



KARSTEN SILZ

3. MAI 2022



# WIE DATEN-VERSIONEN IN FRONT-ENDS VON JAVA-ANWENDUNGEN ZEIGEN?

	<b>Version 2 by Karsten</b> Sat, 26 September 2020, 14:38		<b>Version 1 by Frank</b> Thu, 17 September 2020, 22:14
		<b>2 changes</b> , marked <b>in red</b> .	
<b>First name</b>	Jon	<b>First name</b>	John
<b>Last name</b>	Snow	<b>Last name</b>	Snow
<b>Email</b>	<a href="mailto:jon.snow@nights-watch.com">jon.snow@nights-watch.com</a>	<b>Email</b>	<a href="mailto:john.snow@nights-watch.com">john.snow@nights-watch.com</a>

**WENIGER** SUPPORT-  
ANFRAGEN AN  
**JAVA**-ENTWICKLER!

**WARUM BIN ICH EXPERTE?**





**23 JAHRE JAVA-ENTWICKLER**

**INFOQ-JAVA-EDITOR, PK QCON LONDON**

**6 KRITERIEN**

**NUTZE JAVERS, SPENDE \$10/MONAT –  
VERKAUFE NIX**



**SIE ENTSCHEIDEN!**



**FOLIEN &  
MEHR**

**BPF.LI/JOY**



**DAS SCHLIMMSTE FÜR  
ANWENDUNGEN?**





**ALLE DATEN WEG!**

**LÖSUNG: BACK-UP**





**LÖSUNG: BACK-UP**

**ALLES GUT!**

**NEIN!**



**JEDEN TAG:**

**EIN PAAR DATEN WEG!**

VERTIPPEN, **LÖSCHEN**, FALSCH BEDIENEN, ...

DB-BACKUP **KEINE LÖSUNG**

NUTZER  => **ENTWICKLER**

**NUTZER MACHEN FEHLER!**

**BEISPIEL:**  
**KUNDE BEKAM ON-**  
**BOARDING-EMAIL NICHT**

**TICKET:** FACH-ABTEILUNG, SUPPORT, OPS, ENTWICKLER

EMAIL-ADRESSE **OK**

LOG-DATEI: EMAIL-ADRESSE ANFÄNGLICH **FALSCH**

**WARUM** – UND WER HAT SIE **GEÄNDERT?**

**OHNE DATEN-VERSIONEN**



VERSIONEN WICHTIGER DATEN **SPEICHERN**

DATEN-VERSIONEN IN **BACK-END-LOGS**

**TICKET:** FACH-ABTEILUNG, SUPPORT, OPS, ENTWICKLER

**ENTWICKLER** SUCHEN & LIEFERN VERSIONEN **MANUELL**

**DATEN-VERSIONEN #1**



VERSIONEN WICHTIGER DATEN **SPEICHERN**

DATEN-VERSIONEN IM FRONT-END **ANZEIGEN**

**NUTZER** SUCHEN & FINDEN VERSIONEN

KEIN **TICKET!**

**DATEN-VERSIONEN #2**





## 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)



### 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)

**ANFÄNGLICH**  
**FALSCH**



**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)

**KORRIGIERT**

**ANFÄNGLICH**

**FALSCH**

**NUTZER** SUCHEN &  
FINDEN VERSIONEN

**WENIGER** SUPPORT-  
ANFRAGEN AN  
**JAVA**-ENTWICKLER!





**KEINE** VERSIONS-HISTORIE: TICKET-PING-PONG,  
OFFENE FRAGEN

VERSIONS-HISTORIE IN **LOG-DATEIEN**: TICKET,  
ENTWICKLER FINDEN LÖSUNG MANUELL

VERSIONS-HISTORIE IM **FRONT-END**: KEIN  
TICKET, NUTZER SUCHEN & FINDEN VERSIONEN

**WIE DATEN-VERSIONEN  
IN FRONT-ENDS VON  
JAVA-ANWENDUNGEN  
ZEIGEN?**



# DATEN- VERSIONEN

ANFORDERUNGEN

KANDIDATEN

BEWERTUNG

JAVERS-ERFAHRUNGEN

**ANFORDERUNGEN**

**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**

**ARCHITEKTUR: MICROSERVICE & MONOLITH**

**DB: RELATIONAL & NOSQL**

**MINIMAL-INVASIV**

**GERINGER AUFWAND IN ENTWICKLUNG & BETRIEB**

**MEHR ANFORDERUNGEN**

~~ANFORDERUNGEN~~  
ANFORDERUNGEN

KANDIDATEN

BEWERTUNG

JAVERS-ERFAHRUNGEN

DATEN-  
VERSIONEN

**KANDIDATEN**

**SPEICHERN: JAVA SERIALIZATION**

**1 TABELLE**

**ANNOTATIONEN FÜR KLASSEN-ATTRIBUTE**

**EIGEN-ENTWICKLUNG**

WIE OBJEKTE VERGLEICHEN?

NUTZUNG IN ANDEREN PROJEKTEN?

EIGEN-ENTWICKLUNG?



# HIBERNATE-ERWEITERUNG

1 GLOBALE TABELLE + 1 TABELLE PRO ENTITY

FÜR JEDE ENTITY "AUDIT ENTITY"-KOPIEN

@AUDITED FÜR KLASSEN-ATTRIBUTE

# HIBERNATE ENVERS

WIE OBJEKTE VERGLEICHEN?

WIE RELATIONAL OHNE HIBERNATE?

WIE NOSQL?

HIBERNATE ENVERS?

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

OPEN-SOURCE

SPEICHERT **JSON** IN **4** TABELLEN

1.0: DEZEMBER 2014

**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?

**"IMMUTABLE RECORDS" => ÄNDERUNGSSICHERHEIT**

**OPEN-SOURCE + KOMMERZIELL**

**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~~  
~~ANFORDERUNGEN~~

~~KANDIDATEN~~  
~~KANDIDATEN~~

BEWERTUNG

JAVERS-ERFAHRUNGEN

DATEN-  
VERSIONEN

**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**

**BEWERTUNG**

~~ANFORDERUNGEN~~

~~KANDIDATEN~~

~~BEWERTUNG~~

JAVERS-ERFAHRUNGEN

DATEN-  
VERSIONEN

# JAVERS- ERFAHRUNGEN



"YOUR HOME IN GOOD HANDS"

