

Fakultät Umweltwissenschaften
Professur für Geoinformatik

Webbasierte Kartenanwendungen und WebGIS

Rückblick, Ausblick, Projekt

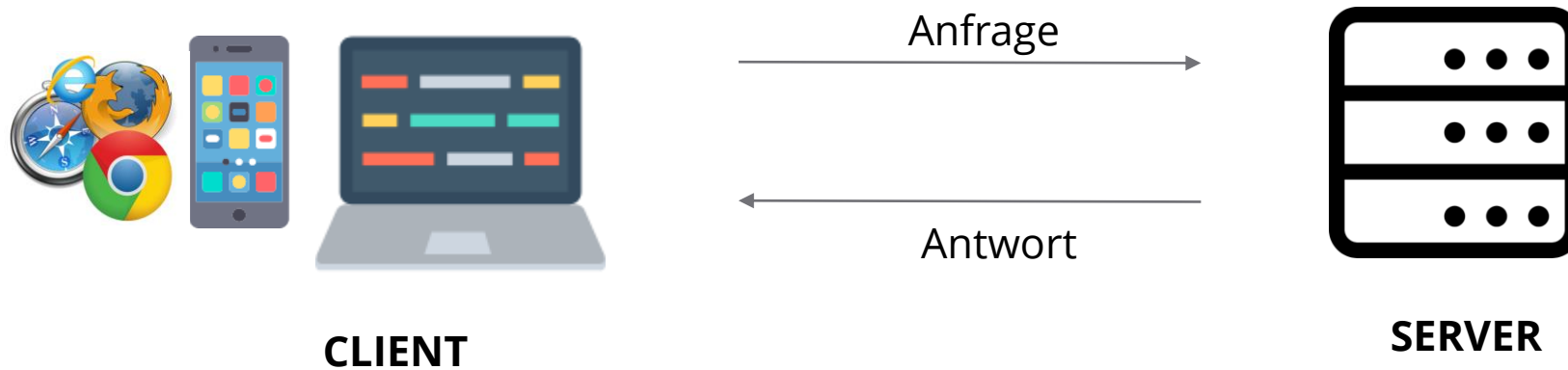
SoSe 2024

Rückblick

... Dieser Kurs vermittelt die **Grundlagen** der **praktischen Entwicklung** von geoinformatischer Software auf Basis der Programmiersprache **JavaScript** ...

„Grundkonzepte der Client-Server Architektur verstehen und Komponente einer Webkarte erlernen; Einführung in Sprachen für die Webentwicklung (HTML, CSS, JavaScript); Programmieraufgabe: eine Webkarte selbst implementieren, mit open-source Bibliotheken, z.B. Leaflet, Bootstrap, Chart.js“

