

Anwenderhandbuch



Generic Pseudonym Administration Service

Version 1.10.2 vom 12.07.2021

Herausgeber:

Unabhängige Treuhandstelle der Universitätsmedizin Greifswald

Autor:

Christopher Hampf, M.Sc.

Ellernholzstr. 1-2
17475 Greifswald

Tel. 03834 / 86-7851, Fax: 03834 / 86-6843

E-Mail: christopher.hampf@uni-greifswald.de

Versionierung

Version	Datum	Bearbeitungsart / Betroffene Abschnitte	Bearbeiter
1.0	25.03.2019	Update und Erweiterung der Doku von mosaic-greifswald.de	Christopher Hampf
1.0.1	19.06.2019	Hinzufügen des Hinweises auf etwaige Probleme hinsichtlich der Verwendung von Volumes und VPN-Clients	Christopher Hampf
1.0.2	15.07.2019	Anpassung von Begrifflichkeiten	Christopher Hampf
1.0.3	09.08.2019	Domänenschnittstelle ergänzt	Martin Bialke
1.9.0	21.11.2019	Abgleich der Versionierung mit gPAS Diverse Ergänzungen und Korrekturen	Christopher Hampf
1.9.1	04.03.2020	Web-Auth für Version 1.9.1 ergänzt	Martin Bialke
1.10.2	11.03.2021	Aktualisierung der Bilder zu Benutzeroberflächen Ergänzungen zu neuen Funktionalitäten Autorisierung/Authentifizierung überarbeitet FHIR Gateway ergänzt Optimierungen ergänzt Logging ergänzt	Martin Bialke Christopher Hampf
1.10.2	12.07.2021	Link zu https://www.ths-greifswald.de/ttp-tools/keycloak ergänzt und Links FHIR GW aktualisiert	Martin Bialke

Inhalt

Anwenderhandbuch	1
Versionierung	2
Inhalt	4
Abbildungsverzeichnis	5
Tabellenverzeichnis	6
1 Hintergrund	8
2 Begriffsbestimmungen	8
3 Funktionalitäten	9
3.1 Was leistet der Dienst.....	9
3.2 Was leistet der Dienst nicht.....	10
4 Installation	10
4.1 Systemanforderungen	10
4.2 Download und Starten des Dienstes.....	10
5 Die grafische Benutzeroberfläche des gPAS®	12
5.1 Anwendungsfall 1: Anlegen einer Domäne	13
5.2 Anwendungsfall 2: Generieren von Pseudonymen	16
5.3 Anwendungsfall 3: Originalwerte und Pseudonyme suchen.....	18
5.4 Anwendungsfall 4: Suchen von Originalwerten (Depseudonymisierung).....	19
5.5 Anwendungsfall 5: Technische Anonymisierung (Virtuelle Anonymisierung).....	19
5.6 Anwendungsfall 6: Löschen von Pseudonymen	20
5.7 Anwendungsfall 7: Integration von Alt-Pseudonymen.....	21
5.8 Anwendungsfall 8: Anzeige von Pseudonym-Hierarchien	21
5.9 Anwendungsfall 9: Listenverarbeitung	22
5.10 Anwendungsfall 10: Statistik ermitteln	23
5.11 Anwendungsfall 11: Domäne bearbeiten oder löschen	24
5.12 Anwendungsfall 12: Pseudonyme exportieren.....	24
6 Logging	25
7 Authentifizierungs- und Autorisierung	25
7.1 Verwendung von KeyCloak	26
7.2 Verwendung von gRAS.....	26
8 Empfehlungen zur Absicherung des Anwendungsservers	29
9 Nutzung der SOAP-Schnittstelle	30

9.1	Anlegen einer Domäne	30
9.2	Anlegen von Pseudonymen	32
9.3	Pseudonymisieren (Abfragen von Pseudonymen)	32
9.4	De-Pseudonymisieren (Abfragen von Originalwerten).....	33
10	FHIR-Unterstützung für gPAS per FHIR-Gateway	34
11	Health Checks	35
11.1	Abfrage.....	35
11.2	Konfiguration	36
12	Optimierungen	37
12.1	Speicher für MySQL erhöhen.....	37
12.2	Batch-Writing.....	37
12.3	Lange Zeiten zum Hochfahren des Applikationsservers.....	37
13	Publikationen und Vorträge	38
14	Weiterführende Informationen	39

Abbildungsverzeichnis

Abbildung 1-1: Zuweisung und Verwaltung von projektspezifischen Pseudonymen zu einem Personenidentifikator mittels gPAS®.	8
Abbildung 4-1: Architektur des gPAS® am Beispiel der Version 1.10.0.	11
Abbildung 5-1: Oberfläche zum Anzeigen aller Domänen. Der Baum zeigt die hierarchische Struktur der Domänen. Das Kontextmenü enthält weitere Optionen.	13
Abbildung 5-2: Oberfläche zum Anlegen einer neuen Domäne.	14
Abbildung 5-3: Oberfläche zum Anlegen eines neuen Pseudonyms.	17
Abbildung 5-4: Kontextmenü zum Erzeugen von nebengeordneten oder übergeordneten Domänen.	17
Abbildung 5-5: Exemplarische Struktur bei mehreren Pseudonymen und Stufen für einen Studienteilnehmer.....	18
Abbildung 5-6: Oberfläche zum Suchen von Originalwerten oder Pseudonymen.	18
Abbildung 5-7: Mit einem Rechtsklick auf den Eintrag, kann das Kontextmenü aufgerufen werden..	19
Abbildung 5-8: Der unumkehrbare Vorgang der Anonymisierung muss nochmals bestätigt werden.	20
Abbildung 5-9: Anonymisierter Eintrag.....	20

Abbildung 5-10: Oberfläche beim Anzeigen der Pseudonymhierarchie für ein selektiertes Pseudonym (blau dargestellt).	21
Abbildung 5-11: Oberfläche zum Verarbeiten von Listen.	22
Abbildung 5-12: Wählen der Verarbeitungsoperation. Hier am Beispiel von <i>Pseudonymisieren</i>	22
Abbildung 5-13: Oberfläche zum Einsehen von der Anzahl von Pseudonymen, Anonymen und Domänen.	23
Abbildung 5-14: Kontextmenü mit den Schaltflächen zum Anzeigen der Domänendetails, zum Bearbeiten der Domäne und zum Löschen der Domäne. Hierüber können weitere Domänen erzeugt werden (s. Anwendungsfall 1: Anlegen einer Domäne).	24
Abbildung 5-15: Oberfläche zum Exportieren beliebiger Domänen.	25
Abbildung 8-1: XML-Repräsentation zum Anlegen einer Domäne über die SOAP-Schnittstelle.	31
Abbildung 8-2: XML-Repräsentation der Rückgabe beim erfolgreichen Anlegen einer Domäne über die SOAP-Schnittstelle.	31
Abbildung 8-3: XML-Repräsentation einer beispielhaften Anfrage zum Anlegen oder Abfragen eines Pseudonyms über die SOAP-Schnittstelle.	32
Abbildung 8-4: XML-Repräsentation der Rückgabe zu einer Anfrage zum Anlegen oder Abrufen eines Pseudonyms über die SOAP-Schnittstelle.	32
Abbildung 8-5: XML-Repräsentation einer beispielhaften Anfrage zum Abfragen eines Pseudonyms über die SOAP-Schnittstelle.	33
Abbildung 8-6: XML-Repräsentation der Rückgabe zu einer Anfrage zum Abrufen eines Pseudonyms über die SOAP-Schnittstelle.	33
Abbildung 8-7: XML-Repräsentation einer beispielhaften Anfrage zum Abfragen eines Originalwerts über die SOAP-Schnittstelle.	33
Abbildung 8-8: XML-Repräsentation der Rückgabe zu einer Anfrage zum Abrufen eines Originalwerts über die SOAP-Schnittstelle.	34

Tabellenverzeichnis

Tabelle 5-1: Bereitgestellte Alphabete im gPAS®.	14
Tabelle 5-2: Mögliche Prüfziffern-Generatoren und dessen Bedingungen.	15
Tabelle 5-3: Mögliche Domain Properties.	15
Tabelle 5-4: Mögliche Verarbeitungsoperationen.	23
Tabelle 6-1: Nutzer der Gruppe Admin und User haben unterschiedliche Zugriffsrechte in der Web-Oberfläche.	26
Tabelle 8-1: Alphabete und die jeweils kompatiblen Prüfziffernalgorithmen.	30

Tabelle 9-1: Übersicht der durch FHIR unterstützten gPAS-Funktionen. 34

1 Hintergrund

Die Durchführung klinisch-epidemiologischer Studien, aber auch der Aufbau von Registern und Kohorten, erfordern eine datenschutzkonforme Datenverarbeitung. Gemäß Art. 32 Abs. 1a 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 (kurz: gPAS®) entwickelt. Das Web-service-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.

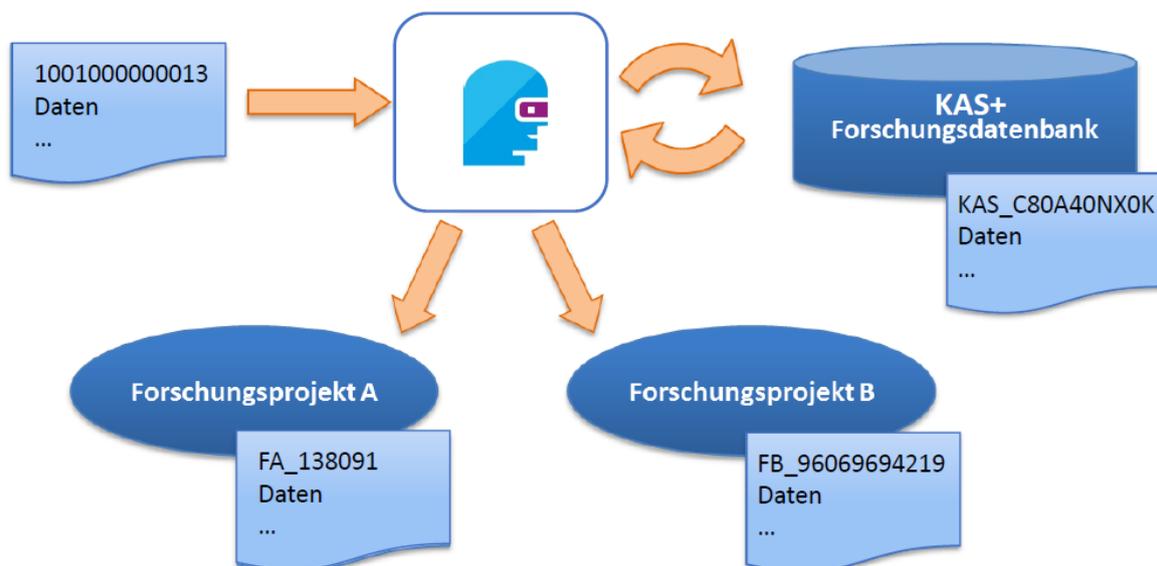


Abbildung 1-1: Zuweisung und Verwaltung von projektspezifischen Pseudonymen zu einem Personenidentifikator mittels gPAS®.

2 Begriffsbestimmungen

Domäne (Domain)

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

Originalwert

Der Originalwert bezeichnet den Wert, für den ein Pseudonym höherer Stufe generiert und diesem Originalwert im Anschluss zugewiesen wird.