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

SEIT **MAI 2021** IN CLOSED BETA

**JAVERS** FÜR KUNDEN-DATEN

**MEIN PROJEKT**

**CAT-  
SITTER  
FRONT-  
END**

**FLUTTER**

**IOS & ANDROID**

**REST JSON**

**SPRING BOOT**

**PG**

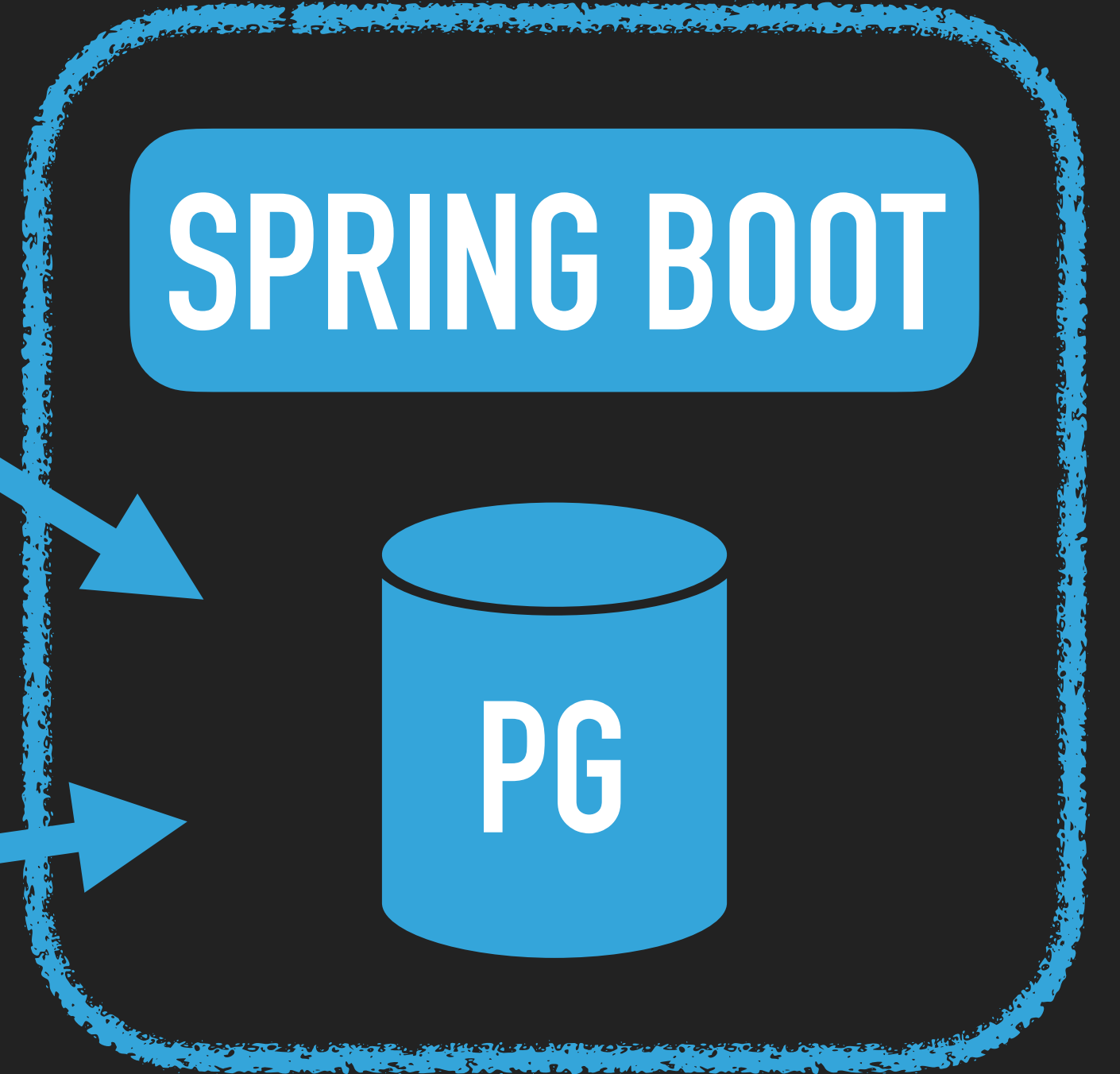
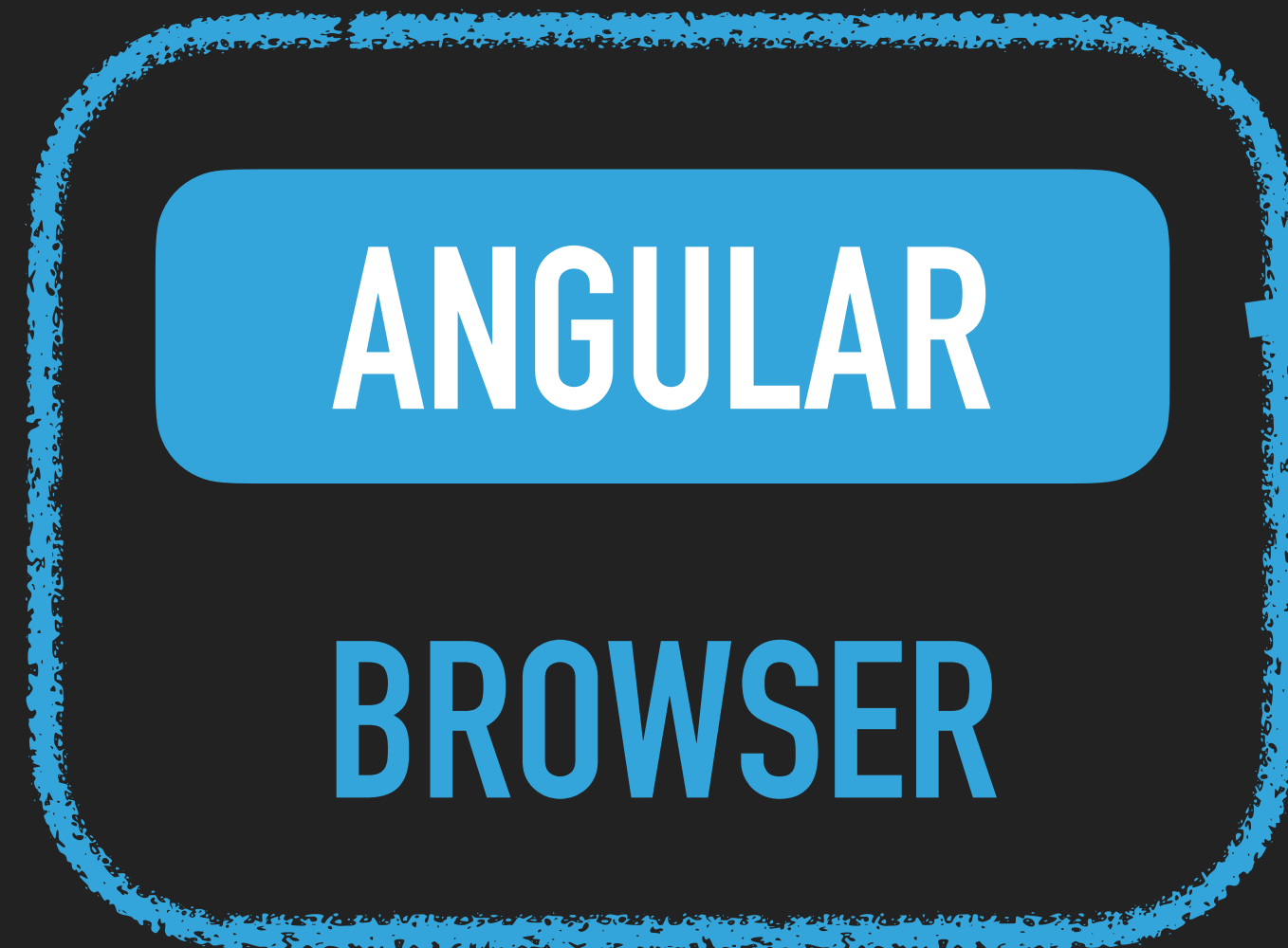
**MANAGER  
FRONT-  
END**

**ANGULAR**

**BROWSER**

**REST JSON**

**BACK-END**



HAT **FUNKTIONIERT**

WÜRD<sup>Ü</sup>E ICH **NOCH MAL** NEHMEN

KANN ICH **EMPFEHLEN**

**BEWERTUNG**

ZEIGE **ALLE** VERSIONEN

ZEIGE **EINE** VERSION

ZEIGE UNTERSCHIEDE  
**ZWEIER** VERSIONEN

**FRONT**-END

ERZEUGE VERSION

FRAGE **ALLE** VERSIONEN AB

FRAGE **EINE** VERSION AB

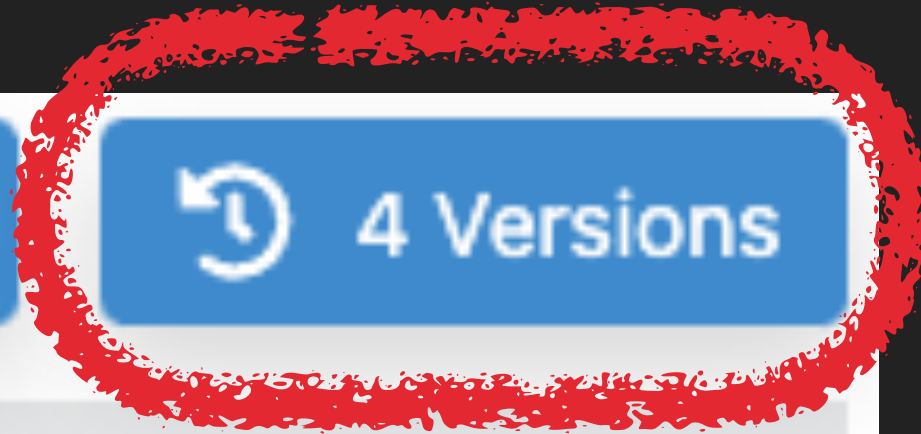
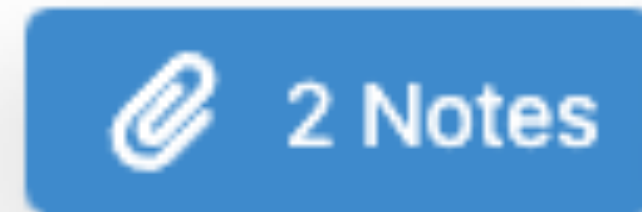
VERGLEICHE **ZWEI** VERSIONEN

