



Hochschule  
München  
University of  
Applied Sciences

Hochschule München – Hochschule für angewandte Wissenschaften  
Fakultät für Geoinformation  
Karlstraße 6  
80333 München

# GeoApp-Entwicklung

Prototypische Entwicklung und Dokumentation einer GeoApp zum  
Thema "Abfrage und Geovisualisierung von Zeitreihen zu CO2-  
Ausstoß und Einsatz von fossilen und erneuerbaren Energien auf  
Länderebene"

Sommersemester 2023

Nicolaj Heidemann

Betreut durch Prof. Dr. Markus Oster und Prof. Dr. Ludwig Hoegner

# Inhaltsverzeichnis

Einführung .....	3
Daten und Server.....	3
Daten .....	3
Datenbank .....	3
Geoserver .....	4
WebApp.....	5
CO2 Seite .....	6
Energiemix und Elektrizitätsmix Seiten .....	8
AndroidApp .....	11
CO2 Layer.....	12
Energie- und Elektrizitätslayer.....	14
Fazit .....	16
Quellen .....	17

# Abbildungsverzeichnis

Abbildung 1 Navigationsleiste, Kartenkopf und Karte .....	5
Abbildung 2 Karte mit eingefärbten Layer .....	6
Abbildung 3 Jahresänderung.....	7
Abbildung 4 Liniendiagramm .....	7
Abbildung 5 CO2 Ländervergleich .....	8
Abbildung 6 Popup mit Trend .....	8
Abbildung 7 Leaflet Kreisdiagramme .....	9
Abbildung 8 Leaflet Kreisdiagramme vereinfacht.....	9
Abbildung 9 Balkendiagramm .....	10
Abbildung 10 Ländervergleich Energiemix.....	10
Abbildung 11 Android Menüauswahl.....	11
Abbildung 12 Android CO2 gesamt Layer.....	13
Abbildung 13 Android CO2 gesamt Popup.....	13
Abbildung 14 Android Jahresänderung.....	13
Abbildung 15 Android CO2 pro Kopf mit Popup .....	14
Abbildung 16 Android Energiemix.....	15
Abbildung 17 Android Energiemix Popup .....	15
Abbildung 18 Android Elektrizitätsmix.....	16
Abbildung 19 Android Elektrizitätsmix Popup .....	16

# Einführung

Das Projektziel war die Entwicklung eines Prototyps einer GeoApp zum Thema "Abfrage und Geovisualisierung von Zeitreihen zu CO<sub>2</sub>-Ausstoß und Einsatz von fossilen und erneuerbaren Energien auf Länderebene". Die GeoApp besteht aus drei Hauptkomponenten: dem Datenbankserver, der Webanwendung und der Android-App.

## Daten und Server

Um die benötigten Daten zu persistieren und über einen Service abzurufen, wurde der Zugang zu einem PostgreSQL Datenbankserver und einem Geo-Server bereitgestellt.

### Daten

Die verwendeten Daten zum Thema CO<sub>2</sub> Ausstoß und Einsatz von fossilen und erneuerbaren Energien auf Länderebene stammen von der Website [Our World in Data](#) und wurden dort im CSV-Format heruntergeladen. Die Daten umfassen Zeitreihen des CO<sub>2</sub> Ausstoßes, sowohl insgesamt als auch pro Kopf, und der Energieerzeugung aus verschiedenen fossilen und erneuerbaren Quellen.

Die Daten zum CO<sub>2</sub>-Ausstoß sind für jedes Land und jedes Jahr in Tonnen angegeben. Für die Energieerzeugung wurden separate Datensätze für den Energiemix und den Elektrizitätsmix erstellt. Sie zeigen die prozentuale Verteilung der einzelnen Energieträger in den jeweiligen Ländern pro Jahr. Der Energiemix beinhaltet Kohle, Gas, Öl und Nuklearenergie als fossile Quellen sowie Solarenergie, Wind- und Wasserkraft als erneuerbare Energieträger. Der Elektrizitätsmix setzt sich, bis auf Öl, aus denselben Quellen zusammen.

Zur Verknüpfung der erfassten Daten mit geographischen Informationen wurde eine Shapefile von der Webseite [Natural Earth Data](#) heruntergeladen. Sie wurde mithilfe von QGIS in die Datenbank importiert.

Um Diagramme an den korrekten geografischen Positionen darstellen zu können, wurde zusätzlich eine CSV-Datei mit den geographischen Mittelpunkten ([Centroids](#)) der einzelnen Länder heruntergeladen und in die Datenbank eingefügt.

### Datenbank

Die heruntergeladenen Daten wurden nach dem Import in verschiedene Tabellen innerhalb der PostgreSQL-Datenbank aufgeteilt. Separate Tabellen wurden sowohl für den gesamten CO<sub>2</sub>-Ausstoß als auch für den CO<sub>2</sub>-Ausstoß pro Kopf erstellt. Auch der Energiemix wurde strukturiert und auf einzelne Tabellen verteilt, wobei jede Tabelle einen spezifischen Energieträger repräsentiert.

Nach der Erstellung der verschiedenen Tabellen wurden Views angelegt. Diese nutzen die Funktionen `to_json` und `array_agg`, um die Datensätze pro Land in einem Array zu bündeln.

Auf diese Weise enthält jeder Ländereintrag ein Array mit den entsprechenden Jahresdaten sowie den CO<sub>2</sub>-Emissionen bzw. den prozentualen Anteilen der jeweiligen Energieträger.

Sobald alle Daten in den aggregierten Views verfügbar waren, erfolgte die Verknüpfung mit den geographischen Daten. Dies wurde durch den in den CSV-Dateien und der Shapefile enthaltenen ISO-Code jedes Landes ermöglicht. Um ein vollständiges Bild der Verteilung der Energieträger zu erhalten, wurden die Daten der einzelnen Energieträger sowohl für den Energiemix als auch für den Elektrizitätsmix in die entsprechenden Views zusammengefasst.

Es wurden Views für die folgenden Kategorien erstellt: CO<sub>2</sub>-Emissionen gesamt, CO<sub>2</sub>-Emissionen pro Kopf, detaillierter Energiemix, vereinfachter Energiemix (mit den Unterkategorien: fossile Energieträger, erneuerbare Energieträger, CO<sub>2</sub>-arme Energiequellen), detaillierter Elektrizitätsmix und vereinfachter Elektrizitätsmix.

