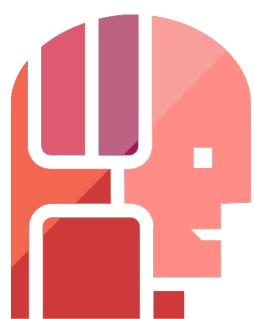


2025.2



E-PIX

Identitätsmanagement &  
Record Linkage

**HERAUSGEBER:** Unabhängige Treuhandstelle der Universitätsmedizin Greifswald

**AUTOR:** Christopher Hampf

**WEBSEITE:** [www.ths-greifswald.de](http://www.ths-greifswald.de)

**KONTAKT:** [kontakt-ths@med.uni-greifswald.de](mailto:kontakt-ths@med.uni-greifswald.de)

**VERÖFFENTLICHUNG:** 09. Dezember 2025



# Inhaltsverzeichnis

## I

## Einführung

<b>1</b>	<b>Grundlagen</b>	<b>12</b>
1.1	Hintergrund	12
1.2	E-PIX	14
1.3	Das Konzept von Haupt- und Nebenidentitäten	14
<b>2</b>	<b>Betrieb</b>	<b>16</b>
2.1	Funktionalitäten	16
2.1.1	Was leistet der Dienst	16
2.1.2	Was leistet der Dienst nicht	17
2.2	Installation	17
2.2.1	Systemanforderungen	17
2.2.2	Download	18
2.2.3	Starten unter Linux	20
2.2.4	Starten unter Windows	21
2.3	Update	21
2.4	Trennung von Applikations- und Datenbankserver	24

<b>3</b>	<b>Allgemein</b>	<b>26</b>
3.1	Datenquellen	26
3.2	Identifier-Domänen	26
3.3	Domänen	27
<b>4</b>	<b>Weboberfläche</b>	<b>28</b>
4.1	Anlegen einer Datenquelle	29
4.2	Anlegen einer Identifier-Domäne	29
4.3	Anlegen einer Domäne	29
4.3.1	Einstellungen	30
4.3.2	Personenfelder	32
4.3.3	Validatoren	32
4.3.4	Vorverarbeitung	34
4.3.5	Matching	35
4.3.6	Privatsphäre	38
<b>5</b>	<b>SOAP-Schnittstelle</b>	<b>42</b>
5.1	Senden von Nachrichten an den E-PIX	42
5.1.1	SOAP-UI	43
5.1.2	Postman	44
5.1.3	PowerShell (Windows)	45
5.2	Anlegen einer Datenquelle	46
5.3	Anlegen einer Identifier-Domäne	46
5.4	Anlegen einer Domäne	47
<b>6</b>	<b>XML-Konfiguration</b>	<b>49</b>
6.1	Match Modus	50
6.2	MPI Generator	51
6.3	MPI Präfix	51
6.4	Benachrichtigungen	51
6.5	Speicher-Reduktion	53
6.6	Aktualisierungsverhalten	53
6.7	Speicher-Modus	54

<b>6.8</b>	<b>Pflichtfelder</b>	55
<b>6.9</b>	<b>Zusatzfelder</b>	55
<b>6.10</b>	<b>Validatoren</b>	56
<b>6.11</b>	<b>Dublettenauflösungsgründe</b>	59
<b>6.12</b>	<b>Privatsphäre</b>	59
6.12.1	Bloomfilter-Konfiguration	60
<b>6.13</b>	<b>Vorverarbeitung</b>	64
6.13.1	Felder	65
6.13.2	Feldnamen	65
6.13.3	Einfache Transformationen	65
6.13.4	Komplexe Transformationen	66
6.13.5	Filter	67
<b>6.14</b>	<b>Matching</b>	67
6.14.1	Schwellwert für mögliche Matches	68
6.14.2	Schwellwert für automatische Matches	68
6.14.3	CEMFIM	69
6.14.4	Paralleles Record Linkage	70
6.14.5	Multithreading	70
6.14.6	Matching-Feld	70
6.14.7	Multiple-Value Feld	74

<b>7</b>	<b>Anwendungsbeispiele</b>	77
<b>7.1</b>	<b>Standardkonfiguration</b>	78
<b>7.2</b>	<b>Krebsregister</b>	80
<b>7.3</b>	<b>Matching mit Krankenversicherungsnummern</b>	81
<b>7.4</b>	<b>Privacy-Preserving Record Linkage</b>	82
7.4.1	Bloomfilter erzeugen	82
7.4.2	Bloomfilter abgleichen	83

### III

### Bedienung

<b>8</b>	<b>Weboberfläche</b>	86
<b>8.1</b>	<b>Registrierung einer Person</b>	86
<b>8.2</b>	<b>Suchen anhand von Personendaten</b>	89
<b>8.3</b>	<b>Einsehen von Details zu einer Person</b>	90
<b>8.4</b>	<b>Bearbeiten und Löschen von Personendaten</b>	91

8.5	Dublettenauflösung .....	93
8.6	Daten exportieren .....	95
8.7	Daten importieren .....	96
8.8	Einsehen von Protokollen .....	98
8.9	Statistiken einsehen .....	99
<b>9</b>	<b>SOAP-Schnittstelle .....</b>	<b>101</b>
9.1	Registrierung einer Person .....	101
9.1.1	Aktualisieren der Hauptidentität .....	105
9.1.2	Beeinflussung der Persistierung .....	105
9.2	Suchen anhand von Personendaten .....	106
9.3	Suchen anhand von Identifiern .....	107
9.4	Nachträgliches Erzeugen von Bloomfiltern .....	109
9.5	Ändern von Personendaten .....	112

## IV

## Integration

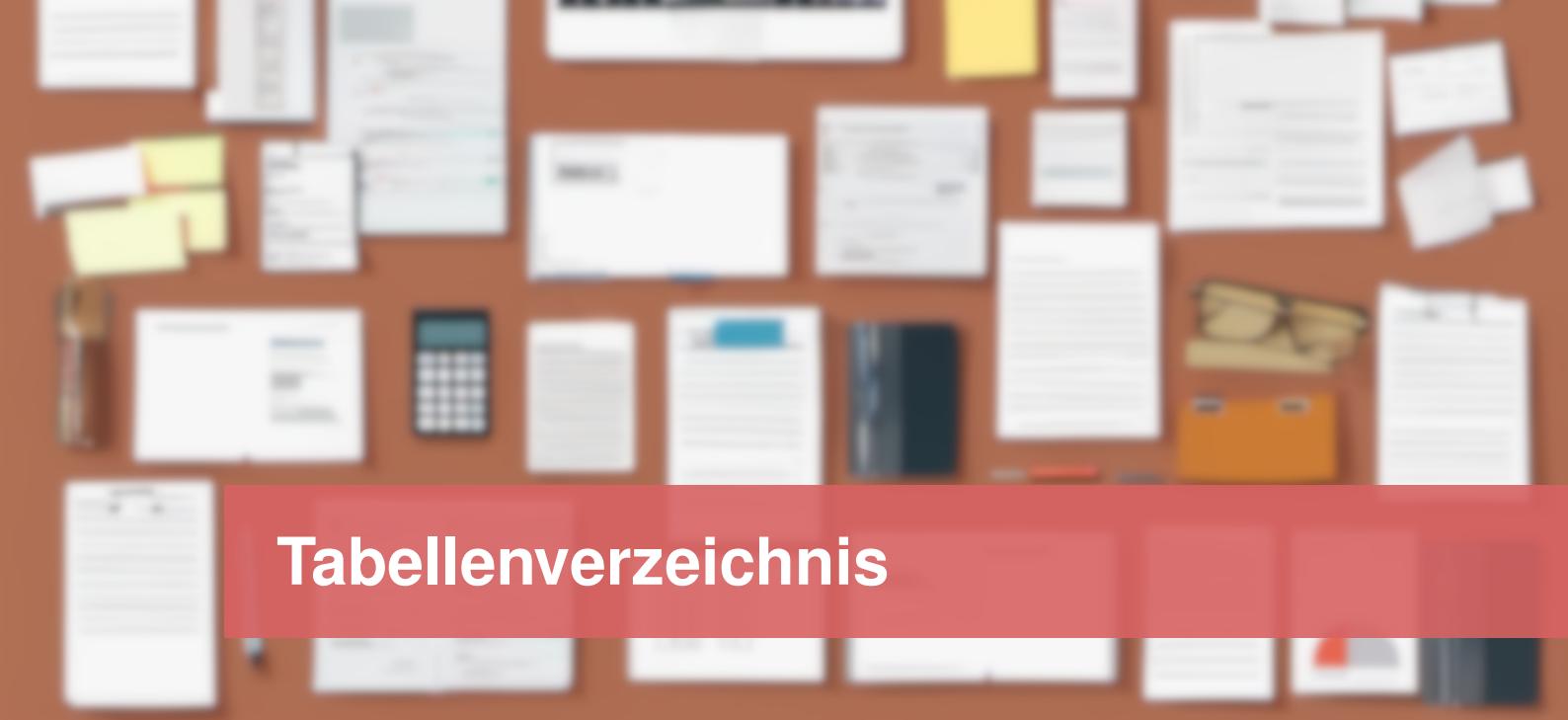
<b>10</b>	<b>Logging .....</b>	<b>115</b>
<b>11</b>	<b>Benachrichtigungen .....</b>	<b>116</b>
<b>12</b>	<b>FHIR-Unterstützung .....</b>	<b>117</b>
<b>13</b>	<b>Authentifizierung &amp; Autorisierung .....</b>	<b>119</b>
13.1	Global .....	119
13.1.1	Übersicht Nutzerrollen und Rechte .....	120
13.1.2	Verwendung von KeyCloak .....	120
13.1.3	Verwendung von gRAS .....	120
13.2	Domänen-spezifische Rollen mit OpenID-Connect .....	120
<b>14</b>	<b>Empfehlungen zur Absicherung .....</b>	<b>122</b>
<b>15</b>	<b>Optimierungen .....</b>	<b>123</b>
15.1	Optimierungen bei Multi-Millionen Beständen .....	123
15.2	Optimierungen bei Betrieb ohne Docker .....	124
15.2.1	Speicher für MySQL erhöhen .....	124

15.2.2 Batch-Writing .....	124
15.2.3 Lange Zeiten zum Hochfahren des Applikationsservers .....	124
<b>Weitere Literatur .....</b>	<b>126</b>
<b>Publikationen .....</b>	<b>126</b>
<b>Glossar .....</b>	<b>127</b>
<b>Abkürzungsverzeichnis .....</b>	<b>132</b>



# Abbildungsverzeichnis

1.1	Anwendungsfall Patientenregistrierung	13
2.1	E-PIX Docker-Architektur	19
6.1	XML-Struktur der Konfiguration	49
8.1	Person hinzufügen	88
8.2	Personsuche	90
8.3	Detailseite zu einer Person	90
8.4	Person bearbeiten	92
8.5	Dublettenauflösung	94
8.6	Export	95
8.7	Import	96
8.8	Import Vorschau	96
8.9	Protokoll	99
8.10	Dashboard	100



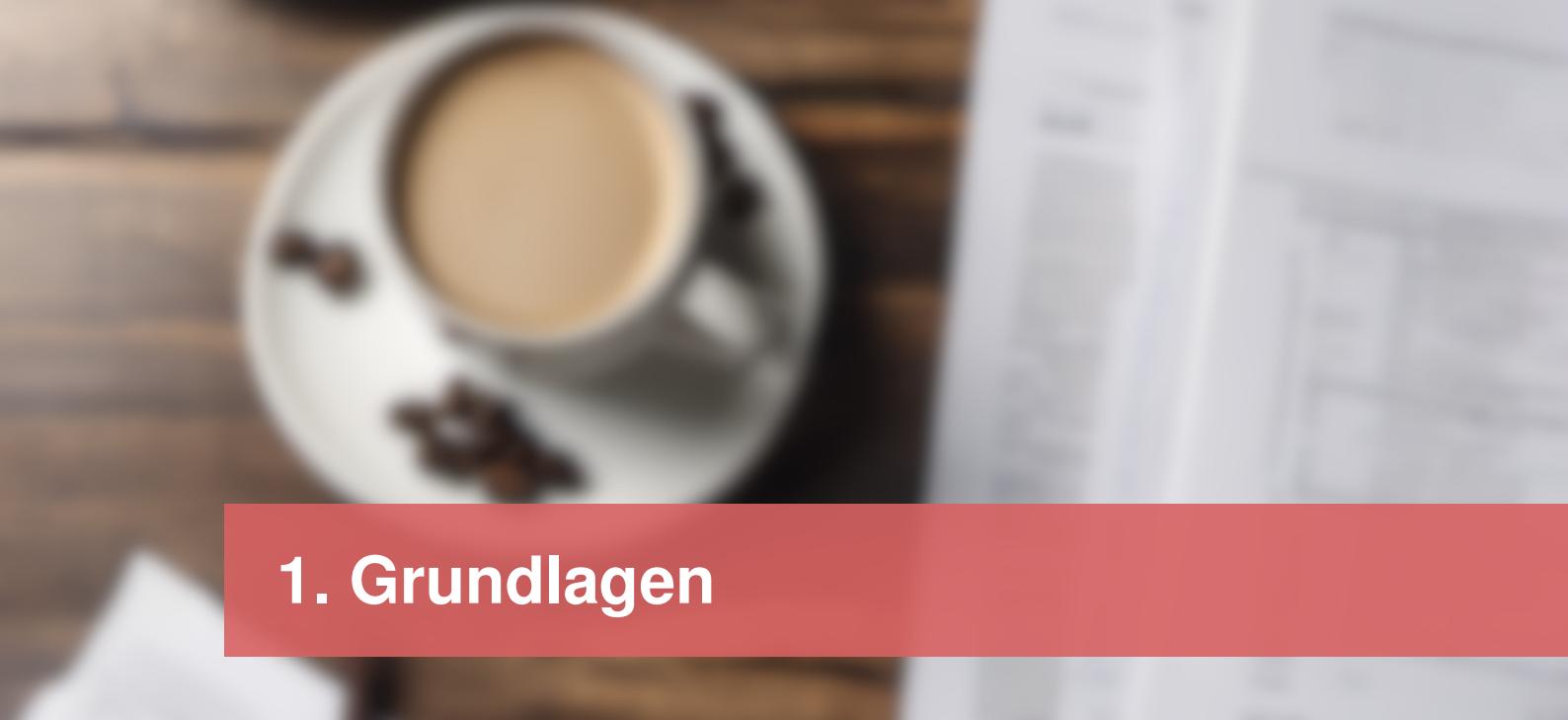
# Tabellenverzeichnis

2.1	Unterstützte DBMS vom E-PIX und Notification-Service. . . . .	18
6.1	Unterstützte Matching-Modes. . . . .	50
6.2	Unterstützte Benachrichtigungen im E-PIX. . . . .	52
6.3	Unterstützte Werte für <code>identity-update-behaviour</code> . . . . .	54
6.4	Operatoren, um Validator-Gruppen miteinander zu verknüpfen. . . . .	57
6.5	Unterstützte Validatoren mit den erforderlichen Parametern. . . . .	57
6.6	Elemente der Bloomfilter-Konfiguration. . . . .	60
6.7	Unterstützte Algorithmen zur Generierung von Bloomfiltern. . . . .	63
6.8	Unterstützte Transformationen für <code>complex-transformation-type</code> . . . . .	66
6.9	Schwellwerte für einen Automatischen Match und einen Möglichen Match. . . . .	68
6.10	Verhalten des E-PIX, je nachdem wie das Element <code>use-cemfim</code> definiert wurde. . . . .	69
6.11	Unterstützte Algorithmen für das Matching. . . . .	73
7.1	Felder, Schwellwerte und Wichtungen der Standardkonfiguration. . . . .	79
7.2	Schwellwerte für automatische und mögliche Matches. . . . .	79
7.3	Verwendete Felder mit Schwellwerten und Wichtungen im Krebsregister MV. . . . .	80
7.4	Schwellwerte für automatische und mögliche Matches im Krebsregister MV. . . . .	80
7.5	Matching mit KVNR mit Schwellwert und Wichtung. . . . .	82
8.1	Match-Typen, die Ergebnis vom Record Linkage sein können. . . . .	88
9.1	Alle im E-PIX definierten Felder. . . . .	102
9.2	Verhalten des E-PIX, je nachdem welche Save-Action gewählt wurde. . . . .	106
9.3	Methoden zum Abrufen von Personen anhand von Identifiern. . . . .	108
9.4	Unterstützte Werte für <code>updateBehaviour</code> . . . . .	113

13.1 Nutzer-Zugriffsrechte in der Weboberfläche. . . . .	120
--	-----

# Einführung

<b>1</b>	<b>Grundlagen</b>	<b>12</b>
1.1	Hintergrund	12
1.2	E-PIX	14
1.3	Das Konzept von Haupt- und Nebenidentitäten	14
<b>2</b>	<b>Betrieb</b>	<b>16</b>
2.1	Funktionalitäten	16
2.2	Installation	17
2.3	Update	21
2.4	Trennung von Applikations- und Datenbankserver	24



# 1. Grundlagen

## 1.1 Hintergrund

Um beispielsweise **Medizinische Daten (MDAT)** einer Person eindeutig zuordnen zu können, verwenden Einrichtungen wie Kliniken oder Register typischerweise lokal eindeutige Kennungen (sog. **Lokaler Identifier**). Diese Kennungen haben jedoch nur innerhalb der jeweiligen Domäne (z.B. Klinik) Gültigkeit. Zudem können **Identifizierende Daten (IDAT)** einer Person, wie Name und Geburtsdatum, aus verschiedenen Quellen aufgrund von Schreibfehlern oder zwischenzeitlichen Änderungen voneinander abweichen, so dass eine Zusammenführung von Daten (**Record Linkage**) gegebenenfalls nicht erfolgen kann. In diesem Fall spricht man von einem **Synonymfehler**. Derartige Fehler sind in der Regel nur unter Zuhilfenahme weiterer Daten auflösbar. Werden Daten verschiedener Personen fälschlicherweise einer einzigen Person zugeordnet, entsteht ein **Homonymfehler**. Diese Fehlerform ist fatal und im Nachgang nur mit sehr hohem Aufwand korrigierbar.

Um Forschungsdaten aus mehreren Projekten und Studien zusammenführen und einer einzigen Person zuordnen zu können, ist sowohl ein **Record Linkage** als auch eine eindeutige systemweite Kennung erforderlich, der sowohl die **IDAT** einer Person, als auch die einzelnen lokalen Kennungen des Quellsystems (z.B. Labore, Studienzentralen, etc.) zugeordnet sind. Da dies auch bei unvollständigen oder fehlerhaften Personendaten fehlertolerant und nachvollziehbar erfolgen muss, ist ein nachhaltiges ID-Management erforderlich.

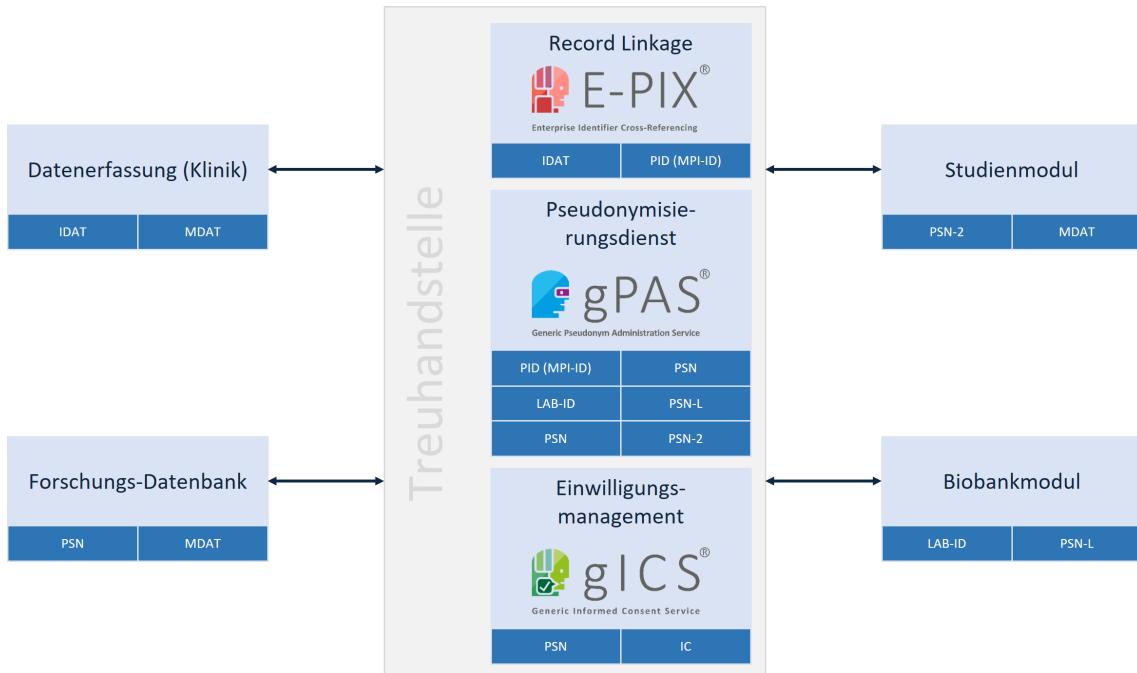


Abbildung 1.1: Das Identitätsdatenmanagement stellt eine zentrale Komponente im medizinischen Forschungskontext dar. Verschiedene Module verwalten modulspezifische Daten und ordnen diese Personen mittels spezifischen Pseudonymen zu. Die Abbildung ist adaptiert vom Maximalmodell des Generischen Datenschutzkonzepts der TMF.

Zweck des ID-Managements ist es, Personendaten unter Vermeidung von **Homonymfehlern** sicher bereits vorhandenen Datensätzen zuzuordnen und potentielle Dubletten zu erkennen und zusammen zu führen. Ergebnis dieser Zuordnung ist eine systemübergreifende eindeutige Kennung. Diese stellt gemäß den Konzepten<sup>1</sup> der TMF ein **Pseudonym** erster Stufe dar (Quelle: TMF 2004, [https://www.tmf-ev.de/Themen/Projekte/V015\\_01\\_PID\\_Generator.aspx](https://www.tmf-ev.de/Themen/Projekte/V015_01_PID_Generator.aspx), Stand: 07. Dezember 2015).

In der Abteilung Versorgungsepidemiologie und Community Health des Instituts für Community Medicine der Universitätsmedizin Greifswald wurde hierfür der Webservice **E-PIX** entwickelt. Der **E-PIX** ist als Open Source Software lizenziert (AGPLv3) und kostenfrei für kommerzielle und nicht-kommerzielle Zwecke einsetzbar.

<sup>1</sup> POMMERENING, Klaus; HELBING, Krister; GANSLANDT, Thomas; DREPPE, Johannes: Leitfaden zum Datenschutz in medizinischen Forschungsprojekten. Berlin: Medizinisch Wissenschaftliche Verlagsgesellschaft mbH & Co. KG, 2014. - ISBN 978-3-95466-123-7

## 1.2 E-PIX

Der **E-PIX** setzt das Konzept eines **Master Patient Index (MPI)** um und stellt die notwendige technische Funktionalität zur eindeutigen Identifizierung von Personen in Form eines Webservices bereit. Frei konfigurierbare Personenattribute, typischerweise Vorname, Nachname, Geburtsdatum, Geschlecht, sind Grundlage für die probabilistischen Verfahren zur Zusammenführung von Datensätzen.

Zur Dublettenerkennung wird ein Algorithmus nach Fellegi-Sunter verwendet. Für den Vergleich von Attributen stehen mehrere Vergleichsfunktionen zur Verfügung. Standardmäßig kommt die Levenshtein-Distanz zum Einsatz. Auf diese Weise kann die Zuordnung von Person und eindeutiger systemübergreifender Kennung auch bei unvollständigen bzw. fehlerhaften demografischen Informationen korrekt erfolgen.

Der **E-PIX** unterstützt neben den erwähnten Vergleichsfunktionen auf Basis von Personendaten im Klartext auch ein **Privacy-Preserving Record Linkage (PPRL)**. Hierbei werden Personendaten derart codiert, sodass keine Rückschlüsse mehr auf die eigentliche Person gezogen werden kann, jedoch dennoch auf Basis dieser codierten Daten vergleiche durchgeführt werden können.

Der **E-PIX** ermöglicht außerdem die Speicherung domänenpezifischer **Lokaler Identifier** und standardisierter IHE-Profile (PIX, PDQ). Zudem setzt der **E-PIX** das Konzept multipler **Identitäten** um, d.h. einer real existierenden Person können mehrere Ausprägungen (ähnlicher) demografischer Daten zugeordnet sein. Darüber hinaus wird die Auflösung von **Synonymfehlern** (s. Abschnitt 4) unterstützt.

## 1.3 Das Konzept von Haupt- und Nebenidentitäten

Vor allem bei epidemiologischen Kohortenstudien ist es oftmals erforderlich, die Variationen von **IDAT** beispielsweise in Bezug auf die (möglicherweise fehlerhafte) Schreibweise eines Namens (z.B.: Müller, Mueller, Muller, Mülller, etc.) im jeweiligen Quellsystem zu erhalten und dennoch die Datensätze eindeutig einer real existierenden Person fehlerfrei zuordnen zu können.

Innerhalb des **E-PIX** kann eine Person daher mehrere (Personen-)Identitäten besitzen, wovon nur eine als **Hauptidentität** (auch als **Referenzidentität** bezeichnet) deklariert werden kann. Die **Hauptidentität** wird als “die korrekte Ausprägung” der **IDAT** angesehen. Jede weitere Ausprägung wird als **Nebenidentität** gespeichert. Ein nachträgliches Ändern der Identitätenbeziehungen ist problemlos möglich,

sollte jedoch nur durch autorisiertes Personal und nach eingehender Recherche der Sachlage erfolgen.

Das Konzept von **Hauptidentitäten** und **Nebenidentitäten** ist in epidemiologischen Kohortenstudien von besonderer Relevanz und ist gleichzeitig Grundlage für das Beheben möglicher **Synonymfehler**.

Insbesondere bei der Verwaltung von **IDAT**, die aus mehreren Quellen stammen, in Abhängigkeit der Eingabemethode und Zeitpunkt der **IDAT** verschiedene Ausprägungen entstehen (Tippfehler, Namensänderung durch Heirat, etc.). Der **E-PIX** vereint all diese Ausprägungen zu einer Person und ermöglicht, die Person über die verschiedenen Ausprägungen zu finden. Mittels der **Hauptidentität** ist es möglich, die korrekte Ausprägung anzugeben und so bei Bedarf andere Systeme zu aktualisieren.



## 2. Betrieb

### 2.1 Funktionalitäten

#### 2.1.1 Was leistet der Dienst

- Erstellung und Verwaltung einer systemweit eindeutigen Kennung mittels Indexgenerator nach dem Konzept des **MPI**
- Zusammenführung von Personendaten aus unterschiedlichen Quellsystemen anhand demographischer Informationen
- Umgang mit fehlerhaften/unvollständigen Personendaten
- Unterstützung bei der Rekontakteierung durch die integrierte Personenverwaltung
- Unterstützung beim Auflösen bei **Möglichen Matches** durch das Konzept von **Hauptidentitäten** und **Nebenidentitäten** (siehe Abschnitt 1.3)
- Unterstützung der IHE-Profile PIX & PDQ (PIX ist derzeit noch ohne Update Notification)
- Protokollierung von Systemprozessen und (kritischen) Systementscheidungen
- Beschleunigtes Matching durch Caching: die für den Matching-Prozess erforderliche Datenbasis wird vollständig im Zwischenspeicher gehalten und erlaubt beispielsweise Antwortzeiten beim Anlegen oder Aktualisieren einer Person und einem Datenbestand von bereits 1.000.000 Personen in deutlich weniger als 1 Sekunde
- Einfache Bedienung durch eine intuitive grafische Oberfläche
- Versenden von Notifications bei Zustandsänderungen, um andere Systeme zu informieren

## 2.1.2 Was leistet der Dienst nicht

- Eine automatisierte Transkription und Transliteration von demografischen Informationen sind nicht möglich. Diese erfolgt im Bedarfsfall vor der Eintragung in den **E-PIX**.
- Die Vergabe von **Pseudonymen** zweiter und weiterer Stufen findet nicht im **E-PIX** statt, sondern kann in Kombination mit dem **gPAS** erzielt werden.

## 2.2 Installation

Der **E-PIX** wird als standardmäßig als Docker-Container bereitgestellt. Die Verwendung von Docker wird empfohlen. Alternativ dazu kann der **E-PIX** als Servlet im Applikationsserver WildFly betrieben werden. Die Voraussetzungen hierfür sind im Abschnitt 2.2.1 aufgeführt.

### 2.2.1 Systemanforderungen

#### Technisch / Infrastruktur

- Installierte aktuelle Version von Docker<sup>1</sup> und Docker-Compose<sup>2</sup>
- Administrative Rechte
- Keine Nutzungsbeschränkungen auf die bereitgestellten Service- und Client-URLs
- Windows<sup>3</sup> oder Ubuntu Server (oder vergleichbar) mit min. 8 GB Arbeitsspeicher, 5 GB Festplattenspeicher, Prozessor (benötigter Arbeitsspeicher und Prozessor-Leistung sind abhängig von erwarteter Datenmenge und -durchsatz)

#### Anwendungs- und Datenbankserver (ohne Verwendung von Docker)

- JDK 17 oder höher
- WildFly 26 oder höher
- EclipseLink 2.7.11 oder höher
- MySQL-Connector 8 oder höher
- MySQL-Server 8 oder höher

Der **E-PIX** und der Notification-Service<sup>4</sup> unterstützen derzeit die in Tabelle 2.1 aufgeführten **DBMS**. Künftig werden auch weitere **DBMS** unterstützt.

<sup>1</sup> Weitere Informationen unter <https://docs.docker.com/install/>

<sup>2</sup> Weitere Informationen unter <https://docs.docker.com/compose/install/>

<sup>3</sup> Beim Betrieb unter Windows ist zu beachten, dass bei der Verwendung von Volumes und parallel betriebenen VPN-Clients Probleme auftreten können.

<sup>4</sup> Der Notification-Service wird für Benachrichtigungen an externe Systeme benötigt. Weitere Informationen dazu im Kapitel 11.

Tabelle 2.1: Unterstützte DBMS vom E-PIX und Notification-Service.

DBMS	E-PIX	Notification-Service
MySQL	×	×
MariaDB	×	×

## Personell

- Mitarbeiter mit grundlegenden IT-Kenntnissen zur Administration des Servers und zur Einrichtung des E-PIX-Dienstes (zuzüglich der Wartung und regelmäßiger Sicherungen der E-PIX-Datenbank)
- Ein autorisierter Verantwortlicher zur Administration der E-PIX-Inhalte inkl. zur Auflösung bei Möglichen Matches nach ausführlicher Prüfung der individuellen Sachlage

### 2.2.2 Download

Um den E-PIX als Docker-Container zu starten, werden die Programme Docker und Docker-Compose benötigt. Beide Programme müssen hierfür installiert sein. Da zwischen beiden Programmen Inkompatibilitäten auftreten können, wird empfohlen die jeweils aktuellsten Versionen zu installieren.

Der E-PIX benötigt zur Ausführung zwei Container (vgl. Abbildung 2.1). Damit diese nicht einzeln gestartet und entsprechend zusammengeschaltet werden müssen, wird der Dienst mit Docker-Compose gestartet. Die entsprechenden Ressourcen können von der THS-Webseite<sup>5</sup> heruntergeladen werden.

<sup>5</sup> <https://www.ths-greifswald.de/forscher/e-pix/>

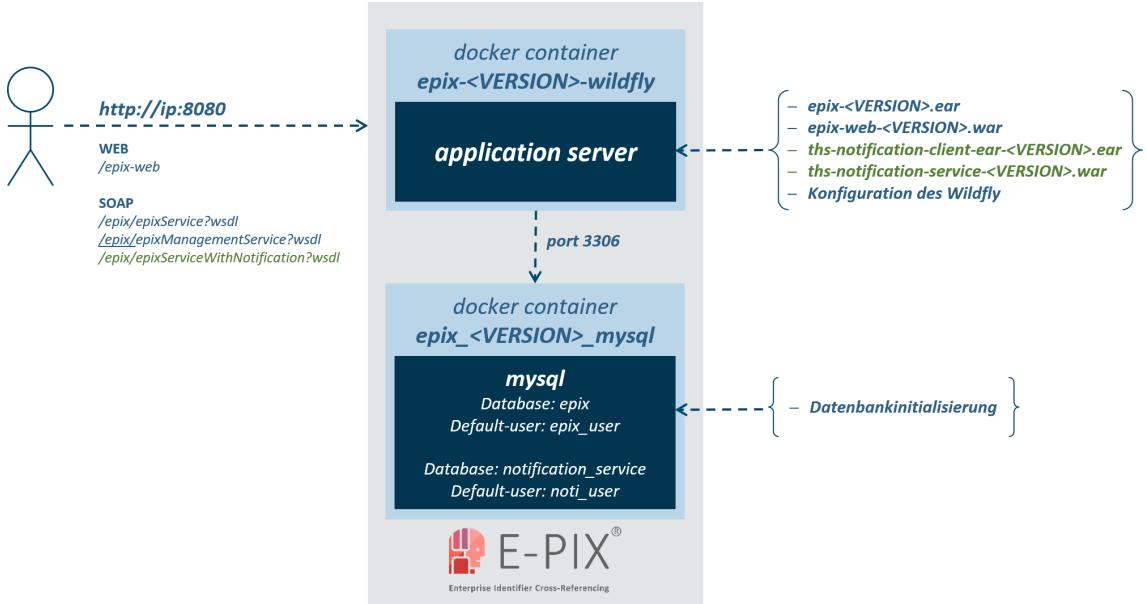


Abbildung 2.1: Architektur des **E-PIX** mit Docker.

Das Docker-System besteht aus zwei getrennten Containern. Zum einen aus einer Datenbankinstanz (MySQL) und zum anderen aus dem Anwendungsserver (WildFly inkl. Datenbank-Konnektoren). Der Anwendungsserver kommuniziert mit dem MySQL-Server über den Port 3306. Der Zugriff auf das System von “außen” erfolgt über den Web-Browser. Die Inhalte werden über den Port 8080 (**E-PIX**) für den Anwender bereitgestellt.

**Hinweis:** Weitere Details zur Nutzung von Docker-Compose und **E-PIX** sind der beigelegten Beschreibung `docker-compose/README_E-PIX.md` zu entnehmen.

**Hinweis:** Für einen **Produktivbetrieb** sollte die `docker-compose.yml` angepasst werden. Hierzu sollte der Speicherpfad des MySQL-Volumes festgelegt werden. Andernfalls sind alle Daten, die im Container liegen, nach einem Herunterfahren gelöscht. Die Datenbank-Skripte prüfen selbst, ob die entsprechenden Datenbanken bereits angelegt wurden. Die Datenbanken werden bei einem Neustart daher **nicht** überschrieben.

**Hinweis:** Beachten Sie, dass beim Wechsel von **E-PIX** Versionen die Docker-Compose Komponenten stets komplett aktualisieren sollten. Dies beinhaltet die Aktualisierung von \*.yml-Dateien, CLI-Dateien und die Übernahme eventueller individueller Konfigurationen auf neue ENV-Files. Eine Übersicht aller Konfigurationsdateien, deren Zweck und aller relevanten Parameter ist der beigelegten Beschreibung `docker-compose/README_E-PIX.md` zu entnehmen. Eine ausführliche Anleitung zur Aktualisierung von produktiv genutzten Containern ist dem Produkt beigelegt (Docker-Update.md) und online verfügbar (<https://www.ths-greifswald.de/e-pix/update>).