**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

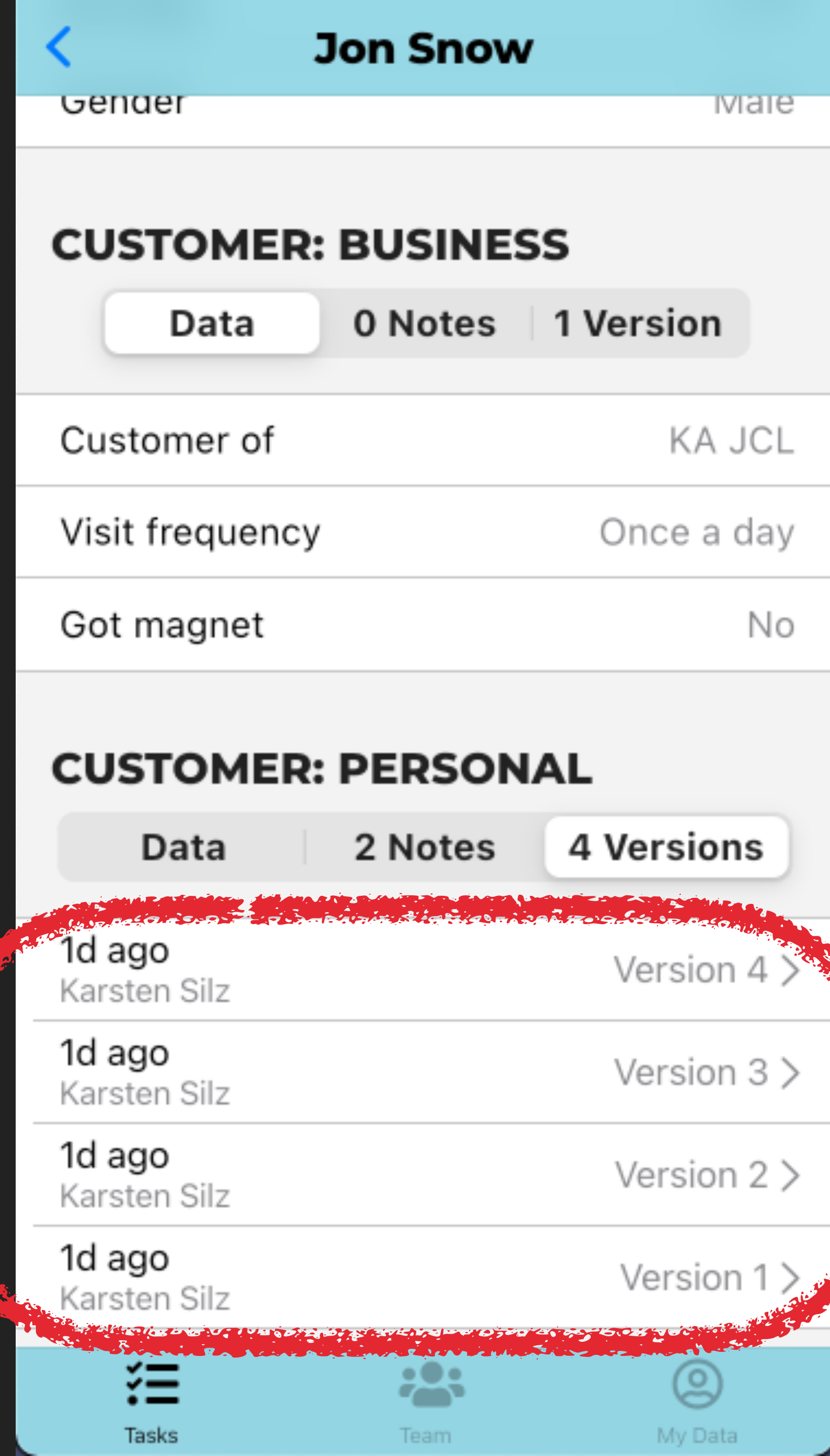
Total number of versions: 4

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

Version 1 (previous) ▾

↺ 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)



## Version 1 by Karsten

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

**2 changes**, marked **in red**.

**First name**

John

**Last name**

Snow

**Email**

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

# 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

**GENERISCH**

Email

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

**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

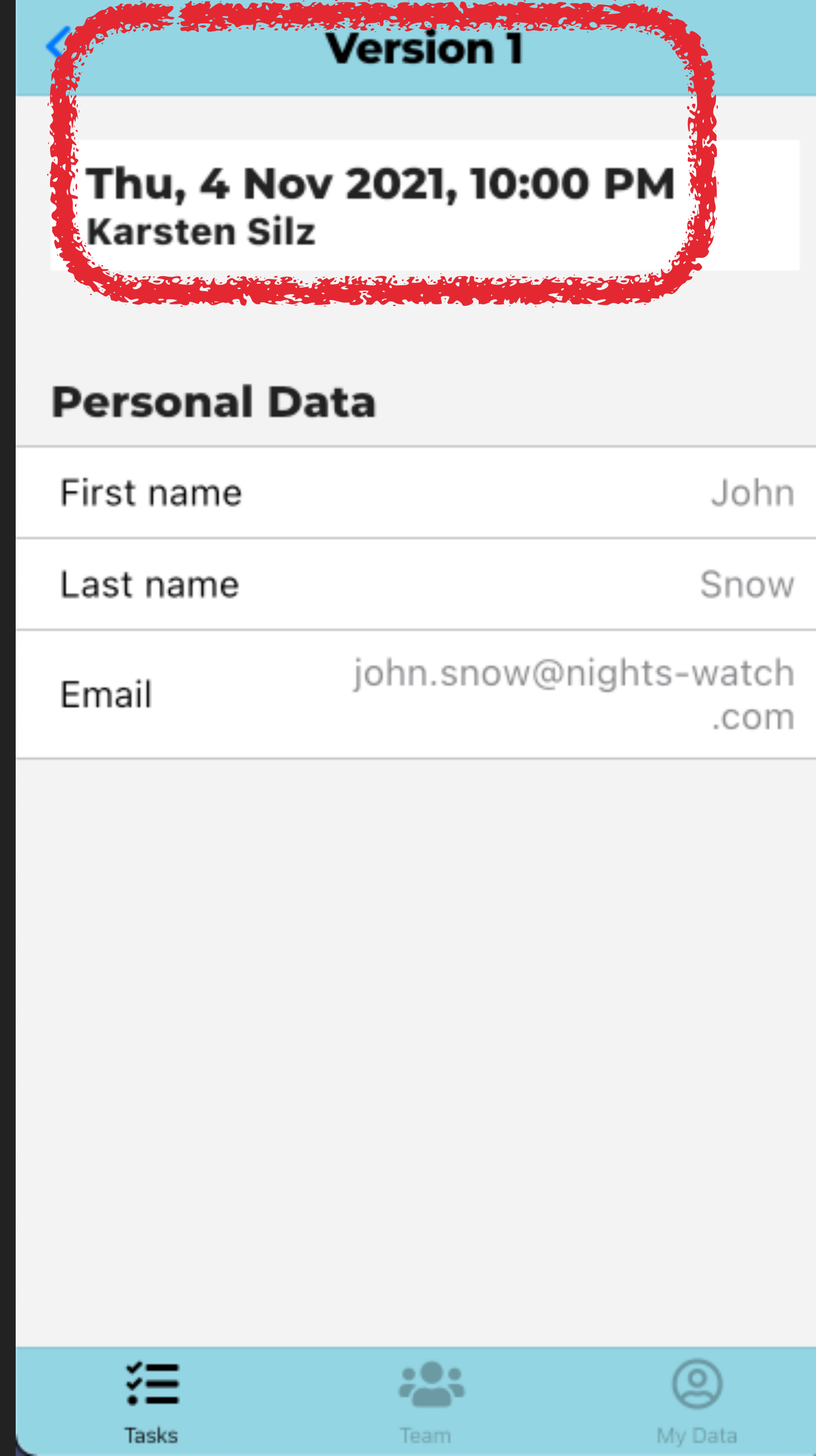
[jon.snow@nights-watch.com](mailto:jon.snow@nights-watch.com)