Pseudonym

Nicht-sprechender Identifikator, welcher einer Person zugewiesen ist. Ein Pseudonym erster Stufe verweist dabei direkt auf die Personendaten, ein Pseudonym höherer Stufe jeweils nur auf das Pseudonym der jeweils niedrigeren Stufe.

Pseudonymisieren

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

Depseudonymisieren

Ermitteln des Originalwerts, eines gegebenen Pseudonyms.

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 Pseudonymen vorsieht. Ein Rückschluss auf die Person ist dann nicht mehr möglich.

3 Funktionalitäten

3.1 Was leistet der Dienst

- Generierung von Pseudonymen (PSN)
- Zuordnung von Pseudonymen zu beliebigen Originalwerten
- Technische Anonymisierung durch Löschung von Zuordnungen zwischen Pseudonym und Originalwert
- Konfiguration von Pseudonymparametern: Prüzfifferalgorithmus, Länge, Alphabet
- Verwaltung von Pseudonymdomänen 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
- Health Checks
- Unterstützung für KeyCloak-Authentifizierung und Autorisierung

3.2 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

4 Installation

4.1 Systemanforderungen

Technisch

- Windows oder Ubuntu Server (oder vergleichbar)²
- Empfohlen: Mindestens 8 GB Arbeitsspeicher³
- Empfohlen: Mindestens 5 GB Festplattenspeicher
- Installierte aktuelle Version von *Docker*⁴ und Docker Compose⁵
- Administrative Rechte
- Keine Nutzungsbeschränkungen auf die bereitgestellten Service- und Client-URLs

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

4.2 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 12** berücksichtigt werden. In der ausgelieferten Docker-Variante sind diese bereits berücksichtigt und es sind keine weiteren Anpassungen erforderlich.

¹ Nutzen Sie hierzu bitte ggf. den E-PIX: <http://ths-greifswald.de/e-pix>

² Beim Betrieb unter Windows ist zu beachten, dass bei der Verwendung von *Volumes* und parallel betriebenen VPN-Clients Probleme auftreten können.

³ Zur Verwaltung von bis zu 4 Millionen Pseudonymen genügen 4 GB Arbeitsspeicher von denen etwa 2 GB durch die Datenbank (MySQL) reserviert werden.

⁴ Weitere Informationen unter <https://docs.docker.com/install/>

⁵ Weitere Informationen unter <https://docs.docker.com/compose/install/>

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

Der gPAS® benötigt zur Ausführung mehrere Container (vgl. **Abbildung 4-1**). Damit diese nicht einzeln gestartet werden müssen und entsprechend zusammengeschaltet werden müssen, wird der Dienst mit Docker-Compose gestartet. Alle hierzu benötigten Ressourcen werden im öffentlichen Repository des MOSAIC-Projekts bereitgestellt⁶.

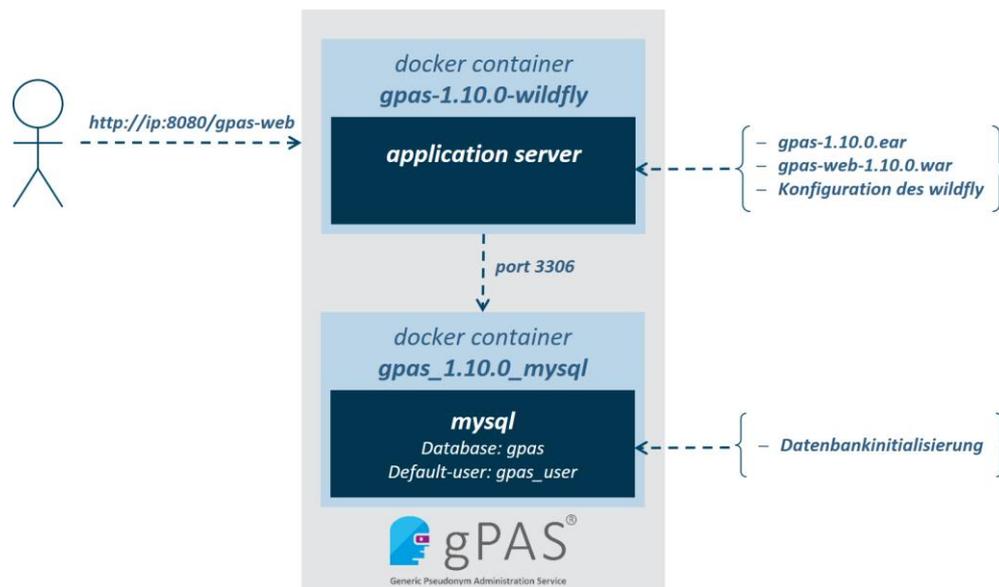


Abbildung 4-1: Architektur des gPAS® am Beispiel der Version 1.10.0.

Das entstehende Docker-System besteht aus getrennten Containern für die erforderliche Datenbankinstanz (MySQL) und den benötigten Anwendungsserver (Wildfly inkl. Datenbank-Konnektoren) zur Verfügbarmachung der Programmdateien. Beide Container kommunizieren intern über den MySQL-Port 3306. Der Zugriff auf das System von „außen“ erfolgt über den Web-Browser. Die Inhalte werden über den Port 8080 (gPAS®) für den Anwender bereitgestellt.

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

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

Download der benötigten Dateien

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

Vergabe Schreibrechten

```
sudo chmod -R 777 gPAS/docker/standard/deployments
```

⁶ Weitere Informationen unter <https://github.com/mosaic-hgw/gPAS>
Version 1.10.2 vom 11.03.2021

Wechseln in das gPAS®-Verzeichnis

```
cd gPAS/docker/standard
```

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

```
sudo service mysql stop
```

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

```
sudo docker -v
```

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

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

Starten des gPAS® mithilfe von Docker Compose

```
sudo docker-compose up
```

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

```
Wildfly 18.0.1.Final (Wildfly Core 5.0.0.Final) started in 63373ms -  
Started 824 of 1024 services (324 services are lazy, passive or on-demand)
```

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

5 Die grafische Benutzeroberfläche des gPAS®

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 an typischen Arbeitsabläufen eines Datentreuhänders. Die daraus resultierenden Anwendungsfälle umfassen:

1. Anlegen einer Domäne
2. Generieren von Pseudonymen
3. Originalwerte und Pseudonyme suchen
4. Depseudonymisierung
5. Technische Anonymisierung (Virtuelle Anonymisierung)
6. Löschen von Pseudonymen
7. Integration von Alt-Pseudonymen
8. Anzeige von Pseudonym-Hierarchien
9. Listenverarbeitung

10. Statistik ermitteln
11. Domäne bearbeiten oder löschen
12. Pseudonyme exportieren

5.1 Anwendungsfall 1: Anlegen einer Domäne

Um ein neues Pseudonym zu erstellen, muss vorab eine entsprechende **Domäne** vorhanden sein.

Dabei gilt:

- ein Pseudonym ist innerhalb einer Pseudonymdomäne eineindeutig
- eine Domäne kann für ein Projekt oder ein eingebettetes Studienvorhaben stehen, aber auch zur Beherbergung von Zweitpseudonymen angelegt werden
- für jede Domäne lassen sich eigene Pseudonymparameter festlegen

Unter dem Menüpunkt *Domänen* werden alle bereits angelegten Domänen als Baum-Struktur dargestellt (vgl. **Abbildung 5-1**). Mit einem Rechtsklick auf einen Eintrag werden weitere Optionen zur jeweiligen Domäne angeboten. So kann beispielsweise direkt die Seite *Suchen / Anlegen* 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 *Nebengeordnete* oder *Untergeordnete Domäne* angelegt werden. Daraufhin öffnet sich ein Fenster mit den auszufüllenden Feldern. Bei einer *Nebengeordneten Domäne* wird dieselbe *Übergeordnete Domäne* gesetzt, wie bei der ausgewählten Domäne. Bei einer *Untergeordneten Domäne* wird die angewählte Domäne als *Übergeordnete Domäne* der neuen Domäne gesetzt.

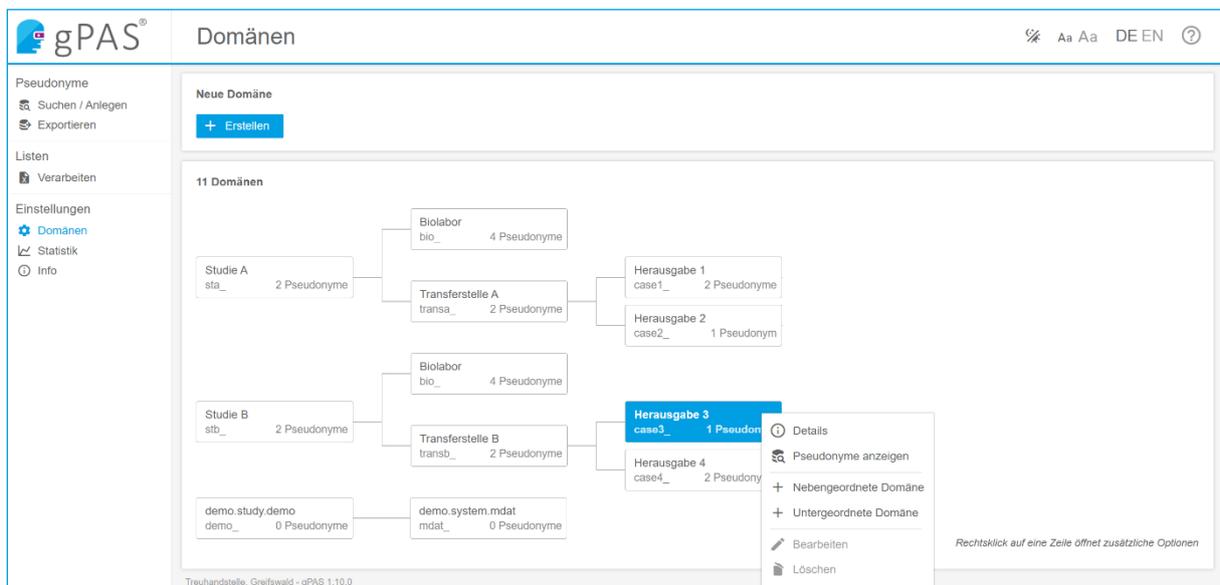


Abbildung 5-1: Oberfläche zum Anzeigen aller Domänen. Der Baum zeigt die hierarchische Struktur der Domänen. Das Kontextmenü enthält weitere Optionen.

Beim Anlegen einer neuen Domäne ist der eindeutige **Name** festzulegen. Optional kann eine oder mehrere **Übergeordnete Domänen** (um Pseudonymhierarchien zu realisieren) angegeben werden.

Eine Domäne mit mehreren *Übergeordneten Domänen* wird entsprechend oft im jeweiligen Domänen-Baum dargestellt. Zusätzlich kann eine **Beschreibung** ergänzt werden.

Das zu nutzende **Alphabet** (vgl. **Tabelle 5-15-15-1**) und der jeweilige **Prüfziffern-Generator** (vgl. **Tabelle 5-25-25-2**) können in sinnvollen Kombinationen ausgewählt werden. Nur erlaubte Kombinationen werden dabei angeboten. Die Oberfläche zum Anlegen einer neuen Domäne ist in **Abbildung 5-2** dargestellt.