### 2.2.3 Starten unter Linux

Um die folgenden Schritte problemlos durchführen zu können, wird ein Account mit administrativen Rechten benötigt. Exemplarisch werden die folgenden Befehle mit `sudo` ausgeführt.

#### Download der benötigten Dateien

Laden Sie die aktuellste Version von <https://www.ths-greifswald.de/forscher/e-pix/#download> herunter und entpacken Sie die ZIP-Datei. Diese enthält alle relevanten Docker-Compose-Dateien. Im Folgenden wird davon ausgegangen, dass der Ordner in das Verzeichnis `/opt/` entpackt wurde. Der Pfad kann bei Bedarf angepasst werden.

#### Vergabe von Schreibrechten

```
1 sudo chmod -R 755 /opt/compose-wildfly/
2 sudo chown -R 1111:1111 /opt/compose-wildfly/logs/\
3   /opt/composewildfly/deployments/
```

Aus Gründen von Leistung und Ausfallsicherheit sollten die Container des **E-PIX** auf einem dedizierten Server eingerichtet werden. Zur Administration werden der User `epix` (uid 1111) aus der Gruppe `users` (gid 1111) genutzt.

#### Wechseln in das E-PIX-Verzeichnis für die Standard-Version

```
1 cd /opt/compose-wildfly/
```

#### Starten des E-PIX mithilfe von Docker-Compose

```
1 sudo docker compose up
```

Damit werden die benötigten Komponenten heruntergeladen<sup>6</sup> und die Konfiguration

<sup>6</sup> Sollte Ihre Maschine keinen Zugang zum Internet haben, können die benötigten Images

on von MySQL und WildFly gestartet. Danach wird die aktuelle Version des **E-PIX** bereitgestellt. Der Installationsvorgang kann in Abhängigkeit der vorhandenen Internetverbindung etwa 5 Minuten dauern. Der erfolgreiche Start des Dienstes wird mit der folgenden Ausgabe abgeschlossen.

```
Wildfly 26.1.2.Final [...] started in ...
```

## 2.2.4 Starten unter Windows

Zur Installation und Starten des **E-PIX** ist ein Benutzeraccount mit Adminrechten erforderlich.

Entpacken Sie das Archiv an der gewünschten Stelle. Danach kann der **E-PIX** über Docker-Compose gestartet werden. Die notwendigen Schritte hierzu auf einem Windows-System sind im Folgendem beschrieben.

Starten Sie die Windows Console CMD mit Adminrechten und wechseln Sie in das gewählte Verzeichnis (enthält die Datei `docker-compose.yml`). Damit der **E-PIX** bei Windows problemlos gestartet werden kann, setzen Sie in der Datei `envs/ttp_commons.env` den Parameter `#WF_MARKERFILES = AUTO` auf `FALSE` und entfernen Sie die vorangehende Raute (#)<sup>7</sup>.

Anhand des folgenden Befehls können Sie über die Konsole den **E-PIX** über Docker-Compose starten:

```
1 docker compose up
```

Das Starten der Software kann wenige Minuten in Anspruch nehmen. **E-PIX** wurde erfolgreich installiert, wenn Ihnen folgender Befehl angezeigt wird:

```
Wildfly 26.1.2.Final [...] started in ...
```

## 2.3 Update

Am folgenden Beispiel wird die Aktualisierung der Docker-Container vom **E-PIX** gezeigt.

Im Beispiel wird die bestehende und laufende Instanz vom **E-PIX** als `<epix-old>` bezeichnet. Die existierende Version (`<old-version>`) soll gesichert und ein Update auf eine neue Version vom **E-PIX** (`<epix-new>`, `<new-version>`) durchgeführt werden, ohne die bereits vorhandenen Daten in der MySQL-Datenbank zu verän-

---

(MySQL und WildFly) von einer anderen Maschine heruntergeladen werden und dann auf Ihr Zielsystem kopiert werden (siehe [https://docs.docker.com/engine/reference/commandline/image\\_save/](https://docs.docker.com/engine/reference/commandline/image_save/) und <https://docs.docker.com/engine/reference/commandline/load/>).

<sup>7</sup> Dies muss mittlerweile normalerweise nicht mehr händisch durchgeführt werden. Bei Problemen kann diese Option jedoch händisch angepasst werden.

dern.

Ob die Instanzen vom **E-PIX** laufen, kann mit folgenden Befehl geprüft werden:

```
1 sudo docker ps -a
```

### Neue Tool-Version von der THS-Webseite herunterladen

Die aktuelle Version von <https://www.ths-greifswald.de/forscher/e-pix/> herunterladen und entpacken, sowie auf das Host-System kopieren und sicherstellen, dass entsprechende Berechtigungen zum Ausführen der Dateien gesetzt sind.

```
1 sudo chmod -R 755 /PFAD
```

### Sichern der aktuellen Docker-Konfiguration

Um auf dem Host-System den derzeitigen Stand der **E-PIX**-Konfiguration (WildFly-Skripte, etc.) zu sichern, den entsprechenden Ordner per TAR-Archiv sichern:

```
1 tar czf backup-epix-2022-03-31.tgz <epix-old>/
```

### Sichern der existierenden Datenbank

Um zusätzlich die Sicherung der existierenden Datenbank durchzuführen, wird ein MySQL-Dump über die Docker-Konsole angestoßen und die resultierende Export-Datei im Dateisystem vom Host abgelegt.

```
1 sudo docker exec epix-<old-version>-mysql \
2     /usr/bin/mysqldump -u epix_user -p epix \
3     > backup-epix-<old-version>-2022-03-31.sql
```

Der Name der bestehenden MySQL-Instanz muss entsprechend angepasst werden.

### Aktualisieren der Datenbank

Für alle Versionen sind die Datenbank-Aktualisierungsskripte jeweils im Docker-Verzeichnis unter `<epix-new>/update_scripts` zu finden. Die Update-Skripte müssen in den Docker-Container kopiert werden, wobei nur die Skripte erforderlich sind, welche die Version zwischen `<epix-old>` zu `<epix-new>` betreffen.

```
1 sudo docker cp <epix-new>/update_scripts/ \
2     epix-<old-version>-mysql:/update-files/
```

Je nachdem von welcher Version aus **E-PIX** aktualisiert werden soll, müssen die relevanten SQL-Skripte chronologisch durchlaufen werden.

**Beispiel:** Für ein Update von Version 2.11.0 auf 2.13.0 sind demzufolge die Skripte `update_database_epix_2.11.x-2.12.x.sql` und `update_database_epix_2.12.x-2.13.x.sql` auszuführen.

Hierzu muss per MySQL Client eine Verbindung mit der bestehenden Datenbank erfolgen und die Update-Skripte nacheinander durchlaufen werden. Dies kann per

Docker realisiert werden (Nutzernamen und Passwörter ggf. anpassen).

### Beispiel:

```
1 docker exec -it epix-2.11.0-mysql /usr/bin/mysql -u epix_user -p
  -e "USE epix;$(cat
  epix-new/standard/update_database_epix_2.11.x-2.12.x.sql)"
2 docker exec -it epix-2.11.0-mysql /usr/bin/mysql -u epix_user -p
  -e "USE epix;$(cat
  epix-new/standard/update_database_epix_2.12.x-2.13.x.sql)"
```

## Aktualisierung der Deployments und Wildfly-Konfiguration

Den Datenbank-Container herunterfahren:

```
1 docker epix-<old-version>-mysql down
```

Die Deployments im <epix-old> Verzeichnis auf dem Host-System löschen und die neuen Deployments hinein kopieren:

```
1 rm -f <epix-old>/deployments/*
2 cp -R <epix-new>/deployments/ <epix-old>/deployments/
```

Aktualisierung der Bezeichnung des MySQL Containers:

```
1 sudo docker rename epix-<old-version>-mysql \
  epix-<new-version>-mysql
```

JBOSS Konfiguration aktualisieren:

```
1 cp -R <epix-new>/jboss/ <epix-old>/jboss/
```

Docker-Compose-Konfiguration aktualisieren:

```
1 cp -R <epix-new>/docker-compose.yml <epix-old>/docker-compose.yml
```

Anpassen des Eigentümer-Benutzers:

```
1 chown 999 <epix-new>/sqls
2 chown 1111 <epix-new>/deployments
3 chown 1111 <epix-new>/logs
4 chown 1111 <epix-new>/jboss
```

## Starten des aktualisierten Containers

Den aktualisierten Container mittels folgendem Befehl starten (-d um Container im Hintergrund zu starten):

```
1 docker compose up -d
```

Den Erfolg der Aktualisierung prüfen durch Aufruf des Web-Frontends unter <http://IPADDRESS:8080/epix-web>.

## Im Fehlerfall: Wiederherstellung der Datenbank

Im Fehlerfall, kann die bisherige Datenbank wiederhergestellt werden (sofern die Anleitung befolgt wurde). Nutzernamen und Passwort ggf. anpassen.

```
1 docker exec -it epix-<new-version>-mysql /usr/bin/mysql -u  
epix_user -p -e "USE epix;$(cat backup-epix-2022-03-31.sql)"
```

**Hinweis:** Bei großen Beständen kann es passieren, dass der **E-PIX** nach einem Update nicht mehr hochfährt. Daher kann es erforderlich sein, die Konfiguration für den WildFly anzupassen. Hierzu werden in der `./envs/wf_commons.env` die Variablen `WF_BLOCKING_TIMEOUT` und `WF_TRANSACTION_TIMEOUT` entsprechend verändert. Die Werte müssen dabei heraufgesetzt werden<sup>a</sup>.

<sup>a</sup> Eine detaillierte Beschreibung aller Variablen ist zu finden unter: <https://hub.docker.com/r/mosaicgreifswald/wildfly/>

## 2.4 Trennung von Applikations- und Datenbankserver

Der **E-PIX** bzw. der Applikationsserver (Wildfly) kann separat vom Datenbankserver (MySQL) betrieben werden. Dies kann beispielsweise erwünscht sein, wenn der Datenbankserver auf einem anderen System betrieben werden soll. Standardmäßig werden die THS-Werkzeuge als Docker bereitgestellt. Im Folgenden sind die Anpassungen erläutert, um auf dieser Basis eine Trennung von Applikationsserver und Datenbankserver zu erreichen.

Zunächst muss die beiliegende `docker-compose.yml` angepasst werden. Diese besteht aus den zwei *Services* "mysql" und "wildfly". Der "mysql"-Teil beginnend mit `mysql:` muss entfernt werden. Im "wildfly"-Teil müssen folgende Anpassungen vorgenommen werden:

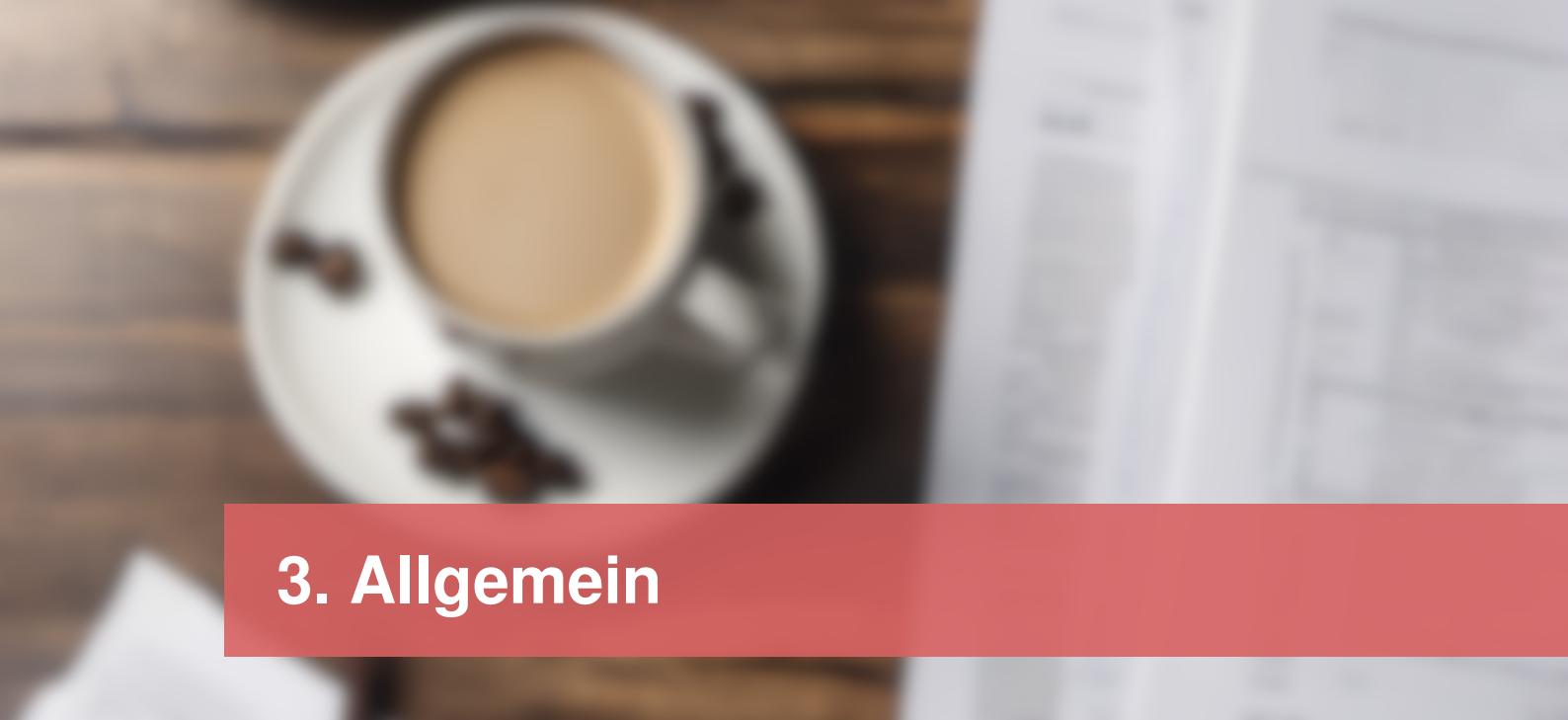
- Die Zeilen `depends_on:` und `- mysql` müssen entfernt werden.
- Die Zeile `entrypoint: /bin/bash` muss entfernt werden.
- Die Zeile `command: ...` muss entfernt werden.

Mit diesen Anpassungen wird kein MySQL-Server im Docker-Compose hochgefahren und der Wildfly-Server wartet entsprechend nicht mehr darauf, dass ein MySQL-Server hochgefahren wird. Dies hat zur Folge, dass der MySQL-Server im Vorfeld gestartet werden muss.

Die Verbindung zum MySQL-Server wird in der `/envs/tp_epix.env` definiert. In der Datei müssen folgende Variablen auskommentiert werden (# entfernen) und entsprechend angepasst werden: `TTP_EPIX_DB_HOST`, `TTP_EPIX_DB_PORT`, `TTP_EPIX_DB_NAME`, `TTP_EPIX_DB_USER` und `TTP_EPIX_DB_PASS`.

# Konfiguration

<b>3</b>	<b>Allgemein</b>	<b>26</b>
3.1	Datenquellen	26
3.2	Identifier-Domänen	26
3.3	Domänen	27
<b>4</b>	<b>Weboberfläche</b>	<b>28</b>
4.1	Anlegen einer Datenquelle	29
4.2	Anlegen einer Identifier-Domäne	29
4.3	Anlegen einer Domäne	29
<b>5</b>	<b>SOAP-Schnittstelle</b>	<b>42</b>
5.1	Senden von Nachrichten an den E-PIX	42
5.2	Anlegen einer Datenquelle	46
5.3	Anlegen einer Identifier-Domäne	46
5.4	Anlegen einer Domäne	47
<b>6</b>	<b>XML-Konfiguration</b>	<b>49</b>
6.1	Match Modus	50
6.2	MPI Generator	51
6.3	MPI Präfix	51
6.4	Benachrichtigungen	51
6.5	Speicher-Reduktion	53
6.6	Aktualisierungsverhalten	53
6.7	Speicher-Modus	54
6.8	Pflichtfelder	55
6.9	Zusatzfelder	55
6.10	Validatoren	56
6.11	Dublettenauflösungsgründe	59
6.12	Privatsphäre	59
6.13	Vorverarbeitung	64
6.14	Matching	67
<b>7</b>	<b>Anwendungsbeispiele</b>	<b>77</b>
7.1	Standardkonfiguration	78
7.2	Krebsregister	80
7.3	Matching mit Krankenversicherungsnummern	81
7.4	Privacy-Preserving Record Linkage	82



## 3. Allgemein

### 3.1 Datenquellen

Eine **Datenquelle** gibt an, woher die später registrierten Personendaten stammen. Je nachdem wo der **E-PIX** betrieben wird, kann dies eine bestimmte Studie, ein Forschungsnetzwerk oder ein konkretes System wie einem **Krankenhaus-informationssystem (KIS)** sein. Beim Anlegen einer **Domäne** (Abschnitt 4.3 per Weboberfläche oder Abschnitt 5.4 per SOAP-Schnittstelle) wird eine **Sichere Datenquelle** definiert. Diese gibt an, woher die **Hauptidentität** einer Person stammt. Die **Datenquelle** kann bei einer Personenregistrierung über die Weboberfläche aus der Liste der zuvor angelegten Einträge ausgewählt werden (Abschnitt 8.1) oder wird über die SOAP-Schnittstelle in der Anfrage angegeben (Abschnitt 9.1). Entspricht die angegebene **Datenquelle** nicht der **Sichere Datenquelle**, so werden im Fall abweichender **IDAT**, diese der Person als **Nebenidentität** angefügt. Die **Datenquelle** hat daher Einfluss darauf, ob eine **Identität** als **Hauptidentität** oder als **Nebenidentität** hinterlegt wird. Weitere Informationen zu diesem Konzept, sind in Abschnitt 1.3 zu finden.

### 3.2 Identifier-Domänen

In einer **Identifier-Domäne** werden alle **Identifier** zu einem Kontext gespeichert. Dies umfasst zum einen **MPIs**, die der **E-PIX** automatisch für Personen erzeugt, als auch **Identifier**, die von externen Systemen vergeben wurden und im **E-PIX** hinterlegt werden. Letzteres umfasst zum Beispiel Fallnummern oder Patienten-IDs. Jede **Identifier-Domäne** erhält einen eindeutigen Namen und einen eindeutigen **Objekt-Identifikator (OID)**. Jede Forschungseinrichtung besitzt typischerweise einen **OID**, welcher hier angegeben werden kann. Für andere **Quellen** wie ein **KIS**,

eine Studie etc., kann der **OID** frei gewählt werden. Wird kein **OID** angegeben, erzeugt der **E-PIX** automatisch eine eindeutige Kennung. Im **E-PIX** ist standardmäßig bereits eine **Identifier-Domäne** für einen **MPIs** angelegt. Diese kann beim Anlegen einer **Domäne** als **Identifier-Domäne** angegeben werden. Der **E-PIX** erzeugt in dieser **Identifier-Domäne** bei einer späteren Personenregistrierung die eindeutigen Kennungen. Dieselbe **Identifier-Domäne** kann für mehrere **Domänen** eingetragen werden. Dabei werden dann **Domänen**-übergreifend eindeutige Kennungen vergeben. Soll für jede **Domäne** eine eigene **Identifier-Domäne** verwendet werden, so muss für jede **Domäne** zunächst eine **Identifier-Domäne** angelegt werden und bei der Konfiguration als **Identifier-Domäne** angegeben werden. Dabei ist zu beachten, dass **MPIs** im **E-PIX** immer eindeutig sein müssen. Es ist daher erforderlich, dass in den **Domäne** verschiedene Präfixe (siehe Abschnitt 4.3.1) angegeben werden. Dies ist nicht erforderlich, wenn eine übergreifende **Identifier-Domäne** für die **MPIs** genutzt wird.

**Hinweis:** Für **Identifier** aus externen Systemen, sollte eine separate **Identifier-Domäne** angelegt werden. Nur so ist sichergestellt, dass es keine Überschneidungen zwischen externen und vom **E-PIX** vergebenen **Identifiern** gibt.

### 3.3 Domänen

Eine **Domäne** stellt den Kontext dar, in dem das **Record Linkage** ausgeführt wird. Eine **Domäne** kann eine Studie, ein Standort-übergreifendes Forschungsprojekt oder die Personenverwaltung eines Institutes oder Systems darstellen. Innerhalb vom **E-PIX** können mehrere **Domänen** verwaltet werden. Eine im **E-PIX** registrierte Person ist innerhalb einer **Domäne** immer eindeutig. Diese Person kann aber, sofern mehrere **Domänen** im **E-PIX** verwaltet werden, mehrfach registriert werden, jedoch immer nur einmal pro **Domäne**. Zu jeder Person können jedoch mehrere Ausprägungen von **IDAT** in Form von **Identitäten** vorliegen. Jede **Domäne** hat dabei eine **Sichere Datenquelle** hinterlegt (Abschnitt 3.1). Wird eine Person registriert, so wird die **Datenquelle** angegeben, von wo die **IDAT** stammen. Jeder **Domäne** kann eine eigene **Identifier-Domäne** hinterlegt werden (Abschnitt 3.2). Jede **Domäne** hat eine spezifische Konfiguration hinterlegt, die unter anderem das Verhalten vom **Record Linkage** bestimmt. Der **E-PIX** definiert Felder für die **IDAT** vor. Dabei kann festgelegt werden, welche Felder Pflichtangaben sind, ob weitere Felder definiert werden sollen und ob und wie diese für das **Record Linkage** verwendet werden sollen. Diese Einstellungen sind oft spezifisch für Projekte, da nicht immer alle Angaben vorliegen. Einige Standardfälle werden in Kapitel 7 dargestellt.

## 4. Weboberfläche

Der **E-PIX** erlaubt die Verarbeitung von Personendatensätzen mehrerer Mandanten innerhalb einer Datenbank, durch die Verwendung von **Domänen**. Die registrierten Personen sind nur innerhalb einer **Domäne** eindeutig. Ein **Record Linkage** findet demnach ebenfalls nur innerhalb einer **Domäne** statt. Um Personen registrieren zu können, muss eine entsprechende **Domäne** angelegt werden. Für jede **Domäne** müssen eine **Sichere Datenquelle** und eine **Identifier-Domäne** angegeben werden. Diese müssen vor dem Anlegen der **Domäne** im System angelegt werden. Die nötigen Schritte sind unter dem Menüpunkt **Domänen** vorzunehmen und werden im Folgenden beschrieben.

Einstellungen

Domänen				
Name	Schlüssel	Record Linkage	MPI Identifier-Domäne	Sichere Datenquelle
Demo (aktiv)	Demo	Ja	MPI	dummy_safe_source

1-1 von 1 |< << 1 >> >| Rechtsklick auf eine Zeile öffnet zusätzliche Optionen

+ Erstellen Importieren

Datenquellen	
Name	Schlüssel
dummy_safe_source	dummy_safe_source

1-1 von 1 |< << 1 >> >| Rechtsklick auf eine Zeile öffnet zusätzliche Optionen

+ Erstellen

Identifier-Domänen		
Name	Schlüssel	OID
MPI	MPI	1.2.276.0.76.3.1.132.1.1.1

1-1 von 1 |< << 1 >> >| Rechtsklick auf eine Zeile öffnet zusätzliche Optionen

+ Erstellen

## 4.1 Anlegen einer Datenquelle

Unter dem Menüpunkt *Domänen / Quellen / Identifier* können bestehende **Datenquellen** eingesehen und neue **Datenquellen** hinzugefügt werden. Mithilfe der Schaltfläche **+ Erstellen** unter der Gruppe *Datenquellen* wird ein neuer Eintrag für eine neue **Datenquelle** angelegt. Im Folgenden muss ein eindeutiger *Name* und idealer Weise eine *Beschreibung* angegeben werden. Der *Schlüssel* wird automatisch erzeugt, sofern dieser nicht explizit angegeben wurde. In der Weboberfläche wird stets der *Name* verwendet. Bei Nutzung der SOAP-Schnittstelle muss der *Schlüssel* angegeben werden. Im Gegensatz zum *Schlüssel* kann der *Name* zu einem späteren Zeitpunkt geändert werden.

## 4.2 Anlegen einer Identifier-Domäne

Unter dem Menüpunkt *Domänen / Quellen / Identifier* können bestehende **lokale Identifier** eingesehen werden. Der **E-PIX** hat standardmäßig eine **Identifier-Domäne MPI** hinterlegt. Das Anlegen weiterer **Identifier-Domänen** ist nur erforderlich, wenn **Identifier** anderer Systeme hinterlegt werden sollen. Hierzu wird unter der Gruppe *Identifier-Domänen* mit der Schaltfläche **+ Erstellen** ein neuer Eintrag angelegt. Dabei muss ein eindeutiger *Name* vergeben werden. Dieser wird später in der Weboberfläche angezeigt. Der *Schlüssel* wird automatisch generiert, sofern dieser nicht explizit angegeben wurde. Dieser wird bei der Verwendung der SOAP-Schnittstelle verwendet. Optional kann eine kurze Beschreibung der **Identifier-Domäne** angegeben werden. Außerdem kann ein **OID** angegeben werden. Wenn dieser explizit angegeben wird, muss dieser eindeutig sein. Wird kein **OID** angegeben, so erzeugt der **E-PIX** automatisch einen eindeutigen **OID**. Nach dem Anlegen der **Identifier-Domäne** kann diese beim Anlegen einer **Domäne** angegeben werden.

## 4.3 Anlegen einer Domäne

Die Konfiguration der **Domäne** kann vollständig per Weboberfläche durchgeführt werden. Alternativ kann die Konfiguration im **XML**-Format (Kapitel 6) erfolgen und über die Weboberfläche eingespielt werden. Die Konfiguration ist auch über die SOAP-Schnittstelle möglich (Abschnitt 5).

Nachdem die **Sichere Datenquelle** und die **Identifier-Domäne** angelegt wurden, kann ein neuer **Domänen**-Eintrag über die Schaltfläche **+ Erstellen** erzeugt werden. Die Konfiguration der **Domäne** erfolgt in mehreren Schritten. Hierfür können verschiedene Reiter angewählt werden und die entsprechenden Einstellungen darin vorgenommen werden. Einige Felder sind bereits entsprechend einer Standard-

Konfiguration (vgl. Abschnitt 7.1) vor ausgefüllt, die bei Bedarf angepasst werden können.

**Hinweis:** Nach der ersten Personenregistrierung in eine **Domäne**, kann die Konfiguration nur noch eingeschränkt bearbeitet werden. Andernfalls müsste der **E-PIX** alle Ergebnisse des **Record Linkages** anhand der neuen Konfiguration prüfen und ggf. zusammengeführte **Identitäten** auftrennen. Soll tatsächlich eine neue Konfiguration auf einen Bestand angewandt werden, muss eine neue **Domäne** angelegt werden und alle Datensätze der vorhandenen **Domäne** dort registriert werden.

Die Beschreibung der **Domänen**-Konfiguration mit den einzelnen Reitern (*Einstellungen* in Abschnitt 4.3.1, *Personenfelder* in Abschnitt 4.3.2, *Vorverarbeitung* in Abschnitt 4.3.4, *Matching* in Abschnitt 4.3.5, *Privatsphäre* in Abschnitt 4.3.6,) erfolgt im Folgendem.

**Info:** In der Weboberfläche sind bereits einige Einstellungen vorausgefüllt. Diese entsprechen der mitgelieferten Standardkonfiguration (siehe Abschnitt 7.1). Die Einstellungen können belassen, ergänzt oder entfernt werden. Für viele Projekte kann die Standardkonfiguration bereits zufriedenstellende Ergebnisse liefern. Werden Projekt-spezifische Parameter benötigt, können diese entsprechend ergänzt werden.

### 4.3.1 Einstellungen

Unter dem Reiter Einstellungen werden der Name, die Beschreibung, die **Sichere Datenquelle**, die **Identifier-Domäne** und weitere allgemeine Einstellungen vorgenommen.

Eine **Domäne** muss einen eindeutigen Namen aufweisen. Der **E-PIX** erzeugt anhand dessen einen Schlüssel (der wahlweise auch manuell definiert werden kann), welcher zum Ansprechen der **Domäne** über die SOAP-Schnittstelle verwendet wird. Der Name wird in der Oberfläche angezeigt und kann zu einem späteren Zeitpunkt geändert werden. Der Schlüssel hingegen kann nachträglich nicht mehr geändert werden und bleibt daher beim Ansprechen über die SOAP-Schnittstelle auch nach einer Änderung des Namens unverändert. Eine Beschreibung sollte insbesondere bei der Verarbeitung von Personen für mehrere Mandanten oder Projekte innerhalb eines **E-PIX** eingetragen werden. Die **Sichere Datenquelle** kann aus der Liste der vorhandenen Einträge ausgewählt werden. Mit Aktivierung der Checkbox *Senden Benachrichtigungen...*, benachrichtigt der **E-PIX** den Notification-Service (vgl. Kapitel 11), bei Änderungen in der Oberfläche (z.B. nach Bearbeitung

eines Personendatensatzes).

Der **E-PIX** erzeugt für jede Person einen **MPI**. Der **E-PIX** wird hierfür mit einem entsprechenden Generator (*EAN13Generator*) ausgeliefert. Soll der **MPI** ein anderes Format aufweisen, können eigene Generatoren implementiert werden. Das Präfix gibt dabei an, ob und welche Zeichenkette einem **MPI** vorangestellt wird (Standardmäßig: 1001). Das Präfix darf dabei nur Zahlen enthalten. Der *EAN13Generator* berücksichtigt dieses Präfix, eine etwaige eigene Implementierung muss dies nicht. Zusätzlich wird die **Identifier-Domäne** ausgewählt, in der die **MPIs** erzeugt werden sollen (der **E-PIX** hat standardmäßig hierfür die **Identifier-Domäne** "MPI" hinterlegt).

The screenshot shows the configuration interface for the E-PIX system. The top navigation bar includes tabs for Einstellungen (selected), Personenfelder, Vorverarbeitung, Matching, and Privatsphäre. The main form is divided into sections:

- Master Patient Index (MPI):** Includes fields for Generator (set to EAN13Generator), Prefix (set to MPI), and Identifier-Domäne (set to MPI).
- Geschwindigkeit:** Includes a field for Aktiviere paralleles Matching ab (set to 1.000) and a checkbox for Limitiere Suche auf Matching-Felder (reduziert Speichernutzung aber verhindert Suche nach anderen Feldern).

**Hinweis:** Wird dieselbe **Identifier-Domäne** für mehrere **Domänen** verwendet, so erzeugt der **E-PIX Domänen**-übergreifende eindeutige Kennungen (**MPIs**). Soll pro **Domäne** eine eigene **Identifier-Domäne** verwendet werden, so müssen zunächst mehrere **Identifier-Domänen** angelegt werden (Abschnitt 4.2). Es ist zu beachten, dass wenn derselbe Generator verwendet wird (z.B. *EAN13Generator*) auch verschiedene Präfixe vergeben werden müssen. Andernfalls würde der **E-PIX** versuchen, dieselben Kennungen in mehreren **Domänen** zu vergeben, was eine spätere Personenregistrierung verhindert.

Zur Verbesserung der Performance können weitere Einstellungen vorgenommen werden. Diese Einstellungen können in der Regel unverändert bleiben. Der **E-PIX**

führt dabei standardmäßig, bevor 1.000 **Identitäten** registriert wurden, das **Record Linkage** seriell durch. Danach werden Berechnungen auf einem Mehrkern-System auf die verschiedenen Prozessorkerne aufgeteilt. Zudem kann der Arbeitsspeicherbedarf reduziert werden, indem nur die Felder, die für das **Record Linkage** erforderlich sind, im Arbeitsspeicher bleiben. Dabei ist zu beachten, dass dabei auch die Suche auf diese Felder beschränkt wird.

### 4.3.2 Personenfelder

Unter dem Reiter Personenfelder werden die Pflichtfelder und Zusatzfelder festgelegt.

Feld	Bezeichnung
Keine Datensätze gefunden.	

Standardmäßig sind die Felder Vorname, Nachname, Geschlecht und Geburtsdatum als **Pflichtfelder** hinterlegt. Bei Bedarf kann diese Restriktion durch entfernen der Einträge aufgehoben werden. Dabei ist zu beachten, dass mindestens die Felder, die später für das **Record Linkage** verwendet werden sollen, als Pflichtfelder anzugeben sind. Pflichtfelder müssen bei einer Personenregistrierung ausgefüllt sein. Weitere Felder können aus der Liste ausgewählt werden.

Darüber hinaus können **Zusatzfelder** definiert werden (die bei Bedarf auch als Pflichtfelder gesetzt werden können). Der **E-PIX** hat hierfür zehn Freitextfelder, die aus einer Liste gewählt werden können (*Zusatzfeld hinzufügen*). Dabei ist zu beachten, dass diese Felder Restriktionen bzgl. der Länge der eingegebenen Daten aufweisen. Die maximale Anzahl der Zeichen, ist hinter dem jeweiligen Feld angegeben (vgl. *value1 - value10* in Tabelle 9.1). Für jedes Zusatzfeld kann ein Bezeichner gewählt werden, der bei der Personenregistrierung am entsprechenden Feld steht.

### 4.3.3 Validatoren

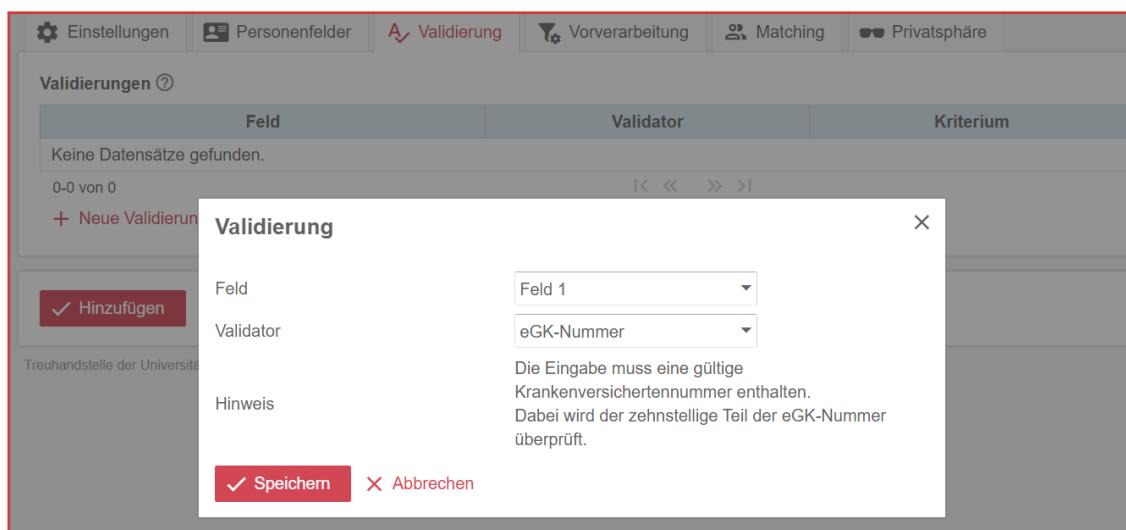
Bei der Personenregistrierung können die eingegebenen Angaben validiert werden. Der Registrierungsvorgang wird abgebrochen, wenn mindestens eine Angabe nicht valide ist. Eine Validierung findet dabei nur statt, wenn für ein entsprechendes Feld zumindest ein Validator hinterlegt wurde. Hierbei ausgenommen sind Geschlechtsangaben, welche einem internen Format entsprechen müssen und

das Geburtsdatum, welches nur valide Datumseingaben akzeptiert.

