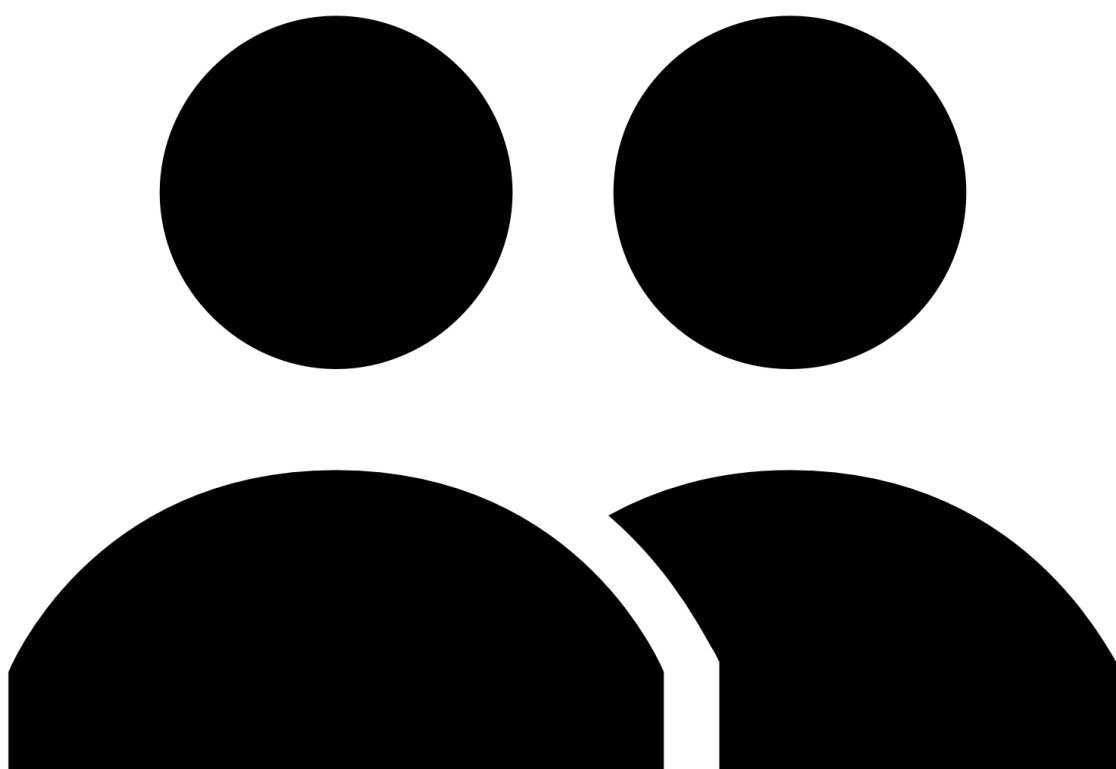


# Informationen Benutzerverwaltung



## Projekt WISSLearnCards – Benutzerverwaltung

Mitarbeiter dieses Projekts:

- Frithjof Hoppe
- Philippe Krüttli
- Hugo Lucca (Lerncoach)

# Inhaltsverzeichnis

|       |  |    |
|-------|--|----|
| 1     | Ausgangslage .....                             | 3  |
| 1.1   | Erweiterbarkeit .....                          | 3  |
| 2     | Berechtigungskonzept .....                     | 4  |
| 2.1   | Standardberechtigungen .....                   | 4  |
| 2.2   | Derive-Berechtigung .....                      | 4  |
| 2.3   | Teamwork-Berechtigung .....                    | 4  |
| 3     | Datenbankkonzept .....                         | 5  |
| 3.1   | Datenbanken .....                              | 5  |
| 3.2   | Datenbankstruktur der Benutzerverwaltung ..... | 6  |
| 3.2.1 | Erläuterungen zur DB-Struktur .....            | 7  |
| 3.3   | Wichtiges .....                                | 8  |
| 3.3.1 | Nachfolge eines DMO .....                      | 8  |
| 3.3.2 | Eindeutigkeit .....                            | 8  |
| 3.3.3 | Papierkorb .....                               | 8  |
| 3.4   | Zusatzfunktionalitäten .....                   | 9  |
| 3.4.1 | Download Stacks .....                          | 9  |
| 3.4.2 | Upload Stacks .....                            | 9  |
| 4     | Restriktionen .....                            | 10 |
| 4.1   | ACID .....                                     | 10 |
| 4.2   | CRUD .....                                     | 10 |
| 4.3   | CAP .....                                      | 11 |

# 1 Ausgangslage

In diesem Dokument sind alle nötigen Informationen zur Benutzerverwaltung bezüglich Berechtigungen und Möglichkeiten festgehalten und erklärt.