**SPEZIFISCH**

**BROWSER**

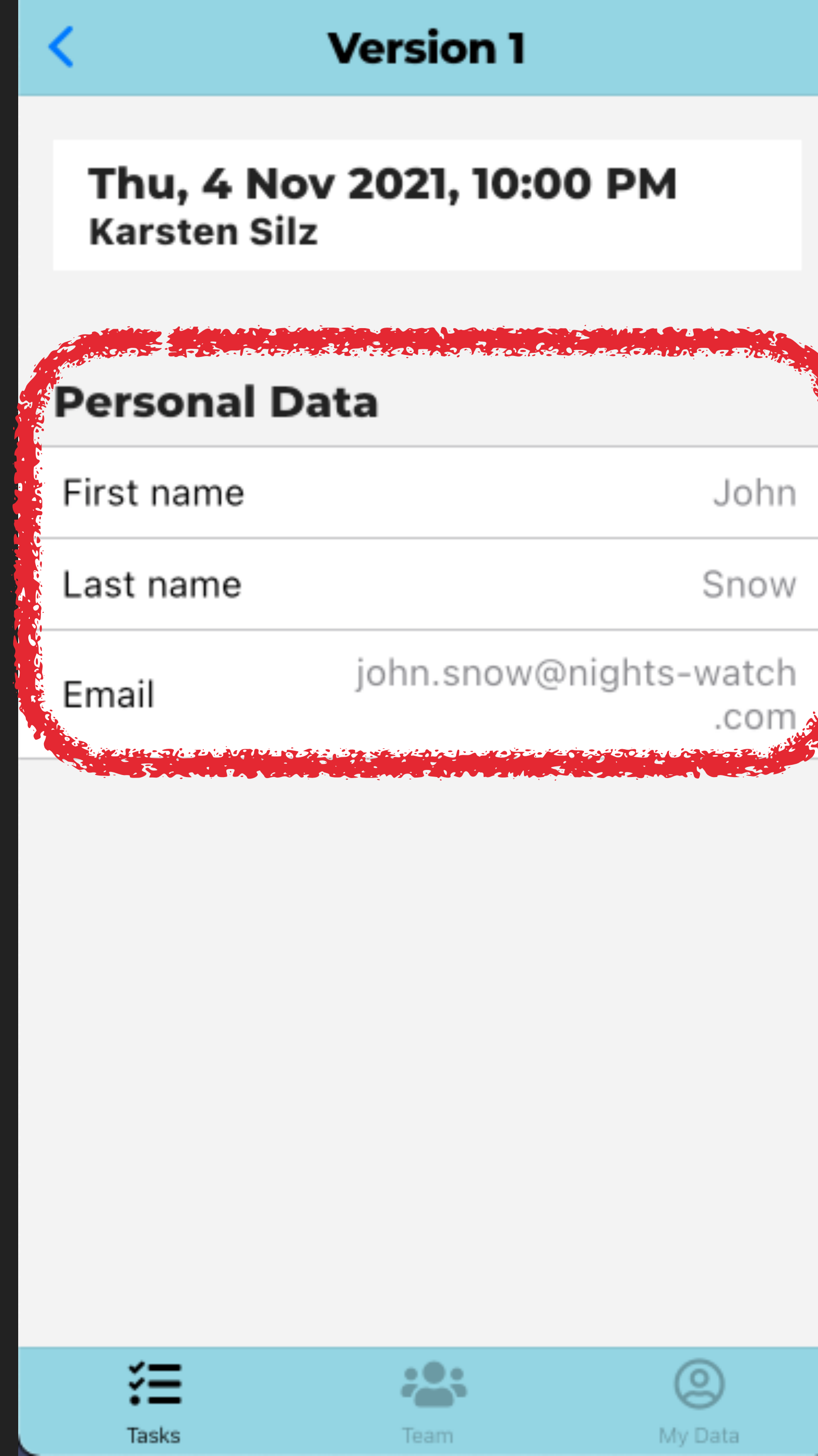
GENERISCH

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)

**First name**

Jon

**Last name**

Snow

**Email**

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



## Version 1 by Karsten

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

**2 changes**, marked **in red**.

**First name**

John

**Last name**

Snow

**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.

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  
**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)

**BROWSER**

**FRONT-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

**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**

**ERZEUGE VERSION**

**AUßERHALB DER APPLIKATIONS-PERSISTENZ**

**VERSIONIERUNG EXISTIERENDER DTO**

**MEINE ARCHITEKTUR**



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 into the top of the first box. A blue rounded rectangular label 'EMPFANGE DTO' is positioned at the tip of the arrow, overlapping 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

