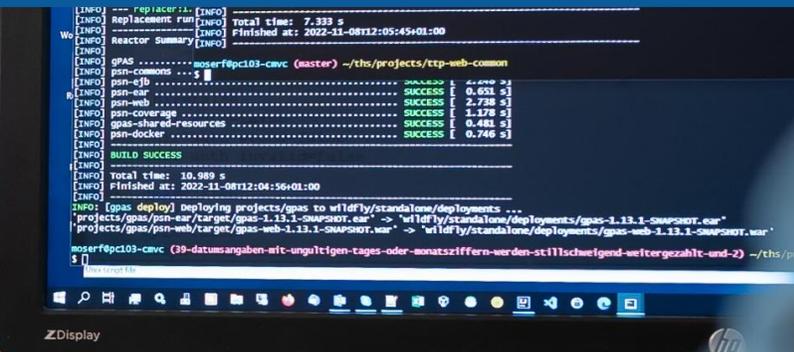
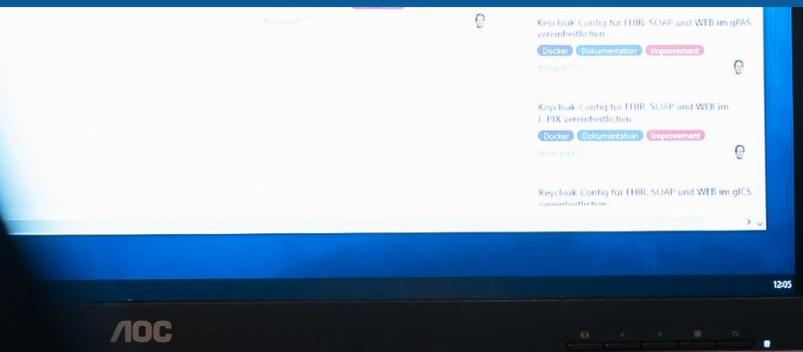




Unabhängige Treuhandstelle

UNIVERSITÄTS MEDIZIN GREIFSWALD

Community Dialog #10



Christopher Hampf | Universitätsmedizin Greifswald

Der THS Community Dialog:

- bietet eine zentrale Anlaufstelle für **Anwender-Fragen, die über den Inhalt der Handbücher hinausgehen**
- hilft Hintergründe besser zu verstehen und gemeinsam mit den Entwicklern erforderliche Antworten zu finden.
- schafft die Basis für eine gemeinsam aufgebaute FAQ.
- ist Möglichkeit mit anderen eigene Ideen zu diskutieren und zu teilen

ths-greifswald.de/community





Top 1 Q&A

Eingehende Fragen und ausgehende Antworten

Top 2 Umgang mit Updates

Ein Blick auf künftige Neuerungen im Update-Prozess

Top 3 Fokus Qualität von Einwilligungen

Ausblick auf neue Qualitäts-Features im kommenden gICS-Release

Top 4 Die Community stellt sich vor

Niclas-Slim Engels Jr. (EY): Aufbau und Betrieb einer Treuhandstelle für die RUB



TOP 1 - FRAGEN UND ANTWORTEN



Im ersten Schritt sollen nur die Einwilligungen für den Broad Consent mit gICS gemanaged werden. Perspektivisch ist allerdings die Erhebung weiterer Einwilligungen nicht ausgeschlossen. Wäre es vorteilhaft, die Domänen (Mandantentrennung) nach Einwilligungsart zu separieren, beispielsweise eine Domäne nur für den Broad Consent anzulegen?



Vorhaben/Projekte/Mandanten sollten in unterschiedlichen Domänen verwaltet werden. Dies erlaubt auch eine Domänenspezifische Rechtevergabe. Innerhalb der Domänen können verschiedene Einwilligungen erhoben und verwaltet werden. Die Vorlagen können auch zu einem späteren Zeitpunkt ergänzt werden.