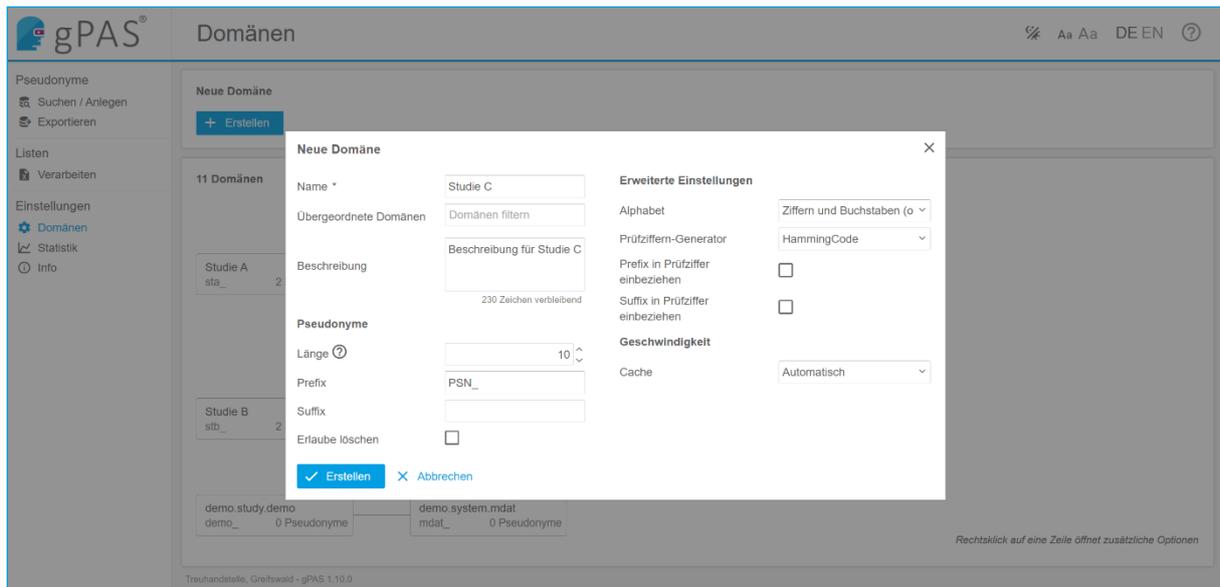


Abbildung 5-2: Oberfläche zum Anlegen einer neuen Domäne.

Mögliche Alphabete sind in **Tabelle 5-15-15-1** aufgelistet. Alternativ kann ein eigenes Alphabet (aus beliebigen Zeichen bis auf Kommas und Leerzeichen) definiert werden. Hierzu wählt man im Alphabet Auswahlmenu den Eintrag *Benutzerdefiniert* und trägt die gewünschten Zeichen in das erscheinende Textfeld ein (vgl. **Abbildung 5-2**).

Tabelle 5-1: Bereitgestellte Alphabete im gPAS®.

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
Symbols32	32 Zeichen: 0-9, A-Z (ohne B, I, O, S – wegen der Ähnlichkeit zu 8, 1, 0, 5)
Symbols31	Wie Symbol32, nur ohne V – wegen der Ähnlichkeit zu U

Dabei ist zu beachten, dass einzelne Prüfzeichenalgorithmen nur unter bestimmten Bedingungen angewendet werden können. Die entsprechenden Bedingungen der jeweiligen Prüfzifferalgorithmen

sind in **Tabelle 5-25-25-2** dargestellt. Sollen keine Prüfziffern enthalten sein, wird die Option *Keine Prüfziffern* angewählt.

Tabelle 5-2: Mögliche Prüfziffern-Generatoren und dessen Bedingungen.

Prüfzifferalgorithmus	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

Ergänzend können bei jeder Domäne noch weitere Eigenschaften definiert werden. Diese sind mit entsprechenden Beispielen in **Tabelle 5-35-35-3** dargestellt.

Tabelle 5-3: Mögliche Domain Properties.

Feld (SOAP-Parameter)	Bedeutung	Beispiel
Anzahl detektierbarer Fehler (<i>MAX_DETECTED_ERRORS</i>)	Max. Anzahl fehlerhafter Zeichen, die auch sicher als fehlerhaft erkannt werden (nur bei Reed-Solomon)	2
Länge (<i>PSN_LENGTH</i>)	Länge des erzeugten Pseudonyms ohne Prüfzeichen (max. 49)	10
Präfix (<i>PSN_PREFIX</i>)	Zeichenfolge, die als Präfix genutzt werden soll (max. 20 Zeichen)	abc_
Suffix (<i>PSN_SUFFIX</i>)	Zeichenfolge, die als Suffix genutzt werden soll (max. 20 Zeichen)	_xyz
Präfix in Prüfziffernberechnung einbeziehen (<i>INCLUDE_PREFIX_IN_CHECK_DIGIT_CALCULATION</i>) ↔	Einschließen des Präfix in die Berechnung der Prüfziffern	ja (true)/ nein (false)
Erlaube löschen (<i>PSNS_DELETABLE</i>)	Pseudonyme innerhalb der Domäne sind löschar (z.B. zur Erstellung von temporären Pseudonymen, der Standard-Wert ist nein(false).)	ja (true)/ nein (false)

Pseudonym Cache (<i>FORCE_CACHE</i>)	ON = Immer an OFF = Immer aus DEFAULT = An, wenn Alphabet + PSN Länge klein genug, dass 120 MB Cache unterschritten werden. ⁷	ON, OFF, Automatisch (DEFAULT)
Abstand für Delimiter bzw. Trennzeichen (<i>useLastCharAs</i> ↔ <i>DelimiterAfterXChars</i>)	Zahlenwert im Intervall [0, 50[. Wenn größer 0, dann gibt dies den Abstand an, in der das Trennzeichen (letztes Zeichen im benutzerdefinierten Alphabet) im Pseudonym gesetzt wird. Bei 0 wird kein Trennzeichen gesetzt.	Wenn z.B. der Wert 3 ist: 123-456-789 („-“ ist das Trennzeichen)

Das **Löschen einer Domäne** ist nur möglich, sofern keine Pseudonyme für diese Domäne hinterlegt sind.

Hingegen ist das **Löschen eines Pseudonyms** möglich, wenn die Domänenkonfiguration dies gestattet (*Erlaube löschen* unter dem Punkt *Pseudonyme* (*PSNS_DELETABLE=TRUE*)).

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

5.2 Anwendungsfall 2: Generieren von Pseudonymen

Generieren eines einzelnen Pseudonyms

Unter dem Menüpunkt *Suchen / Anlegen* können einzelne Pseudonyme angelegt werden. Hierzu muss zunächst die Schaltfläche *Neues Pseudonym generieren* ausgewählt werden. 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 5-3** wird die Oberfläche zum Anlegen eines Pseudonyms dargestellt.

⁷ Memory consumption is one bit per possible pseudonym: $\text{mem_for_cache} = \text{alphabet_length} \wedge \text{pseudonym_length} / 8 / 1024 / 1024$ MB, e.g. alphabet = numbers, length = 8
-> $\text{mem_for_cache} = 10 \wedge 8 / (8 * 1024 * 1024) = 11.92$ MB

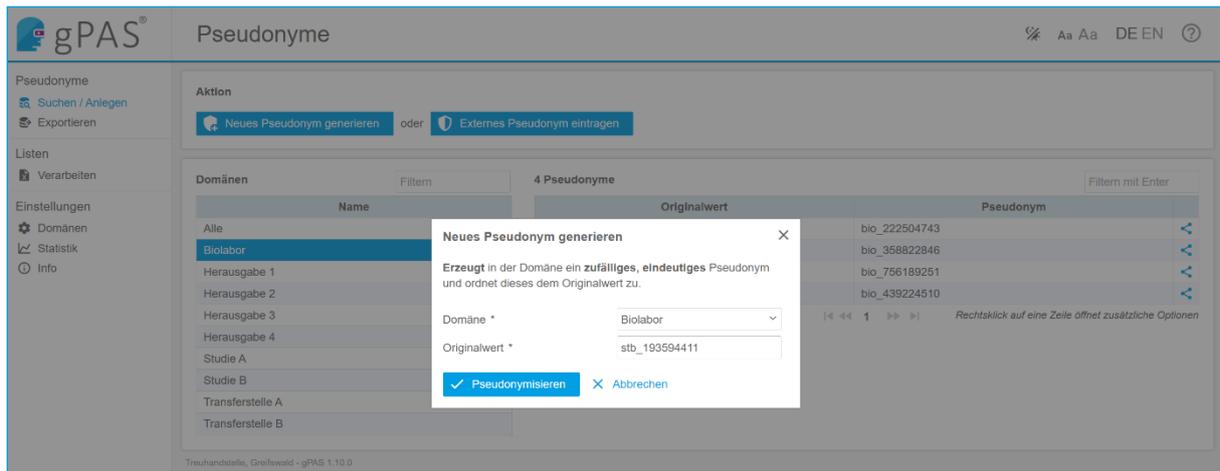


Abbildung 5-3: Oberfläche zum Anlegen eines neuen Pseudonyms.

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.

Bei einem Pseudonym höherer Stufe wird als Originalwert ein zuvor generiertes Pseudonym verwendet. Um ein Zweit-Pseudonym zu generieren, kann wie eben beschrieben vorgegangen oder das Kontextmenü per Rechtsklick genutzt werden. Über die Schaltfläche *Nebengeordnetes Pseudonym* wird ein Pseudonym derselben Stufe generiert. Über die Schaltfläche *Untergeordnetes Pseudonym* wird das Pseudonym als Originalwert verwendet und es wird ein Pseudonym höherer Stufe erzeugt. In **Abbildung 5-4** wird das entsprechende Kontextmenü dargestellt.

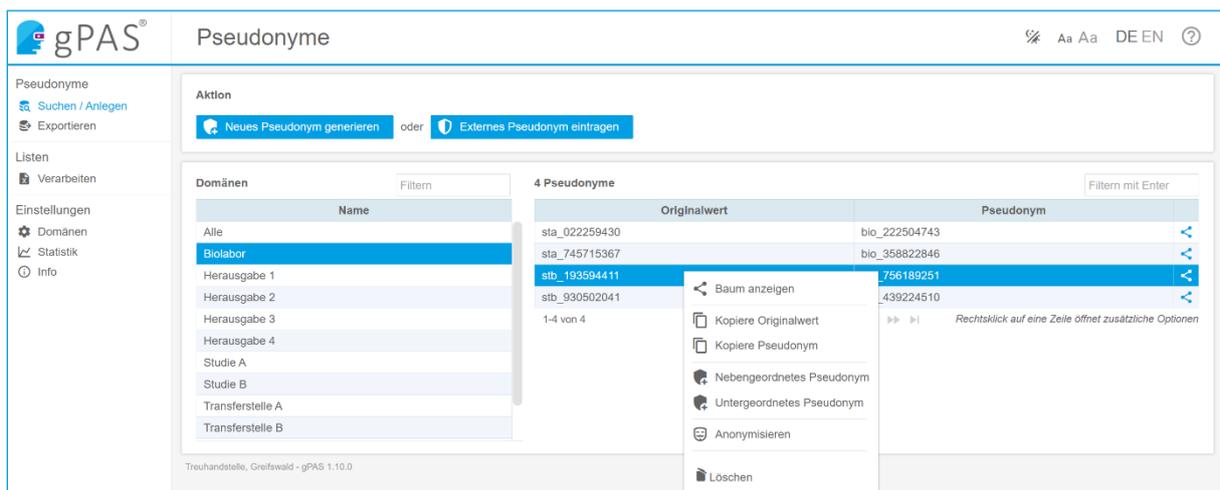


Abbildung 5-4: Kontextmenü zum Erzeugen von nebengeordneten oder übergeordneten Domänen.

