

# LIFI Live Feedback

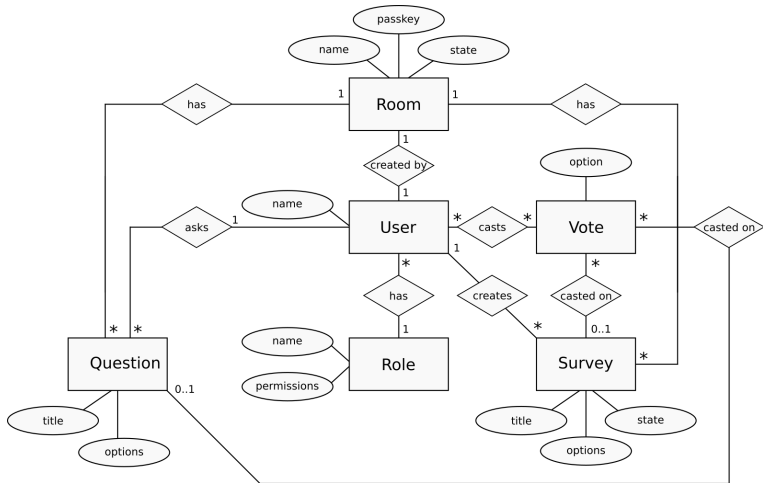
## Meilenstein 3

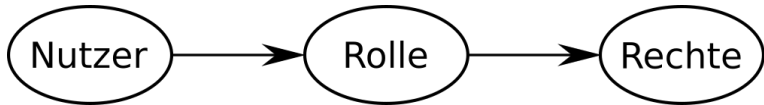
Gruppe 5 (Paul Walger, Sebastian Götte)

TU Berlin

17. Juli 2015

# Datenmodell





# Rollen

## Teilnehmer

view\_index  
view\_room  
create\_question  
join\_lecture  
vote\_tempo  
vote\_question  
vote\_survey  
list\_permissions

## Veranstalter

manage\_lecture  
create\_survey  
create\_room  
view\_tempo  
close\_survey  
delete\_question  
delete\_survey

## Administrator

create\_account  
delete\_account  
assign\_role  
create\_role  
edit\_role  
delete\_role  
list\_roles  
list\_users

# Serversoftware

- Postgresql
- Python 3
- nginx

# Python-Server

- Flask (HTTP-Microframework)
- SQLAlchemy (ORM)

# Authentifizierung

## auth-Decorator

```
@app.route('/api/r/<room_name>')
@auth
@room
def view_room(room):
    return jsonify(name=room.name,
                    questions=[ q.as_dict() for q in
                               ↪ room.questions ],
                    surveys=[ sv.as_dict() for sv in
                              ↪ room.surveys ],
                    user_is_lecturer=(room.creator ==
                                       ↪ request.user))
```

# Unit-Testing

## test\_for-Decorator

```
@test_for('list_rooms', """:HTTP method:    GET
:Response JSON:  '{"rooms": ["some_room",
↪  "another_room"]}' """)
def testListRooms(self):
    with testserver() as url:
        cred = self.login(url, 'user1')
        r = rq.get(url+'list_rooms', cookies=cred)
        self.json(r, {'rooms': {'test_room_access',
↪  'test_room_deny'}})
```



# Api-Dokumentation

## Automatische Dokumentation in rst, tex und pdf

```
list_rooms
=====
:Routes:
    '/api/list_rooms'
:HTTP method:  GET
:Response JSON: '{"rooms": ["some_room",
↪  "another_room"]}'
```