# LU06a - Berechtigungskonzepte

## Lernziele

- Die drei Varianten von Berechtigungskonzepten nennen und erläutern können.
- Zu jeder der drei Varianten mindestens ein konkretes Produkt/Anwendung nennen können.
- Die Datengrundlage (Datenbankmodell) für jedes Konzept aufzeichnen und erläutern können.

## Einleitung

In modernen IT-Systemen greifen viele Benutzer gleichzeitig auf Daten, Anwendungen und Dienste zu. Damit nicht jeder alles darf, braucht es Berechtigungskonzepte. Sie legen fest, wer welche Ressourcen nutzen, ändern oder verwalten darf. Ziel ist es, Sicherheit, Datenschutz und Stabilität zu gewährleisten, ohne die Arbeit unnötig zu behindern.

Ein gutes Berechtigungskonzept sorgt für:

- Schutz sensibler Informationen (z. B. personenbezogene Daten, Finanzdaten)
- Minimierung von Risiken durch das Prinzip „least privilege“ (nur die Rechte, die wirklich nötig sind)
- Nachvollziehbarkeit von Zugriffen durch klare Rollen und Verantwortlichkeiten

Berechtigungskonzepte sind damit ein zentrales Element von IT-Sicherheit und unverzichtbar in Unternehmen, Verwaltungen und auch in privaten IT-Umgebungen.

## Grundlagen

Ein Berechtigungskonzept beschreibt ein System, in dem die Nutzung von Ressourcen nicht uneingeschränkt möglich ist, sondern eine genaue Definition der Nutzung je Benutzer und Ressource erfolgt. Obwohl ursprünglich aus dem organisatorischen Umfeld kommend haben Berechtigungskonzepte, vor allem im Zusammenhang mit IT-Systemen eine wichtige Bedeutung.

Ressourcen sind hier beispielsweise Daten und Informationen, aber auch die technische Infrastruktur wie Systemzugänge, Speicherplatz, Rechnerleistung, Computerprogramme usw.

**Merke:** Ein Berechtigungskonzept dient dem Schutz eine Ressource vor Veränderung oder Zerstörung, verhindert aber auch ihren unrechtmäßigen Gebrauch.

Schützenswerte Ressourcen können beispielsweise Hardware sein:

- Computer
- Rechenzeit
- Prozess-Priorisierung
- Drucker

Aber auch Software oder Daten müssen oft geschützt werden:

- Datenbankobjekte wie Datenbanktabelle
- Files
- Ordner
- Zugriffsart

*Schutz* bedeutet also die Ressourcen vor Veränderung oder Zerstörung (z. B. Datensicherheit) zu schützen. Es bedeutet aber auch zu ihren unrechtmässigen Gebrauch (z. B. Datenschutz) zu verhindern.

# Varianten

Es gibt unterschiedliche Kriterien für Benutzerkonzepte. Im vorliegenden Fall wollen wir uns auf die drei Elemente *Rollen*, *Gruppen* und *Rollen* konzentrieren.

## Benutzerebene (User-Based Access Control – UBAC)

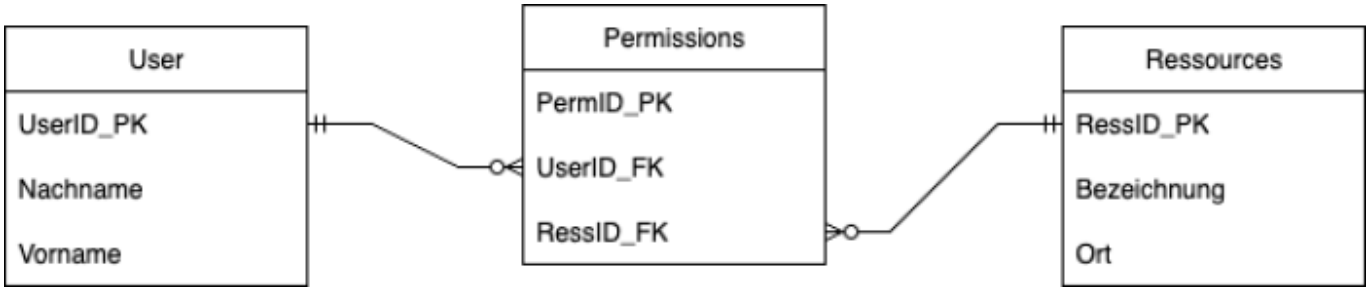
Bei diesem Konzept werden die Rechte direkt einem Benutzerkonto zugewiesen.