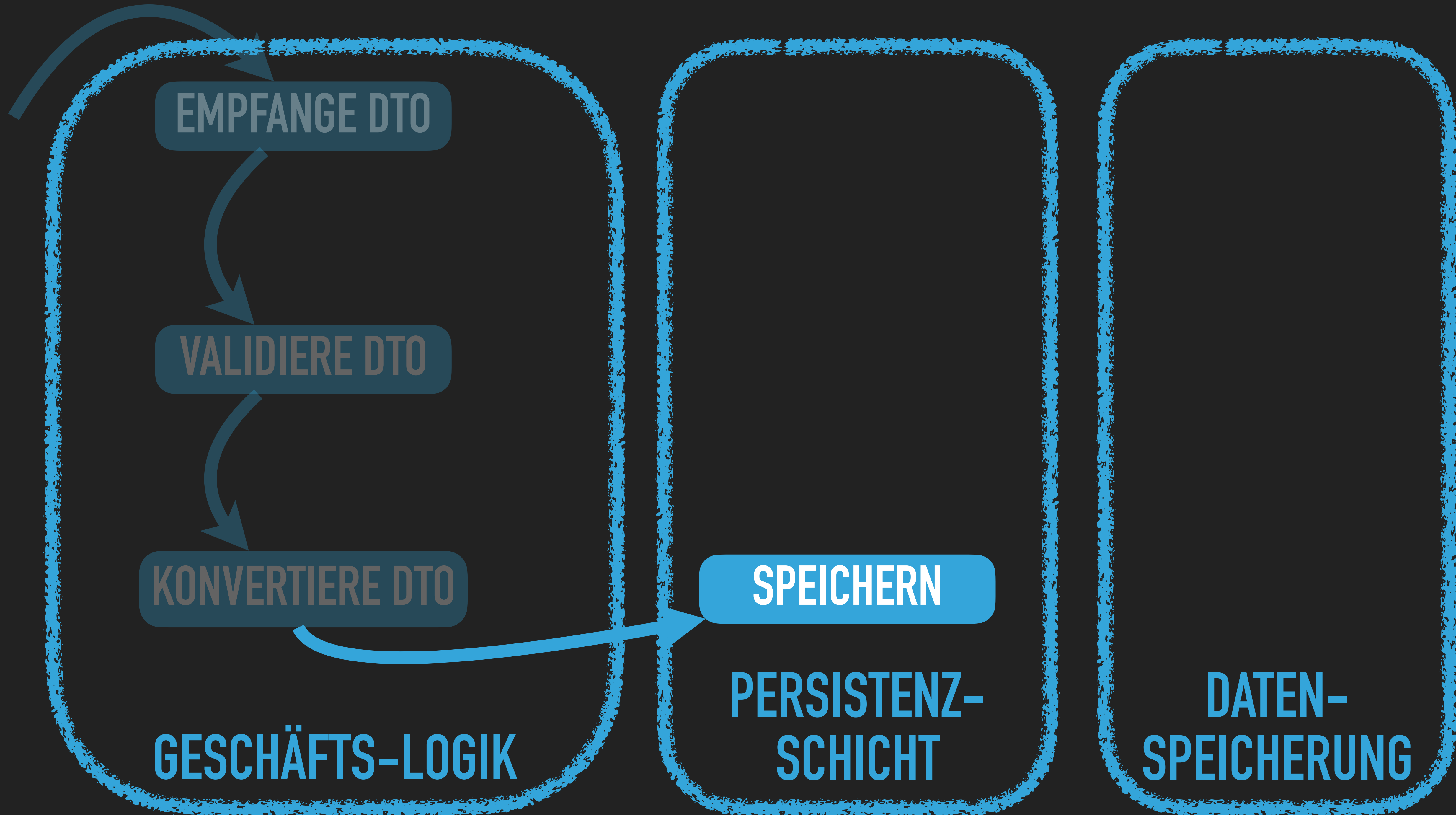
VALIDIERE DTO

KONVERTIERE DTO

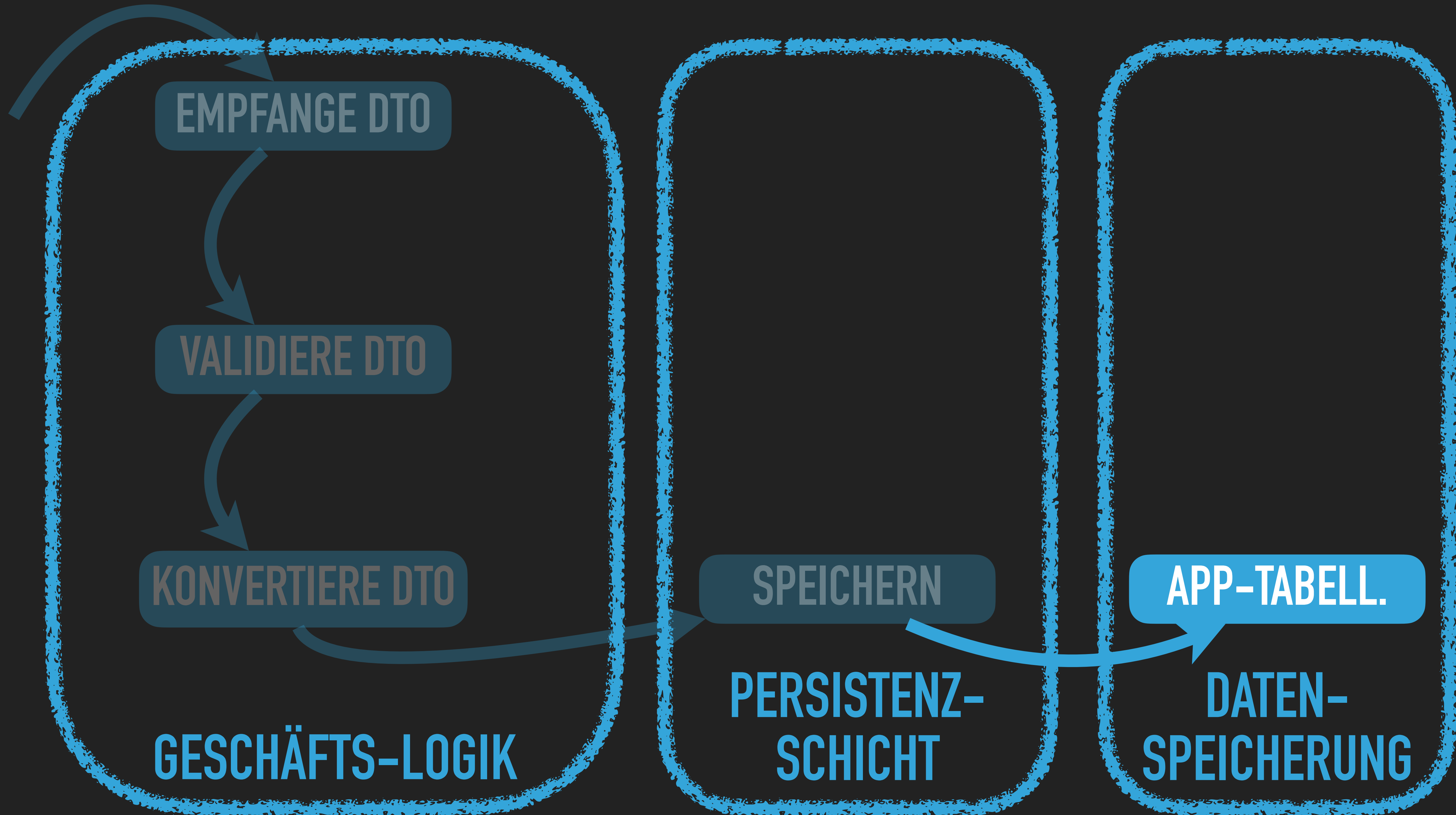
GESCHÄFTS-LOGIK

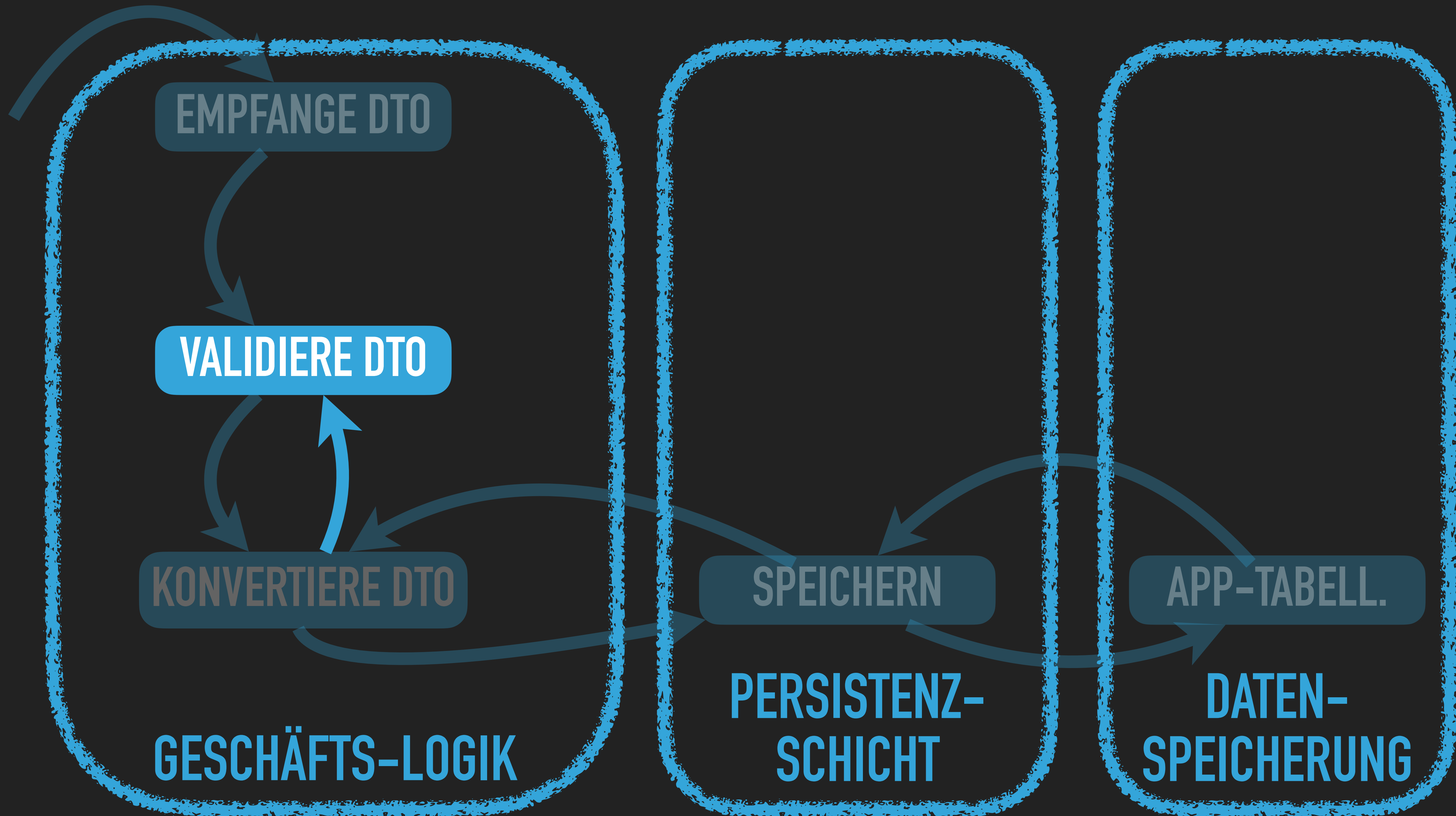
PERSISTENZ-  
SCHICHT

