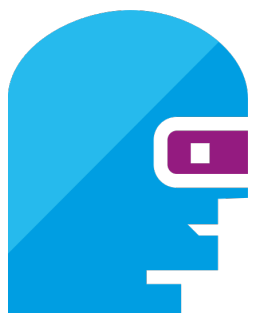


2024.2



gPAS

Digitales Pseudonym- Management



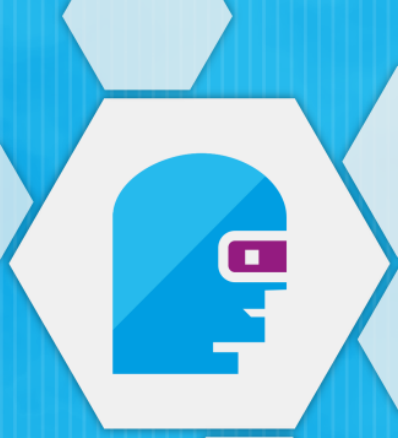
HERAUSGEBER: Unabhängige Treuhandstelle der Universitätsmedizin Greifswald

AUTOR: Christopher Hampf

WEBSEITE: <https://www.ths-greifswald.de>

KONTAKT: kontakt-ths@uni-greifswald.de

VERÖFFENTLICHUNG: 16. Oktober 2024



Inhaltsverzeichnis

I	Einführung	
1	Grundlagen	10
1.1	Domänen	11
1.2	Eltern- und Kind-Domänen	11
1.3	Originalwerte	12
1.4	Pseudonyme	12
1.5	Anonymisierung im gPAS	13
1.6	Pseudonym-Kardinalitäten	13
2	Betrieb	14
2.1	Funktionalitäten	14
2.1.1	Was leistet der Dienst	14
2.1.2	Was leistet der Dienst nicht	14
2.2	Installation	15
2.2.1	Systemanforderungen	15
2.2.2	Download und Starten des Dienstes	16
2.2.3	Starten unter Linux	17
2.2.4	Starten unter Windows	18
2.3	Update	19
2.4	Trennung von Applikations- und Datenbankserver	19

II**Konfiguration**

3	Weboberfläche	22
3.1	Anlegen einer Domäne	22
4	SOAP-Schnittstelle	27
4.1	Anlegen einer Domäne	28
4.1.1	Prüfziffernalgorithmen	31
4.1.2	Alphabete	33
4.1.3	Validierung von Pseudonymen	34
5	Anwendungsbeispiele	35
5.1	genomDE – Phase 0	35
5.2	Verwaltung von Bioproben	37

III**Bedienung**

6	Weboberfläche	42
6.1	Generieren von Pseudonymen	42
6.2	Originalwerte und Pseudonyme suchen	44
6.3	Depseudonymisierung (Suche von Originalwerten)	45
6.4	Technische Anonymisierung	45
6.5	Löschen von Pseudonymen	47
6.6	Integration von Alt-Pseudonymen	48
6.7	Anzeige von Pseudonym-Hierarchien	48
6.8	Listenverarbeitung	49
6.9	Dashboard für Statistiken	51
6.10	Domäne bearbeiten oder löschen	52
6.11	Pseudonyme importieren	53
6.12	Pseudonyme exportieren	54
7	SOAP-Schnittstelle	55
7.1	Pseudonyme anlegen	55
7.1.1	Single-Pseudonym-Domäne	55
7.1.2	Multi-Pseudonym-Domäne	57

8	Logging	62
9	Benachrichtigungen	63
10	FHIR-Unterstützung	64
11	Authentifizierung & Autorisierung	66
11.1	Global	66
11.1.1	Übersicht Nutzerrollen und Rechte	66
11.1.2	Übersicht Nutzerrollen und Rechte	67
11.1.3	Verwendung von gRAS	67
11.2	Domänen-spezifische Rollen mit OpenID-Connect	67
12	Empfehlungen zur Absicherung	69
13	Optimierungen	70
13.1	Speicher für MySQL erhöhen	70
13.2	Batch-Writing	70
13.2.1	Lange Zeiten zum Hochfahren des Applikationsservers	71
	Weitere Literatur	72
	Publikationen	72
	Glossar	73
	Abkürzungsverzeichnis	75



Abbildungsverzeichnis

1.1	Beispiel eines Domänen-Baums	12
2.1	gPas Docker-Architektur	16
3.1	Oberfläche zum Anzeigen aller Domänen . Der Baum stellt die hierarchische Struktur der Domänen dar. Mit einem Rechtsklick auf eine Domäne öffnet sich das Kontextmenü, welches weitere Optionen enthält.	23
6.1	Oberfläche zum Anlegen eines neuen	43
6.2	Kontextmenü zum Erzeugen von Pseudonymen derselben Stufe (<i>Pseudonymisiere Originalwert</i>) und einer höheren Stufe in einer Kind-Domäne (<i>Pseudonymisiere Pseudonym</i>).	44
6.3	Exemplarische Struktur bei mehreren Pseudonymen und Stufen für einen Studienteilnehmer.	44
6.4	Oberfläche zum Suchen von Originalwerten oder Pseudonymen .	45
6.5	Anonymisierung in der Baumstruktur durch Auftrennen der Verbindung (Schere).	46
6.6	Der unumkehrbare Vorgang der Anonymisierung muss nochmals bestätigt werden.	46
6.7	Anonymisierter Eintrag.	47
6.8	Oberfläche beim Anzeigen der Pseudonymhierarchie für ein selektiertes Pseudonym (blau dargestellt).	49
6.9	Oberfläche zum Verarbeiten von Listen.	49
6.10	Wählen der Verarbeitungsoperation. Hier am Beispiel von Pseudonymisieren .	50
6.11	Oberfläche zum Einsehen von der Anzahl von Pseudonymen , Anonymen und Domänen . Die Daten sind in Diagrammen aufgeführt.	51
6.12	Kontextmenü mit den Schaltflächen zum Anzeigen der Domänendetails, zum Bearbeiten der Domäne und zum Löschen der Domäne . Hierüber können weitere Domänen erzeugt werden (siehe Abschnitt 3.1).	52

6.13	Oberfläche zum Importieren von Pseudonympaaren.	53
6.14	Oberfläche zum Exportieren beliebiger Domänen.	54



Tabellenverzeichnis

3.1	Bereitgestellte Alphabete im gPAS	26
3.2	Prüfziffern-Generatoren und dessen Bedingungen.	26
4.1	Elemente einer SOAP-Anfrage zum Anlegen einer neuen Domäne	29
4.2	Unterstützte Prüfzifferalgorithmen mit den jeweiligen Anforderungen an das Alphabet.	32
4.3	Vordefinierte Alphabete mit den enthaltenen Zeichen und der daraus resultierenden Anzahl von Zeichen innerhalb eines Alphabets.	33
4.4	Alphabete mit den jeweils kompatiblen Prüfzifferalgorithmen.	33
6.1	Mögliche Verarbeitungsoperationen.	50
11.1	Nutzer-Zugriffsrechte in der Weboberfläche.	67

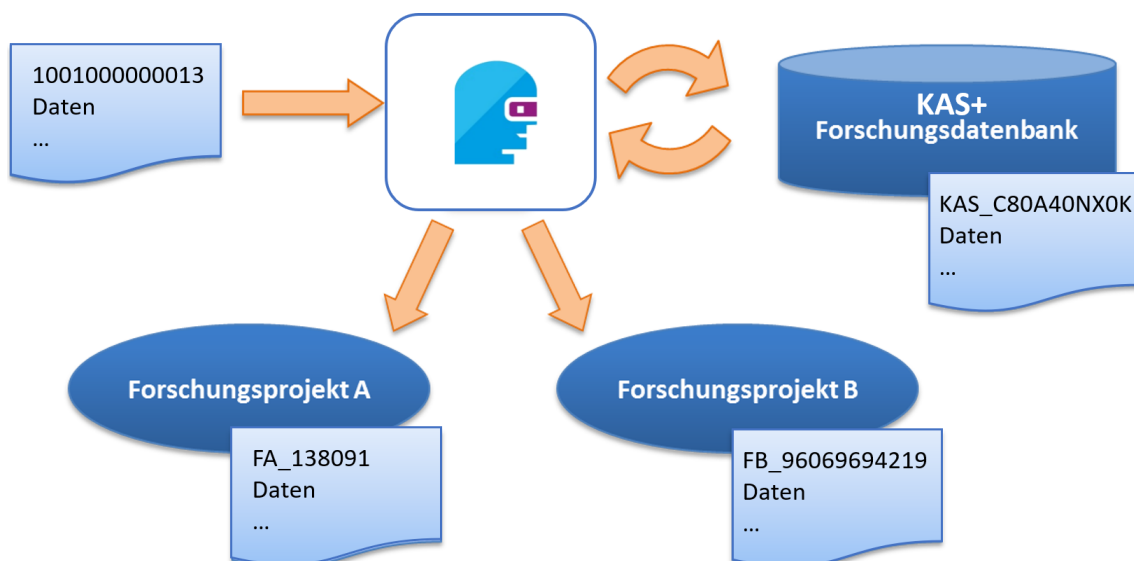


Einführung

1	Grundlagen	10
1.1	Domänen	11
1.2	Eltern- und Kind-Domänen	11
1.3	Originalwerte	12
1.4	Pseudonyme	12
1.5	Anonymisierung im gPAS	13
1.6	Pseudonym-Kardinalitäten	13
2	Betrieb	14
2.1	Funktionalitäten	14
2.2	Installation	15
2.3	Update	19
2.4	Trennung von Applikations- und Datenbankserver ..	19

1. Grundlagen

Die Durchführung klinisch-epidemiologischer Studien, aber auch der Aufbau von Registern und Kohorten, erfordern eine datenschutzkonforme Datenverarbeitung. Gemäß Art. 32 Abs. 1a [Datenschutz-Grundverordnung \(DSGVO\)](#) unterstützt die Verwendung von [Pseudonymen](#) dabei, ein angemessenes Schutzniveau der Datenverarbeitung zu gewährleisten. Am Institut für Community Medicine der Universitätsmedizin Greifswald wurde hierfür der [Generic Pseudonym Administration Service \(gPAS\)](#) entwickelt. Das Web-basierte Werkzeug [gPAS](#) dient der Generierung und Verwaltung von [Pseudonymen](#). Das Domänenkonzept sowie die freie Definition von Alphabeten als auch Generatoralgorithmen erlauben unterschiedliche [Pseudonyme](#) je Datenquelle, Anwendungskontext (z.B. Erhebung oder Herausgabe) und Standort zu generieren. Der [gPAS](#) ist als Open Source Software lizenziert (AGPLv3) und kostenfrei für kommerzielle und nicht-kommerzielle Zwecke einsetzbar.



1.1 Domänen

Domänen geben die Kontexte an, die **pseudonymisiert** werden sollen. Dies können z.B. ganze Projekte, Teile von Projekten oder einzelne Systeme sein. Oft werden mehrere **Domänen** zu einer Pseudonymhierarchie zusammengeschaltet, um mehrere Kontexte innerhalb eines Projektes oder verschiedene Arten von **Pseudonymen** für verschieden Bereiche abbilden zu können (siehe auch Abschnitt 1.2).

Jede **Domäne** hat eine Konfiguration hinterlegt, welche die Art und Weise der Pseudonymgenerierung definiert. Dies umfasst die Länge eines **Pseudonyms**, ob Präfixe oder Suffixe vorhanden sind und welches Alphabet und welche Prüfziffern verwendet werden sollen.

Die **Pseudonymisierung** in einer **Domäne** kann als Liste verstanden werden, die **Originalwerte** mit **Pseudonymen** verknüpft.

1.2 Eltern- und Kind-Domänen

Über mehrere **Domänen** können Pseudonymhierarchien abgebildet werden. Diese beginnt immer mit der Wurzel-**Domäne** und stellt damit immer die niedrigste Stufe der **Pseudonyme** dar. Als **Originalwert** dient meist ein **Pseudonym** erster Stufe, also jenes **Pseudonym** welches direkt mit den **Identifizierende Daten (IDAT)** einer Person zugeordnet ist. Hierbei kommt oft ein **Master Patient Index (MPI)** oder **Personenidentifikator (PID)** zum Einsatz. In der Wurzel-**Domäne** findet die erste **Pseudonymisierung** statt. Jede **Domäne** kann außerdem beliebig viele **Kind-Domänen** aufweisen. Diese stellen jeweils höhere Pseudonymisierungsstufen dar und pseudonymisieren das jeweilige **Pseudonym** der **Eltern-Domäne**. Eine **Kind-Domäne** kann dabei einen spezifischen Kontext in einem Projekt abbilden. Werden für eine Person z.B. für Bilddaten, Datenherausgaben, verschiedene Studien mehrere **Pseudonyme** benötigt, so kann für jeden dieser Kontexte eine **Domäne** angelegt werden. Die Verknüpfung mit einer **Eltern-Domäne** ermöglicht es dabei, dass die **Pseudonyme** später wieder zu einer niedrigeren Stufe (bis hin zum **Pseudonym** erster Stufe) aufgelöst werden können¹. In Abbildung 1.1 ist eine exemplarische Darstellung von einem **Domänen-Baum** mit verschiedenen beispielhaften Kontexten.

¹ Diese Auflösung kann mittels einer technischen Anonymisierung unterbunden werden (siehe Abschnitt 1.5).

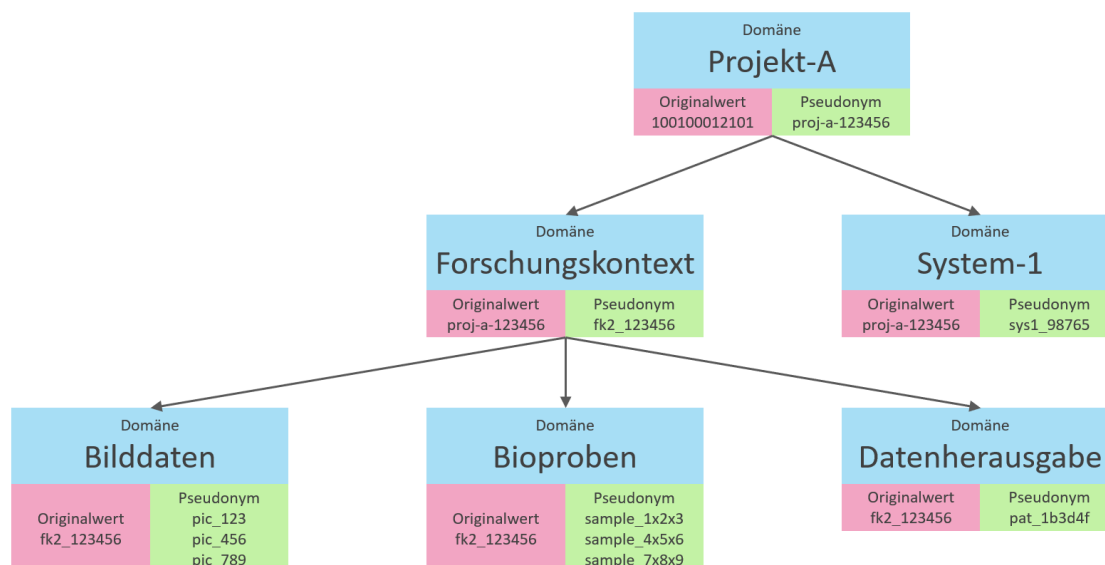


Abbildung 1.1: Exemplarischer Aufbau einer Pseudonymhierarchie über verknüpfte Eltern-Domänen und Kind-Domänen.

1.3 Originalwerte

Der **Originalwert** ist jener Wert, der **pseudonymisiert** werden soll. Dies geschieht so, dass hierfür ein **Pseudonym** generiert und zugeordnet wird. Um höhere Stufen der Pseudonymisierung zu erreichen, wird das **Pseudonym** als **Originalwert** in eine **Kind-Domäne** eingetragen, in der wiederum ein **Pseudonym** für diesen **Originalwert** erzeugt wird. Der **Originalwert** stellt somit immer ein **Pseudonym** einer niedrigeren Stufe dar.

1.4 Pseudonyme

Ein **Pseudonym** ist eine definiert lange Zeichenkette, basierend auf zufällig gewählten Zeichen eines bestimmten Alphabets. Der **gPAS** ordnet ein oder mehrere **Pseudonyme** einem **Originalwert** zu. Der **gPAS** nutzt demnach keine Hashfunktionen, um einen **Originalwert** in ein **Pseudonym** zu überführen. Bei einer Hashfunktion zu einem Eingangswert nach definierten mathematischen Regeln ein Hashwert errechnet, welcher als **Pseudonym** verwendet werden könnte. Durch die zufällige Erzeugung von **Pseudonyme** im **gPAS**, existiert keine Rechenregel, welche den **Originalwert** in das **Pseudonym** überführt. Dadurch sind keinerlei Informationen in den **Pseudonyme** kodiert, welche kompromittiert werden könnten. Demnach können auch keine sensiblen Informationen herausgerechnet werden. Außerdem kann der **gPAS** so auch eine **Anonymisierung** umsetzen (siehe Abschnitt 1.5) indem die Zuordnungen von **Originalwert** und **Pseudonym** entfernt werden.

1.5 Anonymisierung im gPAS

Anonymisierung beschreibt einen Vorgang, der die Möglichkeit auf Rückschlüsse auf eine Person verhindert. Im **gPAS** werden keine **IDAT** gespeichert, sodass keine **Anonymisierung** basierend auf diesen Daten stattfinden kann. Es kann aber erwünscht sein, dass eine Auflösung von **Pseudonymen** nicht mehr möglich ist. Diese Form der **Anonymisierung** wird im **gPAS** als technische **Anonymisierung** bezeichnet. Hierbei wird der Bezug von **Originalwert** und **Pseudonym** entfernt, sodass eine Aufklösung später nicht mehr möglich ist. Dies erlaubt, dass Systeme oder Kontexte, weiterhin mit den **Pseudonymen** arbeiten können, ohne dass diese entfernen oder ersetzt werden müssen.

