Anwenderhandbuch

E-PIX®

Enterprise Identifier Cross-Referencing

Version 2.12 vom 07.07.2021

Herausgeber:

Unabhängige Treuhandstelle der Universitätsmedizin Greifswald

Autor:

Christopher Hampf, M.Sc.

Ellernholzstr. 1-2
17475 Greifswald

Tel. 03834 / 86-7851, Fax: 03834 / 86-6843
E-Mail: christopher.hampf@uni-greifswald.de

Unabhängige Treuhandstelle
UNIVERSITÄTSMEDIZIN GREIFSWALD

Versionierung

Version	Datum	Bearbeitungsart / Betroffene Abschnitte	Bearbeiter
0.9	19.06.2019	Update und Erweiterung der Dokumentation des Mosaic-Projektes	Christopher Hampf
2.9.2	02.04.2020	Aktualisierung auf E-PIX Version 2.9.2 und Erweiterung um Domänen-Konfiguration	Christopher Hampf
2.10	11.02.2021	Aktualisierung auf E-PIX Version 2.10	Christopher Hampf
2.12	07.07.2021	Aktualisierung auf E-PIX Version 2.12 Ergänzung der Domänen-Konfiguration um Bloomfilter Aktualisierung aller Bilder	Christopher Hampf

Inhalt

Abbildungsverzeichnis

Tabellenverzeichnis

1 Hintergrund

Um beispielsweise medizinische Daten einer Person eindeutig zuordnen zu können, verwenden Einrichtungen wie Kliniken oder Register typischerweise lokal eindeutige Kennungen (sog. Local Identifier). Diese Kennungen haben jedoch nur innerhalb der jeweiligen Domäne (z.B. Klinik) Gültigkeit. Zudem können identifizierende Daten 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 eineindeutige systemweite Kennung erforderlich, der sowohl die identifizierenden Daten (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.

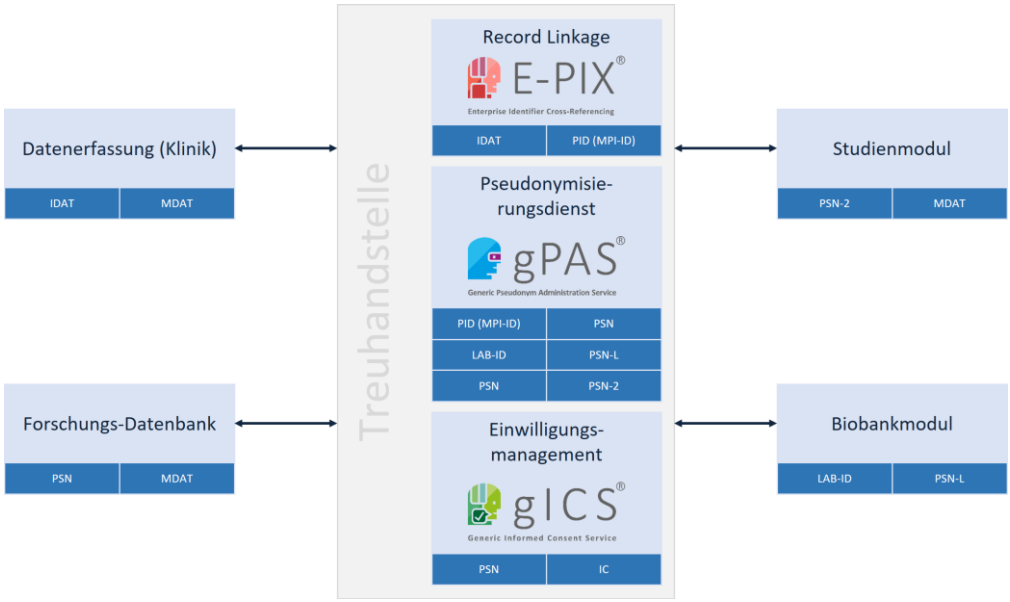

Abbildung 1-1: Das Identitätsdatenmanagement stellt eine zentrale Komponente im medizinischen Forschungskontext dar. Verschiedene Module verwalten modulspezifische Daten und ordnen diese Personen mittels spezifischen Pseudonymen zu. Die Abbildung ist adaptiert vom Maximalmodell des Generischen Datenschutzkonzepts der TMF[1].

[1] POMMERENING, Klaus; HELBING, Krister; GANSLANDT, Thomas; DREPPER, Johannes: Leitfaden zum Datenschutz in medizinischen Forschungsprojekten. Berlin: Medizinisch Wissenschaftliche Verlagsgesellschaft mbH & Co. KG, 2014. – ISBN 978–3–95466–123–7

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 eineindeutige Kennung**. Diese stellt gemäß den Konzepten der TMF ein Pseudonym erster Stufe dar. (Quelle: TMF 2004, 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 lizensiert (AGPLv3) und kostenfrei für kommerzielle und nicht-kommerzielle Zwecke einsetzbar.

2 Der Enterprise Identifier Cross-Referencing (E-PIX)

Der **E-PIX-Service** (kurz für: Enterprise Identifier Cross-Referencing) 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 unterscheidet sich vom Ansatz des PID-Generators der TMF nicht nur in Bezug auf die verwendeten Algorithmen, sondern auch in Bezug auf die Speicherung domänenspezifischer Lokaler Identifier und die Unterstützung standardisierter IHE-Profile (PIX, PDQ). Zudem ermöglicht das **Konzept multipler Personenidentitäten**, d.h. einer real existierenden Person können mehrere Ausprägungen (ähnlicher) demografischer Daten zugeordnet sein, die technische Unterstützung beim Auflösen von Synonymfehlern (s. **Abschnitt 4**).

3 Begriffsbestimmungen

Person

Eine natürliche Person, beschrieben durch eine oder mehrere Personenidentitäten.

Personendaten / Identifizierende Daten (IDAT)

Alle Attribute wie Vorname, Nachname, Kontaktdaten, etc. die einer Person zugeordnet sind. Attribute, die eine Person eindeutig identifizieren, werden als identifizierende Daten bezeichnet.

Personenidentität

Bezeichnet eine konkrete Ausprägung eines IDAT-Satzes einer Person. Eine Person kann mehre Identitäten (Haupt- und Nebenidentitäten) besitzen, die sich zum Beispiel in ihrer Schreibweise oder Aktualität unterscheiden (s. **Abschnitt 4**).

Identifier einer Personenidentität

Eineindeutiger Identifikator (z.B. eine ID) um eine Personenidentität eindeutig zu identifizieren.

Lokaler Identifier

Ein Lokaler Identifier ist ein Identifikator, der durch ein externes System vergeben wurde, wie beispielsweise einem KIS-System. 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 Personenidentifikator (PID) kann in seiner Funktion als Identifier auch als LID ("*Lokaler (externer) Identifier*") betrachtet werden.

Domain (Domäne)

Eine Domain ist eine organisatorische Einheit (Mandant), z.B. eine Studie, ein Projekt oder ein Institut.

Lokale Domain (oder auch Identifier Domain)

Domäne des Lokalen Identifiers. Diese muss nicht dem Quellsystem entsprechen. Aus einem Quellsystem können mehrere Lokale Identifier stammen, bspw. Patienten-ID und Fallnummer aus einem KIS. Gleichzeitig kann die gleiche Lokale ID aus unterschiedlichen Quellen stammen, bspw. eine Fallnummer aus einem elektronischen KIS-Export sowie die gleiche Fallnummer von einem Arztbrief.

Matching-Parameter

Frei wählbares Personenattribut (z.B. Vorname, Nachname, Geburtsdatum, etc.), das für das Matching-Verfahren verwendet werden.

Record Linkage

Verfahren um Datensätze einer Person einander zuzuordnen. Hierzu wird die Ähnlichkeit definierter Personendaten (vgl. Matching-Parameter) ermittelt und bei hinreichender Übereinstimmung ein und derselben Person (als Personenidentität) zugeordnet.

Quelle

Datenquelle, aus der IDAT stammen können, z.B. ein Krankenhaus oder ein Forschungsprojekt. Bei der Registrierung einer Personenidentität wird die Quelle ausgewählt, aus der die jeweiligen IDAT stammen. Es kann eine *Sichere Quelle* einer Domäne zugeordnet werden. IDAT die über die Sichere Quelle registriert werden, werden als korrekte Ausprägung einer Personenidentität angesehen (Hauptidentität, vgl. Personenidentität).

4 Das Konzept der Nebenidentitäten

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

Innerhalb des E-PIX kann eine Person mehrere Identitäten besitzen, wovon nur eine als Hauptidentität 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 (Datentreuhänder) und nach eingehender Recherche der Sachlage erfolgen.

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

5 Funktionalitäten

5.1 Was leistet der Dienst

- Erstellung und Verwaltung einer systemweit eindeutigen Kennung mittels Indexgenerator nach dem Konzept des Master Person Index
- Zusammenführung von Personendaten aus unterschiedlichen Quellsystemen anhand demographischer Informationen
- Umgang mit fehlerhaften/unvollständigen Personendaten
- Unterstützung bei der Rekontaktierung durch die integrierte Personenverwaltung
- Unterstützung beim Auflösen von möglichen Matches durch das Konzept von Haupt- und Nebenidentitäten
- 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 10.00.000 Patienten in deutlich weniger als 1 Sekunde (Entsprechende Tests zur Performance sind in aufgezeigt)

5.2 Was leistet der Dienst nicht

- Eine automatisierte Transkription und Transliteration von demografischen Informationen ist nicht möglich.
- Die Vergabe von Pseudonymen zweiter Stufe ist nur durch Kombination des E-PIX- und des gPAS-Services möglich.

6 Installation per Docker

6.1 Systemanforderungen

Technisch

- Windows oder Ubuntu Server (oder vergleichbar)[2]
- Mindestens 8 GB Arbeitsspeicher
- Mindestens 5 GB Festplattenspeicher
- Installierte aktuelle Version von *Docker*[3] und Docker Compose[4]
- Administrative Rechte
- Keine Nutzungsbeschränkungen auf die bereitgestellten Service- und Client-URLs

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 möglicher Matches NACH ausführlicher Prüfung der individuellen Sachlage

6.2 Download und Starten des Dienstes

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 6-1). Damit diese nicht einzeln gestartet und entsprechend zusammengeschaltet werden müssen, wird der Dienst mit Docker-Compose gestartet. Alle hierzu benötigten Ressourcen werden im öffentlichen Repository des MOSAIC-Projekts bereitgestellt[5].

[2] Beim Betrieb unter Windows ist zu beachten, dass bei der Verwendung von Volumes und parallel betriebenen VPN-Clients Probleme auftreten können.
[3] Weitere Informationen unter https://docs.docker.com/install/
[4] Weitere Informationen unter https://docs.docker.com/compose/install/
[5] Weitere Informationen unter https://github.com/mosaic-hgw/E-PIX

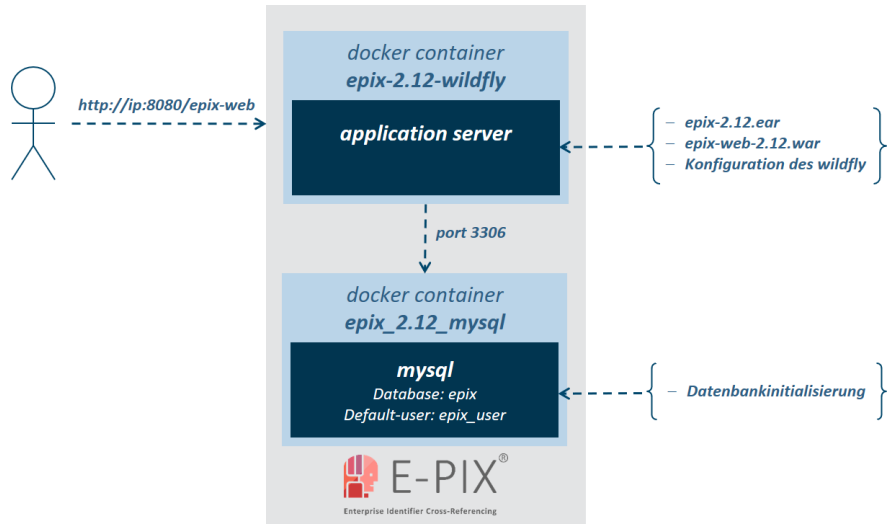

Abbildung 6-1: Aktuelle Architektur des E-PIX mit Docker.

Das entstehende Docker-System besteht aus getrennten Containern für die erforderliche Datenbankinstanz (MySQL) und den benötigten Anwendungsserver (Wildfly inkl. Datenbank-Konnektoren) zur Verfügbarmachung der Programmdateien. Beide Container kommunizieren intern über den MySQL-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.

Die Konfiguration erfolgt mittels Docker Compose. Der Installationsvorgang nimmt rund 3 Minuten in Anspruch und erfordern rund 1.4 GByte an Speicherplatz.