Mehr zu Rechtevergaben unter <https://www.ths-greifswald.de/ttp-tools/domain-auth>



Wir würden gerne Docker aus Sicherheitsgründen im Rootless Modus unter Debian Linux (kein sudo) betreiben. Gibt es auch eine Version, die den Rootles Modus unterstützt?



Wir haben derzeit (noch) keine Erfahrung mit Rootless-Images. Der mit den THS-Tools ausgelieferte Wildfly benötigt selbst zwar keine Root-Rechte, Docker jedoch schon.

<http://www.ths-greifswald.de/e-pix/handbuch>

Vergabe von Schreibrechten



```
sudo chmod -R 755 /opt/compose-wildfly/  
chown -R 1000:1000 /opt/compose-wildfly/logs/ /opt/compose-  
wildfly/deployments/
```

Aus Gründen von Leistung und Ausfallsicherheit sollten die Container des E-PIX auf einem dedizierten Server eingerichtet werden. Zur Administration werden der User `epix` (uid 1000) aus der Gruppe `users` (gid 1000) genutzt.



[Für] die Nutzung [der THS-Softwarelösungen] wird Linux (Ubuntu) empfohlen. Gibt es ihrerseits Erfahrungswerte mit der Umsetzung unter RedHat? Falls ja, funktioniert die Umsetzung in RedHat auch mittels Podman statt Docker?



Bislang keine Erfahrungen mit RedHat in diesem Kontext. Ein Community Projekt mit gICS unter podman existiert bereits.



Mehr unter <https://github.com/thz-community/podman>

The screenshot shows the GitHub repository page for 'thz-community / podman'. The repository is public and has 2 watchers, 1 fork, and 0 stars. The main branch is selected, and there is 1 branch and 0 tags. The repository contains a commit by 'chrismade' titled 'fix more typos after review' from 8 months ago, with 12 commits. The file list includes 'config', 'etc', 'ignition', 'LICENSE', and 'README.md'. The 'About' section shows 'podman scripts', a README, AGPL-3.0 license, and 0 stars.

File	Description	Commit Date
config	section bunkerweb added	8 months ago
etc	wireguard frontend, config and user systemd files added	8 months ago
ignition	fstab and wireguard backend added	8 months ago
LICENSE	Initial commit	9 months ago
README.md	fix more typos after review	8 months ago



Im zweiten Quartal 2024 startet das Modellvorgaben genomDE, an welches sich unser Standort angeschlossen [hat]. Die Transfer-Vorgangsnummer soll durch die lokale Treuhandstelle erzeugt werden. Wird dies zünftig durch gPAS möglich sein?



Die Vertrauensstelle des RKI ist in diesem Vorhaben federführend und macht entsprechende Vorgaben. Wir stehen im Austausch mit dem RKI und geben dem RKI Hinweise zur Konfiguration des gPAS. Die resultierenden Referenzkonfigurationen werden vom RKI bereitgestellt. Etwaige Anpassungen im gPAS werden als Anforderungen aufgenommen und gemäß der Release-Zyklen eingetaktet. Der gPAS kann daher für das Vorhaben an den genomDE-Standorten in Betracht gezogen werden.



Es gibt die Funktion des automatischen Auslesens der Scaninformationen. Muss die benötigte Applikation OpenCV vorab separat installiert werden oder ist diese im Release enthalten? (Fehlermeldung: OpenCV steht auf Ihrer Plattform nicht zur Verfügung)



Die erforderlichen Bibliotheken sind standardmäßig Teil der Releases. Für eine detaillierte Einschätzung der Fehlermeldung sind weitere Informationen erforderlich.



Gibt es Empfehlungen ihrerseits bezüglich Backups der Datenbank[en der THS-Softwarelösungen]?



Die Dokumentation für Softwareupdates beinhaltet sowohl Sicherung- als auch Wiederherstellungshinweise. Der Umgang mit Backups sollte vom jeweiligen Standort bewertet und umgesetzt werden. In der THS Greifswald werden in produktiven Projekten automatisch täglich Backups angelegt und verschlüsselt auf einem separaten System abgelegt.