1.6 Pseudonym-Kardinalitäten

Wenn **Pseudonyme** für Personen generiert werden, dann besteht meist eine 1 zu 1 Beziehung zwischen den verschiedenen Kontexten. D.h. eine Person hat ein **Pseudonym** in einem Projekt und auch in höheren Pseudonymisierungsstufen hat diese Person pro **Domäne** nur ein **Pseudonym**. Daraus ergibt sich, dass es in jeder **Domäne** zu jedem **Originalwert** auch immer nur ein **Pseudonym** gibt (**Single-Pseudonym-Domäne**). Dieses Verhalten ist meistens gewünscht und der **gPAS** stellt sicher, dass diese Struktur eingehalten wird. In bestimmten Fällen kann es jedoch erforderlich sein, dass es zu einer Person mehrere **Pseudonyme** geben soll. Dies kann z.B. bei der Verwaltung von Bioproben oder Bilddaten erforderlich sein, wenn innerhalb eines Kontextes (**Domäne**) zu einer Person mehrere **Pseudonyme** erzeugt werden sollen. Hierzu unterstützt der **gPAS** sogenannte **Multi-Pseudonym-Domänen**. Diese erlauben es pro **Originalwert** mehrere **Pseudonyme** zu erzeugen. Im **gPAS** taucht dann in diesen **Domänen** derselbe **Originalwert** mehrfach auf und ist mit unterschiedlichen **Pseudonymen** verknüpft. In diesem Fall besteht dann eine 1 zu n Beziehung, wobei n beliebig groß sein kann. Die Auflösung von **Pseudonymen** erfolgt auf die gleiche Weise wie bei 1 zu 1 Beziehungen. Der umgekehrte Fall muss jedoch berücksichtigen, dass es mehrere **Pseudonyme** geben kann und diese ggf. in höheren Pseudonymstufen nicht als **Originalwert** auftauchen.



2. Betrieb

2.1 Funktionalitäten

2.1.1 Was leistet der Dienst

- Generierung von **Pseudonymen**
- Zuordnung von **Pseudonymen** zu beliebigen **Originalwerten**
- Technische **Anonymisierung** durch Löschung von Zuordnungen zwischen **Pseudonym** und **Originalwert**
- Konfiguration von Pseudonym-Parametern: Prüzifferalgorithmus, Länge, Alphabet
- Verwaltung von Pseudonym-**Domänen** und Zweitpseudonymen
- Validierung von **Pseudonymen**
- **Depseudonymisierung**
- Darstellung von Pseudonymbäumen (-hierarchien)
- Import und Export vorhandener **Pseudonyme**
- Löschen von temporären **Pseudonymen** (dabei werden sowohl **Pseudonym** als auch zugeordneter **Originalwert** gelöscht)
- Listenverarbeitung
- Hohe Performance durch Caching
- Unterstützung für KeyCloak-Authentifizierung und Autorisierung

2.1.2 Was leistet der Dienst nicht

- Maskierung: der erforderliche Schritt zur Trennung von personenidentifizierenden und medizinischen Daten ist nicht Teil des **gPAS**-Systems und muss vom nutzenden Projekt geleistet werden
- Extraktion und Schwärzung identifizierender Merkmale in Dokumenten oder

Datensätzen

- Record Linkage/Identitäts-Matching¹

2.2 Installation

Der **gPAS** wird als standardmäßig als Docker-Container bereitgestellt. Die Verwendung von Docker wird empfohlen. Alternativ dazu kann der **gPAS** als Servlet im Applikationsserver WildFly betrieben werden. Die Voraussetzungen hierfür sind im folgenden Abschnitt 5 unter *Software: Anwendungs- und Datenbankserver* aufgeführt.

2.2.1 Systemanforderungen

Technisch/Infrastruktur (mit und ohne Docker)

- Installierte aktuelle Version von Docker² und Docker-Compose³ (mittlerweile in Docker enthalten)
- Administrative Rechte
- Keine Nutzungsbeschränkungen auf die bereitgestellten Service- und Client-URLs
- Windows 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)

Software: Anwendungs- und Datenbankserver (ohne Docker)⁴

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

Personell

- Verantwortlicher mit grundlegenden IT-Kenntnissen zur Administration des Servers und zur Einrichtung des **gPAS**-Dienstes (zuzüglich der Wartung und regelmäßiger Sicherungen der **gPAS**-Datenbank)
- Verantwortlicher zur Administration und Pflege der **gPAS**-Inhalte

¹ Dies leistet der E-PIX: <http://ths-greifswald.de/e-pix>

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

³ Weitere Informationen unter <https://docs.docker.com/compose/install/>

⁴ Beim Betrieb unter Windows ist zu beachten, dass bei der Verwendung von Volumes und parallel betriebenen VPN-Clients Probleme auftreten können.

2.2.2 Download und Starten des Dienstes

Hinweis: Die hier beschriebene Installation erfolgt standardmäßig mit *Docker-Compose*. Wenn der **gPAS** davon abweichend ohne Docker betrieben werden soll, kann eine Performancesteigerung erreicht werden, indem die Hinweise in Kapitel 13 berücksichtigt werden. In der ausgelieferten Docker-Variante sind diese bereits berücksichtigt und es sind keine weiteren Anpassungen erforderlich.

Um den **gPAS** als Docker-Container zu starten, werden die Programme *Docker* und *Docker Compose* (in der aktuellen Version ist dies bereits in *Docker* enthalten) benötigt. Beide Programme müssen hierfür installiert sein. Da zwischen beiden Programmen Inkompatibilitäten auftreten können, wird empfohlen die jeweils aktuellsten Versionen zu installieren.

Der **gPAS** benötigt zur Ausführung mehrere Container (vgl. Abbildung 2.1). Damit diese nicht einzeln gestartet werden müssen und entsprechend zusammengesaltet werden müssen, wird der Dienst mit *Docker-Compose* gestartet. Die entsprechenden Ressourcen können von der THS-Webseite heruntergeladen werden⁵.

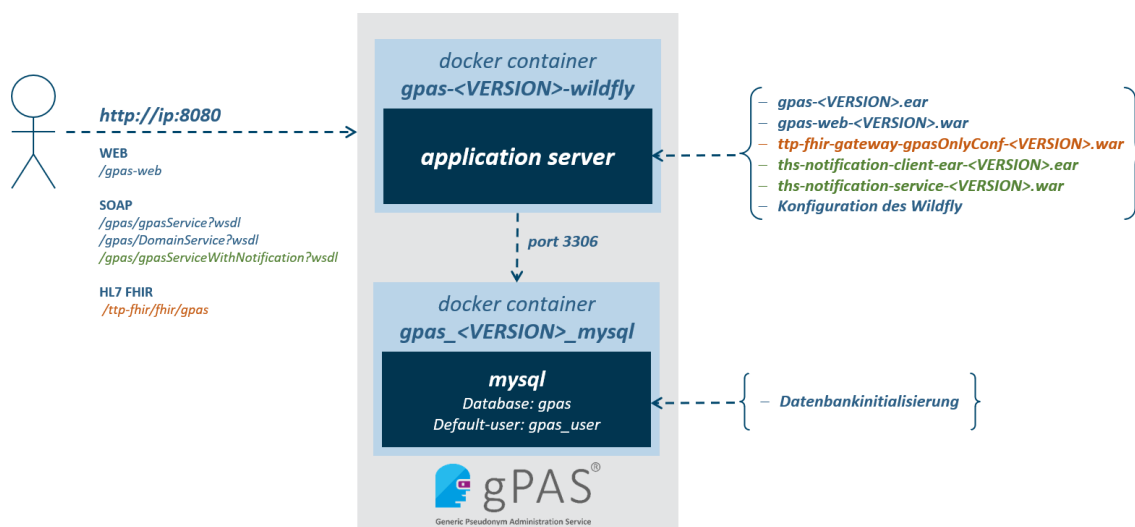


Abbildung 2.1: Architektur des **gPAS** mit Docker.

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“

⁵ <https://www.ths-greifswald.de/forscher/gpas/> bzw. <https://www.ths-greifswald.de/forscher/gpas/#download>

erfolgt über den Web-Browser. Die Inhalte werden über den Port 8080 (gPAS) für den Anwender bereitgestellt.

Hinweis: Weitere Details zur Nutzung von Docker-Compose und gPAS sind der beigelegten Beschreibung `docker-compose/README_gPAS.md` zu entnehmen.

Hinweis: Für einen **Produktivbetrieb** sollte die `docker-compose.yml` angepasst werden. Hierzu sollte der Speicherpfad des MySQL-Volumens 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 gPAS 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_gPAS.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/gpas/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/gpas/#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 Schreibrechten

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

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

Wechseln in das **gPAS**-Verzeichnis für die Standard-Version

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

Starten des **gPAS** mithilfe von Docker Compose

```
1 sudo docker-compose up
```

Damit werden die benötigten Komponenten heruntergeladen⁶ und die Konfiguration von MySQL und WildFly gestartet. Danach wird die aktuelle Version des **gPAS** bereitgestellt. Der Installationsvorgang kann in Abhängigkeit der vorhandenen Internetverbindung etwa 5 Minuten dauern. Der erfolgreiche Start des Dienstes wird mit der folgenden Ausgabe abgeschlossen.

```
1 Wildfly [Version] [...] started in ...
```

2.2.4 Starten unter Windows

Zur Installation und Starten des **gPAS** ist ein Benutzeraccount mit Adminrechten erforderlich.

Entpacken Sie das Archiv an der gewünschten Stelle. Danach kann der **gPAS** 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 **gPAS** 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 (`#`).

Anhand folgender Befehle können Sie nun den **gPAS** über Docker-Compose starten:

```
1 sudo docker compose up
```

⁶ Sollte Ihre Maschine keinen Zugang zum Internet haben, können die benötigten Images (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/ und <https://docs.docker.com/engine/reference/commandline/load/>).

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

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

2.3 Update

Hinweis: Manuelle Anpassungen bei einem Versionswechsel sind nur erforderlich, wenn der **gPAS** in einem Applikationsserver betrieben wird und nicht die bereitgestellten Docker benutzt werden.

Hinweis: Bitte beachten Sie die Update-Hinweise im Abschnitt 2.2

Beim Versionswechsel auf **gPAS** 1.11.0 kann es beim Start und zyklisch im Betrieb zu Warnungen (Failed to reinstate timer 'gpas.psn-ejb.StatisticManagerBean') im Serverlog kommen. Grund hier ist eine verschobene Timer-Funktion. Dies kann behoben werden, indem im Verzeichnis vom WildFly die unter [WildFly-Verzeichnis]/standalone/data/timer-service-data enthaltenen Dateien entfernt werden. Danach kann der Dienst neu gestartet werden. Die neuen Dateien werden danach automatisch erzeugt und es kommt diesbezüglich zu keinen Warnungen mehr.

2.4 Trennung von Applikations- und Datenbankserver

Der **gPAS** 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/ttp_gpas.env` definiert. In der Datei müssen folgende Variablen auskommentiert werden (`#` entfernen) und entsprechend angepasst werden: `TTP_GPAS_DB_HOST`, `TTP_GPAS_DB_PORT`, `TTP_GPAS_DB_NAME`, `TTP_GPAS_DB_USER` und `TTP_GPAS_DB_PASS`.



Konfiguration

3	Weboberfläche	22
3.1	Anlegen einer Domäne	22
4	SOAP-Schnittstelle	27
4.1	Anlegen einer Domäne	28
5	Anwendungsbeispiele	35
5.1	genomDE – Phase 0	35
5.2	Verwaltung von Bioproben	37



3. Weboberfläche

3.1 Anlegen einer Domäne

Um **Pseudonyme** erstellen zu können, muss vorab eine **Domäne** angelegt werden.

Dabei gilt:

- ein **Pseudonym** ist innerhalb einer **Single-Pseudonym-Domäne** eindeutig
- es kann in einer **Multi-Pseudonym-Domäne** mehrere **Pseudonyme** pro **Originalwert** geben
- eine **Domäne** kann für ein Projekt oder ein eingebettetes Studienvorhaben stehen, aber auch zur Verwaltung von Zweitpseudonymen angelegt werden
- für jede **Domäne** lassen sich eigene **Pseudonym-Parameter** festlegen

Unter dem Menüpunkt *domain* werden alle bereits angelegten **Domänen** als Baumstruktur dargestellt (siehe Abbildung 3.1). Mit einem Rechtsklick auf einen Eintrag werden weitere Optionen zur jeweiligen **Domäne** angeboten. So kann beispielsweise direkt die Seite *Suchen / Generieren* aufgerufen werden, wobei die jeweilige **Domäne** vorausgewählt ist. Um eine neue **Domäne** zu erstellen, kann entweder die Schaltfläche **+ Erstellen** angewählt werden oder über das Kontextmenü je nach angewählter **Domäne** eine **Geschwister-Domäne** oder eine **Kind-Domäne** angelegt werden. Daraufhin öffnet sich ein Fenster mit den auszufüllenden Feldern. Bei der Wahl einer **Geschwister-Domäne** wird die **Eltern-Domäne** der ausgewählten **Domäne** übernommen. Bei einer **Kind-Domäne** wird die angewählte **Domäne** als **Eltern-Domäne** gesetzt.

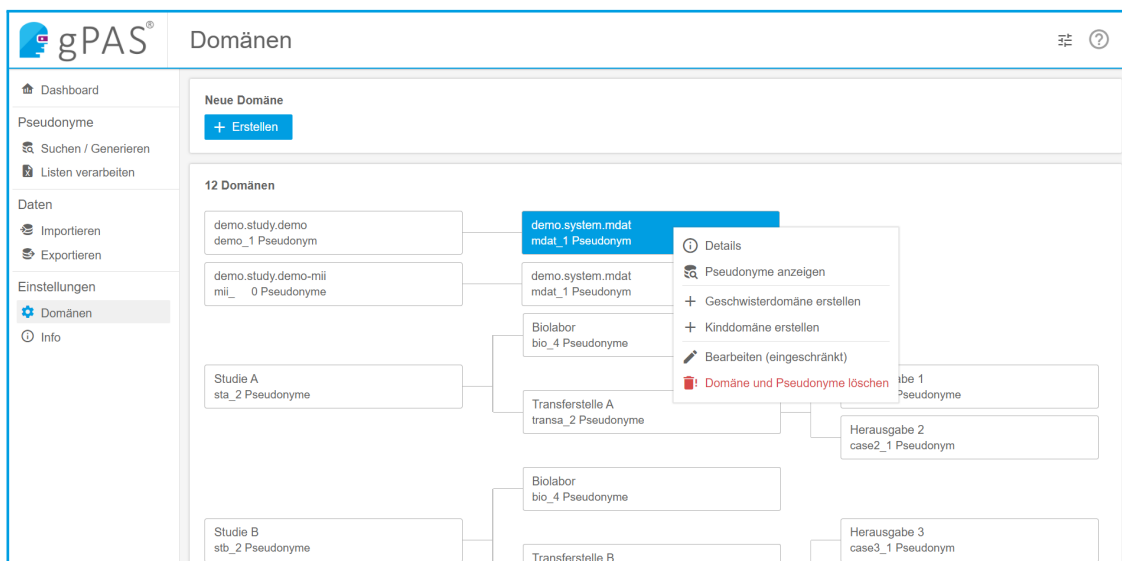


Abbildung 3.1: Oberfläche zum Anzeigen aller **Domänen**. Der Baum stellt die hierarchische Struktur der **Domänen** dar. Mit einem Rechtsklick auf eine **Domäne** öffnet sich das Kontextmenü, welches weitere Optionen enthält.

Beim Anlegen einer neuen **Domäne** ist der eindeutige Name als *Bezeichnung* festzulegen. Zudem kann ein *Schlüssel* vergeben werden, der intern genutzt wird. Wird kein Schlüssel angegeben, erzeugt der **gPAS** diesen automatisch. Dabei ist zu beachten, dass eine **Domäne** über diesen Schlüssel über die SOAP-Schnittstelle angesprochen wird. Wenn der Name der **Domäne** nachträglich geändert wird, betrifft dies nur die Darstellung in der Oberfläche. Eine Änderung des Schlüssels ist nachträglich nicht mehr möglich. Optional können eine oder mehrere **Eltern-Domänen** (um **Pseudonym**-Hierarchien zu realisieren) angegeben werden. Eine **Domäne** mit mehreren **Eltern-Domänen** wird entsprechend oft in den jeweiligen **Domänen**-Bäumen dargestellt. Zusätzlich kann eine Beschreibung ergänzt werden.

Erweiterte Einstellungen

Standardmäßig wird eine **Single-Pseudonym-Domäne** angelegt. Mit der Option *Erlaube mehrere Pseudonyme für denselben Originalwert* kann stattdessen eine **Multi-Pseudonym-Domäne** erzeugt werden. Dies ist dann erforderlich, wenn mehrere **Pseudonyme** pro **Originalwert** erzeugt werden sollen. Dies kann z.B. bei Biomaterialproben erforderlich sein, wenn pro Person mehrere pseudonymisierte Proben verwaltet werden. Dabei ist zu beachten, dass ggf. die Abfragezeiten höher sind, als bei **Single-Pseudonym-Domänen**. Je nach Projekt muss entschieden werden, welche Anforderungen die **Domäne** erfüllen muss. Diese Option kann später nicht mehr verändert werden. Es ist jedoch möglich, eine **Single-Pseudonym-Domäne** später zu exportieren und diese in eine **Multi-Pseudonym-Domäne** zu

importieren. Auf dieser Basis können dann auch mehrere **Pseudonyme** pro **Originalwert** erzeugt werden.