## Geoserver

Die erstellten Views wurden anschließend im Geoserver als Layer publiziert. Dies eröffnete die Möglichkeit, die Daten über Web Map Service (WMS) und Web Feature Service (WFS) vom Geoserver abzufragen. Die Daten konnten daraufhin in verschiedenen Formaten, wie GeoJSON und KML, angefordert werden.

# WebApp

Anhand der vorhandenen Datensätze wurde die Entscheidung getroffen, die WebApp in drei Teilbereiche - CO<sub>2</sub>, Energiemix und Elektrizitätsmix - zu unterteilen. Diese Struktur ermöglicht es, die umfangreichen Datensätze auf eine klare und übersichtliche Weise zu präsentieren.

Beim Interaktionsdesign wurde besonderer Wert daraufgelegt, eine intuitive Benutzeroberfläche zu erstellen. Dafür wurden alle interaktiven Elemente, darunter Slider und Schaltflächen, im Bootstrap Kartenkopfbereich positioniert. Diese Anordnung erleichtert dem Benutzer die Navigation und Individualisierung der Karte sowie des Diagramms.

Im Zuge des Screendesigns wurde ursprünglich in Erwägung gezogen, das Diagramm unter der Karte zu platzieren. Dies hätte jedoch ein kontinuierliches Scrollen erfordert. Daher wurde beschlossen, Karte und Diagramm nebeneinander in einer Bootstrap Karte darzustellen.

Die grafische Benutzeroberfläche (GUI) wurde mit Bootstrap realisiert, wodurch eine moderne und benutzerfreundliche Oberfläche ermöglicht wurde. Eine Navigationsleiste für einen schnellen Themenwechsel und leicht zugängliche Bedienelemente waren Priorität beim Erstellen der GUI.

Für die kartographische Gestaltung wurden drei verschiedene Kartenformate ausgewählt: OpenStreetMap (OSM), Google Satellite und Google Streets. Diese Auswahl bietet den Nutzern unterschiedliche Perspektiven und unterstützt das Verständnis der dargestellten Daten.

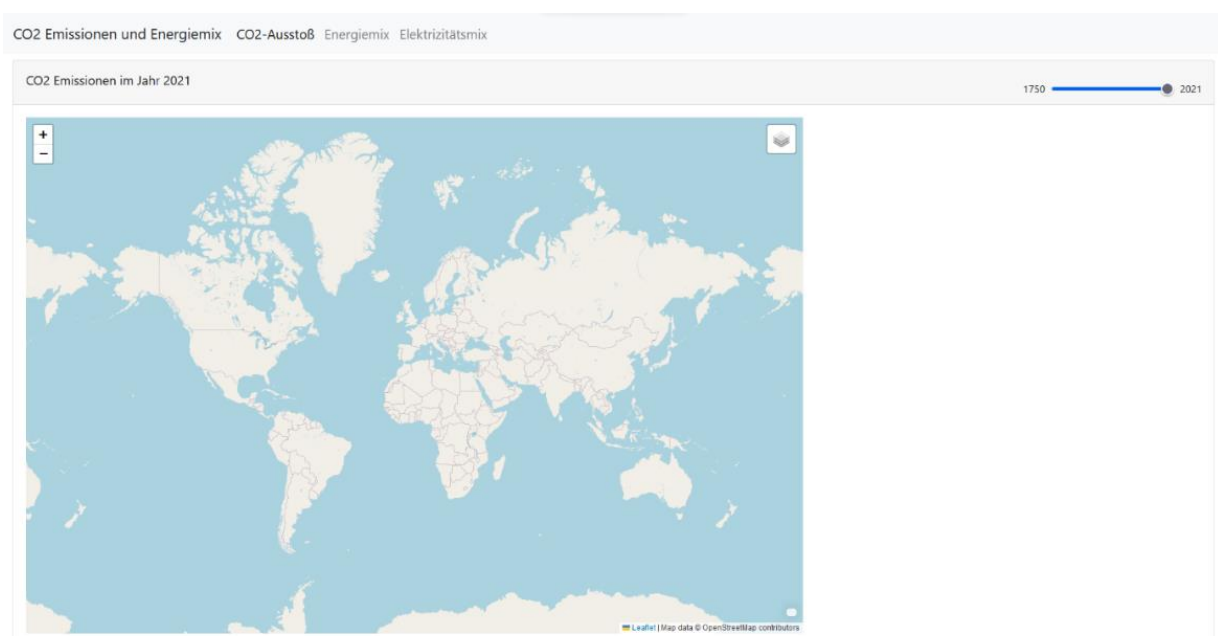


Abbildung 1 Navigationsleiste, Kartenkopf und Karte

## CO2 Seite

Für die CO2-Seite wurden zunächst HTTP-Anfragen mit Ajax an den Geoserver gesendet, um die aufbereiteten Daten im GeoJSON-Format abzurufen. Die Gesamtdaten für den CO2 Ausstoß und die pro Kopf Daten wurden in separaten Anfragen abgerufen.

Nach Erhalt der Daten werden in den *handleCO2Json* und *handleJsonPerCapita* Funktionen die Layer zur Karte hinzugefügt. Mit Hilfe der *setStyle* Funktion, kann dem Layer eine farbliche Einteilung zugewiesen werden. Dafür wurde die *chroma.js* Bibliothek genutzt, um eine Farbskala zu erzeugen. Die Klassenzahl wurde auf 10 festgelegt, um eine angemessene Unterscheidung zwischen den verschiedenen Emissionsebenen zu ermöglichen, während gleichzeitig eine klare und verständliche Visualisierung beibehalten wurde. Die Funktionen *totalCO2Style* überprüft, ob ein Land alle nötigen Daten besitzt, um die Einfärbung vorzunehmen, ansonsten wird das Land grau gefärbt. Falls Daten zu CO2 Emissionen vorhanden sind, wird anhand der mit *chroma.js* erstellten Farbskala die Farbe ermittelt, in der das Land gefärbt werden soll.

Für die CO2-Emissionen wurde ein Maximalwert von 900 Millionen Tonnen CO2 und für die pro-Kopf-Emissionen ein Maximalwert von 18 Tonnen festgelegt. Das Startjahr wurde auf 2021 gesetzt, da dies das neueste Jahr in den Daten ist.

Damit die Klassifizierung der Länder für den Benutzer verständlich ist, wird in der Karte eine Legende angezeigt. Diese wird bei einem Layerwechsel über eine *updateLegend* Funktion aktualisiert.