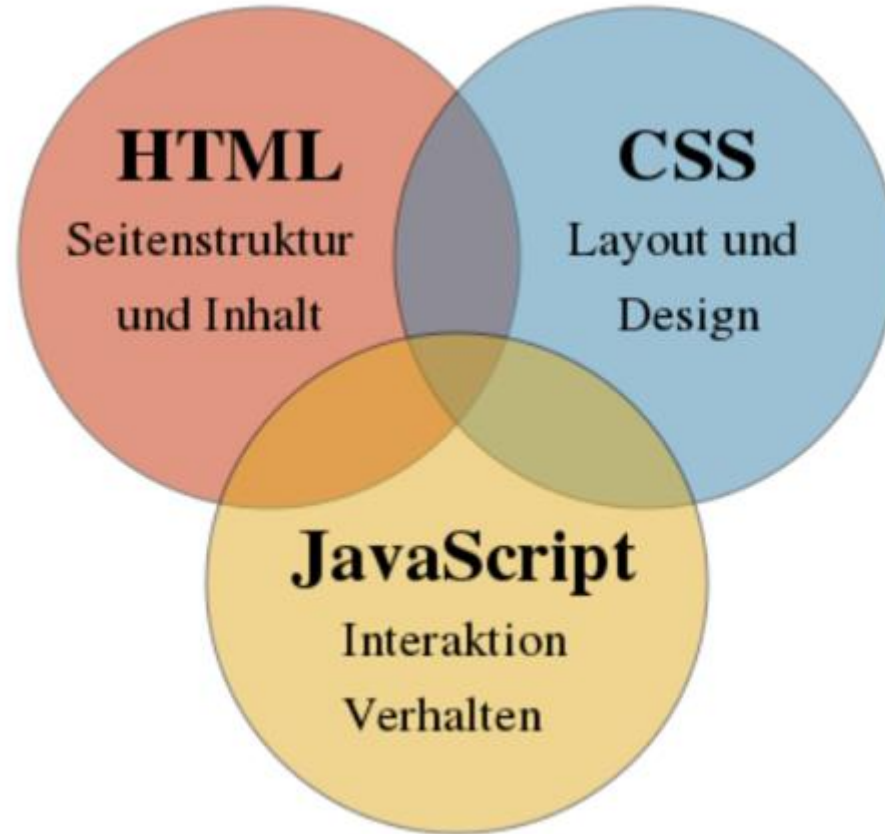
Das Client-Server Modell



- Dient als Schnittstelle zum Benutzer
- Übermittelt Anfragen an den Server
- Bereitet Daten so, dass Sie auf dem Endgerät abgelesen werden können

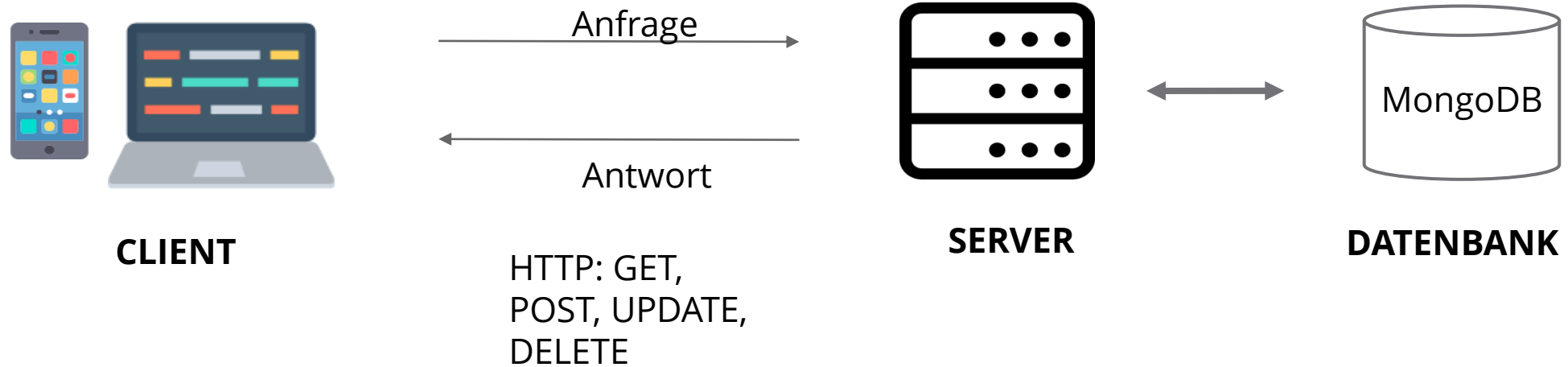
- Nimmt Anfragen vom Client, leitet sie weiter und beantwortet sie
- Stellt einen Dienst (z.B. Daten, Komplexe Berechnung) zur Verfügung

