

RESTful Web Services

Stefan Klotz (589152)
Fynn Jeske (589225)
Paul Dubinski (590183)

Ablauf

- Webdienste
- Vorstellung von REST
 - Definition
 - Prinzipien
 - Vorteile
- REST im Kontext verteilter Systeme
- HTTP-Methoden & Postman
- Implementation mit DJANGO

Webdienste

- Kommunikation zwischen Systemen unabhängig von Plattform, Technik und Programmiersprache



- Herausforderungen
 - Effizienz
 - Skalierbarkeit
 - Verständlichkeit

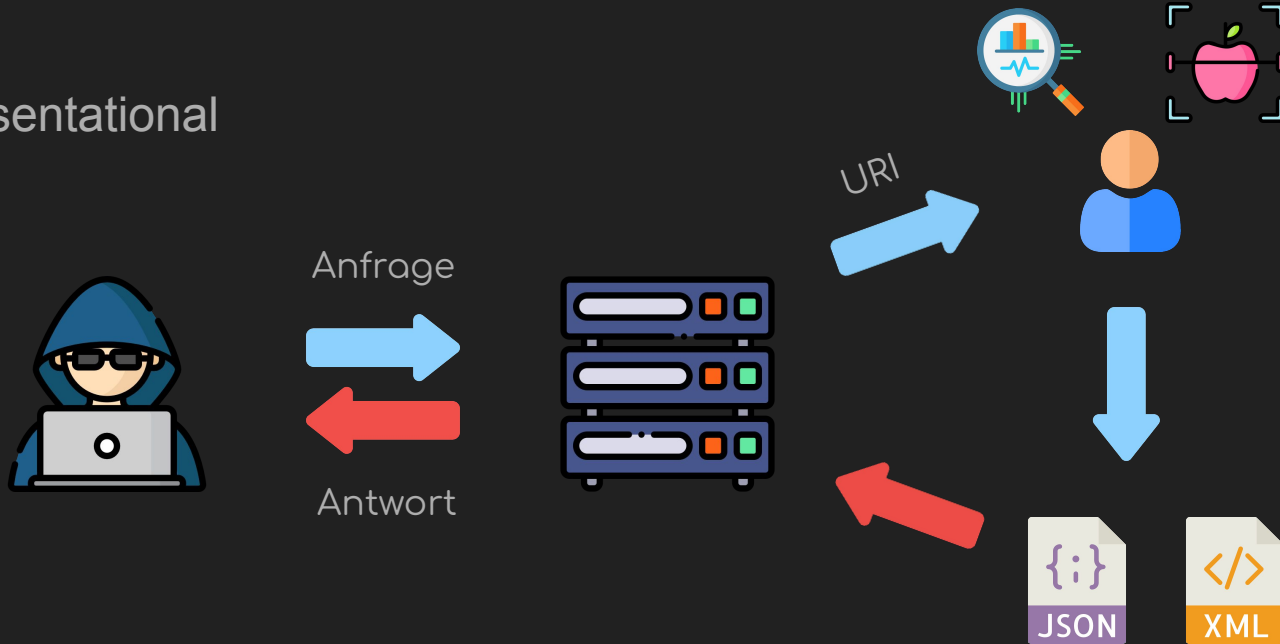
REST

- **Representational State Transfer**
- Roy Fielding, 2000
- Architekturstil für die Software von Webservices [1]