Neue Domäne
✕

Bezeichnung *	<input type="text" value="Studie C"/>	Pseudonyme	
Schlüssel	<input type="text" value="study_c"/>	Länge	<input type="text" value="10"/>
Elterndomänen	<input type="text" value="Domänen filtern"/>	Präfix	<input type="text" value="PSN_"/>
Beschreibung	<input type="text" value="Beschreibung der Domäne zu Studie C."/>	Suffix	<input type="text"/>
	219 Zeichen verbleibend	Alphabet	<input type="text" value="Ziffern und Buchstaben (ohn..."/> <small>Mögliche Pseudonyme: 1.125.899.906.842.624</small>
Erweiterte Einstellungen		Prüfziffern-Generator	<input type="text" value="HammingCode"/>
Erlaube mehrere Pseudonyme für denselben Originalwert (langsamer)	<input type="checkbox"/>	Präfix in Prüfziffer einbeziehen	<input type="checkbox"/>
Erlaube löschen von Pseudonymen	<input checked="" type="checkbox"/>	Suffix in Prüfziffer einbeziehen	<input type="checkbox"/>
Sende Benachrichtigungen durch die Weboberfläche	<input type="checkbox"/>		
Cache	<input type="text" value="Automatisch"/>		
Originalwert in Elterndomänen validieren	<input type="text" value="Nicht validieren"/>		
<input checked="" type="button" value="Erstellen"/> <input type="button" value="Abbrechen"/>			

Das Löschen einer **Domäne** ist nur möglich, sofern keine **Pseudonyme** für diese **Domäne** hinterlegt sind. Das Löschen eines **Pseudonyms** ist möglich, wenn dies gestattet werden soll (Option: *Erlaube löschen*).

Darüber hinaus kann der **gPAS** Benachrichtigungen versenden, um andere Systeme über Änderungen zu informieren. Weitere Informationen können dem Kapitel 9 entnommen werden.

Die Geschwindigkeit bei großen Beständen kann erheblich verbessert werden, wenn das integrierte Caching verwendet wird. Standardmäßig aktiviert der **gPAS** dies selbst (Option: *automatisch*). Es kann jedoch explizit für eine **Domäne** aktiviert oder deaktiviert werden.

Grundsätzlich erzeugt der **gPAS** selbstständig korrekte **Pseudonyme**, welche in höheren Stufen als **Originalwerte** fungieren. Es können jedoch auch **Pseudonyme** aus z.B. Altbeständen importiert werden, die in die **Pseudonym**-Hierarchie aufgenommen werden sollen. Sollen diese der definierten Struktur entsprechen, kann die Validierung in verschiedenen Optionen durchgeführt werden: **Originalwert** muss valide sein und in der Elterndomäne vorhanden sein (Option: *Muss valides*

Pseudonym sein), **Pseudonym** muss in der Elterndomäne vorhanden sein (Option: *Muss vorhandenes Pseudonym sein*) oder **Pseudonym** muss in der Elterndomäne vorhanden sein, andernfalls wird der Eintrag gelöscht (Option: *Muss vorhandenes Pseudonym sein und kaskadiere bei Löschung in Elterndomäne*). Dabei wird die Option *Erlaube löschen* ignoriert.

Pseudonyme

Mit der *Länge* wird die Anzahl der zufälligen Zeichen im **Pseudonym** bestimmt. Die tatsächliche Länge des **Pseudonyms** kann davon abweichen, wenn zusätzlich ein *Präfix*, *Suffix* oder Prüfziffern verwendet werden¹.

Das *Präfix* und *Suffix* definieren einen konstanten Teil, der jedem **Pseudonym** voran oder nach gestellt wird. Wird beispielsweise das *Präfix* `psn_` gewählt, so könnte ein resultierendes **Pseudonym** so aussehen: `psn_123456`. Zu beachten ist, dass wenn im späteren die Option *Präfix/Suffix in Prüfziffer einbeziehen* gewählt wird, jeweils nur Zeichen verwendet werden dürfen, die Teil des gewählten Alphabets sind. Ansonsten können beliebige Zeichen verwendet werden.

Mit dem *Alphabet* kann der Raum der verwendeten Zeichen des **Pseudonyms** ausgewählt werden. In Tabelle 3.1 sind die Alphabete angegeben, welche direkt ausgeliefert werden. Alternativ dazu, können eigene Alphabete definiert werden. Hierzu wird der Eintrag *Benutzerdefiniert* gewählt. Danach können die gewünschten Zeichen eingetragen werden. Hierbei besteht die Limitation, dass keine Leerzeichen oder Kommas erlaubt sind. Die Zeichen werden hintereinander ohne Trennzeichen angegeben. Es besteht die Möglichkeit, dass **Pseudonyme** mit einem Separator bzw. Trennzeichen versehen werden. Dieses Trennzeichen wird dann nach einer beliebigen Anzahl von Zeichen im **Pseudonym** eingetragen. Die Länge verändert sich dabei nicht, da das Trennzeichen Teil des **Pseudonyms** ist. Zu beachten ist, dass sich dadurch die Anzahl der möglichen Kombinationen reduziert. Ein Trennzeichen kann definiert werden, indem es an das benutzerdefinierte Alphabet angefügt wird und die Option *Trennzeichen nach jedem n-ten Zeichen setzen* größer als 0 gesetzt wird. Bei 0 wird kein Trennzeichen verwendet und alle Zeichen des Alphabets werden verwendet. Andernfalls wird nach jedem n-ten Zeichen das definierte Trennzeichen eingefügt. Bei $n = 4$ mit einem numerischen Alphabet und dem Trennzeichen “-” und einer Pseudonymlänge von 14, könnte ein resultierendes **Pseudonym** so aussehen: `1234-4567-7890`. Das **Pseudonym** besteht demnach aus 12 zufällig gewählten Zeichen des Alphabets und zwei

¹ Wird ein benutzerdefiniertes Alphabet mit Trennzeichen verwendet, wirkt sich dies zwar nicht auf die Länge aus, reduziert jedoch die Anzahl der Stellen mit zufällig gewählten Zeichen.

Trennzeichen.

Tabelle 3.1: Bereitgestellte Alphabete im gPAS.

Alphabet	Enthaltene Zeichen
Hex-Zeichen	16 Zeichen: 0-9, A-F
Ziffern 0-9	10 Zeichen: 0-9
Ziffern 1-9	9 Zeichen: 1-9
Ziffern 0-9, X	11 Zeichen: 0-9, X
Ziffern und Buchstaben ohne B, I, O, S	32 Zeichen: 0-9, A-Z (ohne B, I, O, S – wegen der Ähnlichkeit zu 8, 1, 0, 5)
Ziffern und Buchstaben ohne B, I, O, S, V	Wie Symbol32, nur ohne V – wegen der Ähnlichkeit zu U

Mit der Wahl eines *Prüfziffern-Generators* können zusätzliche Prüfziffern an die **Pseudonyme** angefügt werden. Diese erlauben es, Fehler im Pseudonym zu erkennen. Dies ist vor allem dann sinnvoll, wenn die Prozesse **Pseudonyme** händische Eingaben vorsehen. Bei der Wahl des Prüfziffern-Generators muss die Kompatibilität mit dem verwendeten Alphabet beachtet werden. In der Weboberfläche werden in Abhängigkeit des gewählten Alphabets nur kompatible Prüfziffern-Generatoren angezeigt. Wird ein benutzerdefiniertes Alphabet verwendet, so müssen die jeweiligen Voraussetzungen des Generators beachtet werden. Die verfügbaren Prüfziffern-Generatoren sind in Tabelle 3.2 aufgeführt. Bei Verwendung eines Prüfziffern-Generators können zusätzlich die Präfixe und Suffixe (*Präfix/Suffix in Prüfziffer einbeziehen*) eingebunden werden, andernfalls bezieht sich die Prüfung nur auf den zufälligen Teil des **Pseudonyms**. Sollen keine Prüfziffern enthalten sein, muss die Option *Keine Prüfziffern* ausgewählt werden.

Tabelle 3.2: Prüfziffern-Generatoren und dessen Bedingungen.

Algorithmus	Bedingung
HammingCode	Alphabet-Länge ist eine Primzahlpotenz (Achtung: Derzeitige Implementierung erlaubt nur den Wert 32)
Verhoeff	Alphabet-Länge ist gleich 10
VerhoeffGumm	Alphabet-Länge ist gleich 10
Damm	Alphabet-Länge ist gleich 10
Reed-Solomon-Lagrange	Alphabet-Länge ist eine Primzahl, wobei die maximale Anzahl der Prüfzeichen gleich der Alphabet-Länge ist



4. SOAP-Schnittstelle

Neben der grafischen Oberfläche, steht eine Maschinen-verständliche Schnittstelle zur Verfügung. Diese wird mit per SOAP-Protokoll angesprochen. Die Schnittstelle teilt sich dabei in verschiedene Bereiche auf. So stehen unterschiedliche Schnittstellen für die Domänenverwaltung, die Pseudonymverwaltung mit und ohne Versenden von Benachrichtigungen zu Verfügung. Die jeweiligen Schnittstellendefinitionen können beim laufenden Dienst abgerufen werden. Die jeweiligen Definitionen können unter den folgenden Pfaden abgerufen werden (die URLs müssen entsprechend angepasst werden).

Pseudonymverwaltung:

<http://example.org:8080/gpas/gpasService?wsdl>

Konfiguration und Domänen-Management:

<http://example.org:8080/gpas/DomainService?wsdl>

Versenden von Notifications:

<http://example.org:8080/gpas/gpasServiceWithNotification?wsdl>

Die Entwicklerdokumentation ist unter der folgenden URL zu finden:

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

Für das Anlegen einer [Domäne](#) (Abschnitt 4.1) wird die Management-Schnittstelle zur Konfiguration verwendet.

4.1 Anlegen einer Domäne

Im Folgendem wird das Anlegen einer **Domäne** über die Management-Schnittstelle exemplarisch vorgestellt. Nachfolgend werden die wichtigsten Angaben erläutert. In Listing 4.1 ist eine exemplarisch vollständige Anfrage dargestellt. Die einzelnen Elemente werden in Tabelle 4.1 erläutert. Auf die wichtigsten Aspekte wird in den nachfolgenden Abschnitten detailliert eingegangen.

```
1 <soapenv:Envelope
   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:psn="http://psn.ttp.ganimed.icmvc.emau.org/"
2 <soapenv:Header/>
3 <soapenv:Body>
4   <psn:addDomain>
5     <domainDTO>
6       <name>test_domain</name>
7       <label>Test-Domäne</label>
8       <checkDigitClass>
9         org.emau.icmvc.ganimed.ttp.psn.generator.HammingCode
10      </checkDigitClass>
11      <alphabet>
12        org.emau.icmvc.ganimed.ttp.psn.alphabets.Symbol32
13      </alphabet>
14      <config>
15        <forceCache>OFF</forceCache>
16        <includePrefixInCheckDigitCalculation>
17          false
18        </includePrefixInCheckDigitCalculation>
19        <includeSuffixInCheckDigitCalculation>
20          false
21        </includeSuffixInCheckDigitCalculation>
22        <maxDetectedErrors>1</maxDetectedErrors>
23        <psnLength>8</psnLength>
24        <psnPrefix>psn_</psnPrefix>
25        <psnSuffix></psnSuffix>
26        <psnsDeletable>true</psnsDeletable>
27        <sendNotificationsWeb>false</sendNotificationsWeb>
28        <useLastCharAsDelimiterAfterXChars>
29          0
30        </useLastCharAsDelimiterAfterXChars>
31        <validateValuesViaParents>
32          OFF
33        </validateValuesViaParents>
34        <multiPsnDomain>false</multiPsnDomain>
35      </config>
36      <comment>Dies ist eine Domäne für Testzwecke.</comment>
```

```

37         </domainDTO >
38     </psn:addDomain >
39 </soapenv:Body >
40 </soapenv:Envelope >

```

Listing 4.1: SOAP-Anfrage zum Anlegen einer Domäne.

Tabelle 4.1: Elemente einer SOAP-Anfrage zum Anlegen einer neuen [Domäne](#).

Element	Beschreibung
name	Name der Domäne. Mit diesem wird stets auf diese Domäne über die SOAP-Schnittstelle verwiesen. Der Name kann später nicht mehr verändert werden.
label	Das Label gibt den Namen an, der in der Oberfläche dargestellt wird. Das Label kann später geändert werden.
checkDigitClass	Definiert den Prüfzifferalgorithmus an. Hierbei werden Prüfziffern dem Pseudonym beigefügt, sodass eine Prüfung auf Korrektheit eines Pseudonyms erfolgen kann. Dies ist sinnvoll, wenn Pseudonyme händisch eingetragen werden und dabei Übertragungsfehler zu erwarten sind. Zu beachten ist, dass der gewählte Prüfziffern-Generator mit dem verwendeten Alphabet kompatibel ist. Eine Auflistung kompatibler Kombinationen ist in Abschnitt 4.1.1 zu finden.
alphabet	Angabe zum zu verwendenden Alphabets. Die vordefinierten Alphabete sind in Abschnitt 4.1.2 aufgeführt. Eigene Alphabete können ebenfalls definiert werden. Pseudonyme enthalten nur Zeichen, die im jeweiligen Alphabet enthalten sind.
forceCache	Wird eine sehr hohe Anzahl von Pseudonymen erwartet, kann ein Cache zugeschaltet werden, der die Abrufzeiten verringert ¹ . Standardmäßig ist der Wert <code>DEFAULT</code> . Der gPAS entscheidet selbst, wann der Cache dazugeschaltet wird. Mit <code>ON</code> oder <code>OFF</code> kann der Cache hingegen explizit ein- oder ausgeschaltet werden.

¹ Es werden keine [Pseudonyme](#) dauerhaft im Arbeitsspeicher gehalten. Der Cache besteht aus einem Datencontainer, der sehr performant erkennen kann, ob ein [Pseudonym](#) bereits vergeben wurde oder noch generiert werden muss.

includePrefixInCheckDigitCalculation	Beim Erzeugen von Prüfziffern, kann das Präfix berücksichtigt werden. Voraussetzung hierfür ist, dass ein Prüfzifferalgorithmus angegeben wurde. Mit <code>true</code> wird diese Option ein- und mit <code>false</code> ausgeschaltet.
includeSuffixInCheckDigitCalculation	Beim Erzeugen von Prüfziffern, kann das Suffix berücksichtigt werden. Voraussetzung hierfür ist, dass ein Prüfzifferalgorithmus angegeben wurde. Mit <code>true</code> wird diese Option ein- und mit <code>false</code> ausgeschaltet.
maxDetectedErrors	Gibt die maximale Anzahl von Fehlern an, die erkannt werden sollen. Diese Option kann nur verwendet werden, wenn der Prüfzifferalgorithmus <i>Reed-Solomon-Lagrange</i> gewählt wurde.
psnLength	Gibt die Länge des Pseudonyms an. Diese gibt die Anzahl an Zeichen an, die pro Pseudonym zufällig generiert werden. Die tatsächliche Länge des Pseudonyms ergibt sich aus dieser Pseudonym -Länge und ggf. vergebenen Präfix und Suffix. Die Gesamtlänge des Pseudonyms (inkl. Präfix und Suffix) darf die Länge 255 nicht überschreiten.
psnPrefix	Ein Präfix, der vor den zufälligen Teil eines Pseudonyms angefügt wird.
psnSuffix	Ein Suffix, der nach dem zufälligen Teil eines Pseudonyms angefügt wird.
psnsDeletable	Standardmäßig können Pseudonyme nicht gelöscht werden. Mit Einschalten dieser Option (<code>true</code>) können Pseudonyme gelöscht werden. Standardmäßig wird diese Option für Testzwecke verwendet. In Produktivumgebungen wird üblicherweise diese Option deaktiviert (<code>false</code>), damit vergebene Pseudonyme im Sinne der Auskunftspflicht stets dokumentiert sind. Die Beziehung zwischen Pseudonymen kann im Bedarfsfall anonymisiert werden (vgl. Abschnitt 1.5). Diese Option kann auch aktiviert werden, wenn temporäre Pseudonyme erzeugt werden sollen und diese später restlos entfernt werden sollen.

sendNotifications-Web	Änderungen in der Weboberfläche können über Benachrichtigungen an andere Systeme bekannt gemacht werden. Mit <code>true</code> und <code>false</code> kann diese Option ein- oder angeschaltet werden. Weitere Informationen sind in Kapitel 9 zu finden.
useLastCharAs-DelimiterAfterXChars	Wird ein benutzerdefiniertes Alphabet verwendet, kann das letzte Zeichen des Alphabets als Trennzeichen innerhalb des Pseudonyms verwendet werden. Dabei wird das Trennzeichen nach jeweils n Zeichen eingefügt (wobei n dem angegebenen Wert entspricht). Der Wert muss dabei zwischen 0 und einschließlich 49 gewählt werden. Bei 0 werden keine Trennzeichen eingefügt. Ist das letzte Zeichen im benutzerdefinierten Alphabet ein Minus (-) und $n=3$, dann könnte ein resultierendes Pseudonym so aussehen: 123-456-789.
validateValues-ViaParents	Es kann eine Validierung von Pseudonymen eingestellt werden. Weitere Informationen hierzu, sind im Abschnitt 4.1.3 zu finden.
comment	Beschreibungstext für die Domäne (maximal 255 Zeichen).
parentDomainNames	Name der Eltern-Domäne . Soll die Domäne mehrere Eltern-Domänen besitzen, so muss dieses Element entsprechend oft mit dem jeweiligen Namen angegeben werden. Fungiert diese Domäne als Wurzel, so wird dieses Element nicht angegeben.
multiPsnDomain	Sollen mehrere Pseudonyme pro Originalwert erzeugt werden können, kann diese Option auf <code>true</code> gesetzt werden. Die Domäne wird dann zu einer Multi-Pseudonym-Domäne . Standardmäßig ist diese Option auf <code>false</code> gesetzt und damit ist die Domäne eine Single-Pseudonym-Domäne .