Programmiersprachen für die Webentwicklung



Bildquelle: <https://wiki.selfhtml.org/wiki/Datei:HTML-CSS-JS.svg>

Erweiterte Architektur



Grund: Oft möchte man manche Daten persistent speichern

Überblick der Themen



**VL02: HTML, DOM, Events,
Chart.js, GitHub Pages**

VL03: CSS, Bootstrap

VL00: JSON

**VL01: JavaScript Funktionen,
Arrow Funktionen, OOP**

VL04: Leaflet, Leaflet Draw, Turf.js

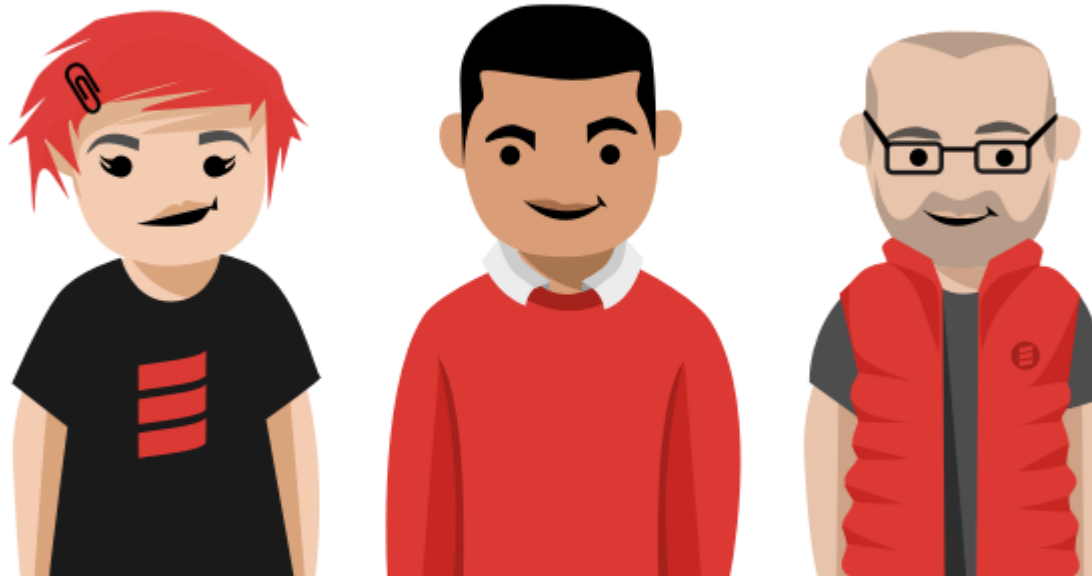
**VL05: Node.js, Routes, Express.js
TemplateEngine(pug),**

APIs, MongoDB

Alternative Formulierung

Aus VL00

... Dieser Kurs vermittelt die **Grundlagen** der **praktischen**
Entwicklung von geoinformatischer Software für werdende
Fullstack **JavaScript** Entwickler:Innen...



What is XML
What is AJAX
What is JSON
What is CSS Icons
What is Bootstrap
What is W3.CSS
What is CLI
What is npm
What is GitHub
What is jQuery
What is AngularJS
What is React
What is Vue.js
What is W3.JS
What is Front-End Dev.
What is Fullstack

What is Fullstack JS

What is SQL

Amazon AWS

What is AWS EC2
What is AWS RDS
What is AWS Cloudfront
What is AWS SNS
What is Elastic Beanstalk
What is AWS Auto Scaling
What is AWS IAM
What is AWS Aurora
What is AWS DynamoDB
What is AWS Personalize
What is AWS Rekognition
What is AWS Quicksight

Full Stack JavaScript Developer

A full stack JavaScript developer is a person who can develop both **client** and **server** software.

In addition to mastering HTML and CSS, he/she also knows how to:

- Program a **browser** (like using JavaScript, jQuery, Angular, or Vue)
- Program a **server** (like using Node.js)
- Program a **database** (like using MongoDB)

Client Software (Front End)

- HTML
- CSS
- Bootstrap
- W3.CSS
- JavaScript
- ES5
- HTML DOM
- JSON
- XML
- jQuery
- Angular
- React
- Backbone.js
- Ember.js
- Redux
- Storybook
- GraphQL
- Meteor.js
- Grunt
- Gulp

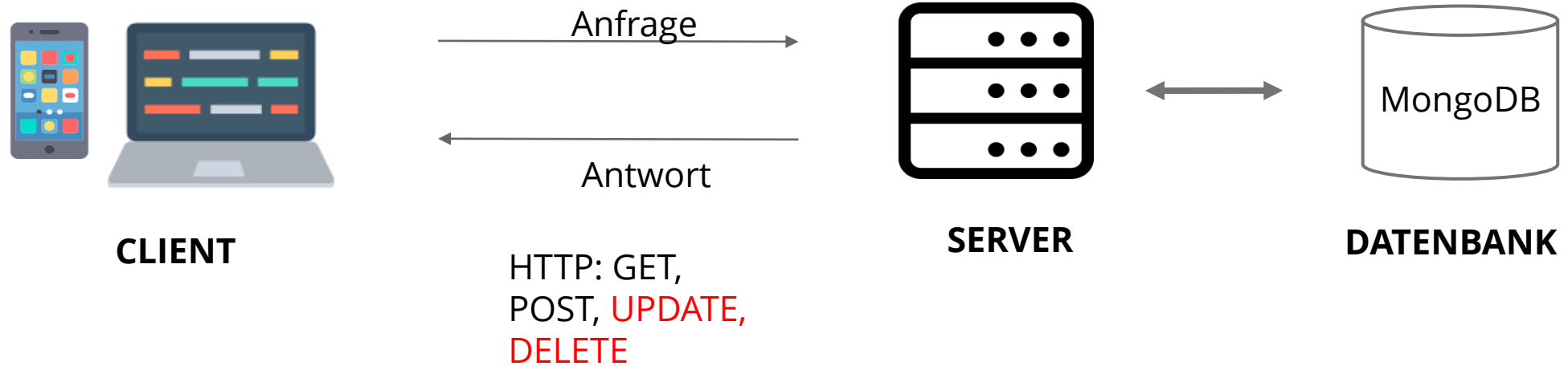
Server Software (Back End)