- **Vorteile:**
  - einfach zu verstehen
  - individuell anpassbar
  - Kontrolle der Berechtigung ist direkt beim Owner
- **Nachteile:**
  - Unübersichtlich und schwer wartbar, wenn viele Benutzer existieren
  - Die Koordination der Berechtigungen wird komplex und aufwendig

**Beispiel:**

- Benutzer Max Mustermann darf Ordner C:\\Projekte lesen und schreiben.
- Benutzer Anna Müller hat nur Leserechte.

Rein benutzerbezogene Konzepte neigen zur Unübersichtlichkeit und sind deshalb oft nur rudimentär ausgeprägt. Das nachfolgende ERD einer Benutzersteuerung zeigt eine mögliche Umsetzung auf reiner Benutzerebene, bei dem die Berechtigung in einer eigenen Datentabelle verwaltet wird. Denkbar wäre es auch die Rechte direkt in der *User-Tabelle* zu verlagen.



## Gruppenbezogene Zugriffssteuerung

Besser ist ein Konzept über Benutzergruppen. Damit lassen sich Berechtigungen zusammenfassen, beispielsweise alle Berechtigungen, die Mitarbeiter in der Personalbuchhaltung benötigen, wie es sich aus den dortigen Geschäftsprozessen eben ergibt.

Das Betriebssystem UNIX ist ein Beispiel eine Benutzerverwaltung über Gruppen. Für jedes Objekt (Prozess, Datei, Verzeichnis, etc.) werden im Kern drei Gruppen zugelassen. Diese lassen sich beispielweise durch den Befehl ls –lrt ermitteln:

- Eigentümer (user)
- Gruppe (group)
- Die restliche Welt/Sonstige (others)

Jedem der eben genannten Objekte können via dem shell-Befehl *chmod* drei Arten von Berechtigungen gegeben werden:

- Lesen (r=read)
- Schreiben (w=write)
- Ausführen (x=execute)

```
jane@daisy > ~ $ ll
total 48
-rw-r--r-- 1 jane jane   44 Nov 10 12:37 contacts2
-rw-r--r-- 2 jane jane   39 Nov 10 12:33 contacts-link
drwxr-xr-x 3 jane jane 4096 Nov 16 08:34 Desktop/
drwxr-xr-x 2 jane jane 4096 Nov 10 22:46 Documents/
drwxr-xr-x 2 jane jane 4096 Feb 26  2007 Movies/
drwxr-xr-x 2 jane jane 4096 Feb 25  2007 Music/
drwxr-xr-x 3 jane jane 4096 Nov 10 20:45 mydir/
drwxr-xr-x 2 jane jane 4096 Nov 10 12:06 mydir1/
-rw-r--r-- 1 jane jane   69 Nov 10 12:10 newfile
-rw-r--r-- 1 jane jane 2174 Nov 10 12:12 newfile2
drwxr-xr-x 2 jane jane 4096 Nov 11 00:38 Pictures/
-rw-r--r-- 1 root root 3084 Nov 13 06:56 test.txt
lrwxrwxrwx 1 jane jane    5 Nov 12 12:39 tmp -> //tmp/
```

Neben der symbolischen Darstellung (z.B. rwxrwxr-x) gibt es auch noch eine oktale Darstellung. Die Grundrechte (Lesen, Schreiben, Ausführen) und Kombinationen daraus werden hierbei durch eine einzelne Ziffer repräsentiert und dem Eigentümer, der Gruppe und allen anderen zugeordnet. Je nach Anwendung wird dabei von unterschiedlichen Grundwerten ausgegangen und entweder Rechte gegeben oder entzogen.