Weitere konkrete Anwendungsbeispiele sind im Kapitel 5 zu finden.

4.1.1 Prüffiffernalgorithmen

Mit dem **gPAS** werden mehrere Prüffiffernalgorithmen mit ausgeliefert, mit denen die Korrektheit von **Pseudonymen** geprüft werden kann. Die verfügbaren Algorithmen sind in Tabelle 4.2 aufgeführt. Nicht jeder Prüffiffernalgorithmus kann mit jedem Alphabet verwendet werden. Welche Alphabete welche Prüffiffernalgorithmen

men unterstützen, ist in Tabelle 4.4 in Abschnitt 4.1.2 aufgelistet.

Wird der Prüfzifferalgorithmus in der SOAP-Anfrage angegeben, so muss beachtet werden, dass der Paketpfad `org.emaui.cmvc.ganimed.ttp.psn.generator` vorangestellt wird (z.B. `org.emaui.cmvc.ganimed.ttp.psn.generator.Damm`).

Tabelle 4.2: Unterstützte Prüfzifferalgorithmen mit den jeweiligen Anforderungen an das Alphabet.

Algorithmus	Voraussetzung
HammingCode	Alphabet-Länge ist eine Primzahlpotenz (Bitte Hinweis unter Tabelle beachten).
Verhoeff	Alphabet-Länge ist gleich 10.
VerhoeffGumm	Alphabet-Länge ist gleich 10.
Damm	Alphabet-Länge ist gleich 10.
ReedSolomonLagrange	Alphabet-Länge ist eine Primzahl, wobei die maximale Anzahl der Prüfzeichen gleich der Alphabet-Länge ist.
NoCheckDigits	Es werden keine Prüfziffern erzeugt. Dies ist der Standard.

Hinweis: Die derzeitige Implementierung des `HammingCode`-Prüfzifferalgorithmus unterstützt derzeit nur eine Alphabet-Länge von 32 (z.B. mit dem Alphabet `Symbol32`).

Bei der Verwendung von Prüfziffern muss bedacht werden, dass die Prüfziffern in Abhängigkeit des verwendeten Algorithmus zusätzlich an das [Pseudonym](#) angefügt werden. Die Prüfziffern werden mit Zeichen des verwendeten Alphabets codiert. Daraus ergibt sich, dass zusätzlich zum zufällig erzeugten Teil mit der angegebenen Länge, die jeweilige Anzahl an Prüfziffern angefügt wird.

Hinweis: Wenn das vergebene Präfix und/oder Suffix bei der Prüfziffernberechnung einbezogen werden soll (`includePrefixInCheckDigitCalculation` und/oder `includeSuffixInCheckDigitCalculation`), muss darauf geachtet werden, dass nur Zeichen verwendet werden, die im ausgewählten Alphabet enthalten sind.

4.1.2 Alphabete

Das verwendete Alphabet definiert die verwendeten Zeichen beim Erzeugen eines **Pseudonyms**². Im **gPAS** sind mehrere Alphabete implementiert und können direkt verwendet werden. Im folgenden Abschnitt werden diese erläutert. Darüber hinaus können spezifische Alphabete definiert werden (Abschnitt 4.1.2.2).

4.1.2.1 Vordefinierte Alphabete

In Tabelle 4.3 sind die mitgelieferten Alphabete aufgelistet. Bei der Verwendung eines mitgelieferten Alphabets muss der Paketpfad `org.emaui.icmvc.ganimed.ttp.psn.alphabets` vorangestellt werden (z.B. `org.emaui.icmvc.ganimed.ttp.psn.alphabets.Symbol32`).

Tabelle 4.3: Vordefinierte Alphabete mit den enthaltenen Zeichen und der daraus resultierenden Anzahl von Zeichen innerhalb eines Alphabets.

Alphabet	Beschreibung
Hex	16 Zeichen: 0-9, A-F
Numbers	10 Zeichen: 0-9
NumbersWithoutZero	9 Zeichen: 1-9
NumbersX	11 Zeichen: 0-9, X
Symbol31	31 Zeichen: 0-9, A-Z (ohne B, I, O, S, V – wegen der Ähnlichkeit zu 8, 1, 0, 5, U)
Symbol32	32 Zeichen: 0-9, A-Z (ohne B, I, O, S – wegen der Ähnlichkeit zu 8, 1, 0, 5)

Sollen zusätzliche Prüfziffern verwendet werden, so muss beachtet werden, dass das jeweilige Alphabet mit dem entsprechenden Prüfzifferalgorithmus kompatibel sein muss. Kompatible Kombinationen werden in Tabelle 4.4 aufgelistet.

Tabelle 4.4: Alphabete mit den jeweils kompatiblen Prüfzifferalgorithmen.

Alphabet	Kompatible Prüfzifferalgorithmen
Hex	<i>keine</i>
Numbers	Verhoeff, VerhoeffGumm, Damm
NumbersWithoutZero	<i>keine</i>
NumbersX	ReedSolomonLagrange
Symbol31	ReedSolomonLagrange
Symbol32	HammingCode

² Unabhängig davon können Präfixe und Suffixe definiert werden, welche auch Zeichen verwenden können, die nicht Teil des Alphabets sind.

4.1.2.2 Benutzerdefinierte Alphabete

Eigene Alphabete können ebenfalls definiert werden. Dabei ist zu beachten, dass diese keine Kommas oder Leerzeichen enthalten dürfen. Dabei können auch Prüfzifferalgorithmen verwendet werden, wenn das Alphabet den Voraussetzungen des Prüfzifferalgorithmus genügt. Die Voraussetzungen sind in Tabelle 4.2 in Abschnitt 4.1.1 aufgeführt. Im Folgenden Listing 4.2 wird ein Alphabet definiert. Dieses enthält die Buchstaben A, B, C und die Ziffern 5, 6, 7, 8, 9. Wie gezeigt, werden im Element `alphabet` die einzelnen Zeichen Komma-separiert aufgelistet. Ein resultierendes **Pseudonym** könnte z.B. so aussehen: C75B79CA.

```

1 [...]
2 <alphabet>A,B,C,5,6,7,8,9</alphabet>
3 [...]
```

Listing 4.2: Beispiel eines benutzerdefinierten Alphabets.

Soll das **Pseudonym** Trennzeichen aufweisen, so muss an das Alphabet das gewünschte Trennzeichen angefügt werden. Außerdem muss für das Element `useLastCharAsDelimiterAfterXChars` ein Wert zwischen jeweils einschließlich 1 und 49 gewählt werden. Nach jedem jeweils n-ten Zeichen wird dann das Trennzeichen eingefügt. Die Länge des **Pseudonyms** ändert sich dabei nicht, da die Trennzeichen nicht zusätzlich eingefügt werden, sondern nur die jeweilige Position einnehmen. Wird z.B. der Wert $n = 2$ gewählt und das Trennzeichen “-” angefügt so könnte das resultierende **Pseudonym** so aussehen: C7-B7-CA.

4.1.3 Validierung von Pseudonymen

Grundsätzlich erzeugt der gPAS selbstständig korrekte **Pseudonyme**, welche in höheren Stufen als **Originalwerte** fungieren. Es können jedoch auch **Pseudonyme** aus z.B. Altbeständen importiert werden, die in die **Pseudonym**-Hierarchie aufgenommen werden sollen. Sollen diese der definierten Struktur entsprechen, kann die Validierung in verschiedenen Optionen durchgeführt werden. Hierzu muss das Element `validateValuesViaParents` mit dem entsprechenden Wert belegt werden. Wenn der **Originalwert** valide und in der **Eltern-Domäne** vorhanden sein muss (VALIDATE). Wenn das **Pseudonym** in der **Eltern-Domäne** vorhanden sein muss (ENSURE_EXISTS). Außerdem kann eingestellt werden, dass das **Pseudonym** in der **Eltern-Domäne** vorhanden sein muss. Wenn dies nicht (mehr) der Fall ist, wird der Eintrag gelöscht (CASCADE_DELETE). Dabei wird `psnsDeletable` ignoriert. Wenn keine Validierung erfolgen soll, muss der Wert OFF gesetzt werden.

5. Anwendungsbeispiele

5.1 genomDE – Phase 0

In Phase 0 von genomDE müssen die Standorte bzw. Leistungserbringer sogenannte Vorgangsnummern erzeugen. Diese werden bei Datenübertragungen an die *klinischen Datenknoten* und *Genomrechenzentren* mit übermitteln. Für beide Kontexte werden unterschiedliche Vorgangsnummern verwendet. Im gPAS werden hierzu entsprechend zwei separate **Domänen** angelegt. Die Vorgangsnummern selbst werden jedoch nach demselben Schema erzeugt. Diese bestehen aus einer 32-Byte langen Zeichenkette, die als Hexadezimalzahl¹ dargestellt wird.

Im gPAS werden hierzu zwei **Domänen** angelegt. Im Folgenden wird exemplarisch die Erzeugung der **Domäne** für den *klinischen Datenknoten* dargestellt. Der Name kann sprechend gewählt werden. Der Schlüssel wird bei der Verwendung der SOAP-Schnittstelle benötigt und wird standardmäßig automatisch erzeugt. Im Beispiel wird dieser mit *ccdn* als Kürzel für *central clinical data node* explizit gesetzt. Es muss keine **Eltern-Domäne** gesetzt werden, da als **Originalwert** der **MPI** einer Person verwendet werden kann. Wird eine davon abweichende Pseudonym-Hierarchie verwendet, so kann die jeweilige **Domäne**, dessen **Pseudonym** als **Originalwert** fungieren soll, entsprechend als **Eltern-Domäne** eingetragen werden. Es kann ein kurzer Beschreibungstext eingetragen werden. Unter *Erweiterte Einstellungen* kann die Option *Erlaube löschen von Pseudonymen* gewählt werden, sodass die Vorgangsnummern im Bedarfsfall entfernt werden können. Unter *Pseudonyme* wird eine Länge von 64 angegeben. Um den Zahlenbereich von einem Byte abzubilden, werden zwei Hexadezimalzahlen benötigt. Da die

¹ Das Hexadezimalsystem ist ein Stellenwertsystem, welches Zahlen zur Basis 16 darstellt. Für die Darstellung werden die Zahlen 0-9 und die Buchstaben A-F verwendet.

Vorgangsnummer aus 32-Byte bestehen soll, ergibt sich eine Länge von 64 Hexadezimalziffern. Es werden weder *Präfix*, noch *Suffix* benötigt. Als *Alphabet* wird das vordefinierte Alphabet *Hex-Zeichen* gewählt. Es sollen keine Prüfziffern erzeugt werden, weshalb bei der Option *Prüfziffern-Generator* die Standardauswahl *Keine Prüfziffern* gewählt wird.

In Listing 5.1 wird die Erzeugung der **Domäne** mit den gezeigten Einstellungen dargestellt.

```

1 <soapenv:Envelope
2   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
3   xmlns:psn="http://psn.ttp.ganimed.icmvc.emau.org/"
4   <soapenv:Header/>
5   <soapenv:Body>
6     <psn:addDomain>
7       <domainDTO>
8         <name>ccdn</name>
9         <label>Klinischer Datenknoten</label>
10        <checkDigitClass>
11          org.emau.icmvc.ganimed.ttp.psn.
12          generator.NoCheckDigits
13        </checkDigitClass>
14        <alphabet>
15          org.emau.icmvc.ganimed.ttp.psn.alphabets.Hex

```

```

14         </alphabet>
15         <config>
16             <forceCache>DEFAULT</forceCache>
17             <includePrefixInCheckDigitCalculation>
18                 false
19             </includePrefixInCheckDigitCalculation>
20             <includeSuffixInCheckDigitCalculation>
21                 false
22             </includeSuffixInCheckDigitCalculation>
23             <psnLength>64</psnLength>
24             <psnPrefix></psnPrefix>
25             <psnSuffix></psnSuffix>
26             <psnsDeletable>true</psnsDeletable>
27             <sendNotificationsWeb>
28                 false
29             </sendNotificationsWeb>
30             <useLastCharAsDelimiterAfterXChars>
31                 0
32             </useLastCharAsDelimiterAfterXChars>
33             <validateValuesViaParents>
34                 OFF
35             </validateValuesViaParents>
36         </config>
37         <comment>
38             Domäne zur Erzeugung von Vorgangsnummern
39             für Datenübertragungen an einen
40             klinischen Datenknoten.
41         </comment>
42     </domainDT0>
43 </psn:addDomain>
44 </soapenv:Body>
45 </soapenv:Envelope>

```

Listing 5.1: SOAP-Anfrage zum Anlegen einer Domäne für die Erzeugung von Vorgangsnummern für Klinische Datenknoten in genomDE.

Die Erzeugung der **Domäne** für die Vorgangsnummern des *Genomrechenzentrums* erfolgt analog zum gezeigten Beispiel. Es müssen lediglich der *Name*, der *Schlüssel* und ggf. der *Beschreibungstext* angepasst werden.

5.2 Verwaltung von Bioproben

Bei der Verwaltung von Bioproben kann es erforderlich sein, für jede Probe ein **Pseudonym** zu vergeben. Daraus ergibt sich, dass pro Personen-Pseudonym (ggf. auch pro Projekt und Person) auf mehrere Bioproben-Pseudonyme referenziert werden soll. Der **gPAS** unterstützt hierfür Multi-Pseudonym-Domänen. Diese erlau-

ben es, pro **Originalwert** mehrere **Pseudonyme** zuzuweisen. Um dies zu aktivieren, wird in der Oberfläche die Option *Erlaube mehrere Pseudonyme für denselben Originalwert* aktiviert. Per SOAP-Request muss die Option *multiPsnDomain* auf `true` gesetzt werden.

Neue Domäne
✕

Bezeichnung *	<input type="text" value="Bioproben"/>	Pseudonyme	
Schlüssel	<input type="text"/>	Länge	<input type="text" value="8"/>
Elterndomänen	<input type="text" value="Domänen filtern"/>	Präfix	<input type="text" value="bio_"/>
Beschreibung	<input type="text" value="Verwaltung von Pseudonymen zu Bioproben."/>	Suffix	<input type="text"/>
	215 Zeichen verbleibend	Alphabet	<input type="text" value="Ziffern und Buchstaben (ohne £)"/>
			Gesamtlänge: 12 Mögliche Pseudonyme: 1.099.511.627.776
Erweiterte Einstellungen		Prüfziffern-Generator	<input type="text" value="HammingCode"/>
Erlaube mehrere Pseudonyme für denselben Originalwert (langsamer)	<input checked="" type="checkbox"/>	Präfix in Prüfziffer einbeziehen	<input type="checkbox"/>
Erlaube löschen von Pseudonymen	<input checked="" type="checkbox"/>	Suffix in Prüfziffer einbeziehen	<input type="checkbox"/>
Sende Benachrichtigungen durch die Weboberfläche	<input type="checkbox"/>		
Cache	<input type="text" value="Automatisch"/>		
Originalwert in Elterndomänen validieren	<input type="text" value="Nicht validieren"/>		
<input checked="" type="button" value="Erstellen"/> <input type="button" value="Abbrechen"/>			

Der SOAP-Request zum gezeigten Beispiel ist in Listing 5.2 dargestellt.