Muss die Eingabe eines Feldes mehreren Validierungskriterien entsprechen, so können mehrere Validatoren angegeben und gruppiert werden. Validatoren innerhalb einer Gruppe werden logisch verknüpft. Die Verknüpfung bestimmt, ob alle, keins oder nur exakt ein Validierungskriterium erfüllt sein soll. Auch mehrere Validator-Gruppen können logisch miteinander verknüpft werden. Eine detaillierte Beschreibung ist im Kapitel 6.10 in Tabelle 6.4 zu finden.

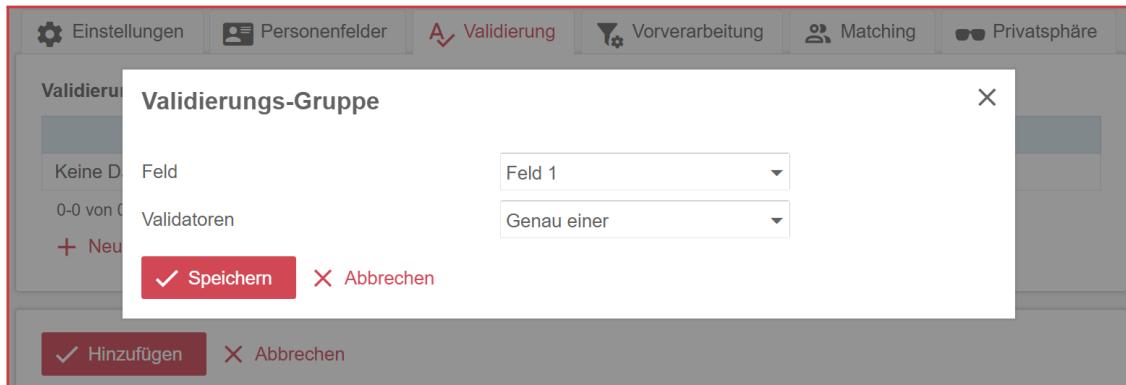
Es werden mehrere Validatoren bereitgestellt, welche entweder einen spezifischen Fall prüfen (z.B. die **KVNR**) oder mittels zusätzlicher Parameter flexibel konfiguriert werden können. Eine Auflistung aller Validatoren mit den dazugehörigen Parametern ist im Kapitel 6.10 in Tabelle 6.5.

Unter dem Reiter *Validierung* können die entsprechenden Validatoren konfiguriert werden. Über die Schaltfläche **+ Neue Validierung** kann im Dialog ein Validator konfiguriert werden. Hierbei wird das Feld angegeben, welches validiert werden soll. Außerdem wird der zu verwendende Validator ausgewählt. Für jeden Validator gibt es einen kurzen Hinweistext. In der folgenden Abbildung wurde für *Feld 1* der *eGK-Nummer*-Validator ausgewählt, welcher das Feld auf eine korrekte **KVNR** prüft.



Über die Schaltfläche **+ Neue Validierungs-Gruppe** kann eine neue Gruppe angelegt werden. Hierbei wird im Dialog das Feld angegeben werden, welche durch die Gruppe validiert wird und die Art der Verknüpfung. In der folgenden Abbildung wurde für *Feld 1* eine Validator-Gruppe hinterlegt. Bei einer Validierung darf nur das Kriterium eines Validators erfüllt sein (*Genau einer*). Nach dem Speichern können der Validator-Gruppe über die Schaltflächen **+** und **+** mehrere Validatoren

oder weitere Validator-Gruppen hinzugefügt werden.



**Info:** Mittels RegEx-Validator können sehr flexibel Validierungsregeln modelliert werden. Für komplexe Validierungen, bieten sich die Validierungs-Gruppen an, welche die Validierung auf mehrere Validatoren aufteilen. Soll ein Feld beispielsweise nur eine bestimmte Anzahl von definierten Zeichen enthalten, so kann dies per RegEx erfolgen. Alternativ kann auch der Alphabet- und Längen-Validator per Validierungs-Gruppe kombiniert werden, wobei die Verknüpfung die Erfüllung beider Bedingungen vorsieht.

#### 4.3.4 Vorverarbeitung

Bei der Personenregistrierung eingegebene **IDAT** können für ein **Record Linkage** aufbereitet werden. Dies umfasst bspw. das Entfernen von unerwünschten Zeichenketten oder die Vereinheitlichung von Umlauten. Dies betrifft aber nur die interne Verarbeitung. Die **IDAT** werden wie eingegeben in der Oberfläche dargestellt. Die Vorverarbeitung verbessert das **Record Linkage** und damit die Zusammenführung von Datensätzen, die zu einer Person zugehörig sind.

Felder mit Vorverarbeitungen					
Feld	Ersetzungen	Umwandlungen	Filter		
Vorname	23	2	0		
Nachname	23	2	0		
+ Feld zur Vorverarbeitung hinzufügen					

Standardmäßig sind für die Felder Vorname und Nachname entsprechende Vorverarbeitungen hinterlegt. Diese können bearbeitet oder entfernt werden. Zudem kann für weitere Felder eine Vorverarbeitung definiert werden. Der **E-PIX** unterscheidet zwischen Ersetzungen, Umwandlungen und Filtern. Es können jeweils mehrere Vorverarbeitungen pro Feld hinterlegt werden. Bei einer Ersetzung wird eine definierte Zeichenkette, mit einer anderen ersetzt (wenn die ersetzende Zeichenkette

leer ist, wird die zu ersetzende Zeichenkette entfernt. Bsp.: Zu ersetzen: „Dr.“, „Ersetzung: „“. Damit wird die Zeichenkette „Dr.“ restlos aus dem entsprechenden Feld entfernt.). Dabei ist zu beachten, dass die Groß- und Kleinschreibung berücksichtigt wird. Für Standardfälle, wie die Ersetzung von Umlauten, gibt es Umwandlungen. Der **E-PIX** wird mit vier Umwandlungen ausgeliefert:

- **ToUpperCaseTransformation:** Ersetzt alle Zeichen durch den entsprechenden Großbuchstaben. Beim **Record Linkage** werden so Unterschiede bei der Groß- und Kleinschreibung nicht berücksichtigt.
- **CharsMutationTransformation:** Ersetzt alle Umlaute: „ä“ durch „ae“, „Ä“ durch „AE“, „ü“ durch „ue“, „Ü“ durch „UE“, „ö“ durch „oe“, „Ö“ durch „OE“ und „ß“ durch „SS“.
- **CharNormalizationTransformation:** Überführt eine Zeichenkette in ASCII<sup>1</sup>. Dies entfernt z.B. Akzente. Dabei ist zu beachten, dass Umlaute wie ä nicht in ae, sondern in a überführt werden. Eine Kombination mit *CharsMutation-Transformation* ist möglich.
- **TrimTransformation:** Entfernt führende und folgende Leerzeichen.  
Bsp.: „ Müller “ → „Müller“.

Beim Filtern kann ein Alphabet mit zulässigen Zeichen angegeben werden. Alle anderen Zeichen, werden bei der Vorverarbeitung durch das angegebene Zeichen ersetzt. Wenn letzteres leer ist, dann werden unzulässige Zeichen entfernt. Dieser Filter sollte nur dann angewandt werden, wenn die Menge der zulässigen Zeichen bekannt ist (z.B. die Postleitzahl darf nur Zahlen enthalten) oder begrenzt werden muss (z.B. um **Bloomfilter** zu erzeugen).

**Hinweis:** Die Vorbearbeitung hat Einfluss auf das **Record Linkage** und kann zu unerwarteten Verhalten führen. So führt das Entfernen der Trennzeichen von *multiple-values* (Abschnitt 6.14.7) dazu, dass z.B. mehrere Vornamen nicht mehr einzeln betrachtet werden und zu schlechteren Matching-Ergebnissen führt.

#### 4.3.5 Matching

Unter dem Reiter *Matching* werden die Parameter für das **Record Linkage** festgelegt. Dies umfasst das Setzen von Schwellwerten, also ab wann zwei Datensätze zu einer Person zugeordnet werden und welche Felder für den Abgleich verwendet werden sollen.

<sup>1</sup> [wikipedia.org/wiki/American\\_Standard\\_Code\\_for\\_Information\\_Interchange](https://en.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange)

Feld	Algorithmus	Modus	Blocking ab Übereinstimmung von	Matching ab Übereinstimmung von	Gewichtung	
Vorname	LevenshteinAlgorithm	Text	40 %	80 %	8.0	
Nachname	LevenshteinAlgorithm	Text	0 %	80 %	6.0	
Geschlecht	LevenshteinAlgorithm	Text	0 %	75 %	3.0	
Geburtsdatum	LevenshteinAlgorithm	Zahlen	60 %	100 %	9.0	

Der **E-PIX** unterscheidet zwischen zwei Modi. Zum einen kann der **E-PIX** Personendaten selbst mittels **Record Linkage** zusammenführen, **MPIs** vergeben usw. (Führe **Record Linkage** durch: ja). Es besteht ebenso die Möglichkeit, dass der **E-PIX** Personendatensätze nur ablegt (Führe **Record Linkage** durch: nein). Dies kann gewünscht sein, wenn ein **Record Linkage** bereits in einem anderen System durchgeführt wurde (z.B. in einem KAS). In beiden Fällen wird eine Matching-Konfiguration hinterlegt, damit der **E-PIX** Personendatensätze korrekt zuordnen kann. Sollen die Personendatensätze nur abgelegt werden, erfolgt dies unter bestimmten Bedingungen. Beispielsweise müssen zwei Personendatensätze mit derselben übergebenen (externen ID) komplett übereinstimmen, oder zumindest laut der angegebenen Konfiguration eine gewisse Übereinstimmung aufweisen. Soll der **E-PIX** ein **Record Linkage** durchführen, bestimmt die Konfiguration, wann zwei Personendatensätze zur selben Person als **Identitäten** zugeordnet werden und dementsprechend dieselbe **MPI** erhalten.

Beim **Record Linkage** klassifiziert der **E-PIX** die Datensätze in **Match**-Typen (eine detaillierte Beschreibung ist in Abschnitt 8.1 zu finden). Ein **Möglicher Match** entsteht, wenn die Übereinstimmung über dem Schwellwert für einen **Möglichen Match** liegt, jedoch niedriger als der Schwellwert für einen **Automatischer Match** ist. Bei einem **Möglicher Match** kann später manuell entschieden werden (siehe Abschnitt 8.5), ob zwei Datensätze zur selben Person zugehörig sind, oder zwei verschiedene Personen darstellen. Bei der Entscheidung können entsprechend weitere Informationen zugezogen werden. Sind beide Schwellwerte identisch, so werden keine **Möglichen Matches** angelegt.

Liegt die ermittelte Übereinstimmung über dem Schwellwert für einen **Automatischer Match**, so führt der **E-PIX** die entsprechenden Datensätze entsprechend zusammen, auch wenn keine vollständige Übereinstimmung (z.B. durch Tippfehler) vorliegt. Im Ergebnis werden die Datensätze als **Identitäten** einer Person zugeordnet. Ein automatisches Zusammenführen kann unterbunden werden, indem das Kontrollkästchen „*nie*“ angewählt oder der Wert auf 1000 gesetzt wird.

Die Übereinstimmung zweier Datensätze, ermittelt der **E-PIX** anhand der definierten *Matching Felder*. Für jedes Feld kann ein Vergleichsalgorithmus, eine Wichtung und Schwellwerte für das **Blocking** und den Abgleich definiert werden. Der **E-PIX** unterstützt verschiedene Vergleichsalgorithmen. Für die meisten Fälle ist jedoch der Algorithmus **LevenshteinAlgorithm** zu empfehlen. Dieser ermittelt die Levenshtein-Distanz zweier Zeichenketten, anhand derer die Übereinstimmung berechnet werden kann. Alle unterstützten Algorithmen sind in Tabelle 6.11 aufgelistet und beschrieben.

Das **Blocking** beschleunigt das **Record Linkage**, indem es zunächst nur grob Datensätze miteinander abgleicht und bei hinreichender Übereinstimmung alle *Matching Felder* zum Abgleich verwendet. Der Schwellwert sollte daher nicht zu hoch gewählt werden, damit das **Blocking** nicht Datensätze aussortiert, die bei einem genaueren Vergleich einer Person zugeordnet werden würden. Der Modus gibt an, welcher Datentyp im Feld enthalten ist (Text oder Zahlen) und betrifft nur das **Blocking**. Dieser optimiert den internen Abgleich und wird in den meisten Fällen auf „Text“ gesetzt.

Der Schwellwert für das Matching gibt an, ab welcher Übereinstimmung zwei *Matching Felder* übereinstimmen. Das Ergebnis fließt der angegebenen *Gewichtung* entsprechend, in das Ergebnis mit ein. Wird anhand aller *Matching Felder* eine der oben genannten Schwellwerte überschritten, werden die betreffenden Datensätze entsprechend als **Möglicher Match** oder **Automatischer Match** klassifiziert. Andernfalls wird der zu registrierende Datensatz als **Kein Match** klassifiziert und entsprechend als neue Person angelegt.

Felder können als *Multi-Wert Feld* angegeben werden. Dabei werden die Inhalte eines Feldes anhand eines *Trennsymbols* aufgeteilt und separat abgeglichen. Wird z.B. erwartet, dass im Feld Vorname mehrere Vornamen angegeben werden, können so die einzelnen Vornamen zwischen zwei Personendatensätzen abgeglichen werden. Eine detailliertere Beschreibung, inkl. der hierfür anzugebenden Schwellwerte, ist in Abschnitt 6.14.7 zu finden.

**Hinweis:** Die Konfiguration basiert auf Erfahrungswerten und ist häufig Projekt-abhängig. Je nach zu erwartender Datenqualität können höhere Schwellwerte gewählt werden, um beispielsweise weniger von **Möglichen Matches** zu erzeugen. Für verschiedene Anwendungsszenarien sind in Kapitel 7 diverse Konfigurationen vorgestellt.

Bei der **Dublettenauflösung** können Gründe angegeben werden. Dies erfolgt mittels Freitextfeld. Für häufig auftretende Gründe, können entsprechende Vorlagen definiert werden.

Gründe für Dublettenauflösung		
Bezeichnung	Hinweis	
Tippfehler	Vertauschte, fehlende oder zu viele Zeichen	 
Namensänderung durch Heirat	Änderung des Nachnamens auf Grund einer Heirat	 
<a href="#">+ Grund hinzufügen</a>		

Hierfür wird für jeden Grund ein Bezeichner gewählt, der bei der **Dublettenauflösung** angewählt werden kann. Der angegebene Hinweis wird dann entsprechend protokolliert.

#### 4.3.6 Privatsphäre

Der **E-PIX** ermöglicht das Anlegen eines **Bloomfilters** für eine **Identität**, um ein **PPRL** durchzuführen. Dies kommt normalerweise bei Standort-übergreifenden Abgleichen zum Einsatz. Der **E-PIX** kann sowohl **Bloomfilter** anlegen, als auch miteinander vergleichen. Der Vergleich wird mittels Matching Felder definiert. Standort-interne Vergleiche finden üblicherweise über die Klartextdaten der **IDAT** statt. Standardmäßig wird kein **Bloomfilter** angelegt. Die Konfiguration erfolgt üblicherweise projektspezifisch.

Einstellungen	Personenfelder	Vorverarbeitung	Matching	Privatsphäre
Speichermodus	Identifizierende Daten und Bloomfilter			
Bloomfilter				
Algorithmus	Quell-Felder	Speicherfeld		
RandomHashingStrategy	Vorname Nachname Geburtsdatum Geschlecht	Feld 6	 	<a href="#">+ Bloomfilter hinzufügen</a>

Im Bild wurde exemplarisch die Konfiguration eines **Bloomfilters** hinterlegt.

**Hinweis:** Der **E-PIX** unterstützt mehrere Algorithmen zur Erzeugung von **Bloomfiltern** und zusätzliche Härtungsverfahren, die kombiniert werden können. Achten Sie darauf, dass die **Bloomfilter**-Konfiguration auf die Bedürfnisse des jeweiligen Projekts angepasst werden muss und so mitunter ein Abwagen zwischen Sicherheit und Qualität stattfinden muss. Ein **Bloomfilter** stellt immer eine Verallgemeinerung der Eingangsdaten dar und kann zu schlechteren Matching-Ergebnissen führen, sofern der **Bloomfilter** zum **Record Linkage** genutzt wird.

Über die Schaltfläche  **Bloomfilter hinzufügen** wird eine neue **Bloomfilter**-Konfiguration angelegt. Zunächst wird der zu verwendende Algorithmus angegeben. Eine Auflistung mit kurzer Erläuterung ist in Tabelle 6.7 zu finden.

Je nach verwendetem Algorithmus, kann ein Alphabet angegeben werden. Dabei ist zu beachten, dass zur **Bloomfilter**-Generierung die vor-verarbeiteten Werte verwendet werden. Damit muss sichergestellt werden, dass die verwendeten **IDAT**-Felder so vor-verarbeitet wurden (Abschnitt 4.3.4 und 6.13), dass diese nur Zeichen enthalten, die auch im angegebenen Alphabet enthalten sind. Besteht das Alphabet nur aus Großbuchstaben, so sollte zuvor das Feld zuvor mit `ToUpperCaseTransformation` transformiert worden sein. Umlaute sollten zuvor mit `CharsMutationTransformation` und Akzente etc. per `CharNormalizationTransformation` entfernt worden sein. Mit einem Filter kann sichergestellt werden, dass Felder nur Zeichen beinhalten, die auch im Alphabet vorkommen. Zu beachten ist, dass die Groß- und Kleinschreibung beachtet wird. Sollen die Zustände vom Feld Geschlecht berücksichtigt werden (intern kodiert mit m, f, o, u, x), so müssen diese Zeichen entsprechend auch im Alphabet vorkommen.

Die Länge gibt die Anzahl der Bits pro **Bloomfilter** an. Zwar ist die Wahl des Speicherfeldes frei, jedoch ist zu beachten, dass der **E-PIX** die Feldlängen intern begrenzt. Außerdem werden die **Bloomfilter** intern im Base64-Format kodiert. Die meisten Felder vom **E-PIX** erlauben eine maximale Länge von 255 Zeichen<sup>2</sup>. Werden längere **Bloomfilter** benötigt, sollten die frei definierbaren Felder (`value8` - `value10`, Tabelle 9.1) verwendet werden. Die tatsächlich benötigte Länge kann durch die Verwendung von Härtungsverfahren beeinflusst werden. So halbiert jede Faltung beim *XOR-Folding* die resultierende Länge. Die Nutzung eines *Balanced Bloomfilters* verdoppelt die resultierende Länge.

Mit der Länge der N-Gramme wird angegeben, wie lang die Teil-Zeichenketten

<sup>2</sup> Die benötigte Speicherlänge kann über die folgende Formel ermittelt werden:  $z = 4 \times \lceil \frac{\frac{Bits}{8}}{3} \rceil$   
Für eine Länge von 1000 Bits ergibt sich ein Bedarf von  $z = 4 \times \lceil \frac{\frac{1000}{8}}{3} \rceil = 167$  Zeichen.

beim kodieren der Felder in den **Bloomfilter** sein sollen. Üblicherweise werden hierfür Bigramme (N=2) genutzt.

Mit Bits pro N-Gramm kann die Anzahl der Bit-Positionen pro N-Gramm angegeben werden. Je höher dieser Wert gewählt wird, desto mehr Positionen werden im resultierenden **Bloomfilter** belegt.

Die Anzahl der XOR-Faltungen (*XOR-Folding*<sup>3</sup>) gibt an, wie oft ein **Bloomfilter** gefaltet werden soll. Dies härtet den **Bloomfilter** gegen Angriffe. Mit jeder Faltung halbiert sich die Länge des **Bloomfilters**. Zu beachten ist, dass die Anzahl der Faltungen ein ganzzahliger Teiler der Länge sein muss. Die Anzahl der Faltungen sollte gering gehalten werden, da andernfalls die Qualität des **Record Linkages** negativ beeinflusst werden kann.

Mit der Aktivierung des Kontrollkastens **Balanced Bloomfilter**<sup>4</sup>, wird bei der Erzeugung des **Bloomfilters** eine negierte Kopie angefügt und die Bit-Positionen mittels des angegebenen Werts (*Seed*) zufällig vertauscht. Der *Seed* muss eine Ganzzahl sein.

Das *Speicherfeld* gibt an, in welchem Feld der resultierende **Bloomfilter** gespeichert werden soll. Dabei muss beachtet werden, dass zum einen der **Bloomfilter** in das ausgewählte Feld passt (siehe auch *Länge*) und zum anderen, dass etwaige Informationen im Feld überschrieben werden (Bsp.: Wenn das Feld Vorname als Speicherfeld gewählt wurde, ist nach einer Personenregistrierung der Vorname durch den **Bloomfilter** überschrieben). Es ist daher ratsam, dass Speicherfeld auf ein Value-Feld (*Zusatzfeld*) zu setzen.

Jedem **Bloomfilter** können beliebig viele *Quell-Felder* zugeordnet werden. Auf Basis der darin enthaltenen Werte, wird bei der Registrierung der **Bloomfilter** erzeugt. Je nach Verfahren muss zusätzlich ein *Seed* (als Ganzzahl), ein *fester Salt* (beliebige Zeichenkette) oder ein Feld als *Salt* angegeben werden. Ein *Salt* ist ein Wert, der intern vor der Kodierung jedem N-Gramm angefügt wird. Wird ein Feld als *Salt* gewählt, so wird vom jeweiligen Datensatz der Wert des Feldes hierzu

<sup>3</sup> Schnell, Rainer and Borgs, Christian, XOR-Folding for Bloom Filter-based Encryptions for Privacy-preserving Record Linkage (December 22, 2016). German Record Linkage Center, NO. WP-GRLC-2016-03, DECEMBER 22, 2016, Available at SSRN: <https://ssrn.com/abstract=3527984> or <http://dx.doi.org/10.2139/ssrn.3527984>

<sup>4</sup> R. Schnell and C. Borgs, "Randomized Response and Balanced Bloom Filters for Privacy Preserving Record Linkage," 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), Barcelona, Spain, 2016, pp. 218-224, doi: 10.1109/ICDMW.2016.0038.

verwendet. Hierzu eignen sich festgelegte Pflichtfelder (z.B. das Geburtsdatum).

**Info: Field-Level Bloomfilter oder Cryptographic Long Term Key (CLK)?**

Der **E-PIX** unterstützt sowohl die Erzeugung von **Field-Level Bloomfilter** (ein **Bloomfilter** pro Feld), als auch die Erzeugung von **Cryptographic Long Term Keys** (**Bloomfilter** kodiert mehrere Felder). Zum Erzeugen von **Field-Level Bloomfilter**, wird pro Feld ein **Bloomfilter** definiert. Dabei wird als Quell-Feld nur das entsprechende Feld ausgewählt. Beim **Cryptographic Long Term Keys** werden mehrere Quell-Felder angegeben, die alle im selben **Bloomfilter** kodiert werden.

Soll der **E-PIX** nur zur Erzeugung von **Bloomfiltern** genutzt werden (bspw. weil die Verwaltung der **IDAT** in einem anderen System erfolgt), so kann der *Speichermodus* zu *Nur Bloomfilter* geändert werden. Die angegebenen Quell-Felder werden nur zu Generierung des **Bloomfilters** verwendet. Alle **IDAT**-Felder werden nichtpersistiert. Ein **Record Linkage** kann dann nur über die **Bloomfilter** durchgeführt werden. Standardmäßig werden sowohl **Bloomfilter**, als auch **IDAT**-Felder persistiert.

**Hinweis:** Abweichend von anderen Konfigurationen, kann während eines laufenden Projekts das Hinzufügen der **Bloomfilter**-Einstellungen notwendig sein. In Abschnitt 9.4 wird dies entsprechend erläutert.

## 5. SOAP-Schnittstelle

Neben der grafischen Benutzerschnittstelle, steht eine maschinenverständliche Web-Schnittstelle zur Verfügung. Diese kann mit dem SOAP-Protokoll angesprochen werden. Beim laufenden Dienst werden je nach Zweck die dazu vorhandenen Definitionen der SOAP-Schnittstellen mit dem folgenden Pfaden abgerufen (die URLs müssen entsprechend angepasst werden).

### **Personenverwaltung (inkl. Record Linkage):**

<http://example.org:8080/epix/epixService?wsdl>

### **Konfiguration und Domänenmanagement (Management-Schnittstelle):**

<http://example.org:8080/epix/epixManagementService?wsdl>

### **Versenden von Notifications:**

<http://example.org:8080/epix/epixServiceWithNotification?wsdl>

### **Die Entwicklerdokumentation ist unter der folgenden URL zu finden:**

<https://www.ths-greifswald.de/epix/doc>

Für das Anlegen einer **Datenquelle** (Abschnitt 5.2), **Identifier-Domäne** (Abschnitt 5.3) und **Domäne** (Abschnitt 5.4) wird die Management-Schnittstelle zur Konfiguration verwendet.

### **5.1 Senden von Nachrichten an den E-PIX**

Zum Absenden von SOAP-Nachrichten an den **E-PIX** kann ein beliebiger http-Client verwendet werden. An dieser Stelle werden drei Beispiele vorgestellt, die

den Einstieg erleichtern sollen. Die meisten Funktionalitäten können über die Weboberfläche ausgeführt werden. Die Nutzung der SOAP-Schnittstelle hingegen, ist meistens nur dann erforderlich, wenn ein anderer Dienst direkt an den **E-PIX** angebunden werden soll. Die jeweiligen Werkzeuge werden anhand des Beispiels zum Löschen einer **Domäne** beschrieben.

SOAP-UI und/oder Postman müssen zunächst installiert werden. Der Befehl in der PowerShell (Windows) kann meistens direkt ausgeführt werden.

### 5.1.1 SOAP-UI

**Info:** SOAP-UI kann über <https://www.soapui.org/> heruntergeladen werden.

Der **E-PIX** liefert eine Beschreibung der Schnittstelle in Form einer **Web Services Description Language (WSDL)**, welche durch SOAP-UI eingelesen werden kann. So kann die Struktur der Nachricht für die jeweilige Methode schnell erzeugt werden. Es müssen dann nur noch die jeweiligen Werte ausgefüllt werden.

Hierzu muss zunächst diese Beschreibung im **XML**-Format heruntergeladen werden. Hierfür sollte die eigene Instanz verwendet werden, da andernfalls die URLs angepasst werden müssen. Es gibt zwei Schnittstellen die angesprochen werden können. Zum einen die Management-Schnittstelle welche vorrangig zum Bearbeiten von **Domänen** erforderlich ist und zum anderen eine Schnittstelle, welche zum Registrieren und Abrufen von Personendaten dient (siehe oben).

Im Folgenden wird die Management-Schnittstelle benötigt. Hierzu muss die jeweilige URL aufrufen werden und die dargestellte Seite als **XML**-Datei gespeichert werden. Die URL ist oben angegeben.

Nun kann SOAP-UI geöffnet werden. Danach oben links auf **SOAP** klicken, um ein neues SOAP-Projekt anzulegen. Nun unter *Initial WSDL* die eben heruntergeladene **XML**-Datei angeben. Der Projektname kann bei Bedarf angepasst werden. Zum Anlegen **Ok** klicken.

Auf der linken Seite unter *Navigator* ist unter *Projects* nun das Projekt angelegt. SOAP-UI erzeugt für alle Methoden der Schnittstelle eine Beispiel-Nachricht. Zunächst muss hierfür der Projektordner und das darunterliegende **SOAP-Binding** aufgeklappt werden. Nun kann die gewünschte Methode ausgewählt werden (in diesem Fall `deleteDomain`). Der Eintrag zur Methode muss ebenfalls aufgeklappt werden. Danach per Doppelklick auf *Request 1* die vorbereitete Nachricht öffnen.

Dieser Request beinhaltet alle Elemente, die zum Aufruf der Schnittstelle erforderlich sind.

Nun können die Parameter angepasst werden. Der Domänenname (`domainName`) enthält den Schlüssel der **Domäne**. Der Parameter `force` kann auf `true` oder `false` gesetzt werden. Wird hierbei `true` angegeben, wird die **Domäne** auch dann gelöscht, wenn in dieser bereits Personendaten registriert wurden.

```
1  <soapenv:Envelope
2      xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
3      xmlns:ser="http://service.epix.ttp.icmvc.emau.org/">
4      <soapenv:Header/>
5      <soapenv:Body>
6          <ser:deleteDomain>
7              <domainName>domain-key</domainName>
8              <force>true</force>
9          </ser:deleteDomain>
10         </soapenv:Body>
11     </soapenv:Envelope>
```

Listing 5.1: SOAP-Anfrage zum Löschen einer **Domäne**.

Bevor die Anfrage abgesendet wird, sollte noch oben die URL geprüft werden. Diese sollte bereits korrekt hinterlegt sein, wenn die **WSDL**-Datei von der eigenen Instanz erzeugt wurde (`http(s)://<URL:PORT>/epix/epixManagementService`). Die Anfrage wird mit Klick auf den grünen Pfeil abgeschickt. Als Antwort sollte ein „HTTP 200 OK“ geliefert werden. Dies bedeutet, der Vorgang wurde erfolgreich abgeschlossen.

## 5.1.2 Postman

**Info:** Postman kann über <https://www.postman.com/> heruntergeladen werden.

Bei Postman muss die Anfrage selbst komplett definiert werden. Hierzu muss Postman zunächst geöffnet werden. Mit einem Klick auf den oberen Button `New` wird eine neue Anfrage erzeugt. Danach muss die Methode auf `POST` gesetzt werden. Als URL wird die Management-Schnittstelle angegeben (`http(s)://<URL:PORT>/epix/epixManagementService`). Diese muss an die eigene Instanz angepasst werden. Unter dem Reiter `Headers` muss ein neues `Key-Value`-Paar angegeben werden. Als `Key` wird `Content-Type` und als `Value` wird `text/xml; charset=UTF-8` angegeben. Unter dem Reiter `Body` muss der Radio-Button `raw` ausgewählt werden. Danach kann in das Textfeld die folgende Nachricht eingetragen werden. Die Parameter können wie unter Abschnitt 5.1.1 bear-

beitet werden.

---

```

1  <soapenv:Envelope
2      xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
3      xmlns:ser="http://service.epix.ttp.icmvc.emau.org/">
4      <soapenv:Header/>
5      <soapenv:Body>
6          <ser:deleteDomain>
7              <domainName>domain-key</domainName>
8              <force>true</force>
9          </ser:deleteDomain>
10         </soapenv:Body>
11     </soapenv:Envelope>

```

---

Listing 5.2: SOAP-Anfrage zum Löschen einer Domäne.

Mit einem Klick auf *Send* oben rechts, wird die Nachricht abgeschickt. Unten rechts sollte der Status „200 OK“ angezeigt werden.

### 5.1.3 PowerShell (Windows)

**Info:** Die PowerShell ist meistens unter Windows vorinstalliert. Grundsätzlich kann diese auch unter Linux oder MacOS verwendet werden.

Die PowerShell kann unter Windows geöffnet werden. Um eine SOAP-Nachricht an den E-PIX zu übermitteln, kann Folgendes vollständig eintragen und abgeschickt werden.

```

1  $headers = New-Object
2      "System.Collections.Generic.Dictionary[[String],[String]]"
3  $headers.Add("Content-Type", "text/xml; charset=UTF-8")

4  $body = @@
5  <soapenv:Envelope>
6      xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
7      xmlns:ser="http://service.epix.ttp.icmvc.emau.org/">
8      <soapenv:Header/>
9      <soapenv:Body>
10     <ser:deleteDomain>
11     <domainName>domain-key</domainName>
12     <force>true</force>
13     </ser:deleteDomain>
14     </soapenv:Body>
15   </soapenv:Envelope>
16   "@

17  Invoke-WebRequest
18      'http(s)://<URL:PORT>/epix/epixManagementService' -Method

```

```
'POST' -Headers $headers -Body $body
```

Listing 5.3: SOAP-Anfrage zum Löschen einer Domäne.

Die URL muss entsprechend angepasst werden. Die Nachricht kann wie in den vorherigen Abschnitten angepasst werden. Erscheint als Antwort in der ersten Zeile ein „StatusCode: 200“, wurde der Vorgang erfolgreich abgeschlossen.

## 5.2 Anlegen einer Datenquelle

In Listing 5.4 ist die exemplarische Darstellung eines SOAP-Requests zur Erstellung einer neuen **Datenquelle** gezeigt. Mit dem Element `description` kann eine kurze Beschreibung für die **Datenquelle** angegeben werden. Über das Element `label` wird ein Bezeichner gewählt, der in der Weboberfläche angezeigt wird. Die Referenzierung der **Datenquelle** erfolgt stets über den Namen. Der Name kann über das Element `name` festgelegt werden. Dieser muss eindeutig sein und kann im Gegensatz zu dem Label nicht mehr geändert werden. Wann immer über die SOAP-Schnittstelle eine **Datenquelle** angegeben werden muss, muss der Name dieser **Datenquelle** angegeben werden.

```
1  <soapenv:Envelope
  2      xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  3      xmlns:ser="http://service.epix.ttp.icmvc.emau.org/">
  4
  5      <soapenv:Header />
  6
  7      <soapenv:Body>
  8          <ser:addSource>
  9              <source>
 10                  <description>Eine kurze Beschreibung</description>
 11                  <label>Neue Datenquelle</label>
 12                  <name>data_source</name>
 13              </source>
 14          </ser:addSource>
 15      </soapenv:Body>
 16  </soapenv:Envelope>
```

Listing 5.4: SOAP-Anfrage zur Erstellung einer neuen **Datenquelle**.

## 5.3 Anlegen einer Identifier-Domäne

In Listing 5.5 ist exemplarisch das Anlegen einer **Identifier-Domäne** gezeigt. Mit dem Element `name` wird ein eindeutiger Name vergeben. Eine Referenzierung der **Identifier-Domäne** erfolgt über die SOAP-Schnittstelle stets über den Namen. Dieser kann später nicht mehr verändert werden. Mit dem Element `label` kann ein sprechender Name vergeben, der in der Weboberfläche angezeigt wird. Das

Label kann später geändert werden. Optional kann mit dem Element `description` eine kurze Beschreibung der **Identifier-Domäne** angegeben werden. Außerdem kann mit dem Element `oid` ein **OID** angegeben werden. Wenn dieser explizit angegeben wird, muss dieser eindeutig sein. Wird kein **OID** angegeben, so erzeugt der **E-PIX** automatisch einen eindeutigen **OID**. Nach dem Anlegen der **Identifier-Domäne** kann diese beim Anlegen einer **Domäne** angegeben werden.

---

```
1 <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ser="http://service.epix.ttp.icmvc.emau.org/"
2   <soapenv:Header/>
3   <soapenv:Body>
4     <ser:addIdentifierDomain>
5       <identifierDomain>
6         <description>Beschreibung zum Identifier</description>
7         <label>Personenidentifikator</label>
8         <name>PID</name>
9         <oid>123.456.789</oid>
10        </identifierDomain>
11      </ser:addIdentifierDomain>
12    </soapenv:Body>
13 </soapenv:Envelope>
```

Listing 5.5: SOAP-Anfrage zur Erstellung einer neuen **Identifier-Domäne**.