Abbildung 2 Karte mit eingefärbtem Layer

Ein eingebauter Slider ermöglicht es den Benutzern, durch die Jahre zu wechseln. Wenn der Slider bewegt wird, wird der Style der Karte entsprechend dem ausgewählten Jahr aktualisiert. Änderungen am Slider werden über eine *onChange* Funktion erkannt. Diese Funktion triggert dann die Einfärbung der Länder anhand des gewählten Jahres neu.

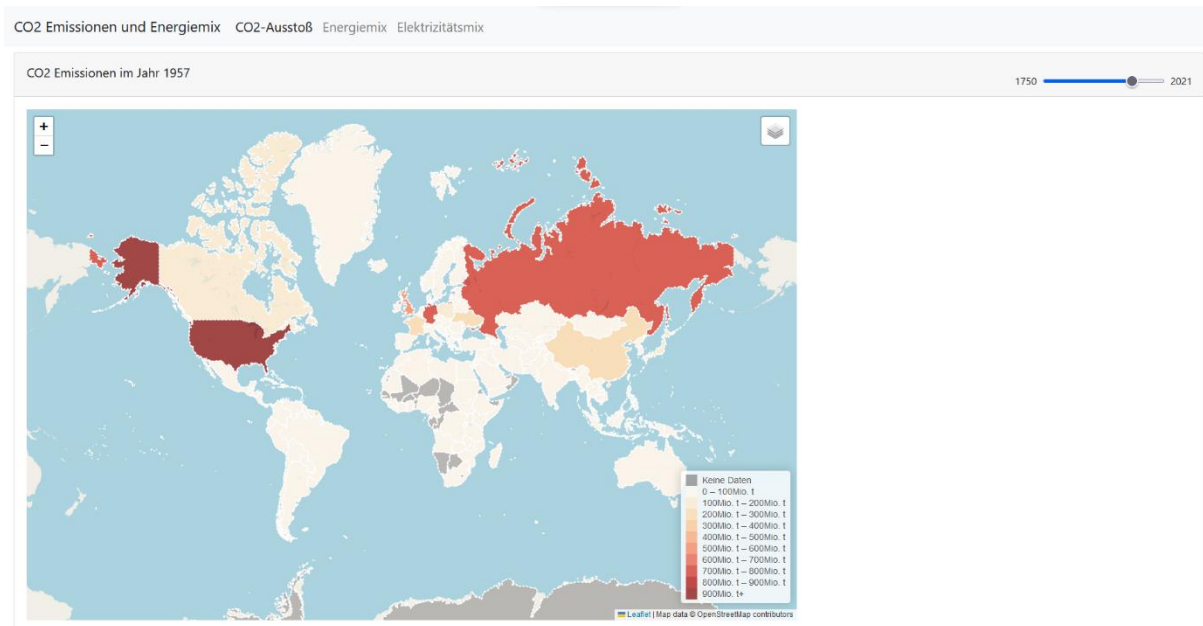


Abbildung 3 Jahresänderung

Beim Klicken auf ein Land wird über den Aufruf der *drawChart* Funktion mit Hilfe von google charts ein Liniendiagramm erstellt, das den zeitlichen Verlauf der CO2-Emissionen des betreffenden Landes darstellt.

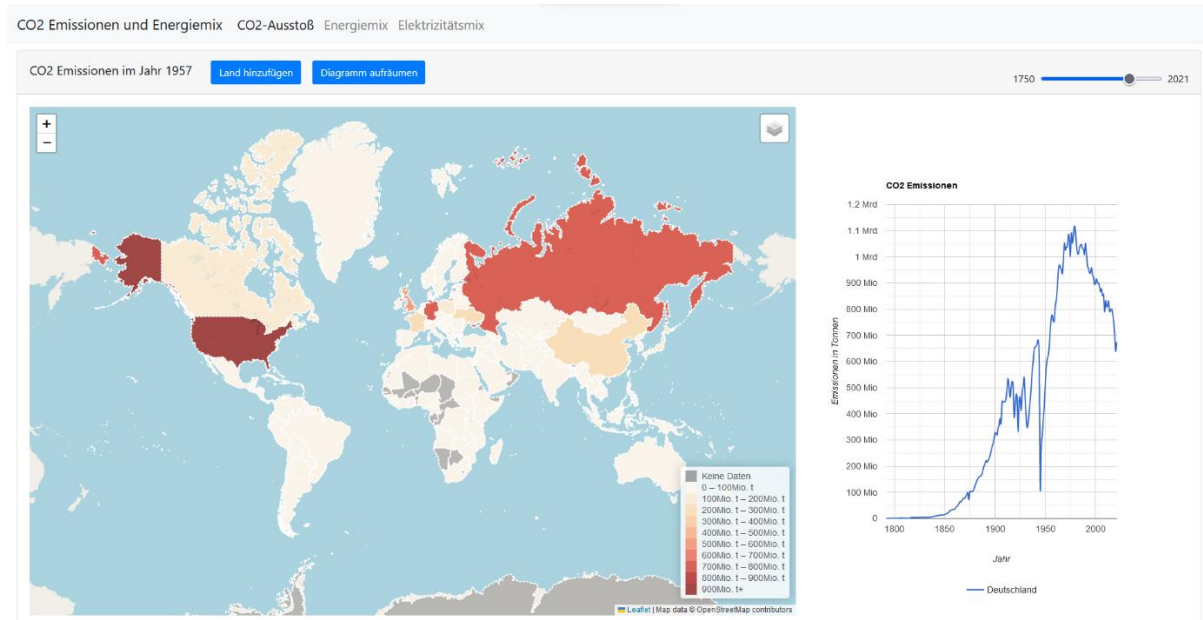


Abbildung 4 Liniendiagramm

Nachdem ein Land ausgewählt wurde, erscheinen zwei Buttons: "Land hinzufügen" und "Diagramm aufräumen". Der Button "Land hinzufügen" ermöglicht es, weitere Länder zum Diagramm hinzuzufügen, wobei sich das Styling des Buttons ändert. Der Button "Diagramm aufräumen" macht das Diagramm unsichtbar. Wählt man ein neues Land aus, ohne den Button

"Land hinzufügen" zu betätigen, wird das zuvor ausgewählte Land im Diagramm durch das neue ersetzt.

