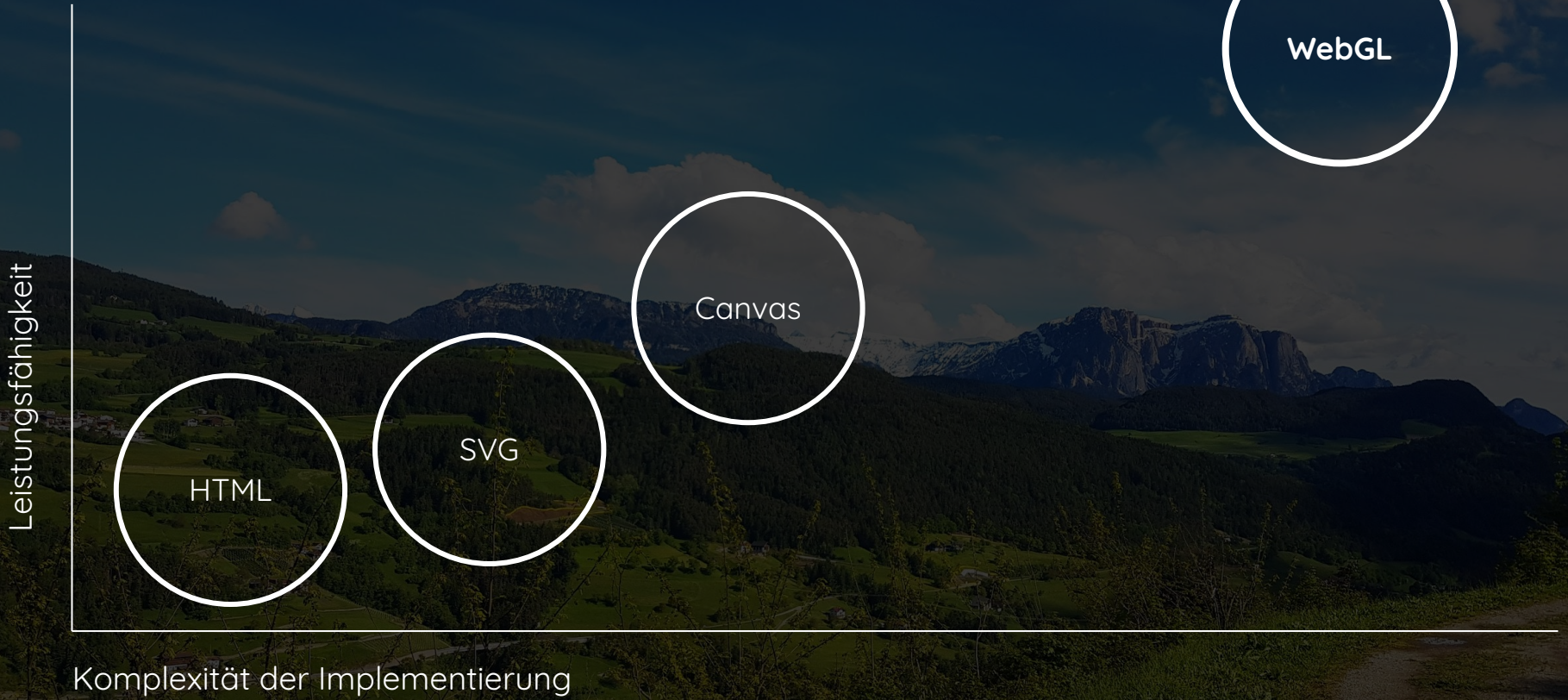
VERBAALPINA
Der alpine Kulturraum im Spiegel seiner Mehrsprachigkeit.
Animierte, geographische Massendatenvisualisierung am Beispiel exportierter „VerbaAlpina“ - Webtechnologie
Dr. David Englmeier
VERBAALPINA