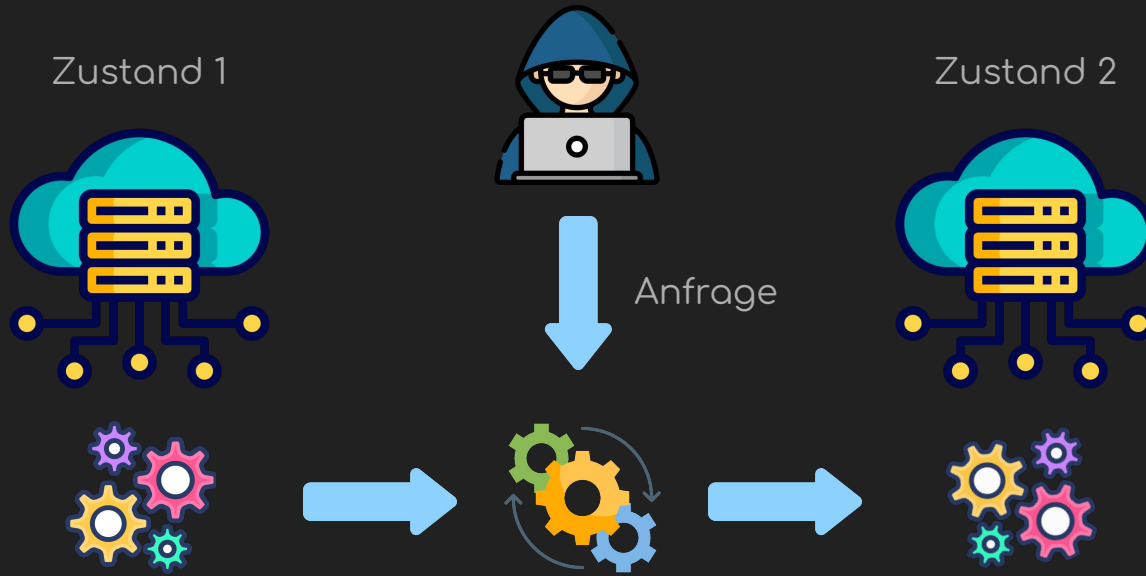
REST

- **Representational**



REST

- State Transfer



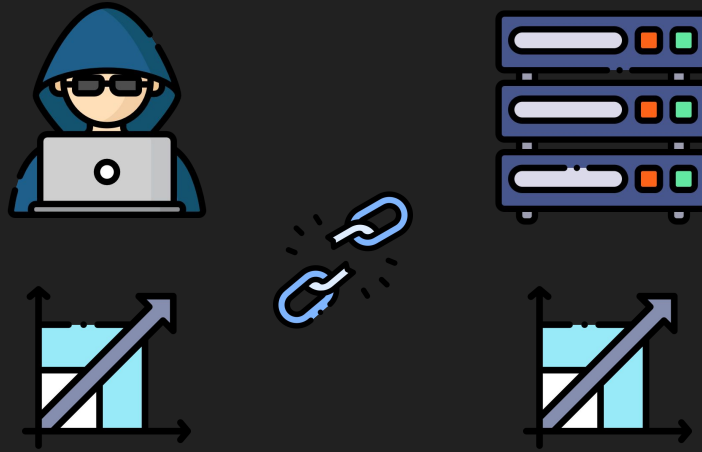
REST

- **Representational State Transfer**
- Client interagiert mit Repräsentationen von Ressourcen
- Anfrage leitet Zustandsänderung ein