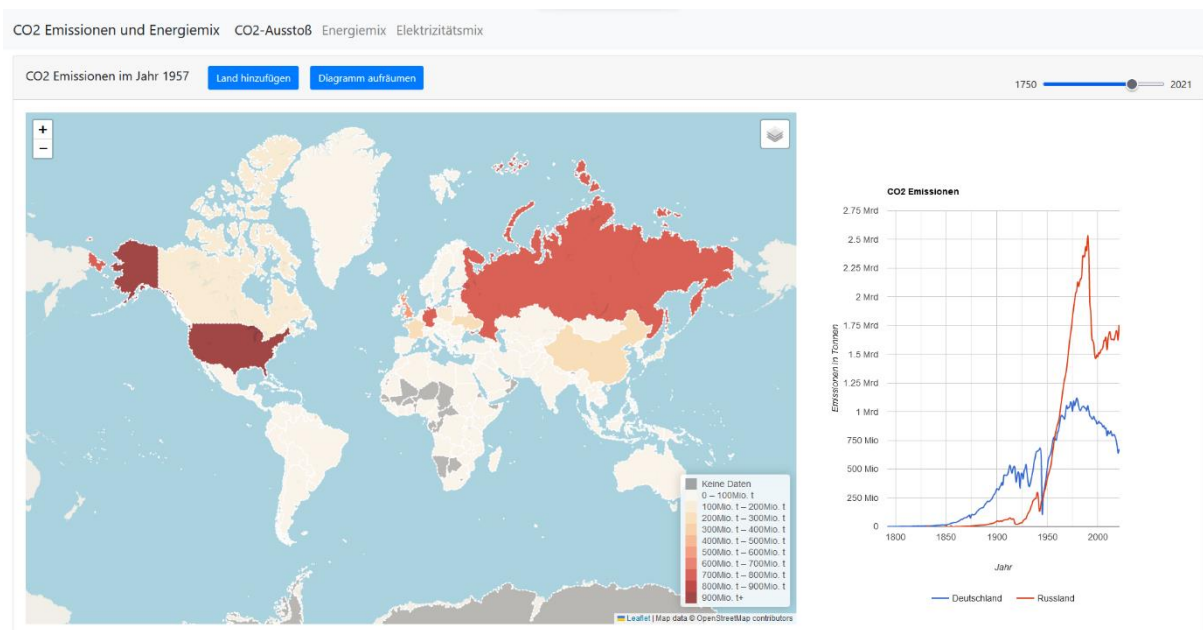


Abbildung 5 CO2 Ländervergleich

Wenn ein Land angeklickt wird, erscheint außerdem ein Popup, welches den Namen des Landes anzeigt, den CO2 Ausstoß gesamt oder pro Kopf, je nachdem welcher Layer gerade ausgewählt ist und den Trend der letzten 10 Jahre. Hierfür wurde eine Funktion geschrieben, die überprüft, ob sich die Emissionen in den letzten 10 Jahren verringert haben oder gestiegen sind. Im Popup wird dann entweder ein grüner Pfeil nach unten oder ein roter Pfeil nach oben angezeigt, um den Trend darzustellen. Für die Icons wurde [Font Awesome](#) in das Projekt mit eingebunden.



Abbildung 6 Popup mit Trend

## Energiemix und Elektrizitätsmix Seiten

Die Energiemix und Elektrizitätsmix Seiten der WebApp bieten einen detaillierten Einblick in die Nutzung verschiedener Energiequellen auf Länderebene. Für diese Darstellung werden jeweils zwei Datenebenen vom Geoserver geladen: der Gesamt-Energiemix bzw. Elektrizitätsmix und eine vereinfachte Version. Beide werden über HTTP-Anfragen mit Ajax



abgerufen und in den entsprechenden handle-Funktionen (handleEnergyJson und handleSimpleEnergyJson) als Layer zur Karte hinzugefügt.

Ein zentrales Feature dieser Seiten sind die Minicharts, die direkt auf der Karte angezeigt werden. Die kreisförmigen Diagramme werden mithilfe der *onEachFeature* Funktion von Leaflet erstellt und geben einen visuellen Überblick über den Energiemix bzw. Elektrizitätsmix eines jeden Landes. Hier kommen die zuvor importierten Centroids Daten ins Spiel, um die Minicharts korrekt auf der Karte zu positionieren. Abhängig vom ausgewählten Layer - entweder "Energiemix" oder "Energiemix vereinfacht" - werden in den Minicharts entweder alle Energiequellen dargestellt oder eine vereinfachte Darstellung, die nur zwischen fossilen und erneuerbaren Energiequellen unterscheidet. Die Farbgebung der Diagramme ist dabei auf die jeweiligen Energiequellen abgestimmt (z.B. Kohle in Braun, Öl in Schwarz usw.).