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

```
sudo git clone https://https://github.com/mosaic-hgw/E-PIX
```

Vergabe Schreibrechten

```
sudo chmod -R 777 E-PIX/docker
```

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

```
cd E-PIX/docker/standard
```

Sollte der MySQL-Dienst auf der Maschine ausgeführt werden, dann stoppen des Dienstes

```
sudo service mysql stop
```

Prüfen der Docker-Version (vorausgesetzt ist Version 1.13.1 oder höher)

```
sudo docker -v
```

Prüfen der Docker Compose-Version (vorausgesetzt ist Version 1.8.0 oder höher)

```
sudo docker-compose -v
```

Starten des E-PIX mithilfe von Docker Compose

```
sudo docker-compose up
```

Damit werden die benötigten Komponenten heruntergeladen und die Konfiguration von MySQL und Wildfly gestartet. Danach wird die aktuelle Version des E-PIX bereitgestellt. Der Installationsvorgang kann in Abhängigkeit der vorhandenen Internetverbindung etwa 7 Minuten dauern. Der erfolgreiche Start des Dienstes wird mit der folgenden Ausgabe abgeschlossen.

```
Wildfly 18.0.0.Final (Wildfly Core 5.0.0.Final) started in 63373ms –
Started 824 of 1024 services (324 services are lazy, passive or on-demand)
```

7 Die grafische Benutzeroberfläche des E-PIX

Um dem Datentreuhänder 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 eines Datentreuhänders. Die daraus resultierenden Anwendungsfälle umfassen das

1. Anlegen von Domänen, Quellen und Identifier-Domänen
2. Anlegen neuer Personen
3. Suchen von Personen anhand demografischer Informationen
4. Bearbeiten von Personen
5. Auflösen möglicher Synonymfehler
6. Exportieren von Daten
7. Importieren von Daten
8. Einsehen von Protokollen
9. Einsehen von Statistiken

7.1 Anlegen von Domänen, Quellen und Identifier-Domänen

Der E-PIX erlaubt die Verarbeitung von Patienten mehrerer Mandanten innerhalb einer Datenbank durch die Verwendung von Domänen (vgl. **Abschnitt 3** Begriffsbestimmungen). 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 Quelle* (vgl. Quelle) und eine Identifier-Domain angegeben werden. Diese sollten vor dem Anlegen der Domäne im System angelegt werden. Die nötigen Schritte sind unter dem Menüpunkt *Domänen* vorzunehmen und werden im

Folgenden beschrieben. **Abbildung 7-1** zeigt die Ansicht der grafischen Oberfläche.

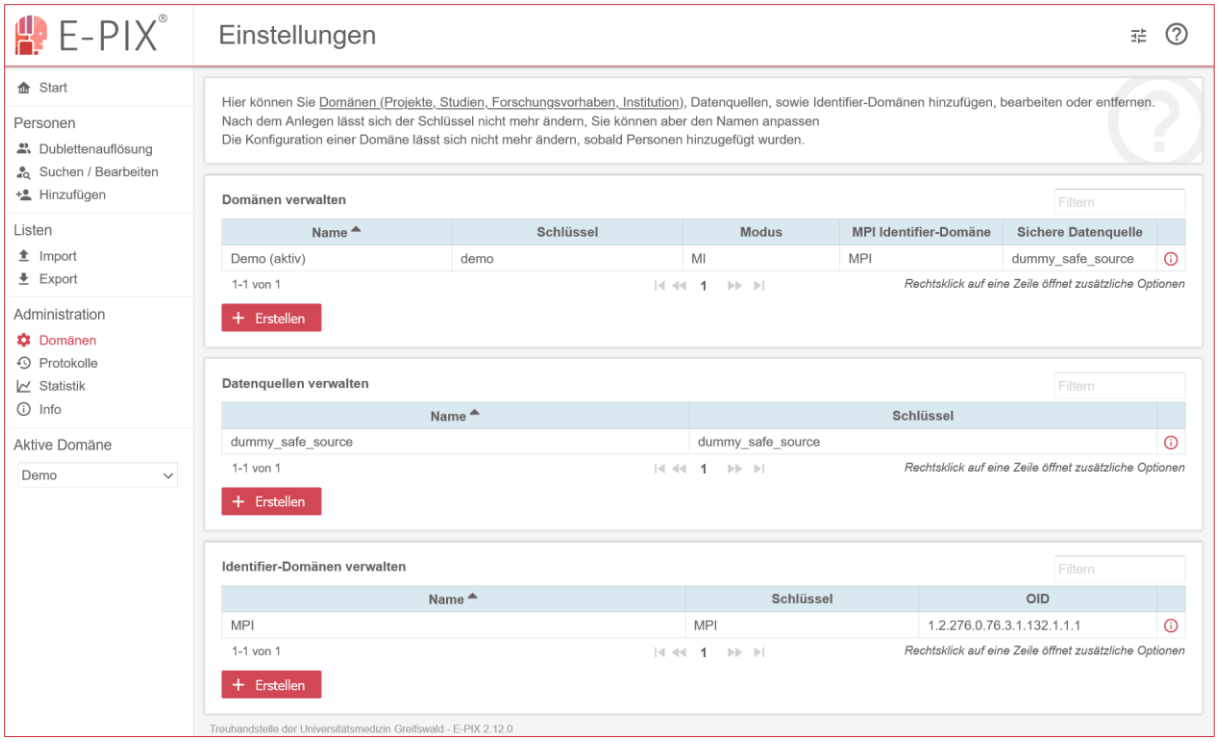

Abbildung 7-1: Oberfläche zum Anlegen von Domänen, Quellen und Identifier-Domänen.

7.1.1 Anlegen einer neuen Quelle

Eine Quelle gibt an, woher die später registrierten Personendaten stammen, also bspw. aus einer bestimmten Studie oder einem Krankenhausinformationssystem (vgl. **Abschnitt 3** Begriffsbestimmungen). Die Quelle kann bei einer Personenregistrierung aus der Liste der zuvor angelegten Einträge ausgewählt werden. Mithilfe der Schaltfläche *Neue Quelle* wird ein neuer Eintrag angelegt. Nachdem ein eindeutiger Name und idealer Weise eine Beschreibung angegeben wurden, kann mit der „Haken"-Schaltfläche die Quelle bestätigt werden. Die *Sichere Quelle* einer Domäne definiert, woher die Hauptidentitäten (respektive die Personendaten, welche als korrekte Ausprägung angesehen werden) stammen.

7.1.2 Anlegen einer Identifier-Domäne

Die Domäne eines Lokalen Identifiers, die so genannte Identifier-Domäne wird auf ähnliche Weise angelegt, wie die Quelle. Hierbei müssen der Name und die OID eindeutig sein. Jede Forschungseinrichtung besitzt typischerweise eine OID, welche hier angegeben werden kann.

7.1.3 Anlegen einer neuen Domäne

Nachdem die *Sichere Quelle* sowie die *Identifier-Domäne* angelegt wurden, kann ein neuer Domänen-Eintrag über die Schaltfläche *Neue Domäne* erstellt werden. Hierfür muss ein eindeutiger Name eingetragen sowie die *MPI Domäne* und die *Sichere Quelle* für diese konkrete Domäne über die Listen ausgewählt werden. Eine Beschreibung sollte insbesondere bei der Verarbeitung von Personen für mehrere Mandanten oder Projekte innerhalb eines E-PIX-Systems eingetragen werden. Die

Konfiguration ist im XML-Format vorzunehmen (Die Domänenkonfiguration wird umfangreich im **Kapitel 9** erklärt).

7.2 Anlegen einer neuen 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 Personendaten eingetragen werden können. Pflichtfelder sind mit einem Stern (*) gekennzeichnet. Welche Felder Pflichtfelder sind, wird in der Konfiguration der Domäne festgelegt (siehe Domänenkonfiguration Pflichtfelder in **Abschnitt 9.4.7**). Es können zu jeder Person außerdem noch beliebig viele Adress- bzw. Kontaktdaten und Lokale Identifier hinterlegt werden. Mithilfe der Domänen-Konfiguration können noch weitere Felder definiert und benannt werden (vgl. **Abschnitt 9.4.8**). Die *Quelle* aus der die Daten stammen muss ebenfalls angegeben werden. Entspricht die angegebene *Quelle* der *Sicheren Quelle* der jeweiligen Domäne, dann wird bei Feststellung eines Duplikates die Identität als Hauptidentität deklariert. Diese gilt dann als fehlerfrei. Andernfalls wird eine neue Nebenidentität angelegt. In Abbildung 7-2 wird exemplarisch das Eintragen der Pflichtfelder dargestellt.

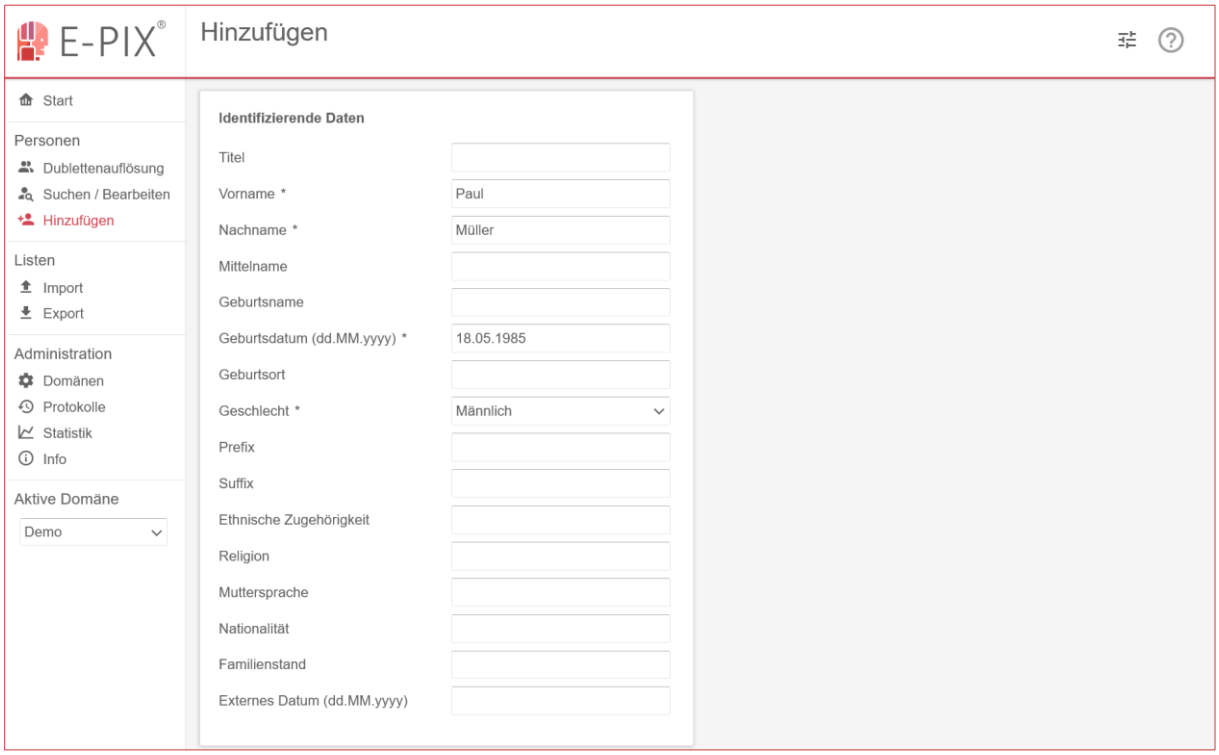

Abbildung 7-2: Oberfläche zum Eintragen von Personendaten.

ⓘ Was passiert, wenn ein lokaler 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 geliefert. Der Grund hierfür ist, dass jeder Identifier nur einer Person zugeordnet sein darf (mehrere Identitäten (Ausprägungen einer Person) können denselben Identifier aufweisen, diese müssen dann aber alle derselben Person zugeordnet sein).

ⓘ Was passiert, wenn zwei Identitäten identisch (*perfect Match*) sind, aber die lokalen Identifier verschieden sind?

Die lokalen 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 lokale Identifier niemals unterschiedlichen Personen zugeordnet ist.

Record Linkage und Match-Typen

Nach dem Anwählen der Schaltfläche *Person hinzufügen* findet ein Abgleich der IDAT und bei hinreichender Ähnlichkeit eine Zusammenführung von Personen mittels Record Linkage statt. Mit einer Mitteilung wird über Erfolg oder Misserfolg informiert. Abhängig von der jeweiligen Konfiguration unterscheidet man nach einem Record Linkage unterschiedliche Matchtypen. Diese sind in **Tabelle 7-1** dargestellt.