- PHP
- ASP
- C++
- C#
- Java
- Python
- Node.js
- Express.js
- Ruby
- REST
- GO
- SQL
- MongoDB
- Firebase.com
- Sass
- Less
- Parse.com
- PaaS (Azure and Heroku)



Ausblick

Erweiterte Architektur



Grund: Oft möchte man manche Daten persistent speichern

Weiterbildung – Siehe Roadmaps



<https://github.com/kamranahmedse/developer-roadmap>

... Aufnahme ab hier...

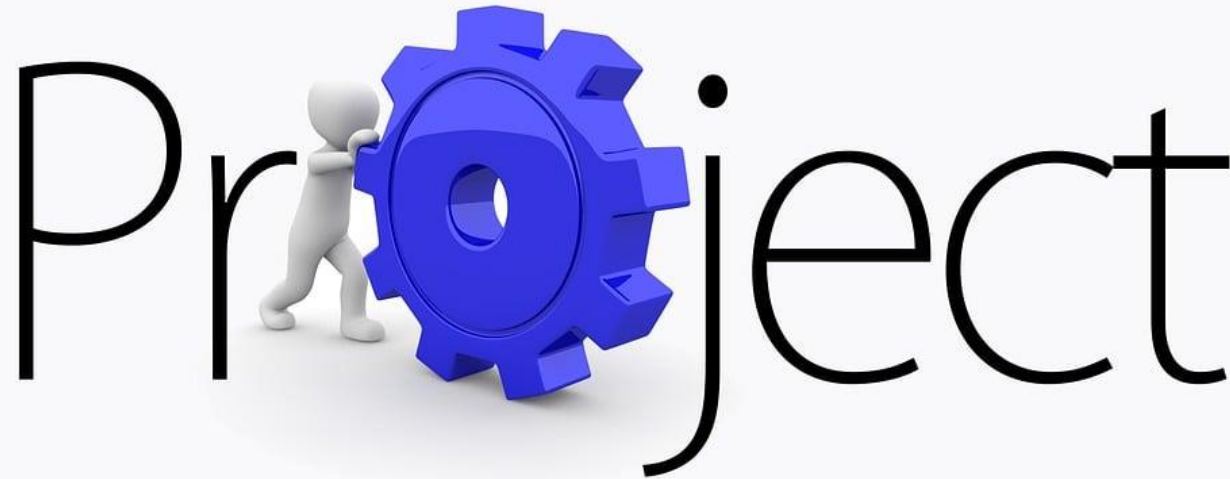
Projekt

- 5x Übungen (Einzelabgaben: 60%)

Einzelabgabe

- 1x Projekt (~~Gruppenabgabe~~: 40%), Frist: 06.09.2024

Entwickelt eine interaktive WebGIS-Anwendung zum Hinzufügen von Standorten aus Sachsen und Anzeigen von Wetterinformationen an diesen Standorten.





*Kerngedanke: was wir während
Aufgaben 1-5 gelernt haben in
ein neues Szenario einsetzen*



Neuer Standort hinzufügen

Es soll möglich sein neue Standorte zu hinzufügen entweder:

- als Einzeleinträge in einem Formular, oder
- als JSON-Datei, die gelesen wird aus einem HTML-Input-Feld, oder
- als Input von der GeoJSON Datei über ein HTML-Upload-Element

Fehlerfreie Einträge sollen in die Datenbank gespeichert werden, sonst bekommt der Nutzer bzw. Nutzerin eine Fehlermeldung (kein Alert)

Das Formular besteht aus vier Text-Input-Felder: Name (Pflichtfeld), Längengrad (Pflichtfeld), Breitengrad (Pflichtfeld), und URL (Optional)

Fehlerüberprüfung

In folgenden Fällen wird eine Fehlermeldung angezeigt

- Einzeleinträge in einem Formular: Leere Angaben
- JSON-Datei, die gelesen wird aus einem HTML-Input-Feld: Leere Angaben, invalides JSON
- Input von der JSON Datei über ein HTML-Upload Button: Leere Angaben, andere Dateiendungen als JSON/GeoJSON, invalides JSON
- Der neue Standort liegt nicht in Sachsen (die Grenzen von Sachsen dürfen hartkodiert sein)
- Beispieldaten zur Fehlerüberprüfung liegen auf [GitHub](#)