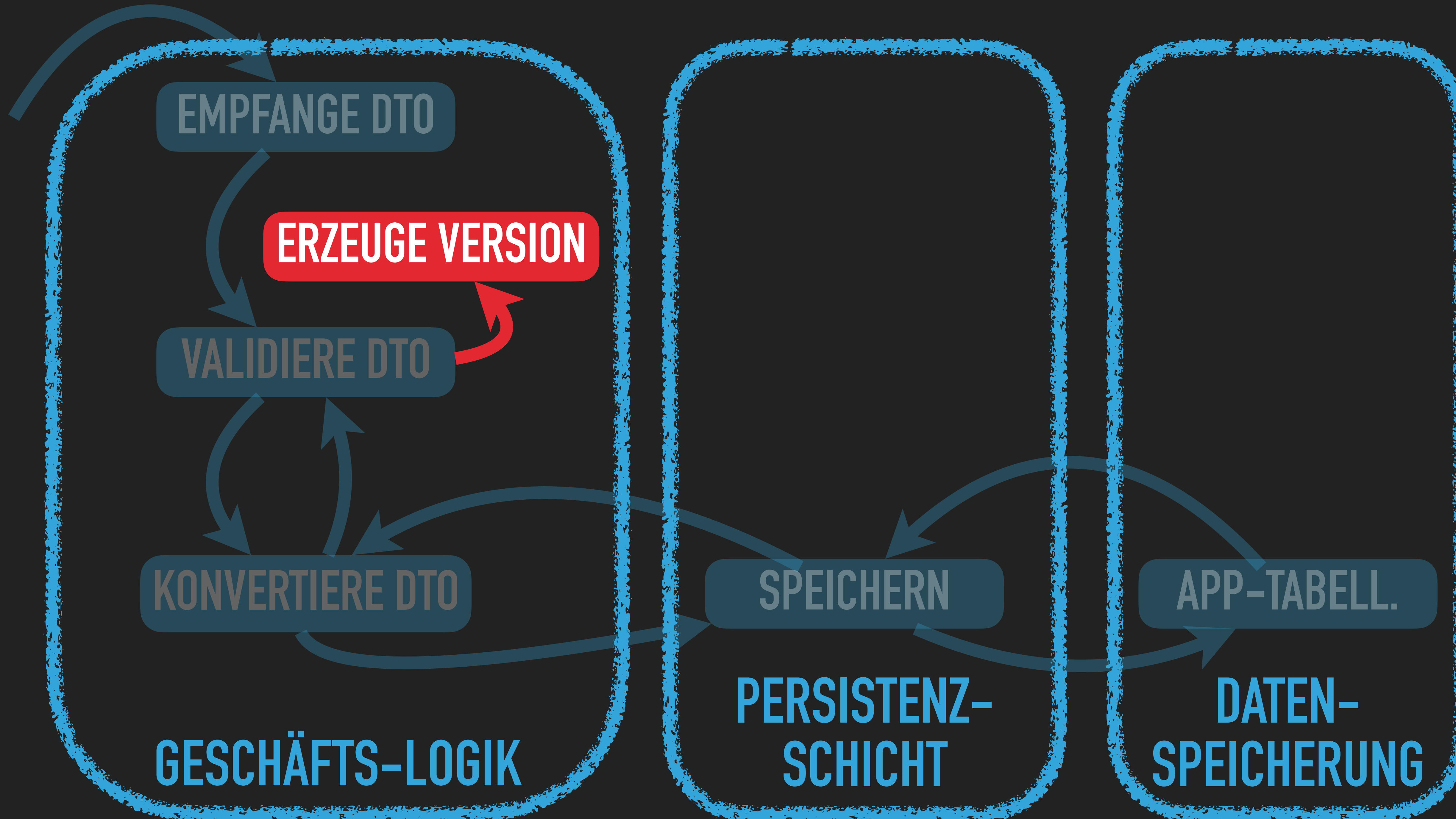
DATEN-  
SPEICHERUNG



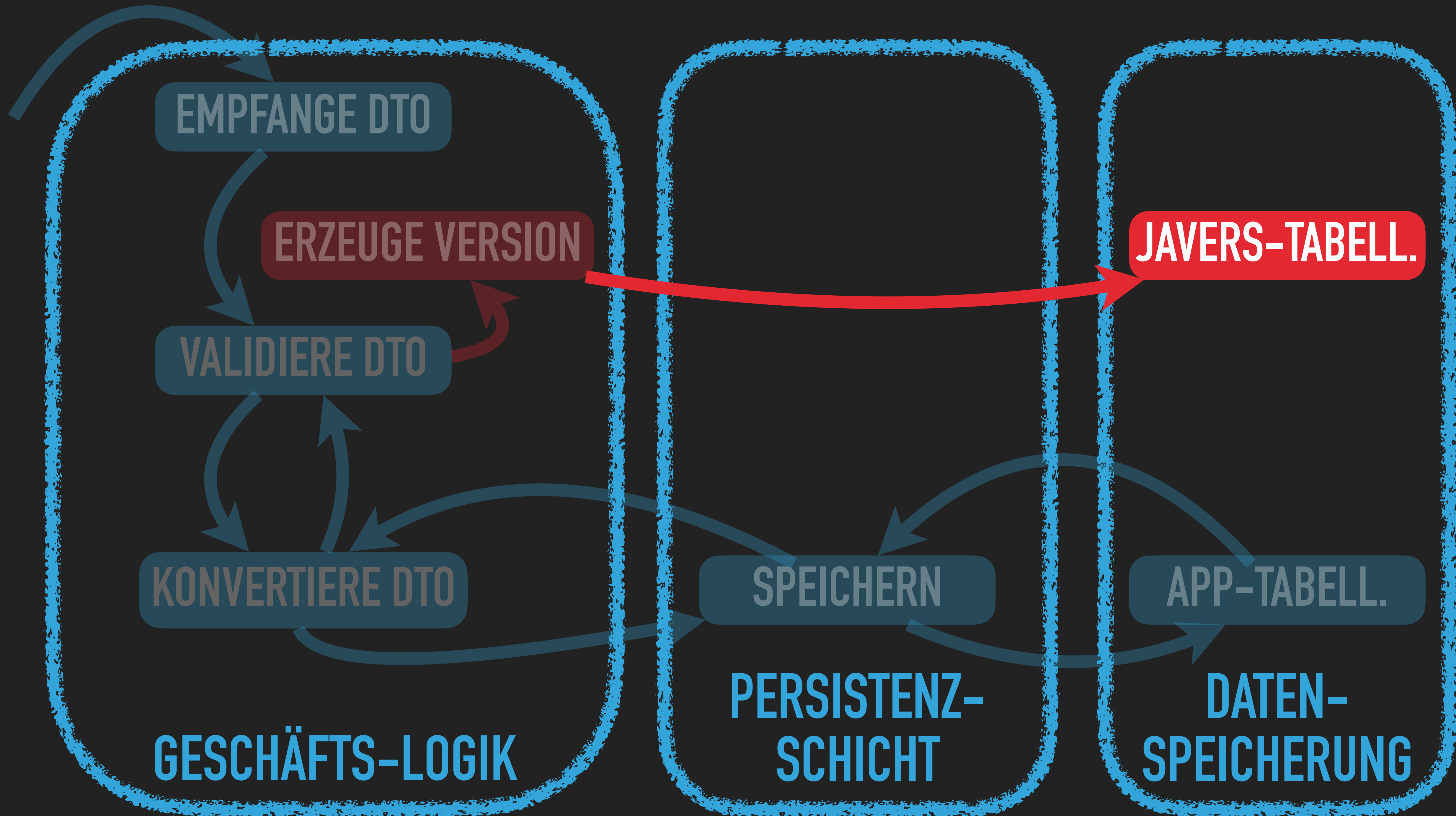




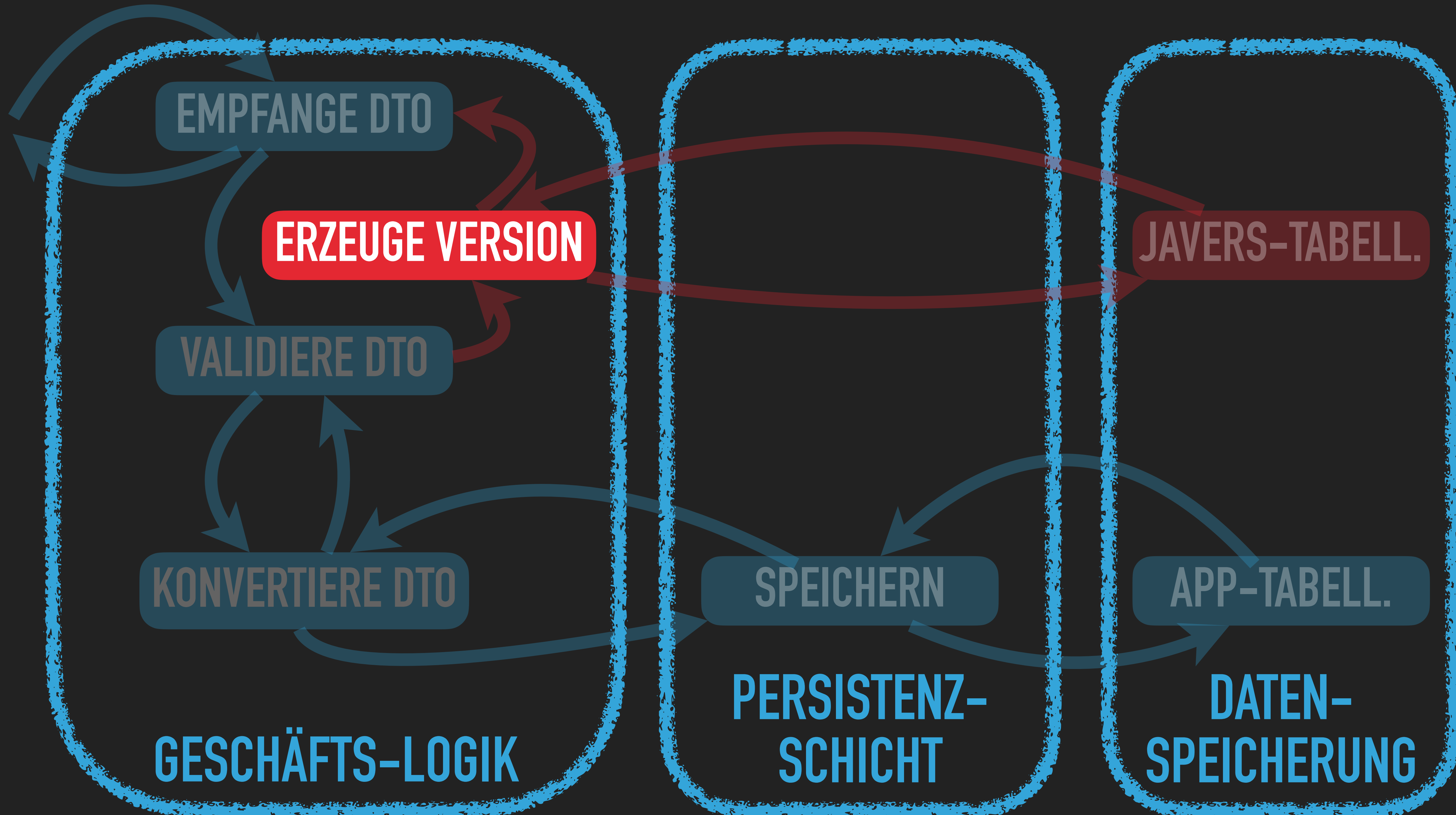


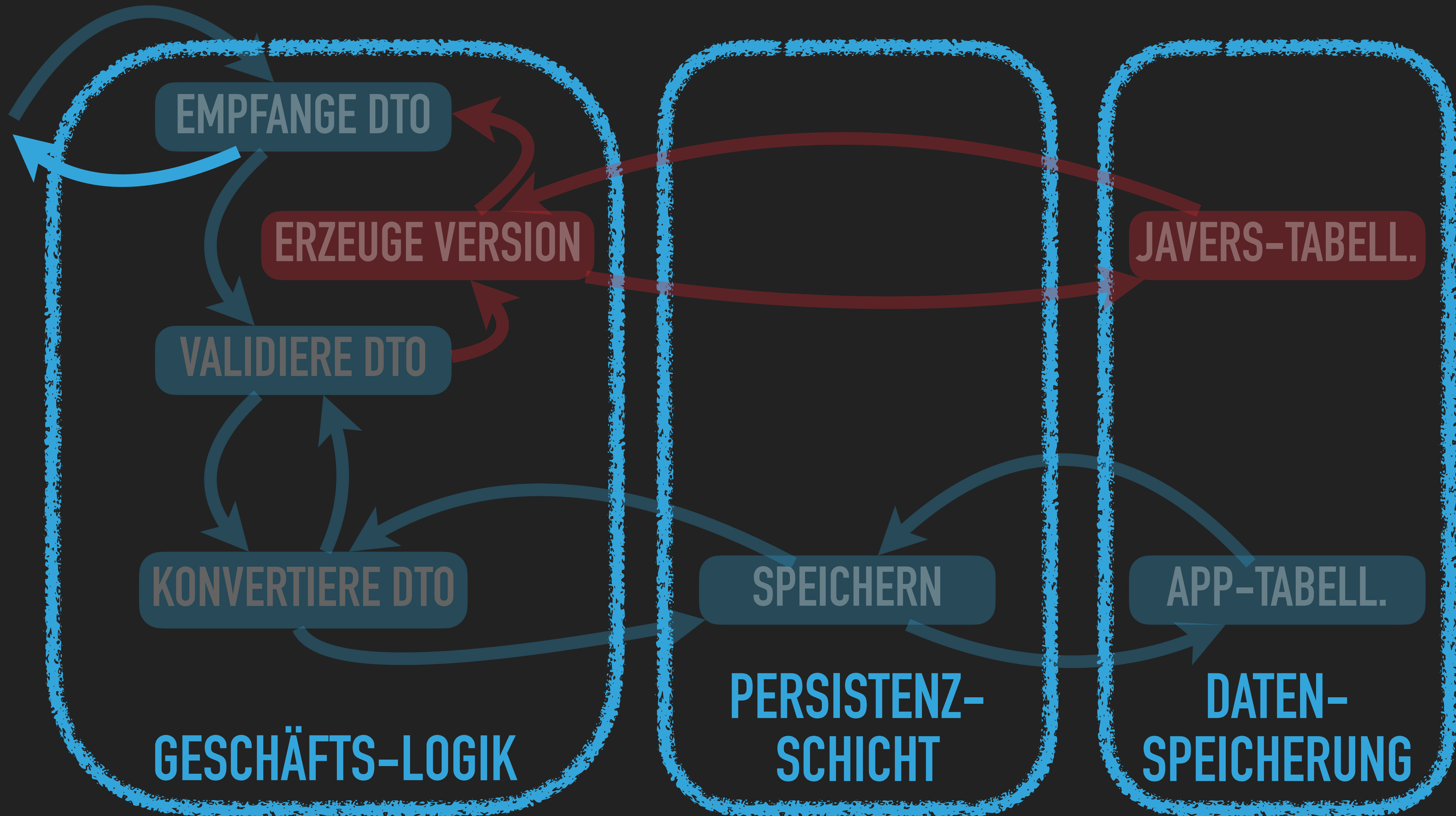