Tabelle 7-1: Mögliche Match-Typen

Match-Typ	Beschreibung
Perfect Match / Perfekter Match	Exakte Übereinstimmung zweier Datensätze in Bezug auf die Matching-Parameter. Es wird keine neue Person und keine neue Identität angelegt, da die Personendaten bereits in gleicher Form hinterlegt sind.
Automatic Match bzw. Match	Im Hinblick auf den konfigurierten Schwellwert haben zwei Datensätze eine hinreichende Ähnlichkeit. Die neu angegebenen Personendaten werden der bereits bestehenden Person als neue Identität zugeordnet.
Possible Match / Möglicher Match	Es besteht eine Ähnlichkeit zwischen zwei Datensätzen. Bei einem *möglichen Match* findet jedoch keine automatische Zusammenführung statt. Eine Dublettenauflösung kann nur manuell im Nachgang unter Zuhilfenahme weiterer Informationen erfolgen (siehe Anwendungsfall 5 in **Abschnitt 7.5**).

Non-Match / Kein Match	Keine Ähnlichkeit zu einem bestehenden Datensatz. Wenn kein Duplikat festgestellt wurde respektive die Person noch nicht bekannt ist, dann wird eine neue Person angelegt.

7.3 Suchen einer Person anhand demografischer Informationen

Unter dem Menüpunkt *Suchen / Bearbeiten* kann nach Personen gesucht werden, welche mit den angegebenen demographischen Daten übereinstimmen. Dabei kann nach Attributen wie Vornamen, Nachnamen, Geburtsdatum und Geschlecht oder nach (Lokalen) Identifiern gesucht werden. Es müssen hierbei nicht alle Attribute ausgefüllt werden. Die Attribute sind dabei standardmäßig UND-Verknüpft, sodass die Ergebnisliste nur Personen enthält, die alle angegebenen Attribute aufweisen. Alternativ kann auch eine ODER-Verknüpfung erfolgen, sodass die Ergebnisliste nur Personen aufweist, die zumindest mit einem der angegebenen Attribute übereinstimmt. Zum Umschalten ist ein Schalter mit der Bezeichnung *Verknüpfung der Suchparameter* vorhanden. In **Abbildung 7-3** wird exemplarisch eine Person anhand der Attribute Vorname, Nachname und Geschlecht gesucht. Die Ergebnisliste enthält genau einen Eintrag.

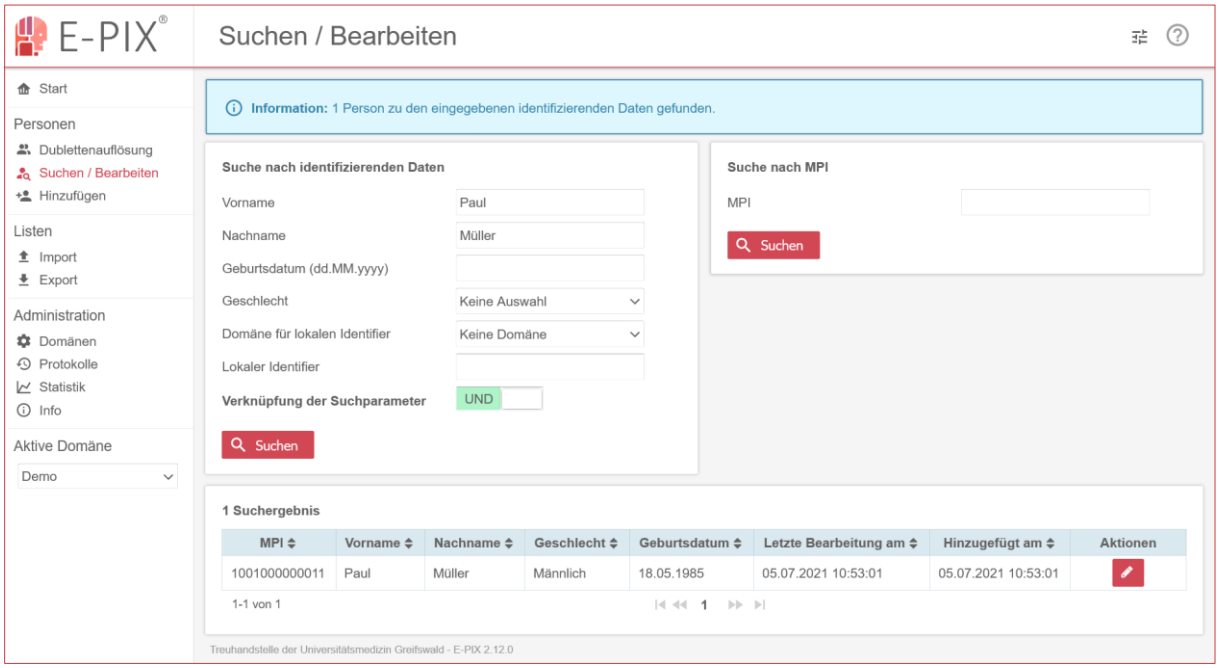

Abbildung 7-3: Oberfläche zum Suchen von Personen anhand von demographischen Daten.

7.4 Bearbeiten einer Person

Um beispielsweise fehlerhafte Eingaben zu korrigieren oder fehlende Attribute zu ergänzen, kann es erforderlich sein, die Attribute einer Person zu bearbeiten. Hierzu wird zunächst die betreffende Person im E-PIX gesucht (siehe **Abschnitt 7.3**). In der Ergebnisliste kann über die *Bearbeiten*-Schaltfläche die jeweilige Person aufgerufen werden. Die hinterlegten Daten zur Person sind bereits im Formular vorausgefüllt. Zur Gewährleistung der Integrität der Daten sollte ein Grund für die

Änderung der Daten angegeben werden. Mit der Anwahl der Schaltfläche *Person aktualisieren*, werden die ergänzten oder veränderten Daten gespeichert. Eine entsprechende Meldung weißt auf Erfolg oder Misserfolg der Aktualisierung hin. Bei Misserfolg müssen ggf. die Angaben korrigiert werden. In **Abbildung 7-4** ist die Oberfläche zum Bearbeiten einer Person abgebildet.

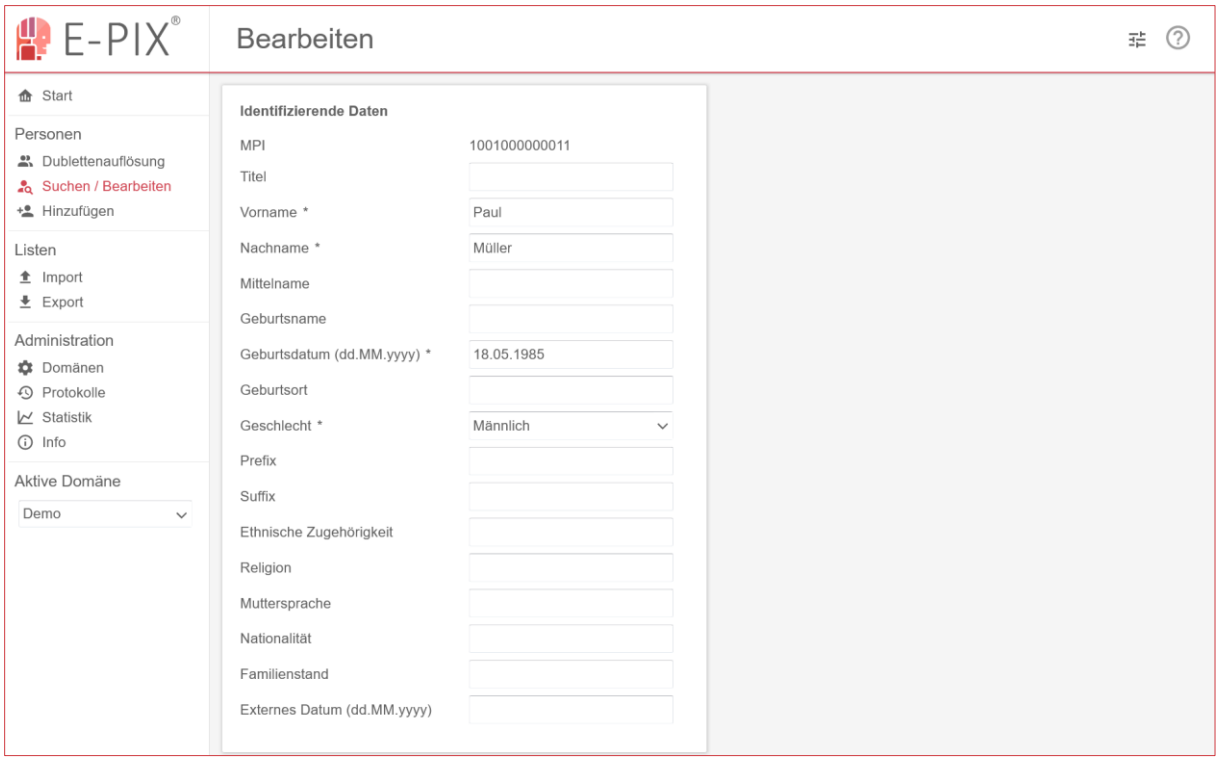

Abbildung 7-4: Oberfläche zum Bearbeiten einer Person.

7.5 Auflösen möglicher Synonymfehler

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 möglicher Dubletten angewählt. Beide Personendatensätze werden tabellarisch gegenübergestellt und Unterschiede bei den jeweiligen Attributen farbig hervorgehoben. 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 natürlichen Person, wird mit der Schaltfläche *Behalte beide Personen* der jeweilige Datensatz als korrekte Ausprägung ausgewählt werden. Der andere Datensatz wird der Person als Nebenidentitäten zugeordnet (dabei bleiben alle etwaigen Nebenidentitäten der beiden Personen erhalten). Wenn beide Datensätze unterschiedlichen Personen zugehörig sind, respektive keine Dublette darstellen, wird über die Schaltfläche *Personen trennen* ein Ausschluss als potentielle Dublette angegeben. Die Personen bleiben dabei getrennt und die Einträge werden aus der Dublettenauflösung entfernt. In **Abbildung 7-5** ist die Oberfläche zur Gegenüberstellung von Personendaten abgebildet.

Wenn zwei Identitäten nicht ähnlich genug sind, um automatisch als *Mögliches Duplikat* erkannt zu werden, kann händisch ein entsprechender Eintrag angelegt werden. Hierzu kann die Schaltfläche *Mögliche Dublette anlegen* angewählt. Dabei können Dubletten zwischen Personen oder Identitäten

angegeben werden. Zwischen zwei Personen werden hierzu die entsprechenden MPIs angegeben. Bei zwei Identitäten werden die jeweiligen IDs angegeben. Danach erfolgt die Auflösung wie zuvor beschrieben.

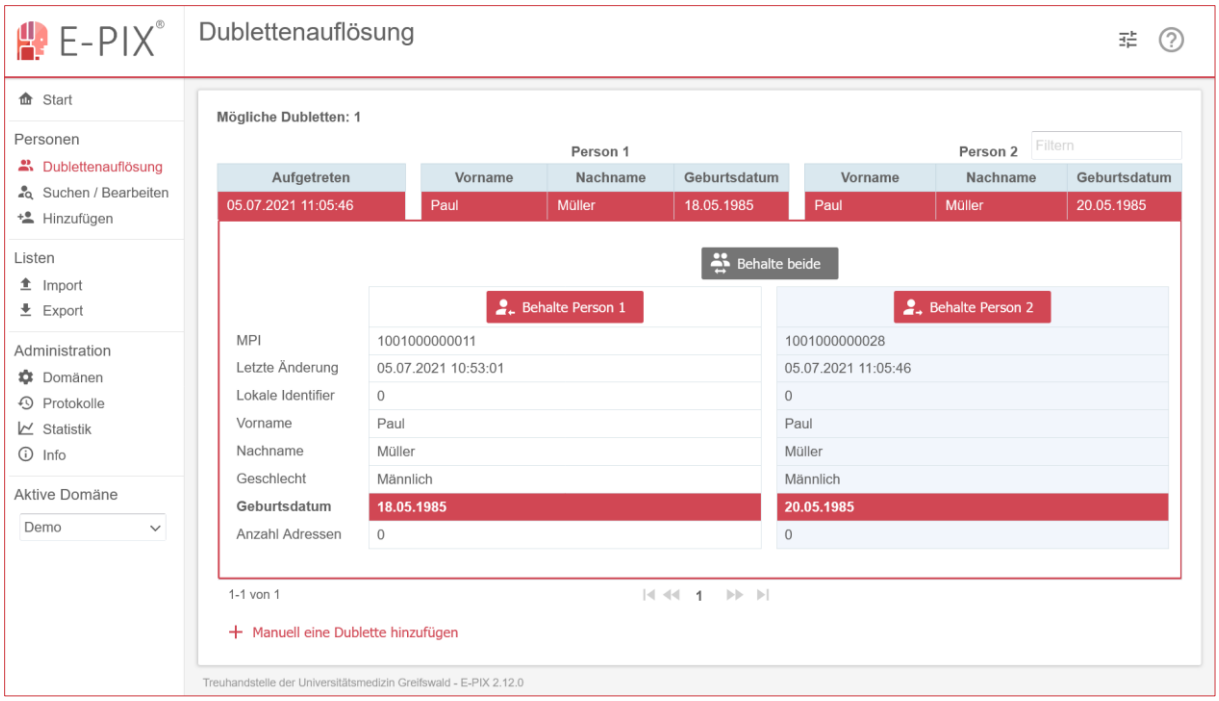

Abbildung 7-5: Gegenüberstellung von Personendaten zum Auflösen einer Dublette.

