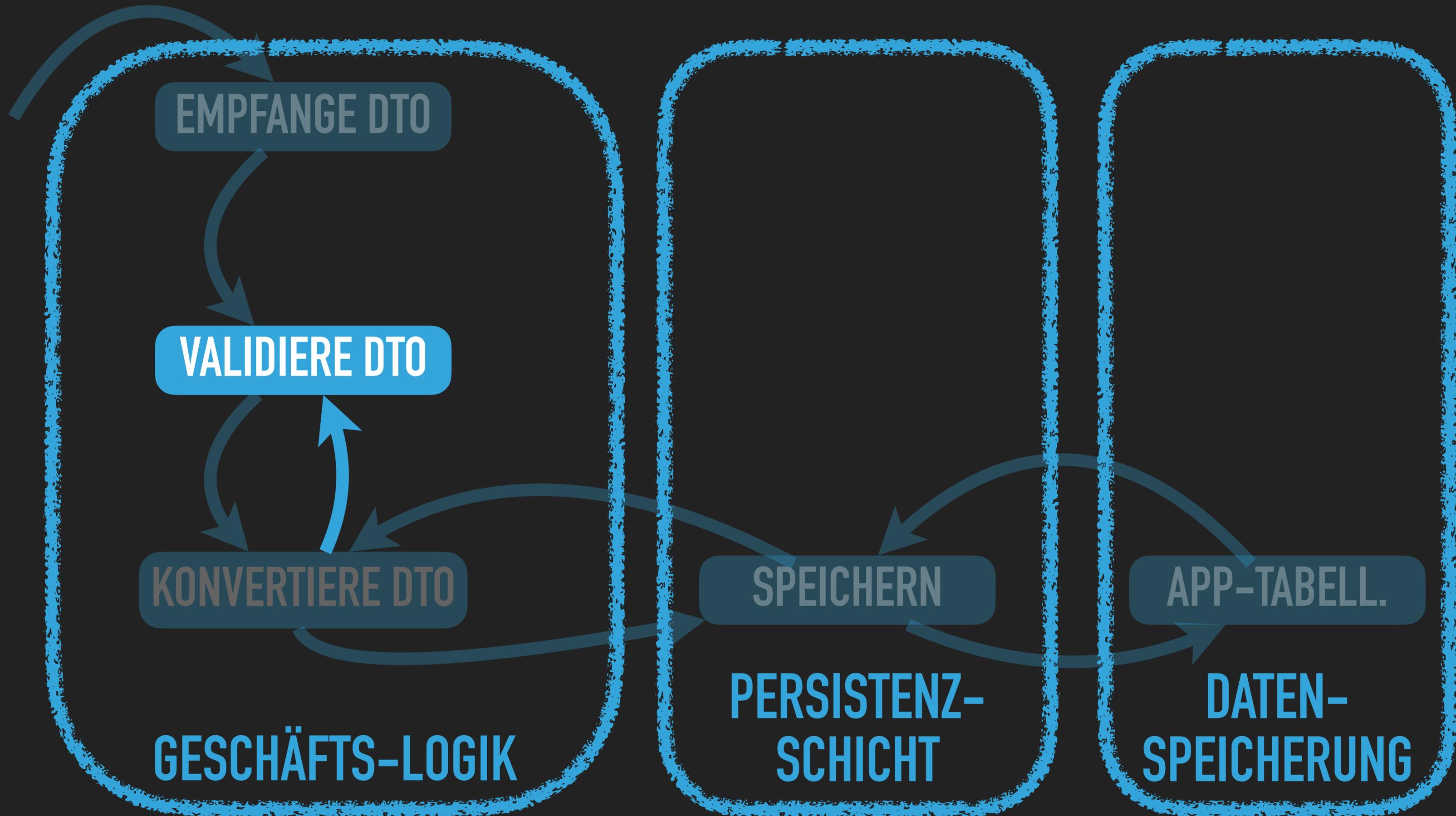
GESCHÄFTS-LOGIK

PERSISTENZ-  
SCHICHT

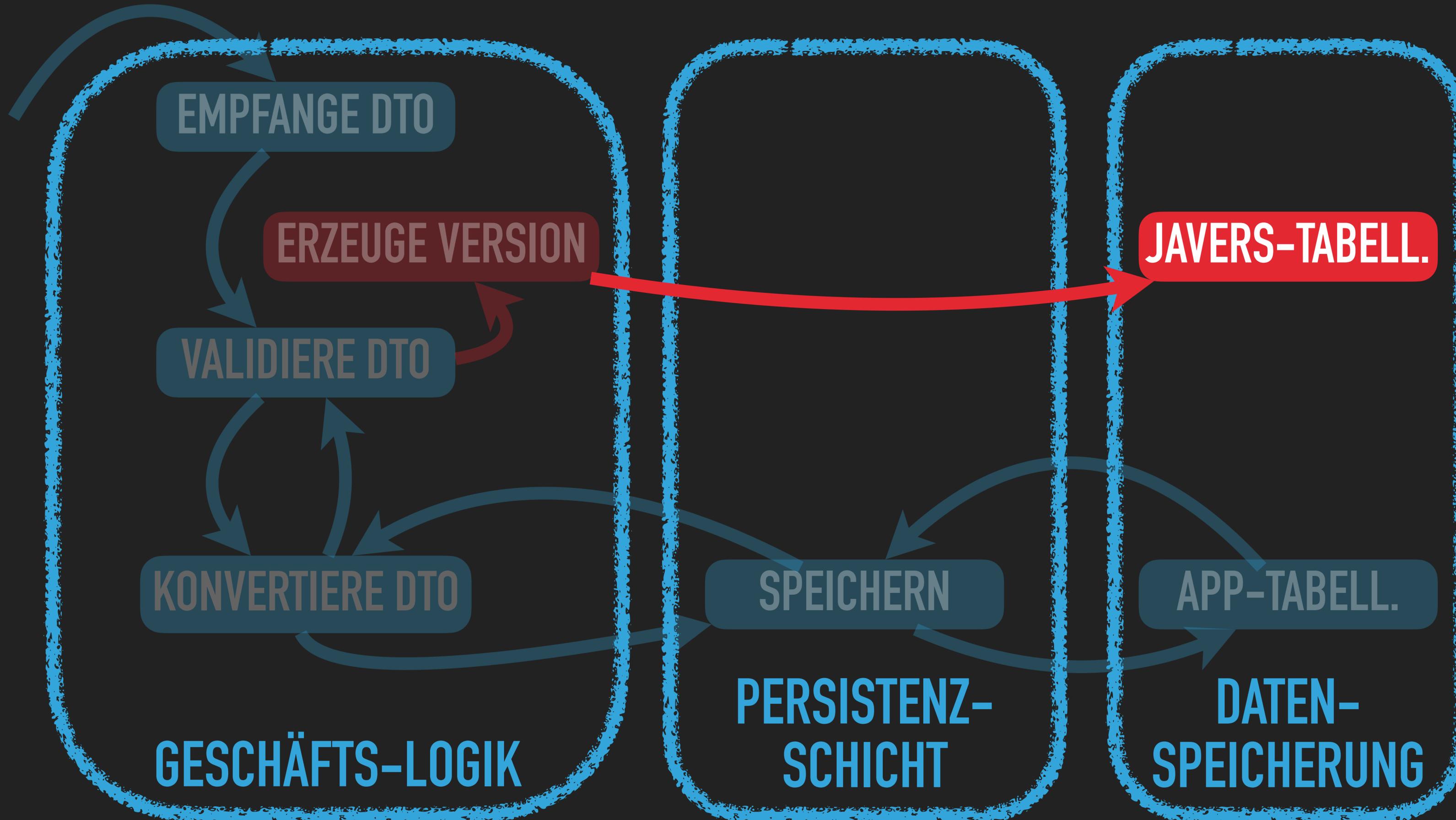
DATEN-  
SPEICHERUNG

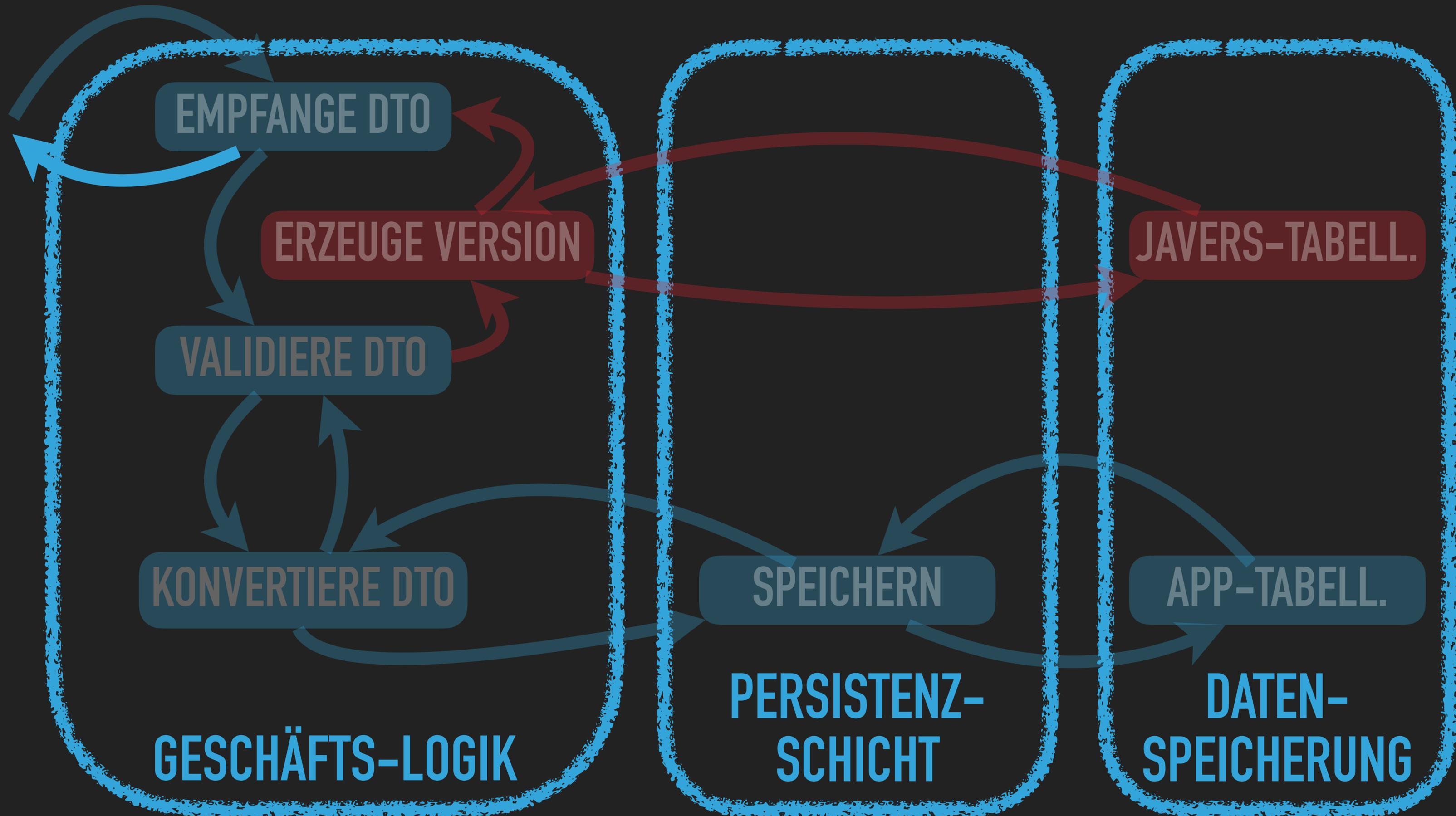












**AUßERHALB** DER APPLIKATIONS-PERSISTENZ

VERSIONIERUNG **EXISTIERENDER** FRONT-END-DTO

**ERZEUGE** VERSION

**KEINE** ÄNDERUNG AN PERSISTENZ

**KEINE** NEUE ENTITY/DTO FÜR VERSIONIERUNG

**KEINE** NEUE DTO FÜR FRONT-END