In **Abbildung 5-5** 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 könnte beispielsweise ebenfalls ein Master Patient Index 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.

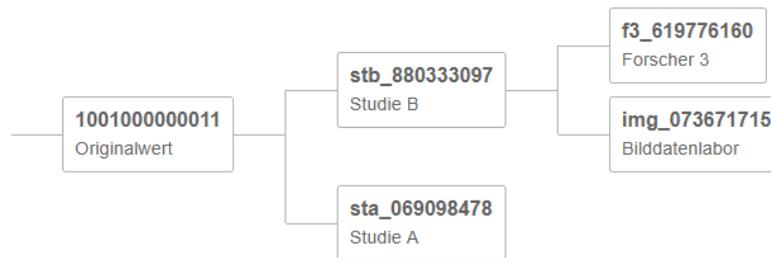


Abbildung 5-5: Exemplarische Struktur bei mehreren Pseudonymen und Stufen für einen Studienteilnehmer.

5.3 Anwendungsfall 3: Originalwerte und Pseudonyme suchen

Originalwerte und Pseudonyme werden auf die gleiche Weise gesucht. Hierzu wird unter dem Menüpunkt *Suchen / Anlegen* eine Domäne angewählt. 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 5-6** ist die Oberfläche zum Suchen von Originalwerten und Pseudonymen dargestellt.

Domänen	Name	Originalwert	Pseudonym
Alle		stb_193594411	bio_756189251
Biolabor		stb_930502041	bio_439224510
Herausgabe 1			
Herausgabe 2			
Herausgabe 3			
Herausgabe 4			
Studie A			
Studie B			
Transferstelle A			
Transferstelle B			

Abbildung 5-6: Oberfläche zum Suchen von Originalwerten oder Pseudonymen.

Eine Suche über alle Domänen hinweg, kann durchgeführt werden, indem in der Domänenauflistung 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.

5.4 Anwendungsfall 4: Suchen von Originalwerten (Depseudonymisierung)

Eine Depseudonymisierung entspricht der Suche eines Originalwerts anhand eines gegebenen Pseudonyms. Es wird demnach so vorgegangen, wie im **Abschnitt 5.3** beschrieben. Hierzu wird das vorhandene Pseudonym in der jeweiligen Domäne gesucht. Die Suche liefert den dazugehörigen Originalwert bzw. im Fall einer Pseudonymhierarchie das Pseudonym geringerer Stufe.

5.5 Anwendungsfall 5: Technische Anonymisierung (Virtuelle Anonymisierung)

Bei der technischen Anonymisierung (bekannt auch 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. Zur Anonymisierung wird hierzu unter dem Menüpunkt *Suchen / Anlegen* die entsprechende Domäne gewählt. Danach wird in der Liste das zu anonymisierende Pseudonym gewählt (zuvor kann mittels Suche die Liste gefiltert werden, vgl. **Abschnitt 5.3**). Mit einem Rechtsklick auf den entsprechenden Eintrag wird das Kontextmenü geöffnet (siehe **Abbildung 5-7**).

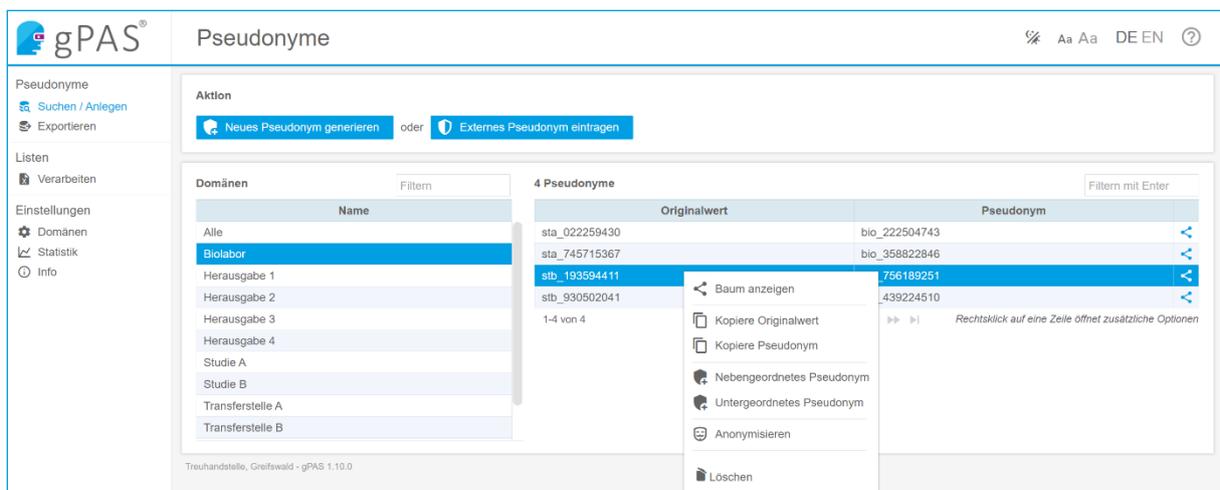


Abbildung 5-7: Mit einem Rechtsklick auf den Eintrag, kann das Kontextmenü aufgerufen werden.

Mit dem Auswählen des Eintrags *Anonymisieren*, wird die Zuordnung aufgehoben. Bevor dieser unumkehrbare Vorgang ausgeführt wird, muss dies bestätigt werden (siehe **Abbildung 5-8**).

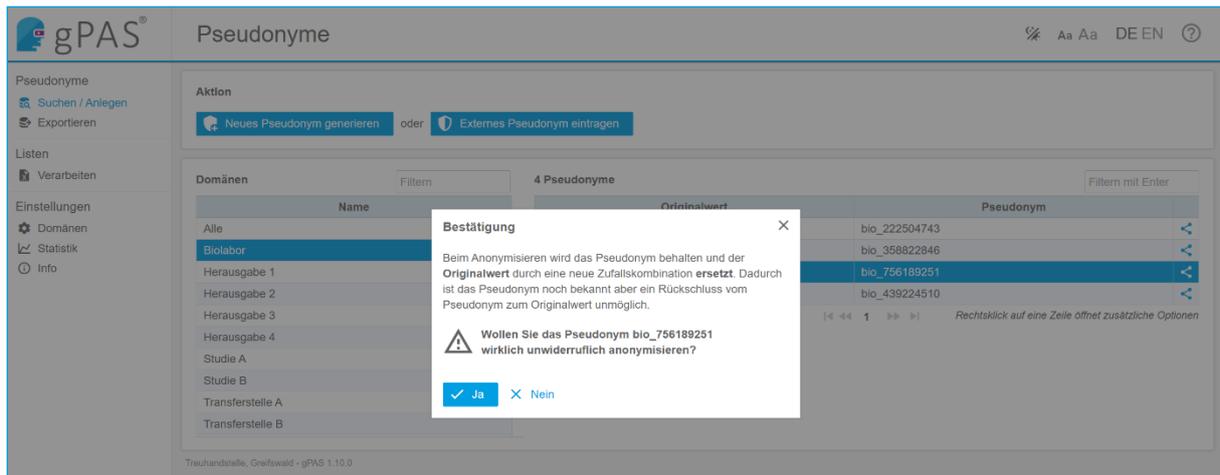


Abbildung 5-8: Der unumkehrbare Vorgang der Anonymisierung muss nochmals bestätigt werden.

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 5-9**).

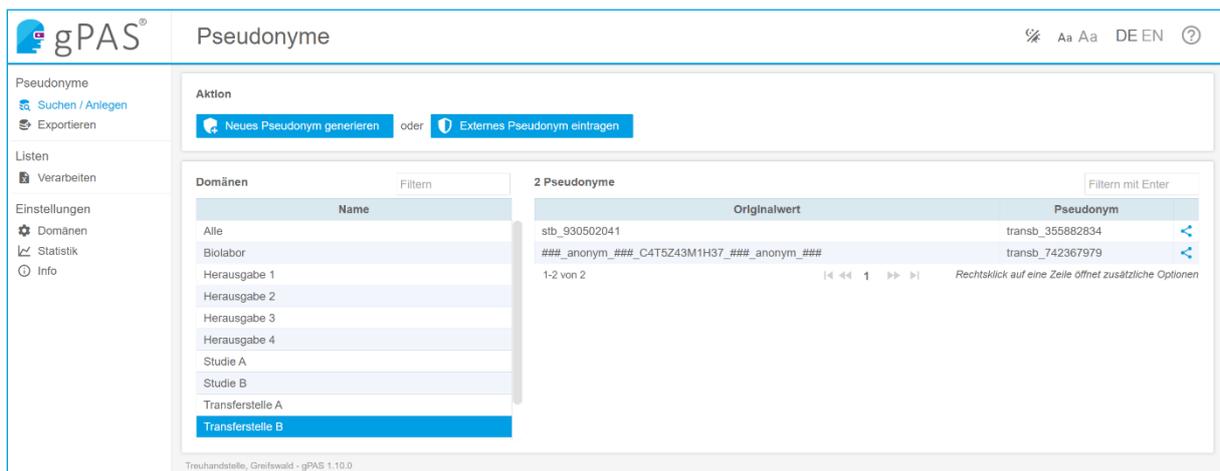


Abbildung 5-9: Anonymisierter Eintrag.

5.6 Anwendungsfall 6: Löschen von Pseudonymen

Beim Löschen eines Pseudonym-Paares, wird der Originalwert und das entsprechende 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 Pseudonyms nicht mehr möglich. Um ein Pseudonym löschen zu können, muss eine Domäne das zulassen und entsprechend konfiguriert sein (vgl. **Tabelle 5-35-35-3** in **Abschnitt 5.1**). Zum Löschen wird unter dem Menüpunkt *Suchen / Anlegen* die entsprechende Domäne und das zu löschende Pseudonym angewählt. Mit einem Rechtsklick auf den Eintrag wird das Kontextmenü geöffnet (siehe **Abbildung 5-7**). Mit dem Auswählen des Eintrags

Löschen, wird der Eintrag gelöscht. Bevor dieser unumkehrbare Vorgang ausgeführt wird, muss dies bestätigt werden.

5.7 Anwendungsfall 7: Integration von Alt-Pseudonymen

Um ein Pseudonym anzulegen, welches als Originalwert eines bereits existenten Pseudonyms dient, muss unter dem Menüpunkt *Suchen / Anlegen* die Schaltfläche *Von Extern 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.

5.8 Anwendungsfall 8: Anzeige von Pseudonym-Hierarchien

Wurden bei der Konfiguration der Pseudonym-Domäne eine Eltern-Domäne zur Angabe der Beziehung zweier Domänen (Eltern-Kind-Beziehung, 1:N möglich) angegeben, so kann der dadurch entstehende Pseudonymbaum übersichtlich dargestellt werden. Hierzu wird unter dem Menüpunkt *Suchen / Anlegen* die Domäne und ein Pseudonym gewählt. Die entsprechende Hierarchie wird in einem Fenster dargestellt. Dabei ist die angewählte Domäne und Pseudonym farblich hervorgehoben. In **Abbildung 5-10** ist die entsprechende Oberfläche dargestellt.