7.6 Daten exportieren

Die registrierten Personendaten können in eine CSV-Datei exportiert werden. Hierzu wird unter dem Menüpunkt *Export* der Modus gewählt, anhand dessen die Liste der zu exportierenden Personendaten bestimmt wird. Es können alle Personen oder gefiltert nach einer bestimmten Identifier-Domäne selektiert werden. Je nach Modus können verschiedene Optionen gewählt werden. Die zu exportierenden Personendaten werden nach der Anwahl der Schaltfläche *Suche* 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 Attribute des resultierenden Exports durch verschieben der Spalten beeinflusst werden. Die resultierende CSV-Datei wird mit der Anwahl der Schaltfläche *CSV herunterladen* heruntergeladen. Die Daten werden standardmäßig mit einem Semikolon separiert, was mit `sep=;` in der ersten Zeile der CSV-Datei dargestellt wird. In **Abbildung 7-6** wird die entsprechende Oberfläche exemplarisch dargestellt.

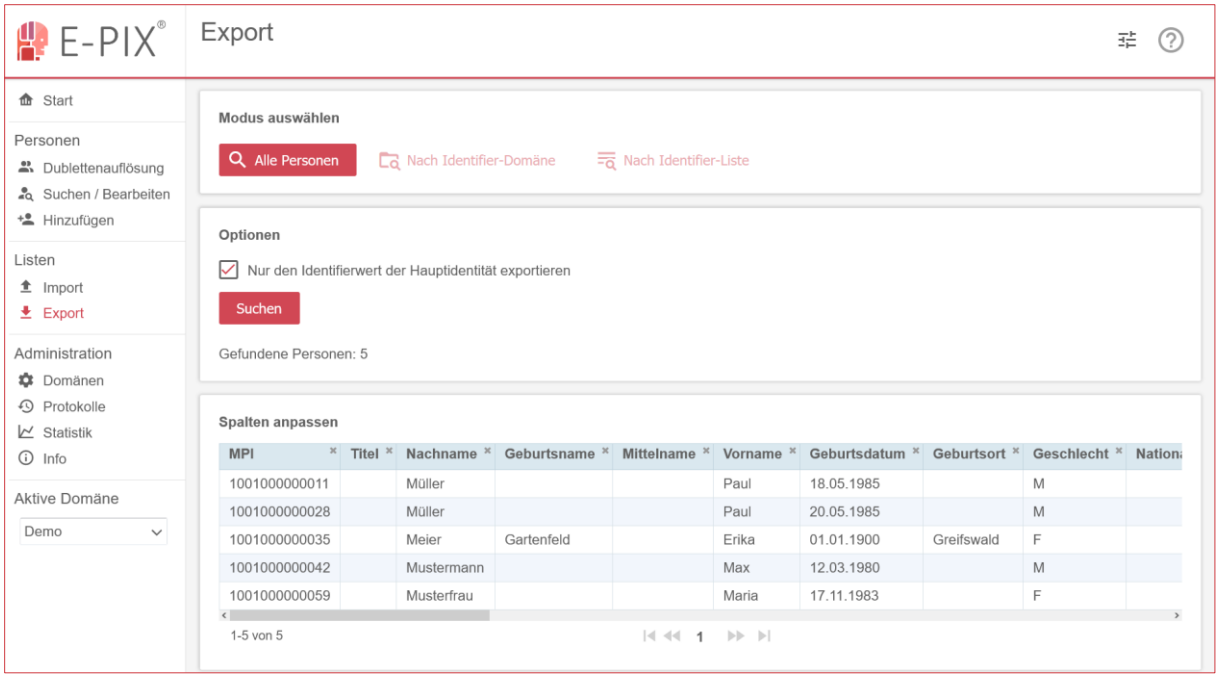

Abbildung 7-6: Oberfläche zum Exportieren von Personendaten.

7.7 Daten importieren

Um Personendaten zu importieren, kann über den Reiter *Import* eine CSV-Datei ausgewählt werden. In **Abbildung 7-7** ist die Oberfläche zum Wählen der CSV-Datei dargestellt.

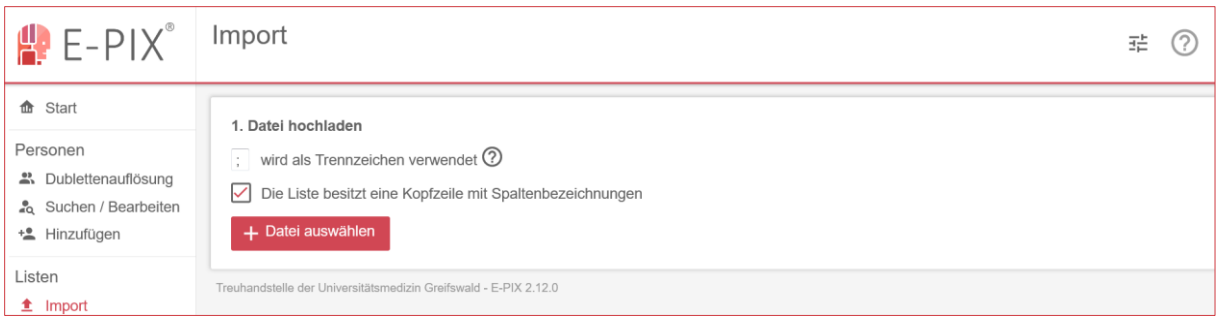

Abbildung 7-7: Oberfläche zum Importieren von Personendaten.

Ist eine Überschrift in der CSV-Datei enthalten, so kann dies mittels der Checkbox *Datei besitzt eine Kopfzeile mit Spaltennamen* mitgeteilt 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 Komma. Soll ein anderes Trennzeichen verwendet werden, bspw. ein Semikolon, so kann dies mittels `sep=;` in der ersten Zeile der CSV-Datei definiert werden[6].

Als Vorschau werden die Daten der ersten enthaltenen Zeilen dargestellt. Wurden in der CSV-Datei Spaltennamen verwendet, die den Attributnamen der E-PIX-Datenbank entsprechen, erfolgt

[6] Dieser Eintrag wird beim Import nicht als Zeile eingelesen und beeinflusst nicht eine etwaig vorhandene Kopfzeile.

automatisch eine Zuordnung. Sollen die Spalten anderen Attributen zugeordnet werden oder wurden keine Spaltennamen vorgegeben, so kann über das Auswahlmenü jeder Spalte ein beliebiges Attribut zugewiesen werden. Welche Spalten importiert werden sollen, kann über die Checkboxen bestimmt werden. In **Abbildung 7-8** ist die entsprechende Oberfläche dargestellt.

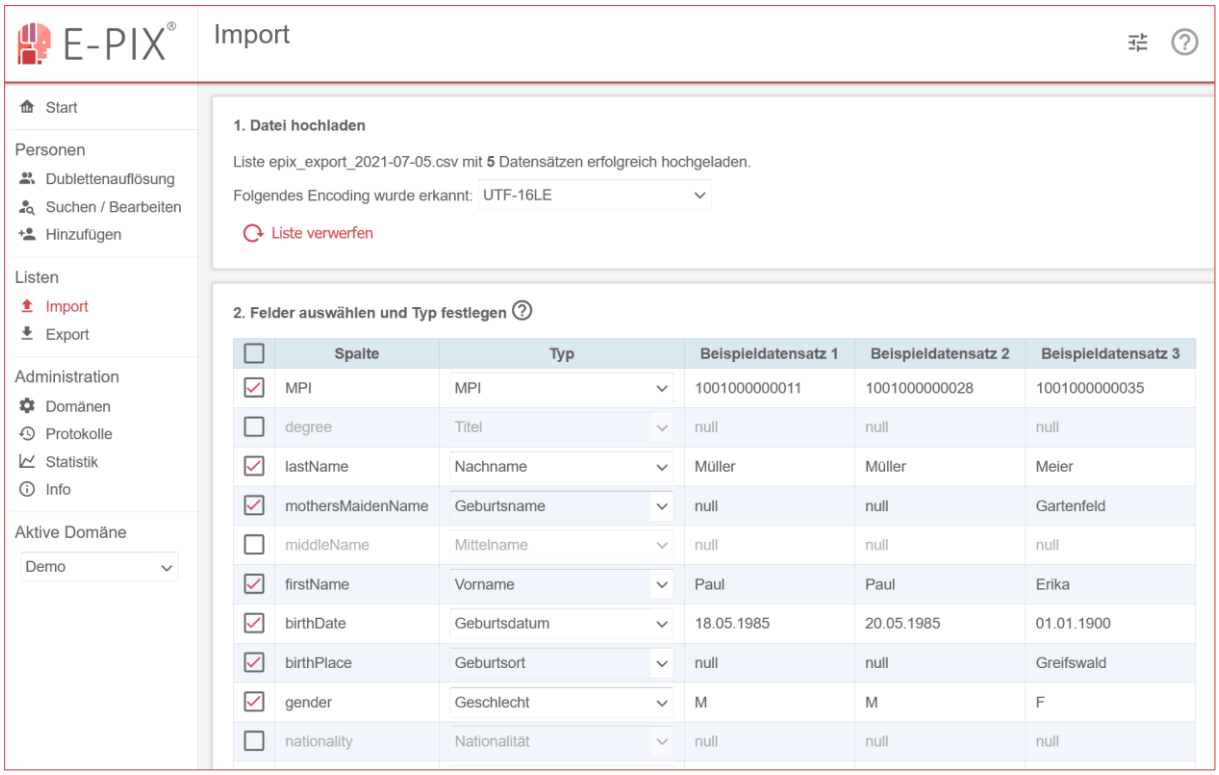

Abbildung 7-8: Oberfläche mit Vorschau der ersten eingelesenen Zeilen.

7.8 Protokoll einsehen

Um nachzuvollziehen, welche Ereignisse eingetreten sind, kann ein Protokoll unter dem Menüpunkt *Protokolle* eingesehen werden. Es stellt dar, welcher Match-Typ durch das Record Linkage für die übertragenden Personendaten errechnet wurde (*Match*, *Möglicher 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 7-9** 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.

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

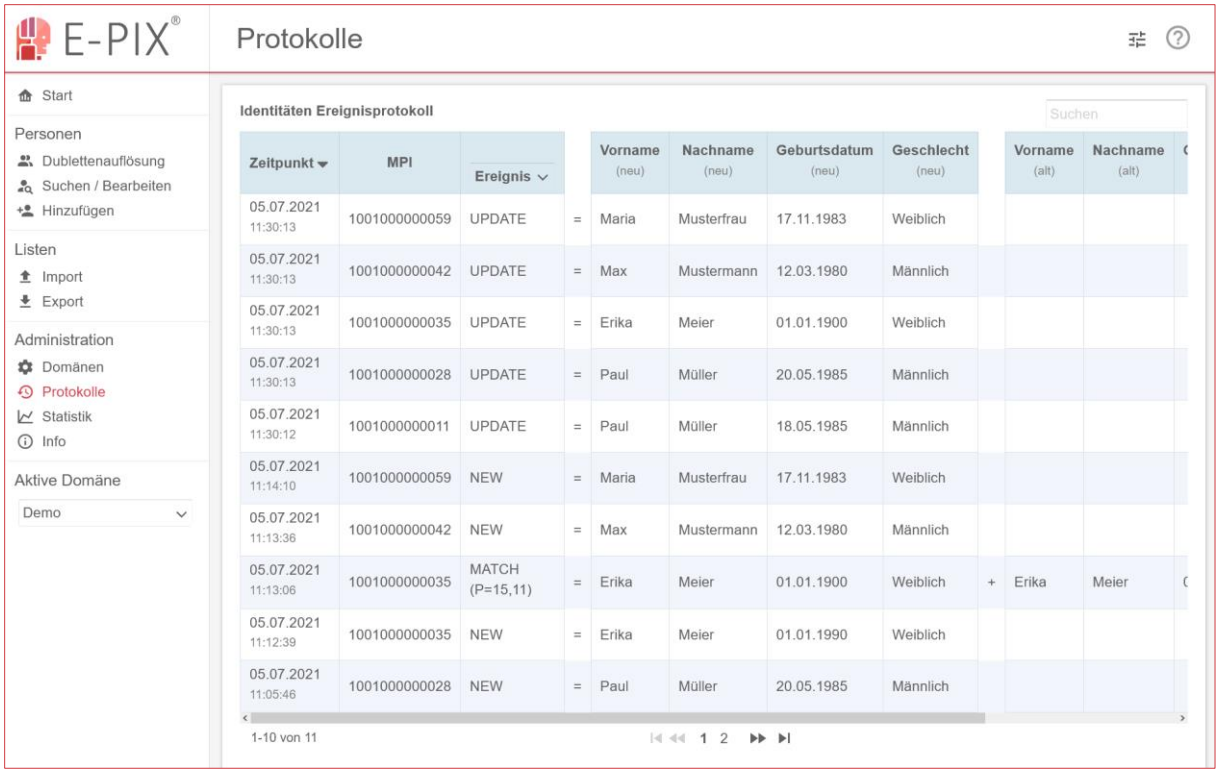

Abbildung 7-9: Oberfläche zum Einsehen des Protokolls.