**ERZEUGE VERSION: GUT**

WAS, WENN VERSIONIERUNG SCHEITERT?

VERSION FEHLT!

IN TRANSAKTION VON PERSISTENZ?

ERZEUGE VERSION: SCHLECHT

**WAS** LANDET IN DER DB?

SNAPSHOT SHOT _PK	TYPE	STATE	CHANGED_PROPERTIES	MANAGED_TYPE
-------------------------	------	-------	--------------------	--------------

100

**GLOBAL ID**

**JV\_SNAPSHOT**

SNAPSHOT SHOT _PK	TYPE	STATE	CHANGED_PROPER TIES	MANAGED_TY PE
100	INITIAL			

**"UPDATE"** BEI GEÄNDERTER ENTITY

**JV\_**  
**SNAPSHOT**

SNAPSHOT_SHOT_PK	TYPE	STATE	CHANGED_PROPERTIES	MANAGED_TYPE
100	INITIAL	{ "firstName": "Karsten", "lastName": "Silz", "id": 50 }		

**ENTITY IN JSON**

# JV\_SNAPSHOT

SNAPSHOT_SHOT_PK	TYPE	STATE	CHANGED_PROPERTIES	MANAGED_TYPE
100	INITIAL	{ "firstName": "Karsten", "lastName": "Silz", "id": 50 }	changedProperties=[ "firstName", "lastName", "id" ]	