Prinzipien von REST

Client-Server

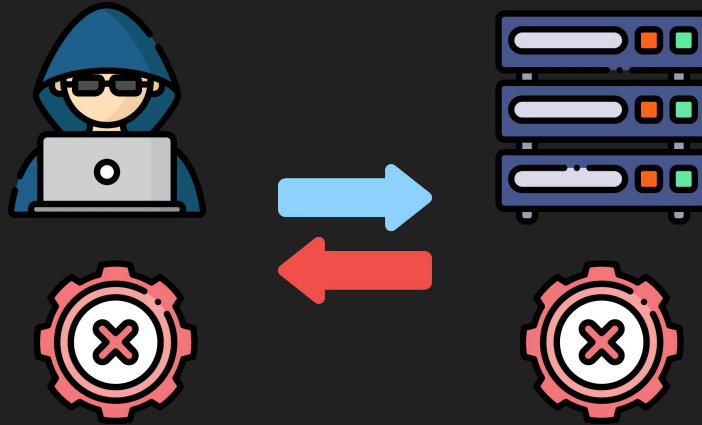
- Separation of Concerns



Prinzipien von REST

Zustandslosigkeit

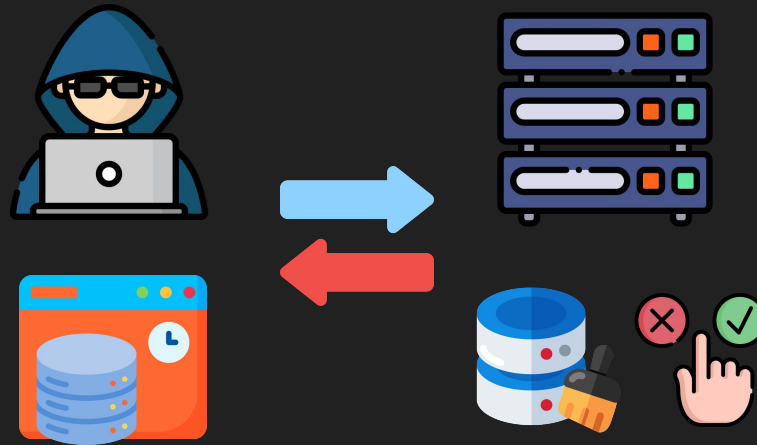
- Client - Server keine Informationen über Zustand des Anderen



Prinzipien von REST

Cache

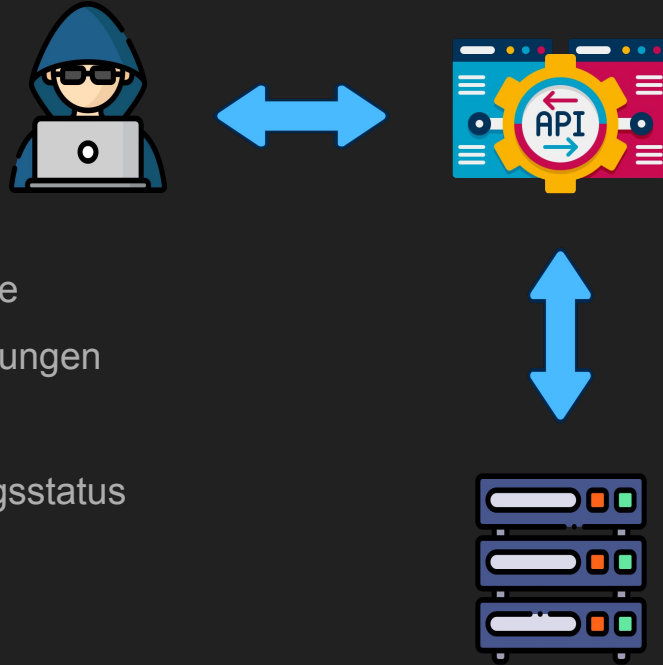
- Clientseitiges Caching



Prinzipien von REST

Einheitliche Schnittstelle

- RESTful APIs
 - Ressourcenidentifizierung in der Anfrage
 - Ressourcenmanipulation durch Darstellungen
 - Selbstschreibende Nachrichten
 - Hypermedia als Engine des Anwendungsstatus



Prinzipien von REST

Schichtensystem

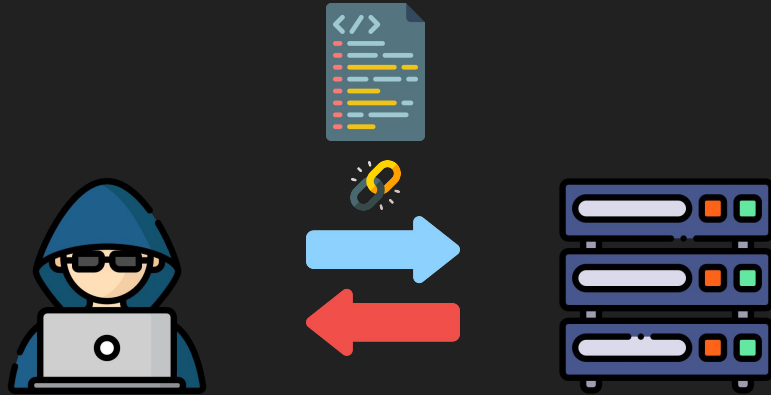
- Systemschichten sind voneinander getrennt