Die Berechtigungen werden stets an Gruppen vergeben und ein Datamanagement-Objekt (mit den beinhalteten Tabellen Card<sup>i</sup>-Table, Stack<sup>ii</sup>-Table und Door<sup>iii</sup>-Table kann vom Besitzer (Ersteller) mit verschiedenen Berechtigungen an verschiedene Gruppen verteilt werden.

Die Berechtigungen werden in den folgenden Abschnitten noch genauer erklärt.

Grundsätzlich werden Berechtigungen auf Stacks vergeben, da somit alle Karten innerhalb eines Stacks problemlos bearbeitet werden können (wenn die benötigten Rechte vorhanden sind).

Des Weiteren wäre es unnötig die Rechte auf Doors zu vergeben, da ja niemals alle Stacks einer Door die gleichen Berechtigungen haben müssen und das Herunterladen von ganzen Doors viel zu viel Zeit und Datenvolumen beanspruchen würde.

## 1.1 Erweiterbarkeit

Die Erweiterbarkeit des Berechtigungskonzeptes ist sicherlich ein wenig eingeschränkt, da die Berechtigung mit bestimmten Attributen in der servergespeicherten Datenbank geregelt wird.

Dennoch können Nachfolger oder freiwillige Weiterentwickler zu jedem Zeitpunkt neue Berechtigungs-Attribute hinzufügen oder die Funktion bzw. die Rechte der bereits gesetzten Berechtigungen bearbeiten.

Somit ist eine Weiterführbarkeit des Programms sichergestellt und auch bei kompletter Änderung des Konzepts (von Seiten der WISS) würde die Neu-Programmierung vergleichsweise relativ wenig Zeit und Aufwand in Anspruch nehmen.

## 2 Berechtigungskonzept

### 2.1 Standardberechtigungen

Die Standardberechtigungen<sup>iv</sup>, welche jeder Benutzer automatisch auf jedes Datamanagement-Objekt erhält, beinhalten die folgenden Punkte:

- Leserechte auf alle Datamanagement-Objekte
- Download-Berechtigung (dazu wird nur Leseberechtigung benötigt)  
Somit wird ein Klon auf der lokalen Datenbank des Benutzers erstellt.
- Lokale Klone können dann jedoch lediglich auf sein eigenes Datamanagement-Objekt und nicht auf das ursprüngliche „Repository“ hochgeladen werden.

### 2.2 Derive-Berechtigung

Die sogenannte Derive-Berechtigung<sup>v</sup> ist eine Erweiterung der Standardberechtigung und enthält dadurch nicht nur die Standardrechte, sondern ebenso die folgenden Berechtigungen:

- Upload des lokalen Klons auf das ursprüngliche Datamanagement-Objekt ist erlaubt.
- Die hochgeladenen Klone werden im ursprünglichen Datamanagement-Objekt als neuen Stack angezeigt.
- Das Integrieren oder auch mergen von lokalen mit servergespeicherten Tabellen bzw. Stacks ist nicht erlaubt.

### 2.3 Teamwork-Berechtigung

Die Teamwork-Berechtigung<sup>vi</sup> ist dazu gedacht, Gruppen die gemeinsame Arbeit an einem Stack zu ermöglichen. Sie ist wiederum eine Erweiterung der Derive-Berechtigung und enthält somit alle Standardberechtigungen, alle Derive-Rechte sowie die im Folgenden aufgelisteten Rechte:

- Upload des lokalen Klons auf das ursprüngliche Datamanagement-Objekt sowie Integration in den bestehenden Stack ist erlaubt.
- Mergen ist bei bestehender Internetverbindung stets erlaubt, einzige Einschränkung ist das untenstehende Lock-System.
- Wenn zu dem Zeitpunkt, an dem ein Merge-Vorgang gestartet wird, bereits ein anderer Merge-Vorgang (von einem anderen Benutzer) auf den gleichen Stack ausgeführt wird, so wird der neu gestartete Merge-Vorgang abgebrochen und der Benutzer wird gebeten dies später (wenn der erste Merge-Vorgang abgeschlossen ist) erneut zu versuchen.