UNTERSCHIEDE DER ENTITY

JV\_SNAPSHOT

SNAPSHOT_PK	TYPE	STATE	CHANGED_PROPERTIES	MANAGED_TYPE
100	INITIAL	{ "firstName": "Karsten", "lastName": "Silz", "id": 50 }	changedProperties=[ "firstName", "lastName", "id" ]	com.company.product.dto.Customer

**KLASSEN-NAME DER ENTITY**

**JV\_SNAPSHOT**

SNAPSHOT_PK	TYPE	STATE	CHANGED_PROPERTIES	MANAGED_TYPE
100	INITIAL	{ "firstName": "Karsten", "lastName": "Silz", "id": 50 }	changedProperties=[ "firstName", "lastName", "id" ]	com.company.product.dto.Customer

**ÜBERRASCHUNGSARM!**

**JV\_SNAPSHOT**

~~ERZEUGE VERSION~~

~~ZEIGE ALLE VERSIONEN~~

FRAGE **ALLE** VERSIONEN AB

~~ZEIGE EINE VERSION~~

FRAGE **EINE** VERSION AB

~~ZEIGE UNTERSCHIEDE~~

VERGLEICHE **ZWEI** VERSIONEN

~~ZWEIER VERSIONEN~~

**FRONT-~~END~~**

**BACK-~~END~~**

AUSFÜHRLICH, ENGLISCH, SELBST-STUDIUM:  
VORTRAGS-**WEBSEITE** [BPF.LI/SPA](https://bpf.li/spa)

**ODER**

KURZ, DEUTSCH, GLEICH: **VORTRAGS-ENDE**

**CODE-BEISPIELE**

~~ERZEUGE VERSION~~

~~ZEIGE ALLE VERSIONEN~~

~~FRAGE ALLE VERSIONEN AB~~

~~ZEIGE EINE VERSION~~

~~FRAGE EINE VERSION AB~~

~~ZEIGE UNTERSCHIEDE  
ZWEIER VERSIONEN~~

~~VERGLEICHE ZWEI VERSIONEN~~

FRONT-END

BACK-END

~~ERZEUGE VERSION~~

~~ZEIGE ALLE VERSIONEN~~

~~FRAGE ALLE VERSIONEN AB~~

~~ZEIGE EINE VERSION~~

~~FRAGE EINE VERSION AB~~

~~ZEIGE UNTERSCHIEDE  
ZWEIER VERSIONEN~~

~~VERGLEICHE ZWEI VERSIONEN~~

FRONT-END

BACK-END

**FAZIT: JAVERS**

**FUNKTIONIERT**

**LEICHT ZU ENTZIFFERN**

**JAVERS: GUT**

SNAPSHOT_SHOT_PK	TYPE	STATE	CHANGED_PROPERTIES	MANAGED_TYPE
100	INITIAL	{ "firstName": "Karsten", "lastName": "Silz", "id": 50 }	changedProperties=[ "firstName", "lastName", "id" ]	com.company.product.dto.Customer