7.9 Statistiken einsehen

Unter dem Menüpunkt *Statistik* kann eine **domänenübergreifende** Statistik eingesehen werden. Hierbei werden vorhandene *Mögliche Matches*, registrierte Personen, vorhandene Identitäten und aufgelöste Dubletten (separat aufgeführt als zusammengeführte und getrennte Personen) aufgelistet. Die Statistik kann als CSV oder als PDF über die jeweilige Schaltfläche heruntergeladen werden. In **Abbildung 7-10** ist exemplarisch eine Statistik dargestellt.

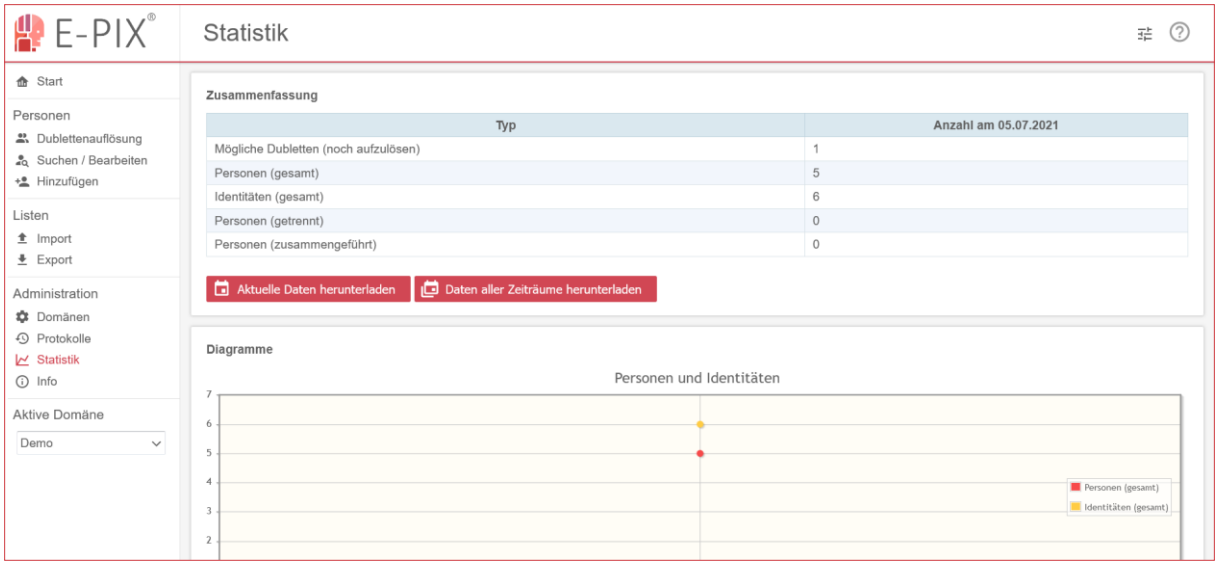

Abbildung 7-10: Oberfläche zum Einsehen der Statistik.

8 Nutzung der 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).

Patientenverwaltung (inkl. Record Linkage) und ID Administration:

```
http://example.org:8080/epix/epixService?wsdl
```

Konfiguration und Domänenmanagement:

```
http://example.org:8080/epix/epixManagementService?wsdl
```

Die dazugehörige Entwicklerdokumentation ist unter der folgenden URL zu finden.

```
https://www.ths-greifswald.de/epix/doc
```

8.1 Registrierung von Personen

Im Folgenden wird exemplarisch die Registrierung einer Person vorgestellt. Mit der Funktion `requestMPI` wird die entsprechende Person registriert, sofern diese noch nicht in der jeweiligen Domäne enthalten ist. Wenn die Person bereits registriert wurde, bzw. Personendaten mit einer sehr hohen Übersteinstimmung vorhanden sind, so wird keine neue Person angelegt, sondern eine neue Identität der entsprechenden Person zugeordnet (vgl. Nebenidentitäten in **Abschnitt 4**). Der Abgleich findet mittels Record Linkage statt. Das Ergebnis dieses Vorgangs (Match-Typ, vgl. **Tabelle 7-1**) wird in der Antwort auf die Anfrage geliefert. Zudem wird ein MPI (vgl. **Abschnitt 2**) geliefert, der für die angegebene Domäne eindeutig ist. Eine exemplarische Anfrage ist in **Abbildung 8-1** abgebildet.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:ser="http://service.epix.ttp.icmvc.emau.org/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:requestMPI>
         <domainName>Dummy</domainName>
         <identity>
            <birthDate>1985-05-21</birthDate>
            <firstName>Maximilian</firstName>
            <gender>M</gender>
            <lastName>Mustermann</lastName>
            <contacts>
               <city>Musterstadt</city>
               <street>Musterstraße 11</street>
               <zipCode>12345</zipCode>
            </contacts>
         </identity>
         <sourceName>dummy_safe_source</sourceName>
      </ser:requestMPI>
   </soapenv:Body>
</soapenv:Envelope>
```

Abbildung 8-1: Exemplarische Anfrage zur Registrierung einer Person.

Die Antwort auf die eben gezeigte Anfrage ist in **Abbildung 8-2** dargestellt.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:requestMPIResponse xmlns:ns2="http://service.epix.ttp.icmvc.emau.org/">
         <return>
            <matchStatus>NO_MATCH</matchStatus>
            <person>
               <mpiId>
                  [...]
                  <value>1001000000059</value>
               </mpiId>
               [...]
            </person>
         </return>
      </ns2:requestMPIResponse>
   </soap:Body>
</soap:Envelope>
```

Abbildung 8-2: Gekürzte Antwort auf die Anfrage zur Registrierung einer Person.

Mit der Funktion `requestMPIBatch` können mehrere Personen innerhalb einer Anfrage registriert werden.

Wenn eine Person angelegt wurde und nachträglich Attribute verändert werden sollen, erfolgt dies mit der Funktion `updatePerson`.

ⓘ Was bedeutet der `matchCode` MULTIPLE-MATCH?

Der übergebene Datensatz führt würde bei mehreren Personen zu einem *automatischen Match* (siehe **Tabelle 7-1**) führen. Deshalb werden in diesem Fall die Datensätze nicht zusammengeführt, sondern eine neue Person angelegt und eine zugehörige Liste mit *möglichen Matches* (siehe **Tabelle 7-1**) angelegt.

ⓘ Welche MPI-ID wird bei einem *possible Match/möglichen Match* zurückgegeben?

Bei einem *möglichen Match* (siehe **Tabelle 7-1**) wird zunächst eine neue Person angelegt und eine neue MPI-ID erzeugt. Die *möglichen Matches* können im Nachgang aufgelöst werden. Zurückgeliefert wird die neue MPI-ID der neu angelegten Person.

8.2 Personen per MPI suchen

Innerhalb einer Domäne kann mittels des eindeutigen MPI eine Person gesucht werden. Hierzu steht die Funktion `getPersonByMPI` bereit. In der entsprechenden Anfrage muss der Domänenname und der MPI der gesuchten Person angegeben werden. In **Abbildung 8-3** ist eine exemplarische Anfrage zum Suchen einer Person mittels des MPIs dargestellt.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:ser="http://service.epix.ttp.icmvc.emau.org/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:getPersonByMPI>
         <domainName>Dummy</domainName>
         <mpiId>1001000000059</mpiId>
      </ser:getPersonByMPI>
   </soapenv:Body>
</soapenv:Envelope>
```

Abbildung 8-3: Exemplarische Anfrage zum Suchen einer Person mittels des dazugehörigen MPIs.

In **Abbildung 8-4** ist ein Auszug aus der entsprechenden Antwort dargestellt. In der Antwort sind alle Personendaten enthalten.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns2:getPersonByMPIResponse xmlns:ns2="http://service.epix.ttp.icmvc.emau.org/">
         <return>
            <mpiId>
               [...]
               <value>1001000000059</value>
            </mpiId>
            [...]
         </return>
      </ns2:getPersonByMPIResponse>
   </soap:Body>
</soap:Envelope>
```

Abbildung 8-4: Antwort auf die Anfrage zum Suchen einer Person mittels des MPIs.

Mit der Funktion `getPersonByLocalIdentifier` können statt mittels MPI mit einem Lokalem Identifier die dazugehörigen Personendaten abgerufen werden.