Mehr unter <https://www.ths-greifswald.de/gics/update>

Sichern der existierenden Datenbank

Um zusätzlich die Sicherung der existierend
Docker-Konsole angestoßen und die resultier

```
sudo docker exec gics-<old-versi  
> backup-gics-<old-version>-2022
```

Der Name der bestehenden MySQL-Instanz

Im Fehlerfall: Wiederherstellung der Datenbank

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

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



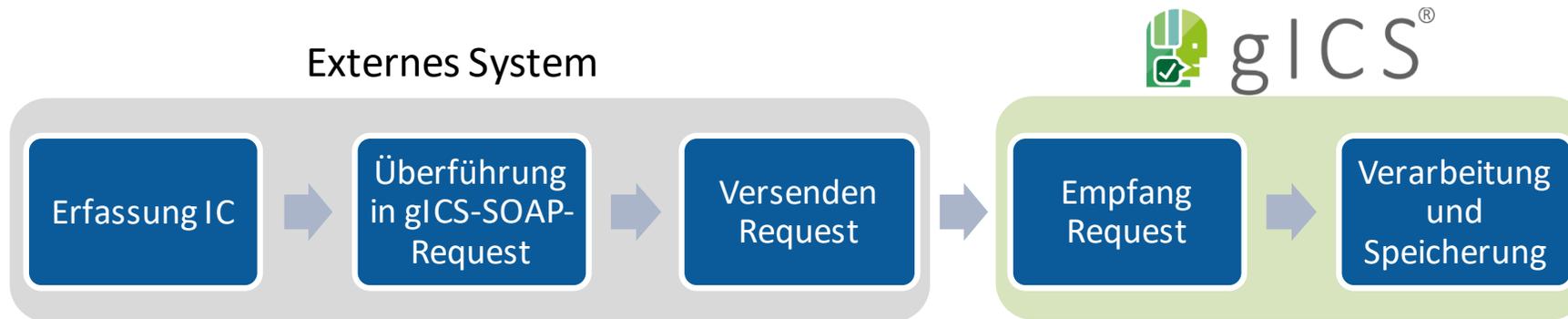
Was ist zu beachten, wenn Consents sowohl auf digitalem als auch analogem Weg eingeholt und in gICS gepflegt werden? Der digitale Weg würde hier vermutlich über Thieme E-Consent Pro und die eventuelle Nutzung des von Thieme angebotenen Integrationsmoduls laufen?



<https://www.ths-greifswald.de/live-demo-neues-integrationsmodul-fuer-gics-in-thiemes-e-consentpro-vorgestellt>



Es kommt drauf an. Thieme E-Consent Pro ist hier nur ein Client, der die SOAP-Schnittstelle nutzt.





Können die Thieme Einwilligungsvorlagen für den Broad Consent dann auch innerhalb von gICS genutzt werden oder muss dafür zwingend die Vorlage der Medizininformatik-Initiative benutzt werden?



<https://www.ths-greifswald.de/live-demo-neues-integrationsmodul-fuer-gics-in-thiemes-e-consentpro-vorgestellt>



Folgende Punkte sind dabei zu beachten:

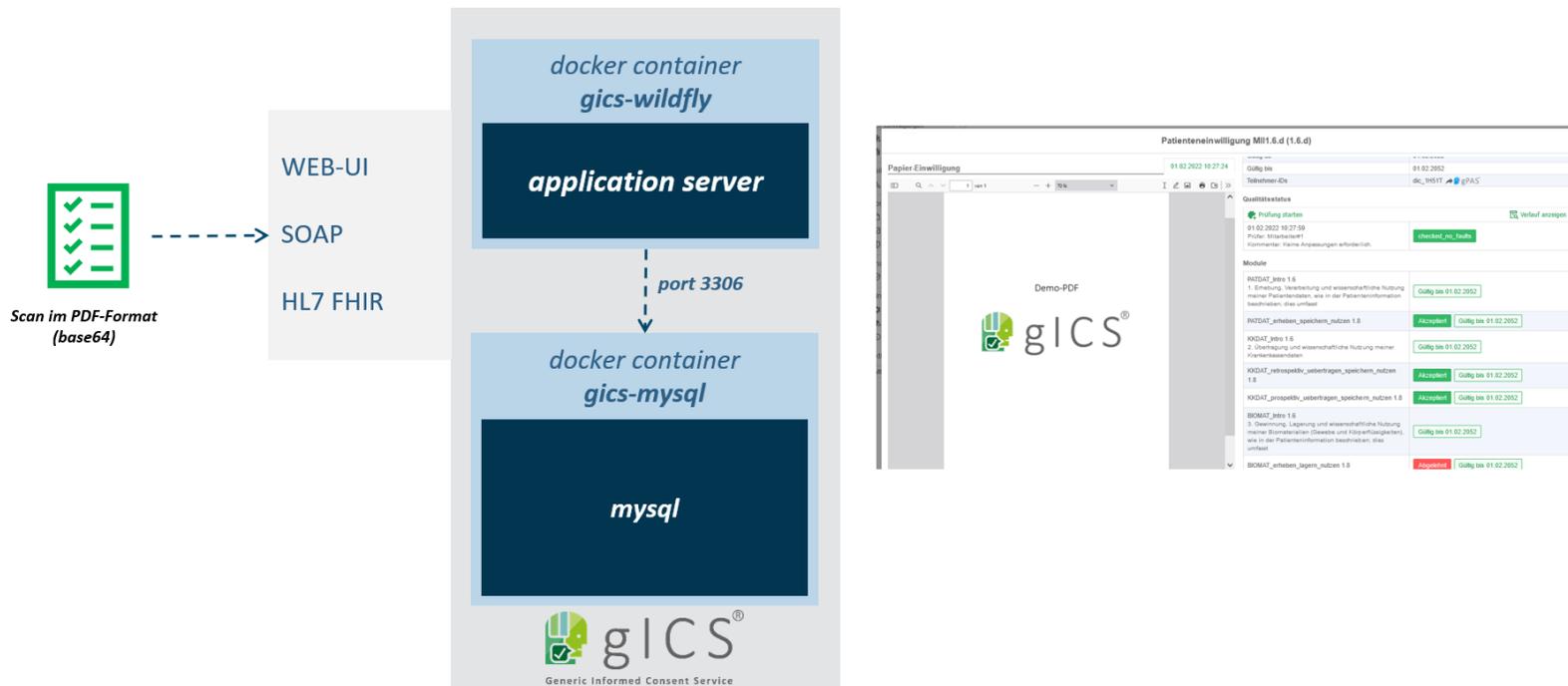
- gICS kann Einwilligungsvorlagen für Forschungsprojekte verwalten.
- Der MII-Broad Consent kann im gICS verwaltet werden.
- Thieme kann seit Feb. 2022 Einwilligungen an die gICS-Schnittstelle versenden.
- Thieme kann auch den MII-Broad Consent erfassen und an den gICS versenden.
- Voraussetzung: Thieme muss die im gICS verwalteten Vorlagen (z.B. MII-Broad Consent) korrekt konfigurieren und verwenden.



Wie werden Scans an gICS übermittelt und in gICS gespeichert?



Sowohl die Übermittlung über die Schnittstellen (Web-UI, SOAP, FHIR), als auch die Speicherung in der MySQL Datenbank erfolgt in einer base64-Kodierung.