LEICHT ZU ENTZIFFERN

**1 MAIN COMMITTER**

**SPEICHERPLATZ-VERBRAUCH**

**KEIN LÖSCHEN VON VERSIONEN**

**KEINE VERSIONS-SICHERHEIT**

**JAVERS: SCHLECHT**

WAS, WENN ER **AUSFÄLLT?**

GITHUB-PROJEKT **FORKEN?**

DATENBANK DIREKT **LESEN?**

JAVERS-NUTZER – **SPENDEN?**

**1 MAIN COMMITTER**

**1 MAIN COMMITTER**

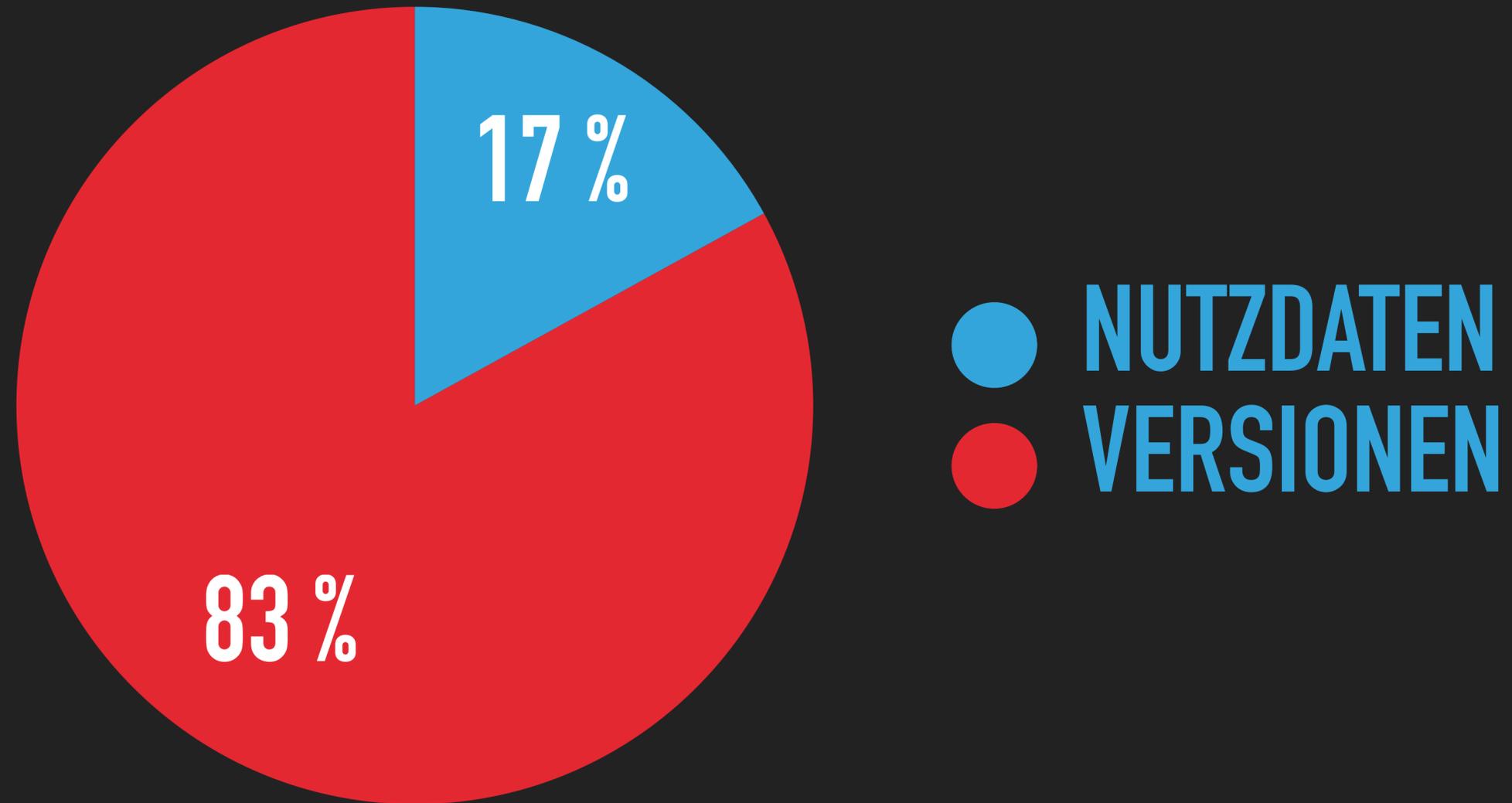
**SPEICHERPLATZ-VERBRAUCH**

**KEIN LÖSCHEN VON VERSIONEN**

**KEINE VERSIONS-SICHERHEIT**

**JAVERS: SCHLECHT**

# MEINE POSTGRES-DB



# SPEICHERPLATZ-VERBRAUCH

**5/6 DER DB FÜR VERSIONS-DATEN!**

**JSON-GRÖÖE SCHON OPTIMIERT...**

**...ABER VERSCHACHELTE OBJEKTE BEI MIR  
"MEHR PLATZ, DAFÜR EINFACHERE ABFRAGE"**

**SPEICHERPLATZ-VERBRAUCH**

ALTE VERSIONEN **LÖSCHEN?**

JAVERS IN LANGSAMERER,  
BILLIGERER, **SEPARATER DB?**

**SPEICHERPLATZ-VERBRAUCH**

**1 MAIN COMMITTER**

**SPEICHERPLATZ-VERBRAUCH**

**KEIN LÖSCHEN VON VERSIONEN**

**KEINE VERSIONS-SICHERHEIT**

**JAVERS: SCHLECHT**

NICHT IN API

LÖSUNG: JAVERS-RECORDS SELBST LÖSCHEN

KEIN LÖSCHEN  
VON VERSIONEN

**1 MAIN COMMITTER**

**SPEICHERPLATZ-VERBRAUCH**

**KEIN LÖSCHEN VON VERSIONEN**

**KEINE VERSIONS-SICHERHEIT**

**JAVERS: SCHLECHT**

NICHT IN API

LÖSUNG: PRÜFSUMMEN FÜR JAVERS-  
RECORDS IN EIGENER TABELLE?