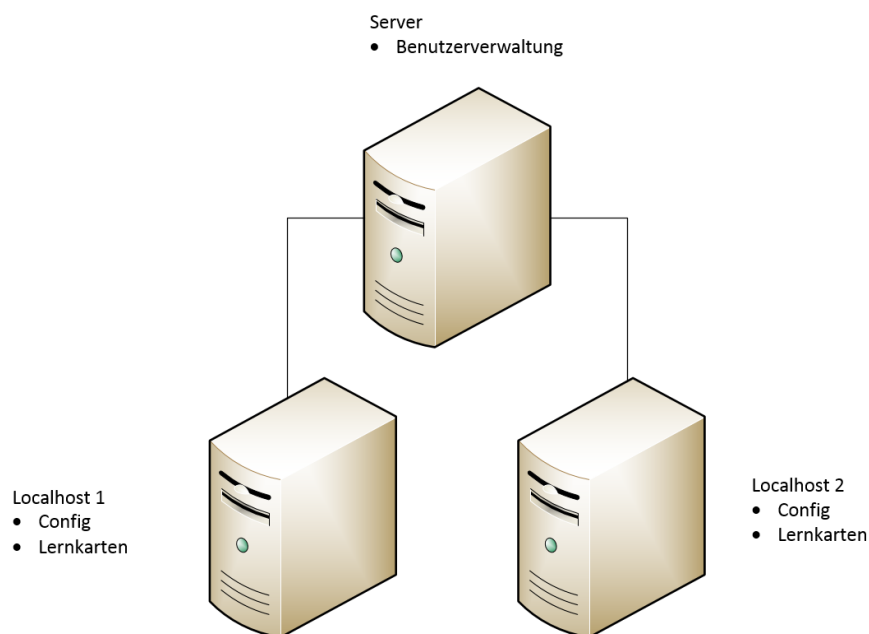
## 3 Datenbankkonzept

### 3.1 Datenbanken

Es handelt sich um eine verteilte Datenbank mit einer servergespeicherten Datenbank (auf dem WISS-Server in Bern) sowie beliebig vielen lokalen Datenbanken (auf den Clients).

Da bei der lokalen und servergespeicherten Datenbanken lediglich die Tabellen „CardTable + DatamanagementID“, „StackTable + DatamanagementID“ und „DoorTable + DatamanagementID“ synchronisiert werden, müssen die anderen Tabellen lokal auch nicht vorhanden sein:

#### Datenbanken der Applikation «WISSLearnCards»

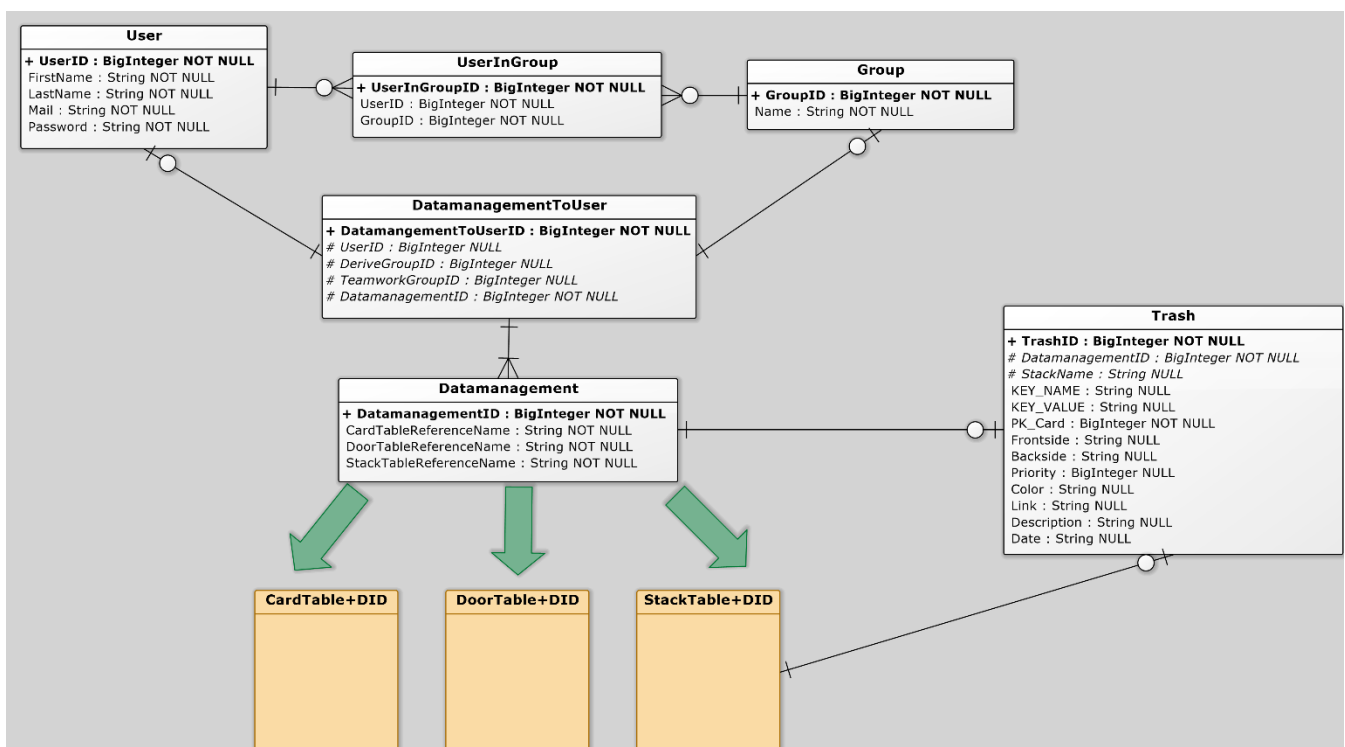


## 3.2 Datenbankstruktur der Benutzerverwaltung

Wir hatten in unserer Datenbankstruktur verschiedenste Faktoren zu berücksichtigen. Einige Fragen, die wir uns stellen mussten waren:

- Wie soll der Zugriff geregelt sein?
- Welche Funktionalitäten wollen wir?
- Welche Tabellen sind dazu geeignet?

Wir haben die Lösungsvariante aus dem oben erwähnten Dokument nochmals überarbeitet, dieses Mal jedoch mit einer richtigen Planung im Voraus.



### 3.2.1 Erläuterungen zur DB-Struktur

Die Grundidee unserer Datenbank-Struktur ist die folgende:

- Der Zugriff auf Cards / Stacks / Doors wird in einem zentralen Manager gespeichert, jedoch vom Programm selber verwaltet (durch Berechtigungs-Abfragen).
- Jeder Benutzer erhält jeweils eine eigene Cards-, Stacks- und Doors-Tabelle
- Jede Cards-, Stacks- und Doors-Tabelle hat einen eindeutigen Namen (Tabellenname + DatamanagementID)
- Firmen / Klassen können einen gemeinsamen Zugriff mittels Gruppen regeln

In unserer Lösungsvariante sind diese Funktionalitäten so geregelt, dass die Tabellen *Datamanagement* und *DatamanagementToUser* den Zugriff auf die oben beschriebenen Ebenen beinhalten.

Die Tabelle *Datamanagement* verweist immer jeweils auf eine Cards-, Stacks- und Doors-Tabelle, auf welche mit der Tabelle *DatamanagementToUser* der Zugriff gesetzt werden kann.

Das Zugriffskonzept ist im oberen Abschnitt (Siehe Punkt 2 Berechtigungskonzept) festgehalten und erklärt.

Als Benutzername in der Tabelle *User* wird die E-Mail Adresse des Benutzers verwendet, da diese in jedem Falle eindeutig ist und somit eine Überprüfung (auf Eindeutigkeit) recht einfach zu implementieren ist.

In der Tabelle *UserInGroup* wird die Gruppenzugehörigkeit aller User geregelt. Somit kann ein Benutzer sowohl in mehreren Gruppen Mitglied sein, als auch mehrere Benutzer in einer Gruppe.

Des Weiteren wird so der Benutzer nicht dazu gezwungen in einer Gruppe Mitglied zu sein, und es werden keine unnötigen Gruppen erstellt.

Auch Gruppennamen müssen in jedem Falle eindeutig und einmalig sein.

### **3.3 Wichtiges**

#### **3.3.1 Nachfolge eines DMO**

Da DMOs sowohl von einzelnen Benutzern als auch im Zuge der Derive- und Teamwork Berechtigungen von anderen Personen genutzt werden können, muss der Besitzer eines Objekts bei dessen Abgabe einen Nachfolger bestimmen der die Verantwortung übernimmt und somit die jeweiligen Berechtigungen zur Manipulation erhält. Wird kein Erbe ernannt, wird das DMO gelöscht.