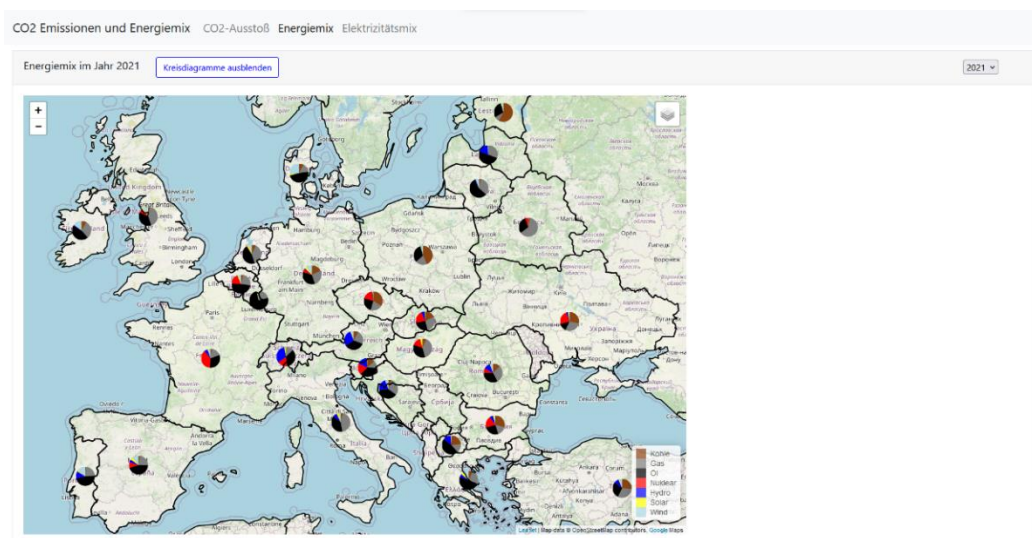


Abbildung 7 Leaflet Kreisdiagramme

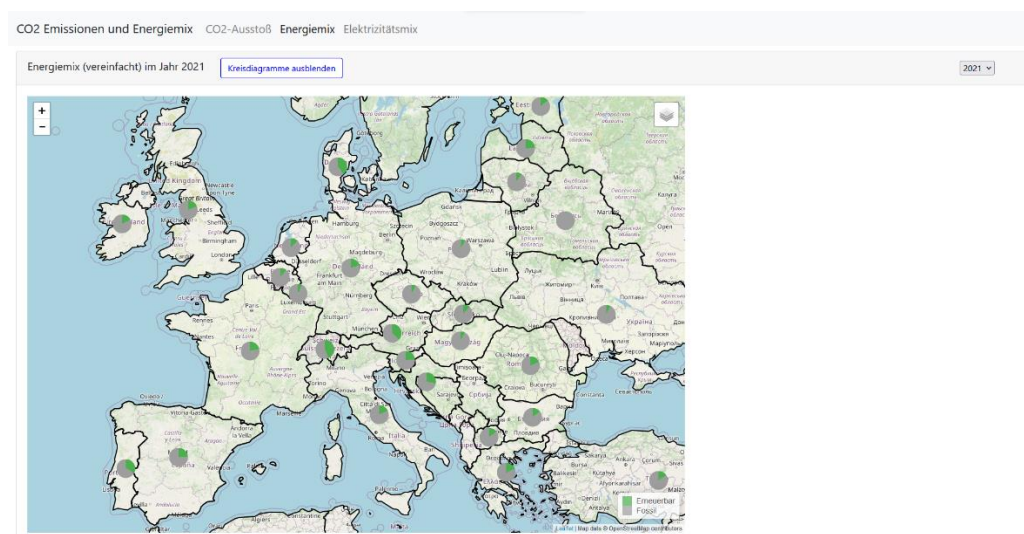


Abbildung 8 Leaflet Kreisdiagramme vereinfacht

Ein integrierter Button ermöglicht es den Benutzern, die Minicharts je nach Bedarf ein- oder auszublenden. Bei einem Klick auf ein Land erscheint ein Pop-up, das detaillierte Informationen zum jeweiligen Energiemix bzw. Elektrizitätsmix anzeigt, inklusive der prozentualen Anteile aller Energiequellen.

Parallel dazu wird ein Balkendiagramm mit Hilfe von Google Charts erstellt, das die prozentuale Verteilung der Energieträger visualisiert. In der detaillierten Ansicht sind dabei alle Energiequellen zu sehen, während in der vereinfachten Ansicht zwischen fossilen, erneuerbaren und "low carbon"-Energieträgern unterschieden wird. Über einen zusätzlichen Button können die Nutzer weitere Länder zum Vergleich zum Diagramm hinzufügen und das Diagramm bei Bedarf aufräumen.

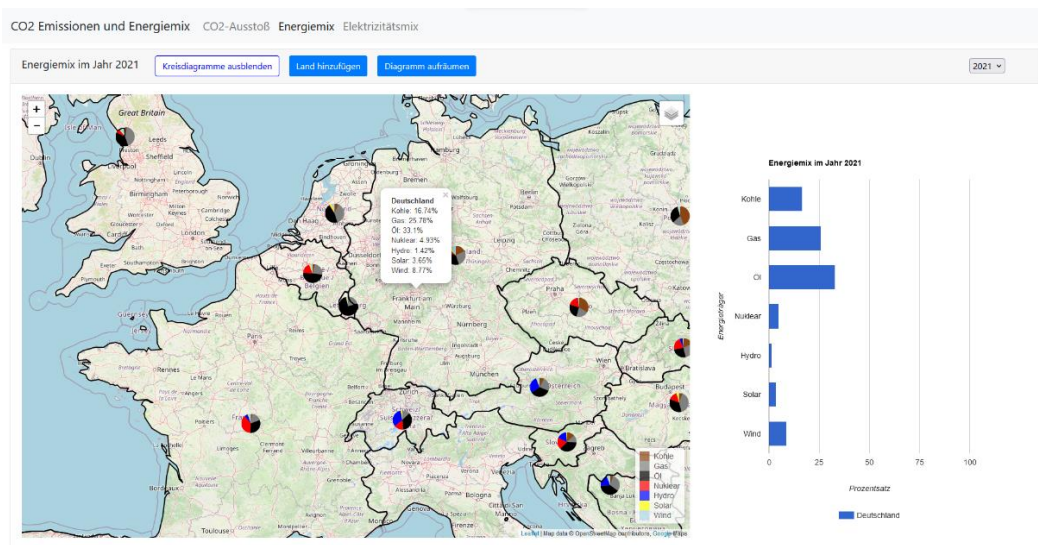


Abbildung 9 Balkendiagramm

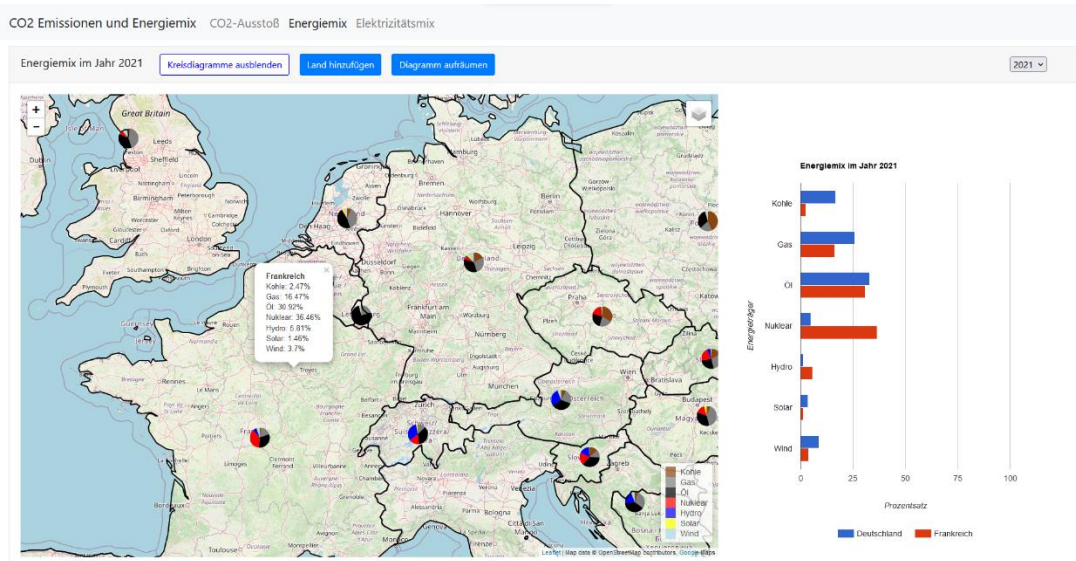


Abbildung 10 Ländervergleich Energiemix

# AndroidApp

Die Android App visualisiert die globalen CO2 und Energie- und Elektrizitätsmix Daten. Über ein Menü kann durch vier verschiedenen Datenlayer navigiert werden: CO2-Gesamt, CO2 pro Kopf, Energienutzung und Elektrizitätsmix. Ein Regler ermöglicht es durch die Jahre zu wischen und die Veränderungen im Laufe der Zeit zu beobachten.

