

# Die barocken Schloss- und Gartenveduten

AR

2024-06-15



# Table of contents

1	Katalog zur Ausstellung: Die barocken Schloss- und Gartenveduten	1
2	Die barocken Schloss- und Gartenveduten	3



## Chapter 1

# Katalog zur Ausstellung: Die barocken Schloss- und Gartenveduten

Ein Katalog mit Kunstwerken aus der CbDD-Sammlung. Textteil: 6e73f774-4b7f-4e37-937b-e11cc35c5bc8

Die barocken Schloss- und Gartenveduten [Sammlung]

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 4.0 International License.



## Chapter 2

# Die barocken Schloss- und Gartenveduten

```
from datetime import datetime
import sys
import time
from SPARQLWrapper import SPARQLWrapper, JSON
import requests
from PIL import Image
import html

endpoint_url = "https://computational-publishing-service.wikibase.cloud/query/sparql"

query_txt = """PREFIX cps: <https://computational-publishing-service.wikibase.cloud/entity/>
PREFIX cpss: <https://computational-publishing-service.wikibase.cloud/entity/statement/>
PREFIX cpsv: <https://computational-publishing-service.wikibase.cloud/value/>
PREFIX cpspt: <https://computational-publishing-service.wikibase.cloud/prop/direct/>
PREFIX cpsp: <https://computational-publishing-service.wikibase.cloud/prop/>
PREFIX cpsps: <https://computational-publishing-service.wikibase.cloud/prop/statement/>
PREFIX cpspq: <https://computational-publishing-service.wikibase.cloud/prop/qualifier/>

SELECT ?textItem ?kuratorLabel ?textUrl
WHERE
{
  <placeholder>
  ?textItem cpsp:P46 ?kuratorStatement.
  ?kuratorStatement cpsps:P46 ?kuratorItem.
  ?kuratorItem rdfs:label ?kuratorLabel.
  ?textItem cpsp:P57 ?urlstatement.
  ?urlstatement cpsps:P57 ?textUrl.
}"""

query_img = """PREFIX cps: <https://computational-publishing-service.wikibase.cloud/entity/>
```

```

PREFIX cps: <https://computational-publishing-service.wikibase.cloud/entity/statemen
PREFIX cpsv: <https://computational-publishing-service.wikibase.cloud/value/>
PREFIX cpspt: <https://computational-publishing-service.wikibase.cloud/prop/direct/>
PREFIX cpsp: <https://computational-publishing-service.wikibase.cloud/prop/>
PREFIX cpsps: <https://computational-publishing-service.wikibase.cloud/prop/statement
PREFIX cpspq: <https://computational-publishing-service.wikibase.cloud/prop/qualifier

SELECT DISTINCT ?itemLabel ?itemDescr ?imgItem ?imgUrl ?publishDate
WHERE
{
  ?imgItem cpsp:P107 ?urlStatement.
  ?urlStatement cpsps:P107 ?imgUrl.
  ?imgItem cpsp:P60 ?dateStatement.
  ?dateStatement cpsps:P60 ?publishDate.
  ?imgItem cpsp:P6 ?partOfStatement.
  ?partOfStatement cpsps:P6 ?partOfItem.
  <placeholder>
  SERVICE wikibase:label {
    bd:serviceParam wikibase:language "de,en".
    ?imgItem rdfs:label ?itemLabel.
    ?imgItem schema:description ?itemDescr.
  }
}"""

query_graph = """PREFIX cps: <https://computational-publishing-service.wikibase.cloud
PREFIX cps: <https://computational-publishing-service.wikibase.cloud/entity/statemen
PREFIX cpsv: <https://computational-publishing-service.wikibase.cloud/value/>
PREFIX cpspt: <https://computational-publishing-service.wikibase.cloud/prop/direct/>
PREFIX cpsp: <https://computational-publishing-service.wikibase.cloud/prop/>
PREFIX cpsps: <https://computational-publishing-service.wikibase.cloud/prop/statement
PREFIX cpspq: <https://computational-publishing-service.wikibase.cloud/prop/qualifier

SELECT ?x ?y
WHERE
{
  ?a cpsp:P2 ?c.
  ?c cpsps:P2 ?d.
  ?a rdfs:label ?x.
  ?d rdfs:label ?y.
}LIMIT 1"""

query_graph2 = """
SELECT ?a ?b ?c
WHERE
{
  ?a rdfs:label ?c
}LIMIT 100"""