8.3 Alle Personendaten zu einer Domain

Mit der Funktion `getPersonsForDomain` können alle Personendaten aller Personen einer Domäne abgerufen werden. Hierzu muss in der entsprechenden Anfrage die jeweilige Domäne angegeben werden. In der Antwort sind alle Personendaten aufgelistet. In **Abbildung 8-5** ist exemplarisch eine Anfrage dargestellt.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:ser="http://service.epix.ttp.icmvc.emau.org/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:getPersonsForDomain>
         <domainName>Dummy</domainName>
      </ser:getPersonsForDomain>
   </soapenv:Body>
</soapenv:Envelope>
```

Abbildung 8-5: Anfrage um alle Personendaten einer Domain abzurufen.

Analog dazu können mit der Funktion `getIdentitiesForDomain` alle Identitäten aus einer Domäne abgerufen werden.

9 Konfiguration von E-PIX Domänen

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 255.000 Patienten) aus den beteiligten Registerstellen und bei der Identifikation der Patienten erforderlich. Jede

Abweichung in den demografischen Informationen, sei es nur ein Zeichen, soll dem Datentreuhänder 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 Teilnehmern die nötige manuelle Nacharbeit zum Auflösen potentieller Matches, 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 Patienten, 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 sollen potentielle Matches signalisiert werden) kann über einer Vielzahl von Schwellwerten und Parametern konfiguriert werden.

Je niedriger die Schwellwerte für potentielle Matches gewählt werden desto mehr Matching-Paare von Patienten werden signalisiert und umso mehr manuelle Kontrolle dieser möglichen Matches durch den Datentreuhänder ist erforderlich.

Die Konfiguration des E-PIX erfolgt je Domäne. Um die Vielzahl der Anpassungsmöglichkeiten zu verstehen, werden nachfolgende grundlegend die Matching-Mechanismen und die möglichen Konfigurationsoptionen beschrieben.

⚠ **Hinweis:** Die Konfiguration des E-PIX sollte stets vor produktivem Beginn des Vorhabens erfolgen. Der E-PIX entscheidet über den Matching-Zustand eines Patienten auf Basis der bereits vorhandenen Daten und der aktuellen Konfiguration. Aktualisiert man die Konfiguration obwohl bereits Daten in der Datenbank vorhanden sind, müssten diese erneut in eingespielt werden (idealerweise in ein leeres System), um die Korrektheit der Matching-Bewertung gemäß der neuen Konfiguration gewährleisten zu können.

⚠ **Hinweis:** Eine Standardkonfiguration wird beim E-PIX mitgeliefert und kann als Grundlage für Änderungen oder Erweiterungen verwendet werden. Zu finden ist diese im Verzeichnis `E-PIX/docker/standard/demo_import`.

9.1 Hintergrund

Für die Registrierung eines Personendatensatzes, kann der Matching-Prozess in mehrere Teilschritte gegliedert werden. Hierzu wird zunächst eine Vorselektion der Personendatensätze vorgenommen (Blocking), sodass eine verringerte Anzahl von Datensätzen für ein unschärferes Matching (siehe **Abschnitt 9.1.1**) abgeglichen werden muss. Die Personendatensätze, die dabei eine hinreichende Ähnlichkeit aufweisen, werden mithilfe eines genaueren Abgleichs unter Zuhilfenahme von höheren Schwellwerten und weiteren Feldern vergleichen (siehe **Abschnitt 9.1.2**). Je nach Ähnlichkeit können

die verbleibenden Datensätze als Dublette, keine Dublette oder mögliche Dublette klassifiziert werden. In **Abbildung 9-1** ist der Prozess der Registrierung stark vereinfacht dargestellt.

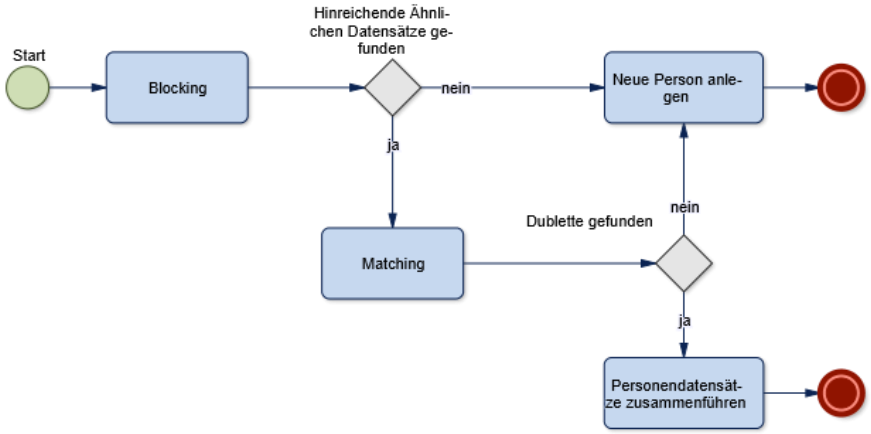

Abbildung 9-1: Vereinfachter Ablauf des Matching-Prozesses.

9.1.1 Blocking

Grundsätzlich dient das Blocking einer ersten unscharfen Selektierung potentieller Duplikate. Im E-PIX ist dieser Vorgang frei konfigurierbar, indem eine reduzierte Teilmenge von Attributen verwendet wird, um einen ersten Abgleich durchzuführen. Dabei wird eine Ähnlichkeit zwischen jeweils zwei Datensätzen (dem zu registrierenden und einem bereits registrierten Datensatz) ermittelt. Die Attribute, welche hierzu abgeglichen werden sollen, können gewählt werden und betreffend des Schwellwerts konfiguriert werden (siehe **Abschnitt 9.4.11**). Das Blocking dient der Steigerung der Performance und verringert insbesondere bei großen Datenbeständen die Dauer eines Abgleichs.

9.1.2 Matching

Wird beim Blocking eine hinreichende Ähnlichkeit mit bestimmten Datensätzen ermittelt, so werden diese in einem weiteren Abgleich genauer verglichen. Hierzu werden weitere Attribute hinzugezogen, welche ebenfalls ähnlich wie beim Blocking konfiguriert (siehe **Abschnitt 9.4.11**) werden können. Mithilfe dieses genaueren Abgleichs kann klassifiziert werden, ob ein Duplikat vorliegt oder nicht[7].

9.2 XML-basierte Konfiguration

Die Konfiguration des E-PIX wird im XML-Format definiert. Über das Web-Frontend des E-PIX kann die Konfiguration von E-PIX-Domänen (Projekte, Studien, Quellsysteme) angezeigt und editiert werden.

[7] Tatsächlich findet eine feinere Unterteilung statt (vgl. **Tabelle 7-1**)

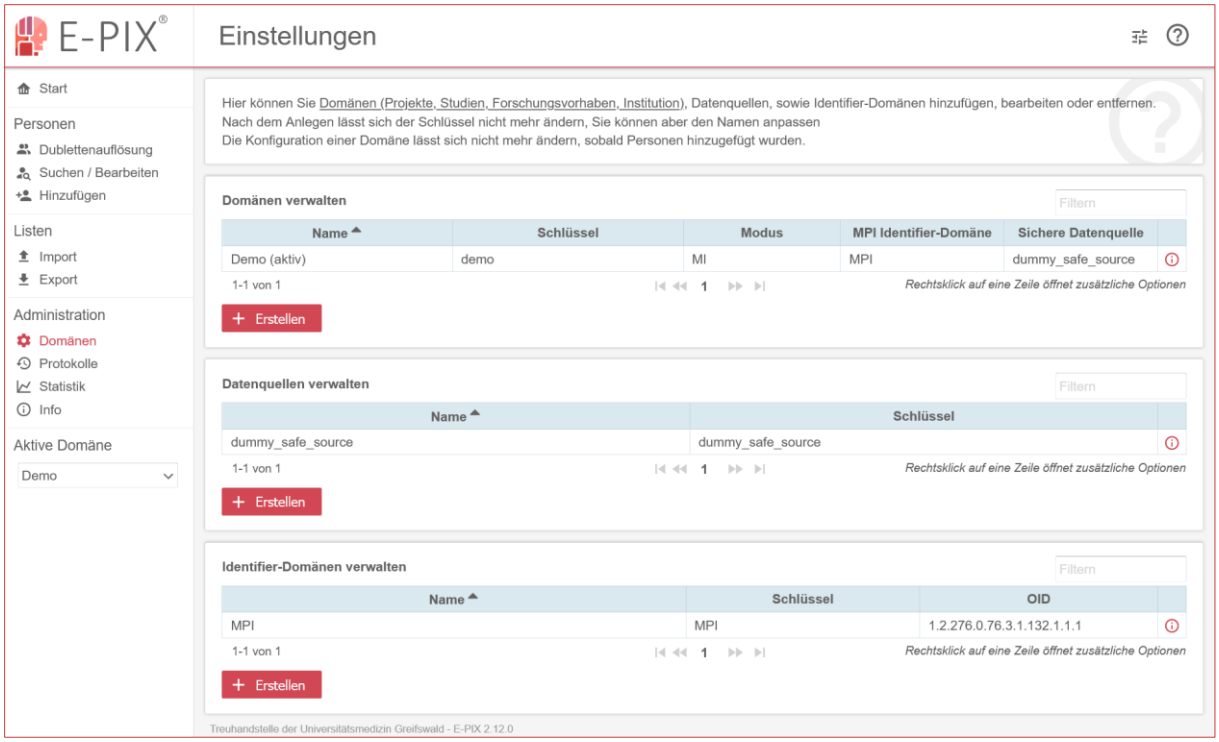

Abbildung 9-2: Das Anzeigen und Editieren der aktuellen Konfiguration einer E-PIX-Domäne ist direkt über das Web-Frontend möglich.

Abbildung 9-3 zeigt die Struktur der Konfiguration und listet alle Elemente, die bei der Domänenkonfiguration verwendet werden. Die Struktur gibt dabei an, welche Elemente anderen Elementen untergeordnet sind. Eine Erläuterung aller Elemente mit Beispielen und validen Wertebereichen folgt im nächsten Abschnitt.

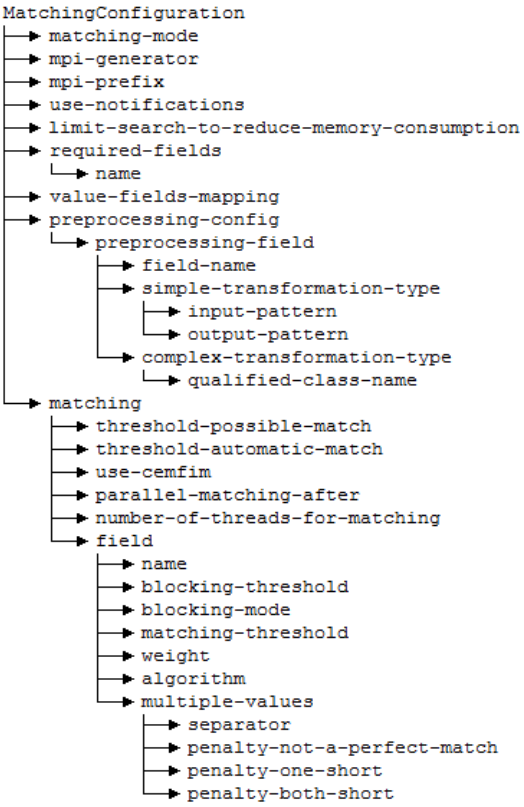

Abbildung 9-3: Alle Elemente, die bei der Konfiguration der Domäne verwendet werden können.

Im E-PIX sind mehrere Felder vorgegeben. Je nach Feld wird standardmäßig eine formale Prüfung von Eingaben durchgeführt. So würde beispielsweise der 31.02. nicht als Geburtsdatum angenommen werden. In **Tabelle 9-1** sind alle vordefinierten Felder aufgelistet. Hierbei ist zu beachten, dass die Felder der Kontaktdaten nicht für das Matching verwendet werden können.

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	1980-03-12
gender	Geschlecht (wird intern auf mittels eines Buchstaben angegeben) *m* für male (männlich), *f* für female (weiblich), *o* für other (sonstige), *u* für Unknown (unbekannt) und *x* für divers	f

externalDate	Freies Feld für ein Datum, welches im E-PIX nur gespeichert, aber nicht weiter prozessiert wird Format: JJJJ-MM-TT	2019-04-30
birthPlace	Geburtsort	Berlin
race	Ethnizität	Kaukasier
religion	Religion	Christentum
mothersMaidenName	Geburtsname	Müller
degree	Abschluss	Mittlerer Schulabschluss
motherTongue	Muttersprache	deutsch
nationality	Nationalität/Staatsangehörigkeit	deutsch
civilStatus	Familienstand	ledig
value1 - value10	Felder dessen Werte je Projekt/Studie individuell belegt werden können. Zusätzlich kann der Feldname für die Weboberfläche mittels eines Labels geändert werden (siehe **Abschnitt 9.4.8**). Die Felder haben in der Datenbank unterschiedliche Längen: value1-5 50 Zeichen value6+7 255 Zeichen value8+9 1.000 Zeichen value10 10.000 Zeichen	Todesdatum
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)	anna.schmidt@beispiel.de
externalDate	Freies Feld für ein Datum, welches im E-PIX nur gespeichert, aber nicht weiter prozessiert wird (Kontaktdaten)	2019-06-27

	Format: JJJJ-MM-TT	
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*

9.3 Die Standard-Konfiguration

Dem E-PIX ist eine Standard-Konfiguration für Domänen beigelegt. Diese kann ohne Weiteres für viele Projekte vorerst ausreichend sein. Hierbei ist jedoch, wie oben bereits erwähnt zu beachten, dass eine nachträgliche Änderung der Domänen-Konfiguration für eine korrekte Bewertung des Matchings eine komplette Neuregistrierung aller bereits bekannten Datensätze nach sich ziehen muss.

Die Standard-Konfiguration 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 pre-processing (siehe **Abschnitt 9.4.10**) aufbereitet. Für das Blocking werden die Felder `firstName` und `birthDate` verwendet. Für das Feld `firstName` werden zudem Multiple-Values (siehe **Abschnitt 9.4.11.7**) genutzt. Ein Matching findet mithilfe aller vier Felder statt. Für einen Abgleich wird immer die Levenshtein-Distanz verwendet[8]. In **Tabelle 9-2** sind die Felder zur Übersicht dargestellt.

Tabelle 9-2: Verwendete Felder mit Schwellwerten und Wichtung in der Standard-Domänenkonfiguration.

Feldname	Blocking-Schwellwert	Matching-Schwellwert	Wichtung des Felds
firstName	0,4	0,8	8
lastName	*Nicht für Blocking verwendet*	0,8	6
birthDate	*Nicht für Blocking verwendet*	0,75	3
gender	0,6	1,0	9

[8] Weitere Vergleichsmöglichkeiten sind implementiert (vgl. **Tabelle 9-9**)

9.4 Struktur und Inhalt der Konfiguration

Das Element `MatchingConfiguration` ist das Wurzelelement. Alle Elemente sind diesem Element untergeordnet.

9.4.1 matching-mode

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 KAS/KIS übermittelt werden und bereits Identifizierer vergeben wurden und bereits ein Record Linkage durchgeführt wurde. In **Tabelle 9-3** sind die zwei Modi im Detail erläutert.

Tabelle 9-3: Unterstützte Matching-Modes

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

In **Listing 1** ist exemplarisch gezeigt, wie der Modus definiert wird.

Beispiel:

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

Listing 1: XML-Code zum Definieren des Matching-Modes.

9.4.2 mpi-generator

Wird eine Person im E-PIX erstmalig eingetragen, so erhält diese eine MPI-ID. Die Erzeugung einer MPI-ID wird dabei durch einen Generator durchgeführt. Derzeit ist im E-PIX ein Generator (`EAN13Generator`) integriert, welcher eindeutige MPI-IDs erzeugt. Weitere Generatoren können implementiert werden. In **Listing 2** ist die Angabe des Generators dargestellt.

```
<mpi-generator>
      org.emau.icmvc.ttp.epix.gen.impl.EAN13Generator
</mpi-generator>
```

Listing 2: XML-Code zum Definieren des MPI-Generators.

9.4.3 mpi-prefix

Die ersten Ziffern im MPI können mithilfe eines Präfixes festgelegt werden. Jeder MPI enthält damit die angegebene Ziffernfolge[9]. Wird beispielsweise das Präfix *1001* gesetzt, so könnte ein resultierender MPI so aussehen: *1001000000035*. In **Listing 3** ist dargestellt, wie ein Präfix definiert werden kann.

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

Listing 3: XML-Code zum Definieren des MPI-Präfixes.

9.4.4 use-notifications

Das Element `use-notifications` dient dazu, bei Änderungen von Datensätzen im E-PIX andere Systeme zu benachrichtigen. Derzeit läuft diese Benachrichtigung über den Dispatcher und ist nicht mit beliebigen Systemen kombinierbar. Mit dem Wert `true` wird die Benachrichtigung aktiviert und mit dem Wert `false` deaktiviert. Im **Listing 4** ist beispielhaft die Benachrichtigung deaktiviert.

Aktuell ist diese Funktionalität nur für die Funktion „mergePerson“ implementiert.