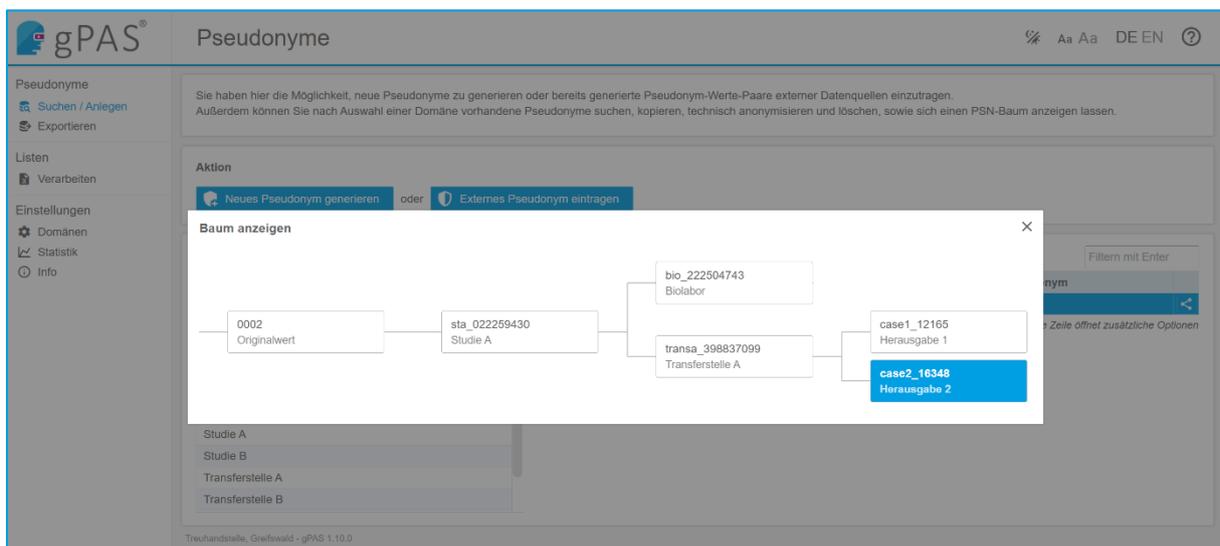


Abbildung 5-10: Oberfläche beim Anzeigen der Pseudonymhierarchie für ein selektiertes Pseudonym (blau dargestellt).

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

5.9 Anwendungsfall 9: Listenverarbeitung

Es ist möglich, eine Liste von Eingabewerten zu pseudonymisieren, zu depseudonymisieren (ermitteln des Originalwerts), zu anonymisieren oder zu löschen. Hierzu kann unter dem Menüpunkt *Verarbeiten* eine CSV-Datei ausgewählt werden. In **Abbildung 5-11** ist die entsprechende Oberfläche abgebildet.

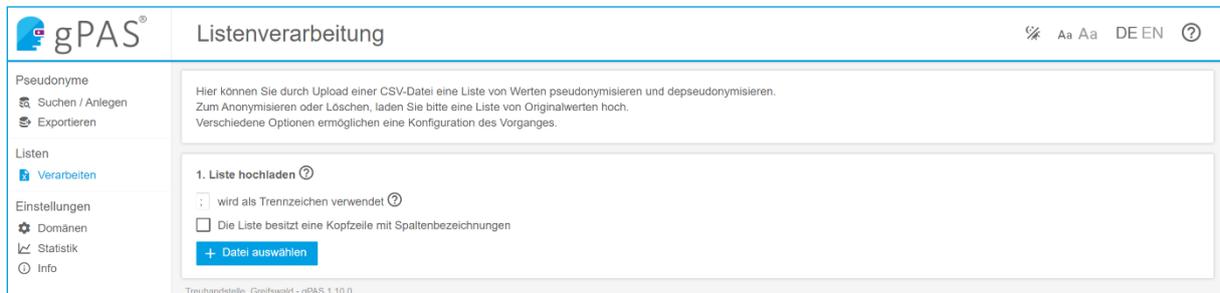


Abbildung 5-11: Oberfläche zum Verarbeiten von Listen.

Ist eine Überschrift enthalten, so kann dies mittels Anwählen der Checkbox *Die Liste besitzt eine Kopfzeile mit Spaltenbezeichnung* mitgeteilt werden. In diesem Fall wird die Kopfzeile/erste Zeile nicht mitverarbeitet. Wenn eine mehrspaltige Tabelle enthalten ist, erfolgt eine Separierung der Spalten standardmäßig mit einem Semikolon. Soll ein eigenes Trennzeichen verwendet werden, so kann dies entweder mittels `sep=X` (X steht für das verwendete Trennzeichen) in der ersten Zeile der CSV-Datei definiert werden oder in dem entsprechenden Feld das Trennzeichen definiert werden. 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 5-12**).

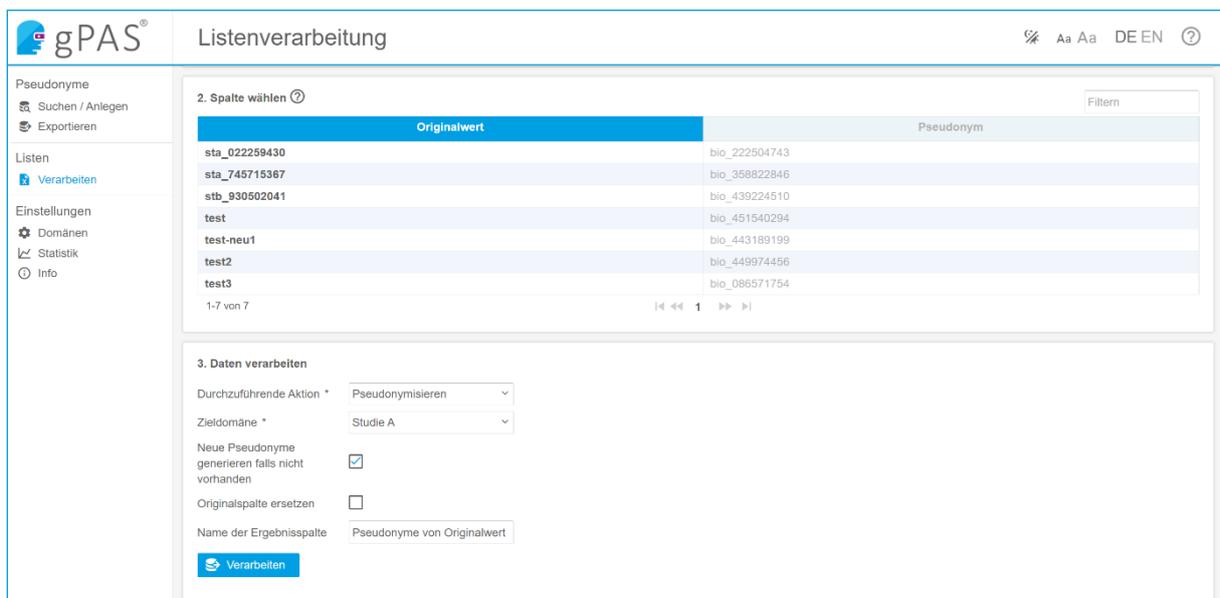


Abbildung 5-12: Wählen der Verarbeitungsoperation. Hier am Beispiel von *Pseudonymisieren*.

Es kann zwischen vier Verarbeitungsoperationen gewählt werden. Die entsprechenden Optionen sind in **Tabelle 5-45-45-4** aufgelistet.

Tabelle 5-4: Mögliche Verarbeitungsoperationen.

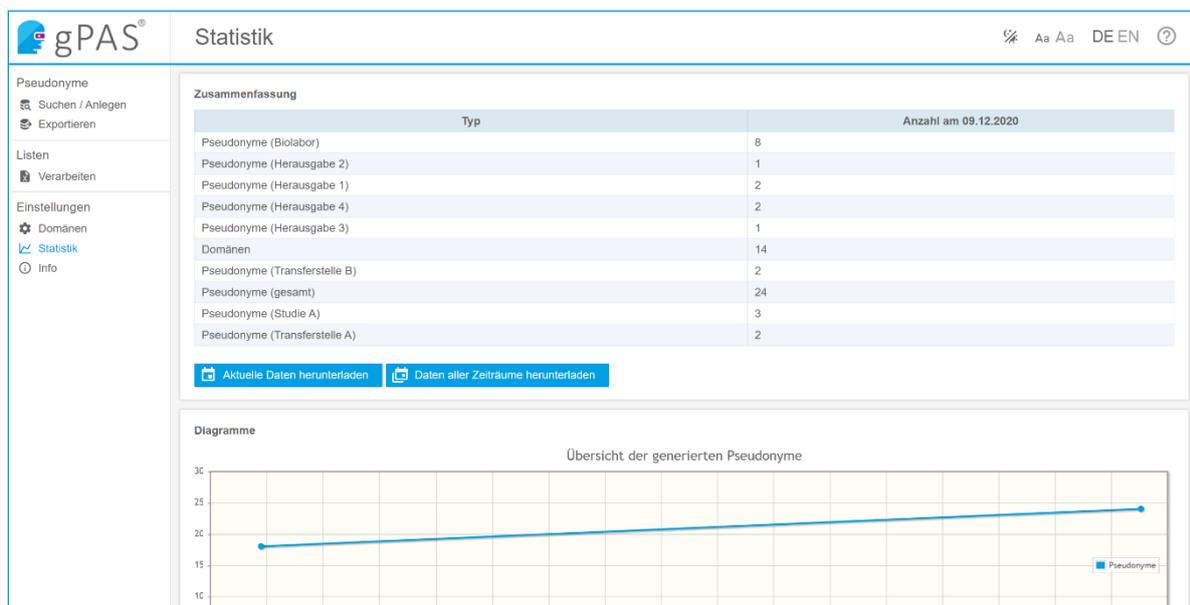
Operation	Beschreibung
Pseudonymisieren	Für jedes Element der Liste ein neues Pseudonym erzeugen (sofern noch nicht bekannt).
Depseudonymisieren	Für jedes Pseudonym der Liste die Originalwerte ermitteln (sofern bekannt).
Anonymisieren	Jedes Pseudonym der Liste wird anonymisiert.
Löschen	Jedes Pseudonym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

Die zu nutzende Pseudonym-Domäne muss angegeben werden.

Nach dem Wählen der Schaltfläche *Verarbeiten* wird das Ergebnis der Verarbeitung in die dargestellte Tabelle ergänzt. Hierfür kann die Ergebnisspalte benannt werden. Die aktualisierte Liste kann im Anschluss als CSV-Datei heruntergeladen werden.

5.10 Anwendungsfall 10: Statistik ermitteln

Die Anzahl der vorhandenen Pseudonyme, Anonyme und Pseudonymdomänen im gPAS® können angezeigt werden und als Ergebnis im CSV- oder PDF-Format exportiert werden. Hierzu kann unter dem Menüpunkt *Statistik* eine Tabelle eingesehen werden, welche die jeweiligen Daten auflistet. In **Abbildung 5-13** ist die Oberfläche der Statistik abgebildet.