```



```

def run_query(endpoint_url, query):
    user_agent = "WDQS-example Python/%s.%s" % (sys.version_info[0], sys.version_info[1])
    # TODO adjust user agent; see https://w.wiki/CX6
    sparql = SPARQLWrapper(endpoint_url, agent=user_agent)
    sparql.setQuery(query)
    sparql.setMethod("POST") #this NEEDS to be added to get results (not included in the wikiba
    sparql.setReturnFormat(JSON)
    return sparql.query().convert()

def get_text(textitem_id):
    q = ""
    if textitem_id:
        q = query_txt.replace("<placeholder>", "cps:"+textitem_id+" cpsp:P46 ?kuratorStatement.")
    else:
        q = query_txt.replace("<placeholder>", "")

    results_txt = run_query(endpoint_url, q)
    for item in results_txt["results"]["bindings"]:
        print('Wikibase link: ' + '[' + item['textItem']['value'] + ']' + '(' + item['textItem']
        print('Kurator: ' + item['kuratorLabel']['value'] + '\n')
        headers = {'User-Agent': 'Ex_Books_conference_bot/0.0 (https://github.com/SimonXIX/Expe
        r = requests.get(item['textUrl']['value'], headers=headers, stream=True)
        text = str(r.text) #changed from r.content
        text = text.replace("ä", "&auml;")
        text = text.replace("Ä", "&Auml;")
        text = text.replace("ö", "&ouml;")
        text = text.replace("Ö", "&Ouml;")
        text = text.replace("ü", "&uuml;")
        text = text.replace("Ü", "&Uuml;")
        text = text.replace("ß", "&szlig;")
        text = text.replace('\n', "<br>")
        text = str(text)
        text = text.removeprefix("<!DOCTYPE html>").removesuffix("") #changed from b'<!DOCTYPE
        print(text)

def get_delay(date):
    try:
        date = datetime.datetime.strptime(date, '%a, %d %b %Y %H:%M:%S GMT')
        timeout = int((date - datetime.datetime.now()).total_seconds())
    except ValueError:
        timeout = int(date)
    return timeout

def fetch_image_by_url(url, headers):
    r = requests.get(url, headers=headers, stream=True)
    if r.status_code == 200:
        im = Image.open(r.raw)
        return im
    if r.status_code == 500:

```

```

        return None
    if r.status_code == 403:
        return None
    if r.status_code == 429:
        timeout = get_delay(r.headers['retry-after'])
        print('Timeout {} m {} s'.format(timeout // 60, timeout % 60))
        time.sleep(timeout)
        fetch_image_by_url(url, headers)

def get_img(partOfItem_id):
    q = ""
    if partOfItem_id:
        q = query_img.replace("<placeholder>", "?partOfStatement cpsps:P6 cps:"+partOfItem_id)
    else:
        q = query_img.replace("<placeholder>", "")
    results_img = run_query(endpoint_url, q)
    for item in results_img["results"]["bindings"]:
        #print(item)
        print('Wikibase link: ' + '[' + item['imgItem']['value'] + ']' + '(' + item['itemLabel']['value'] + ')')
        print('Title: ' + item['itemLabel']['value'] + '\n')
        print('Year: ' + item['publishDate']['value'] + '\n')
        print('Description: ' + html.unescape(item['itemDescr']['value']) + '\n')

        # get image from image URL and display resized version
        image_url=item['imgUrl']['value']
        headers = {'User-Agent': 'Ex_Books_conference_bot/0.0 (https://github.com/SimonSteuernagel/exbooks-conference-bot)'}
        im = fetch_image_by_url(image_url, headers)
        im.thumbnail((500, 500), Image.Resampling.LANCZOS)
        display(im)
        print('\n\n')

def get_graph():
    import VizKG.visualize as vkg
    results_graph1 = run_query(endpoint_url, query_graph)
    #print(results_graph1)
    #print('---')
    results_graph2 = run_query(endpoint_url, query_graph2)
    #print(results_graph2)

    chart = vkg(sparql_query=query_graph2, sparql_service_url=endpoint_url, chart='world')
    chart.plot()

get_text("Q232")

```

Wikibase link: <https://computational-publishing-service.wikibase.cloud/entity/Q232>

Kurator: Seeger, Ulrike

Die Tafelstube

## Beschreibung

Östlich an den Rittersaal schließt ein großer, 1837 unterteilter Raum an, bei dem es sich um die einstige Tafelstube handelt.[1] Als Eckraum mit vier Doppelfenstern zur Gartenseite und weiteren drei Doppelfenstern zur Grabenseite erhielt die Tafelstube viel Licht. Auch konnte der Fürst von dort aus auf die Stadt und den Lustgarten blicken, der in der Renaissance dem Schloss südöstlich vorgelegt war.[2] Gemessen an der Größe des Raumes war die Tafelstube nicht sehr hoch. Die Decke mit kräftigen Unterzügen ruhte ursprünglich auf vier Stützen, deren Position einem Plan des 19. Jahrhunderts zu entnehmen ist. Die Fensternischen waren in Fortsetzung der Saaldekoration mit Roll- und Beschlagwerk stuckiert, wofür Christoph Limmerich in Frage kommt, der auch im Saal gearbeitet hat.