## 5.4 Anlegen einer Domäne

Zum Anlegen einer **Domäne** ist es erforderlich, eine Konfiguration im **XML**-Format anzugeben. Die Zusammensetzung ist im Kapitel 6 erläutert. Die **XML**-Konfiguration wird im Element `config` angegeben. Weitere Einstellungen, werden direkt in der SOAP-Anfrage vorgenommen. In Listing 5.6 ist exemplarisch eine SOAP-Anfrage zum Anlegen einer **Domäne** gezeigt. Die **XML**-Konfiguration ist im Sinne der Übersichtlichkeit nicht aufgeführt. Mit dem Element `description` kann eine kurze Beschreibung für die **Domäne** hinterlegt werden. Mit dem Element `label` wird ein sprechender Name für die **Domäne** hinterlegt. Dieser wird in der Weboberfläche angezeigt und kann jederzeit geändert werden. Mit dem Element `name` wird ein Name vergeben, mit dessen die **Domäne** referenziert wird. Dieser Name kann später nicht mehr geändert werden. Mit dem Element `name` unter `mpiDomain` wird der Name der **Identifier-Domäne** angegeben. Der **E-PIX** erzeugt später die eindeutigen Kennungen innerhalb dieser **Identifier-Domäne**. Der Name entspricht dem Element `name`, der beim Anlegen der **Identifier-Domäne** gewählt wurde (Abschnitt 5.3). Mit dem Element `name` unter `safeSource` wird der Name der **Sichere Datenquelle** angegeben. Dieser entspricht dem Namen der im Element `name`

beim Anlegen der **Datenquelle** angegeben wurde (Abschnitt 5.2).

---

```
1  <soapenv:Envelope
2      xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
3      xmlns:ser="http://service.epix.ttp.icmvc.emau.org/">
4      <soapenv:Header/>
5      <soapenv:Body>
6          <ser:addDomain>
7              <domain>
8                  <config>
9                      <![CDATA[ XML-Konfiguration ]]>
10                 </config>
11                 <description>Beschreibung des Projekts</description>
12                 <label>Projekt-A</label>
13                 <mpiDomain>
14                     <name>PID</name>
15                 </mpiDomain>
16                 <name>project-a</name>
17                 <safeSource>
18                     <name>data_source</name>
19                 </safeSource>
20             </domain>
21         </ser:addDomain>
22     </soapenv:Body>
23 </soapenv:Envelope>
```

---

Listing 5.6: SOAP-Anfrage zur Erstellung einer neuen **Domäne**.

```

</simple-transformation-type xsi:type="ma:SimpleTransformation">
  <input-pattern>Dipl.</input-pattern>
  <output-pattern></output-pattern>
</simple-transformation-type>
<simple-transformation-type xsi:type="ma:SimpleTransformation">
  <input-pattern></input-pattern>
  <output-pattern></output-pattern>
</simple-transformation-type>
<simple-transformation-type xsi:type="ma:SimpleTransformation">
  <input-pattern></input-pattern>
  <output-pattern></output-pattern>
</simple-transformation-type>
<complex-transformation-type xsi:type="ma:ComplexTransformation">
  <qualified-class-name>org.emau.icmvic.ttp.deduplication.preprocessing.impl.ToUpperCaseTransformation</qualified-class-name>
</complex-transformation-type>
<complex-transformation-type xsi:type="ma:ComplexTransformation">
  <qualified-class-name>org.emau.icmvic.ttp.deduplication.preprocessing.impl.CharsMutationTransformation</qualified-class-name>
</complex-transformation-type>
</preprocessing-field>
<preprocessing-field>
  <field-name>lastName</field-name>
  <simple-transformation-type xsi:type="ma:SimpleTransformation">
    <input-pattern></input-pattern>
    <output-pattern></output-pattern>
  </simple-transformation-type>
  <simple-transformation-type xsi:type="ma:SimpleTransformation">
    <input-pattern></input-pattern>
    <output-pattern></output-pattern>
  </simple-transformation-type>
  <simple-transformation-type xsi:type="ma:SimpleTransformation">
    <input-pattern>Dr.</input-pattern>
    <output-pattern></output-pattern>
  </simple-transformation-type>
</preprocessing-field>

```

## 6. XML-Konfiguration



Abbildung 6.1: Alle Elemente, die bei der Konfiguration der Domäne verwendet werden können.

Die Konfiguration einer Domäne kann vollständig über die Weboberfläche (siehe Abschnitt 4.3) erfolgen. Alternativ kann über diese eine vordefinierte XML-Datei

importiert werden. Über die SOAP-Schnittstelle erfolgt die Konfiguration ausschließlich im **XML**-Format.

In Abbildung 6.1 ist die Struktur der Konfiguration illustriert. Es sind alle Elemente aufgelistet, die bei der Konfiguration verwendet werden können. Das Element **MatchingConfiguration** ist das Wurzelement. Alle Elemente sind diesem Element untergeordnet. Die Struktur gibt an, welche Elemente anderen Elementen untergeordnet sind. Die angegebene Reihenfolge der Elemente ist dabei einzuhalten. Eine Erläuterung aller Elemente mit Beispielen und validen Wertebereichen folgt im nächsten Abschnitt.

## 6.1 Match Modus

Mithilfe des Elements `matching-mode` kann definiert werden, ob ein **Record Linkage** durchgeführt werden soll, oder nicht. Mit dem Modus **MATCHING\_IDENTITIES**, findet ein **Record Linkage** statt. Mit dem Modus **NO\_DECISION** wird kein **Record Linkage** durchgeführt und Personendaten werden nur übernommen und im **E-PIX** hinterlegt. Dies kann gewünscht sein, wenn Personendaten z.B. durch ein **KIS** übermittelt werden und bereits Identifizierer vergeben wurden und bereits ein **Record Linkage** durchgeführt wurde. In Tabelle 6.1 sind die zwei Modi im Detail erläutert.

Tabelle 6.1: Unterstützte Matching-Modes.

Wert	Beschreibung
<code>MATCHING_IDENTITIES</code>	Bei der Registrierung von Personen wird ein <b>Record Linkage</b> durchgeführt (Verwendung von <code>addPerson</code> nicht möglich). Die Konfiguration des <b>Record Linkages</b> wird mit dem Element <code>matching</code> angegeben.
<code>NO_DECISION</code>	Bei der Registrierung von Personen findet kein <b>Record Linkage</b> statt und die Personendaten werden nur übernommen. Bei jedem Registriervorgang (mit der Funktion <code>addPerson</code> ) wird dabei eine neue Person angelegt.

```
1 <matching-mode>MATCHING_IDENTITIES</matching-mode>
```

Listing 6.1: XML-Code zum Definieren des Matching-Modes.

**Hinweis:** Auch im *Matching-Mode* NO\_DECISION wird eine Matching-Konfiguration hinterlegt. Der **E-PIX** prüft anhand dessen, ob die **IDAT** der **Identitäten** mit verschiedenen **Identifier** auch verschiedenen Personen zugeordnet werden würde.

## 6.2 MPI Generator

Wird eine Person im **E-PIX** erstmalig eingetragen, so erhält diese einen **MPI**. Die Erzeugung eines **MPI** wird dabei durch einen Generator durchgeführt. Derzeit ist im **E-PIX** ein Generator (**EAN13Generator**) integriert, welcher eindeutige **MPIs** erzeugt. Weitere Generatoren können implementiert werden. In Listing 6.2 ist die Angabe des Generators dargestellt.

```
1 <mpi-generator>
2   org.emau.icmvcttp.epix.gen.impl.EAN13Generator
3 </mpi-generator>
```

Listing 6.2: XML-Code zum Definieren des MPI-Generators.

## 6.3 MPI Präfix

Die ersten Ziffern im **MPI** können mithilfe eines Präfixes festgelegt werden. Jeder **MPI** enthält damit die angegebene Ziffernfolge (es können nur Zahlen angegeben werden). Ob das Präfix verwendet wird, hängt davon ab, ob der genutzte MPI-Generator das Präfix berücksichtigt. Der mitgelieferte Generator (**EAN13Generator**) berücksichtigt das Präfix. Wird beispielsweise das Präfix 1001 gesetzt, so könnte ein resultierender **MPI** so aussehen: 100100000035. In Listing 6.3 ist dargestellt, wie ein Präfix definiert werden kann.

```
1 <mpi-prefix>1001</mpi-prefix>
```

Listing 6.3: XML-Code zum Definieren des MPI-Präfixes.

## 6.4 Benachrichtigungen

Das Element `use-notifications` dient dazu, bei Änderungen von Datensätzen im **E-PIX** andere Systeme zu benachrichtigen. Diese Benachrichtigungen werden beispielsweise vom **THS-Dispatcher** abgerufen. Mit dem Wert `true` wird die Benachrichtigung aktiviert und mit dem Wert `false` deaktiviert. Sind Benachrichti-

gungen aktiviert, so werden diese versendet, wenn das Web-Interface verwendet wird.

**Hinweis:** Die SOAP-Schnittstelle stellt für die jeweiligen Methoden eine Variante mit und ohne Versendung von Benachrichtigungen bereit. Beim Aufruf einer Methode mit Versenden von Benachrichtigungen, wird in jedem Fall eine Benachrichtigung versendet, auch wenn in der Domänen-Konfiguration dies anders definiert wurde. Die Domänen-Konfiguration bezieht sich hierbei nur auf die Weboberfläche.

**Hinweis:** Der Abruf der Benachrichtigungen erfolgt über einen separaten Dienst, der mit dem E-PIX ausgeliefert wird (ths-notification-service-<version>.war). Die Konfiguration ist in der beiliegenden Anleitung unter /docs/notification-service-<version>-README.pdf beschrieben.

In Tabelle 6.2 sind alle derzeit unterstützten Benachrichtigungen aufgelistet.

Tabelle 6.2: Unterstützte Benachrichtigungen im E-PIX.

Name	Beschreibung
EPIX.AddIdentifierToPersonNotification	Anfügen eines neuen Identifiers an eine Person.
EPIX.AddLocalIdentifierToIdentifierNotification	Anfügen eines neuen lokalen Identifiers an eine Person mit vorhandenen Identifier.
EPIX.UpdatePersonNotification	Aktualisierung von Personendaten.
EPIX.AddPersonNotification	Person hinzugefügt.
EPIX.DeactivatePersonNotification	Person deaktiviert.
EPIX.DeletePersonNotification	Person gelöscht.
EPIX.SetReferencelentityNotification	Identität als Hauptidentität einer Person gesetzt.
EPIX.DeactivateldentityNotification	Identität einer Person deaktiviert.
EPIX.DeleteldentityNotification	Identität einer Person gelöscht.
EPIX.AddContactNotification	Kontaktinformation an eine Person angefügt.
EPIX.MoveldentitiesForIdentifierToPersonNotification	Identitäten einer Person mit dem Identifier an eine andere Person übertragen.

EPIX.AssignIdentity	Mögliche Dublette zusammengeführt.
---------------------	------------------------------------

Im Listing 6.4 ist beispielhaft die Benachrichtigung aktiviert.

```
1 <use-notifications>true</use-notifications>
```

Listing 6.4: XML-Code zum Aktivieren der Benachrichtigungen.

## 6.5 Speicher-Reduktion

Das Element `limit-search-to-reduce-memory-consumption` dient zur Reduzierung der Belegung des Arbeitsspeichers. Diese Option reduziert den benötigten Arbeitsspeicher, schränkt dafür jedoch die Attribute ein, nach denen eine Person gesucht werden kann. Wenn die Option auf `true` gesetzt wird, dann können die Personen nur anhand der Felder gesucht werden, die auch für das Matching (Abschnitt 6.14.6) verwendet werden. In Listing 6.5 wird exemplarisch das deaktivieren dieser Option dargestellt.

```
1 <limit-search-to-reduce-memory-consumption>
2   false
3 </limit-search-to-reduce-memory-consumption>
```

Listing 6.5: XML-Code zum Deaktivieren der Option zur Reduzierung des benötigten Arbeitsspeichers.

## 6.6 Aktualisierungsverhalten

Das Aktualisierungsverhalten bezieht sich auf das Verhalten, wenn Personendaten im E-PIX aktualisiert werden. Dies passiert z.B. über die SOAP-Schnittstelle mittels der Methode `updatePerson`. Sofern die **Matching Attribute** übereinstimmen (andernfalls wird nur eine neue **Identität** angelegt), wird standardmäßig die vorhandene **Identität** überschrieben. Dies führt dazu, dass befüllte Datenfelder mit einem leeren Feld überschrieben werden.

Um **Domänen**-spezifisch dieses Verhalten anzupassen, kann in der **Domänen**-Konfiguration das Verhalten angepasst werden. Dies passiert über das Element `identity-update-behaviour`. Es kann dabei zwischen zwei Verhalten gewählt werden. Wird das Element nicht angegeben, so entspricht dies dem Wert `DEFAULT`, was aufgrund von Abwärtskompatibilitäten wiederum dem Wert `OVERWRITE_ALL` entspricht. In der folgenden Tabelle 6.3 sind alle Werte näher beschrieben.

Tabelle 6.3: Unterstützte Werte für `identity-update-behaviour`

Wert	Beschreibung
DEFAULT	Dies ist der Standardwert und wird intern automatisch auf <code>OVERWRITE_ALL</code> gesetzt. Dies entspricht dem standardmäßigem Verhalten vom E-PIX.
OVERWRITE_ALL	Die übermittelten Daten überschreiben alle Daten im Bestand. Vorhandene Felder werden dabei überschrieben. Enthält der übermittelte Datensatz leere Felder, so überschreiben diese befüllte Felder.
UPDATE_ALL_ADD_MISSING	Die Daten im Bestand, werden mit jenen Daten aus dem übermittelten Datensatz überschrieben. Felder, die im Bestandsdatensatz enthalten sind, jedoch nicht im übermittelten Datensatz, bleiben erhalten. Felder die nur im übermittelten Datensatz enthalten sind, werden im Bestandsdatensatz ergänzt.

**Hinweis:** In Abschnitt 9.5 wird beschrieben, wie über die SOAP-Schnittstelle das Aktualisierungsverhalten bestimmt wird. Diese Angabe hat immer Vorrang vor der Domänen-Konfiguration. Das definierte Aktualisierungsverhalten der Domänen-Konfiguration kommt immer dann zur Anwendung, wenn mittels SOAP-Schnittstelle keine konkrete Angabe oder der DEFAULT-Wert übermittelt wurde.

In Listing 6.7 wird exemplarisch die Festlegung des Aktualisierungsverhaltens dargestellt.

1 `<identity-update-behaviour>DEFAULT</identity-update-behaviour>`

Listing 6.6: XML-Code zum Wählen des Aktualisierungsverhaltens.

## 6.7 Speicher-Modus

Das Element `persist-mode` legt den Modus fest, wie IDAT gespeichert werden. Dabei kann zwischen IDENTIFYING und PRIVACY\_PRESERVING gewählt werden. Standardmäßig wird (wenn dieses Element nicht angegeben wurde) der Modus IDENTIFYING verwendet. Dabei werden alle Daten, die bei der Personenregistrierung übermittelt wurden im E-PIX persistiert. Wird der Modus PRIVACY\_PRESERVING gewählt, werden alle Daten die nicht einem Ziel-Feld eines Bloomfilters entsprechen, entfernt. Die Daten werden zu keiner Zeit persistiert. Ein Record Linkage kann dann nur auf Basis von Bloomfiltern durchgeführt werden. Weitere Informationen zum Bloomfilter sind unter Abschnitt 6.12.1 zu finden. In Listing 6.7 wird

exemplarisch die Festlegung des Persist-Modes dargestellt.

```
1 <persist-mode>IDENTIFYING</persist-mode>
```

Listing 6.7: XML-Code zum Wählen des Persist-Modes.

## 6.8 Pflichtfelder

Mit dem Element `required-fields` kann festgelegt werden, welche Felder für eine Registrierung verpflichtend übermittelt werden müssen. Eine Auflistung der entsprechenden Felder findet über das Element `name` statt. Eine Auflistung der Feldnamen ist in Tabelle 9.1 zu finden. In dem nachfolgenden Listing 6.8 ist exemplarisch eine Konfiguration dargestellt, wodurch zur Registrierung die Felder Vorname, Nachname, Geburtsdatum und Geschlecht übermittelt werden müssen.

```
1 <required-fields>
2   <name>firstName</name>
3   <name>lastName</name>
4   <name>birthDate</name>
5   <name>gender</name>
6 </required-fields>
```

Listing 6.8: XML-Code zur Festlegung der Pflichtfelder, die für eine Registrierung übermittelt werden müssen.

## 6.9 Zusatzfelder

Die Felder `value1` – `value10` können für beliebige Werte verwendet werden. Die entsprechenden Felder können mit sprechenden Namen versehen werden, welche in der Weboberfläche (*Zusatzfelder* in Abschnitt 4.3.2) dargestellt werden. Es handelt sich dabei jedoch nur um ein Label, für etwaige weitere Konfigurationen oder spätere Anfragen über die SOAP-Schnittstelle wird weiterhin der Feldname (also `value` – `value10`) verwendet. In Listing 6.9 wird exemplarisch die Vergabe von Labels für die Felder `value1` und `value2` dargestellt.

```
1 <value-fields-mapping>
2   <value1>KV-Name</value1>
3   <value2>KV-Nummer</value2>
4   <value3>Postleitzahl Hauptwohnsitz</value3>
5   <value8>Bloomfilter Projekt-A</value8>
6 </value-fields-mapping>
```

Listing 6.9: XML-Code zum Definieren von Labels für `value`-Felder.

Im Menüpunkt *Hinzufügen* werden die Zusatzfelder unter dem Abschnitt *Projektdaten* aufgeführt. Die Sortierung ergibt sich anhand des Feldnamens (1-10). In der folgenden Abbildung ist die entsprechende Oberfläche gezeigt, die sich aus dem gezeigten Code-Beispiel ergibt.

Projektdaten	
KV-Nummer	KV-Name
<input type="text"/>	<input type="text"/>
Postleitzahl Hauptwohnsitz	Bloomfilter Projekt-A
<input type="text"/>	<input type="text"/>

## 6.10 Validatoren

Mithilfe von Validatoren können Eingaben im Registrierungsprozess überprüft werden. Eine Registrierung wird dabei abgebrochen, wenn ein Eingabewert nicht die Bedingung zum jeweiligen Feld entspricht. Standardmäßig sind keine Validatoren für die Felder hinterlegt. Das Geburtsdatum und das Geschlecht hingegen, werden immer validiert, da diese im **E-PIX** in einem speziellen Format abgelegt werden. So werden z.B. nur gültige Datumsangaben akzeptiert. Weitere Informationen dazu sind in der Tabelle 9.1 zu finden.

Ein Validator liefert entweder `true`, wenn das Feld den Validierungskriterien entspricht oder `false`, wenn dies nicht der Fall ist. Für jedes Feld können beliebig viele Validatoren angegeben werden. Mehrere Validatoren werden dabei gruppiert und innerhalb dieser Gruppe logisch verknüpft. Mehrere dieser Gruppen können ebenfalls logisch verknüpft werden. Die Operatoren sind in Tabelle 6.4 aufgeführt. Erfüllen alle Felder die jeweils hinterlegten Validierungskriterien, so wird die Registrierung abgeschlossen. Andernfalls liefert der **E-PIX** jene Felder zurück, welche die Validierungskriterien nicht erfüllen. Ein Beispiel ist unten im Listing 6.11 zu finden.

Tabelle 6.4: Operatoren, um Validator-Gruppen miteinander zu verknüpfen.

Verknüpfung	Beschreibung
ALL	Alle Validierungskriterien müssen erfüllt sein.
AT_LEAST_ONE	Mindestens ein Validierungskriterium muss erfüllt sein.
EXACT_ONE	Genau ein Validierungskriterium muss erfüllt sein.
ALL_OR_NONE	Alle oder keines der Validierungskriterien muss erfüllt sein.

Der **E-PIX** definiert bereits einige Validatoren, welche auf konkrete Kriterien zugeschnitten sind. Darüber hinaus werden Validatoren angeboten, die flexibel Kriterien zulassen. In Listing 6.10 ist eine generelle Definition eines Validators eines Feldes dargestellt. Pro Feld, welches validiert werden soll, wird per `validator-config`-Element ein oder mehrere Validatoren bzw. Gruppen hinterlegt. Der *FELDNAME* gibt das Feld an, welches validiert werden soll. Der *VALIDATOR* gibt den Klassennamen des zu verwendenden Validators an (beginnend mit dem Namespace `org.emau.icmvc.ttp.deduplication.impl.validation.`). Mit *PARAMETER* kann das Verhalten des jeweiligen Validators beeinflusst werden.

```

1 <validation>
2   <validator-config>
3     <field>FELDNAME</field>
4     <validator>
5       <qualified-class-name>VALIDATOR</qualified-class-name>
6       <validation-criterion>PARAMETER</validation-criterion>
7     </validator>
8   </validator-config>
9   ...
10 </validation>

```

Listing 6.10: Allgemeiner XML-Code zum Definieren von Validatoren.

In Tabelle 6.5 sind die unterstützten Validatoren aufgeführt.

Tabelle 6.5: Unterstützte Validatoren mit den erforderlichen Parametern.

Validator	Parameter	Beschreibung
Alphabet-Validator	Zeichen, die im Feld akzeptiert werden.	Prüft, ob ein Feld nur Zeichen des angegebenen Alphabets beinhaltet.
Balanced-BloomFilter-Validator	Keiner	Prüft, ob ein Feld den Kriterien eines <b>Balanced Bloomfilter</b> entspricht. Dieser muss Base64 kodiert sein.

Base64- Validator	<i>Keiner</i>	Prüft, ob ein Feld in Base64 kodiert ist.
EGKValidator	<i>Keiner</i>	Prüft, ob das Feld der zehnstelligen <b>KVNR</b> auf der <b>Elektronische Gesundheitskarte (eGK)</b> entspricht.
EMail- Validator	<i>Keiner</i>	Prüft, ob das Feld einer E-Mail Adresse entspricht.
EmptyField- Validator	true, wenn Leerzei- chen ignoriert wer- den sollen, ansonst false.	Prüft, ob ein Feld leer ist. Dabei können wahlweise ignoriert oder berücksichtigt werden.
GermanZipCode- Validator	<i>Keiner</i>	Prüft, ob das Feld eine deutsche Postleitzahl beinhaltet.
Length- Validator	Zahl, welche die er- laubte Länge des Feldes angibt.	Prüft, ob das Feld die angegebene Anzahl an Zeichen aufweist.
RegEx- Validator	Regex- Zeichenkette	Prüft, ob das Feld der Bedingung des angegebenen Regex entspricht.
PhoneNumber- Validator	<i>Keiner</i>	Prüft, ob das Feld eine Telefonnummer enthält. Dabei wird nur geprüft, ob die Zeichen im entsprechendem Alphabet beinhaltet sind. Es wird auf kein spezifisches Format geprüft.

In Listing 6.11 ist ein Beispiel dargestellt, welches für das Feld `value1` eine Validatorgruppe definiert. Das Feld ist valide, wenn es entweder (Operator `EXACT_ONE`) eine **KVNR** beinhaltet oder leer (woebi Leerzeichen erlaubt sind) ist.

```

1  <validator-config>
2      <field>value1</field>
3      <validator-group>
4          <validator>
5              <qualified-class-name>
6                  org.emau.icmvc.ttp.deduplication.impl.
7                      validation.EGKValidator
8              </qualified-class-name>
9          </validator>
10         <validator>
11             <qualified-class-name>
12                 org.emau.icmvc.ttp.deduplication.impl.
13                     validation.EmptyFieldValidator

```

```

14         </qualified-class-name>
15         <validation-criterion>True</validation-criterion>
16     </validator>
17     <link>EXACT_ONE</link>
18   </validator-group>
19 </validator-config>

```

Listing 6.11: XML-Code zum Definieren von einer Validator-Gruppe.

## 6.11 Dublettenauflösungsgründe

Bei einer Dublettenauflösung (Abschnitt 8.5) kann entweder eine Begründung in einem Freitextfeld angegeben werden, oder eine zuvor definierte Begründung ausgewählt werden. Letztere Auswahlmöglichkeiten werden in der Domänen-Konfiguration hinterlegt. Im Element `deduplication` kann hierfür eine Liste von Begründungen angelegt werden, welches im Form eines oder mehrerer `reason`-Elemente stattfindet. Jede Begründung erhält einen Namen und eine kurze Beschreibung. In Listing 6.12 wird exemplarisch eine Dublettenauflösungsbegründung definiert.

```

1 <deduplication>
2   <reason>
3     <name>Tippfehler</name>
4     <description>
5       Vertauschte oder fehlende Zeichen
6     </description>
7   </reason>
8   ...
9 </deduplication>

```

Listing 6.12: Exemplarisches Beispiel zum Anlegen von Dublettenauflösungsbegründungen. Hier am Beispiel der Begründung “Tippfehler” mit einer kurzen erklärenden Beschreibung.

## 6.12 Privatsphäre

Das `privacy`-Element ist ein Container für alle Bloomfilter-Konfigurationen. Der E-PIX unterstützt die Generierung mehrerer Bloomfilter (mittels unterschiedlicher Konfiguration) auf Basis der IDAT. Jeder Bloomfilter besteht dabei aus einem `bloomfilter-config`-Element, welches die jeweilige Konfiguration beinhaltet.

### 6.12.1 Bloomfilter-Konfiguration

Die **Bloomfilter**-Konfiguration enthält alle Einstellungen für einen **Bloomfilter**. Dabei ist zu beachten, dass das Feld in dem der **Bloomfilter** gespeichert wird, die Länge des **Bloomfilters** zulässt (vgl. Tabelle 9.1). Außerdem wird der **Bloomfilter** aus normalisierten bzw. aus aufbereiteten Werten generiert (Abschnitt 6.13). Der **Bloomfilter** kann wie andere Felder auch zum Matching verwendet werden. Hierzu stehen entsprechende Vergleichsverfahren zur Verfügung. Im Abschnitt 6.14.6.6 sind weitere Informationen dazu enthalten. Zu beachten ist, dass **Bloomfilter** im **E-PIX** im **Base64**-Format gespeichert werden.

In der nachfolgenden Tabelle 6.6 sind alle Elemente zur **Bloomfilter**-Konfiguration aufgeführt. Ein Beispiel ist in Listing 6.13 dargestellt.

Tabelle 6.6: Elemente der **Bloomfilter**-Konfiguration.

Element	Beschreibung	Beispiel
algorithm	Angabe des Algorithmus, welcher das Verfahren zur Erzeugung des <b>Bloomfilters</b> implementiert. Eine Auflistung von den unterstützten Algorithmen ist in Tabelle 6.7 zu finden.	org.emau.icmvc.ttp.-deduplication.impl.-bloomfilter.-RandomHashingStrategy
field	Das Feld in das der <b>Bloomfilter</b> gespeichert werden soll. Dabei zu ist beachten, dass das Feld ggf. überschrieben wird und die Länge des <b>Bloomfilters</b> durch das Feld unterstützt werden muss. Obwohl alle Felder grundsätzlich verwendet werden können, wird die Wahl der Value-Felder 6-8 (Tabelle 9.1) empfohlen (je nach Konfiguration).	value8
length	Länge des <b>Bloomfilters</b> in Bits.	1000
ngrams	Länge der N-Gramme, die für die Erzeugung des <b>Bloomfilters</b> verwendet werden. Klassischerweise wird hier ein Wert von 2 angegeben, um Bi-Gramme zu erzeugen.	2

bits-per-ngram	Anzahl der Bits, die pro N-Gramm im <b>Bloomfilter</b> gesetzt werden. Beim Double-Hashing wird von Iterationen gesprochen. Beim Random-Hashing handelt es sich um die Anzahl der generierten Zufallspositionen.	25
fold	Der <b>E-PIX</b> unterstützt ein <b>XOR-Folding</b> von <b>Bloomfiltern</b> nach Schnell et al. <sup>1</sup> . Der Wert gibt die Anzahl der Faltungen an. Zu beachten ist, dass der Wert+1 ein ganzzahliger Teiler von der Länge des <b>Bloomfilters</b> sein muss $n + 1   Laenge$ . Wird 0 angegeben, wird der <b>Bloomfilter</b> nicht gefaltet. Pro Faltung halbiert sich die Länge des resultierenden <b>Bloomfilters</b> .	Bei <b>Bloomfilter</b> der Länge 1000 wären möglich: 0, 1, 3, 4, 7, ...
alphabet	Das Alphabet, welches beim Random-Hashing berücksichtigt werden soll (nur erforderlich, wenn das Random-Hashing verwendet wird).	ABCDEF12345-.
balanced	Der <b>E-PIX</b> unterstützt das Generieren von <b>Balanced Bloomfiltern</b> (Schnell et al. <sup>2</sup> ). Das Element <b>balanced</b> enthält ein Feld <b>seed</b> , welches einen Zahlenwert enthält. Dieser stellt den Seed-Wert des Zufallsgenerators dar. Wird dieses Element ( <b>balanced</b> ) nicht angegeben, wird kein <b>Balanced Bloomfilter</b> erzeugt. Der <b>Balanced Bloomfilter</b> führt zu einer Verdopplung der resultierenden <b>Bloomfilter</b> -Länge.	462945623209

<sup>1</sup> Schnell, Rainer and Borgs, Christian, XOR-Folding for Bloom Filter-based Encryptions for Privacy-preserving Record Linkage (December 22, 2016). German Record Linkage Center, NO. WP-GRLC-2016-03, DECEMBER 22, 2016, Available at SSRN: <https://ssrn.com/abstract=3527984> or <http://dx.doi.org/10.2139/ssrn.3527984>

<sup>2</sup> <https://ieeexplore.ieee.org/document/7836669>

---

source-field Jeder **Bloomfilter** kann aus einem oder mehreren Feldern zusammengesetzt werden. Dabei wird je Feld (Element: `field` (enthält Feldnamen, siehe Tabelle 9.1)) der Wert entsprechend gehashed. Beim Random-Hashing kann pro Feld ein Seed-Wert (Element: `seed` (enthält einen Zahlenwert)) gesetzt werden. Beim Double-Hashing kann ein Salt auf Basis einer statischen Zeichenkette (Element: `salt-value` (enthält eine feste Zeichenkette (z.B.: `a3ghd5o36#sz3`)) oder dynamisch auf Basis eines anderen Feldes (Element: `salt-field` (enthält Feldnamen, siehe Tabelle 9.1)) der **Identität** gesetzt werden.

---

**Hinweis:** Der **E-PIX** unterstützt mehrere Generierungsverfahren und zusätzliche Härtungsverfahren, die kombiniert werden können. Dabei ist zu beachten, dass die **Bloomfilter**-Konfiguration auf die Bedürfnisse des jeweiligen Projekts angepasst werden muss und so mitunter ein Abwegen zwischen Sicherheit und Qualität stattfinden muss. Ein **Bloomfilter** stellt immer eine Verallgemeinerung der Eingangsdaten dar und kann bei falscher Konfiguration zu schlechteren Matching-Resultaten führen, sofern der **Bloomfilter** zum **Record Linkage** genutzt wird. Untersuchungen zeigen, dass **Bloomfilter** zu vergleichbaren Ergebnissen führen, wie der Abgleich von **IDAT**<sup>a</sup>.

---

<sup>a</sup> Randall SM, Ferrante AM, Boyd JH, Bauer JK, Semmens JB. Privacy-preserving record linkage on large real world datasets. *J Biomed Inform.* 2014 Aug;50:205–12.

Der **E-PIX** unterstützt mehrere Verfahren, um **Bloomfilter** zu erzeugen. In der nachfolgenden Tabelle sind alle unterstützten Algorithmen angegeben.

Tabelle 6.7: Unterstützte Algorithmen zur Generierung von Bloomfiltern.

Algorithmus	Beschreibung
org.emau.icmvc.ttp.deduplication.impl.bloomfilter.RandomHashingStrategy	Random Hashing <sup>3</sup>
org.emau.icmvc.ttp.deduplication.impl.bloomfilter.DoubleHashingStrategy	Double Hashing <sup>4</sup>
org.emau.icmvc.ttp.deduplication.impl.bloomfilter.DoubleHashingStrategyFaster	Optimierte Variante vom Double Hashing (Nicht Kompatibel mit DoubleHashingStrategy)

```

1 <privacy>
2   <bloomfilter-config>
3     <algorithm>
4       org.emau.icmvc.ttp.deduplication.impl.bloomfilter.
5         RandomHashingStrategy
6     </algorithm>
7     <field>value8</field>
8     <length>1000</length>
9     <ngrams>2</ngrams>
10    <bits-per-ngram>15</bits-per-ngram>
11    <fold>1</fold>
12    <alphabet>
13      ABCDEFGHIJKLMNOPQRSTUVWXYZ .-0123456789
14    </alphabet>
15    <balanced>
16      <seed>4623829476</seed>
17    </balanced>
18    <source-field>
19      <name>firstName</name>
20      <seed>456542343</seed>
21    </source-field>
22    <source-field>
23      <name>lastName</name>
24      <seed>374027465</seed>
25    </source-field>
26  </bloomfilter-config>
27  <bloomfilter-config>
28    <algorithm>
29      org.emau.icmvc.ttp.deduplication.impl.bloomfilter.
30        DoubleHashingStrategy
31    </algorithm>

```

```

32      <field>value6</field>
33      <length>500</length>
34      <ngrams>2</ngrams>
35      <bits-per-ngram>15</bits-per-ngram>
36      <source-field>
37          <name>firstName</name>
38          <salt-field>birthDate</salt-field>
39      </source-field>
40      <source-field>
41          <name>gender</name>
42          <salt-value>Q2fh-Fk2#CjP+s5#</salt-value>
43      </source-field>
44      </bloomfilter-config>
45  </privacy>

```

Listing 6.13: Verkürzte exemplarische Konfiguration von zwei Bloomfiltern.

## 6.13 Vorverarbeitung

Mithilfe der Vorverarbeitung können Felder aufbereitet werden. Dies ermöglicht beispielsweise, dass für das **Record Linkage** z.B. die Vornamen ohne Berücksichtigung der Groß- und Kleinschreibung miteinander verglichen werden. Eine Vorverarbeitung muss maximal für die Felder durchgeführt werden, die beim **Record Linkage** verwendet werden. Die Felder werden in jedem Fall im unbearbeiteten Zustand, demnach so wie diese übermittelt wurden, im **E-PIX** abgelegt.

Im Element `preprocessing-config` werden alle `preprocessing-fields` aufgelistet. In Listing 6.14 ist ein einfaches Beispiel aufgeführt, welches die Konfiguration zur Aufbereitung des Vornamen-Feldes zeigt. In den folgenden Abschnitten werden die einzelnen Elemente erläutert.

```

1  <preprocessing-config>
2      <preprocessing-field>
3          <field-name>firstName</field-name>
4          <simple-transformation-type
5              xsi:type="ma:SimpleTransformation">
6              <input-pattern> </input-pattern>
7              <output-pattern></output-pattern>
8          </simple-transformation-type>
9          <complex-transformation-type
10             xsi:type="ma:ComplexTransformation">
11             <qualified-class-name>org.emau.icmvcttp.
12                 deduplication.preprocessing.impl.
13                 ToUpperCaseTransformation
14             </qualified-class-name>

```

```
15     </complex-transformation-type>
16   </preprocessing-field>
17 </preprocessing-config>
```

Listing 6.14: Exemplarischer XML-Code mit allen Elementen für eine Vorverarbeitung eines Feldes.