**Abbildung 5-13:** Oberfläche zum Einsehen von der Anzahl von Pseudonymen, Anonymen und Domänen.

5.11 Anwendungsfall 11: 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 5-14**).

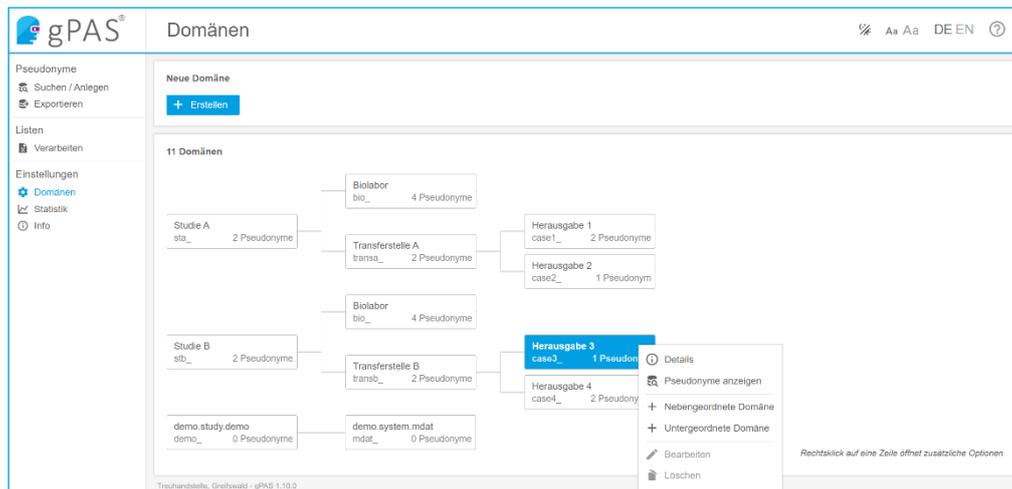


Abbildung 5-14: Kontextmenü mit den Schaltflächen zum Anzeigen der Domänendetails, zum Bearbeiten der Domäne und zum Löschen der Domäne. Hierüber können weitere Domänen erzeugt werden (s. **Anwendungsfall 1: Anlegen einer Domäne**).

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

⚠ Hinweis: Wird der Name 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.

5.12 Anwendungsfall 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. Mit einem Klick auf die Schaltfläche *CSV herunterladen* wird der Export gestartet. Die resultierende Datei kann ohne weitere Anpassungen beispielsweise wieder importiert werden. In **Abbildung 5-15** ist die entsprechende Oberfläche dargestellt.

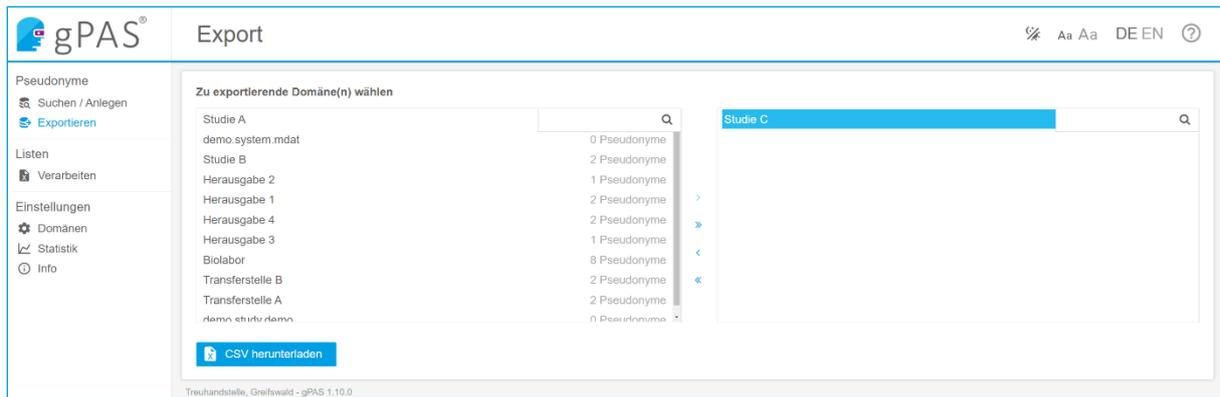


Abbildung 5-15: Oberfläche zum Exportieren beliebiger Domänen.

6 Logging

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

7 Authentifizierungs- und Autorisierung

Die bereitgestellte gPAS-Version (ab 1.10.0) bietet unterschiedliche Umsetzungsoptionen der Authentifizierungs- und Autorisierung sowohl in der **Docker-** als auch in der **Docker-Compose-**Variante.

In der Standard-Ausgabe vom gPAS ist keine Authentifizierung notwendig. Möchte man den gPAS jedoch nur für bestimmte **Nutzergruppen (Admin-Nutzer, Standard-Nutzer)** zugänglich machen (vgl. Tabelle 7-1), oder sogar das Anlegen von neuen Domänen beschränken, können zwei Authentifizierungsverfahren angewendet werden. **gRAS** und **keycloak**, wobei es für KeyCloak zwei verschiedene Varianten gibt.

⚠ Hinweis: Die Auswahl der einzelnen Varianten erfolgt in der Docker-Compose Version innerhalb der `docker-compose.yml`. Details für die notwendige Anpassung der Docker-Konfiguration können der beigelegten Beschreibungen <https://www.ths-greifswald.de/ttp-tools/keycloak> sowie `docker-compose/README.md`.

Tabelle 7-1: Nutzer der Gruppe **Admin** und **User** haben unterschiedliche Zugriffsrechte in der Web-Oberfläche.

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

7.1 Verwendung von KeyCloak

Details zur Vorbereitung des KeyCloak-Servers sind in der beigelegten Beschreibung **KeyCloak-Einrichtung.md** zu finden.

Die Client-seitige KeyCloak-Konfiguration kann sowohl per Config-Datei als auch per Environment-Variablen bei Start des Docker-Compose erfolgen. Details sind in **docker-compose/README.md** beschrieben.

7.2 Verwendung von gRAS

Die gRAS Authentifizierungs- und Autorisierungsmechanismen basiert auf MySQL und JAXWS per Wildfly-Konfiguration. Nutzer müssen sich für die Nutzung des gPAS mittels Benutzernamen (user@ths) und Passwort einloggen. Je nach zugeordneter Rolle enthalten sie unterschiedliche Berechtigungen. Eine Übersicht der Rollen und Rechte ist in der vorstehenden **Tabelle 6-16-1** zu finden.

Bei der Installation des gPAS mittels Docker werden standardmäßig ein Admin-Nutzer „admin“, sowie ein Standard-Nutzer „employee“ angelegt. Die Anmeldung erfolgt unter Angabe der Domäne „THS“.

```
[Default-Admin] username="admin@ths", password="ttp-tools"
[Default-User]  username="user@ths", password="ttp-tools"
```

Die Autorisierung kann werkzeübergreifend erfolgen. So kann ein und dieselbe Person z.B. beim E-PIX Standard-Nutzer und beim gPAS Admin-Nutzer sein.

7.2.1 Verwaltung der Nutzer, Rollen und Rechte mittels MySQL und Docker EXEC

Die Verwaltung der Nutzer (anlegen, aktivieren, deaktivieren, Passwort ändern) und das Zuweisen von Rechten kann direkt in der entsprechenden Datenbank (gRAS) oder per *Docker EXEC* auf dem entsprechenden Datenbank-Container (hier: gPAS-1.10.0-mysql) erfolgen.

Die Verwaltung erfolgt auf Basis entsprechender MySQL-Prozeduren. Diese werden im Zuge der Docker-Installation der Web-Auth-Version für gPAS automatisch angelegt.

7.2.1.1 Nutzer anlegen

⚠ Hinweis: Das Nutzer-Passwort wird zwar im Klartext angegeben, die Prozedur jedoch erzeugt automatisch einen SHA-256-Hash, der in der gRAS-Datenbank gespeichert wird.

in Docker

```
docker exec -it gPAS-1.10.0-mysql mysql -ugras_user -pgras_password  
-e "use gras;call createUser('BENUTZERNAME','PASSWORT','KOMMENTAR');"
```

in SQL

```
use gras;  
call createUser("BENUTZERNAME","PASSWORT","KOMMENTAR");
```

7.2.1.2 Rechtevergabe

Nutzer haben standardmäßig keine Berechtigungen. Die Vergabe dieser erfolgt je Werkzeug („Projekt“) unter der Angabe von *epix*, *gpas* oder *gics* (Schreibweise wie hier dargestellt).

7.2.1.2.1 Admin-Rechte vergeben

⚠ Hinweis: Admin-Rechte beinhalten die Standard-Rechte.

in Docker

```
docker exec -it gPAS-1.10.0-mysql mysql -ugras_user -pgras_password  
-e "use gras;call grantAdminRights('PROJEKTNAME','BENUTZERNAME');"
```

in SQL

```
use gras;  
call grantAdminRights("gPAS","BENUTZERNAME");
```

7.2.1.2.2 Standard-Rechte vergeben

in Docker

```
docker exec -it gPAS-1.10.0-mysql mysql -ugras_user -pgras_password  
-e "use gras;call grantStandardRights('PROJEKTNAME','BENUTZERNAME');"
```

in SQL

```
use gras;
call grantStandardRights("epix","BENUTZERNAME");
```

*Passwort ändern***in Docker**

```
docker exec -it gPAS-1.10.0-mysql mysql -ugras_user -pgras_password -e "use
gras;call changePassword('BENUTZERNAME','NEUES_PASSWORT');"
```

in SQL

```
use gras;
call changePassword("BENUTZERNAME","NEUES_PASSWORT");
```

7.2.1.4 Nutzer aktivieren/deaktivieren

⚠ Hinweis: Diese Änderungen werden erst nach einem Neustart des *Wildfly* übernommen.

in Docker

```
docker exec -it gPAS-1.10.0-mysql mysql -ugras_user -pgras_password -e "use
gras;call disableUser('BENUTZERNAME');"
```

in SQL

```
call disableUser("BENUTZERNAME"); -- Benutzer deaktivieren
call enableUser("BENUTZERNAME"); -- Benutzer aktivieren
```

7.2.1.5 Erweiterte Rechte-Konfiguration**7.2.1.5.1 Domain anlegen**

⚠ Hinweis: Die Authentifizierungsdomain ist unabhängig von den Pseudonymisierungsdomänen.