Prinzipien von REST

Code on Demand (optional)

- Übertragung von Code durch API



Zusammenfassung REST

- **Representational State Transfer**
- Client interagiert mit Repräsentationen von Ressourcen
- Anfrage leitet Zustandsänderung ein
- Client-Server
- Zustandslosigkeit
- Cache
- Einheitliche Schnittstelle
- Schichten
- Code on Demand (optional)

Vorteile von REST

- nötige Infrastruktur im WWW vorhanden
- Einfachheit
- Skalierbarkeit
- Flexibilität

REST im Kontext verteilter Systeme

- REST Architektur ist abgeleitet aus verschiedenen Netzwerkarchitektur Stilen [1, p. 76]
- zusätzliche Einschränkungen, z.B.: Client-Server Architektur [1, p. 78]
- Vorteile von RESTful Webservices:
 - Skalierbarkeit: da kein Zustand gespeichert wird
 - Interoperabilität: Nutzung von Standards wie HTTP, JSON und XML
 - Caching: Antworten können clientseitig gespeichert werden

HTTP-Methoden

wichtige HTTP-Methoden [2, pp. 74 - 80] für Restful APIs:

POST:	erstellen einer Ressourcen	Create
GET:	anfordern einer Ressource	Read
PUT:	updaten einer Ressource	Update
DELETE:	löschen einer Ressource	Delete

HTTP-Methoden: Beispiel GET

GET <https://jsonplaceholder.typicode.com/posts?userId=4&id=32>

|

Methode

|

URL

|

Query

Common HTTP Response Codes

- 200: Success
- 301: Permanent Redirect
- 404: Not Found
- 500: Internal Server Error

API-Test mit Postman

- freemium Software, API-Plattform
- hilft beim “designen, testen, dokumentieren und mocking” [3] von APIs
- einfache Möglichkeit HTTP-Requests an eine API zu stellen
- übersichtliche Darstellung der Serverantwort (Response Code und Body)

Implementation in Django

- Django
 - OpenSource Web-Framework
 - in Python
 - entwickelt von Django Software Foundation [4] (non-profit)
 - Verwendet von Instagram [5], Mozilla [6] uvm.
- Implementation zweier Endpunkte für potenzielle ToDo-App
 - Todo Items können abgerufen und angelegt werden

Quellen

- [1] R. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Ph.D. dissertation, University of California, Irvine, 2000.
Available: https://ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf
[Accessed Jan. 14, 2025]
- [2] R. Fielding, J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content," RFC 9110, Internet Engineering Task Force (IETF), June 2022. [Standard specification].
Available: <https://www.rfc-editor.org/rfc/rfc9110.pdf> [Accessed Jan. 14, 2025]
- [3] Postman, "What is Postman?," postman.com. [Info page].
Available: <https://www.postman.com/product/what-is-postman/>. [Accessed Jan. 14, 2025]

Quellen

- [4] Django Software Foundation, "Django Software Foundation,"
Available: <https://www.djangoproject.com/foundation/> [Accessed Jan. 13, 2025].
- [5] Instagram Engineering, "What Powers Instagram: Hundreds of Instances, Dozens of Technologies,"
Available: <https://instagram-engineering.com/what-powers-instagram-hundreds-of-instances-dozens-of-technologies-adf2e22da2ad> [Accessed Jan. 13, 2025].
- [6] Mozilla Developer Network, "Use of Python at Mozilla," Internet Archive,
Available: https://web.archive.org/web/20120208002537/https://developer.mozilla.org/en/Python#Use_of_Python_at_Mozilla [Accessed Jan. 13, 2025].