KEINE VERSIONS-SICHERHEIT

**1 MAIN COMMITTER**

**SPEICHERPLATZ-VERBRAUCH**

**KEIN LÖSCHEN VON VERSIONEN**

**KEINE VERSIONS-SICHERHEIT**

**JAVERS: SCHLECHT**

**ZUSAMMENFASSUNG**

**NUTZER MACHEN FEHLER**

**VERSIONEN IM FRONT-END: NUTZER LÖSEN  
PROBLEME SELBST, OHNE UNS ENTWICKLER!**

**VERSIONIERUNG=  
ZEITMASCHINE**

EIGEN-ENTWICKLUNG 12 ★

HIBERNATE ENVERS 12 ★

JAVERS 13 ★

XTDB & DATOMIC 13 ★

# KANDIDATEN & BEWERTUNG

**ÄNDERUNGS-SICHERHEIT EIN MUSS: XTDB ODER DATOMIC**

**DB NICHT IN JAVERS: JAVERS + MYSQL/POSTGRES/  
MONGO DB... SEPARAT (MICROSERVICE?)**

**SONST: JAVERS**

**EMPFEHLUNG**

HAT FUNKTIONIERT

WÜRDEN SIE NOCH MAL NEHMEN

KANN ICH EMPFEHLEN

JAVERS: MEINE ERFAHRUNG

**AUßERHALB** DER APPLIKATIONS-PERSISTENZ

VERSIONIERUNG **EXISTIERENDER** FRONT-END-DTO

**ERZEUGE** VERSION

**FUNKTIONIERT**

**LEICHT ZU ENTZIFFERN**

**JAVERS: GUT**

**1 MAIN COMMITTER**

**SPEICHERPLATZ-VERBRAUCH**

**KEIN LÖSCHEN VON VERSIONEN**

**KEINE VERSIONS-SICHERHEIT**

**JAVERS: SCHLECHT**

**WERBUNG**

**MAI 2022**

**FULL-STACK **JAVA**: SPRING BOOT, ANGULAR, FLUTTER**

**FREIBERUFLER/BEFRISTETE ANSTELLUNG**

**MILTON KEYNES, LONDON, ODER **REMOTE****

**SUCHE **PROJEKT!****



*The End*



**FOLIEN**

**ZUSATZ-INFO**

**FEEDBACK**

**JAVERS-TUTORIAL:  
AUSFÜHRLICH & ENGLISCH**

**BPF.LI/SPA**

**SIE KÖNNEN JETZT GEHEN!**

**ODER EIN KURZES  
JAVERS-TUTORIAL SEHEN!**

**KURZE VERSION DES  
LANGEN JAVERS-  
TUTORIALS AUF WEBSITE**

**ERZEUGE VERSION**

**FRAGE ALLE VERSIONEN AB**

**FRAGE EINE VERSION AB**

**VERGLEICHE ZWEI VERSIONEN**

**BACK-END**

**ERZEUGE VERSION**

```
public class Customer {  
  
    private Long id;  
    private String firstName;  
    private String lastName;  
}
```

**DTO: OHNE JAVERS**

```
public class Customer {  
    @Id  
    private Long id;  
    private String firstName;  
    private String lastName;  
}
```

**DTO: MIT JAVERS**

```
public class Customer {  
    @Id  
    private Long id;  
    private String firstName;  
    private String lastName;  
}
```

**JAVERS ID: ENTITY-ID + KLASSEN-NAME**

**DTO: MIT JAVERS**

```
private Javers javers;  
    [...]
```

```
Customer customer =  
    new Customer(50L, "Karsten", "Silz");  
this.javers.commit("hmeier", customer);
```

**NUTZER-NAME**

**SPEICHERN**

~~ERZEUGE VERSION~~

FRAGE ALLE VERSIONEN AB

FRAGE EINE VERSION AB

VERGLEICHE ZWEI VERSIONEN

BACK-END

**FRAGE **ALLE** VERSIONEN AB**

```
JqlQuery allVersionsQuery = QueryBuilder  
    .byInstanceId(50L, Customer.class)
```

**JAVERS ID: ENTITY-ID + KLASSEN-NAME**

**ALLE VERSIONEN: CODE 1**

```
JqlQuery allVersionsQuery = QueryBuilder  
    .byInstanceId(50L, Customer.class)  
    .withShadowScope(ShadowScope.SHALLOW)
```

**WENIG DATEN LESEN**

**ALLE VERSIONEN: CODE 1**

```
JqlQuery allVersionsQuery = QueryBuilder
    .byInstanceId(50L, Customer.class)
    .withShadowScope(ShadowScope.SHALLOW)
    .build();
List<CdoSnapshot> versions=
    this.javers
        .findSnapshots(allVersionsQuery);
```