Innerhalb der gRAS-Authentifizierung wird **per Default die Parent Domain „ths“ genutzt**, um Anmeldung von Nutzern für eine ausgewählte Authentifizierungsdomain (user@domain) zu ermöglichen. Die Individualisierung der Domänen ist möglich.

in SQL

```
use gras;
call createDomain('domainName', 'parent domain');
```

7.2.1.5.2 Projekt anlegen

Innerhalb der gRAS-Authentifizierung wird mit dem Projekt das abzusichernde Werkzeug (konkret hier: gPAS) bezeichnet. **Per Default wird hier entsprechend das Projekt „gpas“ genutzt**. Die Individualisierung des Projektnamens ist möglich.

in SQL

```
use gras;
call createProject('domainName', 'parent domain');
```

7.2.1.5.3 Gruppe anlegen

Rechte werden stets nur Gruppen zugeordnet. Per Default werden für die Authentifizierung per gRAS im gPAS-Frontend die Gruppen **gPAS-users** und **gPAS-admins** genutzt. Anpassungen und Ergänzungen sind möglich.

in SQL

```
-- createGroup(<projectName>, <groupName>, <description>)
CALL createGroup('gpas','gpas-users','this group is for users with basic
right');
CALL createGroup('gpas','gpas-admins','this group is for users with
extended right');
```

7.2.1.5.4 Rolle anlegen

Je Projekt/Werkzeug kann ein-und-derselbe Nutzer unterschiedliche Rollen einnehmen. Per Default werden für die Authentifizierung per gRAS im gPAS-Frontend die Rollen **role.gpas.user** und **role.gpas.admin** genutzt. **Diese Rollen sind fest den jeweiligen Unterseiten des gPAS-Frontends zugeordnet und können nicht geändert werden.** Das Ergänzen zusätzlicher Rollen ist möglich.

in SQL

```
-- createRole(<projectName>, <roleName>, <description>)
CALL createRole('gpas','role.gpas.user','gPAS userspace');
CALL createRole('gpas','role.gpas.admin','gPAS adminsace');
```

7.2.1.5.5 Zuordnung von Gruppen und Rollen

Die finalen Rechte erhält jeder Nutzer durch die Zuordnung von Gruppen (sind in der gRAS-Datenbank per Default in ADMIN und USER unterteilt) und Rollen (sind im Frontend definiert und eingebettet). Das Ergänzen zusätzlicher Zuordnungen ist möglich.

in SQL

```
-- createGroupRoleMapping(<projectName>, <groupName>, <roleName>)
CALL createGroupRoleMapping('gpas','gPAS-users','role.gpas.user');
CALL createGroupRoleMapping('gpas','gPAS-admins','role.gpas.admin');
```

8 Empfehlungen zur Absicherung des Anwendungsservers

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 Key-Cloak**

- Zugangsbeschränkung auf URL-Ebene mit Basic Authentication (z.B. mit NGINX oder Apache)

9 Nutzung der SOAP-Schnittstelle

Neben der grafischen Benutzerschnittstelle, stehen maschinenverständliche Web-Schnittstellen für den Regelbetrieb (Pseudonymisieren, Depseudonymisieren, Anonymisieren, Validieren, Strukturdarstellung) und für die Administration (Anlegen, Lesen und Löschen von Domänen) von gPAS® zur Verfügung. Diese können mit dem SOAP-Protokoll angesprochen werden. Beim laufenden Dienst kann die Definition der SOAP-Schnittstelle mit den folgenden Pfaden, an der jeweiligen URL, abgerufen werden.

Regelbetrieb

```
http://example.org:8080/gpas/gpasService?wsdl
```

Administration

```
http://example.org:8080/gpas/DomainService?wsdl
```

Die neueste Entwicklerdokumentation ist unter der folgenden URL zu finden.

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

9.1 Anlegen einer Domäne

Im Folgendem wird das Anlegen einer Domäne über die `DomainService`-Schnittstelle exemplarisch vorgestellt. In **Abbildung 9-1** ist die entsprechende Anfrage für das Anlegen einer Domäne dargestellt. Mittels des Parameters `name` wird ein eindeutiger Name der Domäne festgelegt. Der Prüfzifferalgorithmus wird mit dem Parameter `checkDigitClass` angegeben. Die verfügbaren Algorithmen können der **Tabelle 5-25-25-2** entnommen werden. Dabei muss vor den Namen des jeweiligen Algorithmus Paketpfad `org.emau.icmvc.ganimed.ttp.psn.generator` vorangestellt werden. Mit dem Parameter `alphabet` wird das verwendete Alphabet angegeben. Hierbei muss der Paketpfad `org.emau.icmvc.ganimed.ttp.psn.alphabets` vorangestellt werden. Die mitgelieferten Alphabete und die jeweils kompatiblen Prüfzifferalgorithmen, jeweils mit den kompletten Pfadangaben, sind in **Tabelle 9-1** aufgelistet. Daneben können auch eigene Alphabete angegeben werden. Mit dem Parameter `properties` werden die semikolonseparierten **Domain Properties** im Format `SCHLÜSSEL=WERT` übermittelt. Alle verfügbaren Eigenschaften sind in **Tabelle 5-35-35-3** aufgelistet. Falls die zu erstellende Domäne einer anderen Domäne untergeordnet werden soll (Kind-Domäne), so kann mittels des Parameters `parentDomainName` der Name der übergeordneten Domäne angegeben werden.

Tabelle 9-1: Alphabete und die jeweils kompatiblen Prüfzifferalgorithmen.

Alphabet	Kompatible Prüfzifferalgorithmen
<code>org.emau.icmvc.ganimed.ttp.psn.alphabets.Hex</code>	←

<code>org.emau.icmvc.ganimed. ttp.psn.alphabets.Numbers</code>	↔	<code>org.emau.icmvc.ganimed.ttp. psn.generator.Verhoeff</code>	↔
		<code>org.emau.icmvc.ganimed.ttp. psn.generator.VerhoeffGumm</code>	↔
		<code>org.emau.icmvc.ganimed.ttp. psn.generator.Damm</code>	↔

<code>org.emau.icmvc.ganimed. ttp.psn.alphabets. NumbersWithoutZero</code>	↔		↔
--	---	--	---

<code>org.emau.icmvc.ganimed. ttp.psn.alphabets.NumbersX</code>	↔	<code>org.emau.icmvc.ganimed.ttp. psn.generator.ReedSolomonLagrange</code>	↔
---	---	--	---

<code>org.emau.icmvc.ganimed. ttp.psn.alphabets.Symbols31</code>	↔	<code>org.emau.icmvc.ganimed.ttp. psn.generator.ReedSolomonLagrange</code>	↔
--	---	--	---

<code>org.emau.icmvc.ganimed. ttp.psn.alphabets.Symbols32</code>	↔	<code>org.emau.icmvc.ganimed.ttp. psn.generator.HammingCode</code>	↔
--	---	--	---

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:psn="http://psn.ttp.ganimed.icmvc.emau.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <psn:addDomain>
      <domainDTO>
        <!--Optional:-->
        <name>DEMO_DOMAIN</name>
        <!--Optional:-->
        <checkDigitClass>org.emau.icmvc.ganimed.ttp.psn.generator.NoCheckDigits</checkDigitClass>
        <!--Optional:-->
        <alphabet>org.emau.icmvc.ganimed.ttp.psn.alphabets.Symbol132</alphabet>
        <!--Optional:-->
        <comment>Schulungsdomäne</comment>
        <label>Schulungsdomäne</label>
        <!--Optional:-->
        <parentDomainNames>DOMAIN</parentDomainNames>
        <config>PSNS_DELETABLE=true;PSN_PREFIX=fp_;INCLUDE_PREFIX_IN_CHECK_DIGIT_CALCULATION=false;PSN_LENGTH=8;FORCE_CACHE=DEFAULT</config>
      </domainDTO>
    </psn:addDomain>
  </soapenv:Body>
</soapenv:Envelope>
```

Abbildung 9-1: XML-Repräsentation zum Anlegen einer Domäne über die SOAP-Schnittstelle.

Wurde die Domäne erfolgreich angelegt, antwortet der Dienst mit einer Antwort, wie sie in **Abbildung 9-2** dargestellt ist.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:addDomainResponse xmlns:ns2="http://psn.ttp.ganimed.icmvc.emau.org/">
  </soap:Body>
</soap:Envelope>
```

Abbildung 9-2: XML-Repräsentation der Rückgabe beim erfolgreichen Anlegen einer Domäne über die SOAP-Schnittstelle.

9.2 Anlegen von Pseudonymen

Im Folgendem wird das Anlegen von Pseudonymen mittels `gpasService`-Schnittstelle exemplarisch vorgestellt. Dabei kann mithilfe der Funktion `getOrCreatePseudonymFor` ein Pseudonym angelegt werden, wenn zum gegebenen Originalwert und Domäne noch kein Pseudonym vorhanden ist. Andernfalls wird das entsprechende Pseudonym zurückgeliefert. In **Abbildung 9-3** wird die eine exemplarische Anfrage dargestellt.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:psn="http://psn.ttp.ganimed.icmvc.emau.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <psn:getOrCreatePseudonymFor>
      <value>12AB34CD56</value>
      <domainName>KAS+</domainName>
    </psn:getOrCreatePseudonymFor>
  </soapenv:Body>
</soapenv:Envelope>
```

Abbildung 9-3: XML-Repräsentation einer beispielhaften Anfrage zum Anlegen oder Abfragen eines Pseudonyms über die SOAP-Schnittstelle.

In **Abbildung 9-4** wird die entsprechende Antwort dargestellt. Darin enthalten ist das angelegte bzw. abgefragte Pseudonym.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:getOrCreatePseudonymForResponse xmlns:ns2="http://psn.ttp.ganimed.icmvc.emau.org/">
      <psn>KAS_EVNT1XZJHJ</psn>
    </ns2:getOrCreatePseudonymForResponse>
  </soap:Body>
</soap:Envelope>
```

Abbildung 9-4: XML-Repräsentation der Rückgabe zu einer Anfrage zum Anlegen oder Abrufen eines Pseudonyms über die SOAP-Schnittstelle.

Zum Anlegen von Alt-Pseudonymen kann die Funktion `insertValuePseudonymPair` verwendet werden. Dabei wird unter Angabe des Originalwerts, des Pseudonyms und der Domäne der jeweilige Eintrag angelegt. Um mehrere Pseudonyme innerhalb einer Anfrage anzulegen, kann die Funktion `insertValuePseudonymPairs` verwendet werden.

9.3 Pseudonymisieren (Abfragen von Pseudonymen)

Wie in **Abschnitt 9.1** gezeigt, kann mittels der Funktion `getOrCreatePseudonymFor` mittels `gpasService`-Schnittstelle ein Pseudonym angelegt oder wenn dieses bereits vorhanden ist, abgefragt werden. Wenn ein Pseudonym abgefragt werden soll, ohne ein etwaiges Anlegen von diesem, kann die Funktion `getPseudonymFor` verwendet werden. Dabei wird der Originalwert und die Domäne angegeben. In **Abbildung 9-5** wird ein exemplarischer Abruf dargestellt.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:psn="http://psn.ttp.ganimed.icmvc.emau.org">
  <soapenv:Header/>
  <soapenv:Body>
    <psn:getPseudonymFor>
      <value>12AB34CD56</value>
      <domainName>KAS+</domainName>
    </psn:getPseudonymFor>
  </soapenv:Body>
</soapenv:Envelope>

```

Abbildung 9-5: XML-Repräsentation einer beispielhaften Anfrage zum Abfragen eines Pseudonyms über die SOAP-Schnittstelle.

Die entsprechende Antwort wird in **Abbildung 9-6** dargestellt. Darin ist das entsprechende Pseudonym der Domäne zum angegebenen Originalwert enthalten.

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:getPseudonymForResponse xmlns:ns2="http://psn.ttp.ganimed.icmvc.emau.org">
      <psn>KAS_EVNT1XZJHJ</psn>
    </ns2:getPseudonymForResponse>
  </soap:Body>
</soap:Envelope>

```

Abbildung 9-6: XML-Repräsentation der Rückgabe zu einer Anfrage zum Abrufen eines Pseudonyms über die SOAP-Schnittstelle.

Mit der Funktion `getPseudonymForList` kann ein Abfragen mehrerer Pseudonyme mit nur einer Anfrage erreicht werden.