### 6.13.1 Felder

Im Element `preprocessing-field` ist zum einen das betroffene Feld angegeben und alle Transformationen, die für die Aufbereitung eines Feldes verwendet werden sollen. Dabei wird zwischen einfachen und komplexen Transformationen unterschieden, die sich jeweils in ihrer Konfiguration unterscheiden. Eine einfache Transformation stellt ein einfaches Ersetzen dar. Hierbei wird eine bestimmte Zeichenkette in einem Feld gesucht und durch eine andere Zeichenkette ersetzt. Eine komplexe Transformation bezieht sich auf den Inhalt eines gesamten Feldes. Die durchgeführte Operation hängt dabei von der verwendeten Transformation ab.

**Hinweis:** Die Reihenfolge der Transformationen ist nicht sichergestellt und kann von der Reihenfolge der Definition in der XML-Datei abweichen. Es gilt jedoch, dass `complex-transformation-type` stets nach `simple-transformation-type` verarbeitet werden.

### 6.13.2 Feldnamen

Das Element `field-name` gibt das Feld an, welches aufbereitet werden soll. Die Bezeichnungen für die jeweiligen Felder sind in Tabelle 9.1 angegeben.

### 6.13.3 Einfache Transformationen

Mithilfe des Elements `simple-transformation-type` kann eine definierte Zeichenkette durch eine andere ersetzt werden. Hierzu wird mittels des Elements `input-pattern` die Zeichenkette definiert, die ersetzt werden soll. Mit dem Element `output-pattern` kann die Zeichenkette angegeben werden, die eingefügt wird. Diese kann auch leer sein, dann wird die gefundene Zeichenkette nur entfernt. In Listing 6.15 sind zwei `simple-transformation-type` dargestellt. Die erste Transformation dient zum Entfernen von allen Leerzeichen aus einem Feld, die Zweite ersetzt das Zeichen A durch a.

```
1 ...
2 <simple-transformation-type xsi:type="ma:SimpleTransformation">
3   <input-pattern> </input-pattern>
4   <output-pattern></output-pattern>
5 </simple-transformation-type>
```

```

6  <simple-transformation-type xsi:type="ma:SimpleTransformation">
7      <input-pattern>A</input-pattern>
8      <output-pattern>a</output-pattern>
9  </simple-transformation-type>
10 ...

```

Listing 6.15: XML-Code zur Definition zweier einfacher Transformationen.

**Hinweis:** Das Entfernen der definierten Trennzeichen von `multiple-values` (vgl. Abschnitt 6.14.7) führt dazu, dass die Werte nicht mehr voneinander getrennt werden können. Werden bei der Vorverarbeitung z.B. Leerzeichen entfernt, so können im späteren nicht mehr mehrere Vornamen anhand von Leerzeichen entfernt werden.

## 6.13.4 Komplexe Transformationen

Mithilfe des Elements `complex-transformation-type` kann eine Transformation auf ein gesamtes Feld angewendet werden. Dies bedeutet nicht, dass alle Zeichen betroffen sind. Welche Transformation angewendet werden soll, wird mithilfe des Elements `qualified-class-name` angegeben. Die derzeit implementierten Transformationen sind in Tabelle 6.8 genannt und beschrieben. Dabei ist zu beachten, dass bei der Angabe der Transformation immer noch `org.emau.icmvc.ttp.deduplication.preprocessing.impl.` vorangestellt werden muss.

Tabelle 6.8: Unterstützte Transformationen für `complex-transformation-type`.

Transformation	Beschreibung	Beispiel
ToUpperCase- Transformation	Alle Kleinbuchstaben werden durch Großbuchstaben ersetzt.	Anna → ANNA
CharsMutation- Transformation	Ersetzt Umlaute.	München → Muenchen
TrimTrans- formation	Entfernt führende und nachfolgende Leerzeichen.	“ An na ” → “AN NA”
CharNormalization- Transformation	Normalisiert alle Zeichen nach ASCII <sup>5</sup>	â → a é → e

In Listing 6.16 wird exemplarisch gezeigt, wie führende und nachfolgende Leerzeichen für das **Record Linkage** mittels Transformator entfernt werden.

1 ...

<sup>5</sup> [wikipedia.org/wiki/American\\_Standard\\_Code\\_for\\_Information\\_Interchange](https://en.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange)

```

2 <complex-transformation-type xsi:type="ma:ComplexTransformation">
3   <qualified-class-name>
4     org.emau.icmvc.ttp.deduplication.preprocessing.impl.
5       TrimTransformation
6   </qualified-class-name>
7 </complex-transformation-type>
8 ...

```

Listing 6.16: XML-Code zur Definition eines Transformators zum Entfernen führender und nachfolgender Leerzeichen.

### 6.13.5 Filter

Mit dem Element `simple-filter-type` kann ein Alphabet (`pass-alphabet`) bestimmt werden. Alle Zeichen die davon abweichen, werden durch das angegebene Ersatz-Zeichen (`replace-character`) ersetzt. Ist das Ersatz-Zeichen leer, so werden die Zeichen entfernt, die nicht im Alphabet enthalten sind. In Listing 6.17 ist ein einfaches Beispiel zum Entfernen ungültiger Zeichen dargestellt.

```

1 ...
2 <simple-filter-type xsi:type="ma:SimpleFilter">
3   <pass-alphabet>ABCDEFGHIJKLMNPQRSTUVWXYZ</pass-alphabet>
4   <replace-character></replace-character>
5 </simple-filter-type>
6 ...

```

Listing 6.17: XML-Code zur Definition eines Filters, zum Entfernen aller Zeichen, die nicht Teil des Alphabets A-Z sind.

**Hinweis:** Das Entfernen der definierten Trennzeichen von `multiple-values` (vgl. Abschnitt 6.14.7) führt dazu, dass die Werte nicht mehr voneinander getrennt werden können. Werden bei der Vorverarbeitung z.B. Leerzeichen entfernt, so können im späteren nicht mehr mehrere Vornamen anhand von Leerzeichen entfernt werden.

## 6.14 Matching

Das **Record Linkage** wird mithilfe des Elements `matching` konfiguriert. Im **E-PIX** wird das Verfahren von Fellegi-Sunter zur Bestimmung von Wahrscheinlichkeiten verwendet. Hierzu werden die Felder konfiguriert, welche für das **Blocking** und das Matching verwendet werden sollen. Mithilfe von zwei Schwellwerten (`threshold-possible-match` und `threshold-automatic-match`) kann zwischen 4 **Match**-Typen unterschieden werden. In Tabelle 8.1 sind alle **Match**-Typen aufgeführt und entsprechend erläutert. Die Schwellwerte können dem Verfahren

entsprechend angepasst werden. Werden die Elemente nicht angegeben, werden Standardwerte gesetzt. In Tabelle 6.9 sind die empfohlenen und Standard-Schwellwerte dargestellt.

Tabelle 6.9: Schwellwerte für einen **Automatischen Match** und einen **Möglichen Match**.

Schwellwert	Wert (gemäß Standardkonfiguration in Abschnitt 7.1)	Standardwert (wenn nicht angegeben)
threshold-automatic-match	14,5	20
threshold-possible-match	2,99	4

### 6.14.1 Schwellwert für mögliche Matches

Mit dem Element `threshold-possible-match` kann der Schwellwert für einen **Möglichen Match** (vgl. Tabelle 6.9) definiert werden. Überschreitet die ermittelte Ähnlichkeit den angegeben Wert (und unterschreitet den Schwellwert für einen **Automatischen Match** (`threshold-automatic-match`)), so wird der **Match**-Typ **Möglicher Match** als Ergebnis des **Record Linkages** zurückgegeben. In Listing 6.18 ist die Definition des Schwellwert dargestellt.

```
1 <threshold-possible-match>2.99</threshold-possible-match>
```

Listing 6.18: XML-Code zur Definition des Schwellwerts zur Klassifizierung von **Möglichen Matches**.

### 6.14.2 Schwellwert für automatische Matches

Mit dem Element `threshold-automatic-match` kann der Schwellwert für einen **Automatischer Match** (vgl. Tabelle 6.9) definiert werden. Unterscheiden sich die abgeglichenen Datensätze voneinander und die ermittelte Ähnlichkeit liegt jedoch über den angegebenen Wert, so wird der **Match**-Typ **Automatischer Match** als Ergebnis des **Record Linkages** zurückgegeben. In Listing 6.19 ist die Definition des Schwellwert dargestellt.

```
1 <threshold-automatic-match>14.5</threshold-automatic-match>
```

Listing 6.19: XML-Code zur Definition des Schwellwerts zur Klassifizierung von **Automatischen Matches**.

**Info: Wie können automatische Zusammenführungen deaktiviert werden?**

Eine automatische Zusammenführung kann auf **Perfekte Matches**. **mgenitive** beschränkt werden. Fälle mit sehr höher Übereinstimmung, die trotz kleiner Unterschiede zusammengeführt werden würden (**Automatischer Match**), können somit manuell geprüft werden. Hierzu wird der Schwellwert für **threshold-automatic-match** auf 1000 gesetzt. Damit liegt dieser beim internen Wert für **Perfekte Matches**. **mgenitive** und wird so niemals "vor" einem **Perfekter Match** erreicht.

### 6.14.3 CEMFIM

**CEMFIM** steht für *Check Equal Match for Identifier Match* und dient dazu das Matchingergebnis zu beeinflussen. Dabei kann definiert werden, wie sich der **E-PIX** verhalten soll, wenn ein übermittelter Identifier (siehe auch **Lokaler Identifier**) mit dem einer **Identität** übereinstimmt, jedoch mindestens ein **Match** mit einer **Identität** einer anderen Person vorhanden ist. Das Element kann die Werte `true` oder `false` annehmen. Das Verhalten des **E-PIX** kann aus Tabelle 6.10 entnommen werden.

Tabelle 6.10: Verhalten des **E-PIX**, je nachdem wie das Element `use-cemfim` definiert wurde.

CEMFIM	Mehr als 1 Match vorhanden (mit anderer Person)	Verhalten
<code>true</code>	Ja	Fehler: Ein Identifier darf nur einer Person pro <b>Domäne</b> zugeordnet sein.
<code>false</code>	Ja	Die <b>Identität</b> wird gespeichert und als <b>Möglicher Match</b> hinterlegt.
<code>true</code>	Nein	Die <b>Identität</b> wird gespeichert und als <b>Möglicher Match</b> hinterlegt.
<code>false</code>	Nein	Die <b>Identität</b> wird gespeichert und als <b>Möglicher Match</b> hinterlegt.

In Listing 6.20 ist exemplarisch die Definition dargestellt.

<sup>1</sup> `<use-cemfim>true</use-cemfim>`

Listing 6.20: **XML**-Code zur Definition des `use-cemfim`-Wertes.

#### 6.14.4 Paralleles Record Linkage

Der **E-PIX** unterstützt Multithreading, wodurch die Performance gesteigert wird. Bei einer niedrigen Anzahl von registrierten **Identitäten** ist es performanter einen sequenziellen Abgleich durchzuführen. Mit dem Element `parallel-matching-after` kann daher definiert werden, ab wie viel registrierten **Identitäten** ein paralleler Abgleich, also verteilt auf mehrere Threads, stattfinden soll. Der Wert ist abhängig von der Rechenleistung des verwendeten Systems. Bei einem erwarteten Datenbestand von mehreren Tausend registrierten **Identitäten** sollte der Wert nicht zu hoch gewählt werden. Wird der Wert nicht definiert, so wird standardmäßig 1000 gesetzt. In Listing 6.21 ist exemplarisch die Definition dargestellt.

```
1 <parallel-matching-after>1000</parallel-matching-after>
```

Listing 6.21: **XML**-Code zur Definition der Anzahl registrierter Personen, ab denen der **E-PIX** Multithreading verwendet.

#### 6.14.5 Multithreading

Mithilfe des Elements `number-of-threads-for-matching` kann die Anzahl der verwendeten Threads definiert werden. Dabei wird diese in Abhängigkeit des verwendeten Systems eingestellt. Wenn das Element nicht definiert wird, liegt der Wert standardmäßig bei 4 Threads. Je nachdem, wie viele Threads der **E-PIX** verwenden soll, kann der Wert erhöht oder verringert werden. Eine höhere Anzahl von Threads bedeutet, dass im optimalen Fall ein Abgleich von Personen schneller durchgeführt werden kann, da die Vergleiche auf mehrere Threads aufgeteilt werden. Insbesondere bei großen Datenbeständen kann eine Verteilung auf mehrere Threads deutlich performanter sein. In Listing 6.22 ist die exemplarische Definition der Anzahl der verwendeten Threads dargestellt.

```
1 <number-of-threads-for-matching>4</number-of-threads-for-matching>
```

Listing 6.22: **XML**-Code zur Definition der Anzahl der verwendeten Threads.

**Hinweis:** Eine Parallelisierung findet erst statt, wenn die jeweilige **Domäne** die definierte Anzahl an **Identitäten** überschreitet (vgl. Abschnitt 6.14.4).

#### 6.14.6 Matching-Feld

Mit dem Element `field` werden alle Felder definiert, die im Rahmen des **Blockings** oder/und Matchings verwendet werden. Jedes Feld wird hierfür separat konfiguriert. Dabei ist zu beachten, dass wenn nur ein Feld zu Matching genutzt wird, dass das Gewicht auf 100 gesetzt wird. Werden mehrere Felder verwendet, werden die

Felder im Verhältnis ihres Gewichts in die Berechnung einbezogen. In Listing 6.23 ist exemplarisch angegeben, wie eine Konfiguration eines Feldes aussehen kann. Im Folgenden werden die einzelnen Elemente erläutert.

```
1 <field>
2   <name>gender</name>
3   <matching-threshold>0.75</matching-threshold>
4   <weight>3</weight>
5   <algorithm>
6     org.emau.icmvc.ttp.deduplication.impl.LevenshteinAlgorithm
7   </algorithm>
8 </field>
```

Listing 6.23: XML-Code zur exemplarischen Konfiguration eines Felders, welches zum Matching verwendet wird.

#### 6.14.6.1 Feldname

Das Element `name` gibt an, welches Feld für das **Blocking** oder/und Matching verwendet werden soll. Die Bezeichnungen für die jeweiligen Felder sind in Tabelle 9.1 angegeben. In Listing 6.24 ist exemplarisch der Wert `gender` angegeben, wenn das Geschlecht z.B. für das **Blocking** verwendet werden soll.

```
1 <name>gender</name>
```

Listing 6.24: XML-Code zur Definition des Feldes für das **Record Linkage**.

#### 6.14.6.2 Schwellwert für Blocking

Beim **Blocking** wird ein erster Abgleich durchgeführt, um eine erste Selektierung durchzuführen. Die Schwellwerte sollten hierfür niedriger angesetzt werden, damit potentielle **Duplikate** nicht aufgrund eines Abgleichs mit reduzierter Anzahl von abgeglichenen Feldern aussortiert werden. Wird keine entsprechende Schwelle gesetzt, wird standardmäßig der Wert 0.0 gesetzt. Dieser Schwellwert besitzt einen Wertebereich von 0.0 bis 1.0, wobei 0.0 als 0% und 1.0 als 100% zu verstehen ist. In Listing 6.25 wird exemplarisch ein Schwellwert definiert.

```
1 <blocking-threshold>0.8</blocking-threshold>
```

Listing 6.25: XML-Code zur Definition eines Schwellwertes für das **Blocking** von einem Feld.

#### 6.14.6.3 Blocking-Modus

Das **Blocking** unterstützt zwei Datentypen für einen Abgleich zweier Felder. Zum einen **TEXT**, für beliebige Zeichenketten und **NUMBERS** für Zahlen. Letzteres stellt

für Zahlen eine Optimierung dar und ist performanter. Dies kann beispielsweise beim Feld Geburtsdatum (birthDate, vgl. Tabelle 9.1) verwendet werden. Wenn das Element `blocking-mode` nicht angegeben wurde, wird standardmäßig `TEXT` verwendet. In Listing 6.26 ist die Definition von `blocking-mode` exemplarisch für Zahlenvergleiche dargestellt.

```
1 <blocking-mode>NUMBERS</blocking-mode>
```

Listing 6.26: XML-Code zur Definition der **Blocking**-Vergleichsmethode.

#### 6.14.6.4 Schwellwert für Match

Ist beim Matching der ermittelte Wert der Übereinstimmung gleich oder höher dem im Element `matching-threshold` definierten Wert, dann liegt ein Match für das entsprechende Feld vor. Anders als beim **Blocking** sollte der Schwellwert höher angesetzt werden, weil beim Matching nur tatsächliche **Duplikate** ermittelt werden sollen. Trotzdem sollte der Schwellwert genug Raum für etwaige Fehler (z.B. Tippfehler, Zahlendreher) lassen, damit beim Abgleich diese dennoch als **Duplikate** erkannt werden können. Der Schwellwert hängt von dem entsprechenden Feld ab und muss dementsprechend an das Feld angepasst werden. Der Schwellwert besitzt einen Wertebereich von 0.0 bis 1.0, wobei 0.0 als 0% und 1.0 als 100% zu verstehen ist. In Listing 6.27 ist exemplarisch eine Schwelle definiert.

```
1 <matching-threshold>0.8</matching-threshold>
```

Listing 6.27: XML-Code zur Definition eines Schwellwertes für das Matching von einem Feld.

#### 6.14.6.5 Gewichtung für Feld

Mit dem Element `weight` kann eine Gewichtung definiert werden. Damit kann bestimmt werden, wie sehr das Ergebnis eines Vergleichs in das Gesamtergebnis einfließt. Je höher der Wert ist, desto höher gewichtet wird das Feld. Wenn kein Wert angegeben wurde, wird der Wert 1 standardmäßig verwendet. In Listing 6.28 ist exemplarisch eine Gewichtung angegeben.

```
1 <weight>3</weight>
```

Listing 6.28: XML-Code zur Gewichtung eines Feldes.

#### 6.14.6.6 Algorithmus

Der Abgleich der Felder kann mittels unterschiedlicher Verfahren durchgeführt werden. Hierfür wird im Element `algorithm` der Algorithmus eingetragen, welcher für das Matching verwendet werden soll. In Tabelle 6.11 sind alle derzeit unterstützten

Verfahren aufgelistet und erläutert. Bei der Angabe des Algorithmus muss immer ein `org.emau.icmvc.ttp.deduplication.impl.` vorangestellt werden.

Tabelle 6.11: Unterstützte Algorithmen für das Matching.

Algorithmus	Beschreibung
<code>ColognePhoneticAlgorithm</code>	Vergleicht zwei Werte nach ihrem Sprachklang. Die Nachnamen Maier, Meyer und Meier würden beispielsweise als gleich gewertet werden.
<code>DeterministicAlgorithm</code>	Vergleicht zwei Werte auf exakte Gleichheit. Bei exakter Gleichheit zweier Werte ist das Ergebnis 1, bei einer Abweichung 0.
<code>LevenshteinAlgorithm</code>	Vergleicht zwei Werte anhand ihrer Levenshtein-Distanz. Dabei werden durch Einfügen oder Löschen von Zeichen zwei Zeichenketten aneinander angeglichen. Je weniger Operationen nötig sind, desto Ähnlicher sind sich zwei Werte. Dies stellt die empfohlene Methode für das Matching dar und wird standardmäßig verwendet.
<code>SorensenDiceCoefficient-Coded</code>	Vergleicht zwei ( <b>Base64</b> -kodierte) <b>Bloomfilter</b> auf Ähnlichkeit. Dabei wird der Sørensen-Dice Koeffizient verwendet. Dieser Algorithmus wird verwendet, wenn der <b>Bloomfilter</b> direkt im <b>E-PIX</b> erzeugt wurde.
<code>JaccardSimilarity-AlgorithmCoded</code>	Vergleicht zwei ( <b>Base64</b> -kodierte) <b>Bloomfilter</b> auf Ähnlichkeit. Dabei wird der Jaccard-Koeffizient verwendet. Dieser Algorithmus wird verwendet, wenn der <b>Bloomfilter</b> direkt im <b>E-PIX</b> erzeugt wurde.

SorensenDiceCoefficient	Vergleicht zwei (0 und 1 basierte Strings) <b>Bloomfilter</b> auf Ähnlichkeit. Dabei wird der Sørensen-Dice Koeffizient verwendet. Dieser Algorithmus wird verwendet, wenn der <b>Bloomfilter</b> nicht im <b>Base64</b> -Format dem <b>E-PIX</b> übermittelt wird, sondern in einem 0 und 1 basierten String vorliegt.
JaccardSimilarityAlgorithm	Vergleicht zwei (0 und 1 basierte Strings) <b>Bloomfilter</b> auf Ähnlichkeit. Dabei wird der Jaccard-Koeffizient verwendet. Dieser Algorithmus wird verwendet, wenn der <b>Bloomfilter</b> nicht im <b>Base64</b> -Format dem <b>E-PIX</b> übermittelt wird, sondern in einem 0 und 1 basierten String vorliegt.

In Listing 6.29 wird exemplarisch die Definition eines Algorithmus zum Abgleich von einem Feld definiert.

```

1 <algorithm>
2   org.emau.icmvc.ttp.deduplication.impl.LevenshteinAlgorithm
3 </algorithm>

```

Listing 6.29: **XML**-Code zur Definition des Algorithmus für das Matching.

### 6.14.7 Multiple-Value Feld

Der **E-PIX** unterstützt sogenannte Multiple-Value Fields. Hierbei werden Teil-Zeichenketten innerhalb eines Feldes in unterschiedlichen Reihenfolgen abgeglichen. Sind beispielsweise mehrere Vornamen innerhalb des Feldes Vorname angegeben, so werden bei einem Vergleich alle Permutationen der Reihenfolgen abgeglichen. Es wäre somit beispielsweise irrelevant, ob eine Person die Vornamen “Klaus Dieter” oder “Dieter Klaus” angegeben hat. Hierzu kann ein Separator definiert werden, anhand dessen die Teil-Zeichenketten ermittelt werden. In Listing 6.30 ist exemplarisch das Element `multi-values` dargestellt. Die enthaltenen Elemente werden im Folgenden erläutert.

```

1 <multiple-values>
2   <separator> </separator>
3   <penalty-not-a-perfect-match>0.1</penalty-not-a-perfect-match>
4   <penalty-one-short>0.1</penalty-one-short>
5   <penalty-both-short>0.2</penalty-both-short>

```

```
6  </multiple-values>
```

Listing 6.30: XML-Code zur Definition eines `multiple-values`-Feldes.

#### 6.14.7.1 Separator

Mit dem Element `separator` kann ein Zeichen definiert werden, anhand dessen ein Wert in mehrere Zeichenketten aufgespalten wird. Beim Feld Vorname könnte dies beispielsweise ein Leerzeichen sein, sodass sich z.B. aus “Klaus Dieter” die Teil-Zeichenketten “Klaus” und “Dieter” ergeben. Ein Abgleich findet dann unabhängig der Reihenfolge der Teil-Zeichenketten statt. Zu beachten ist, dass nur ein Zeichen als Separator dienen kann. In Listing 6.31 ist die Definition eines Leerzeichens als Separator dargestellt.

```
1  <separator> </separator>
```

Listing 6.31: XML-Code zur exemplarischen Definition eines Leerzeichens als Separator eines `multiple-values`-Feldes.

#### 6.14.7.2 Abzug bei Nicht-Perfect Match

Mit dem Element `penalty-not-a-perfect-match` kann ein Wert definiert werden, der dem Wahrscheinlichkeitswert einer Übereinstimmung abgezogen wird, wenn bei einem Multiple-Value-Feld zwar alle Teil-Zeichenketten eine hinreichende Ähnlichkeit haben, aber nicht exakt gleich sind. Beispiel: “Klaus Dieter” und “Klaas Dieter”. “Klaas” und “Klaus” sind ähnlich genug und haben daher eine hinreichende Ähnlichkeit. Sie sind aber nicht identisch. In Listing 6.32 ist exemplarisch gezeigt, wie ein Wert hierfür definiert wird.

```
1  <penalty-not-a-perfect-match>0.1</penalty-not-a-perfect-match>
```

Listing 6.32: XML-Code zur exemplarischen Definition des `penalty-not-a-perfect-match`-Wertes.

#### 6.14.7.3 Abzug bei einzelnen Übereinstimmungen

Mit dem Element `penalty-one-short` kann ein Wert definiert werden, der dem Wahrscheinlichkeitswert einer Übereinstimmung abgezogen wird, wenn bei einem Multiple-Value-Feld nicht alle Teil-Zeichenketten eine hinreichende Ähnlichkeit aufweisen. Beispiel: “Klaus Dieter” und “Klaus”. “Klaus” ist vorhanden, “Dieter” fehlt jedoch in einem Datensatz. In Listing 6.33 ist exemplarisch gezeigt, wie ein Wert hierfür definiert wird.

```
1 <penalty-one-short>0.1</penalty-one-short>
```

Listing 6.33: XML-Code zur exemplarischen Definition des `penalty-one-short`-Wertes.

#### 6.14.7.4 Abzug bei Teilübereinstimmung

Mit dem Element `penalty-both-short` kann ein Wert definiert werden, der dem Wahrscheinlichkeitswert einer Übereinstimmung abgezogen wird, wenn bei beiden Multiple-Value-Feldern nicht alle Teil-Zeichenketten eine hinreichende Ähnlichkeit aufweisen. Beispiel: “Klaus Dieter” und “Dieter Erhardt”. In Listing 6.34 ist exemplarisch gezeigt, wie ein Wert hierfür definiert wird.

```
1 <penalty-both-short>0.2</penalty-both-short>
```

Listing 6.34: XML-Code zur exemplarischen Definition des `penalty-both-short`-Wertes.

## 7. Anwendungsbeispiele

Jedes Projekt und jedes Forschungsvorhaben haben unterschiedliche Anforderungen bei der technischen Umsetzung zu berücksichtigen. Register, wie das Klinische Krebsregister MV (KKR-MV), verzeichnen alle Krebspatienten aus Mecklenburg-Vorpommern. Hier ist eine besonders hohe Genauigkeit bei der Zusammenführung von Informationen (bei bislang mehr als 400.000 Personen) aus den beteiligten Registerstellen und bei der Identifikation der Personen erforderlich. Jede Abweichung in den demografischen Informationen, sei es nur ein Zeichen, soll dem Treuhandstellenpersonal signalisiert werden und muss einer genauen Prüfung unterzogen werden.

In der NAKO Gesundheitsstudie werden die demografischen Daten der potentiellen Studienteilnehmer von den Meldeämtern abgerufen. Da hier von einer gewissen Grundqualität der Daten auszugehen ist, sind die Schwellwerte deutlich höher als im KKR-MV gewählt. Dies hat zur Folge, dass bei mehr als 2 Mio. eingeschlossenen Personen die nötige manuelle Nacharbeit zum Auflösen von **Möglichen Matches (Dublettenauflösung)**, bei gleichzeitiger Gewährleistung der Qualität, auf ein Mindestmaß reduziert werden konnte.

Beide Beispiele lassen sich problemlos über entsprechende Schwellwerte und Parameter mit Hilfe der **E-PIX** Konfiguration abbilden.

Grundlage der Erkennung der Personen, ist der Matching-Prozess des **E-PIX**. Das beabsichtigte Verhalten (welche Felder sollen wie abgeglichen werden) und die nötige Genauigkeit (wann soll der **E-PIX** entscheiden und wann soll ein **Möglicher Match** signalisiert werden) kann über einer Vielzahl von Schwellwerten und Parametern konfiguriert werden.

Je niedriger die Schwellwerte für einen **Möglicher Match** gewählt wird desto mehr Matching-Paare von Personen werden signalisiert und umso mehr manuelle Kontrolle dieser möglichen Matches durch das Treuhandstellenpersonal ist erforderlich.

Die Konfiguration des **E-PIX** erfolgt je **Domäne**. Um die Vielzahl der Anpassungsmöglichkeiten zu verstehen, werden nachfolgend einige Beispiele im Detail erläutert. Diese können als Grundlage genutzt werden, um projekt-spezifische Anpassungen vorzunehmen. Die einzelnen Matching-Mechanismen und möglichen Konfigurationsoptionen wurden im Kapitel 6 beschrieben.

**Hinweis:** Die Konfiguration des **E-PIX** sollte stets vor produktivem Beginn des Vorhabens erfolgen. Der **E-PIX** entscheidet über den Matching-Zustand einer Person auf Basis der bereits vorhandenen Daten und der aktuellen Konfiguration. Aktualisiert man die Konfiguration bzgl. des Matchings oder der Aufbereitung der Eingabedaten, obwohl bereits Daten in der Datenbank vorhanden sind, müssen diese erneut eingespielt werden (idealerweise in eine leere **Domäne**), um die Korrektheit der Matching-Bewertung gemäß der neuen Konfiguration gewährleisten zu können. Da im produktiven Betrieb keine Änderungen an der Konfiguration vorgesehen sind, kann die **Domänen**-Konfiguration nach erstmalige Personenregistrierung nicht mehr verändert werden.

**Hinweis:** Standardkonfigurationen (mit und ohne **Bloomfilter**) werden beim **E-PIX** mitgeliefert und können als Grundlage für Änderungen oder Erweiterungen verwendet werden. Zu finden sind diese im Verzeichnis `/examples` als `.xml`-Dateien. Zudem befindet sich dort eine Demo-Datenbank (`.sql`) mit exemplarischen Daten.

## 7.1 Standardkonfiguration

Dem **E-PIX** ist eine Standardkonfiguration für Domänen beigelegt. Diese kann für Projekte bereits ohne Anpassungen ausreichend sein. Grundsätzlich gilt, dass während des produktiven Betriebs die Konfiguration nicht mehr geändert werden soll. Für eine korrekte Bewertung des Matchings ist bei einer Änderung der Konfiguration eine komplette Neuregistrierung aller Datensätze erforderlich. Demnach kann die Standardkonfiguration als Grundlage herangezogen werden, sollte jedoch wenn erforderlich durch Projekt-spezifische Parameter angepasst werden.

Die Standardkonfiguration nutzt für das **Record Linkage** die Felder `firstName` (Vorname), `lastName` (Nachname), `birthDate` (Geburtsdatum) und `gender` (Geschlecht). Die Felder `firstName` und `lastName` werden für den Abgleich mittels

Vorverarbeitung (pre-processing) (Abschnitt 4.3.4 oder 6.13) aufbereitet. Diese umfasst das Ersetzen von Zeichen mit Diakritika und Umlauten im Vor- und Nachname. Außerdem werden bekannte Titel oder akademische Grade entfernt. Für das **Blocking** werden die Felder `firstName` und `birthDate` verwendet. Für das Feld `firstName` werden zudem Multiple-Values (Abschnitt 4.3.5 oder 6.14.7) genutzt. Als Trennzeichen wird ein Leerzeichen verwendet, weshalb Leerzeichen nicht bei der Vorverarbeitung entfernt werden. Ein Matching findet mithilfe aller vier Felder statt. Für einen Abgleich wird immer die Levenshtein-Distanz verwendet<sup>1</sup>. In Tabelle 7.1 sind die Felder zur Übersicht dargestellt.

Tabelle 7.1: Felder, Schwellwerte und Wichtungen der Standardkonfiguration.

Feld	Blocking-Schwellwert	Matching-Schwellwert	Wichtung
<code>firstName</code>	0,4	0,8	8
<code>lastName</code>	<i>Kein Blocking</i>	0,8	6
<code>birthDate</code>	0,6	1,0	9
<code>gender</code>	<i>Kein Blocking</i>	0,75	3

Für alle Feldvergleiche wird der Algorithmus zur Berechnung der Levenshtein-Distanz verwendet (Tabelle 6.11). In Tabelle 7.2 sind die Schwellwerte für die automatische Zusammenführung (**Automatischer Match**) und zur Erkennung von **Möglichen Matches**.

Tabelle 7.2: Schwellwerte für automatische und mögliche Matches.

Schwellwert	Wert
<code>threshold-automatic-match</code>	14,5
<code>threshold-possible-match</code>	2,99

Die Standardkonfiguration berücksichtigt *Multiple-Value*-Felder (Abschnitt 6.14.7). Für das Geburtsdatum wird der optimierte Blocking-Modus **NUMBERS** verwendet (Abschnitt 6.14.6.3).

**Info:** Die Weboberfläche hat beim Anlegen einer neuen **Domäne** die Einstellungen der Standardkonfiguration hinterlegt (Abschnitt 4.3). Daher sind die Felder entsprechend vorausgefüllt.

<sup>1</sup> Weitere Vergleichsmöglichkeiten sind implementiert (siehe Tabelle 6.11)

## 7.2 Krebsregister

Die Konfiguration des Krebsregisters MV nutzt für das **Record Linkage** die Felder `firstName` (Vorname), `lastName` (Nachname), `birthDate` (Geburtsdatum), `gender` (Geschlecht) und `value3` (Feld3). Das Feld `value3` enthält die Postleitzahl des Hauptwohnsitzes der registrierten Person. Letzteres wird über ein *Value*-Feld übermittelt, da die Adresse/Kontaktdaten nicht zum Matching verwendet werden können. Es ist daher erforderlich, hierfür ein Freitextfeld zu verwenden und die Postleitzahl zusätzlich dort einzutragen. Die Felder `firstName` und `lastName` werden für den Abgleich mittels Vorverarbeitung (pre-processing) (Abschnitt 4.3.4 oder 6.13) aufbereitet. Für das **Blocking** werden die Felder `firstName` und `birthDate` verwendet. Für das Feld `firstName` werden zudem Multiple-Values (Abschnitt 4.3.5 oder 6.14.7) genutzt. Ein Matching findet mithilfe aller fünf Felder statt. In Tabelle 7.3 sind die Felder zur Übersicht dargestellt.

Tabelle 7.3: Verwendete Felder mit Schwellwerten und Wichtungen im Krebsregister MV.

Feld	Blocking-Schwellwert	Matching-Schwellwert	Wichtung
<code>firstName</code>	0,4	0,8	8
<code>lastName</code>	<i>Kein Blocking</i>	0,8	6
<code>birthDate</code>	0,6	1,0	11
<code>gender</code>	<i>Kein Blocking</i>	0,75	3
<code>value3</code>	<i>Kein Blocking</i>	0,9	5

Für alle Feldvergleiche wird der Algorithmus zur Berechnung der Levenshtein-Distanz verwendet (Tabelle 6.11). In Tabelle 7.4 sind die Schwellwerte für die automatische Zusammenführung (**Automatischer Match**) und zur Erkennung von **Möglichen Matches**.

Tabelle 7.4: Schwellwerte für automatische und mögliche Matches im Krebsregister MV.

Schwellwert	Wert
<code>threshold-automatic-match</code>	1001
<code>threshold-possible-match</code>	3,15

Der Schwellwert für einen **Automatischen Match** wurde mit 1001 so gewählt, dass keine automatischen Zusammenführungen stattfinden. Wird ein **Perfekter Match** erkannt, so liegt der ermittelte Ähnlichkeitswert beim Maximalwert von 1000.

Der Schwellwert für automatische Zusammenführungen kann daher nie erreicht werden (Abschnitt 6.9). Für das Geburtsdatum wird der optimierte Blocking-Modus NUMBERS verwendet (Abschnitt 6.14.6.3).