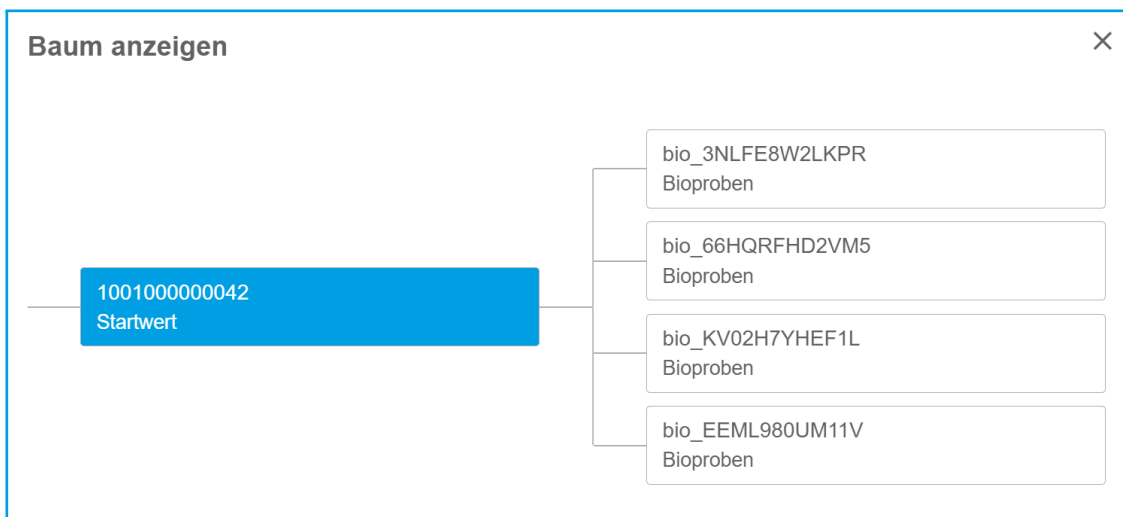
```

1 <soapenv:Envelope
2   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
3   xmlns:psn="http://psn.ttp.ganimed.icmvc.emau.org/"
4   <soapenv:Header/>
5   <soapenv:Body>
6     <psn:addDomain>
7       <domainDT0>
8         <name>Bioproben</name>
9         <label>Bioproben</label>
10        <checkDigitClass>
11          org.emau.icmvc.ganimed.ttp.psn.
12          generator.HammingCode
13        </checkDigitClass>
14        <alphabet>
15          org.emau.icmvc.ganimed.ttp.psn.
16          alphabets.Symbol32
17        </alphabet>
18        <config>

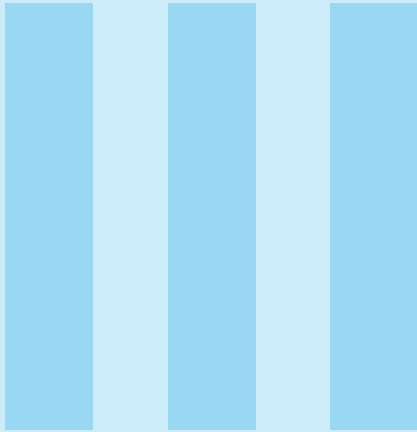
```

```
17         <forceCache>DEFAULT</forceCache>
18         <includePrefixInCheckDigitCalculation>
19             false
20         </includePrefixInCheckDigitCalculation>
21         <includeSuffixInCheckDigitCalculation>
22             false
23         </includeSuffixInCheckDigitCalculation>
24         <maxDetectedErrors>2</maxDetectedErrors>
25         <multiPsnDomain>true</multiPsnDomain>
26         <psnLength>8</psnLength>
27         <psnPrefix>bio_</psnPrefix>
28         <psnSuffix></psnSuffix>
29         <psnsDeletable>true</psnsDeletable>
30         <sendNotificationsWeb>
31             false
32         </sendNotificationsWeb>
33         <useLastCharAsDelimiterAfterXChars>
34             0
35         </useLastCharAsDelimiterAfterXChars>
36         <validateValuesViaParents>
37             OFF
38         </validateValuesViaParents>
39     </config>
40     <comment>
41         Verwaltung von Pseudonymen zu Bioproben.
42     </comment>
43 </domainDTO>
44 </psn:addDomain>
45 </soapenv:Body>
46 </soapenv:Envelope>
```

Listing 5.2: SOAP-Anfrage zum Anlegen einer Multi-Pseudonym-Domäne, am Beispiel einer Pseudonymverwaltung für Bioproben.



Wird nun derselbe **Originalwert** (z.B. derselbe **MPI**) mehrfach verwendet, so werden dennoch verschiedene **Pseudonyme** erzeugt. Diese können dann für verschiedene Bioproben verwendet werden. In der Oberfläche des **gPAS** ergibt sich daraus ein Baum, dessen Wurzel derselbe **Originalwert** ist.



Bedienung

6	Weboberfläche	42
6.1	Generieren von Pseudonymen	42
6.2	Originalwerte und Pseudonyme suchen	44
6.3	Depseudonymisierung (Suche von Originalwerten) .	45
6.4	Technische Anonymisierung	45
6.5	Löschen von Pseudonymen	47
6.6	Integration von Alt-Pseudonymen	48
6.7	Anzeige von Pseudonym-Hierarchien	48
6.8	Listenverarbeitung	49
6.9	Dashboard für Statistiken	51
6.10	Domäne bearbeiten oder löschen	52
6.11	Pseudonyme importieren	53
6.12	Pseudonyme exportieren	54
7	SOAP-Schnittstelle	55
7.1	Pseudonyme anlegen	55



6. Weboberfläche

Um dem Datentreuhänder die Administration der **Pseudonyme** zu ermöglichen, verfügt der **gPAS** über eine grafische Benutzeroberfläche die speziell für den Einsatz im Web-Browser entwickelt wurde. Der Aufbau der Oberfläche orientiert sich dabei an typischen Arbeitsabläufen eines Datentreuhänders.

6.1 Generieren von Pseudonymen

Voraussetzung für die Generierung von **Pseudonymen** ist ein angelegter **Pseudonym**-Baum in Form einer oder mehrerer **Domänen**. Nachträglich können weitere **Domänen** angelegt werden.

Generieren eines einzelnen Pseudonyms

Unter dem Menüpunkt *Suchen / Generieren* können einzelne **Pseudonyme** angelegt werden. Hierzu wird zunächst die gewünschte **Domäne** aus der Liste ausgewählt und danach die Schaltfläche **Neues Pseudonym generieren** ausgewählt. Im sich öffnenden Fenster muss zum einen die **Domäne** und zum anderen der **Originalwert** angegeben werden. Ist der **Originalwert** beispielsweise ein **Pseudonym** erster Stufe, wird ein **Pseudonym** zweiter Stufe generiert. Das generierte **Pseudonym** folgt dann den entsprechenden Vorgaben der **Domäne** (siehe vorheriger Abschnitt). In Abbildung 6.1 wird die Oberfläche zum Anlegen eines dargestellt.

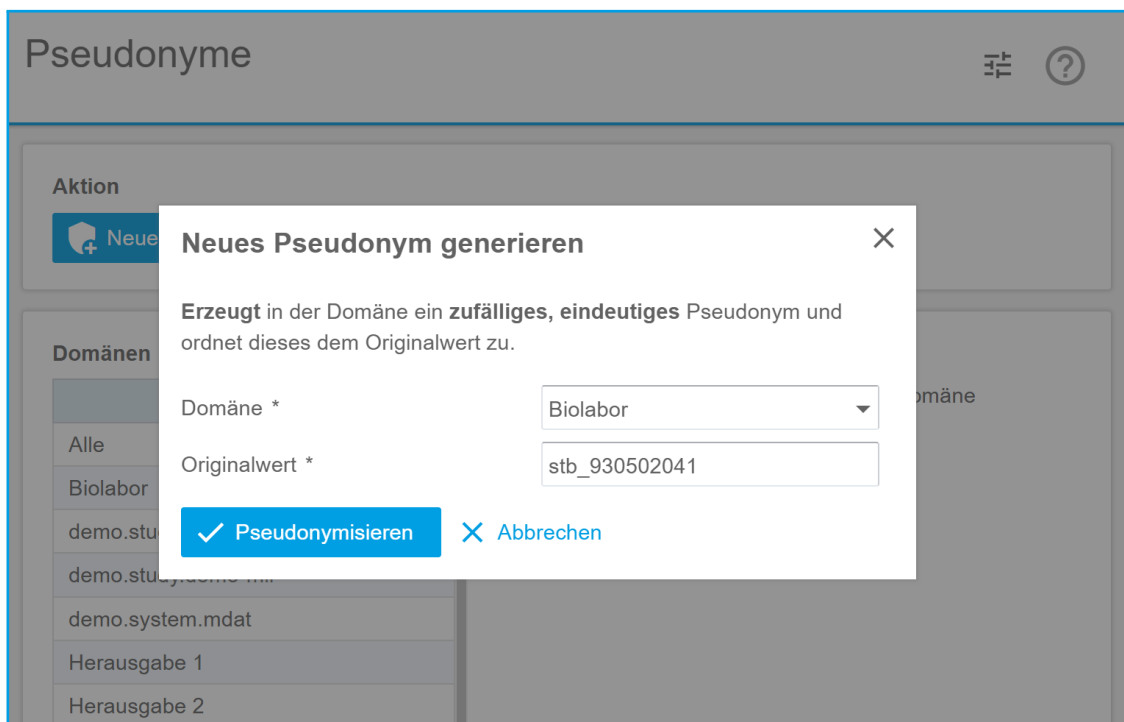


Abbildung 6.1: Oberfläche zum Anlegen eines neuen .

Generieren eines Zweit-Pseudonyms

Oftmals ist das Generieren von Zweit-Pseudonymen (oder beliebig vielen Pseudonymen) erforderlich, z.B. bei unterschiedlichen Pseudonymen je Studienzentrum und Datentyp. In diesem Fall empfiehlt es sich jeweils eigene Domänen anzulegen. Als Originalwerte werden dann die bei der Erst-Pseudonymisierung generierten Pseudonyme bzw. die Pseudonyme der niedrigeren Stufe verwendet (z.B. ein PID oder MPI). Bei einem Pseudonym höherer Stufe wird als Originalwert ein zuvor generiertes Pseudonym verwendet. Dieses kann im Kontextmenü (Rechtsklick auf den Eintrag) über den Eintrag *Kopiere Pseudonym* kopiert werden. Um ein Zweit-Pseudonym zu generieren, wird die Domäne höherer Stufe gewählt (Kind-Domäne) und dort über die Schaltfläche **Neues Pseudonym generieren** das eben kopierte Pseudonym als Originalwert eingetragen.

Der gPAS erkennt die möglichen Kind-Domänen, sodass alternativ ein Pseudonym nur angewählt werden und im Kontextmenü der Eintrag *Pseudonyme Pseudonym* gewählt werden muss. Danach kann im Auswahlmeneü Domäne eine der Kind-Domäne ausgewählt werden. Der Originalwert wird mit dem eben gewählten Pseudonym vorbefüllt. In Abbildung 6.2 wird das entsprechende Kontextmenü dargestellt.

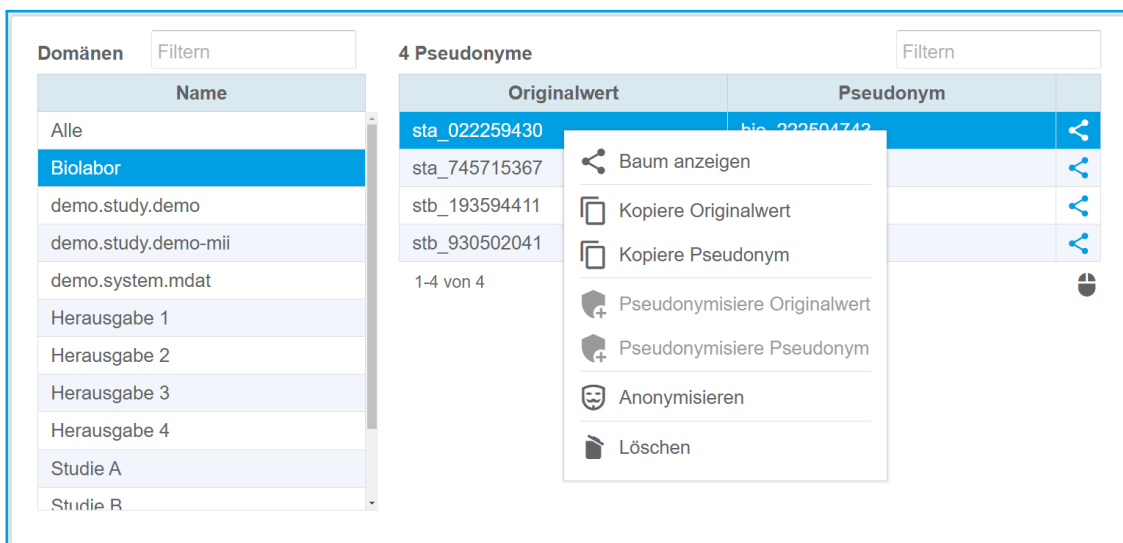


Abbildung 6.2: Kontextmenü zum Erzeugen von **Pseudonymen** derselben Stufe (*Pseudonymisiere Originalwert*) und einer höheren Stufe in einer **Kind-Domäne** (*Pseudonymisiere Pseudonym*).

In Abbildung 6.3 wird eine beispielhafte Struktur für mehrere **Pseudonyme** mit mehreren Stufen dargestellt. Der Studienteilnehmer hätte in diesem Fall zwei **Pseudonyme** zweiter Stufe, jeweils eins für Studie A und eins für Studie B. Bei beiden stellt das **Pseudonym** erster Stufe (100100000011) den **Originalwert** dar. Dies beispielsweise der **MPI** aus dem **E-PIX** sein. Basierend auf dem **Pseudonym** für Studie B, wurden zwei weitere **Pseudonyme** dritter Stufe generiert. Der **Originalwert** der **Pseudonyme** dritter Stufe ist dabei das **Pseudonym** zweiter Stufe.

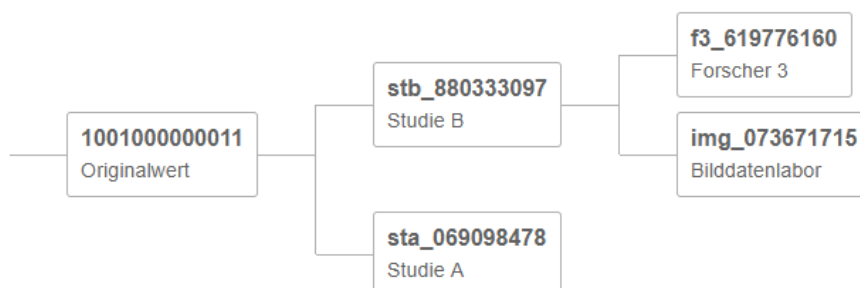


Abbildung 6.3: Exemplarische Struktur bei mehreren **Pseudonymen** und Stufen für einen Studienteilnehmer.

6.2 Originalwerte und Pseudonyme suchen

Originalwerte und **Pseudonyme** werden auf die gleiche Weise gesucht. Hierzu wird unter dem Menüpunkt *Suchen / Generieren* eine **Domäne** ausgewählt. Wenn ein **Originalwert** gesucht wird, muss hierzu die **Domäne** gewählt werden, in der

sich das höherstufige **Pseudonym** befindet. Wenn ein **Pseudonym** gesucht wird, dann wird die **Domäne** gewählt, in der sich das jeweilige **Pseudonym** befindet. In das obere rechte Suchfeld (über der **Pseudonym**-Auflistung) wird die gesuchte Zeichenkette eingetragen. Die dargestellte Tabelle wird nach dem Drücken der Enter-Taste gefiltert. Es werden nur die Einträge angezeigt, bei der eine exakte Übereinstimmung vorhanden ist. Hierbei ist es irrelevant, an welcher Position sich die eingegebene Zeichenkette befindet. In Abbildung 6.4 ist die Oberfläche zum Suchen von **Originalwerten** und **Pseudonymen** dargestellt.

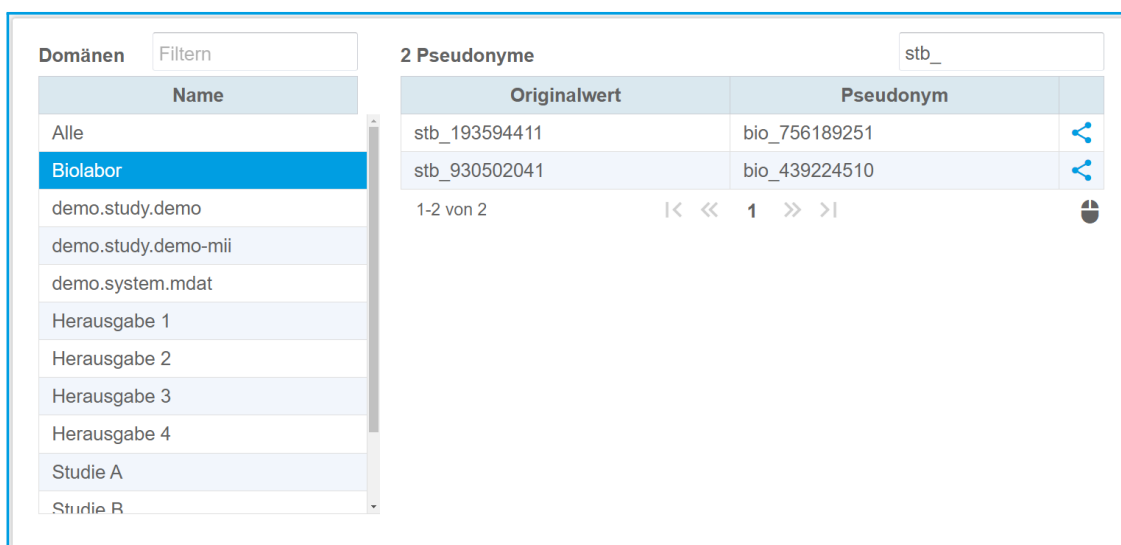


Abbildung 6.4: Oberfläche zum Suchen von **Originalwerten** oder **Pseudonymen**.

Eine Suche über alle **Domänen** hinweg kann durchgeführt werden, indem in der Domänenaufstellung der Eintrag *Alle* gewählt wird. Die Suche erfolgt dann wie eben beschrieben über das Suchfeld. Die gefilterte Tabelle beinhaltet zudem noch die Angabe, aus welcher **Domäne** der jeweilige Eintrag stammt.

6.3 Depseudonymisierung (Suche von Originalwerten)

Eine **Depseudonymisierung** entspricht der Suche eines **Originalwertes** anhand eines gegebenen **Pseudonyms**. Es wird demnach so vorgegangen, wie im Abschnitt 6.2 beschrieben. Hierzu wird das vorhandene **Pseudonym** in der jeweiligen **Domäne** gesucht. Die Suche liefert den dazugehörigen **Originalwert** bzw. im Fall einer Pseudonym-Hierarchie das **Pseudonym** geringerer Stufe.

6.4 Technische Anonymisierung

Bei der technischen **Anonymisierung** (auch bekannt als virtuelle **Anonymisierung** und im Folgenden der Einfachheit halber **Anonymisierung** genannt) wird die Zuordnung zwischen einem **Originalwert** und einem **Pseudonym** unwiederbringlich

aufgehoben. Das **Pseudonym** bleibt dabei erhalten, der **Originalwert** hingegen wird durch einen neu generierten Platzhalter ersetzt. Der **Originalwert** als **Pseudonym** niedrigerer Stufe bleibt ebenfalls erhalten. Da die Zuordnung jedoch aufgehoben wurde, ist eine **Depseudonymisierung** nicht mehr möglich.