**NUR VERSIONS-INFORMATIONEN**

**ALLE VERSIONEN: CODE 1**

```
for (CdoSnapshot aVersion : versions) {
```

**NUR VERSIONS-INFORMATIONEN**

**ALLE VERSIONEN: CODE 2**

```
for (CdoSnapshot aVersion : versions) {  
    CommitMetadata metaData =  
        aVersion.getCommitMetadata();  
}
```

**VERSIONS-INFORMATIONEN**

**ALLE VERSIONEN: CODE 2**

```
for (CdoSnapshot aVersion : versions) {  
    CommitMetadata metaData =  
        aVersion.getCommitMetadata();  
    System.out.println(  
        "Version details: versionNumber="  
        + aVersion.getCommitId().value()
```

**VERSIONS-NUMMER**

**ALLE VERSIONEN: CODE 2**

```
for (CdoSnapshot aVersion : versions) {  
    CommitMetadata metaData =  
        aVersion.getCommitMetadata();  
    System.out.println(  
        "Version details: versionNumber=" +  
        + aVersion.getCommitId().value()  
        + ", author=" + metaData.getAuthor()  
        + ", time=" + metaData.getCommitDate());  
}
```

**ALLE VERSIONEN: CODE 2**

```
Version details:  
  versionNumber=1.00,  
  author=hmeier,  
  time=2021-11-04T22:00:27.567
```

# ALLE VERSIONEN: **ERGEBNIS**

# Versions

Total number of versions: 4

Timestamp 	Relative time 	Version 	Author 
<a href="#">Thu, 4 November 2021, 10:27 PM</a>	A day ago	4	 Karsten
<a href="#">Thu, 4 November 2021, 10:01 PM</a>	A day ago	3	 Karsten
<a href="#">Thu, 4 November 2021, 10:01 PM</a>	A day ago	2	 Karsten
<a href="#">Thu, 4 November 2021, 10:00 PM</a>	A day ago	1	 Karsten

ALLE VERSIONEN: FRONT-END

~~ERZEUGE VERSION~~

~~FRAGE ALLE VERSIONEN AB~~

FRAGE **EINE** VERSION AB

VERGLEICHE **ZWEI** VERSIONEN

**BACK-END**

**FRAGE EINE VERSION AB**

```
CommitId commitId = CommitId.valueOf("1.0");  
                                VERSIONS-NUMMER
```

**EINE VERSION: CODE 1**

```
CommitId commitId = CommitId.valueOf("1.0");  
JqlQuery oneVersionQuery = QueryBuilder  
    .byInstanceId(50L, Customer.class)
```

**JAVERS ID: ENTITY-ID + KLASSEN-NAME**

**EINE VERSION: CODE 1**

```
CommitId commitId = CommitId.valueOf("1.0");  
JqlQuery oneVersionQuery = QueryBuilder  
    .byInstanceId(50L, Customer.class)  
    .withCommitId(commitId)
```

**VERSIONS-NUMMER**

**EINE VERSION: CODE 1**

```
CommitId commitId = CommitId.valueOf("1.0");  
JqlQuery oneVersionQuery = QueryBuilder  
    .byInstanceId(50L, Customer.class)  
    .withCommitId(commitId)  
    .withShadowScope(ShadowScope.DEEP_PLUS)  
        ALLE DATEN LESEN
```

**EINE VERSION: CODE 1**

```
CommitId commitId = CommitId.valueOf("1.0");
JqlQuery oneVersionQuery = QueryBuilder
    .byInstanceId(50L, Customer.class)
    .withCommitId(commitId)
    .withShadowScope(ShadowScope.DEEP_PLUS)
    .build();
List<Shadow<Customer>> singleVersion =
    this.javers
        .findShadows(oneVersionQuery);
```

**EINE VERSION: CODE 1**

```
CommitId commitId = CommitId.valueOf("1.0");
JqlQuery oneVersionQuery = QueryBuilder
    .byInstanceId(50L, Customer.class)
    .withCommitId(commitId)
    .withShadowScope(ShadowScope.DEEP_PLUS)
    .build();
List<Shadow<Customer>> singleVersion =
    this.javers ENTITY LESEN
        .findShadows(oneVersionQuery);
```