```
<use-notifications>false</use-notifications>
```

Listing 4: XML-Code zum Aktivieren der Benachrichtigungen über den Dispatcher.

9.4.5 limit-search-to-reduce-memory-consumption

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 ein Patient gesucht werden kann. Wenn die Option auf `true` gesetzt wird, dann können die Patienten nur anhand der Felder gesucht werden, die auch für das Matching (siehe **Abschnitt 9.4.11.6**) verwendet werden. In **Listing 5** wird exemplarisch das deaktivieren dieser Option dargestellt.

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

Listing 5: XML-Code zum Deaktivieren der Option zur Reduzierung des benötigten Arbeitsspeichers.

9.4.6 persist-mode

Das Element `persist-mode` legt den Modus fest, wie Identitätsdaten 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

[9] Ob das Präfix verwendet wird, hängt davon ab, ob der genutzte MPI-Generator das Präfix berücksichtigt. Der mitgelieferte Generator (`EAN13Generator`) berücksichtigt das Präfix.

auf Basis von Bloomfiltern durchgeführt werden. Weitere Informationen zum Bloomfilter sind unter **Abschnitt 9.4.9** zu finden. In **Listing 6** wird exemplarisch die Festlegung des Persist-Modes dargestellt.

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

Listing 6: XML-Code zum Wählen des Persist-Modes.

9.4.7 required-fields

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 **Listing 7** ist exemplarisch eine Konfiguration dargestellt, wodurch zur Registrierung die Felder Vorname, Nachname, Geburtsdatum und Geschlecht übermittelt werden müssen.

```
<required-fields>
      <name>firstName</name>
      <name>lastName</name>
      <name>birthDate</name>
      <name>gender</name>
</required-fields>
```

Listing 7: XML-Code zur Festlegung der Pflichtfelder, die für eine Registrierung übermittelt werden müssen.

9.4.8 value-fields-mapping

Die Felder *value1* – *value10* können für beliebige Werte verwendet werden. Die entsprechenden Felder können mit einem sprechenden Namen versehen werden, welcher in der Weboberfläche (vgl. **Abbildung 9-4**) dargestellt wird. Es handelt sich dabei jedoch nur um ein Label, für etwaige weitere Konfigurationen wird weiterhin der Feldname verwendet. In **Listing 8** wird exemplarisch die Vergabe von Labeln für die Felder *value1* und *value2* dargestellt.

```
<value-fields-mapping>
      <value1>KV-Name</value1>
      <value2>KV-Nummer</value2>
</value-fields-mapping>
```

Listing 8: XML-Code zum Definieren von Labeln für *value*-Felder.

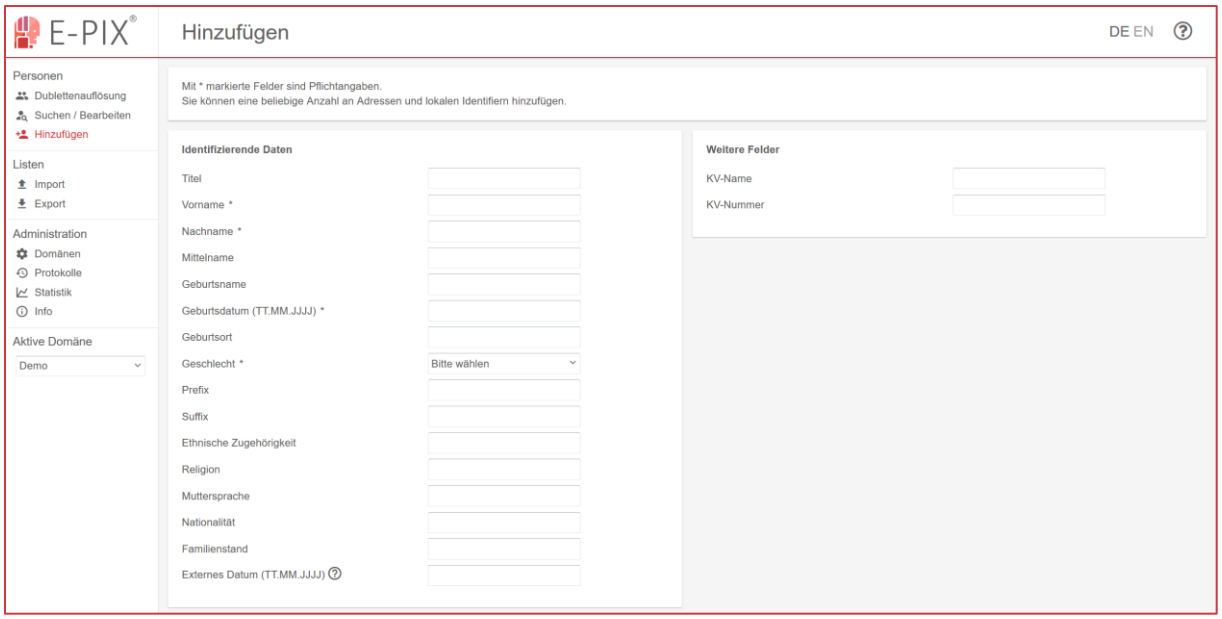

Abbildung 9-4: Weboberfläche zur Registrierung eines Person. Rechts sind die gemappten Felder dargestellt.

9.4.9 privacy

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

9.4.9.1 bloomfilter-config

Die Bloomfilter-Konfiguration enthält alle Einstellungen für einen Bloomfilter. Dabei ist zu beachten, dass 1) das Feld in dem der Bloomfilter gespeichert wird, die Länge des Bloomfilters zulässt und 2) der Bloomfilter aus normalisierten bzw. aus aufbereiteten Werten generiert wird (siehe **Abschnitt 9.4.10**). Der Bloomfilter kann wie andere Felder auch zum Matching verwendet werden. Hierzu stehen entsprechende Vergleichsverfahren zur Verfügung. Im **Abschnitt 9.4.11.6.6** sind weitere Informationen dazu enthalten. Zu beachten ist, dass Bloomfilter im E-PIX im base64-Format gespeichert werden.

⚠ **Hinweis:** Der E-PIX unterstützt mehrere Verfahren 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 Abwägen 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.

In der nachfolgenden Tabelle sind alle Elemente zur Bloomfilter-Konfiguration aufgeführt. Ein exemplarisches Beispiel ist in **Listing 9** aufgeführt.

Tabelle 9-4: Elemente der Bloomfilter-Konfiguration.

Element-Name	Beschreibung	Beispiel
`algorithm`	Angabe des Algorithmus, welcher das Verfahren zur Erzeugung des Bloomfilters implementiert. Eine Auflistung von den unterstützen Algorithmen ist in **Tabelle 9-5** zu finden.	org.emau.icmvc.ttp. ↩ deduplication.impl. ↩ bloomfilter. ↩ RandomHashingStrategy
`field`	Feld der Identität, in dem der Bloomfilter gespeichert werden soll. Dabei zu ist beachten, dass das Feld ggf. überschrieben wird und die Länge des Bloomfilters 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 Bloomfilters in Bits.	1000
`ngrams`	Länge der N-Gramme, die für die Erzeugung des Bloomfilters 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 Bloomfilter gesetzt werden. Beim Doube-Hashing wird von Iterationen gesprochen. Beim Random-Hashing handelt es sich um die Anzahl der generierten Zufallspositionen.	25
`fold`	Der E-PIX unterstützt ein XOR-Folding von Bloomfilern nach Schnell et al.[10]. Der Wert gibt die Anzahl der Faltungen an. Zu beachten ist, dass der Wert+1 ein ganzzahliger Teiler von der Länge des Bloomfilters sein muss ($n+1 \mid Länge$). Wird 0 angegeben, wird der Bloomfilter nicht bearbeitet. Pro Faltung halbiert sich die Länge des resultierenden Bloomfilters.	Bei Bloomfilter 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 E-PIX unterstützt das Generieren von Balanced-Bloomfiltern (Schnell et al.[11]). Das Element	462945623209

[10] https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3527984
[11] https://ieeexplore.ieee.org/document/7836669

	`balanced` enthält ein Feld `seed`, welches einen Zahlenwert enthält. Dieser stellt den Seed-Wert des Zufallsgenerators dar. Wird dieses Element (`balanced`) nicht angegeben, wird kein Balanced-Bloomfilter erzeugt. Der Balanced-Bloomfilter führt zu einer Verdopplung der resultierenden Bloomfilter-Länge.
`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.

Tabelle 9-5: Unterstütze Algorithmen zur Generierung von Bloomfiltern.

Algorithmus	Beschreibung
`org.emau.icmvc.ttp.deduplication.impl.bloomfilter.RandomHashingStrategy`	
`org.emau.icmvc.ttp.deduplication.impl.bloomfilter.DoubleHashingStrategy`	
`org.emau.icmvc.ttp.deduplication.impl.bloomfilter.DoubleHashingStrategyFaster`	

```
<privacy>
      <bloomfilter-config>
            <algorithm>org.emau.icmvc.ttp.deduplication.                ↩
                  impl.bloomfilter.RandomHashingStrategy
            </algorithm>
            <field>value8</field>
            <length>1000</length>
            <ngrams>2</ngrams>
            <bits-per-ngram>15</bits-per-ngram>
            <fold>1</fold>
            <alphabet>ABCDEFGHIJKLMNOPQRSTUVWXYZ .-0123456789</alphabet>
            <balanced>
                  <seed>4623829476</seed>
            </balanced>
            <source-field>
                  <name>firstName</name>
                  <seed>456542343</seed>
            </source-field>
            <source-field>
                  <name>lastName</name>
                  <seed>374027465</seed>
            </source-field>
      </bloomfilter-config>
      <bloomfilter-config>
            <algorithm>org.emau.icmvc.ttp.deduplication.                ↩
                  impl.bloomfilter.DoubleHashingStrategy
            </algorithm>
            <field>value6</field>
            <length>500</length>
            <ngrams>2</ngrams>
            <bits-per-ngram>15</bits-per-ngram>
            <source-field>
                  <name>firstName</name>
                  <salt-field>birthDate</salt-field>
            </source-field>
            <source-field>
                  <name>gender</name>
                  <salt-value>Q2fh-Fk2#CjP+s5#</salt-value>
            </source-field>
      </bloomfilter-config>
</privacy>
```

Listing 9: Verkürzte exemplarische Konfiguration von zwei Bloomfiltern.

9.4.10 preprocessing-config

Mithilfe des pre-prossessings 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. Ein pre-processing 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 10** ist ein einfaches Beispiel für die Konfiguration der Aufbereitung des Feldes für den Vornamen.

```
<preprocessing-config>
      <preprocessing-field>
            <field-name>firstName</field-name>
            <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.icmvc.ttp.                  ↩
                  deduplication.preprocessing.impl.                        ↩
                  ToUpperCaseTransformation                                ↩
                </qualified-class-name>
            </complex-transformation-type>
      </preprocessing-field>
<preprocessing-config>
```

Listing 10: Exemplarischer XML-Code mit allen Elementen für ein pre-processing eines Feldes.

In **Abschnitt 9.4.10.2** wird das Element `field-name`, in **Abschnitt 9.4.10.3** wird das Element `simple-transformation-type` und in **Abschnitt 9.4.10.4** das Element `complex-transformation-type` erläutert.

9.4.10.1 preprocessing-field

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. `complex-transformation-type` werden stets nach `simple-transformation-type` verarbeitet[12].

9.4.10.2 field-name

Das Element `field-name` gibt das Feld an, welches aufbereitet werden soll. Die Bezeichnungen für die jeweiligen Felder sind in **Tabelle 9-1** angegeben.

[12] Die Festlegung der Reihenfolge wird demnächst implementiert.

9.4.10.3 simple-transformation-type

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 11** sind zwei `simple-transformation-type` dargestellt. Die erste Transformation dient zum Entfernen von allen Leerzeichen aus einem Feld, die Zweite ersetzt das Zeichen *é* durch *e*.

```
...
<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>é</input-pattern>
      <output-pattern>e</output-pattern>
</simple-transformation-type>
...
```

Listing 11: XML-Code zur Definition zweier einfacher Transformationen.

9.4.10.4 complex-transformation-type

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 9-6** 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 9-6: Unterstützte Transformationen für `complex-transformation-type`.

Transformation	Beschreibung	Beispiel
`ToUpperCaseTransformation`	Alle Kleinbuchstaben werden durch Großbuchstaben ersetzt.	Anna ➔ ANNA
`CharsMutationTransformation`	Ersetzt Umlaute.	München ➔ Muenchen
`TrimTransformation`	Entfernt führende und nachfolgende Leerzeichen.	␣␣An␣na␣ ➔ An␣na

In **Listing 12** wird exemplarisch gezeigt, wie führende und nachfolgende Leerzeichen für das Record Linkage mittels Transformator entfernt werden.