## 7.3 Matching mit Krankenversicherungsnummern

Der **E-PIX** unterstützt das Matching von **KVNR** der **eGK**. Im Registrierungsprozess kann anhand der enthaltenen Prüfziffer validiert werden, ob die angegebene **KVNR** gültig ist. Es ist zu beachten, dass die vollständige **KVNR** 20 oder 30 Stellen aufweisen kann, wobei diese einen 10-stelligen unveränderlichen Teil aufweist. Dieser bleibt auch beim Wechsel der Krankenkasse der Person erhalten. Die Prüfung bzw. das Matching bezieht sich nur auf diesen Teil, da Datensätze auch dann zusammengeführt werden sollen, wenn zwischenzeitlich ein Wechsel der Krankenkasse erfolgt ist. Da das Matching sich nur auf den 10-stelligen Teil bezieht, wird im Folgendem davon ausgegangen, dass das verglichene Feld nur diesen 10-stelligen Teil beinhaltet. Soll die komplette **KVNR** erfasst werden, so kann dies in einem separaten Feld erfolgen.

Der **E-PIX** bietet Validatoren zur Validierung an (vgl. Abschnitt 4.3.3 oder 6.10). Hierbei steht auch ein Validator zur Prüfung der **KVNR** zur Verfügung<sup>2</sup>. Beim Matching von Identifikatoren wie der **KVNR**, wird auf exakte Übereinstimmung geprüft. Auch wenn zwei **KVNR** sehr ähnlich sind, können diese valide und zu zwei unterschiedlichen Personen gehören. Daher findet ein Abgleich mittels **DeterministicAlgorithm** (vgl. Abschnitt 4.3.5 und 6.14.6.6) statt. Zwei **KVNR** werden somit nur dann als Match gewertet, wenn diese übereinstimmen. Grundsätzlich ist es möglich, noch weitere Matching-Parameter zu definieren.

In Tabelle 7.5 ist die Konfiguration des Abgleichs von der **KVNR** dargestellt. Da in diesem Beispiel nur die **KVNR** verglichen werden soll und diese demnach als einziges Feld verwendet wird, wird der Matching-Schwellwert auf 1,0 festgelegt. Dies stellt einen Sonderfall dar, weil üblicherweise mehrere Felder in die Matching-Berechnung eingezogen werden. Ein Blocking ist nicht erforderlich, da bei nur einem Matching-Feld kein Performance-Gewinn erzielt werden kann.

<sup>2</sup> Dieser prüft nur den 10-stelligen festen Teil der **KVNR**, auch wenn die vollständige **KVNR** angegeben wurde

Tabelle 7.5: Matching mit KVNR mit Schwellwert und Wichtung.

Feld	Matching-Schwellwert	Wichtung
value1	1,0	1

Die Schwellwerte für `threshold-automatic-match` und `threshold-possible-match` können auf den Standardwerten bleiben. Da nur ein Feld verwendet wird und dieses nur bei exakter Übereinstimmung zusammengeführt werden soll, geschieht dies nur bei einem **Perfekten Match**. Alternativ kann `threshold-automatic-match` auf 1001 gesetzt werden, um zu signalisieren, dass nur bei exakter Übereinstimmung eine Zusammenführung erfolgt.

## 7.4 Privacy-Preserving Record Linkage

Der **E-PIX** unterstützt ein **PPRL** mittels **Bloomfilter**. Ein **PPRL** kann in Projekten mit Standort-übergreifenden **Record Linkage** erforderlich sein, damit keine Klartextdaten den Standort verlassen. Hierbei ist zu unterscheiden zwischen einem Daten-liefernden Standort, der auf Basis von lokal vorliegenden **IDAT** einen **Bloomfilter** erzeugt und z.B. einer föderierten Treuhandstelle, welche die **Bloomfiltern** entgegennimmt und miteinander abgleicht.

### 7.4.1 Bloomfilter erzeugen

Der Bloomfilter wird während des Registrierungsprozesses erzeugt. Klassischerweise werden hierbei auch jene Attribute verwendet, die lokal für einen Abgleich herangezogen werden (z.B. Vorname, Nachname, Geburtsdatum, Geschlecht, ggf. weitere Felder). Der **E-PIX** unterstützt zwar mehrere Verfahren zur Generierung (um kompatibel zu anderen Werkzeugen zu sein, siehe Tabelle 6.7), jedoch, sofern es ein Projekt zulässt, wird empfohlen nach aktuellen Verfahren **Bloomfiltern** zu erzeugen. Das Random-Hashing Verfahren wird dem Double-Hashing vorgezogen, erfordert jedoch in jedem Fall eine Abstimmung der **Bloomfilter**-erzeugenden Standorte, um **Bloomfilter** einheitlich zu erzeugen, damit diese vergleichbar sind. Dies ist bei Double-Hashing-Verfahren reduziert auf die Attribute, die codiert werden sollen. Beim Random-Hashing müssen zudem noch einheitliche **Seeds** abgesprochen werden.

Bei der Erzeugung vom **Bloomfilter** wird die vor-verarbeitete **Identität** verwendet. Insbesondere beim Random-Hashing muss darauf geachtet werden, dass die Vorverarbeitung das entsprechend beim Random-Hashing verwendete Alphabet berücksichtigt. Eine gute zusätzliche Härtung des **Bloomfilters** besteht in der

Erzeugung eines **Balanced Bloomfilters**. Hierbei sollte jedoch darauf geachtet werden, dass das Speicherfeld entsprechend lang gewählt wird (siehe auch Tabelle 9.1). Ein **Balanced Bloomfilter** verdoppelt die Bit-Länge des **Bloomfilters**. Die Länge des **Bloomfilters** muss anhand der Anzahl der zu codierenden Attribute und der Anzahl der Bits pro  $n$ -Gramm gewählt werden. Der **E-PIX** unterstützt die Generierung von **Bloomfiltern** pro Attribut oder die Kombination mittels **CLK**. Ist der **Bloomfilter** zu kurz oder die Anzahl der Positionen pro  $n$ -Gramm zu hoch, führt dies im schlimmsten Fall dazu, dass alle Positionen im **Bloomfilter** auf 1 gesetzt werden und damit keine Unterscheidung mehr von unterschiedlichen Datensätzen möglich ist. Eine Länge von 1.000 Bit bei 25-50 Positionen pro Bi-Gramm ( $n = 2$ ) und einer Kombination der Attribute Vorname, Nachname, Geschlecht und Geburtsdatum führt in den meisten Fällen zu zufriedenstellenden Ergebnissen. Eine Wichtung (bzw. unterschiedliche Anzahl von Bit-Positionen) je Attribut ist derzeit im **CLK** nicht möglich.

Es werden weitere Härtungen unterstützt. Mit dem **XOR-Folding** wird der **Bloomfilter** gefaltet. Jede Faltung halbiert die Länge des **Bloomfilters**, kann aber auch die Matching-Qualität verschlechtern. Bei Verwendung oder Kombination mit anderen Härtungsverfahren, sollte geprüft werden, ob ausreichend gute Matching-Resultate anhand eines Test-Datensatzes erzielt werden.

Am **Bloomfilter**-erzeugenden Standort kann und sollte auf Basis der **IDAT** ein **Record Linkage** erfolgen. Ein Abgleich sollte nur erfolgen, wenn nur der **Bloomfilter** zur Verfügung steht (z.B. in übergreifenden Projekten). Der **E-PIX** kann derart betrieben werden, dass dieser nur **Bloomfilter** erzeugt und keine **IDAT** speichert. Der **E-PIX** agiert dann nur als **Bloomfilter**-Generator und das Identitätsmanagement findet in einem anderen Werkzeug oder Instanz statt (siehe Abschnitt 4.3.6 oder 6.7). In diesem Fall muss auch ein Abgleich der **Bloomfilter** konfiguriert werden, damit der **E-PIX** die **Identitäten** unterscheiden kann.

## 7.4.2 Bloomfilter abgleichen

Der Abgleich von **Bloomfiltern** findet unabhängig vom verwendeten Verfahren zur Generierung statt, sodass das Verfahren der abgleichenden Stelle (z.B. der föderierten Treuhandstelle) nicht bekannt sein muss. Werden mehrere **Bloomfilter** erzeugt (z.B. für jedes Attribut), können diese einzeln abgeglichen werden, ähnlich wie andere Attribute abgeglichen werden (siehe Abschnitt 4.3.5 oder 6.14.6). Zu beachten ist, dass der **E-PIX** für **Bloomfilter** entsprechende Vergleichsalgorithmen bereitstellt (siehe Tabelle 6.11). Um **Bloomfilter** auf exakte Übereinstimmung zu vergleichen, kann auch der Algorithmus **DeterministicAlgorithm** verwendet werden.

den, sofern die **Bloomfilter** wie im **E-PIX** üblich, als **Base64**-Zeichenkette verwaltet und übertragen werden.

Werden **CLKs** verwendet, werden diese wie ein Attribut verglichen. Die einzelnen codierten Attribute sind in diesem verschleiert, sodass keine Wichtung einzelner Attribute stattfinden kann. **Bloomfilter** lassen einen Ähnlichkeitsvergleich zu. Damit ist es möglich, dass beim Abgleich ein **Möglicher Match** erzeugt. Da ein manueller Abgleich (**Dublettenauflösung**) mittels **Bloomfilter** nicht möglich ist bzw. auch kein nachgelagerter Prozess dies ermöglichen soll<sup>3</sup>, müssen die Schwellwerte dies entsprechend berücksichtigen.

Die konkrete Umsetzung hängt davon ab, ob automatische Zusammenführungen stattfinden sollen. Sollen die **Bloomfilter** auf **exakte Übereinstimmung** verglichen werden, muss der Matching-Schwellwert (siehe Abschnitt 4.3.5 oder 6.14.6, konkret: 6.14.6.4) auf 1 gesetzt werden (die Wichtung ist bei nur einem Attribut irrelevant), oder der Algorithmus **DeterministicAlgorithm** verwendet werden. Der Schwellwert für einen **Automatischen Match** kann auf 1001 (siehe Abschnitt 6.14.2) gesetzt werden. Der Schwellwert von 1 beim Matching-Schwellwert führt dazu, dass ein Attribut nur bei exakter Übereinstimmung als **Match** gewertet wird. Wenn eine **automatische Zusammenführung** bei hinreichender Ähnlichkeit erfolgen soll, muss der Matching-Schwellwert entsprechend verringert werden (z.B. auf 0.8, sodass bei einer 80%-igen Übereinstimmung der **Bloomfilter** als **Match** gewertet wird). Der Algorithmus **DeterministicAlgorithm** kann hierfür nicht verwendet werden, da dieser nur 1 bei exakter Übereinstimmung und andernfalls 0 zurückliefert! Daher sollte entweder der Algorithmus **SorenseDiceCoefficientCoded** oder **JaccardSimilarityAlgorithmCoded** verwendet werden. Die Schwellwerte für einen **Automatischen Match** und einen **Möglichen Match** sollten identisch sein, um keine **Möglichen Matches** zu erzeugen. Die Schwellwerte müssen dabei so gewählt werden, dass gedultete Abweichungen in einer automatischen Zusammenführung resultieren. Dies hängt sehr stark von den verwendeten Daten und codierten Attributen ab. Zur Ermittlung geeigneter Schwellwerte sollte mittels Test-Datensatz geprüft werden, ob ausreichend gute Matching-Ergebnisse erzielt werden.

<sup>3</sup> Je nach Konzept kann eine föderierte Treuhandstelle einen nachgelagerten Prozess aufweisen, der **nur** für Fälle, die nicht eindeutig aufgelöst werden können, entsprechende **IDAT** nachfordert. Dies erfolgt jedoch in einer getrennten Komponenten, sodass **IDAT** und **Bloomfilter** nicht zusammengeführt werden können. Die **IDAT** werden nach der **Dublettenauflösung** in der föderierten Treuhandstelle gelöscht.

# Bedienung

<b>8</b>	<b>Weboberfläche</b>	<b>86</b>
8.1	Registrierung einer Person	86
8.2	Suchen anhand von Personendaten	89
8.3	Einsehen von Details zu einer Person	90
8.4	Bearbeiten und Löschen von Personendaten	91
8.5	Dublettenauflösung	93
8.6	Daten exportieren	95
8.7	Daten importieren	96
8.8	Einsehen von Protokollen	98
8.9	Statistiken einsehen	99
<b>9</b>	<b>SOAP-Schnittstelle</b>	<b>101</b>
9.1	Registrierung einer Person	101
9.2	Suchen anhand von Personendaten	106
9.3	Suchen anhand von Identifiern	107
9.4	Nachträgliches Erzeugen von Bloomfiltern	109
9.5	Ändern von Personendaten	112

## 8. Weboberfläche

Um dem Treuhandstellenpersonal die Administration der Identitätsdaten zu erleichtern, verfügt der **E-PIX** über eine grafische Benutzeroberfläche, die speziell für den Einsatz im Web-Browser entwickelt wurde. Der Aufbau der Oberfläche orientiert sich an typischen Arbeitsabläufen innerhalb einer Treuhandstelle.

### 8.1 Registrierung einer Person

Bevor eine Person angelegt bzw. registriert werden kann, muss die *Aktive Domäne* ausgewählt werden, für die die Person hinzugefügt wird. Hierzu wird im linken Menü die entsprechende **Domäne** über das Auswahlmenü gewählt. Wenn nur eine **Domäne** angelegt wurde, ist diese standardmäßig aktiv. Über den Menüpunkt *Hinzufügen*, wird ein Formular aufgerufen, in welches die Stammdaten/Personendaten eingetragen werden können. Pflichtfelder sind mit einem Stern (\*) gekennzeichnet. Welche Felder Pflichtfelder sind, wird in der Konfiguration der **Domäne** festgelegt (vgl. Abschnitt 4.3.2). Es können zu jeder Person außerdem noch Adress- bzw. Kontaktdaten und beliebig viele **Lokale Identifier** hinterlegt werden. Weitere Adress- bzw. Kontaktdaten können auf der Detailseite der Person hinzugefügt werden (siehe Abschnitt 8.3). Beim Anlegen können Ein- und ein Auszugsdatum angegeben werden. Die Aktualität einer Adresse kann zusätzlich bearbeitet werden. Mithilfe der **Domäne**-Konfiguration können noch weitere Felder definiert und benannt werden (vgl. Abschnitt 4.3.2). Die **Datenquelle** aus der die Daten stammen muss ebenfalls angegeben werden. Entspricht die angegebene **Datenquelle** der **Sicheren Datenquelle** der jeweiligen **Domäne**, dann wird bei Feststellung eines **Duplikates** die **Identität** als **Hauptidentität** deklariert. Diese gilt dann als fehlerfrei (Änderungen und Fehlerkorrekturen können später trotzdem vorgenommen werden. Grundsätzlich kann die **Hauptidentität** frei gewählt werden). Andernfalls

wird eine neue **Nebenidentität** angelegt. Vor der Registrierung führt der **E-PIX** ein **Record Linkage** durch, welcher ermittelt, ob die Person bereits in dieser oder ähnlichen Form hinterlegt ist. Über das Ergebnis dieses Vorgangs informiert der **E-PIX**. In Abbildung 8.1 wird exemplarisch das Eintragen der Pflichtfelder dargestellt.

**Hinweis:** Jeder **Domäne** wird eine **MPI-Identifier-Domäne** zugeordnet. In diese **Identifier-Domäne** erzeugt der **E-PIX** automatisch die **MPIs**. Daher kann diese nicht für andere **Identifier** ausgewählt werden. Um **Identifier** aus einem anderen System abzubilden, muss zunächst eine entsprechende **Identifier-Domäne** angelegt werden (siehe Abschnitt 4.2), die den Bereich der **Identifier** darstellt (z.B. Fallnummern, Identifikator/ID im **KIS**, usw.).

**Info: Was passiert, wenn ein **Identifier** bei zwei **Identitäten** identisch ist?**

Wenn die beiden **Identitäten** zu einem hohen Grad (konfigurationsabhängig) übereinstimmen, dann werden beide **Identitäten** einer Person zugeordnet. Können die **Identitäten** nicht einer Person zugeordnet werden, weil keine oder nur eine geringe Übereinstimmung vorliegt, so wird ein Fehler zurückgemeldet. Der Grund hierfür ist, dass jeder **Identifier** nur einer Person zugeordnet sein darf (mehrere **Identitäten** können denselben **Identifier** aufweisen, diese müssen dann aber derselben Person zugeordnet sein).

**Info: Was passiert, wenn zwei **Identitäten** identisch sind, aber die **Identifier** aus derselben **Identifier-Domäne** verschieden sind?**

Die **Identifier** werden der bereits vorhandenen **Identität** angefügt. Es können mehrere **Identifier** einer **Identität** angefügt werden, auch wenn diese aus derselben **Identifier-Domäne** stammen (Beispiel: Fallnummern). Voraussetzung ist, dass derselbe **Identifier** niemals unterschiedlichen Personen zugeordnet ist.

Abbildung 8.1: Weboberfläche zum Eintragen von Personendaten.

### Record Linkage und Match-Typen

Bei der Registrierung der Person findet ein Abgleich der **IDAT** statt. Sind diese hinreichend ähnlich zu einer bereits zuvor registrierten Person, so werden diese Personen zusammengeführt. Eine Mitteilung informiert über Erfolg oder Misserfolg. Abhängig von der jeweiligen **Domänen**-Konfiguration unterscheidet man nach einem **Record Linkage** unterschiedliche **Match**-Typen. Diese sind in Tabelle 8.1 dargestellt.

Tabelle 8.1: **Match**-Typen, die Ergebnis vom **Record Linkage** sein können.

Match-Typ	Beschreibung
Perfekter Match	Exakte Übereinstimmung zweier Datensätze in Bezug auf die Matching-Parameter. Es wird keine neue Person und keine neue <b>Identität</b> angelegt, da die <b>IDAT</b> bereits in identischer Form hinterlegt sind.

Automatischer Match / Guter Match	Im Hinblick auf den konfigurierten Schwellwert haben zwei Datensätze eine hinreichende Ähnlichkeit. Die neu angegebenen <b>IDAT</b> werden der bereits bestehenden Person als neue <b>Identität</b> zugeordnet. Je nachdem, ob aus welcher <b>Datenquelle</b> die neue <b>Identität</b> stammt, wird diese als <b>Hauptidentität</b> (siehe auch <b>Sichere Datenquelle</b> ) oder <b>Nebenidentität</b> hinterlegt.
Möglicher Match	Es besteht eine Ähnlichkeit zwischen zwei Datensätzen. Bei einem <b>Möglichen Match</b> findet jedoch keine automatische Zusammenführung statt. Eine <b>Dublettenauflösung</b> kann nur manuell im Nachgang unter Zuhilfenahme weiterer Informationen erfolgen (siehe Abschnitt 8.5).
Kein Match	Keine Ähnlichkeit zu einem bestehenden Datensatz. Wenn kein <b>Duplikat</b> festgestellt wurde, respektive die Person noch nicht bekannt ist, dann wird eine neue Person hinterlegt und die <b>Identität</b> als <b>Hauptidentität</b> angefügt.

## 8.2 Suchen anhand von Personendaten

Unter dem Menüpunkt *Suchen / Bearbeiten* kann nach Personen gesucht werden. Die Suche kann anhand von einem **MPI**, einem **Identifier**, den **IDAT** oder mit Projekt-spezifischen Daten, wie z.B. in der **Domäne** definierten Zusatzfelder (siehe Abschnitt 4.3.2) erfolgen. Hierbei können auch mehrere Felder ausgefüllt werden. Die Suchparameter sind dabei standardmäßig UND-Verknüpft, sodass die Ergebnisliste nur Personen enthält, die alle angegebenen Merkmale aufweisen. Alternativ kann auch eine ODER-Verknüpfung erfolgen, sodass die Ergebnisliste nur Personen aufweist, die zumindest mit einem der angegebenen Merkmale übereinstimmt. Zum Umschalten ist ein Schalter mit der Bezeichnung **Verknüpfung der Suchparameter** vorhanden. In Abbildung 8.2 wird exemplarisch eine Person anhand der Attribute Vorname, Nachname und Geschlecht gesucht. Die Ergebnisliste enthält genau einen Eintrag.

The screenshot shows the E-PIX search interface. The left sidebar includes 'Dashboard', 'Personen' (with 'Dublettenauflösung', 'Suchen / Bearbeiten', and 'Hinzufügen' options), 'Listen' (with 'Import' and 'Export' options), 'Einstellungen' (with 'Domänen / Quellen / Identifier', 'Protokolle', and 'Info' options), and 'Aktive Domäne' (set to 'Demo'). The main area displays a search result for 'Hannah Gasser'. A table header row shows 'Aktionen', 'MPI', 'Vorname', 'Nachname', 'Geschlecht', 'Geburtsdatum', 'Geburtsort', 'Vitalstatus', and 'Block'. Below this is a table row with the data: 'Öffnen' (button), '1001000000264', 'Hannah', 'Gasser', 'Weiblich', '11.09.1985', 'Memmingen', 'Lebendig', and 'jhToGGoaBjwN'. To the left of the table is a 'MPI' input field. To the right is a 'Externer Identifier' section with 'Domäne' and 'Identifier' dropdowns. Below these are 'Stammdaten' and 'Projektdaten' sections, each with various input fields for 'Titel', 'Vorname', 'Nachname', 'Geschlecht', 'Nationalität & Religion', 'Ethnische Zugehörigkeit', 'Religion', and 'Muttersprache'. A 'Sonstige Daten' section is also present.

Abbildung 8.2: Weboberfläche zum Suchen von Personen.

## 8.3 Einsehen von Details zu einer Person

The screenshot shows the E-PIX detail page for 'Hannah Gasser'. The left sidebar is identical to the search interface. The main area starts with a summary box for 'Hannah Gasser' (Weiblich, 38 Jahre, Geboren 11.09.1985 in Memmingen) with a 'Person löschen' button. To the right is a 'Master Identifier (MPI)' section showing '1001000000264' and a 'gPAS' icon. Below this is an 'Externe Identifier' section stating 'Dem E-PIX sind bisher keine Identifier für diese Person in externen Systemen bekannt.' with a 'Externen Identifier hinzufügen' button. The next section is 'Identitäten', which lists known names and allows adding a 'Referenz'. A table shows 'Aktionen', 'Status', 'Vorname', 'Nachname', 'Geschlecht', 'Geburtsdatum', 'Geburtsort', 'Vitalstatus', and 'Block' for the reference 'Hannah Gasser'. A 'Referenz' button is highlighted. Below this is an 'Adressen' section with a table for 'Weilerstraße 34' (PLZ 32425, Stadt Minden, Bundesland Bayern). A 'Adresse hinzufügen' button is at the bottom.

Abbildung 8.3: Detailseite zur Einsicht der hinterlegten Personendaten einer Person.

Um die Detailseite einer Person aufzurufen, muss zunächst nach der betreffenden Person gesucht werden (siehe Abschnitt 8.2). In der Ergebnisliste kann über die Schaltfläche Öffnen die Detailseite zur jeweiligen Person aufgerufen werden.

Neben den **IDAT** können über die Seite die bekannten **Identitäten** eingesehen werden. Darüber hinaus ist eine Auflistung aller bekannten Adressen vorhanden, sowie ein Zeitstrahl mit allen Änderungen, die diese Person betreffen. Wenn parallel auch ein **gPAS** zur Pseudonymverwaltung betrieben wird, kann direkt der Eintrag mit der entsprechenden **MPI** im **gPAS** aufgerufen werden. Änderungen werden ebenso über diese Seite durchgeführt. So lassen sich der Person weitere **Identitäten** oder Adressen hinzufügen. Sind mehrere **Identitäten** zur Person bekannt, so kann im Bereich **Identitäten** die **Hauptidentität** mit der Wahl des Sterns ausgewählt werden. Die **Hauptidentität** wird als **Referenz** markiert. In der Weboberfläche des **E-PIX** werden stets die **IDAT** dieser **Identität** aufgeführt. Einzelne **Identitäten** können in dieser Liste mit der entsprechenden Aktion bearbeitet oder entfernt werden. Existiert zu einer Person nur eine **Identität**, so ist diese automatisch die **Hauptidentität** und kann nicht gelöscht werden. Soll der Personeneintrag aus dem **E-PIX** entfernt werden, kann im oberen Teil die Schaltfläche  **Person löschen** gewählt werden. Das Bearbeiten einer **Identität** führt dazu, dass eine neue **Nebenidentität** angelegt wird. Der bearbeitete Eintrag bleibt demnach erhalten. In Abbildung 8.3 ist exemplarisch die Detailseite einer Person dargestellt.

## 8.4 Bearbeiten und Löschen von Personendaten

Um beispielsweise fehlerhafte Eingaben zu korrigieren oder fehlende Daten zu ergänzen, kann es erforderlich sein, die Personendaten einer Person zu bearbeiten. Hierzu wird zunächst die Detailseite der betreffenden Person aufgerufen (siehe Abschnitt 8.3). Jede **Identität** einer Person kann entsprechend bearbeitet werden. Zur Gewährleistung der Integrität der Daten sollte ein Grund für die Änderungen angegeben werden. Eine Bearbeitung der Personendaten bedeutet, dass im **E-PIX** eine neue **Identität** mit den geänderten Informationen hinzugefügt wird. Daher wird erneut ein **Record Linkage** durchgeführt. In Abbildung 8.4 ist die Oberfläche zum Bearbeiten einer Person abgebildet.

**Info: Was passiert, wenn sich die geänderten IDAT zu sehr von den Vorherigen unterscheiden?**

In diesem Fall teilt der **E-PIX** dies mit einer Fehlermeldung mit. Die geänderten Daten werden dann nicht übernommen. Um dennoch die neuen Daten zu hinterlegen, kann die Checkbox **Neue Identität erzwingen** ausgewählt werden. Dann werden die neuen Daten in jedem Fall der Person zugeordnet.

Da lediglich eine neue **Identität** hinzugefügt wird, müssen die alten bzw. fehlerhaften Personendaten manuell aus der Liste der **Identitäten** entfernt werden. Standardmäßig werden diese **Identitäten** nicht gelöscht, da beispielsweise in ex-

ternen Systemen diese Informationen noch hinterlegt sein könnten und dadurch die Person auch über die zwischenzeitlich geänderten Personendaten noch im **E-PIX** auffindbar sein soll. Das Löschen einer **Identität** ist unwiederbringlich und sorgt dafür, dass jegliche Verweise und Informationen im **E-PIX** hierzu gelöscht werden. Zu jeder Person muss zumindest eine **Identität** vorhanden sein. Sollen alle Personendaten entfernt werden, so muss die Person als ganzes gelöscht werden. Dies beinhaltet auch, dass dazugehörige **MPIs** entfernt werden. Die Schaltfläche zum Löschen von Personen befindet sich im oberen Teil der Detailseite.

Sind bei einer Person mehrere **Identitäten** hinterlegt, kann die gewünschte **Identität** als **Referenz** bzw. **Hauptidentität** ausgewählt werden. Dies kann erforderlich sein, wenn alle Ausprägungen im **E-PIX** hinterlegt sein sollen, jedoch die korrekte Ausprägung von der gesetzten **Hauptidentität** abweicht.

Identität bearbeiten

Wenn Sie Pflichtparameter ändern, wird der E-PIX die Änderung automatisch als neue Identität der aktuellen Person speichern.

**Stammdaten**

Titel

Vorname \* Nachname \*

Hannah Gasser

Geschlecht \*

Weiblich

**Geburt & Tod**

Geburtsdatum \* Geburtsort

11.09.1987 Memmingen

Geburtsname

Vitalstatus

Lebendig

**Projektdaten**

Nationalität & Religion

Nationalität Ethnische Zugehörigkeit

Religion Muttersprache

**Sonstige Daten**

Mittelnname Familienstand

Präfix Suffix

Externes Datum ⓘ

Grund der Bearbeitung Datenquelle \*

Tippfehler im Geburtsjahr dummy\_safe\_source

Speichern

Abbildung 8.4: Weboberfläche zum Bearbeiten der Personendaten.

Zu jeder Person können beliebig viele Adressen verwaltet werden. Dabei kann ein Eintrag als aktuelle Adresse markiert werden. Beim Hinzufügen neuer Einträge wird stets die neuste Adresse als aktuell markiert. Unabhängig davon kann zu jeder Adresse ein Ein- und Auszugsdatum angegeben werden. Vorhandene Einträge können dupliziert und direkt bearbeitet werden. Vorhandene Einträge können entfernt werden.

**Info: Wie können Änderungen an einer Identität anderen Systemen bekannt gemacht werden?**

Dies kann auf zwei Weisen erfolgen. Der E-PIX kann Benachrichtigungen bei Veränderungen versenden. In der Weboberfläche kann dies bei der Einrichtung einer Domäne aktiviert werden (vgl. Abschnitt 4.3.1). Alternativ erfolgt die Aktivierung in der Domäne bei Nutzung einer Konfiguration im XML-Format (vgl. Abschnitt 6.4). Allgemeine Informationen zu Benachrichtigungen, sind im Kapitel 11 aufgeführt.

## 8.5 Dublettenauflösung

Zum Auflösen möglicher **Synonymfehler**, kann unter dem Menüpunkt *Dublettenauflösung* die Liste möglicher Dubletten eingesehen werden. Um einen **Möglichen Match** aufzulösen, wird ein Eintrag aus der Liste angewählt. Beide Personendatensätze werden tabellarisch gegenübergestellt und Unterschiede bei den jeweiligen Feldern farbig hervorgehoben (siehe Abbildung 8.5). So ist eine Entscheidung, ob es sich um ein und dieselbe Person oder zwei unterschiedliche Personen handelt komfortabel möglich. Handelt es sich um zwei Datensätze zu einer Person, wird mit der Schaltfläche **Zusammenführen zur Person 1/2** der jeweilige Datensatz als korrekte Ausprägung ausgewählt. Der jeweils andere Datensatz wird der Person als **Nebenidentität** zugeordnet (dabei bleiben alle etwaigen **Nebenidentitäten** der beiden Personen erhalten). Wenn beide Datensätze zwei unterschiedlichen Personen zugehörig sind, bzw. keine Dublette darstellen, wird über die Schaltfläche **Trennen** ein Ausschluss als potentielle Dublette angegeben. Die Personen bleiben dabei getrennt und die Einträge werden aus der *Dublettenauflösung* entfernt. Für jede *Dublettenauflösung* kann ein entsprechender Kommentar hinterlegt werden, sodass auch später nachvollzogen werden kann, anhand welcher Kriterien die Entscheidung getroffen wurde. Projektspezifische Begründungen können in der **Domänen**-Konfiguration (siehe Abschnitt 4.3.5 bei Nutzung der Weboberfläche oder Abschnitt 6.11 für die Konfiguration im XML-Format) definiert werden und sind dann wählbar. Dies reduziert bei häufig auftretenden Fehlern die Schreibarbeit.

Abbildung 8.5: Gegenüberstellung von Personendaten zum Auflösen einer Dublette.

Sollte eine direkte **Dublettenauflösung** nicht sofort möglich sein, weil beispielsweise zunächst weitere Informationen eingeholt werden müssen, kann die Auflösung zurückgestellt werden (Schaltfläche **Zurückstellen**). Damit wird der Eintrag aus der Liste der offenen Dubletten entfernt. Zurückgestellte Dubletten können über die Schaltfläche **Zurückgestellte anzeigen** eingesehen werden. Beide Listen werden gleichermaßen bedient. Zurückgestellte Dubletten können bei Bedarf wieder als offene Dubletten (Schaltfläche **Als offen markieren**) angezeigt werden. Beide Listen können zudem als CSV-Datei exportiert werden (Schaltfläche **Offene herunterladen**).

Wenn zwei **Identitäten** nicht ähnlich genug sind, um automatisch als Mögliche Dublette erkannt zu werden, kann händisch ein entsprechender Eintrag angelegt werden. Hierzu kann die Schaltfläche **+ Manuell eine Dublette hinzufügen** angewählt werden. Dabei können Dubletten zwischen Personen oder **Identitäten** angegeben werden. Zwischen Personen werden die zugehörigen **MPIs** und bei **Identitäten** die jeweiligen IDs angegeben. Danach erfolgt die Auflösung wie zuvor beschrieben.

## 8.6 Daten exportieren

Die registrierten Personendaten können als CSV-Datei exportiert werden. Hierzu wird unter dem Menüpunkt *Export* der Modus gewählt, anhand dessen die Liste der zu exportierenden Personendaten bestimmt wird. Personendaten können entweder vollständig oder gefiltert nach einer bestimmten **Identifier-Domäne** oder anhand bestimmter **IDAT** exportiert werden. Je nach Modus können verschiedene Optionen gewählt werden. Die zu exportierenden Personendaten werden nach der Anwahl der Schaltfläche **Suchen** in einer Vorschau angezeigt. Dabei können die zu exportierenden Spalten bestimmt werden, indem durch Anwählen des **X** oder **+** die jeweilige Spalte aus- oder einbezogen wird. Außerdem kann die Reihenfolge der Felder des resultierenden Exports durch verschieben der Spalten beeinflusst werden. Die resultierende CSV-Datei wird mit der Anwahl der Schaltfläche **CSV herunterladen** heruntergeladen. Die Spalten in der resultierenden Datei werden standardmäßig mit einem Semikolon separiert. Daher enthält die Datei in der ersten Zeile ein `sep=;`. Falls für den Import (Abschnitt 8.7) andere Separatoren verwendet werden sollen, kann darüber das entsprechende Zeichen angegeben werden. In Abbildung 8.6 wird die entsprechende Oberfläche exemplarisch dargestellt.

The screenshot shows a web-based export interface for person data. At the top, a header reads "Export". Below it, a message says "Information: 4 Personen gefunden." A section titled "Alle Personen exportieren" contains a checked checkbox "Nur den Identifierwert der Hauptidentität exportieren" and a "Suchen" button. The main area is titled "Exportspalten anpassen" and shows a table of 4 rows of data with 11 columns. The columns are: MPI, Titel, Nachname, Geburtsname, Mittelname, Vorname, Geburtsdatum, Geburtsort, Geschlecht, Nationalität, and Mui. The data rows are: 1001000000011 (Meier, Gartenfeld, Max, 01.01.1990, Greifswald, M); 1001000000028 (Maier, Gartenfeld, Max, 01.01.1900, Greifswald, M); 1001000000035 (Mustermann, , Max, 03.12.1980, , M); and 1001000000042 (Musterfrau, , Maria, 17.11.1983, , F). Below the table are buttons "Leere Spalten ausschließen" and "Alle Spalten einschließen". At the bottom, a section titled "Ergebnis speichern" contains a "CSV herunterladen" button.

Abbildung 8.6: Weboberfläche zum Exportieren von Personendaten.

## 8.7 Daten importieren

Um Listen von Personen zu importieren, kann über den Menüpunkt *Import* eine CSV-Datei ausgewählt werden. In Abbildung 8.7 ist die Oberfläche zum Wählen der CSV-Datei dargestellt.

Abbildung 8.7: Weboberfläche zum Importieren von Personendaten.

Ist eine Überschrift in der CSV-Datei enthalten, so kann dies mittels der Checkbox *Datei besitzt eine Kopfzeile mit Spaltennamen* eingestellt werden. In diesem Fall wird die Kopfzeile nicht mitverarbeitet und führt nicht zu einem Eintrag in den Personendaten. Eine Separierung der Spalten erfolgt standardmäßig mit einem Semikolon. Soll ein anderes Trennzeichen verwendet werden, bspw. ein Komma, so kann dies mittels `sep=,` in der ersten Zeile der CSV-Datei definiert werden<sup>1</sup>.