TOP 2 AUSBLICK: UMGANG MIT UPDATES



- Tool-spezifischer Link mit Update-Hinweisen, z.B. <https://www.ths-greifswald.de/gics/update>
- Zusätzlich: Umfassende Übersicht von Konfigurationsdateien und -variablen

```
## 7. Alle verfügbaren Enviroment-Variablen
In den env-Dateien stehen weitere Details zu den einzelnen Variablen.

#### ./envs/ttp_gics.env **<-- ehemals gics.env**
| Kategorie | Variable | verfügbare Werte oder Schema | default |
|-----|-----|-----|-----|
| Logging | TTP_GICS_LOG_TO_FILE **<-- Alias von GICS_FILE_LOG** | true, false | false |
| Logging | TTP_GICS_LOG_LEVEL **<-- Alias von GICS_LOG_LEVEL** | TRACE, DEBUG, INFO, WARN, ERROR, FATAL | INFO |
| Database | TTP_GICS_DB_HOST **<-- Alias von GICS_DB_HOST** | \<STRING\> | mysql |
| Database | TTP_GICS_DB_PORT **<-- Alias von GICS_DB_PORT** | 0-65535 | 3306 |
| Database | TTP_GICS_DB_NAME **<-- Alias von GICS_DB_NAME** | \<STRING\> | gics |
| Database | TTP_GICS_DB_USER **<-- Alias von GICS_DB_USER** | \<STRING\> | gics_user |
| Database | TTP_GICS_DB_PASS **<-- Alias von GICS_DB_PASS** | \<STRING\> | gics_password |
| Security | TTP_GICS_WEB_AUTH_MODE **<-- Alias von GICS_AUTH_MODE** | gras, keycloak, keycloak-json | - |
| Security | TTP_GICS_SOAP_KEYCLOAK_ENABLE **<-- neu** | true, false | - |
| Security | TTP_GICS_SOAP_ROLE_USER_NAME **<-- neu** | \<STRING\> | role.gics.user |
| Security | TTP_GICS_SOAP_ROLE_USER_SERVICES **<-- neu** | \<STRING\> | /gicas/gicsService,/gics/gicsServiceWithNotification |
| Security | TTP_GICS_SOAP_ROLE_ADMIN_NAME **<-- neu** | \<STRING\> | role.gics.admin |
| Security | TTP_GICS_SOAP_ROLE_ADMIN_SERVICES **<-- neu** | \<STRING\> | /gics/gicsManagementService,/gics/gicsFhirService |
| Security | TTP_GICS_AUTH_DOMAIN_ROLES **<-- neu** | DISABLED, FORCED, IMPLIED | IMPLIED |

#### ./envs/ttp_noti.env **<-- ehemals noti.env**
| Kategorie | Variable | verfügbare Werte oder Schema | default |
|-----|-----|-----|-----|
| Database | TTP_NOTI_DB_HOST **<-- Alias von NOTI_DB_HOST | \<STRING\> | mysql |
| Database | TTP_NOTI_DB_PORT **<-- Alias von NOTI_DB_PORT | 0-65535 | 3306 |
| Database | TTP_NOTI_DB_NAME **<-- Alias von NOTI_DB_NAME | \<STRING\> | notification_service |
| Database | TTP_NOTI_DB_USER **<-- Alias von NOTI_DB_USER | \<STRING\> | noti_user |
| Database | TTP_NOTI_DB_PASS **<-- Alias von NOTI_DB_PASS | \<STRING\> | noti_password |
```

- Angefangen hat es vor langer Zeit mit einer Environment-Variable
- Mit wachsender Beliebtheit und steigenden Anforderungen, wurden es schnell mehr und mehr
- Für die bessere Übersichtlichkeit wurden Kategorien (Präfixe) geschaffen
 - Dafür mussten Variablen neu benannt werden, jetzt mit: `WF_` und `TTP_`
 - Für die Kompatibilität blieben die alten Namen erhalten
- Und um noch mehr Einfachheit zu generieren, wurden weitere Variablen erzeugt
 - Main-Variablen, die eine ganze Gruppe von Variablen zusammenfassen können, z.B.:

<code>TTP_DB_HOST</code>	statt	<code>TTP_<TOOL>_DB_HOST</code>
<code>TTP_KEYCLOAK_SERVER_URL</code>	statt	<code>TTP_<PROTOKOLL>_KEYCLOAK_SERVER_URL</code>



Jede unserer Variablen bekommt jetzt Zusatzinformationen, u.a.

- Name
- Default-Wert
- Löschung geplant
- Ersatzvariablen
- Bereits gelöscht

Diese Informationen werden vor jedem Start des Applikation-Servers geprüft.