Der erste Schritt in der Entwicklung war die Einbindung der OpenStreetMap (OSM). Nach dem Hinzufügen der osmdroid Abhängigkeit konnte in der *activity\_main.xml* ein *MapView* eingebaut werden. In der *MainActivity.java* wurde daraufhin eine Map Instanzvariable hinzugefügt, die mit der *MapView* verknüpft wurde. Daraufhin konnten einige Karteneinstellungen vorgenommen werden.

Um später zwischen den verschiedenen Layern wechseln zu können, wurde eine *main\_menu.xml* angelegt. In der *MainActivity.java* wurde daraufhin eine *onOptionsItemSelected* Funktion hinzugefügt, die den Layerwechsel verarbeitet. Das Enum *DataType* hilft, um typsicher zu bestimmen auf welchem Layer man sich befindet. In der *selectedLayer* Instanzvariable wird der aktuelle *DataType* gespeichert.

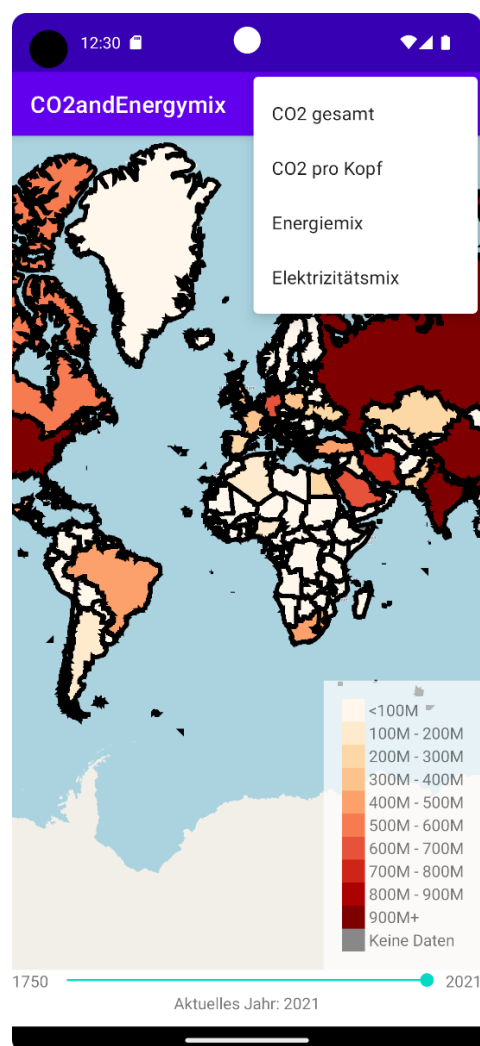


Abbildung 11 Android Menüauswahl

Nach der erfolgreichen Einbindung der OSM-Karte und des Menüs musste nun die Grundlage für die API-Abfragen implementiert werden. Hierfür wurde eine *callGeoserverApi* Funktion geschrieben, die eine URL übergeben bekommt. Der URL hängt vom aktuell ausgewählten Layer ab. Zunächst wird eine *URLConnection* eröffnet und die Abfrage Methode auf GET gesetzt. Falls die Abfrage erfolgreich war, wird der *InputStream* über einen *BufferedReader* in einen String umgewandelt.

## CO2 Layer

Da nun die Grundlage für die API-Abfragen vorhanden war, mussten weitere Funktionen implementiert werden, die die richtigen URLs aufrufen, die Antwort auslesen und Polygone auf die Karte zeichnen.

Für die Verarbeitung der Response wurde dem Projekt eine neue Klasse *AddLayer* hinzugefügt. Diese Klasse enthält die Funktion *addCO2TotalLayer*, welche den Response String in ein *JSONObject* umwandelt. Durch dieses *JSONObject* wird daraufhin durchiteriert und es werden Geometrie, Name, Liste des jährlichen CO2 Ausstoß und Liste der Jahre ausgelesen. Anhand des übergebenen Jahres wird der CO2 Ausstoß ermittelt. Mit der ausgelesenen Geometrie konnten dann Polygone erstellt werden und mit *map.getOverlayManager().addPolygon(polygon)* der Karte hinzugefügt werden. Davor wurde die Farbe des Polygons anhand des CO2 Ausstoßes ermittelt. Hierfür wurde ein Array mit den Farbabstufungen angelegt und über if else Statements bestimmt in welche Kategorie das Land eingestuft wird. Für den CO2-gesamt-Layer wurde eine Klassenzahl von 10 gewählt und der maximale CO2 Ausstoß auf 900 Mio. Tonnen festgelegt. Die Farbe konnte dem Polygon über *polygon.getFillPaint().setColor(color)* zugewiesen werden. Außerdem besitzt jedes Polygon einen *onClickListener*. Dieser wird ausgelöst, wenn auf ein Polygon geklickt wird. Wenn das angeklickte Land Daten zu CO2 Emissionen besitzt, wird dieses in einem Infofenster dargestellt. Sollten keine Daten vorliegen wird darauf im Infofenster hingewiesen.

In der *AcitivityMain.java* wurde eine *getData* Funktion hinzugefügt, die die anhand des aktuell gewählten Layers den richtigen URL an die *callGeoserverApi* übergibt und daraufhin auch die passende *AddLayer* Funktion aufruft.

Um die Einfärbung der Länder für den Benutzer verständlich zu machen, ist dem *activity\_main.xml* eine Legende hinzugefügt worden. Diese Legende wird in der *updatelegenden* Funktion in der *MainActivity.java* erstellt, je nachdem welcher Layer aktuell gewählt ist. Die Funktion wird aufgerufen, wenn sich der Layer ändert.