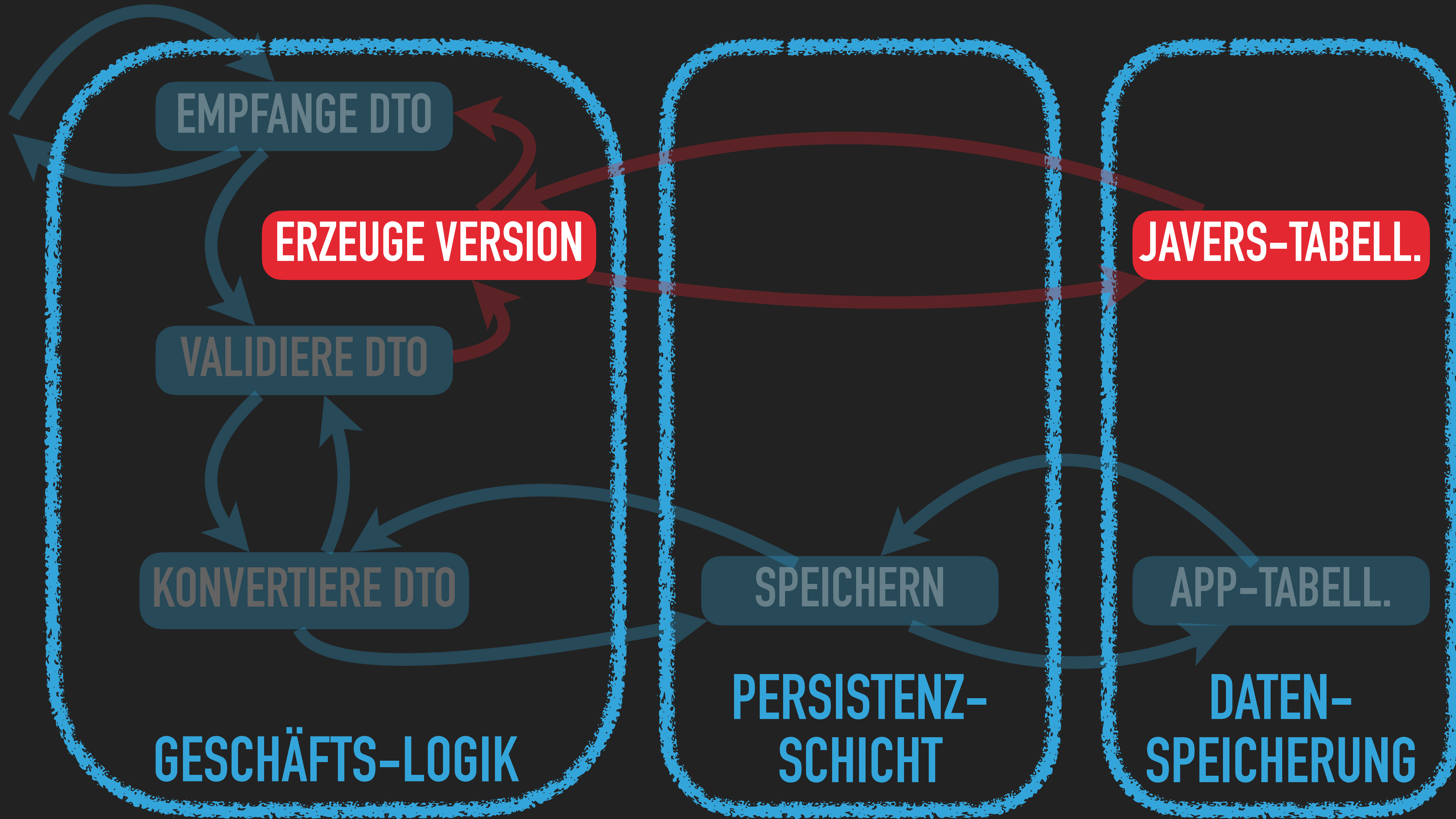












**KEINE** ÄNDERUNG AN PERSISTENZ

**KEINE** NEUE ENTITY/DTO FÜR VERSIONIERUNG

**KEINE** NEUE DTO FÜR FRONT-END

**ARCHITEKTUR: GUT**

VERSIONIERUNG SCHEITERT: VERSION FEHLT!