Zur **Anonymisierung** wird unter dem Menüpunkt *Suchen / Generieren* die entsprechende **Domäne** gewählt. Danach wird in der Liste das zu anonymisierende **Pseudonym** ausgewählt (zuvor kann mittels Suche die Liste gefiltert werden, vgl. Abschnitt 6.2). Mit einem Rechtsklick auf den entsprechenden Eintrag wird das Kontextmenü geöffnet. Mit dem Auswählen des Eintrags *Anonymisieren*, wird die Zuordnung aufgehoben.

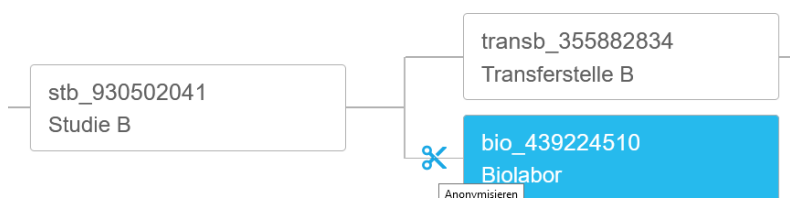


Abbildung 6.5: **Anonymisierung** in der Baumstruktur durch Auftrennen der Verbindung (Schere).

Alternativ dazu, kann bei Betrachtung der Baumstruktur die Verbindung mittels des kleinen Scheren-Symbols aufgelöst werden (siehe Abbildung 6.5). In jedem Fall, muss die Durchführung dieses unumkehrbaren Vorgangs bestätigt werden (siehe Abbildung 6.6).

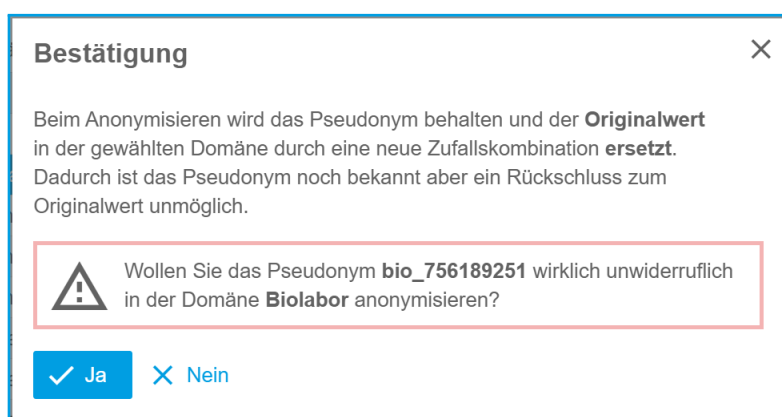


Abbildung 6.6: Der unumkehrbare Vorgang der **Anonymisierung** muss nochmals bestätigt werden.

Nach Bestätigung des Vorgangs, wird die **Anonymisierung** durchgeführt und der Eintrag entsprechend aktualisiert. Die **Anonymisierung** ist dadurch ersichtlich, dass

der **Originalwert** mit dem Wert `###_anonym_###_..._###_anonym_###` ersetzt wird (siehe Abbildung 6.7).

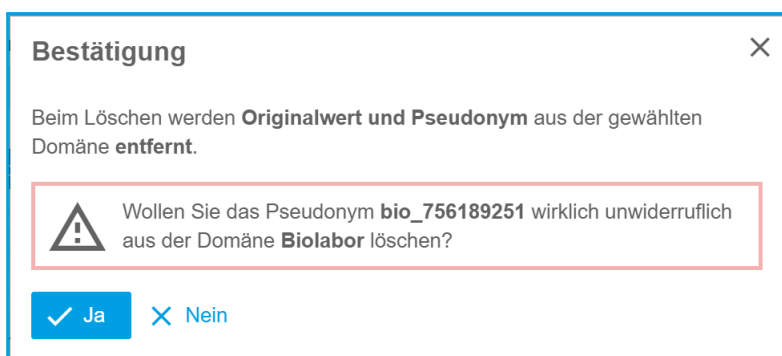
Domänen	4 Pseudonyme																					
<table border="1"> <thead> <tr> <th>Name</th> </tr> </thead> <tbody> <tr><td>Alle</td></tr> <tr><td>Biolabor</td></tr> <tr><td>demo.study.demo</td></tr> <tr><td>demo.study.demo-mii</td></tr> <tr><td>demo.system.mdat</td></tr> <tr><td>Herausgabe 1</td></tr> <tr><td>Herausgabe 2</td></tr> <tr><td>Herausgabe 3</td></tr> <tr><td>Herausgabe 4</td></tr> <tr><td>Studie A</td></tr> </tbody> </table>	Name	Alle	Biolabor	demo.study.demo	demo.study.demo-mii	demo.system.mdat	Herausgabe 1	Herausgabe 2	Herausgabe 3	Herausgabe 4	Studie A	<table border="1"> <thead> <tr> <th>Originalwert</th> <th>Pseudonym</th> </tr> </thead> <tbody> <tr> <td>###_anonym_###_CPNW5NwRA8NQ_###_anonym_###</td> <td>bio_756189251</td> </tr> <tr> <td>sta_022259430</td> <td>bio_222504743</td> </tr> <tr> <td>sta_745715367</td> <td>bio_358822846</td> </tr> <tr> <td>stb_930502041</td> <td>bio_439224510</td> </tr> </tbody> </table>	Originalwert	Pseudonym	###_anonym_###_CPNW5NwRA8NQ_###_anonym_###	bio_756189251	sta_022259430	bio_222504743	sta_745715367	bio_358822846	stb_930502041	bio_439224510
Name																						
Alle																						
Biolabor																						
demo.study.demo																						
demo.study.demo-mii																						
demo.system.mdat																						
Herausgabe 1																						
Herausgabe 2																						
Herausgabe 3																						
Herausgabe 4																						
Studie A																						
Originalwert	Pseudonym																					
###_anonym_###_CPNW5NwRA8NQ_###_anonym_###	bio_756189251																					
sta_022259430	bio_222504743																					
sta_745715367	bio_358822846																					
stb_930502041	bio_439224510																					

Abbildung 6.7: Anonymisierter Eintrag.

6.5 Löschen von Pseudonymen

Beim Löschen eines **Pseudonym**-Paares, wird der **Originalwert** und das dazugehörige **Pseudonym** unwiederbringlich aus der **Domäne** entfernt.

War das **Pseudonym** Teil einer Hierarchie bleiben die anderen Elemente der Hierarchie erhalten und müssen bei Bedarf ebenfalls gelöscht werden. In einer niedrigeren Hierarchiestufe bleibt der gelöschte **Originalwert** also als **Pseudonym** erhalten. In einer höheren Stufe bleibt das gelöschte **Pseudonym** als **Originalwert** erhalten.




Eine **Depseudonymisierung** ist mittels des gelöschten nicht mehr möglich. Um ein **Pseudonym** löschen zu können, muss eine **Domäne** entsprechend konfiguriert sein (vgl. Abschnitt 3). Zum Löschen wird unter dem Menüpunkt *Suchen / Generieren* die entsprechende **Domäne** und das zu löschende **Pseudonym** angewählt. Mit einem Rechtsklick auf den Eintrag wird das Kontextmenü geöffnet. Mit dem Auswählen des Eintrags *Löschen* wird der Eintrag gelöscht. Bevor dieser unumkehrbare Vorgang ausgeführt wird, muss dies bestätigt werden. Wurden etwaige

Kind-Domäne dahingehend konfiguriert, dass diese eine Validierung besitzen und hierbei die Option *Muss vorhandenes Pseudonym sein und kaskadiere bei Löschung in Elterndomäne* gewählt, werden auch in den **Kind-Domänen** die auf das **Pseudonym** verweisenden Einträge gelöscht.

6.6 Integration von Alt-Pseudonymen

Der Import von **Pseudonymen** aus vorherigen Projekten oder von Alt-Beständen kann über zwei Wege erfolgen.


Einzelne Pseudonyme manuell eintragen

Um ein **Pseudonym** anzulegen, welches als **Originalwert** eines bereits existenten **Pseudonyms** dient, muss unter dem Menüpunkt *Suchen / Generieren* die Schaltfläche ** Pseudonympaar eintragen** gewählt werden. In das sich öffnende Fenster kann die **Domäne**, der **Originalwert** und das **Pseudonym** eingetragen werden. Dabei bezieht sich die **Domäne** auf das anzulegende **Pseudonym**. Es ist nicht möglich, ein bereits existierendes **Pseudonym** erneut anzulegen.

Listen importieren

Bei größeren Alt-Beständen, können diese komplett in bestehende **Domänen** importiert werden. Hierzu kann unter dem Menüpunkt *Importieren* eine Datei mit den zu importierenden **Originalwerten** und **Pseudonymen** eingetragen werden. Die Zieldomäne kann manuell ausgewählt werden oder aus einer weiteren Spalte ausgelesen werden. Letzteres funktioniert auch bei verschiedenen Zieldomänen. Das **Originalwert-Pseudonym**-Paar wird dann in die jeweilige **Domäne** importiert. Beim Import sind etwaige Validierungen zu berücksichtigen, die bei der Domänenkonfiguration hinterlegt wurden (siehe Abschnitt 3.1).

6.7 Anzeige von Pseudonym-Hierarchien

Wurden bei der Konfiguration der **Domäne** eine **Eltern-Domäne** zur Angabe der Beziehung zweier **Domänen** (Eltern-Kind-Beziehung) angegeben, so kann der dadurch entstehende **Pseudonym**-Baum übersichtlich dargestellt werden. Hierzu wird unter dem Menüpunkt *Suchen / Generieren* die **Domäne** und ein **Pseudonym** gewählt. Über das Kontextmenü über den Eintrag *Baum anzeigen* oder direkt über die Schaltfläche **** kann die entsprechende Hierarchie in einem Fenster dargestellt werden. Dabei sind die angewählte **Domäne** und das **Pseudonym** farblich hervorgehoben. In Abbildung 6.8 ist die entsprechende Oberfläche dargestellt.

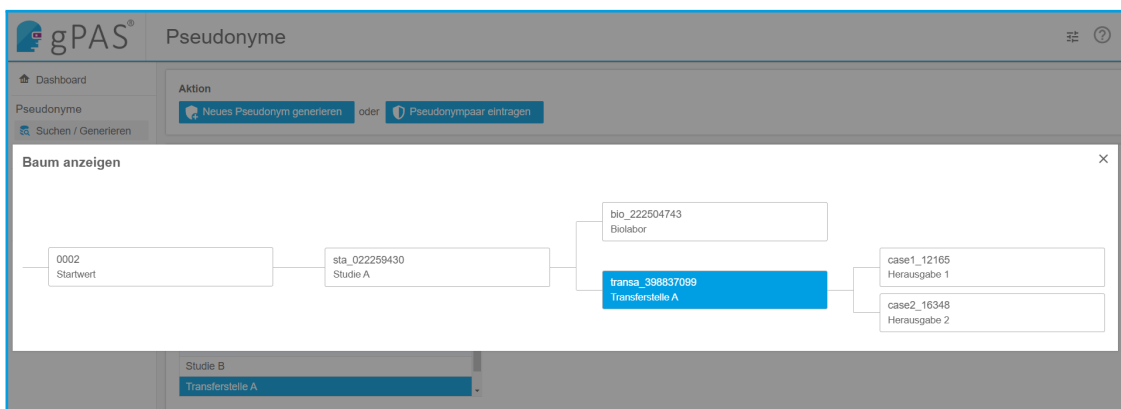


Abbildung 6.8: Oberfläche beim Anzeigen der Pseudonymhierarchie für ein selektiertes Pseudonym (blau dargestellt).

Es werden neben dem gewählten Pseudonym auch der Originalwert und alle verknüpften Pseudonyme dargestellt.

6.8 Listenverarbeitung

Es ist möglich, eine Liste von Eingabewerten zu pseudonymisieren, zu depseudonymisieren (ermitteln des Originalwertes), zu anonymisieren oder zu löschen. Hierzu kann unter dem Menüpunkt *Listen verarbeiten* eine CSV-Datei ausgewählt werden. In Abbildung 6.9 ist die entsprechende Oberfläche zum Auswählen einer Datei abgebildet. Hierbei kann ausgewählt werden, welcher Separator in der CSV-Datei verwendet wurde (standardmäßig “;”) und ob die Datei eine Kopfzeile/Spaltenbezeichner (*Die Liste besitzt eine Kopfzeile mit Spaltenbezeichnung*) aufweist. Exporte aus dem gPAS weisen stets eine Kopfzeile auf. Der zu verwendende Separator kann auch direkt in der CSV-Datei in der ersten Zeile per `sep=X` deklariert werden. Dabei steht X für das Zeichen, welches als Separator verwendet werden soll.

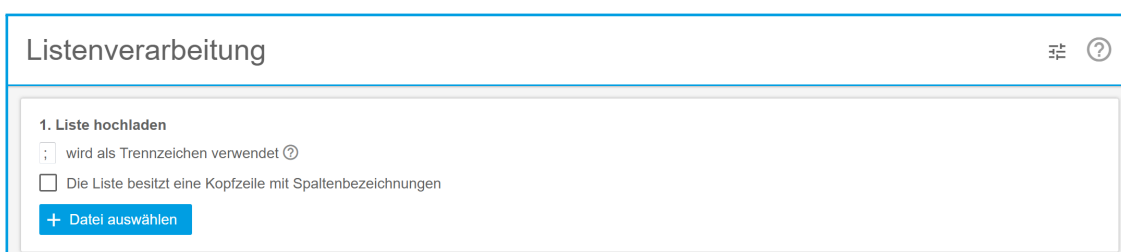


Abbildung 6.9: Oberfläche zum Verarbeiten von Listen.

Beim Hochladen erkennt der gPAS automatisch die Kodierung der Datei. Diese kann bei Bedarf angepasst werden. Danach kann die Spalte gewählt werden,

dessen Werte verarbeitet werden sollen. Die enthaltenen Werte werden dabei bereits in der Oberfläche als Vorschau angezeigt (vgl. Abbildung 6.10).

Listenverarbeitung
⌵ ?

1. Liste hochladen

Liste 2024-05-27 Pseudonym-Export Biolabor demo.study.demo demo.study.demo-mii demo.system.mdat Herausgabe 1 Herau... gPAS.csv mit 20 Datensätzen erfolgreich hochgeladen.

Folgendes Encoding wurde erkannt: UTF-16LE

[Liste verwerfen](#)

2. Spalte wählen Filtern

Originalwert	Pseudonym	Domäne
###_anonym_###_CPNW5NWRA8NQ_###_anonym_###	bio_756189251	Biolabor
sta_022259430	bio_222504743	Biolabor
sta_745715367	bio_358822846	Biolabor
stb_930502041	bio_439224510	Biolabor
1001000000011	demo_56919218148	demo.study.demo
demo_56919218148	mdat_969292	demo.system.mdat
transa_398837099	case1_12165	Herausgabe 1
transa_682719299	case1_96459	Herausgabe 1
transa_398837099	case2_16348	Herausgabe 2
transb_742367979	case3_484881704	Herausgabe 3

1-10 von 20 |< << 1 2 >> >|

3. Daten verarbeiten

Durchzuführende Aktion * Pseudonymisieren

Zieldomäne * Bitte wählen ...

Abbildung 6.10: Wählen der Verarbeitungsoperation. Hier am Beispiel von **Pseudonymisieren**.

Es kann zwischen vier Verarbeitungsoperationen gewählt werden. Die entsprechenden Optionen sind in Tabelle 6.1 aufgelistet.

Tabelle 6.1: Mögliche Verarbeitungsoperationen.

Operation	Beschreibung
Pseudonymisieren	Für jedes Element der Liste ein neues Pseudonym erzeugen (sofern noch nicht bekannt).
Depseudonymisieren	Für jedes Pseudonym der Liste die Originalwerte ermitteln (sofern bekannt).
Anonymisieren	Jedes Pseudonym der Liste wird anonymisiert.
Löschen	Jedes Pseudonym der Liste wird aus dem Bestand entfernt.

Die zu nutzende Pseudonym-Domäne muss angegeben werden. Nach dem Wählen der Schaltfläche Verarbeiten die Operation auf den Datenbestand vom **gPAS** angewandt. Das Ergebnis dieser Verarbeitung wird in der dargestellten

Tabelle ergänzt. Hierfür kann die Ergebnisspalte zusätzlich benannt werden. Diese aktualisierte Liste kann im Anschluss als CSV-Datei heruntergeladen werden.

6.9 Dashboard für Statistiken

Die Anzahl der vorhandenen **Pseudonyme**, Anonyme und Pseudonym-Domänen im **gPAS** können angezeigt werden und als CSV-Datei zum aktuellen Stand oder mit Verlaufsdaten exportiert werden. Unter dem Menüpunkt *Dashboard* können Tabellen und Diagramme eingesehen werden, welche die jeweiligen Daten aufbereitet darstellen. In Abbildung 6.11 ist die Oberfläche der Statistik abgebildet.