- Welche ENV-Variablen stehen dem Container zur Verfügung?
- Welche sind davon bekannt, also haben Zusatzinformationen?
- Bei Verwendung einer veralteten Variable, wird ein `WARNING` ausgegeben.



```
root@docker:~ 10:04 # docker run --rm -it -e "CONSOLE_LOG_LEVEL=INFO" mosaicgreifswald/wildfly:31
=====
https://hub.docker.com/repository/docker/mosaicgreifswald
=====

last updated           : 2024-02-22 09:30:45
Distribution           : Debian GNU/Linux 12.5
zulu-jre               : 21.0.2
WildFly                : 31.0.0.Final
MySQL-Connector        : 8.3.0
EclipseLink            : 4.0.2
=====

[run.sh:14] test write-permission in folder /entrypoint-logs
[run.sh:16] - /entrypoint-logs -> ok
[run.sh:16] - /entrypoint-logs/wildfly -> ok

[run.sh:21] check environment-variables
[run.sh:26] - WARNING You are using an obsolete variable that will be removed in one of the next versions: CONSOLE_LOG_LEVEL
[run.sh:30] -> Use instead: WF_CONSOLE_LOG_LEVEL

[run.sh:37] 1 run-script(s) found to execute in following order and type
[run.sh:38] - 20:service:/opt/mosaic/wildfly/run_wildfly.sh:/opt/mosaic/wildfly/wildfly_started.sh
[run.sh:70] start /opt/mosaic/wildfly/run_wildfly.sh as service and wait for running with /opt/mosaic/wildfly/wildfly_started.sh
=====

This is a Docker image for the Java application server WildFly. The
image is based on slim debian-image and prepared for the tools of the
university medicine greifswald. (but can also be used for other similar
```



```
root@docker:~ 10:04 # docker run --rm -it -e "WF_CONSOLE_LOG_LEVEL=INFO" mosaicgreifswald/wildfly:31
=====
https://hub.docker.com/repository/docker/mosaicgreifswald

=====

last updated           : 2024-02-22 09:30:45
Distribution           : Debian GNU/Linux 12.5
zulu-jre               : 21.0.2
WildFly                : 31.0.0.Final
MySQL-Connector        : 8.3.0
EclipseLink           : 4.0.2

=====

[run.sh:14] test write-permission in folder /entrypoint-logs
[run.sh:16] - /entrypoint-logs -> ok
[run.sh:16] - /entrypoint-logs/wildfly -> ok

[run.sh:21] check environment-variables
[run.sh:32] everything is fine.

[run.sh:37] 1 run-script(s) found to execute in following order and type
[run.sh:38] - 20:service:/opt/mosaic/wildfly/run_wildfly.sh:/opt/mosaic/wildfly/wildfly_started.sh
[run.sh:70] start /opt/mosaic/wildfly/run_wildfly.sh as service and wait for running with /opt/mosaic/wildfly/wildfly_started.sh

=====

This is a Docker image for the Java application server WildFly. The
image is based on slim debian-image and prepared for the tools of the
university medicine greifswald (but can also be used for other similar
media etc.)
```



```
root@docker:~/git/docker/bash-scripts/local-wildfly-from-image/target 16:33 # docker exec gpas-wildfly envs
VARIABLE-NAME          CURRENT-VALUE          DEFAULT-VALUE          STATE
CONSOLE_LOG_LEVEL      DEPRECATED             DEPRECATED             deprecated, use WF_CONSOLE_LOG_LEVEL
DEBUG_PORT              *:8787
ENTRY_JAVA_CACERTS     /entrypoint-java-cacerts
ENTRY_LOGS              /entrypoint-logs
ENTRY_USAGE            /entrypoint-help-and-usage
ENTRY_WILDFLY_ADDINS   /entrypoint-wildfly-addins
ENTRY_WILDFLY_CLI      /entrypoint-wildfly-cli
ENTRY_WILDFLY_DEPLOYS /entrypoint-wildfly-deployments
ENTRY_WILDFLY_LOGS     /entrypoint-wildfly-logs
GPAS_AUTH_MODE         DEPRECATED             DEPRECATED             deprecated, use TTP_GPAS_WEB_AUTH_MODE
GPAS_DB_HOST           DEPRECATED             DEPRECATED             deprecated, use TTP_GPAS_DB_HOST
GPAS_DB_NAME           DEPRECATED             DEPRECATED             deprecated, use TTP_GPAS_DB_NAME
GPAS_DB_PASS           DEPRECATED             DEPRECATED             deprecated, use TTP_GPAS_DB_PASS
GPAS_DB_PORT           DEPRECATED             DEPRECATED             deprecated, use TTP_GPAS_DB_PORT
GPAS_DB_USER           DEPRECATED             DEPRECATED             deprecated, use TTP_GPAS_DB_USER
GPAS_FILE_LOG          DEPRECATED             DEPRECATED             deprecated, use TTP_GPAS_LOG_TO_FILE
GPAS_LOG_LEVEL         DEPRECATED             DEPRECATED             deprecated, use TTP_GPAS_LOG_LEVEL
GRAS_DB_HOST           DEPRECATED             DEPRECATED             deprecated, use TTP_GRAS_DB_HOST
GRAS_DB_NAME           DEPRECATED             DEPRECATED             deprecated, use TTP_GRAS_DB_NAME
GRAS_DB_PASS           DEPRECATED             DEPRECATED             deprecated, use TTP_GRAS_DB_PASS
GRAS_DB_PORT           DEPRECATED             DEPRECATED             deprecated, use TTP_GRAS_DB_PORT
GRAS_DB_USER           DEPRECATED             DEPRECATED             deprecated, use TTP_GRAS_DB_USER
HOME                   /opt/mosaic
HOSTNAME               13b419c0038c
JAVA_HOME              /usr/lib/jvm/zulu
JAVA_VERSION
JBOSS_CLI              /opt/mosaic/wildfly/bin/jboss-cli.sh
KEYCLOAK_CLIENT_ID     DEPRECATED             DEPRECATED             deprecated, use TTP_KEYCLOAK_CLIENT_ID
KEYCLOAK_CLIENT_SECRET DEPRECATED             DEPRECATED             deprecated, use TTP_KEYCLOAK_CLIENT_SECRET
KEYCLOAK_CONFIDENTIAL_PORT DEPRECATED             DEPRECATED             deprecated, use TTP_KEYCLOAK_CONFIDENTIAL_PORT
KEYCLOAK_REALM         DEPRECATED             DEPRECATED             deprecated, use TTP_KEYCLOAK_REALM
```