Abbildung 8.8: Weboberfläche mit Vorschau der ersten eingelesenen Zeilen.

<sup>1</sup> Dieser Eintrag wird beim Import nicht als Zeile eingelesen und beeinflusst nicht eine etwaig vorhandene Kopfzeile.

Als Vorschau wird der erste Datensatz aus der Datei dargestellt. Wurden in der CSV-Datei Spaltennamen verwendet, die den Feldnamen des **E-PIX** entsprechen (z.B., weil die CSV-Datei aus dem **E-PIX** exportiert wurde (Abschnitt 8.6)), erfolgt automatisch eine Zuordnung. Sollen die Spalten anderen Feldern zugeordnet werden oder wurden keine Spaltennamen vorgegeben, so kann über das Auswahlmenü jeder Spalte ein beliebiges Feld zugewiesen werden. Welche Spalten importiert werden sollen, kann über die Checkboxen bestimmt werden. Einträge mit dem Wert `null` zeigen an, dass es sich um einen Eintrag mit einem leeren Feld handelt. Nach dem Import sind diese Felder entsprechend leer. In Abbildung 8.8 ist die entsprechende Weboberfläche dargestellt.

Für den Import können weitere Optionen festgelegt werden:

- **Datenquelle:** **Datenquelle** der zu importierenden Daten.
- **Kennzeichnung von Änderungen bei einem Perfekten Match:** Bei einem **Perfekten Match** bei denen Nicht-Matching-Felder<sup>2</sup> geändert werden, werden diese Datensätze gesondert gekennzeichnet.
- **Vorschau ohne Daten zu speichern:** Der Datenbestand wird nicht verändert. Es wird lediglich das erwartete Ergebnis bei einem Import angegeben.
- **Schutz beim Import mit MPI vor ungültigen Updates**<sup>3</sup>: Der **E-PIX** prüft, ob bei identischen **MPIs** die Personendaten von Bestandsdaten und zu importierenden Personendaten übereinstimmen und ähnlich genug sind. Wenn keine hinreichende Ähnlichkeit erzielt wird, werden die Daten nicht importiert, bzw. der Person nicht zugeordnet. Diese Option ist standardmäßig aktiviert. Wird diese Option deaktiviert, dann werden **Identitäten** auch dann einer Person zugeordnet, wenn keine hinreichende Ähnlichkeit mit den vorhandenen **Identitäten** erreicht wird. Die Zusammenführung findet dann nur anhand des **MPI** statt.
- **Datenquelle:** **Datenquelle** der zu importierenden Daten.

<sup>2</sup> Felder, die nicht für das **Record Linkage** berücksichtigt werden.

<sup>3</sup> Diese Option wird nur dargestellt, wenn der **MPI** importiert wird. Andernfalls wird immer ein neuer **MPI** erzeugt.

**Info: Was passiert, wenn Personendaten aus einer Domäne exportiert werden und in eine andere Domäne importiert werden?**

Dies ist möglich. Dabei ist zu beachten, dass die Personendaten nur innerhalb einer Domäne eindeutig sind. Das heißt die Personendaten werden nicht Domäne-übergreifend abgeglichen und entsprechend in jeder Domäne gespeichert. Jedoch müssen die MPIs im E-PIX stets eindeutig sein. Demnach muss beim Import darauf geachtet werden, dass etwaig exportierte MPIs nicht importiert werden, sofern es Überlappungen gibt. Der E-PIX weiß entsprechend darauf hin, sofern MPIs aus anderen Domänen importiert werden. Der E-PIX vergibt neue MPIs, sofern keine MPIs importiert werden.

## 8.8 Einsehen von Protokollen

Um nachzuvollziehen, welche Ereignisse eingetreten sind, kann ein Protokoll unter dem Menüpunkt *Protokolle* eingesehen werden. Es stellt dar, welcher **Match**-Typ (vgl. Tabelle 8.1 durch das **Record Linkage** für die übertragenden Personendaten errechnet wurde (**Kein Match**, **Möglicher Match**, **Automatischer Match**, **Perfekter Match**)). Es gibt zudem Aufschluss darüber, ob Personendaten aktualisiert oder Personen neu angelegt oder **Identitäten** an bestehende Personen angefügt (**Nebenidentitäten**) wurden. In Abbildung 8.9 ist eine exemplarische Auflistung dargestellt.

Das angezeigte Protokoll kann anhand der Ereignisse bzw. Events gefiltert werden. Hierzu werden in der Spalte Ereignis über eine Auswahlliste die darzustellenden Ereignisse des **Record Linkages** angewählt. Zudem können die Zeilen nach einer bestimmten Zeichenkette durchsucht werden. Hierfür steht ein Suchfeld zur Verfügung. Dabei werden nur jene Zeilen aufgelistet, welche die entsprechende Zeichenkette in zumindest einer beliebigen Spalte aufweisen. Zum Öffnen der Detailseite der jeweiligen Person, kann auf den **MPI** geklickt werden.

Das dargestellte Protokoll kann über die Schaltfläche **CSV herunterladen** heruntergeladen werden.

Identitäten Ereignisprotokoll											
Zeitpunkt $\pm \downarrow$	MPI	Ereignis $\downarrow$	Vorname (neu)	Nachname (neu)	Geburtsdatum (neu)	Geschlecht (neu)	Vorname (alt)	Nachname (alt)	Geburtsdatum (alt)	Geschlecht (alt)	
08.02.2024 16:54:41	1001000000028	MERGE (P=14,04) Tipfehler	= Max	Meier	01.01.1990	Männlich	+	Max	Maier	01.01.1900	Männlich
08.02.2024 16:54:03	1001000000059	MERGE (P=2,50) Tipfehler	= Max	Meier	01.01.1990	Männlich	+	Maxi	Maier	01.01.1900	Männlich
08.02.2024 16:53:22	1001000000059	NEW	= Maxi	Maier	01.01.1900	Männlich					
15.03.2022 16:41:43	1001000000042	UPDATE	= Maria	Musterfrau	17.11.1983	Weiblich					
15.03.2022 16:41:43	1001000000035	UPDATE	= Max	Mustermann	03.12.1980	Männlich					
15.03.2022 16:41:43	1001000000028	UPDATE	= Max	Maier	01.01.1900	Männlich					
15.03.2022 16:41:43	1001000000011	UPDATE	= Max	Meier	01.01.1990	Männlich					
29.03.2019 14:14:39	1001000000042	NEW	= Maria	Musterfrau	17.11.1983	Weiblich					
29.03.2019 14:13:41	1001000000035	NEW	= Max	Mustermann	03.12.1980	Männlich					
29.03.2019 14:13:08	1001000000028	NEW	= Max	Maier	01.01.1900	Männlich					

1-10 von 11 |< << 1 2 >> >|

Abbildung 8.9: WeboberOberfläche zum Einsehen des Protokolls.

## 8.9 Statistiken einsehen

Unter dem Menüpunkt *Dashboard* können **Domänen**-spezifische und -übergreifende Statistiken eingesehen werden. Hierbei werden diverse Werte wie die Anzahl von vorhandenen **Möglichen Matches**, registrierte Personen, vorhandene **Identitäten**, aufgelöste Doublets (separat aufgeführt als zusammengeführte und getrennte Personen), usw. gelistet und grafisch aufbereitet dargestellt.

Die Statistik kann als CSV über die jeweiligen Schaltflächen (⬇️) heruntergeladen werden. In Abbildung 8.10 ist das Dashboard gezeigt, welches die Statistiken für eine **Domäne** dargestellt.

Die gezeigten Statistiken werden asynchron, also nicht automatisch und nicht in Echtzeit, generiert. Die Aktualisierung kann jederzeit manuell über die Schaltfläche **⟳ Aktualisieren** angestoßen werden. Die dabei generierten Daten werden durch den **E-PIX** erzeugt und in der Datenbank dokumentiert.

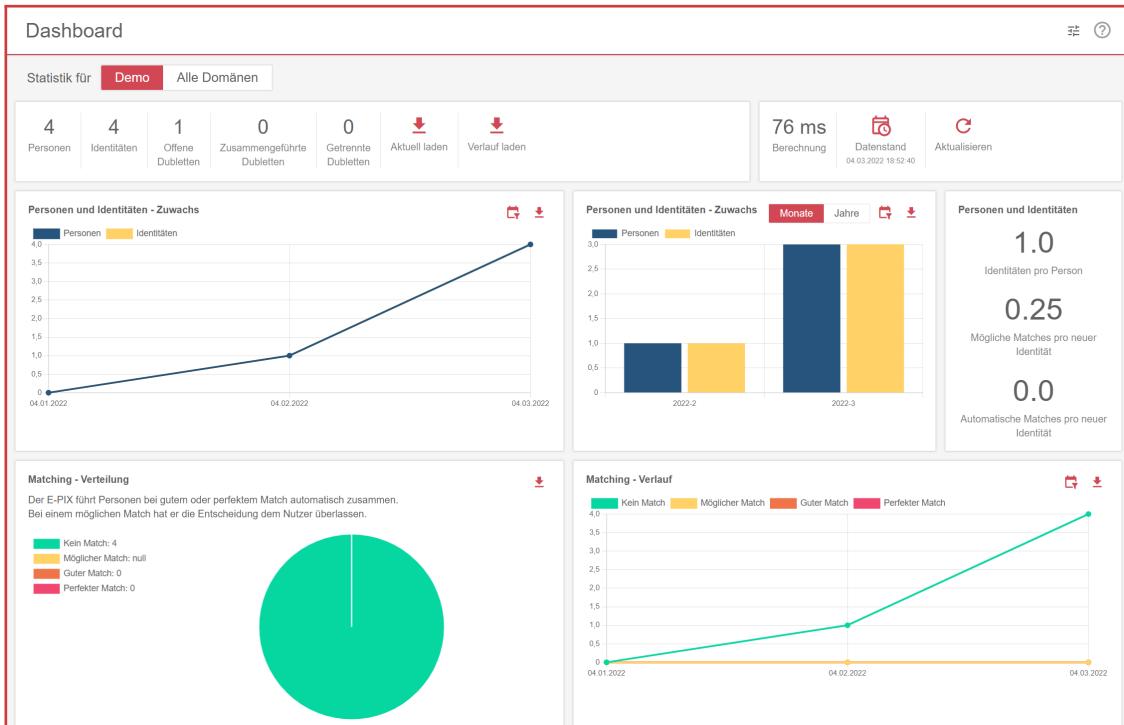


Abbildung 8.10: Dashboard zum Einsehen der Statistiken.

### Info: Unterstützung bei regelmäßiger Community-Kennzahlenerhebung.

Das Dashboard liefert einen schnellen Überblick über Zahlen zu Personen und **Identitäten**. Diese können als CSV-Datei exportiert und der *Unabhängigen Treuhandstelle Greifswald* per E-Mail [kontakt-ths@uni-greifswald.de](mailto:kontakt-ths@uni-greifswald.de) übermittelt werden. Das unterstützt bei statistischen Auswertungen über die Gesamtzahl von Personen und **Identitäten** in der Community. Vielen Dank fürs Mitmachen!

# 9. SOAP-Schnittstelle

## 9.1 Registrierung einer Person

Die Registrierung einer Person über die SOAP-Schnittstelle zur Personenverwaltung (Kapite 5) erfolgt in Abhängigkeit dazu, ob der **E-PIX** ein **Record Linkage** durchführen und **Identitäten** zusammenführen soll, oder nur ablegen soll (beispielsweise, weil das **Record Linkage** bereits in einem anderen System durchgeführt wurde) (*Matching-Mode*: Abschnitt 4.3.5 oder 6.1). Wird der *Matching-Mode* **MATCHING\_IDENTITIES** verwendet, so findet die Registrierung mit der Methode `requestMPI`<sup>1</sup> statt. Dabei führt der **E-PIX** ein **Record Linkage** durch und vergibt eine **MPI**, wenn die Person zuvor noch nicht registriert war. Wenn der *Matching-Mode* **NO\_DECISION** verwendet wird, so findet die Registrierung mit der Methode `addPerson` statt. Dabei führt der **E-PIX** Personen anhand des übergebenen **Identifiers** zusammen.

**Hinweis:** Auch im *Matching-Mode* **NO\_DECISION** wird eine Matching-Konfiguration hinterlegt. Der **E-PIX** prüft anhand dessen, ob die **IDAT** der **Identitäten** mit verschiedenen **Identifier** auch verschiedenen Personen zugeordnet werden würde.

Im Folgenden wird die Registrierung einer Person anhand der Methode `requestMPI` gezeigt. Die Registrierung mit `addPerson` funktioniert analog dazu.

Der **E-PIX** gibt für **Identitäten** verschiedene Felder für die **IDAT** vor. Je nach Feld wird standardmäßig eine formale Prüfung von Eingaben durchgeführt. So würde beispielsweise der 31.02. nicht als Geburtsdatum angenommen werden. Darüber hinaus gibt es Freitextfelder (mit unterschiedlichen Maximal-Längen). In

<sup>1</sup> Bzw. mit `requestMPIBatch` oder `requestMPIWithConfig`

Tabelle 9.1 sind alle vordefinierten Felder aufgelistet.

Tabelle 9.1: Alle im E-PIX definierten Felder.

Feldname	Beschreibung	Beispiel
firstName	Vorname	Anna
middleName	Weitere Vornamen	Lea
lastName	Nachname	Schmidt
birthDate	Geburtsdatum Format: JJJJ-MM-TT <sup>2</sup>	1980-03-12
gender	Geschlecht (wird intern auf mittels eines Buchstabens angegeben) <i>m</i> für male (männlich), <i>f</i> für female (weiblich), <i>o</i> für other (sonstige), <i>u</i> für unknown (unbekannt) und <i>x</i> für divers	f
externalDate	Freies Feld für ein Datum, welches im E-PIX nur gespeichert, aber nicht weiter prozessiert wird Format: JJJJ-MM-TT <sup>3</sup>	2019-04-30
birthPlace	Geburtsort	Berlin
race	Ethnizität	Kaukasier
religion	Religion	Christentum
mothersMaidenName	Geburtsname	Müller
degree	Abschluss	Mittlerer Schulab- schluss
motherTongue	Muttersprache	deutsch
nationality	Nationalität/Staatsangehörigkeit	deutsch
civilStatus	Familienstand	ledig

<sup>2</sup> Betrifft nur SOAP-Schnittstelle

<sup>3</sup> Betrifft nur SOAP-Schnittstelle

---

value1 - value10	Felder dessen Werte je Projekt/ Studie individuell belegt werden können. Zusätzlich kann der Feld- name für die Weboberfläche mit- tels eines Labels geändert werden (Abschnitt 4.3.2 und 6.9). Die Fel- der haben in der Datenbank unter- schiedliche Längen und sind auf diese limitiert:  value1-5: max. 50 Zeichen value6 und 7: max. 255 Zeichen value8 und 9: max. 1.000 Zeichen value10: max. 15.000 Zei- chen	KVNR
prefix	Präfix (Name), Vorsatzwort	von
suffix	Suffix (Name), Namenszusatz	B. Sc.
city	Wohnort (Kontaktdaten)	Berlin
country	Land (Kontaktdaten)	Deutschland
countryCode	Ländercode (Kontaktdaten)	49
district	Bezirk/Stadtteil (Kontaktdaten)	Spandau
email	E-Mail-Adresse (Kontaktdaten)	a.schmidt@bsp.de
externalDate	Freies Feld für ein Datum, welches nur gespeichert, aber nicht weiter prozessiert wird (Kontaktdaten) Format: JJJJ-MM-TT	2019-06-27
municipalityKey	Amtlicher Gemeindeschlüssel (Kontaktdaten)	11000000
phone	Telefonnummer (Kontaktdaten)	030/123 456 789
state	Bundesland (Kontaktdaten)	Berlin
street	Straße (Kontaktdaten)	Spandauer Damm
zipCode	Postleitzahl (Kontaktdaten)	13593
comment	Kommentar	beliebig
vitalStatus	Vitalstatus Unterstützte Werte sind: <i>ALIVE</i> , <i>VE</i> (lebendig), <i>DEAD</i> (verstorben), <i>UNKNOWN</i> (unbekannt)	<i>ALIVE</i>
dateOfDeath	Sterbedatum	2015-03-20

Diese Felder können bei der Registrierung angegeben werden. Dabei ist jedoch zu beachten, dass die Felder der Kontaktdaten nicht für das Matching verwendet werden können. Allerdings können solche Angaben zusätzlich in den Freitextfeldern value1 - value10 übermittelt werden und darüber beim Matching berücksichtigt werden. In Listing 9.1 wird exemplarisch die Registrierung einer Person gezeigt.

---

```
1  <soapenv:Envelope
2      xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
3      xmlns:ser="http://service.epix.ttp.icmvc.emau.org/">
4      <soapenv:Header/>
5      <soapenv:Body>
6          <ser:requestMPI>
7              <domainName>project-a</domainName>
8              <identity>
9                  <birthDate>1990-07-18</birthDate>
10                 <firstName>Anna</firstName>
11                 <lastName>Schmidt</lastName>
12                 <gender>F</gender>
13                 <identifiers>
14                     <identifierDomain>
15                         <name>PID</name>
16                     </identifierDomain>
17                     <value>pid_12345</value>
18                 </identifiers>
19                 <value1>A123456789</value1>
20                 <contacts>
21                     <city>Greifswald</city>
22                     <state>Mecklenburg - Vorpommern</state>
23                     <street>Bahnhofstrasse 3a</street>
24                     <zipCode>17489</zipCode>
25                 </contacts>
26             </identity>
27             <sourceName>data_source</sourceName>
28         </ser:requestMPI>
29     </soapenv:Body>
30 </soapenv:Envelope>
```

---

Listing 9.1: SOAP-Anfrage zur Registrierung einer Person.

Die SOAP-Anfrage muss zumindest alle Felder beinhalten, die in der jeweiligen **Domäne** als Pflichtfelder definiert wurden. **Identifier** werden der **Identität** als **Lokaler-Identifier** hinterlegt. Jeder **Identität** können mehrere Kontakt-Adressen zugeordnet werden. Weitere Adressdaten können mit der Methode `addContact` hinzugefügt werden. Alle weiteren Felder werden in der jeweiligen **Identität** hinterlegt.

Wurde die Person erfolgreich registriert, wird der *HTTP-Code* 200 OK zurückgeliefert. Die SOAP-Antwort enthält Informationen zum **Match**-Status (z.B. **Möglicher Match**) und der vergebenen **MPI**, sowieso der Identity-Id, welche benötigt wird, wenn konkret eine Ausprägung bearbeitet wird (z.B. beim Hinzufügen einer weiteren Adresse per addContact).

Im Vergleich zur Methode `requestMPI` stehen in der Methode `requestMPIWithConfig` über das Element `requestConfig` zwei zusätzliche Parameter zur Verfügung. In Listing 9.2 ist das entsprechende Element exemplarisch aufgeführt. Die Parameter werden in den zwei folgenden Abschritten erläutert.

---

```

1  <soapenv:Envelope
2      xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
3      xmlns:ser="http://service.epix.ttp.icmvc.emau.org/">
4      <soapenv:Header/>
5      <soapenv:Body>
6          <ser:requestMPIWithConfig>
7              ...
8              <requestConfig>
9                  <forceReferenceUpdate>False</forceReferenceUpdate>
10                 <saveAction>DONT_SAVE</saveAction>
11             </requestConfig>
12         </ser:requestMPIWithConfig>
13     </soapenv:Body>
14 </soapenv:Envelope>

```

---

Listing 9.2: SOAP-Anfrage zur Registrierung einer Person mit zusätzlichen Konfigurationsmöglichkeiten. In diesem Beispiel würde die **Identität** nicht gespeichert werden und nur das **Match**-Ergebnis zurückgegeben werden.

### 9.1.1 Aktualisieren der Hauptidentität

Wird bei der Registrierung ein **Match** ermittelt, so kann die zu registrierende **Identität** als **Hauptidentität** hinterlegt werden, auch wenn diese nicht aus der **Sicheren Datenquelle** stammt. Hierzu wird im Element `forceReferenceUpdate` der Wert `true` gewählt. Mit `false` wird die Ermittlung der **Hauptidentität** über den üblichen Weg vorgenommen<sup>4</sup>.

### 9.1.2 Beeinflussung der Persistierung

Mit dem Element `saveAction` kann die Persistierung der zu registrierenden **Identität** beeinflusst werden. So kann beispielsweise ermittelt werden, ob die zu

<sup>4</sup> Dies umfasst z.B. die Herkunft (**Datenquelle**) der **Identität**

registrierende **Identität** einen **Match** erzeugt, ohne die **Identität** zu speichern. Dies ermöglicht einen Abgleich des Datenbestandes, ohne diesen zu ändern. Es werden hierbei verschiedene Modi unterstützt, welche in Tabelle 9.2 aufgelistet sind.

Tabelle 9.2: Verhalten des E-PIX, je nachdem welche Save-Action gewählt wurde.

Save-Action	Beschreibung
SAVE_ALL	Speichert die <b>Identität</b> . Dies ist das standardmäßige Verhalten, wenn z.B. keine zusätzliche Konfiguration hinterlegt wurde. Im Fall eines <b>Perfekter Match</b> wird die vorhandene <b>Identität</b> aktualisiert.
DONT_SAVE_ON_PERFECT_MATCH	Im Fall eines <b>Perfekter Match</b> wird keine Aktualisierung vorgenommen.
DONT_SAVE_ON_PERFECT_MATCH_MINUS_EXCEPT_CONTACTS	Im Fall eines <b>Perfekter Match</b> wird keine Aktualisierung vorgenommen. Etwaig angegebene Kontaktdaten werden der bestehenden <b>Identität</b> angefügt.
DONT_SAVE	Unabhängig vom Ergebnis des <b>Record Linkage</b> werden keine Daten im <b>E-PIX</b> verändert.

## 9.2 Suchen anhand von Personendaten

Die Suche anhand von Personendaten erfolgt über die SOAP-Schnittstelle zur Personenverwaltung (Kapitel 5) mittels der Methode `searchPersonsByPDQ`. Die Suche erfolgt immer innerhalb einer **Domäne**, dessen Name über das Element `domainName` angegeben wird. Über das Element `and` kann angegeben werden, ob alle gesuchten Felder (`true`) oder zumindest ein Feld (`false`) übereinstimmen müssen. Unabhängig von der Angabe des Geburtsdatums (`birthDate`), können Personen anhand des Geburtsjahres (`yearOfBirth`), des Geburtsmonats (`monthOfBirth`) und/oder des Geburtstages (`dayOfBirth`) gesucht werden. Die Suche anhand von **Identifier** ist ebenfalls möglich. Hierfür stehen zudem gesonderte Methoden bereit, um Personen anhand des **MPIs** der **Hauptidentität** (`getPersonByFirstMPI`), anhand eines **MPIs** (`getPersonByMPI`) oder anhand eines **Identifiers** (`getPersonByLocalIdentifier`) zu suchen (Abschnitt 9.3).

**Hinweis:** Adressen/Kontaktdaten werden bei der Suche nicht berücksichtigt, auch wenn diese bei der Suche angegeben werden können.

In Listing 9.3 ist eine exemplarische Suche dargestellt. Es werden alle Personen gefunden, die im Jahr 1983 geboren sind (`1983` in `yearOfBirth`) oder (`false` in `and`) den Nachnamen *Meier* haben (`Meier` in `lastname`).

```
1 <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ser="http://service.epix.ttp.icmvc.emau.org/">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <ser:searchPersonsByPDQ>
5       <searchMask>
6         <and>false</and>
7         <yearOfBirth>1983</yearOfBirth>
8         <domainName>project-a</domainName>
9         <identity>
10           <lastName>Meier</lastName>
11         </identity>
12       </searchMask>
13     </ser:searchPersonsByPDQ>
14   </soapenv:Body>
15 </soapenv:Envelope>
```

Listing 9.3: SOAP-Anfrage zur Suche von Personen anhand von IDAT.

Jede gefundene Person wird innerhalb eines `return` Elements zurückgeliefert. Dabei wird die **Hauptidentität** (auch **Referenzidentität** genannt) im Element `referenceIdentity` aufgeführt. Weitere **Nebenidentitäten** werden im Element `otherIdentities` aufgeführt.

**Hinweis:** Das Element `identity` muss auch dann angegeben werden, wenn keine Elemente davon gesucht werden.

## 9.3 Suchen anhand von Identifiern

Die Suche anhand von **Identifier** erfolgt über die SOAP-Schnittstelle zur Personenverwaltung (Kapitel 5). Hierbei stehen diverse Methoden bereit, welche in Tabelle 9.3 aufgelistet sind.

### Info: Wann erhält eine Identität einen MPI als Lokalen Identifier?

Der **MPI** wird bei der Registrierung einer Person erzeugt und der Person zugeordnet. Bei einer späteren **Dublettenuflösung** können dieser Person weitere **Identitäten** zugeordnet werden. Diese haben jeweils bei der erstmaligen Registrierung bereits einen **MPI** erhalten. Bei einer Zusammenführung werden diesen **Identitäten** die **MPis** als **Lokale Identifier** zugeordnet.

Tabelle 9.3: Methoden zum Abrufen von Personen anhand von Identifiern.

Scope	Methode	Beschreibung
All	getPersonByFirstMPI	Liefert den Personendatensatz zurück, der den angegebenen <b>MPI</b> als <i>First MPI</i> enthält. <b>MPIs</b> , die durch eine <b>Dublettenauflösung</b> als <b>Lokale Identifier</b> angefügt wurden, bleiben hierbei unberücksichtigt. Es werden auch deaktivierte Personen zurückgeliefert, die nach einer <b>Dublettenauflösung</b> keine <b>Identitäten</b> zugeordnet haben.
	getPersonByLocalIdentifier	Liefert die Person, welche den angegebenen <b>Lokalen Identifiers</b> aufweist.
	getPersonByMPI	Liefert die Person anhand eines <b>MPI</b> zurück.
	getPersonByMultipleLocalIdentifier	Liefert eine Person, die alle angegebenen <b>Lokalen Identifier</b> aufweist. Haben verschiedene Personen diese <b>Identifier</b> , so wird ein Fehler geliefert.
	getPersonsByFirstMPIBatch	Wie <code>getPersonByFirstMPI</code> , es können mehrere <b>MPIs</b> angegeben werden, um mehrere Personen innerhalb einer Anfrage abzurufen.
	getPersonsByMPIBatch	Wie <code>getPersonByMPI</code> , es können mehrere <b>MPIs</b> angegeben werden, um mehrere Personen innerhalb einer Anfrage abzurufen.
Active	getActivePersonByLocalIdentifier	Liefert die aktive Person anhand eines <b>Lokalen Identifiers</b> zurück.
	getActivePersonByMPI	Liefert die aktive Person anhand eines <b>MPI</b> zurück.
	getActivePersonByMultipleLocalIdentifier	Liefert die aktive Person anhand mehrerer <b>Lokale Identifier</b> zurück. Haben verschiedene Personen diese <b>Identifier</b> , so wird ein Fehler geliefert.
	getActivePersonsByMPIBatch	Wie <code>getActivePersonByMPI</code> , es können mehrere <b>MPIs</b> angegeben werden, um mehrere aktive Personen innerhalb einer Anfrage abzurufen.

**Info: Was ist eine “aktive” bzw. *active* Person?**

Bei einer Zusammenführung durch eine **Dublettenuflösung** wird die **Identität** ausgewählt, welche der korrekten Ausprägung entspricht. Die jeweilige Person bleibt als Datensatz erhalten (auch Dublettengewinner) und erhält alle **Identitäten** als **Nebenidentitäten** der anderen Person zugeordnet. Diese **Identitäten** erhalten den **MPI** als **Lokalen Identifier** zugeordnet. Die Person nun ohne zugeordnete **Identitäten** (auch Dubettenverlierer) wird deaktiviert, bleibt jedoch erhalten und kann über den **MPI** gefunden werden. Werden explizit nur aktive Personen gesucht und dabei der **MPI** einer deaktivierten Person verwendet, so wird die Person zurückgeliefert, die bei einer **Dublettenuflösung** als Dublettengewinner hervorgegangen ist. Der **First MPI** der gelieferten Person weicht damit vom eigentlich gesuchten **MPI** ab. Diese Person enthält dann jedoch jene **Identitäten**, die zuvor der Dubettenverlierer Person zugeordnet waren und den gesuchten **MPI** als **Lokalen Identifier** zugeordnet haben.

**Info: Was ist der **First MPI**?**

Der **First MPI** wird bei der erstmaligen Registrierung einer Person vergeben. Wird die **Identität** durch eine **Dublettenuflösung** an eine andere Person angefügt, so wird die ursprüngliche Person deaktiviert, behält aber den **First MPI**. Die **Identität** erhält den **MPI** als **Lokalen Identifier**. Wird konkret nach dem **First MPI** gesucht, so werden nur Personendatensätze geliefert, welche den gesuchten **MPI** als **First MPI** hinterlegt hat. **Identitäten** die den **MPI** als **Lokaler Identifier** hinterlegt haben, bleiben in diesem Fall unberücksichtigt.

## 9.4 Nachträgliches Erzeugen von Bloomfiltern

Optimalerweise findet die finale Konfiguration einer **Domäne** vor der ersten Registrierung statt. Der **E-PIX** lässt eine Änderung der Konfiguration nicht über die Oberfläche zu, da der Datenbestand stets die Anforderungen der Matching-Konfiguration erfüllen muss. Bei Bestandsprojekten kann es vorkommen, dass die Konfiguration ergänzt werden muss. Für Datenbestände die nachträglich auch für Standort-übergreifende Zusammenführungen zur Verfügung gestellt werden sollen, kann z.B. die Ergänzung von **Bloomfiltern** erforderlich sein. Grundsätzlich kann hierfür auch eine separate **Domäne** angelegt, eine entsprechende Konfiguration hinterlegt und die Datensätze in diese **Domäne** importiert werden. Dies kann aber zu zusätzlichen Aufwänden bzgl. Dublettenuflösung führen. Daher wird hier beschrieben, wie eine **Bloomfilter**-Konfiguration nachträglich eingespielt werden kann und nachträglich **Bloomfilter** generiert werden können.

## 1. Anpassen der Datenbank

Sobald zumindest eine Registrierung in die **Domäne** erfolgt ist, kann die Konfiguration nicht mehr über die Weboberfläche verändert werden. Dies kann nur noch direkt über die Datenbank erreicht werden, indem der entsprechende Eintrag verändert wird und dann der **E-PIX** neu gestartet wird. Hierzu muss in der Datenbank die Tabelle `domain` ausgewählt werden und der entsprechende Eintrag zur **Domäne** herausgesucht werden. In dem Eintrag muss die Spalte `config` angepasst werden. Die Konfiguration liegt im **XML**-Format vor (siehe auch Kapitel 6). Stets direkt vor dem Element `preprocessing-config` muss das Element `privacy` eingefügt werden. Dieses beinhaltet ein oder mehrere Konfigurationen für **Bloomfilter**. Die Konfiguration ist in Kapitel 6.12.1 beschrieben. Im folgenden Listing 9.4 ist eine exemplarische Konfiguration dargestellt.

```

1  <privacy>
2      <bloomfilter-config>
3          <algorithm>
4              org.emau.icmvc.ttp.deduplication.impl.
5              bloomfilter.RandomHashingStrategy
6          </algorithm>
7          <field>value8</field>
8          <length>1000</length>
9          <ngrams>2</ngrams>
10         <bits-per-ngram>15</bits-per-ngram>
11         <fold>1</fold>
12         <alphabet>
13             ABCDEFGHIJKLMNOPQRSTUVWXYZ . - 0123456789
14         </alphabet>
15         <balanced>
16             <seed>4623829476</seed>
17         </balanced>
18         <source-field>
19             <name>firstName</name>
20             <seed>456542343</seed>
21         </source-field>
22         <source-field>
23             <name>lastName</name>
24             <seed>374027465</seed>
25         </source-field>
26     </bloomfilter-config>
27 </privacy>

```

Listing 9.4: Exemplarische Konfiguration eines **Bloomfilters**.

## 2. Neustarten des Dienstes

Nachdem die Datenbank angepasst wurde, muss der Dienst neu gestartet werden,

damit die neue Konfiguration geladen wird. Der Datenbestand wird dabei nicht verändert. Bei neuen Registrierungen greift nun jedoch die **Bloomfilter**-Konfiguration. Neue Datensätze erhalten demnach bereits einen **Bloomfilter**.

Je nach Größe des Bestands der jeweiligen **Domäne** und den technischen Spezifikationen des Systems, kann das nachträgliche Erzeugen der **Bloomfilter** höhere Laufzeiten verursachen. Es kann daher erforderlich sein zu diesem Zweck die Umgebungsvariablen des WildFlys anzupassen. Dies ist dann zu empfehlen, wenn beispielsweise über 100.000 **Bloomfilter** erzeugt werden sollen. Hierzu können in der Datei `./envs/wf_commons.env` die Variablen `WF_BLOCKING_TIMEOUT` und `WF_TRANSACTION_TIMEOUT` entsprechend angepasst werden<sup>5</sup>. Standardmäßig werden dadurch laufende Prozeduren nach einer gewissen Zeit abgebrochen.

### 3. Bloomfilter für Bestandsdaten erzeugen

Nachdem der Dienst vollständig hochgefahren ist, können die **Bloomfilter** erzeugt werden. Dies ist nur über die SOAP-Schnittstelle möglich. Hierzu wird die Methode `updatePrivacy` verwendet. Mit `domainName` wird der Name der **Domäne** angegeben. Mit dem Element `onlyReferenceIdentity` kann mit `true` angegeben werden, dass nur der **Hauptidentität** ein **Bloomfilter** zugeordnet werden soll. Wird `false` angegeben, wird allen **Identitäten** ein **Bloomfilter** zugeordnet. Mit Listing 9.5 wird allen **Identitäten** ein **Bloomfilter** zugeordnet.

---

```
1 <soapenv:Envelope
2   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
3   xmlns:ser="http://service.epix.ttp.icmvc.emau.org/">
4   <soapenv:Header/>
5   <soapenv:Body>
6     <ser:updatePrivacy>
7       <domainName>project-a</domainName>
8       <onlyReferenceIdentity>false</onlyReferenceIdentity>
9     </ser:updatePrivacy>
10    </soapenv:Body>
11 </soapenv:Envelope>
```

---

Listing 9.5: SOAP-Anfrage zum nachträglichen Erzeugen von **Bloomfiltern**. Dabei wird allen **Identität** ein **Bloomfilter** zugeordnet.

Alternativ kann auch eine Liste mit **MPIs** übermittelt werden. Dann wird nur den Personen mit der jeweiligen **MPI** ein **Bloomfilter** zugeordnet. Hier wird mit dem

<sup>5</sup> Eine detaillierte Beschreibung aller Variablen ist zu finden unter: <https://hub.docker.com/r/mosaicgreifswald/wildfly/>

Element `mpiIds` ein oder mehrere **MPIs** angegeben. In Listing 9.6 ist dies entsprechend mit exemplarisch drei **MPIs** dargestellt.

---

```

1  <soapenv:Envelope
2      xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
3      xmlns:ser="http://service.epix.ttp.icmvc.emau.org/">
4      <soapenv:Header/>
5      <soapenv:Body>
6          <ser:updatePrivacy>
7              <domainName>project-a</domainName>
8              <mpiIds>1001000000011</mpiIds>
9              <mpiIds>1001000000042</mpiIds>
10             <mpiIds>1001000000059</mpiIds>
11             <onlyReferenceIdentity>false</onlyReferenceIdentity>
12         </ser:updatePrivacy>
13     </soapenv:Body>
14 </soapenv:Envelope>

```

---

Listing 9.6: SOAP-Anfrage zum nachträglichen Erzeugen von **Bloomfiltern**. Dabei wird nur den Personen mit den entsprechenden **MPIs** ein **Bloomfilter** zugeordnet.