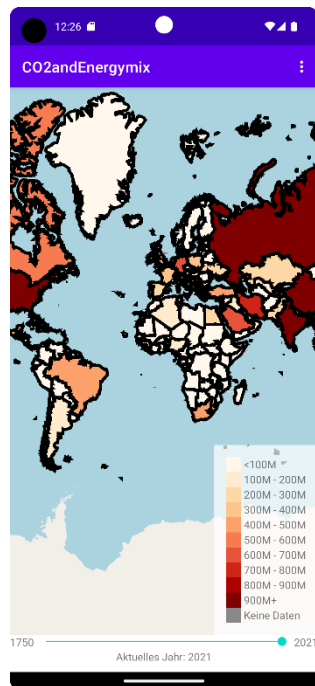


Abbildung 12 Android CO2 gesamt Layer

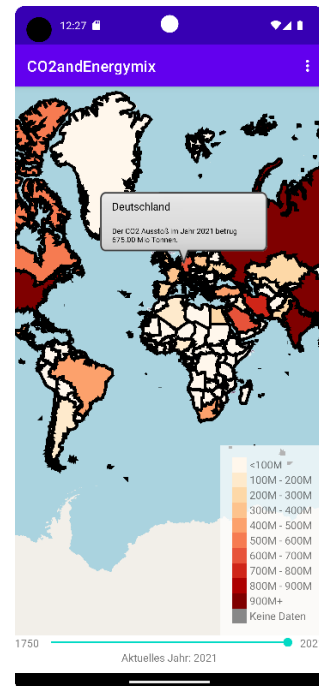


Abbildung 13 Android CO2 gesamt Popup

Um durch die Jahre wischen zu können besitzt die App einen Regler, welcher ein Start- und End-Jahr besitzt. Außerdem wird unter dem Regler das aktuell gewählte Jahr angezeigt. Der Regler ist ein SeekBar und wurde der activity\_main.xml als Layout, sowie der MainActivity.java als Instanzvariable hinzugefügt. Damit bei Änderungen am Regler das aktuell gewählte Jahr aktualisiert wird und auch die Einfärbung der Länder, sowie Erstellung der Popups neu vorgenommen wird, wurde ein `onSeekBarChangeListener` implementiert. Dieser aktualisiert beim Bewegen des Reglers das ausgewählte Jahr in der Anzeige und ruft beim Loslassen des Reglers die `getData` Funktion mit dem ausgewählten Jahr auf.

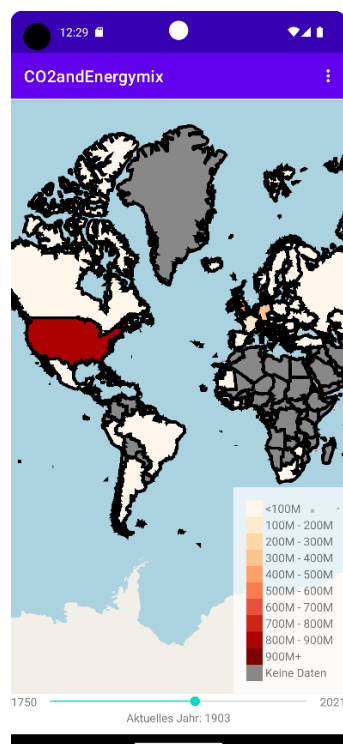


Abbildung 14 Android Jahresänderung

Für den CO2 pro Kopf Layer ist das Verhalten ähnlich. In der *addCO2PerCapLayer* Funktion sind die einzigen Unterschiede, dass die Liste der CO2 Ausstöße im JSONObject anders ausgelesen werden und die Klassifizierung sich etwas unterscheidet. Der maximale CO2 Ausstoß pro Kopf wurde auf 18 Tonnen festgelegt. Die Klassenanzahl bleibt bestehen. Außerdem sind die Popups leicht angepasst und zeigen nun den CO2 Ausstoß pro Kopf an.

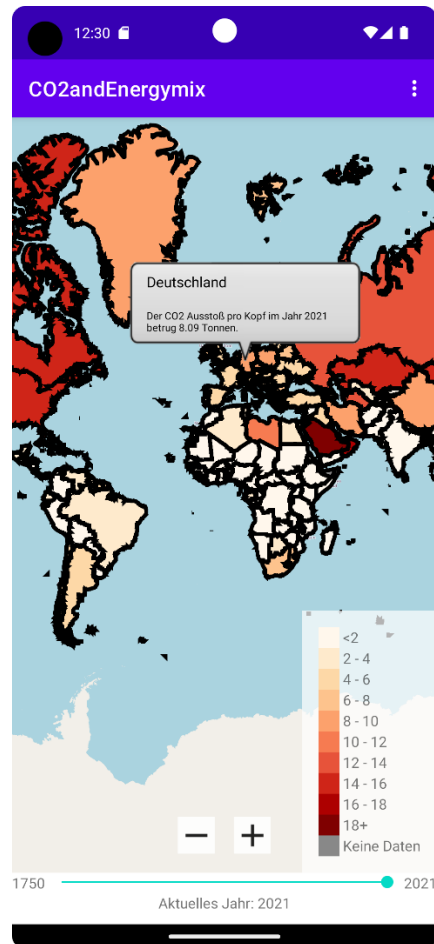


Abbildung 15 Android CO2 pro Kopf mit Popup

## Energie- und Elektrizitätslayer

Um die Energie- und Elektrizitätslayer zu laden, wird beim Layerwechsel die *getData* Funktion mit dem passenden URL aufgerufen. Daraufhin werden in der *addEnergyLayer* und *addElectricityLayer* Funktion die Daten ausgelesen und die Polygone gezeichnet.

In der *addEnergyLayer* Funktion wird ähnlich wie bei den CO2 Funktionen die Geometrie, der Name ausgelesen. Da das Energiemix GeoJSON jedoch mehrere Energieträger beinhaltet wird vor dem Auslesen dieser eine HashMap initialisiert. In dieser wird dann für jeden Energieträger der Prozentsatz aus dem Jahr eingespeichert. Dann werden wie in den vorherigen Funktionen die Polygone gezeichnet. Hier wird allerdings bei der Farbgebung nur unterschieden, ob das Land Daten zum Energiemix besitzt oder nicht. Länder die Daten besitzen werden blau eingefärbt, ansonsten wird grau eingefärbt. Die Legende wird über die *updatelegende*

Funktion auch entsprechend angepasst. Die Daten zum Energiemix werden im Popup angezeigt.