```
...
<complex-transformation-type xsi:type="ma:ComplexTransformation">
      <qualified-class-name>
            org.emau.icmvc.ttp.deduplication.preprocessing.impl.        ↩
            TrimTransformation
      </qualified-class-name>
</complex-transformation-type>
...
```

Listing 12: XML-Code zur Definition eines Transformators zum Entfernen führender und nachfolgender Leerzeichen.

9.4.11 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. Die Schwellwerte können dem Verfahren entsprechend angepasst werden. Werden die Elemente nicht angegeben, werden Standardwerte gesetzt. In **Tabelle 9-7** sind die empfohlenen und Standard-Schwellwerte dargestellt.

Tabelle 9-7: Empfohlene und Standard-Schwellwerte für *Automatic Match* und *Possible Match*.

Schwellwert	Wert	Standardwert
`threshold-automatic-match`	14,5	20
`threshold-possible-match`	2,99	4

Die Match-Typen wurden In **Abschnitt 7.2** erläutert. In **Tabelle 7-1** sind alle Match-Typen aufgeführt und entsprechend erläutert.

9.4.11.1 threshold-possible-match

Mit dem Element `threshold-possible-match` kann der Schwellwert für *Possible Matches* (vgl. **Tabelle 9-7**) definiert werden. Überschreitet die ermittelte Wahrscheinlichkeit den angegeben Wert (und unterschreitet den Schwellwert `threshold-automatic-match`), so wird der Match-Typ *Possible Match* als Ergebnis des Record Linkages zurückgegeben. In **Listing 13** ist die Definition des Schwellwert dargestellt.

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

Listing 13: XML-Code zur Definition des Schwellwerts für *Possible Matches*.

9.4.11.2 threshold-automatic-match

Mit dem Element `threshold-automatic-match` kann der Schwellwert für *Automatic Matches* (vgl. **Listing 14**) definiert werden. Unterscheiden sich die abgeglichenen Datensätze voneinander, die ermittelte Wahrscheinlichkeit überschreitet jedoch den angegebenen Wert, so wird der Match-Typ

Automatic Match als Ergebnis des Record Linkages zurückgegeben. In **Listing 14** ist die Definition des Schwellwert dargestellt.

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

Listing 14: XML-Code zur Definition des Schwellwerts für *Automatic Matches*.

9.4.11.3 use-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 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 9-8** entnommen werden.

Tabelle 9-8: Verhalten des E-PIX, je nachdem wie das Element `use-cemfim` definiert wurde.

Nummer	CEMFIM	Mehr als 1 Match vorhanden (mit anderer Person)	Verhalten
1	`true`	Ja	Fehler: Ein Identifier darf nur einer Person pro Domäne zugeordnet sein.
2	`false`	Ja	Die Identität wird gespeichert und als Possible Match hinterlegt.
3	`true`	Nein	Die Identität wird gespeichert und als Possible Match hinterlegt.
4	`false`	Nein	Die Identität wird gespeichert und als Possible Match hinterlegt.

In **Listing 15** ist exemplarisch die Definition dargestellt.

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

Listing 15: XML-Code zur Definition des *use-cemfim*-Wertes.

9.4.11.4 parallel-matching-after

Der E-PIX unterstützt Multithreading, wodurch die Performance gesteigert wird. Bei einer niedrigen Anzahl von registrierten Identitäten ist es performanter einen sequenziellen Abgleich durchzuführen. Deshalb kann mit dem Element `parallel-matching-after` definiert werden, ab wieviel registrierten Identitäten ein paralleler Abgleich, also verteilt auf mehrere Threads, stattfinden soll. Der Wert ist abhängig von der Rechenleistung des 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 16** ist exemplarisch die Definition dargestellt.

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

Listing 16: XML-Code zur Definition der Anzahl registrierter Personen, ab denen der E-PIX Multithreading verwendet.

9.4.11.5 number-of-threads-for-matching

Die Anzahl der verwendeten Threads kann 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 Optimalfall 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 17** ist die exemplarische Definition der Anzahl der verwendeten Threads dargestellt.

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

Listing 17: XML-Code zur Definition der Anzahl der verwendeten Threads.

9.4.11.6 field

Mit dem Element `field` werden alle Felder definiert, die im Rahmen des Blockings oder/und Matchings verwendet werden. Jedes Feld wird hierfür separat konfiguriert. In **Listing 18** ist exemplarisch angegeben, wie eine Konfiguration eines Feldes aussehen kann. Im Folgenden werden die einzelnen Elemente erläutert.

```
<field>
      <name>gender</name>
      <matching-threshold>0.75</matching-threshold>
      <weight>3</weight>
      <algorithm>
            org.emau.icmvc.ttp.deduplication.impl.LevenshteinAlgorithm
      </algorithm>
</field>
```

Listing 18: XML-Code zur exemplarischen Konfiguration eines Felders, welches zum Matching verwendet wird.

9.4.11.6.1 name

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 19** ist exemplarisch der Wert „gender“ angegeben, wenn das Geschlecht z.B. für das Blocking verwendet werden soll.

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

Listing 19: XML-Code zur Definition des Feldes für das Record Linkage.

9.4.11.6.2 blocking-threshold

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 20** wird exemplarisch ein Schwellwert definiert.

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

Listing 20: XML-Code zur Definition eines Schwellwertes für das Blocking von einem Feld.

9.4.11.6.3 blocking-mode

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 verwendet werden. Wenn das Element `blocking-mode` nicht angegeben wurde, wird standardmäßig `TEXT` verwendet. In **Listing 21** ist die Definition von `blocking-mode` exemplarisch für Zahlenvergleiche dargestellt.

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

Listing 21: XML-Code zur Definition der Blocking-Vergleichsmethode.

9.4.11.6.4 matching-threshold

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 22** ist exemplarisch eine Schwelle definiert.

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

Listing 22: XML-Code zur Definition eines Schwellwertes für das Matching von einem Feld.

9.4.11.6.5 weight

Mit dem Element `weight` kann eine Wichtung 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 23** ist exemplarisch eine Wichtung angegeben.

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

Listing 23: XML-Code zur Wichtung eines Feldes.

9.4.11.6.6 algorithm

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 9-9** 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 9-9: Unterstütze Algorithmen für das Matching.

Algorithmus	Beschreibung
`ColognePhoneticAlgorithm`	Vergleicht zwei Werte nach ihrem Sprachklang. Die Nachnamen Maier, Meyer und Meier würden beispielsweise als gleich gewertet werden.
`DeterministicAlgorithm`	Vergleicht zwei Werte auf exakte Gleichheit. Bei exakter Gleichheit zweier Werte ist das Ergebnis 1, bei einer Abweichung 0.
`LevenshteinAlgorithm`	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.
`SorensenDiceCoefficientCoded`	Vergleicht zwei (base64-kodierte) Bloomfilter auf Ähnlichkeit. Dabei wird der Sørensen-Dice Koeffizient verwendet. Dieser Algorithmus wird verwendet, wenn der Bloomfilter direkt im E-PIX erzeugt wurde.
`JaccardSimilarityAlgorithmCoded`	Vergleicht zwei (base64-kodierte) Bloomfilter auf Ähnlichkeit. Dabei wird der Jaccard-Koeffizient verwendet. Dieser Algorithmus wird verwendet, wenn der Bloomfilter direkt im E-PIX erzeugt wurde.
`SorensenDiceCoefficient`	Vergleicht zwei (0 und 1 basierte String-) Bloomfilter auf Ähnlichkeit. Dabei wird der Sørensen-Dice Koeffizient verwendet. Dieser Algorithmus wird verwendet, wenn der Bloomfilter

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

In **Listing 24** wird exemplarisch die Definition eines Algorithmus zum Abgleich von einem Feld definiert.

```
<algorithm>
      org.emau.icmvc.ttp.deduplication.impl.LevenshteinAlgorithm
</algorithm>
```

Listing 24: XML-Code zur Definition des Algorithmus für das Matching.

9.4.11.7 multiple-values

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 den Vornamen mit „Klaus Dieter“ oder „Dieter Klaus“ angibt[14]. Hierzu kann ein Separator definiert werden, anhand dessen die Teil-Zeichenketten ermittelt werden. In **Listing 25** ist exemplarisch ein `multi-value` dargestellt. Die enthaltenen Elemente werden im Folgenden erläutert.

```
<multiple-values>
      <separator> </separator>
      <penalty-not-a-perfect-match>0.1</penalty-not-a-perfect-match>
      <penalty-one-short>0.1</penalty-one-short>
      <penalty-both-short>0.2</penalty-both-short>
</multiple-values>
```

Listing 25: XML-Code zur Definition eines `multi-value`-Feldes.

9.4.11.7.1 separator

Mit dem Element `separator` kann ein Zeichen definiert werden, anhand dessen ein Wert in mehrere Zeichenketten aufgespalten wird. Beim Feld Vorname könnte dies beispielweise 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 26** ist die Definition eines Leerzeichens als Separator dargestellt.

[14] Dabei ist entscheidend, welcher Separator verwendet wird.

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

Listing 26: XML-Code zur exemplarischen Definition eines Leerzeichens als Separator eines multi-value-Feldes.

9.4.11.7.2 penalty-not-a-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 Teilzeichenketten matchen, aber nicht exakt gleich sind. Beispiel: Klaus Dieter und Klaas Dieter. Klaas und Klaus sind ähnlich genug und matchen daher, sie unterscheiden sich jedoch geringfügig. In Fehler! Verweisquelle konnte nicht gefunden werden. ist exemplarisch gezeigt, wie ein Wert hierfür definiert wird.

9.4.11.7.3 penalty-one-short

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 Teilzeichenketten matchen. Beispiel: Klaus Dieter und Klaus. Klaus matcht, Dieter fehlt jedoch in einem Feld. In

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

Listing 27 ist exemplarisch gezeigt, wie ein Wert hierfür definiert wird.

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

Listing 27: XML-Code zur exemplarischen Definition des *penalty-one-short*-Wertes.

9.4.11.7.4 penalty-both-short

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 Teilzeichenketten matchen. Beispiel: Klaus Dieter und Dieter Erhardt. In **Listing 28** ist exemplarisch gezeigt, wie ein Wert hierfür definiert wird.

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

Listing 28: XML-Code zur exemplarischen Definition des *penalty-both-short*-Wertes.

10 Publikationen und Vorträge

- Hampf C, Geidel L, Zerbe N, Bialke M, Stahl D, Blumentritt A, Bahls T, Hufnagl P, Hoffmann W, et al.
  Assessment of scalability and performance of the record linkage tool E-PIX® in managing multi-million patients in research projects at a large university hospital in Germany (Originalartikel)
  Journal of Translational Medicine. 2020. DOI:10.1186/s12967-020-02257-4
  https://dx.doi.org/10.1186/s12967-020-02257-4
- Bialke M*, Penndorf P, Wegner T, Bahls T, Havemann C, Piegsa J, et al.
  A workflow-driven approach to integrate generic software modules in a Trusted Third Party (Originalartikel)
  Journal of Translational Medicine. 2015; 13(176).
  http://www.translational-medicine.com/content/13/1/176
- Bialke M*, Bahls T, Havemann C, Piegsa J, Weitmann K, Wegner T, et al.
  MOSAIC. A modular approach to data management in epidemiological studies. (Originalartikel)
  METHODS OF INFORMATION IN MEDICINE. 2015; 54(4):364-371.
  http://dx.doi.org/10.3414/ME14-01-0133
- Bialke M, Langner D, Bahls T, Geidel L, Piegsa J, Havemann C, Hoffmann W.
  “Who am I? And if so, how many?” – The E-PIX as innovative system to manage person identities. (Poster)
  2nd Research Data Management Workshop; 2014 Nov 27; Köln.

11 Weiterführende Informationen

Überblicksseite E-PIX

ths-greifswald.de/epix

Produktbroschüre E-PIX

https://www.ths-greifswald.de/epix/produktbrief

E-PIX Service Spezifikation

ths-greifswald.de/epix/doc

E-PIX Demo

ths-greifswald.de/epix/demo

Git-Repository

https://github.com/mosaic-hgw/E-PIX

Offizielle E-PIX Docker Image (Standard-Version, DB+App)

https://hub.docker.com/r/tmfev/epix/

```
docker pull tmfev/epix
```

Docker Installation

https://docs.docker.com/install/

Docker Compose Installation

https://docs.docker.com/compose/install/

Docker Cheat Sheet

https://www.docker.com/sites/default/files/Docker_CheatSheet_08.09.2016_0.pdf

Docker und Docker Compose Cheat Sheet

https://dev-eole.ac-dijon.fr/doc/cheatsheets/docker.html