#### 4. Optional: Anpassungen am WildFly rückgängig machen

Wurden Änderungen an den Umgebungsvariablen vorgenommen, sollten diese in der `./envs/wf_commons.env` wieder rückgängig gemacht werden.

## 9.5 Ändern von Personendaten

Personendaten können mittels der Methode `updatePerson` geändert werden. Dabei ist jedoch zu beachten, dass ein reguläres **Record Linkage** durchgeführt wird. Wenn die **Matching Attribute** (nur) hohe Ähnlichkeiten aufweisen, dann wird eine neue **Identität** mit den übermittelten Daten angelegt. Stimmen alle **Matching Attribute** überein, so wird diese **Identität** aktualisiert. Standardmäßig werden hierbei alle Daten überschrieben. Das führt auch dazu, dass Felder, die im übermittelten Datensatz leer sind, etwaig beschriebene Felder im Bestandsdatensatz überschreiben. Im Resultat ist dieses Feld dann leer/gelöscht.

Über die SOAP-Schnittstelle kann dieses standardmäßige Verhalten geändert werden. Hier kann mit dem Element `updateBehaviour` das entsprechende Verhalten bestimmt werden. In der folgenden Tabelle 9.4 werden die unterstützten Werte aufgelistet und beschrieben.

Tabelle 9.4: Unterstützte Werte für `updateBehaviour`

Wert	Beschreibung
DEFAULT	Wenn in der Domänen-Konfiguration nichts anderes festgelegt wurde, so entspricht das Aktualisierungsverhalten OVERWRITE_ALL. Ansonsten wird das Aktualisierungsverhalten der Domänen-Konfiguration übernommen (siehe Abschnitt 6.6).
OVERWRITE_ALL	Die übermittelten Daten überschreiben alle Daten im Bestand. Vorhandene Felder werden dabei überschrieben. Enthält der übermittelte Datensatz leere Felder, so überschreiben diese befüllte Felder.
UPDATE_ALL_ADD_MISSING	Die Daten im Bestand, werden mit jenen Daten aus dem übermittelten Datensatz überschrieben. Felder, die im Bestandsdatensatz enthalten sind, jedoch nicht im übermittelten Datensatz, bleiben erhalten. Felder die nur im übermittelten Datensatz enthalten sind, werden im Bestandsdatensatz ergänzt.

# IV Integration

<b>10</b>	<b>Logging</b>	115
<b>11</b>	<b>Benachrichtigungen</b>	116
<b>12</b>	<b>FHIR-Unterstützung</b>	117
<b>13</b>	<b>Authentifizierung &amp; Autorisierung</b>	119
13.1	Global	119
13.2	Domänen-spezifische Rollen mit OpenID-Connect	120
<b>14</b>	<b>Empfehlungen zur Absicherung</b>	122
<b>15</b>	<b>Optimierungen</b>	123
15.1	Optimierungen bei Multi-Millionen Beständen	123
15.2	Optimierungen bei Betrieb ohne Docker	124

## 10. Logging

**Hinweis:** Details für die Anpassung der Logging-Konfiguration sind in der beigelegten Beschreibung `README\_E-PIX.md` (Abschnitt Logging) beschrieben.

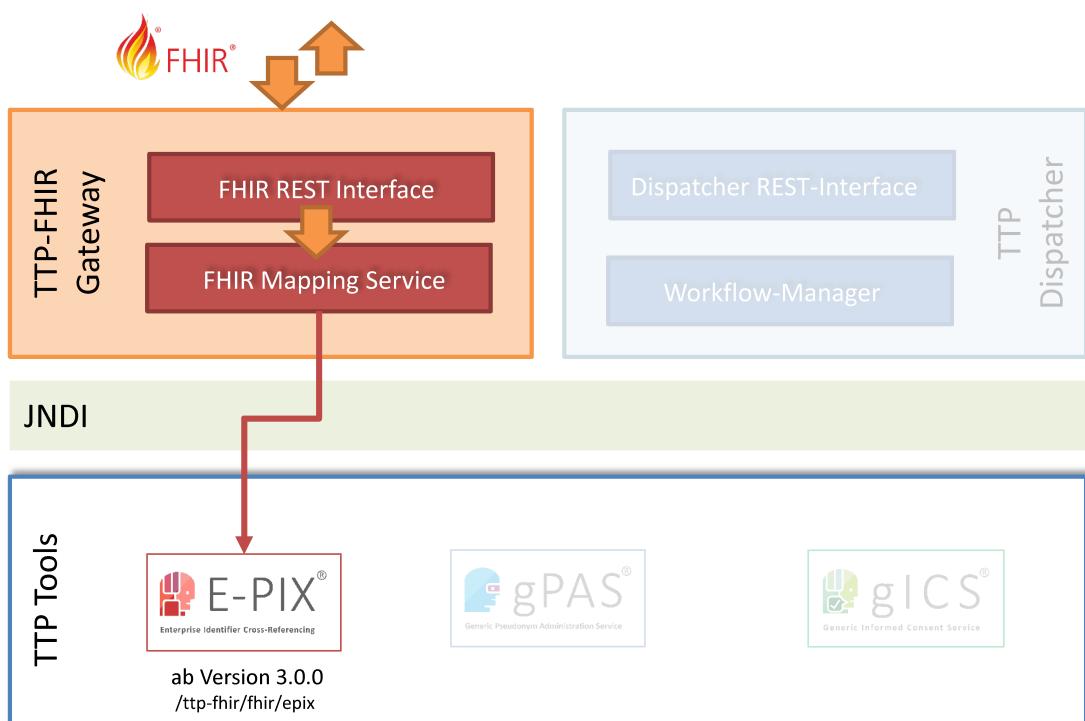
## 11. Benachrichtigungen

Bei Veränderungen im **E-PIX** (z.B. bei Registrierung einer Person oder Änderungen von Personendaten) kann dieser Benachrichtigungen an externe Systeme versenden. Dies kann per `http`, `MQTT` oder `EJB` erfolgen. Die Benachrichtigungen werden in einer separaten Notification-Datenbank im Notification-Dienst dokumentiert. Im **E-PIX** kann das Versenden von Benachrichtigungen pro **Domäne** konfiguriert werden. In Abschnitt 4.3.1 ist die Aktivierung bei der **Domänen**-Konfiguration über die Weboberfläche erläutert. Im **XML**-Format ist dies im Abschnitt 6.4 dargestellt. Unabhängig davon bietet SOAP-Schnittstelle die Möglichkeit, unabhängig von der Konfiguration, Benachrichtigungen zu versenden.

**Hinweis:** Der Abruf der Benachrichtigungen erfolgt über einen separaten Dienst, der mit dem **E-PIX** ausgeliefert wird (`ths-notification-service-<version>.war`). Die Konfiguration ist in der beiliegenden Anleitung unter `/docs/notification-service-<version>-README.pdf` beschrieben.

## 12. FHIR-Unterstützung

„Fast Healthcare Interoperability Resources (kurz: FHIR®) ist ein von HL7 erarbeiteter Standard. Dieser unterstützt den Datenaustausch zwischen Softwaresystemen im Gesundheitswesen. FHIR beschreibt Datenformate und Elemente als sogenannte „Ressourcen“ und bietet eine Schnittstelle an, um diese auszutauschen“<sup>1</sup>.



© Independent Trusted Third Party Greifswald 2022

Um sowohl bestehende Anwenderprojekte als auch künftige Nutzer bei der Umsetzung FHIR-orientierter Infrastrukturen und Prozesse zu unterstützen, wird ein

<sup>1</sup> [https://de.wikipedia.org/wiki/Fast\\_Healthcare\\_Interoperability\\_Resources](https://de.wikipedia.org/wiki/Fast_Healthcare_Interoperability_Resources)

zusätzliches Treuhandstellen-FHIR-Gateway (kurz: TTP-FHIR Gateway) als Mittler von FHIR-spezifischen Infrastrukturkomponenten und **E-PIX** bereitgestellt.

**Hinweis:** Da der **E-PIX** als Daten-haltendes System sämtliche **IDAT** und **Pseudonyme** erster Stufe (**MPI**) verwaltet, ist der **E-PIX** auch für die Generierung und Verwaltung der erforderlichen FHIR-UUIDs verantwortlich.

Für ausgewählte Funktionalitäten zum Anlegen von Personendaten in FHIR wurden nachfolgende Funktionen umgesetzt und sind nach erfolgreichem Deployment des TTP-FHIR Gateways direkt per REST nutzbar. Der aktuelle Funktionsumfang (FHIR-Operations) des TTP-FHIR Gateway umfasst:

- Anlegen von Personendaten
- Aktualisieren von Personendaten

Darüber hinaus gibt es eine Vielzahl von Suchfunktionen. Weitere Funktionalitäten werden sukzessive implementiert und bereitgestellt. Der zugehörige Implementation Guide mit konkreten Beispielen ist zu finden unter <https://www.ths-greifswald.de/e-pix/fhir>.

**Hinweis:** Die Profilierung der erforderlichen Profile, Codesysteme und Operations erfolgte in Zusammenarbeit mit der Fa. Gefyra<sup>a</sup>.

<sup>a</sup> <https://www.gefyra.de/>



# 13. Authentifizierung & Autorisierung

## 13.1 Global

Der **E-PIX** bietet unterschiedliche Umsetzungsoptionen der Authentifizierung und Autorisierung sowohl in der Docker- als auch in der Docker-Compose-Variante.

Standardmäßig ist im **E-PIX** keine Authentifizierung notwendig. Soll der **E-PIX** nur für bestimmte Nutzergruppen (Admin-Nutzer, Standard-Nutzer) zugänglich gemacht werden (vgl. Tabelle 13.1) oder das Anlegen von neuen **Domäne** beschränkt werden, stehen dafür zwei Authentifizierungsverfahren bereit. *gRAS* und *KeyCloak*, wobei es für *KeyCloak* zwei verschiedene Varianten gibt. Die Verwendung von *KeyCloak* wird empfohlen.

**Hinweis:** Die Rollen-spezifische **Domänen**-Absicherung ist unter <https://www.ths-greifswald.de/ttp-tools/domain-auth> oder in der beiliegenden `/docs/TT P-Tools-Domain-Roles.md` beschrieben.

**Hinweis:** Alle THS-Schnittstellen (Weboberfläche, FHIR-Gateway und SOAP-Webservices) können je Endpunkt und somit je Werkzeug (**E-PIX**, **gICS**, **gPAS**) mit *KeyCloak*-basierter (und damit OIDC-konformer) Authentifizierung abgesichert werden. Die Konfiguration der Authentifizierung erfolgt in der Docker-Compose Version innerhalb der `ttp\_epix.env`. Eine detaillierte Beschreibung ist unter <https://www.ths-greifswald.de/ttp-tools/keycloak> oder in der beiliegenden `/docs/TTP-Tools-Keycloak-Einrichtung.md` verfügbar.

### 13.1.1 Übersicht Nutzerrollen und Rechte

Tabelle 13.1: Nutzer-Zugriffsrechte in der Weboberfläche.

Bereich/Seite	Zugang ohne Login	Zugang mit User-Rechten	Zugang mit Admin-Rechten
Info	×	×	×
Dashboard		×	×
Administration: Domänen			×
Administration: Protokolle		×	×
Administration: Statistik		×	×
Personen: Dublettenauflösung		×	×
Personen: Suche / Bearbeiten		×	×
Personen: Hinzufügen		×	×
Listen: Import			×
Listen: Export			×

### 13.1.2 Verwendung von KeyCloak

Die Client-seitige *KeyCloak*-Konfiguration kann sowohl per Konfigurationsdatei als auch per Environment-Variablen bei Start des Docker-Compose erfolgen.

**Hinweis:** Details können unter <https://www.ths-greifswald.de/ttp-tools/keycloak> oder aus der beiliegenden /docs/TPP-Tools-Keycloak-Einrichtung.md entnommen werden.

Neben der Absicherung der Weboberfläche gibt es die Möglichkeit, die SOAP-Schnittstelle per KeyCloak abzusichern. Hierfür wird ähnlich wie bei der Weboberfläche in Zugriffsrechte für Admin und User unterschieden.

### 13.1.3 Verwendung von gRAS

**Hinweis:** Details können unter <https://www.ths-greifswald.de/ttp-tools/gras> oder aus der beiliegenden /docs/gRAS-Einrichtung.md entnommen werden.

## 13.2 Domänen-spezifische Rollen mit OpenID-Connect

Mit der rollenbasiertes **Domänen**-Absicherung können einzelne **Domänen** für authentifizierte Benutzer, basierend auf den ihnen zugeordneten Rollen, ein- bzw. ausgeblendet werden. So werden über spezielle Rollen die **Domänen** beschrieben,

auf die der Zugriff erlaubt sein soll. Alle anderen **Domänen** werden “ausgeblendet” bzw. sind nicht zugänglich.

Als Paradigma wird dabei die transparente “Perspektive” (oder “View”) verwendet: Anfragen zur **Domänen**-Auflistung werden nur mit den **Domänen** beantwortet, zu denen es eine Autorisierung gibt. Zugriffsversuche auf andere **Domänen** werden so beantwortet, als gäbe es diese nicht. So ist es einem Nutzer auch nicht möglich, durch gezielte Anfragen herauszufinden, welche weiteren **Domänen** in der Instanz vorhanden sind.

Die “Filterung” der **Domänen** erfolgt im Backend, so dass die Zugriffe über SOAP und das Weboberfläche entsprechend eingeschränkt werden, sofern diese authentifiziert und mit aktivierter rollenbasierter **Domänen**-Absicherung erfolgen.

Das zweistufige Rollensystem mit Admin- und User-Rollen (vgl. Abschnitt 13.1) bleibt von rollenbasierter **Domänen**-Absicherung unberührt und ist komplementär dazu.

**Hinweis:** Weitere inhaltliche Erläuterungen zur Verwendung und Konfiguration der **Domänen**-spezifischen Rollen und Rechte sind separat unter <https://www.ths-reifswald.de/ttp-tools/domain-auth> dokumentiert.

## 14. Empfehlungen zur Absicherung

Der Zugriff auf relevante Anwendungs- und Datenbankserver des **E-PIX** sollte nur für autorisiertes Personal und über autorisierte Endgeräte möglich sein. Wir empfehlen die Umsetzung nachfolgender IT-Sicherheitsmaßnahmen:

- Betrieb der relevanten Server in separaten Netzwerkzonen (getrennt von Forschungs- und Versorgungsnetz)
- Verwendung von Firewalls und IP-Filtern
- Verwendung von Keycloak (siehe auch Kapitel 13)
- Zugangsbeschränkung auf URL-Ebene mit Basic Authentication (z.B. mit NGINX oder Apache)

# 15. Optimierungen

## 15.1 Optimierungen bei Multi-Millionen Beständen

Bei Datenbeständen mit mehreren Millionen zu verwaltenden Personen, können in Abhängigkeit der Leistungsfähigkeit der verwendeten Hardware, höhere Laufzeiten entstehen. Dies kann es erforderlich machen, weitere Anpassungen vorzunehmen. Diese sollten aber ausdrücklich erst dann vorgenommen werden, wenn entsprechende Datenbestände erreicht oder erwartet werden. Dies umfasst beispielsweise das Hochsetzen von Timeouts, was nur bedingt durch den Datenbestand sinnvoll ist, aber nicht grundsätzlich.

### 1. Wert für Timeout in der Datenbank erhöhen

Bei großen Datenmengen können die standardmäßigen Zeiten bis zum Auslösen von Timeouts zu niedrig sein. Treten diese auf, so können diese in der Datenbank erhöht werden. Hier wird muss die (Datenbank-)Servervariable `innodb_lock_wait_timeout` erhöht werden. Standardmäßig liegt diese bei 50 Sekunden.

### 2. Werte für Timeout des WildFly Applikationsservers erhöhen

Wenn der Start eines Deployments zu lange dauert (standardmäßig mehr als 5 Min.), dann wird ein Timeout ausgelöst. Beim **E-PIX** kann das passieren, wenn der Datenbestand groß ist und nicht schnell genug alle Daten aus der Datenbank in den Cache geladen werden können. Dieser Abschnitt kann hierzu in die Konfiguration des Applikationsservers WildFly eingefügt und der Wert angepasst werden:

```
1 <system-properties>
```

```
2   <property name="jboss.as.management.blocking.timeout"
3     value="DAUER_IN_SEKUNDEN" />
4 </system-properties>
```

Gleiches gilt für die Deployment-Dauer (standardmäßig 60 Sekunden). Folgende bereits vorhandene Konfiguration muss dafür angepasst werden:

```
1 <subsystem xmlns="urn:jboss:domain:deployment-scanner:2.0">
2   <deployment-scanner deployment-timeout="DAUER_IN_SEKUNDEN" ...>
3 </subsystem>
```

## 15.2 Optimierungen bei Betrieb ohne Docker

Wird entgegen der hier beschrieben Vorgehensweise selbst ein Applikationsserver und Datenbankserver aufgesetzt, so kann eine Performance-Steigerung des **E-PIX** durch diverse Optimierungen erzielt werden. In den von der Treuhandstelle Greifswald ausgelieferten Docker Containern (WildFly und MySQL) sind diese bereits vorkonfiguriert. Diese Optimierungen sind relevant, wenn größere Datenbestände mit mehreren Zehn-Tausend Personen erwartet werden.

### 15.2.1 Speicher für MySQL erhöhen

Standardmäßig ist im MySQL-Server eine `innodb_buffer_pool_size` von 128 MB eingestellt. Es wird empfohlen diese auf 2 GB zu erhöhen. Dies geschieht entweder direkt in der Datenbank oder bei der Verwendung eines Docker-Containers als entsprechendes Kommando. Bei der Konfiguration dieses Wertes ist die offizielle MySQL-Dokumentation (<https://dev.mysql.com/doc/refman/5.7/en/innodb-buffer-pool-resize.html>) zu beachten. Die Anpassung dieses Wertes erfolgt unter Beachtung des verfügbaren Arbeitsspeichers.

### 15.2.2 Batch-Writing

Für jede Datenbankoperation (Insert, Update, Delete) wird standardmäßig separat auf die Datenbank zugegriffen. Zur Steigerung der Performance können die Anfragen jedoch zusammengefasst werden. Dies kann erreicht werden, indem in der `standalone.xml` vom WildFly der Parameter `rewriteBatchedStatements=true` an die `jdbc-connection-url` angefügt wird.

### 15.2.3 Lange Zeiten zum Hochfahren des Applikationsservers

Wurden viele Millionen Personen angelegt und ein Neustart des Systems ist erforderlich, so kann das Hochfahren des Applikationsservers WildFly mehr Zeit in Anspruch nehmen, als das konfigurierte Timeout zulässt. Das Timeout wird

standardmäßig nach 5 Minuten ausgelöst, sofern der WildFly bis dahin nicht hochgefahren ist. Es ist dann erforderlich, die Konfiguration des WildFly abzupassen. Hierzu wird in der `standalone.xml` des WildFly-Servers die Komponente `deployment-scanner` um das Attribut `deployment-timeout` ergänzt. Der Wert des Attributes gibt die Zeit in Sekunden an, ab wann ein Timeout ausgelöst wird. Im folgenden Beispiel wird das Timeout auf 15 Minuten (900 Sekunden) hoch gesetzt.

```
1 <subsystem xmlns="urn:jboss:domain:deployment-scanner:2.0">
2   <deployment-scanner [...] scan-interval="5000"
3     deployment-timeout="900" [...] />
4 </subsystem>
```

# Weitere Literatur

## Publikationen

1. Bialke M, Bahls T, Havemann C, Piegsa J, Weitmann K, Wegner T und Hoffmann W. MOSAIC – A Modular Approach to Data Management in Epidemiological Studies. *Methods Inf Med.* 2015; 54:364–71. DOI: [10.3414/ME14-01-0133](https://doi.org/10.3414/ME14-01-0133)
2. Bialke M, Langner D, Geidel L, Bahls T, Havemann C und Piegsa J. Who Am I? And If so, How Many? The E-PIX as Innovative System to Manage Person Identities. Paper Presented at: 2nd Data Management Workshop. Band 2014. 2014
3. Bialke M, Penndorf P, Wegner T, Bahls T, Havemann C, Piegsa J und Hoffmann W. A Workflow-Driven Approach to Integrate Generic Software Modules in a Trusted Third Party. *Journal of Translational Medicine.* 2015 Jun 4; 13. DOI: [ARTN17610.1186/s12967-015-0545-6](https://doi.org/10.1186/s12967-015-0545-6)
4. Hampf C, Geidel L, Zerbe N, Bialke M, Stahl D, Blumentritt A, Bahls T, Hufnagl P und Hoffmann W. Assessment of Scalability and Performance of the Record Linkage Tool E-PIX((R)) in Managing Multi-Million Patients in Research Projects at a Large University Hospital in Germany. *Journal of Translational Medicine.* 2020 Feb 17; 18:86. DOI: [10.1186/s12967-020-02257-4](https://doi.org/10.1186/s12967-020-02257-4)

# Glossar

**Balanced Bloomfilter** Ein Balanced Bloomfilter stellt ein Härtungsverfahren von Bloomfiltern dar, bei dem ein Bloomfilter eine invertierte Kopie angefügt bekommt und die Bit-Positionen anhand eines bekannten Seeds zufällig vertauscht werden. Dies sorgt dafür, dass das Heimming-Weigth immer bei 1 liegt<sup>1</sup>.

**Base64** Base64 ist eine Kodierung zur Übertragung binärer Inhalte. Dabei wird der zu übermittelnde Inhalt in die Zeichen A-Z, a-z, 0-9, +, / überführt und besteht damit nur aus lesbaren Zeichen. Der Inhalt kann unabhängig von der binären Darstellung verschiedener Computersysteme übermittelt werden. Dies kann auch mittels Text-basierter Übertragung wie z.B. per E-Mail erfolgen.

**Blocking** Verfahren um zwei Identitäten anhand einer Teilmenge von Attributen zu vergleichen. Wird dabei eine hinreichende Ähnlichkeit erreicht, können weitere Attribute zum Vergleich verwendet werden (siehe [Record Linkage](#)).

**Bloomfilter** Ein Bloomfilter ist ein Hashing Verfahren, bei dem meist auf Basis von Hashfunktionen die Bit-Positionen eines Bit-Vektors auf 1 gesetzt werden. Bloomfilter können im Vergleich zu vielen anderen Hash-Methoden miteinander auf Ähnlichkeiten verglichen werden. Ähnliche Bloomfilter haben dabei auch ähnliche Eingabewerte Zugrunde. Dabei können keine direkten Rückschlüsse auf die Eingabewerte gezogen werden. Härtungsverfahren sorgen dafür, dass Versuche die zugrundeliegenden Eingabewerte zu ermitteln, erschwert oder verhindert werden.

**Cryptographic Long Term Key** Ein Cryptographic Long Term Key ist ein Härtungsverfahren von Bloomfiltern. Dabei werden mehrere Attribute in einen Bloomfilter codiert. Die Rückschlüsse auf die Eingabedaten eines Attributes werden damit erschwert<sup>2</sup>.

**Datenbankmanagementsystem** Ein Datenbankmanagementsystem ist eine Software zur Verwaltung von Datenbanken und deren Daten. Ein DBMS ermöglicht einem Benutzer meistens die CRUD Operationen auf die Daten anzuwenden. Dies umfasst das Anlegen (**Create**), das Lesen (**Read**), das

---

<sup>1</sup> Schnell R, Borgs C, editors. Randomized Response and Balanced Bloom Filters for Privacy Preserving Record Linkage. 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW); 2016 12-15 Dec. 2016.

<sup>2</sup> Schnell R, Bachteler T, Reiher J. A novel error-tolerant anonymous linking code. 2011. German RLC Working Paper, German Record Linkage Center.

Aktualisieren (**Update**) und das Löschen (**Delete**).

**Datenquelle** Die Datenquelle ist die namentliche Nennung der Quelle, aus denen **IDAT** stammen können, z.B. ein Krankenhaus, ein Forschungsprojekt, eine Abteilung oder ein System (z.B. **KIS**). Eine tatsächliche Verknüpfung findet dabei nicht statt, sondern dient nur der Dokumentation und hat Einfluss auf die Bewertung beim **Record Linkage**. Bei der Registrierung einer **Identität** wird die Quelle ausgewählt, aus der die jeweiligen **IDAT** stammen. Jeder **Domäne** kann eine **Sichere Datenquelle** zugeordnet werden. **IDAT** die über die **Sichere Datenquelle** registriert werden, werden als korrekte Ausprägung einer **Identität** angesehen (**Hauptidentität**).

**Domäne** Eine Domäne definiert die konkrete Konfiguration, welche zum Matchen von **IDAT** verwendet wird. Personen innerhalb einer Domäne sind immer eindeutig. Das **Record Linkage** findet immer nur innerhalb einer Domäne statt. Werden mehrere Mandaten innerhalb einer **E-PIX**-Instanz verwaltet, so muss für jeden Mandaten eine Domäne mit den spezifischen Matching-Parametern angelegt werden.

**Dublettenauflösung** Bei der Dublettenauflösung werden erkannte **Möglichen Matches** geprüft und oft unter Zuhilfenahme weiterer Informationen aus anderen Datenquellen aufgelöst. Dies umfasst häufig auch die Korrektur offensichtlicher Fehler, sodass eine eindeutige Zusammenführung oder Trennung zweier **Identitäten** möglich wird. Dieser Prozess erfolgt meist manuell.

**Duplikat** Ein Duplikat liegt vor, wenn zwei Patientendatensätze denselben Patienten beschreiben und gegebenenfalls dennoch im selben Bestand gehalten werden. Mithilfe einer Duplikaterkennung (siehe **Record Linkage**) können Duplikate vor einer Eintragung ermittelt werden.

**Field-Level Bloomfilter** Ein Field-Level Bloomfilter ist ein **Bloomfilter**, in dem nur ein Attribut codiert wurde.

**Hauptidentität** Eine Hauptidentität ist die **Identität** die als korrekt angesehene Ausprägung einer Person angesehen wird. Jede Person kann beliebig viele **Nebenidentitäten** haben. In der Weboberfläche wird diese Ausprägung z.B. angezeigt, wenn anhand von **IDAT** nach einer Person gesucht wird. Die Hauptidentität wird vereinzelt auch als Referenz oder Referenzidentität bezeichnet.

**Homonymfehler** Ein Homonymfehler entsteht, wenn die Datensätze mehrere Personen fälschlicherweise nur einer Person zugeordnet werden. Im **E-PIX** wäre dies z.B. der Fall, wenn eine Person zwei **Identitäten** zugeordnet hat, die eigentlich zu verschiedenen Personen gehören.

**Identifier** Ein Identifier ist ein Identifikator, der eine **Identität** eindeutig identifiziert. Dieser Identifier kann vom **E-PIX** in Form eines **MPIs** selbst erzeugt werden

sein, oder aus einem externen System stammen (z.B. Fallnummer oder Patienten-ID aus einem **KIS**). Im **E-PIX** können diese Identifier einer Person zugeordnet werden. Hierzu wird eine **Identifier-Domäne** angelegt, welche die Identifier z.B. eines externen Systems entspricht und diese Identifier beinhaltet.

**Identifier-Domäne** In einer Identifier-Domäne werden alle **Identifier** verwaltet, die zu einem Kontext gehören. Dies umfasst z.B. **MPIs**, die der **E-PIX** erzeugt hat oder **Identifier** aus anderen Systemen. Jede **Identität** kann mehrere **Identifier** aus einer oder verschiedenen Identifier-Domänen zugeordnet bekommen.

**Identifizierende Daten** Die identifizierenden Daten (IDAT) einer Person umfassen alle Daten, welche diese identifizieren können. Hierzu zählen z.B. der Vorname und Nachname, das Geburtsdatum, der Wohnort, der Geburtsort, der Geburtsname und gegebenenfalls weitere Attribute. Einzelne Attribute müssen dabei nicht per se identifizierend sein. Mit Zuhilfenahme weiterer Attribute kann die Kombination dieser jedoch identifizierend werden.

**Identität** Eine Identität ist eine Ausprägung von **IDAT**. Jeder Person können mehrere Identitäten in Form von einer **Hauptidentität** beliebig vielen **Nebenidentitäten** zugeordnet werden.

**First MPI** Der First MPI wird bei der erstmaligen Registrierung einer Person vergeben. Wird die **Identität** durch eine **Dublettenauflösung** an eine andere Person angefügt, so wird die ursprüngliche Person deaktiviert, behält aber den First MPI. Die Person ist damit immer über den First MPI findbar.

**Lokaler Identifier** Ein lokaler Identifier ist ein Identifikator, der durch ein externes System vergeben wurde, wie beispielsweise einem **KIS**. Der Lokale Identifier identifiziert dabei die Personenidentität eindeutig in diesem System. Aus einem System können dabei mehrere Lokale Identifier stammen (z.B. Patienten-ID und Fallnummer). Der **Patientenidentifikator** kann in seiner Funktion als Identifier auch als LID ("Lokaler (externer) Identifier") betrachtet werden. Im **E-PIX** können diese Identifier einer Person zugeordnet werden. Hierzu wird eine **Identifier-Domäne** angelegt, welche die Identifier z.B. eines externen Systems entspricht und diese Identifier beinhaltet.

**Match** Ähnlichkeit zweier **Identitäten** überschreitet einen bestimmten Schwellwert. Der **E-PIX** unterscheidet zwischen **Perfekter Match**, **Automatischer Match** und **Möglicher Match**. Sofern die Ähnlichkeit unter dem Schwellwert für einen **Möglicher Match** liegt, dies als **Kein Match** bezeichnet.

**Automatischer Match** (engl.: Automatic Match) Die Ähnlichkeit zweier **Identitäten** überschreitet den Schwellwert für *automatische Matches*. Die zu registrierende **Identität** wird automatisch der vorhandenen Person zugeordnet. Es ist keine **Dublettenauflösung** erforderlich.

**Guter Match** (engl.: Good Match) → **Automatischer Match**

**Matching Attribut/Feld** Ein Attribut, welches zum **Record Linkage** verwendet wird. Wird z.B. das Vorname-Attribut als Matching Attribut verwendet, dann werden die Vornamen der Datensätze miteinander verglichen.

**Kein Match** (engl.: Non-Match) Die Ähnlichkeit zweier **Identitäten** unterschreitet den Schwellwert für einen **Möglicher Match**. Die **Identitäten** werden zwei verschiedenen Personen zugeordnet.

**Möglicher Match** (engl.: Possible Match) Zwei Identitäten weisen eine hohe Ähnlichkeit auf, sind jedoch nicht exakt gleich. Aufgrund der hohen Ähnlichkeit kann ein Verbinden der beiden **Identitäten** in Betracht gezogen werden

**Perfekter Match** (engl.: Perfect Match) Bei einem Perfect Match ergibt der Vergleich zweier **Identitäten** die völlige Übereinstimmung aller verglichenen **IDAT**.

**Nebenidentität** Eine Nebenidentität ist eine Ausprägung von **IDAT**. Jeder Person können mehrere Nebenidentitäten zugeordnet werden. Die **Hauptidentität** zeigt an, welche **Identität** die als korrekt angesehene Ausprägung angesehen wird.

**Objekt-Identifikator** Ein Objekt-Identifikator (OID) ist ein eindeutiger Bezeichner für ein Objekt. Im **E-PIX** erhält jede **Identifier-Domäne** einen OID. Bei der Vergabe von **MPIs** kann z.B. eine entsprechende **Identifier-Domäne** mit dem OID der Forschungseinrichtung hinterlegt werden.

**Patientenidentifikator** Ein Patientenidentifikator (PID) ist ein Pseudonym (siehe **Pseudonym**) erster Stufe. Demnach wird diese Kennung einem Patienten direkt zugeordnet.

**Privacy-Preserving Record Linkage** Das Privacy-Preserving Record Linkage ist ein Verfahren um Datensätze abzugleichen, ohne dabei die Identität einer Person offenbaren zu müssen. Eine Möglichkeit dies umzusetzen, ist der Einsatz von **Bloomfiltern**, welche zwar einen Abgleich von Personendaten ermöglicht, ohne jedoch Rückschlüsse auf diese Daten zu ermöglichen.

**Pseudonym** Ein Pseudonym ist eine nichtssagende Kennung. Mit diesem kann die **Identität** eines Patienten verschleiert werden, da mit alleiniger Nutzung der Kennung keine Rückschlüsse auf die **Identität** gezogen werden können. Pseudonyme können über eine Zufallskennung erzeugt werden. Ein Pseudonym erster Stufe wird durch einen **Patientenidentifikator** realisiert. Bei mehrstufigen Pseudonymen wird einem Pseudonym ein weiteres Pseudonym zugeordnet. So lassen sich beliebig viele Stufen abbilden.

**Quelle** → **Datenquelle**

**Record Linkage** Verfahren um zwei **Identitäten** auf Gleichheit zu prüfen. Dabei werden alle oder nur eine Teilmenge von Personenattributen bzw. **IDAT**

miteinander verglichen. Je nachdem welche Ähnlichkeit diese in Summe aufweisen, werden die **Identitäten** als **Duplikat** erkannt und gehören demnach zur selben Person.

**Referenz** → **Hauptidentität**

**Referenzidentität** → **Hauptidentität**

**Sichere Datenquelle** Eine Sichere Datenquelle ist eine **Datenquelle** bei der die registrierten **IDAT** als korrekte Ausprägung angesehen werden. Siehe auch **Datenquelle**.

**Synonymfehler** Ein Synonymfehler entsteht, wenn mehrere Datensätze, die zu einer Person zugehörig sind, nicht dieser Person zugeordnet werden, sondern auf im Datenbestand als verschiedene Personen betrachtet werden. Im **E-PIX** wäre dies z.B. der Fall, wenn zwei **Identitäten** auf zwei Personen verteilt verwaltet werden, statt diese derselben Person zuzuordnen.

**THS-Dispatcher** Der Treuhandstellen-Dispatcher ist ein Workflowmanagementsystem. Damit lassen sich Prozesse im Treuhandstellenkontext abbilden. Hierbei stehen diverse Schnittstellen zur Verfügung, um z.B. Abläufe zwischen Systemen des Identitäts-, Pseudonym- und Einwilligungsmanagements abzubilden.

# Abkürzungsverzeichnis

- CLK** Cryptographic Long Term Key *Glossar: Cryptographic Long Term Key*  
**DBMS** Datenbankmanagementsystem *Glossar: Datenbankmanagementsystem*  
**E-PIX** Enterprise Identifier Cross-Referencing  
**eGK** Elektronische Gesundheitskarte  
**gPAS** generic pseudonym administration service  
**IDAT** Identifizierende Daten. *Glossar: Identifizierende Daten*  
**KIS** Krankenhausinformationssystem  
**KVNR** Krankenversichertennummer  
**MDAT** Medizinische Daten  
**MPI** Master Patient Index  
**OID** Objekt-Identifikator  
**PPRL** Privacy-Preserving Record Linkage  
    *Glossar: Privacy-Preserving Record Linkage*  
**TMF** Technologie- und Methodenplattform für die vernetzte medizinische Forschung e.V.  
**WSDL** Web Services Description Language  
**XML** Extensible Markup Language