Logistisch gehören zur Tafelstube zwei Service-Kabinetten beiderseits des Durchgangs zwischen Saal und Tafelstube. Sie haben eine geringe Raumhöhe, da über ihnen und dem Durchgang die Empore an der Ostseite des Saals verläuft. Das Kabinett der Gartenseite war von der Tafelstube und vom Durchgang aus zugänglich, das Kabinett der Hofseite außer von der Tafelstube vom Altan aus. Der Altan entlang der Hofseite des Saalbaus verband das hofseitige Kabinett mit der Küche im Erdgeschoss des Küchenbaus, sodass bevorzugt dieses Kabinett dem Anrichten der Speisen gedient haben dürfte. Dank der Verbindung zu dem ja erst in einem zweiten Bauabschnitt errichteten Altan, blieb der Rittersaal vom Transport der Speisen verschont.

Der repräsentative Zugang zur Tafelstube erfolgte vom Saal aus, wo der Besucher das imposante Portal mit der Belagerung von Gran (Eszergom) im Hintergrund einer wilden Türkenschlacht, bekrönt von der Skulptur des heiligen Georg zu durchschreiten hatte. Ein zweiter Zugang bestand oder ließ sich zumindest einrichten von der geradeläufigen Treppe im späteren Langenburger Bau.

## Die ursprüngliche Bezeichnung des Raumes und seine Ausstattung

Im Inventar von 1625–27 wurde der Raum im Anschluss an den Saal als „Saalstube“ bezeichnet.[3] Die Wände waren mit 14 Ledertapeten beschlagen. Im Raum standen zwei längsrechtecke Tische, ein quadratischer Tisch und eine „große Landtafel“ sowie 31 Sessel mit Lederbezügen und goldenem Dekor.[4] Im Schadensinventar von 1639 wurde der Raum sodann als „Große Tafelstube“ geführt.[5]

[1] Die Jahreszahl der Unterteilung: Merten, Weikersheim, o. J., S. 40; Fandrey, Weikersheim, 2010, S. 51.

[2] Münzenmayer/Elfgang, Schlossgarten, 1999, Abb. S. 5.

[3] Die Kenntnis dieses Inventars verdankt die Autorin Dinah Rottschäfer.

[4] Ebd.

[5] HZAN La 130 Bü 152, Schadensinventar von 1639. Die Kenntnis und die Transkription dieser Archivalie verdankt die Autorin Frieder Leipold. Zur Herausbildung der Tafelstube im deutschen Schlossbau der Renaissance: Hoppe, Tafelstube, 2007 ([https://adw-goe.de/en/digital-library/hoefe-und-residenzen-im-spaetmittelalterlichen-reich/gsn/rf15\\_II\\_121207-196/?tx\\_find\\_find%5BunderlyingQuery%5D%5Bq%5D%5Bc](https://adw-goe.de/en/digital-library/hoefe-und-residenzen-im-spaetmittelalterlichen-reich/gsn/rf15_II_121207-196/?tx_find_find%5BunderlyingQuery%5D%5Bq%5D%5Bc)

## Programm und Synthese der einstigen Tafelstube

Tafelstube und Saal hängen konzeptionell eng zusammen. Während der Saal mit der guten Herrschaft des Grafen Wolfgang einen regionalen Radius beschreibt, weitet sich in der Tafelstube der Horizont auf den Beitrag der Grafschaft Hohenlohe zur Rettung der Christenheit vor osmanischer Herrschaft. Räumlich verknüpft sind die beiden Bildprogramme durch das Relief des Innenportals mit der Belagerung von Gran (Eszergom) 1594 und die Deckenmalerei des Durchgangs, die mit der Beweinung des toten Adonis durch Venus und Amor auf den tragischen Tod des jüngsten Sohnes bei der Belagerung von Gran (Eszergom) 1604 vorausweist. Adonis als passionierter Jäger wiederum verband die Tafelstube mit dem Jagdzyklus an der Decke des Saals.

```
get_img("Q208")
```

Wikibase link: <https://computational-publishing-service.wikibase.cloud/entity/Q283>

Title: Die barocken Schloss- und Gartenveduten bild

Year: 2018

Description: Bild für Die barocken Schloss- und Gartenveduten

Wikibase link: <https://computational-publishing-service.wikibase.cloud/entity/Q283>

Title: Die barocken Schloss- und Gartenveduten bild

Year: 2018-01-01T00:00:00Z

Description: Bild für Die barocken Schloss- und Gartenveduten





get\_graph()