Bei *chmod* wird beispielsweise von der Grundeinstellung *keine Rechte* (000) ausgegangen und Rechte gegeben, wohingegen bei *umask* von *alle Rechte vorhanden* (777) ausgegangen und Rechte entzogen werden. Entsprechend sind die Werte je nach Anwendung anders.

| Oktal | Klartext der Berechtigung | chmod | umask | Symbolisch | Binär |
|---|---|---|---|---|---|
| 0 | Keine Rechte | chmod 000 | umask 777 | — | 000 |
| 1 | Ausführen | chmod 100 | umask 666 | –x | 001 |
| 2 | Schreiben | chmod 200 | umask 555 | -w- | 010 |
| 3 | Schreiben + Ausführen | chmod 300 | umask 444 | -wx | 011 |
| 4 | Lesen | chmod 400 | umask 377 | r– | 100 |
| 5 | Lesen + Ausführen | chmod 500 | umask 277 | r-x | 101 |
| 6 | Lesen + Schreiben | chmod 600 | umask 177 | rw- | 110 |
| 7 | Lesen + Schreiben + Ausführen | chmod 700 | umask 077 | rwx | 111 |

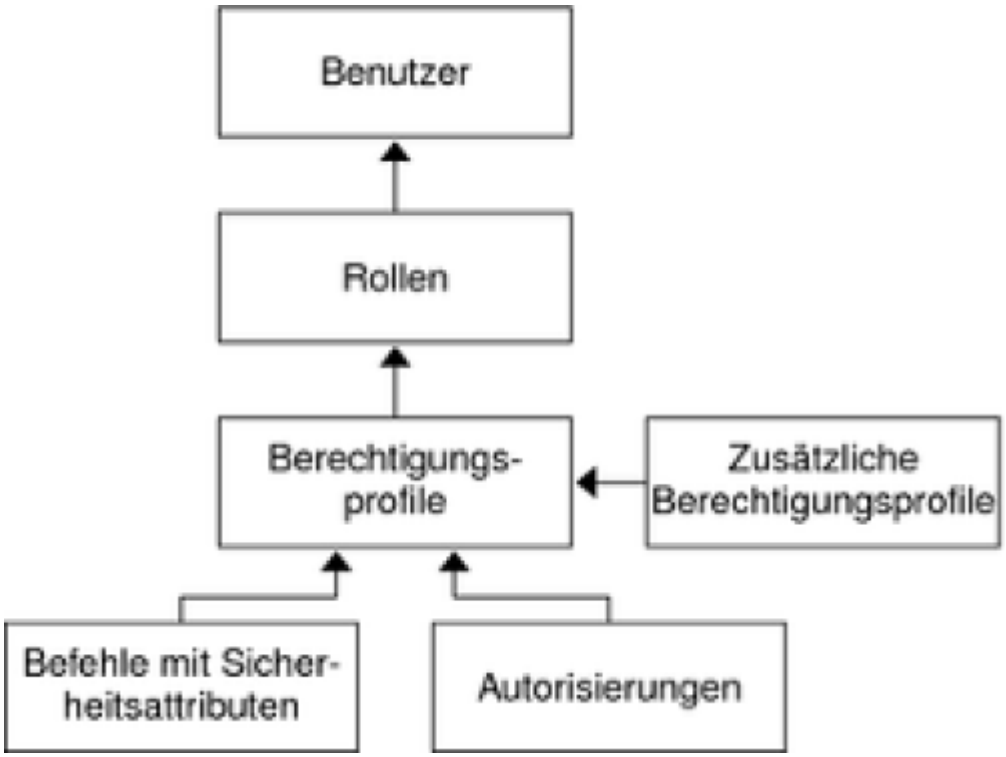
**Erklärung der Spalten**

- **Oktal:** Zahl, die für die Berechtigungen steht (0–7).
- **Klartext:** Welche Rechte gemeint sind (Lesen, Schreiben, Ausführen).
- **chmod:** So setzt man die Rechte direkt.
- **umask:** Die „Maske“, die Standardrechte bei Datei-/Ordnererstellung einschränkt (je größer der Wert, desto weniger Rechte).
- **Symbolisch:** Typische Schreibweise in ls -l.
- **Binär:** Entspricht den Bits (r=4, w=2, x=1).