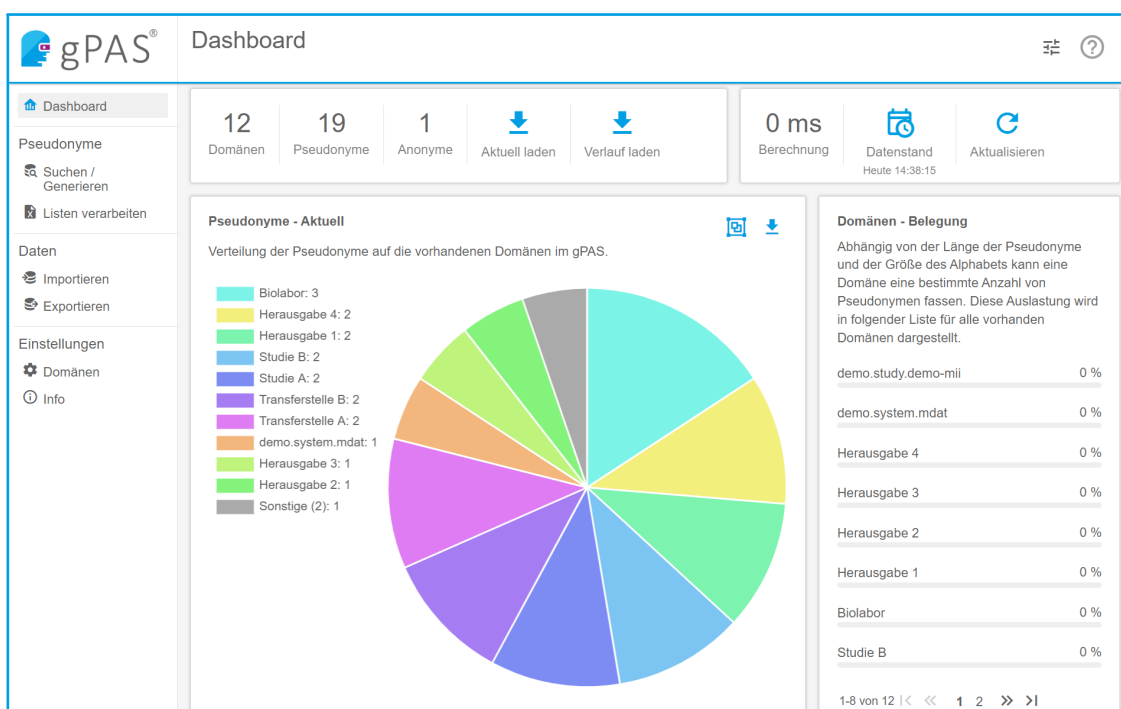



Abbildung 6.11: Oberfläche zum Einsehen von der Anzahl von **Pseudonymen**, Anonymen und **Domänen**. Die Daten sind in Diagrammen aufgeführt.

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 **gPAS** erzeugt und in der Datenbank dokumentiert.

Die Statistik kann als CSV über die jeweiligen Schaltflächen () heruntergeladen werden.

Info: Unterstützung bei regelmäßiger Community-Kennzahlenerhebung.

Das Dashboard liefert einen schnellen Überblick über Zahlen zu **Pseudonymen** und **Domänen**. Diese können als CSV-Datei exportiert und der Unabhängigen Treuhandstelle Greifswald per E-Mail übermittelt werden. Dies beinhaltet ausschließlich aggregierte Daten und keine konkreten **Pseudonyme**. Das unterstützt bei statistischen Auswertungen über die Gesamtzahl von **Pseudonymen** und **Domäne** in der Community. Vielen Dank fürs Mitmachen!

6.10 Domäne bearbeiten oder löschen

Die Voraussetzung zum Bearbeiten oder Löschen einer **Domäne** ist, dass keine **Pseudonyme** in der entsprechenden **Domäne** hinterlegt sind. Um eine **Domäne** zu bearbeiten oder zu löschen, wird unter dem Menüpunkt *Domänen* die entsprechende **Domäne** ausgewählt und mit einem Rechtsklick das Kontextmenü aufgerufen (vgl. Abbildung 6.12).

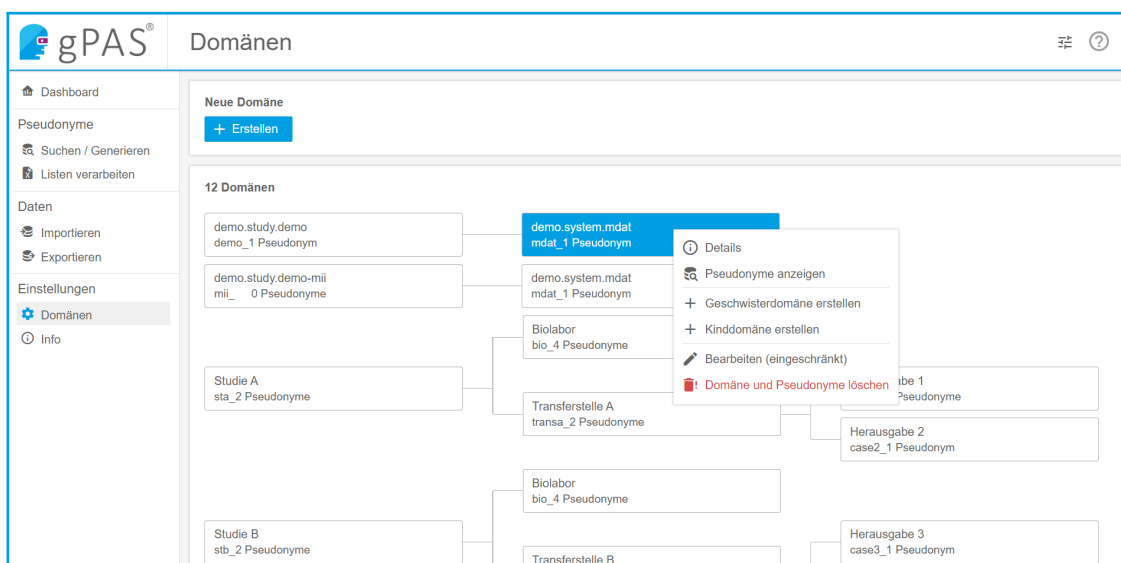


Abbildung 6.12: Kontextmenü mit den Schaltflächen zum Anzeigen der Domänen-details, zum Bearbeiten der **Domäne** und zum Löschen der **Domäne**. Hierüber können weitere **Domänen** erzeugt werden (siehe Abschnitt 3.1).

Zum Bearbeiten wird die Schaltfläche **Bearbeiten**¹ angewählt. Daraufhin können alle Einstellungen, außer der Namen, zur **Domäne** bearbeitet werden. Wenn die Schaltfläche **Domäne und Pseudonyme löschen** gewählt wird, muss der Vorgang bestätigt werden und die **Domäne** wird unwiederbringlich gelöscht.

¹ Bei **Domänen** mit bereits vorhandenen **Pseudonymen** steht hier der Zusatz "(eingeschränkt)".

Hinweis: Wird der Bezeichner der **Domäne** geändert, betrifft dies nur den angezeigten Namen bzw. das Label. Wird die **Domäne** über die SOAP-Schnittstelle angesprochen, muss der ursprüngliche Name verwendet werden.

6.11 Pseudonyme importieren

Unter dem Menüpunkt *Suchen / Generieren* können einzelne Pseudonympaare eingetragen werden. Um Listen dieser Paare (**Originalwert**, **Pseudonym**) zu importieren, können unter dem Menüpunkt *Import* komplette Dateien importiert werden. Dies passiert auf vergleichbare Weise, wie die Listenverarbeitung (vgl. Abschnitt 6.8). Hierzu wird zunächst eine Datei ausgewählt, die importiert werden soll. Dabei kann angegeben werden, ob ein spezieller Separator in der Datei verwendet wurde (... *wird als Trennzeichen verwendet*). Außerdem kann angegeben werden, ob die Datei eine Kopfzeile/Spaltenbezeichner aufweist. Wenn die Datei aus einer **gPAS**-Instanz exportiert wurde, ist standardmäßig eine Kopfzeile enthalten und es wird der Separator “;” verwendet.

1. Datei hochladen
Liste 2024-10-15 Pseudonym-Export T1 gPAS.csv mit 11 Datensätzen erfolgreich hochgeladen.
Folgendes Encoding wurde erkannt: UTF-16LE
[Liste verwerfen](#)

2. Spalten wählen Filtern

Originalwert	Originalwert	Pseudonym	Pseudonym
19283745		893249239	
28464824		390482034	
27391856		390428304	
96245582		217373423	
72379012		897238293	
87354811		678462384	
82648012		386947239	
72358203		238932684	
49204528		236718423	
29836292		892388742	

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

3. Optionen
Zieldomäne * Studie A
 Datei enthält eine Spalte mit Zieldomänen
[Importieren](#)

Abbildung 6.13: Oberfläche zum Importieren von Pseudonympaaren.

Der Inhalt der ausgewählten Datei wird in einer Vorschau dargestellt. Die Zieldomäne kann ausgewählt werden. Enthält die Datei eine Spalte für die Zieldomäne (dies können auch verschiedene **Domänen** sein), so kann alternativ hierzu die entsprechende Option (*Datei enthält eine Spalte mit Zieldomänen*) gewählt werden. In jedem Fall müssen die jeweiligen **Domänen**-Konfigurationen einschließlich etwaiger Validierungen berücksichtigt werden, welche einen Import ggf. unterbin-

den. Der Importvorgang wird über die Schaltfläche **Importieren** gestartet. Die dargestellte Tabelle wird um das Importergebnis ergänzt (Erfolg/Misserfolg). In Abbildung 6.13 ist die Oberfläche mit einem exemplarischen Import dargestellt.

6.12 Pseudonyme exportieren

Die **Pseudonyme** und dessen **Originalwerte** können exportiert werden. Hierbei kann gewählt werden, aus welchen **Domänen** die **Pseudonyme** exportiert werden sollen. Der Export wird über den Menüpunkt *Exportieren* aufgerufen. Die zu exportierenden **Domänen** können hierbei einzeln angewählt werden. Mittels eines Klicks auf die Schaltfläche **>** werden die zu exportierenden **Domänen** in der rechten Liste gesammelt. Soll ein vollständiger Export stattfinden, genügt der Klick auf die Schaltfläche **>>**. Mit einem Klick auf die Schaltfläche **Exportieren** wird der Export gestartet. Die resultierende Datei kann ohne weitere Anpassungen beispielsweise wieder importiert werden. In Abbildung 6.14 ist die entsprechende Oberfläche dargestellt.

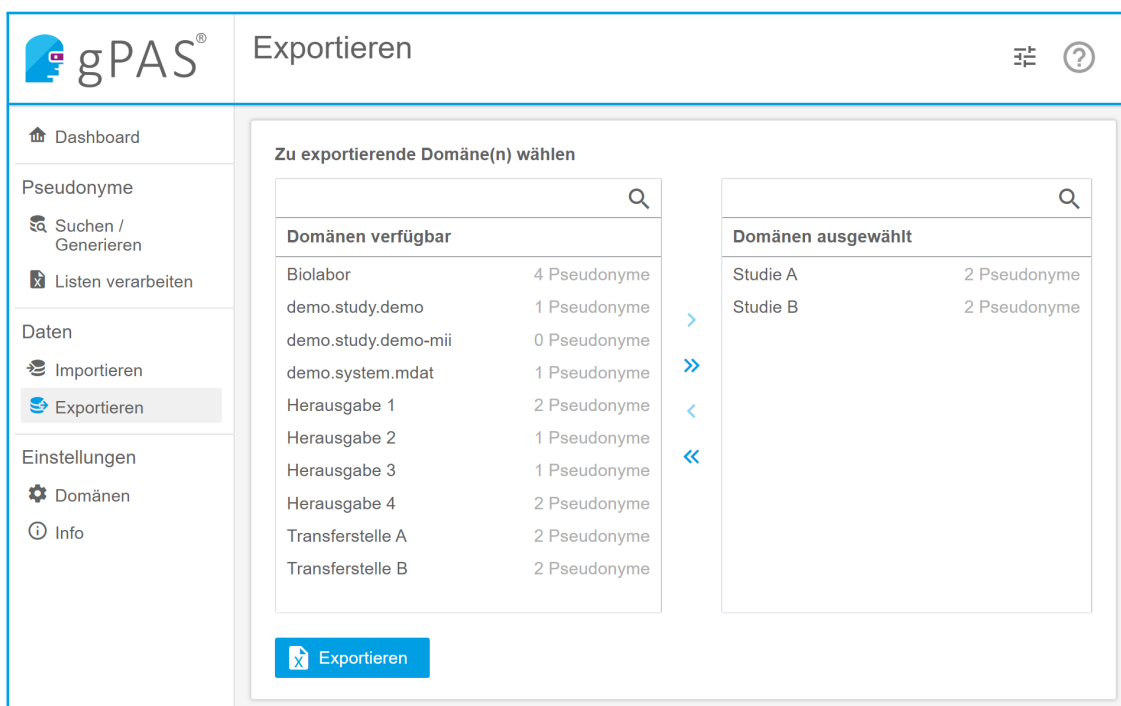


Abbildung 6.14: Oberfläche zum Exportieren beliebiger **Domänen**.

7. SOAP-Schnittstelle

Die SOAP-Schnittstelle zur Pseudonymverwaltung (anlegen, abrufen, etc.) von **Pseudonymen** ist unter folgender URL erreichbar:

<http://example.org:8080/gpas/gpasService?wsdl>

7.1 Pseudonyme anlegen

Beim Anlegen von **Pseudonymen** wird zwischen der Art der **Domäne** (**Single-Pseudonym-Domäne** oder **Multi-Pseudonym-Domäne**) unterschieden. Außerdem erlaubt die SOAP-Schnittstelle, unabhängig von der Art der **Domäne**, das Anlegen von nur einem oder mehreren **Pseudonymen** pro Anfrage. Für die jeweiligen Anfragen sind im Folgenden Beispiele angegeben.

7.1.1 Single-Pseudonym-Domäne

Soll ein oder mehrere **Pseudonyme** in eine **Single-Pseudonym-Domäne** registriert werden, kann pro **Originalwert** nur ein **Pseudonym** angelegt werden. Die entsprechende SOAP-Anfrage zum Anlegen eines **Pseudonyms** wird in Listing 7.1 dargestellt. Hierbei wird die Methode `createPseudonymFor` verwendet. Im Element `value` wird der **Originalwert** hinterlegt. Im Element `domainName` wird der Name der **Domäne** angegeben. Alternativ kann die Methode `getOrCreatePseudonymFor` verwendet werden, welche das etwaig vorhandene **Pseudonym** zum angegebenen **Originalwert** liefert und nur im Bedarfsfall ein neues **Pseudonym** generiert.

```
1 <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:psn="http://psn.ttp.ganimed.icmvc.emau.org/">
```

```
2     <soapenv:Header/>
3     <soapenv:Body>
4         <psn:createPseudonymFor>
5             <value>org_123456</value>
6             <domainName>projekt-a</domainName>
7         </psn:createPseudonymFor>
8     </soapenv:Body>
9 </soapenv:Envelope>
```

Listing 7.1: SOAP-Anfrage zur Erstellung eines neuen **Pseudonyms**.

Der **gPAS** liefert für den angegebenen **Originalwert** ein **Pseudonym** zurück. Dieses ist im Element `psn` zu finden. Eine exemplarische Rückgabe ist in Listing 7.2 dargestellt.

```
1 <soap:Envelope
2     xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
3     <soap:Body>
4         <ns2:createPseudonymForResponse
5             xmlns:ns2="http://psn.ttp.ganimed.icmvc.emau.org/"
6             <psn>psn_959831</psn>
7         </ns2:createPseudonymForResponse>
8     </soap:Body>
9 </soap:Envelope>
```

Listing 7.2: SOAP-Rückgabe mit dem erzeugten **Pseudonym**.

Innerhalb einer Anfrage können auch mehrere **Pseudonyme** erzeugt werden. In einer **Single-Pseudonym-Domäne** müssen dafür entsprechend viele **Originalwerte** angegeben werden. Hierzu wird die Methode `createPseudonymForList` verwendet. Das Element `values` kann mehrfach angegeben werden und muss unterschiedliche **Originalwerte** beinhalten. Mit dem Element `domainName` wird der Name der **Domäne** angegeben. Alternativ kann auch hier die Methode `getOrCreatePseudonymForList` verwendet werden, die nur im Bedarfsfall **Pseudonyme** erzeugt und ansonsten das bereits vorhandene **Pseudonym** zurückliefert. In Listing 7.3 wird exemplarisch gezeigt, wie mit einer Anfrage mehrere **Pseudonyme** für verschiedene **Originalwerte** erzeugt werden können.

```
1 <soapenv:Envelope
2     xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
3     xmlns:psn="http://psn.ttp.ganimed.icmvc.emau.org/"
4     <soapenv:Header/>
5     <soapenv:Body>
6         <psn:createPseudonymForList>
7             <values>org_123_A</values>
8             <values>org_123_B</values>
```



```
7         <values>org_123_C</values>
8         <values>org_123_D</values>
9         <values>org_123_E</values>
10        <domainName>projekt-a</domainName>
11    </psn:createPseudonymForList>
12 </soapenv:Body>
13 </soapenv:Envelope>
```

Listing 7.3: SOAP-Anfrage zur Erstellung mehrerer **Pseudonyme** für verschiedene **Originalwerte** innerhalb einer Anfrage.

Der **gPAS** liefert eine Liste zurück. Die Einträge werden als `entry` Element dargestellt, wobei das Element `key` den **Originalwert** und das Element `value` das jeweils erzeugte **Pseudonym** beinhaltet. Zu beachten ist, dass die Reihenfolge der Einträge der Rückgabe, von der Reihenfolge der Eingabe abweichen kann. In Listing 7.4 ist eine exemplarische Rückgabe dargestellt.

```
1 <soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2   <soap:Body>
3     <ns2:createPseudonymForListResponse
4       xmlns:ns2="http://psn.ttp.ganimed.icmvc.emau.org/">
5       <return>
6         <entry>
7           <key>org_123_B</key>
8           <value>psn_691624</value>
9         </entry>
10        <entry>
11          <key>org_123_A</key>
12          <value>psn_473462</value>
13        </entry>
14        ...
15      </return>
16    </ns2:createPseudonymForListResponse>
17  </soap:Body>
</soap:Envelope>
```

Listing 7.4: SOAP-Rückgabe bei der Erstellung mehrerer **Pseudonyme** für verschiedene **Originalwerte** innerhalb einer Anfrage.