9.4 De-Pseudonymisieren (Abfragen von Originalwerten)

Ähnlich wie Pseudonyme, können Originalwerte mittels `gpasService`-Schnittstelle abgefragt werden. Hierzu kann mittels der Funktion `getValueFor` und unter Angabe des Pseudonyms und der Domäne der zugeordnete Originalwert abgefragt werden. In **Abbildung 9-7** ist eine Abfrage exemplarisch dargestellt.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:psn="http://psn.ttp.ganimed.icmvc.emau.org">
  <soapenv:Header/>
  <soapenv:Body>
    <psn:getValueFor>
      <psn>KAS_EVNT1XZJHJ</psn>
      <domainName>KAS+</domainName>
    </psn:getValueFor>
  </soapenv:Body>
</soapenv:Envelope>

```

Abbildung 9-7: XML-Repräsentation einer beispielhaften Anfrage zum Abfragen eines Originalwerts über die SOAP-Schnittstelle.

In **Abbildung 9-8** wird die entsprechende Antwort dargestellt. Darin ist der Originalwert zum Pseudonym enthalten.

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:getValueForResponse xmlns:ns2="http://psn.ttp.ganimed.icmvc.emau.org/">
      <value>12AB34CD56</value>
    </ns2:getValueForResponse>
  </soap:Body>
</soap:Envelope>

```

Abbildung 9-8: XML-Repräsentation der Rückgabe zu einer Anfrage zum Abrufen eines Originalwerts über die SOAP-Schnittstelle.

Um mehrere Originalwerte mit nur einer Anfrage abzufragen, kann die Funktion `getValueForList` verwendet werden.

10 FHIR-Unterstützung für gPAS per FHIR-Gateway

„Fast Healthcare Interoperability Resources (kurz: FHIR®) ist ein von HL7 erarbeiteter Standard. Er unterstützt den Datenaustausch zwischen Softwaresystemen im Gesundheitswesen. FHIR beschreibt Datenformate und Elemente als sogenannte „Ressourcen“ und bietet eine Schnittstelle an, um diese auszutauschen“⁸.

Um sowohl bestehende Anwenderprojekte als auch künftige Nutzer bei der Umsetzung FHIR-orientierter Infrastrukturen und Prozesse zu unterstützen, stellen wir für ab gPAS-Version 1.10.0 ein zusätzliches Treuhandstellen-FHIR-Gateway (kurz: TPP-FHIR Gateway) als Mittler von FHIR-spezifischen Infrastrukturkomponenten und gPAS bereit.

Für ausgewählte Funktionalitäten zur domänenspezifischen Generierung neuer Pseudonyme und zur Abfrage von Mappings von Pseudonym- und Originalwerten wurden in FHIR nachfolgende Funktionen umgesetzt und sind nach erfolgreichem Deployment direkt per REST nutzbar.

Tabelle 10-1: Übersicht der durch FHIR unterstützten gPAS-Funktionen.

gPAS (SOAP Funktion)	FHIR (REST-Äquivalent)	Details im
<code>getOrCreatePseudonymForList</code>	<code>\$pseudonymizeAllowCreate</code>	https://simplifier.net/guide/ttp-fhir-gateway-ig/pseudonymizeAllowCreate
<code>getPseudonymForList</code>	<code>\$pseudonymize</code>	https://simplifier.net/guide/ttp-fhir-gateway-ig/pseudonymize
<code>getValueForList</code>	<code>\$dePseudonymize</code>	A https://simplifier.net/guide/ttp-fhir-gateway-ig/dePseudonymize

⁸ https://de.wikipedia.org/wiki/Fast_Healthcare_Interoperability_Resources, Zugriff am 22.01.2021

Details zur Nutzung der Funktionen (Aufruf, Parameter, beispielhafte Antworten) entnehmen Sie bitte dem frei verfügbaren Implementation-Guide⁹. Weitere Details unter <https://ths-greifswald.de/fhir>.

11 Health Checks

Der gPAS besitzt ab Version 1.10 einen Health Check. Damit ist es möglich einen Status zum Zustand des Dienstes abzufragen. Dabei werden die Erreichbarkeit der Datenbank und der freie Arbeitsspeicher für den Dienst geprüft. Sollte dieser in einen Zustand wechseln oder dies bevorstehen, in dem dieser nicht mehr ordnungsgemäß arbeiten kann, liefert der Health Check einen entsprechenden Status. Es ist somit möglich, automatisiert den Status abzufragen und im Fall eines unerwünschten Zustands entsprechende Meldung z.B. an einen Administrator zu geben, sodass dieser zeitnah eingreifen kann und mögliche Probleme behebt.

⚠ Hinweis: Der Health Check kann derzeit nur mit dem Wildfly 16 betrieben werden. Eine Anpassung auf spätere Wildfly Versionen erfolgt in der gPAS Version 1.11.

11.1 Abfrage

Der Health Check wird über die URL des Wildfly-Servers unter dem Port 9990 aufgerufen. Bsp.:

```
http://example.org:9990/health
```

Der Wildfly-Server fragt dabei den gICS ab, welcher die entsprechenden Zustandsdaten liefert. Dies erfolgt im JSON-Format und kann demnach auch automatisiert weiterverarbeitet werden. Eine exemplarische Rückgabe ist im Folgendem zu sehen:

```
{
  "outcome": "UP",
  "checks": [
    {
      "name": "gPAS Database Healthcheck",
      "state": "UP",
      "data": {
        "updated": "Mon Jan 25 19:48:51 CET 2021"
      }
    }, {
      "name": "gPAS Memory Healthcheck",
      "state": "UP",
      "data": {
        "total": 6737100800,
        "maximal": 13362528256,
        "threshold": 104857600,
        "free": 5425185360,
        "updated": "Mon Jan 25 19:48:52 CET 2021"
      }
    }
  ]
}
```

```
]
}
```

Die Health Checks für den gPAS sind als „gPAS Database Healthcheck“ und „gPAS Memory Healthcheck“ benannt. Diese beinhalten zusätzliche Informationen zum jeweiligen Health Check. Die Health Checks werden nicht bei jedem Aufruf ausgeführt, sodass ggf. der Dienst damit belastet wird, sondern die Ergebnisse werden standardmäßig 10 Sekunden gecached. Diese Zeitspanne ist konfigurierbar (siehe **Abschnitt 11.2**). Der Zeitstempel wird im Feld `updated` mitgeliefert. Beide Health Checks liefern außerdem den Status im Feld `state` zurück. Solange der Dienst ordnungsgemäß ausgeführt wird, wird der Wert `UP` geliefert. Im Fehlerfall `DOWN`. Der Health Check zur Arbeitsspeicherüberwachung liefert darüber hinaus noch den belegten Speicher der Java Virtual Maschine (JVM) (`total`), den maximal der JVM verfügbaren Speicher (`maximal`), den noch freien Speicher (`free`) und den konfigurierbaren Schwellwert (`threshold`), ab wann der Dienst in den Status `DOWN` wechseln soll (standardmäßig 100 MB). Letzteres dient als Schutzmaßnahme, damit der Dienst zwar grundsätzlich ordnungsgemäß funktioniert, jedoch bevor ein endgültiges volllaufen des Speichers erreicht ist, Gegenmaßnahmen getroffen werden können.

11.2 Konfiguration

Die Konfiguration der Health Checks erfolgt in der Datei `healthcheck-gPAS.xml`. Diese liegt im Servlet des Backends (*.ear).

⚠ Hinweis: Die direkte Bearbeitung der Datei innerhalb des Servlets zur Laufzeit ist nicht zu empfehlen und führt zu einem Neustart des Dienstes. Es wird empfohlen den Dienst für eine Änderung herunterzufahren. Optimalerweise genügt eine einmalige Konfiguration vor dem ersten Hochfahren.

Standardmäßig beinhaltet die Konfigurationsdatei folgenden Inhalt:

```
<ma:HealthCheckConfiguration
  xmlns:ma=http://www.ttp.icmvc.emau.org/gpas/healthcheck/model
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xsi:schemaLocation="http://www.ttp.icmvc.emau.org/gpas/healthcheck/model
healthcheck-config-1.0.0.xsd">
  <service-name>gpas</service-name>
  <database-configuration>
    <persistence-unit-name>gpas</persistence-unit-name>
    <update-time>10</update-time>
  </database-configuration>
  <memory-configuration>
    <available-memory-threshold>104857600</available-memory-threshold>
    <update-time>10</update-time>
  </memory-configuration>
</ma:HealthCheckConfiguration>
```

Mit dem Feld `service-name` kann ein frei wählbarer Name vergeben werden. Dieser wird lediglich vor die konkrete Bezeichnung des Health Checks gelegt. Das Feld `persistence-unit-name` gibt den internen Namen für die Datenbank an. Eine Änderung ist nicht zu empfehlen. Die Zeitspanne des Caches für den Health Check Zustand, wird mit dem Feld `update-time` in Sekunden angegeben. Um den Schwellwert für den Health Check zur Speicherüberwachung anzugeben, wird das Feld `available-memory-threshold` in Byte verwendet. Standardmäßig sind hier 100 MB definiert.

Bei Unterschreitung dieses Schwellwerts wird nur der Status des Health Checks auf `DOWN` gelegt. Der Dienst funktioniert ordnungsgemäß weiter, bis der Speicher komplett belegt ist.

12 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.

12.1 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.

12.2 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.

12.3 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 der konfigurierte Timeout zulässt. Der Timeout wird standardmäßig nach 5 Minuten ausgelöst, sofern der Wildfly bis dahin nicht hochgefahren ist. Es ist dann erforderlich, die Konfiguration des Wildfly anzupassen. 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 der Timeout auf 15 Minuten (900 Sekunden) hochgesetzt.

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

13 Publikationen und Vorträge

Bialke M*, Bahls T, Havemann C, Piegsa J, Weitmann K, Wegner T, et al.

MOSAIC. A modular approach to data management in epidemiological studies. (Originalartikel)

METHODS OF INFORMATION IN MEDICINE. 2015; 54(4):364-371.

<http://dx.doi.org/10.3414/ME14-01-0133>

Bialke M*, Penndorf P, Wegner T, Bahls T, Havemann C, Piegsa J, et al.

A workflow-driven approach to integrate generic software modules in a Trusted Third Party (Originalartikel)

Journal of Translational Medicine. 2015; 13(176).

<http://www.translational-medicine.com/content/13/1/176>

Bialke M*, Rau H*, Thamm O, Schuldt R, Penndorf P, Blumentritt A, Gött R, Piegsa J, Bahls T, Hoffmann W.

Toolbox for Research, or How to facilitate a central data management in small-scale research projects. (Originalartikel)

Journal of Translational Medicine. (open access) 2018; 16(16). IF: 4.197 (2017)

<http://rdcu.be/FynH>

14 Weiterführende Informationen

Produktbroschüre gPAS®

<https://www.ths-greifswald.de/wp-content/uploads/2019/01/gPAS-Produktbrief.pdf>

gPAS® Service Spezifikation

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

gPAS® Demo

<https://ths-greifswald.de/gpas/demo>

Git-Repository

<https://github.com/mosaic-hgw/gpas> **Docker Installation**

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

Docker Compose Installation

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

Docker Cheat Sheet

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

Docker und Docker Compose Cheat Sheet

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