#### **3.3.2 Eindeutigkeit**

Sowohl beim Benutzernamen, als auch beim Gruppennamen und bei den Tabellennamen (der Tabellen CardTable+DID<sup>vii</sup>, StackTable+DID und DoorTable+DID) muss die Eindeutigkeit in jedem Falle eingehalten und überprüft werden.

Ansonsten können Fehler auftreten, welche die ganze Benutzer-Funktionalität unbrauchbar machen und somit viel Zeit und Arbeit kosten würden.

#### **3.3.3 Papierkorb**

Wir haben einen Papierkorb erstellt, in dem die gelöschten Karten kurz zwischengespeichert werden, damit diese bei versehentlichem Löschen wiederhergestellt werden können.

Damit dieser Papierkorb aber nicht zu voll wird, werden die Einträge vom Programm automatisch einen Monat nach dem Eintragen im Trash, aus der Tabelle gelöscht.

Aus dem Trash können sowohl der Besitzer eines Datamanagement-Eintrages, als auch die Mitglieder der TeamworkGroup des entsprechenden Eintrages in der Datamanagement-Tabelle, Stacks bzw. Cards wiederherstellen.



### 3.4 Zusatzfunktionalitäten

Die folgenden Zusatzfunktionalitäten, werden nun quasi zu Pflichtaufgaben, da Herunterladen und Hochladen für die Grundfunktionalität der Benutzerverwaltung benötigt sind.

#### 3.4.1 Download Stacks

Hat der Benutzer (zumindest) Lese-Rechte auf ein Stack-Element, so soll er dieses auch herunterladen und lokal abspeichern können.

Die neuen Daten werden in die bestehende lokale Datenbank integriert.

Diese Funktionalität wird dadurch erleichtert, dass der Benutzer seine eigenen Elemente (bei denen er als User mit der UserID Zugriff hat), durch die bereits bestehende Gliederung in drei Tabellen viel einfacher herunterladen und somit integrieren kann.

#### 3.4.2 Upload Stacks

Genau wie bei der Download-Funktion, würde nun hier ein bestimmter Stapel vom Benutzer in seinen Datamanagement-Eintrag (in die entsprechende Tabelle) integriert werden.

Auch hierbei erleichtert die von uns gewählte Datenbank-Struktur die Arbeit deutlich.

Ebenso ist eine Merge-Funktion geplant, bei der man einen Stack auch in das ursprüngliche Datamanagement-Objekt hochladen kann.

Durch die Upload-Funktion, kombiniert mit der Download-Funktion wäre es dann möglich z.B. einen Quizlet-Stapel herunterzuladen, diesen lokal zu bearbeiten oder zu verändern und anschliessend in die eigene (oder Gruppen-interne) Struktur auf den WISS-internen Server hochzuladen.

## 4 Restriktionen

### 4.1 ACID

Atomarität wird in unserem Projekt angewandt, da es keinen Sinn macht, einen halben Stack herunterzuladen und bei Unterbrechung der Internetverbindung diesen Teil abzuspeichern. Würde dies gemacht, so müsste der Benutzer den Stack zuerst löschen und anschliessend neu herunterladen.

Konsistenzerhaltung wird in den WISSLearnCards mittels Referentiellen Integritätsbedingungen (siehe Referentielle Integritätsbedingungen.docx) auf der Datenbank.

Diese Bedingungen werden beim Erstellen mittels SQL-Befehlen direkt ins DBMS (Datenbank-Management-System) implementiert und von diesem Zeitpunkt an auch umgesetzt.

Isolation wird in unserem Projekt mit einem „Lock“-Mechanismus implementiert. Dadurch ist es dem Benutzer nicht möglich, Änderungen in einem Stack abzuspeichern, solange ein anderer Benutzer am „schreiben“ bzw. hochladen ist. Somit soll das Phantomproblem vermieden werden.

Dauerhaftigkeit wird nicht berücksichtigt, da Backups der servergespeicherten Datenbank nicht erstellt werden sollen. Dies müsste ausserdem vom Server übernommen werden und unsere Applikation soll so viel wie möglich selbst machen. Somit sind die einzigen „Sicherstellungen“ die lokalen Klone der servergespeicherten Daten.