Wetterinformation an einem Standort anzeigen


Der Standort soll als Marker auf einer Karte angezeigt werden

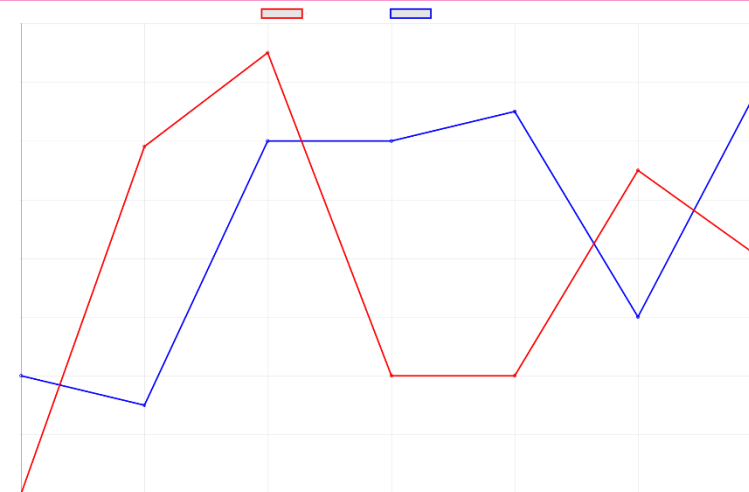
- Standardmäßig soll das Marker-Popup *nicht* aufgeklappt sein
- Beim Klicken auf dem Marker soll Folgendes in einem Popup-Fenster angezeigt werden
 - Name des Standorts
 - Entfernung zu Dresden (Koordinaten von Dresden hartkodiert) in Kilometers
 - Aktuelle Temperatur laut der Weather API (Einheit: Grad)

Die Karte soll die Wahl zwischen drei verschiedenen *base* Layers geben

Wetterprognose als Liniendiagramm

Beim Klicken auf einem Marker soll die Prognose für die nächsten 30 Tagen als Liniendiagramm angezeigt werden.

- Die Informationen sollen aus der [Weather API](#) geholt werden
- Drei Linien sollen in einem Diagramm erstellt werden: eine Linie für max, eine für min, eine Linie für day 
- Der Inhalt der Charts aktualisiert sich automatisch, je nach Standortauswahl (d.h. Beim Klicken auf einem Marker)



Folie 22

Standorte filtern

Es soll hier möglich sein die Menge der angezeigten Standorte anhand einer BoundingBox zu reduzieren

- Nur Rechtecken sollen als Form beim Zeichnen Nutzern erlaubt werden
- Nur Städte innerhalb der Bounding Box bzw. dem Rechteck sollen angezeigt werden
- Es soll möglich sein die Bounding Box zu editieren (die passenden Ergebnisse sollen automatisch aktualisiert werden)
- Es soll auch möglich sein die Bounding Box zu löschen (alle verfügbare Standorte sollen dann wieder angezeigt werden)

Visuelle Gestaltung

- Die Webseite soll *responsive* gestaltet werden.
- Fügt eine Bootstrap navbar hinzu.
- Input-Felder sollen anhand von Bootstrap formatiert werden.
- Das <div>-Element für die Karte soll mit einer sinnvollen Bootstrap-Klasse versehen werden.

Strukturierung

- Die Webseite besteht mindestens aus drei Seiten
 - Eine Seite zum Hinzufügen von neuen Standorten
 - Eine Seite zum Anzeigen der Infos. Die Infos zu einem Standort werden ausschließlich aus der Datenbank gelesen
 - Ein Impressum
- Die navbar soll das Wechseln zwischen verschiedenen Webseiten ermöglichen
- HTML-Elemente sollen von JavaScript aus über das Document Object Model (DOM) manipuliert werden. Der JavaScript Code soll als externe Datei in das HTML-Code eingebunden werden. Der CSS-Code soll auch als externe Datei in das HTML-Code eingebunden werden.

Dokumentation

Achtet darauf, dass...

- der Code ausreichend kommentiert ist
- die Ergebnisse der Zwischenschritten sichtbar gemacht werden (z.B. `console.log`)

Abgabeformat

- Abgabe als GitHub-Link in OPAL
- node-modules nicht mit abgeben
 - nur package.json notwendig
- *npm install & npm start* und es soll laufen



**06.09.2024,
23:55**



Online

**12.09.2024,
10:00-11:00**