IN PERSISTENZ-TRANSAKTION: ALLES SCHEITERT!

ARCHITEKTUR: SCHLECHT



```
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;  
}
```

**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**



**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_PROPERTIES	MANAGED_TYPE
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_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_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

**KLASSEN-NAME DER ENTITY**

**JV\_SNAPSHOT**

**ERZEUGE VERSION**

~~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~~**



**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**





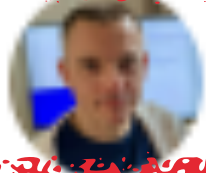





```
Version details:  
  versionNumber=1.00,  
  author=hmeier,  
  time=2021-11-04T22:00:27.567
```

# 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

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

~~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~~**

**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)  
    .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 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

↔ 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

## 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, 4 November 2021, 10:01 PM)

First name

Jon

Last name

Snow

Email

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

**SPEZIFISCH**

**EINE VERSION: FRONT-END**

**FRAGE EINE VERSION AB**

~~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

**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**

**VERGLEICHE ZWEI VERSIONEN**

~~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

**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

**FUNKTIONIERT**

**LEICHT ZU ENTZIFFERN**

**JAVERS: GUT**

**JAVERS: SCHLECHT**



WAS, WENN ER **AUSFÄLLT?**

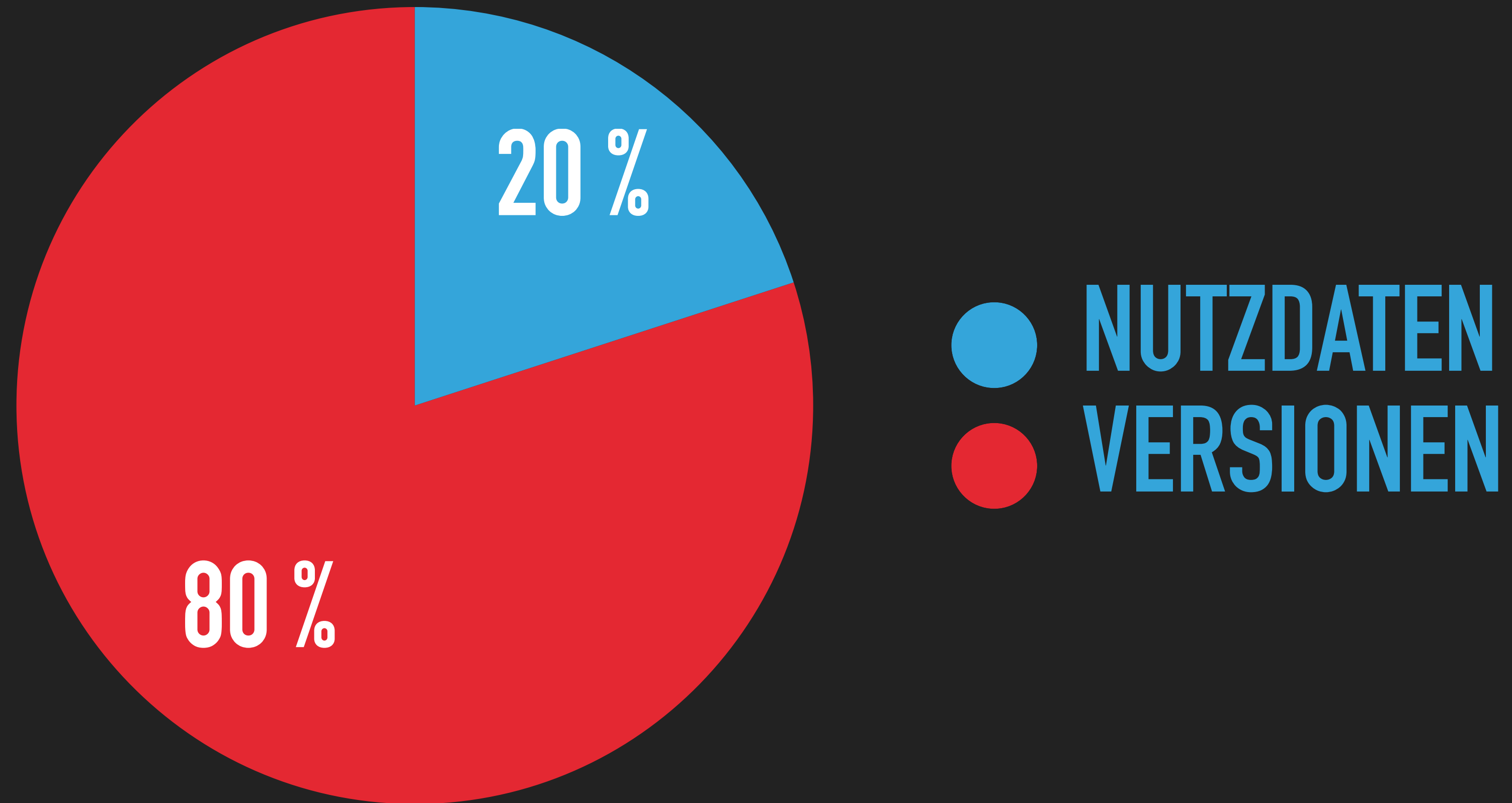
GITHUB-PROJEKT **FORKEN?**

DATENBANK DIREKT **LESEN?**

JAVERS-NUTZER – **SPENDEN?**

**1 MAIN COMMITTER**

# MEINE POSTGRES-DB



# SPEICHERPLATZ-VERBRAUCH

**4/5 DER DB FÜR VERSIONS-DATEN!**

**JSON-GRÖÖE SCHON OPTIMIERT...**

**...ABER VERSCHACHELTE OBJEKTE  
"MEHR PLATZ, ABER EINFACHERE ABFRAGE"**

**SPEICHERPLATZ-VERBRAUCH**

ALTE VERSIONEN **LÖSCHEN?**

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

**SPEICHERPLATZ-VERBRAUCH**

**NICHT IN API**

**LÖSUNG: JAVERS-RECORDS SELBST LÖSCHEN**

**KEIN LÖSCHEN  
VON VERSIONEN**

**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**

# JAVERS- ERFAHRUNGEN



# DATEN- VERSIONEN

~~ANFORDERUNGEN~~

~~KANDIDATEN~~

~~BEWERTUNG~~

~~JAVERS ERFABRUNGEN~~

**ZUSAMMENFASSUNG**

**NUTZER MACHEN FEHLER**

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

**DATEN-VERSIONEN?**

EIGEN-ENTWICKLUNG 12 ★

HIBERNATE ENVERS 12 ★

JAVERS 13 ★

XTDB & DATOMIC 13 ★

KANDIDATEN

**AUßERHALB** DER APPLIKATIONS-PERSISTENZ

VERSIONIERUNG **EXISTIERENDER** FRONT-END-DTO

**JAVERS: ERZEUGE VERSION**

**FUNKTIONIERT**

**LEICHT ZU ENTZIFFERN**

**JAVERS: GUT**

**1 MAIN COMMITTER**

**SPEICHERPLATZ-VERBRAUCH**

**KEIN LÖSCHEN VON VERSIONEN**

**KEINE VERSIONS-SICHERHEIT**

**JAVERS: SCHLECHT**

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

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

**SONST: JAVERS**

**EMPFEHLUNG**



**WENIGER** SUPPORT-  
ANFRAGEN AN  
**JAVA**-ENTWICKLER!



*The End*

**MEIN VORTRAG HÖRT  
NICHT AUF, WENN  
ICH AUFHÖRE  
VORZUTRAGEN! <sup>TM</sup>**



**FOLIEN**

**ZUSATZ-INFO**

**JAVERS-TUTORIAL:  
AUSFÜHRLICH & ENGLISCH**

**JAVA-NEWSLETTER**

**BPF.LI/JOY**





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

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

**BILD-QUELLEN**