TOP 3 - FOKUS QUALITÄT VON EINWILLIGUNGEN



Die Community stellt sich vor

Referent: Niclas-Slim Engels Jr.

EY Business & Technology Consulting | Healthcare



AUFBAU UND BETRIEB EINER TREUHANDSTELLE FÜR DIE RUB





Senior Java Developer*in

in Vollzeit

Wir sind auf der Suche nach Unterstützung für unser Team. Fokus: Architekturentwurf und Implementierungsarbeit zur Weiterentwicklungen unserer Lösungen

[Mehr erfahren >](#)



System-Manager*in IT- Infrastruktur und Software Engineer

in Vollzeit

Wir sind auf der Suche nach Unterstützung für unser Team zur Weiterentwicklung unserer IT-Infrastrukturen

[Mehr erfahren >](#)



Wissenschaftliche*r Mitarbeiter*in

in Vollzeit

Wir sind auf der Suche nach Unterstützung für unser Team. Fokus: Konzept- und Implementierungsarbeit zur Weiterentwicklungen unserer Lösungen

[Mehr erfahren >](#)



Kontakt

Dr. Martin Bialke
Universitätsmedizin Greifswald
Institute für Community Medicine
Ellernholzstr. 1-2
17475 Greifswald

☎ +49 (3834) 86-7580
✉ martin.bialke@uni-greifswald.de
🌐 ths-greifswald.de