Abbildung 16 Android Energiemix

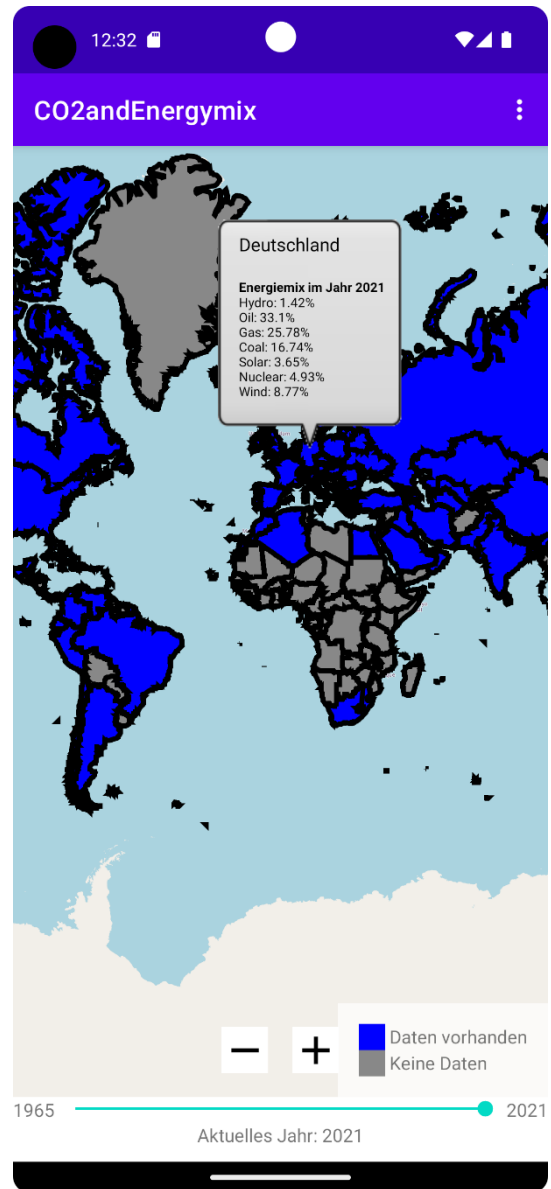


Abbildung 17 Android Energiemix Popup

Der Elektrizitätslayer ist sehr ähnlich aufgebaut. Die Daten im GeoJSON unterscheiden sich ein wenig. Im Elektrizitätsmix ist Öl als Energieträger nicht enthalten und wird deshalb beim Auslesen der Response auch nicht berücksichtigt.



Abbildung 18 Android Elektrizitätsmix

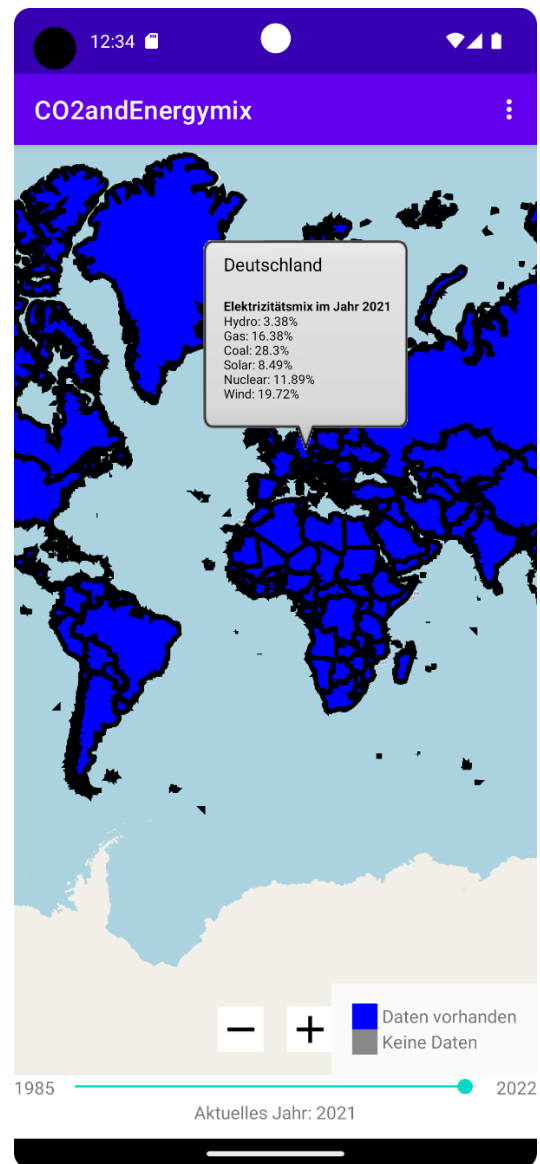


Abbildung 19 Android Elektrizitätsmix Popup

## Fazit

Im Rahmen dieser Projektarbeit wurde erfolgreich eine GeoApp, bestehend aus WebApp und AndroidApp, zur Abfrage und Geovisualisierung von Zeitreihen zu CO<sub>2</sub>-Ausstoß und Energieverbrauch auf Länderebene entwickelt.

Die Kernaufgabe der Arbeit war die Erstellung einer interaktiven Anwendung, die eine visuell ansprechende Darstellung der globalen CO<sub>2</sub>-Emissionen und Energiequellen bietet. Um dies zu erreichen wurde eine Datenbank und ein WMS und WFS über einen Geoserver aufgebaut.

Die WepApp wurde mithilfe von HTML, JavaScript und Leaflet geschrieben. Sie bietet eine umfangreiche Interaktion mit den Daten durch einen Slider bzw. eine Jahreswahl, die es ermöglicht, durch die Jahre zu wechseln. Darüber hinaus wurden drei wechselbare Themenseiten implementiert, die dem Benutzer einen detaillierten Einblick in die globalen CO<sub>2</sub>-Emissionen, Energie- und Elektrizitätsmix bieten.



Die AndroidApp basiert auf einer OSM-Karte und einem Datenlayer. Der Datenlayer kann über ein Menü gewählt werden. Die App ermöglicht ebenfalls das Navigieren durch die Jahre mithilfe eines Sliders.

Die Gestaltung der GeoApp folgte den Prinzipien der thematischen Kartographie und Geomedientechnik. Dabei wurde besonderer Wert auf ein klares und ansprechendes Screendesign und GUI gelegt, welches sowohl funktional als auch visuell ansprechend ist.

Abschließend kann festgehalten werden, dass die vorgenommene Funktionalität der WebApp erreicht wurde und darüber hinaus einige weitere Features implementiert wurden. Es wurde eine solide Grundlage für weitere Verbesserungen und Erweiterungen der App geschaffen, die es ermöglicht weitere Daten und Visualisierungen einzubinden.

## Quellen

Skript und Übungen aus dem Modul GeoApp-Entwicklung

Our World in Data: <https://ourworldindata.org/>

Natural Earth Data: <https://www.naturalearthdata.com/downloads/50m-cultural-vectors/>

Centroids CSV: <https://github.com/gavinr/world-countries-centroids>