7.1.2 Multi-Pseudonym-Domäne

Sollen pro **Originalwert** mehrere **Pseudonyme** erzeugt werden, muss eine **Multi-Pseudonym-Domäne** verwendet werden. Mit der Methode `createPseudonymsFor` können für einen **Originalwert** (Element `value`) mehrere **Pseudonyme** innerhalb einer **Domäne** (Element `domainName`) erzeugt werden. Hierzu wird mit dem

Element `number` die Anzahl der zu erzeugenden **Pseudonyme** angeben (es werden dabei immer entsprechend viele neue **Pseudonyme** erzeugt, unabhängig davon, wie viele **Pseudonyme** dem jeweiligen **Originalwert** bereits zugeordnet sind). In Listing 7.5 ist eine exemplarische SOAP-Anfrage dargestellt. Alternativ kann die Methode `getOrCreatePseudonymsFor` verwendet werden. Diese erzeugt nur im Bedarfsfall neue **Pseudonyme**, wenn die angegebene Mindestanzahl an **Pseudonymen** noch nicht für dem jeweiligen **Originalwert** zugeordnet sind. Wenn bereits **Pseudonyme** vorhanden sind, werden nur so viele **Pseudonyme** erzeugt, bis die Mindestanzahl erreicht wird. Existieren bereits mehr **Pseudonyme** als die Mindestanzahl erfordert, so werden keine neuen **Pseudonyme** erzeugt und nur die bereits vorhandenen **Pseudonyme** zurückgeliefert. Die Anzahl der gelieferten **Pseudonyme** kann dann die Mindestanzahl übersteigen.

```
1 <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:psn="http://psn.ttp.ganimed.icmvc.emau.org/"
2   <soapenv:Header/>
3   <soapenv:Body>
4     <psn:createPseudonymsFor>
5       <value>org_123456</value>
6       <domainName>projekt-b</domainName>
7       <number>3</number>
8     </psn:createPseudonymsFor>
9   </soapenv:Body>
10 </soapenv:Envelope>
```

Listing 7.5: SOAP-Anfrage zur Erstellung mehrerer **Pseudonyme** für einen **Originalwert**.

Der **gPAS** liefert eine Liste mit den erzeugten **Pseudonymen** zurück. Diese sind im Element `psn` enthalten. In Listing 7.6 ist eine exemplarische Antwort auf die eben gezeigt Anfrage dargestellt. Es werden drei **Pseudonyme** zurückgeliefert.

```
1 <soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
2   <soap:Body>
3     <ns2:createPseudonymsForResponse
4       xmlns:ns2="http://psn.ttp.ganimed.icmvc.emau.org/"
5       <return>
6         <psn>psn_332527</psn>
7         <psn>psn_896859</psn>
8         <psn>psn_170217</psn>
9       </return>
10    </ns2:createPseudonymsForResponse>
  </soap:Body>
```

```
11 </soap:Envelope>
```

Listing 7.6: SOAP-Rückgabe zur Erstellung mehrerer **Pseudonyme** für einen **Originalwert**.

Innerhalb einer Anfrage können auch mehrere **Pseudonyme** zu verschiedenen **Originalwerten** erzeugt werden. Hierzu wird die Methode `createPseudonymsForList` verwendet. Das Element `values` kann mehrfach angegeben werden und beinhaltet die **Originalwerte**. Wird derselbe **Originalwert** mehrfach angegeben, so wird trotzdem nur die angegebene Anzahl an **Pseudonymen** erzeugt. Mit dem Element `domainName` wird der Name der **Domäne** angegeben. Alternativ kann auch hier die Methode `getOrCreatePseudonymsForList` verwendet werden, die nur im Bedarfsfall **Pseudonyme** erzeugt und ansonsten das bereits vorhandene **Pseudonym** zurückliefert. Mit dem Element `number` wird die Anzahl an **Pseudonymen** angegeben, die für jeden **Originalwert** erzeugt werden. In Listing 7.7 wird exemplarisch gezeigt, wie mit einer Anfrage mehrere **Pseudonyme** für mehrere **Originalwerte** erzeugt werden können.

```
1 <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:psn="http://psn.ttp.ganimed.icmvc.emau.org/"
2   <soapenv:Header/>
3   <soapenv:Body>
4     <psn:createPseudonymsForList>
5       <values>org_111111</values>
6       <values>org_222222</values>
7       <values>org_333333</values>
8       <domainName>projekt-b</domainName>
9       <number>2</number>
10    </psn:createPseudonymsForList>
11  </soapenv:Body>
12 </soapenv:Envelope>
```

Listing 7.7: SOAP-Anfrage zur Erstellung mehrerer **Pseudonyme** für mehrere **Originalwerte** innerhalb einer Anfrage.

Der **gPAS** liefert eine Liste zurück. Die Einträge werden als `multiPsn` Element gruppiert, wobei das Attribut `value` den **Originalwert** und die Elemente `psn` die jeweils erzeugten **Pseudonyme** beinhalten. Zu beachten ist, dass die Reihenfolge der Einträge der Rückgabe, von der Reihenfolge der Eingabe abweichen kann. In Listing 7.8 ist eine exemplarische Rückgabe dargestellt.

```
1 <soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
```

```
2     <soap:Body>
3         <ns2:createPseudonymsForListResponse
4             xmlns:ns2="http://psn.ttp.ganimed.icmvc.emau.org/">
5             <return>
6                 <multiPsn value="org_333333">
7                     <psn>psn_183962</psn>
8                     <psn>psn_596102</psn>
9                 </multiPsn>
10                <multiPsn value="org_111111">
11                    <psn>psn_808473</psn>
12                    <psn>psn_302243</psn>
13                </multiPsn>
14                <multiPsn value="org_222222">
15                    <psn>psn_655029</psn>
16                    <psn>psn_543285</psn>
17                </multiPsn>
18            </return>
19        </ns2:createPseudonymsForListResponse>
20    </soap:Body>
21</soap:Envelope>
```

Listing 7.8: SOAP-Rückgabe bei der Erstellung mehrerer **Pseudonyme** für mehrere **Originalwerte** innerhalb einer Anfrage.

IV

Integration

8	Logging	62
9	Benachrichtigungen	63
10	FHIR-Unterstützung	64
11	Authentifizierung & Autorisierung ...	66
11.1	Global	66
11.2	Domänen-spezifische Rollen mit OpenID-Connect .	67
12	Empfehlungen zur Absicherung	69
13	Optimierungen	70
13.1	Speicher für MySQL erhöhen	70
13.2	Batch-Writing	70

8. Logging

Hinweis: Details für die Anpassung der Logging-Konfiguration entnehmen Sie bitte der beigelegten Beschreibung `docker-compose/README.md` (Abschnitt Logging).



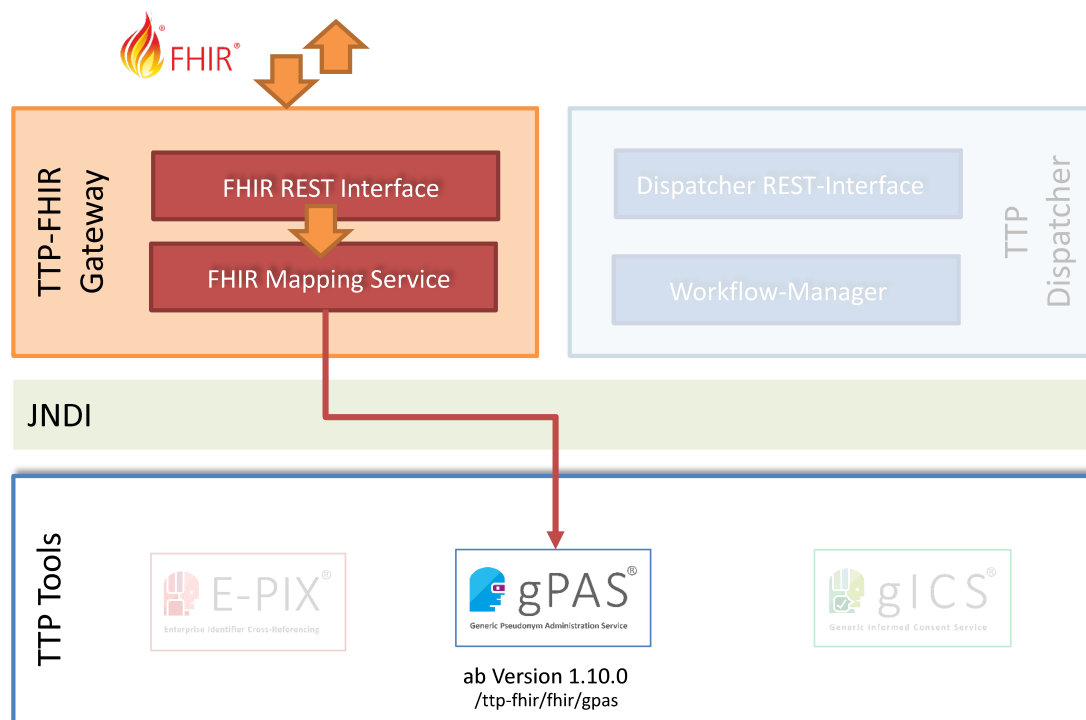
9. Benachrichtigungen

Wie in der Architekturgrafik zu sehen (siehe Abbildung 2.1), ist der **gPAS** seit Version 1.12.0 in der Lage Benachrichtigungen an externe Systeme zu versenden. Dies kann per `http`, `MQTT` oder `EJB` erfolgen. Die Versandmitteilungen werden in einer separaten Notification-Datenbank dokumentiert.

Hinweis: Details zum Umfang der Notification-Schnittstelle, zur Einrichtung, sowie weitere Erläuterungen sind separat unter <https://www.ths-greifswald.de/ttp-tools/notifications> dokumentiert.

10. 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“¹.



© 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 ab

¹ https://de.wikipedia.org/wiki/Fast_Healthcare_Interoperability_Resources

gPAS-Version 1.10.0 ein zusätzliches Treuhandstellen-FHIR-Gateway (kurz: TTP-FHIR Gateway) als Mittler von FHIR-spezifischen Infrastrukturkomponenten und gPAS bereitgestellt.

Für ausgewählte Funktionalitäten zur domänenspezifischen Generierung neuer **Pseudonyme** und zur Abfrage von Mappings von **Pseudonym**- und Originalwerten wurden in FHIR Funktionen umgesetzt und sind nach erfolgreichem Deployment direkt per REST nutzbar. Eine Liste der umgesetzten Funktionen ist unter <https://ths-greifswald.de/gpas/fhir> zu finden.

Hinweis: Die Profilierung der erforderlichen Profile, Codesysteme und Operations erfolgte in Zusammenarbeit mit der Fa. gefyra^a.

^a<https://www.gefyra.de/>, Zugriff am 2021-06-08

Details zur Nutzung der Funktionen (Aufruf, Parameter, beispielhafte Antworten) können unter dem frei verfügbaren Implementation-Guide entnommen werden.

11. Authentifizierung & Autorisierung

11.1 Global

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

Standardmäßig ist im **gPAS** keine Authentifizierung notwendig. Soll der **gPAS** nur für bestimmte Nutzergruppen (Admin-Nutzer, Standard-Nutzer) zugänglich gemacht werden (vgl. Tabelle 11.1) 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/TTP-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.

11.1.1 Übersicht Nutzerrollen und Rechte

Tabelle 11.1: Nutzer-Zugriffsrechte in der Weboberfläche.

Bereich/Seite	Zugang ohne Login	Zugang mit User-Rechten	Zugang mit Admin-Rechten
Einstellungen: Info	×	×	×
Einstellungen: Domänen			×
Einstellungen: Statistiken		×	×
Listen: Importieren			×
Pseudonyme: Suchen/Anlegen		×	×
Pseudonyme: Exportieren			×

11.1.2 Übersicht Nutzerrollen und Rechte

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/TTP-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.

11.1.3 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.

11.2 Domänen-spezifische Rollen mit OpenID-Connect

Mit der rollenbasierter **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 Webinterface entsprechend eingeschränkt werden, sofern diese authentifiziert und mit aktivierter rollenbasierter **Domänen**-Absicherung erfolgen.

Das zweistufige Rollensystem mit Admin- und User-Rollen (vgl. Abschnitt 11.1) 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-greifswald.de/ttp-tools/domain-auth> dokumentiert.



12. Empfehlungen zur Absicherung

Der Zugriff auf relevante Anwendungs- und Datenbankserver des **gPAS** 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
- Zugangsbeschränkung auf URL-Ebene mit Basic Authentication (z.B. mit NGINX oder Apache)

13. Optimierungen

Wird entgegen der hier beschriebenen Vorgehensweise selbst ein Applikationsserver und Datenbankserver aufgesetzt, so kann eine Performance-Steigerung des gPAS durch diverse Optimierungen erzielt werden. In den von der Treuhandstelle Greifswald ausgelieferten Docker-Containern (WildFly und MySQL) sind diese bereits eingerichtet. Diese Optimierungen sind relevant, wenn mehr als 10 Mio. Datenbankeinträge erwartet werden.

13.1 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.

13.2 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` des WildFly-Servers der Parameter `rewriteBatchedStatements=true` an die `jdbc-connection-url` angefügt wird.

13.2.1 Lange Zeiten zum Hochfahren des Applikationsservers

Wurden viele Millionen Pseudonyme 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 abzu-passen. 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) hochgesetzt.

```
1 <subsystem xmlns="urn:jboss:domain:deployment-scanner:2.0">
2   <deployment-scanner [...] scan-interval="5000"
3     deployment-timeout="900" [...] />
</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, 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/ARTN17610.1186/s12967-015-0545-6)
3. Bialke M, Rau H, Thamm OC, Schuldt R, Penndorf P, Blumentritt A, Gott R, Piegsa J, Bahls T und Hoffmann W. Toolbox for Research, or How to Facilitate a Central Data Management in Small-Scale Research Projects. *Journal of Translational Medicine.* 2018 Jan 25; 16:16. DOI: [10.1186/s12967-018-1390-1](https://doi.org/10.1186/s12967-018-1390-1)

Glossar

Anonymisieren Veränderung von Patientendaten zur jeweiligen natürlichen Person, dass eine Zuordnung nicht oder nur mit verhältnismäßig viel Aufwand möglich ist. Im gPAS werden keine Personendaten gespeichert, weshalb eine Anonymisierung das unwiederbringliche Löschen von Zuordnungen zwischen **Originalwert** und **Pseudonym** vorsieht. Ein Rückschluss auf die Person ist dann nicht mehr möglich.

Depseudonymisieren Ermitteln des Originalwerts, eines gegebenen .

Domäne Eine Domäne stellt den Kontext dar, in dem die Pseudonymisierung stattfindet. Die konkrete Konfiguration definiert dabei, wie die Pseudonymisierung umgesetzt wird. Dies umfasst z.B. die Länge und das Alphabet eines **Pseudonyms**. Der Kontext kann z.B. ein Projekt oder ein Fachbereich sein. **Pseudonyme** sind dabei innerhalb einer Domäne immer eindeutig. Das heißt, zu jedem **Pseudonym** existiert genau ein **Originalwert**. Zu jedem **Originalwert** kann es aber mehrere **Pseudonyme** geben (bei **Multi-Pseudonym-Domänen**).

Single-Pseudonym-Domäne Eine Single-Pseudonym-Domäne ist eine **Domäne**, die pro **Originalwert** nur ein **Pseudonym** aufweist. **Multi-Pseudonym-Domäne** erlauben pro **Originalwert** mehrere **Pseudonyme** (*Glossar: Multi-Pseudonym-Domäne*)

Multi-Pseudonym-Domäne Eine Multi-Pseudonym-Domäne ist eine **Domäne**, die pro **Originalwert** mehrere **Pseudonyme** erlaubt. **Single-Pseudonym-Domäne** erlauben pro **Originalwert** nur ein **Pseudonym** (*Glossar: Single-Pseudonym-Domäne*)

Eltern-Domäne Eine Eltern-Domäne ist eine **Domäne**, die eine niedrige Stufe von Pseudonymen abbildet. Eine Eltern-Domäne kann beliebig viele **Kind-Domänen** aufweisen.

Geschwister-Domäne Eine Geschwister-Domäne ist eine **Domäne**, die sich auf gleicher Stufe mit einer anderen **Domäne** befindet. Beide **Domänen** teilen sich dabei dieselbe **Eltern-Domäne**.

Kind-Domäne Eine Kind-Domäne ist eine **Domäne**, die eine höhere Stufe von **Pseudonymen** abbildet. Eine Kind-Domäne hat immer eine **Eltern-Domäne**.

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.

Originalwert Der Originalwert ist der Wert, der innerhalb einer **Domäne** pseudonymisiert werden soll. Dies kann z.B. ein **PID** oder **MPI** sein. Auf höheren Pseudonym-Stufen ist der Originalwert ein **Pseudonym** einer niedrigeren Pseudonym-Stufe.

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

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.

Pseudonymisieren Erzeugen eines nicht-sprechenden Identifikators, basierend auf einem gegebenen **Originalwert**.

Abkürzungsverzeichnis

DSGVO Datenschutz-Grundverordnung

E-PIX Enterprise Identifier Cross-Referencing

gPAS generic pseudonym administration service

IDAT Identifizierende Daten. *Glossar:* [Identifizierende Daten](#)

MPI Master Patient Index

PID Personenidentifikator. *Glossar:* [Patientenidentifikator](#)