**EINE VERSION: CODE 1**

```
for (Shadow<Customer> aVersion : versions) {
```

**GANZE ENTITY**

**EINE VERSION: CODE 2**

```
for (Shadow<Customer> aVersion : versions) {  
    Customer customer = aVersion.get();
```

**VERSION WIEDERHERSTELLEN**

**EINE VERSION: CODE 2**

```
for (Shadow<Customer> aVersion : versions) {  
    Customer customer = aVersion.get();  
    System.out.println(  
        "Version details: versionNumber="  
        + aVersion.getCommitId().value()  
        + " " + VERSIONS-NUMMER  
    );  
}
```

**EINE VERSION: CODE 2**

```
for (Shadow<Customer> aVersion : versions) {  
    Customer customer = aVersion.get();  
    System.out.println(  
        "Version details: versionNumber="  
        + aVersion.getCommitId().value()  
        + ", customer=" + customer);  
}
```

**WIEDERHERGESTELLTE VERSION**

**EINE VERSION: CODE 2**

```
Version details:  
  versionNumber=1.00,  
  customer=Customer {  
    id=50,  
    firstName='Karsten',  
    lastName='Silz' }
```

**EINE VERSION: ERGEBNIS**



# Version Details

← Back

## GENERISCH

↔ Compare

with

↶ Clear comparison



**Version 2 by Karsten**

A day ago (Thu, 4 November 2021, 10:01 PM)

First name

Jon

Last name

Snow

Email

[jon.snow@nights-watch.com](mailto:jon.snow@nights-watch.com)

EINE VERSION: FRONT-END



# Version Details

← Back

↔ Compare

with



Clear comparison



## Version 2 by Karsten

A day ago (Thu, 11 November 2021, 10:01 PM)

**SPEZIFISCH**

First name

Jon

Last name

Snow

Email

[jon.snow@nights-watch.com](mailto:jon.snow@nights-watch.com)

**EINE VERSION: FRONT-END**

~~ERZEUGE VERSION~~

~~FRAGE ALLE VERSIONEN AB~~

~~FRAGE EINE VERSION AB~~

VERGLEICHE ZWEI VERSIONEN

BACK-END

**VERGLEICHE ZWEI VERSIONEN**

```
Customer oldOne =  
    new Customer(50L, "Karsten", "Silz");
```

**VERSION 1**

**ZWEI VERSIONEN: CODE 1**

```
Customer oldOne =  
    new Customer(50L, "Karsten", "Silz");  
Customer newOne =  
    new Customer(50L, "Joe", "Cool");
```

**VERSION 2**

**ZWEI VERSIONEN: CODE 1**

```
Customer oldOne =  
    new Customer(50L, "Karsten", "Silz");  
Customer newOne =  
    new Customer(50L, "Joe", "Cool");  
Diff differences =  
    this.javers.compare(oldOne, newOne);
```

**ZWEI VERSIONEN: CODE 1**

```
Changes listOfDifferences =  
    differences.getChanges();
```

**ALLE UNTERSCHIEDE**

**ZWEI VERSIONEN: CODE 2**

```
Changes listOfDifferences =  
    differences.getChanges();
```

```
for (Change aChange : listOfDifferences) {  
    System.out.println(" Change: "  
    + aChange.toString());  
}
```

**EIN UNTERSCHIED**

**ZWEI VERSIONEN: CODE 2**

```
Change: ValueChange {  
  'firstName' value changed  
  from 'Karsten' to 'Joe' }
```

```
Change: ValueChange {  
  'lastName' value changed  
  from 'Silz' to 'Cool' }
```

# ZWEI VERSIONEN: ERGEBNIS

~~ERZEUGE VERSION~~

~~FRAGE ALLE VERSIONEN AB~~

~~FRAGE EINE VERSION AB~~

~~VERGLEICHE ZWEI VERSIONEN~~

BACK-END



**FOLIEN**

**ZUSATZ-INFO**

**FEEDBACK**

**JAVERS-TUTORIAL:  
AUSFÜHRLICH & ENGLISCH**

**BPF.LI/SPA**

<https://pixabay.com/photos/man-selfie-male-self-portrait-5545030>

<https://pixabay.com/photos/hipster-man-beard-look-glasses-358479>

<https://commons.wikimedia.org/wiki/File:>

[TeamTimeCar.com-BTTF\\_DeLorean\\_Time\\_Machine-OtoGodfrey.com-JMortonPhoto.com-07.jpg](TeamTimeCar.com-BTTF_DeLorean_Time_Machine-OtoGodfrey.com-JMortonPhoto.com-07.jpg)

<https://www.publicdomainpictures.net/en/view-image.php?image=283258&picture=hands-raised-up>

<https://openclipart.org/detail/324093/rolling-up-sleeves>

<https://commons.wikimedia.org/wiki/File:The-end-folks.jpg>

# BILD-QUELLEN