## Rollenbezogene Zugriffskontrolle

Das Verwalten der Benutzerrechte via Rollen erweitert die Gruppenverwaltung. Jedem Mitarbeiter, der nun konkret in der Personalbuchhaltung arbeitet, wird diese Rolle zugeordnet. Ein Mitarbeiter kann aber durchaus mehrere Rollen haben, wenn er mehrere Funktionen bekleidet. Auf diese Weise wird

erreicht, dass sowohl Veränderungen in den Zuständigkeiten der einzelnen Mitarbeiter als auch Veränderungen im Geschäftsprozess, nur an jeweils einer Stelle im Berechtigungskonzept nachvollzogen werden müssen, und dieses konsistent und überschaubar bleibt.




## Vergleich der 3 Konzepte

| Ebene | Fokus / Definition | Vorteile | Nachteile | Beispiel | Aufwand |
|---|---|---|---|---|---|
| **Benutzerebene** | Rechte werden direkt einem einzelnen Benutzerkonto zugewiesen. | Sehr feingranular, individuelle Steuerung möglich | Extrem unübersichtlich bei vielen Usern, Redundanz | Benutzer Anna darf Ordner C:\\Projekte lesen/schreiben | Linear (O(n)) – jeder Benutzer muss einzeln gepflegt werden |
| **Rollenebene** | Rechte werden Rollen (z. B. „Admin“, „Mitarbeiter“) zugeordnet, Benutzer erhalten Rollen. | Gut skalierbar, klare Abbildung von Funktionen | Rollenmodell muss durchdacht sein, sonst Chaos | Rolle Buchhaltung darf Rechnungen bearbeiten | Annähernd konstant (O(1)) pro User nach Einrichtung – Hauptaufwand liegt in der Modellierung (einmalig) |

| Ebene | Fokus / Definition | Vorteile | Nachteile | Beispiel | Aufwand |
|---|---|---|---|---|---|
| **Gruppenebene** | Benutzer werden zu Gruppen (z. B. „HR“, „IT“) zusammengefasst, Berechtigungen gelten für alle Mitglieder. | Sehr einfach bei großen Organisationen, organisatorisch naheliegend | Grobkörnig, wenig flexibel bei abweichenden Einzelrechten | Gruppe Marketing darf auf Laufwerk M: zugreifen | Logarithmisch bis linear (O(log n) – O(n)) – Verwaltung nach Abteilungen einfacher, aber nicht so präzise wie Rollen |

# Quellennachweis

## Grundlagen

- http://de.wikipedia.org/wiki/Berechtigungskonzept
- http://de.wikipedia.org/wiki/Role_Based_Access_Control
- http://de.wikipedia.org/wiki/Zugriffsrecht

## Technische Dokumentation

- http://technet.microsoft.com/de-de/library/cc721640%28v=office.15%29.aspx
- http://de.wikipedia.org/wiki/Unix-Dateirechte
- http://de.wikipedia.org/wiki/Chmod
- http://wiki.ubuntuusers.de/Rechte
- http://de.wikipedia.org/wiki/Darwin_%28Betriebssystem%29
- http://wiki.ubuntuusers.de/MySQL_Workbench
- http://de.wikibooks.org/wiki/Oracle:_Benutzerverwaltung
- http://help.sap.com/printdocu/core/print46c/de/data/pdf/BCCCMUSR/BCCCMUSR.pdf
- http://www.linupedia.org/opensuse/Zugriffsrechte

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
**https://wiki.bzz.ch/modul/m183/learningunits/lu06/01?rev=1756806656**

Last update: **2025/09/02 11:50**