### 4.2 CRUD

#### CREATE:

- Die servergespeicherte Datenbank darf (sofern sie noch nicht existiert) von jedem Benutzer erstellt werden
- Die Tabellen der servergespeicherten Datenbank dürfen (sofern sie noch nicht existieren) von jedem Benutzer erstellt werden
- Die Tabellen „CardTable + DatamanagementID“, „StackTable + DatamanagementID“ und „DoorTable + DatamanagementID“ in der servergespeicherten Datenbank werden automatisch vom Programm erstellt, sobald ein neuer Datamanagement-Eintrag erstellt wird.

#### READ:

- Leserecht auf die Tabellen der servergespeicherten Datenbank hat jeder Benutzer.

#### UPDATE:

- Auf die servergespeicherten Tabellen User, UserInGroup, Group und DatamanagementToUser hat jeweils nur der Besitzer (UserID in der jeweiligen Tabelle) UPDATE-Zugriff.
- Auf die servergespeicherten Tabellen „CardTable + DatamanagementID“, „StackTable + DatamanagementID“ und „DoorTable + DatamanagementID“ hat sowohl der Besitzer (UserID in DatamanagementToUser), als auch die Mitglieder der TeamworkGroup des DatamanagementToUser UPDATE-Berechtigung.
- Die servergespeicherte Tabellen Datamanagement und Trash werden vom Programm automatisch aktualisiert.

#### DELETE:

- Auf die servergespeicherten Tabellen „CardTable + DatamanagementID“, „StackTable + DatamanagementID“ und „DoorTable + DatamanagementID“ hat jeweils nur der Besitzer (UserID in DatamanagementToUser) Lösch-Berechtigung.
- Die servergespeicherte Trash-Tabelle wird vom Programm automatisch jeden Monat gelöscht bzw. geleert.
- Auf die servergespeicherte Tabellen User, UserInGroup, DatamanagementToUser hat der Benutzer nur Lösch-Berechtigung, sofern seine UserID mit der in den Tabellen referenzierenden UserID übereinstimmt.
- Auf der servergespeicherten Group-Tabelle hat jeder Benutzer Lösch-Berechtigung, der in der Tabelle UserInGroup als Gruppenmitglied der zu entfernenden Gruppe vermerkt ist.
- Auf die servergespeicherte Tabelle Datamanagement hat nur der Besitzer (UserID in DatamanagementToUser) Lösch-Berechtigung. Wenn der Benutzer seinen Account löscht, so muss er einen Nachfolger bestimmen, ansonsten werden alle seine Daten unwiderruflich gelöscht.

### 4.3 CAP

Die Schwerpunkte in der Benutzerverwaltung der WISSLearnCards werden auf die Konsistenz sowie auf die Verfügbarkeit gesetzt.

Konsistenz ist im Programm sehr wichtig, da sichergestellt werden muss, dass die Berechtigungen immer auf dem neusten Stand sind. Benutzer dürfen keinen Zugriff haben, nur weil sie bei der Anmeldung noch die entsprechenden Berechtigungen hatten.

Verfügbarkeit ist insofern wichtig, dass alle Anfragen an den Server bzw. an die Serverdatenbank stets bearbeitet werden. Eine schnelle Antwortzeit kann nicht garantiert werden.

Die Verfügbarkeit der Daten ist besonders bei Gruppenarbeiten unablässig.

Partitionstoleranz ist in unserem Projekt ein weniger wichtiger Punkt, da dies nur vom WISS-Server abhängt. Da wir nichts direkt auf dem Server machen, sind alle Daten die lokal gespeichert werden in jedem Falle sicher, und Daten auf dem Server sind nach dem Neustart des Servers wieder verfügbar, sofern keine Daten gelöscht wurden.

---

<sup>i</sup> Siehe im Dokument Begriffserklärung unter Punkt 2 mit dem Referenznamen *Card*

<sup>ii</sup> Siehe im Dokument Begriffserklärung unter Punkt 2 mit dem Referenznamen *Stack*

<sup>iii</sup> Siehe im Dokument Begriffserklärung unter Punkt 2 mit dem Referenznamen *Door*

<sup>iv</sup> Siehe im Dokument Begriffserklärung unter Punkt 2 mit dem Referenznamen *Standardberechtigungen*

<sup>v</sup> Siehe im Dokument Begriffserklärung unter Punkt 2 mit dem Referenznamen *Derive-Berechtigungen*

<sup>vi</sup> Siehe im Dokument Begriffserklärung unter Punkt 2 mit dem Referenznamen *Teamwork-Berechtigungen*

<sup>vii</sup>