Übersicht

- Was ist **WebGL** und wie funktionieren die Kartenvisualisierungen von VerbaAlpina?
- Unterschiedliche Ansprüche der Projekte an die Implementierung
- Technologie-Export an das Projekt “T-Migrants” (LMU München)
- **Demo und Diskussion**

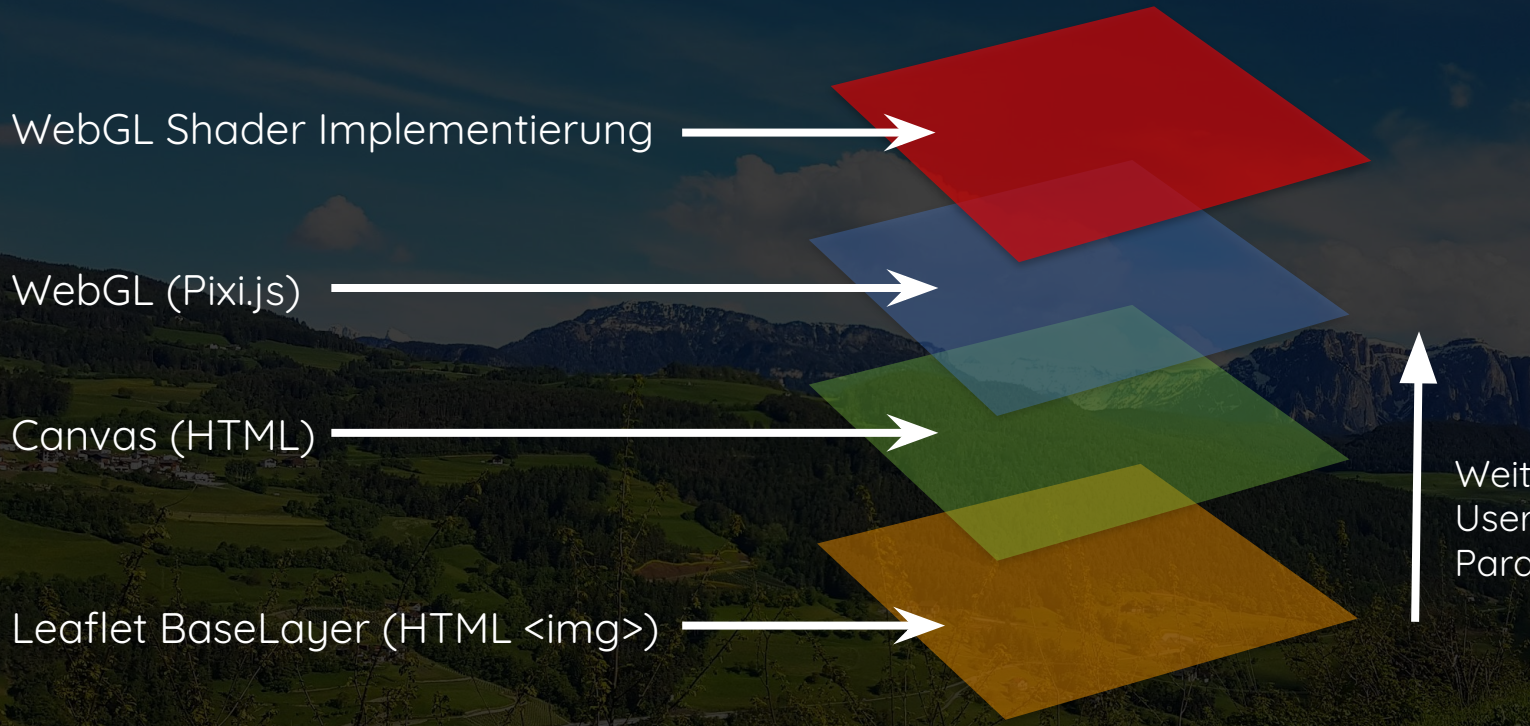
Rendering Technologien im Web



WebGL Grafikpipeline

Unidirektionaler Ablauf!

Funktionsweise



WebGL Setup zur Geovisualisierung

Komponenten / Technology-Stack

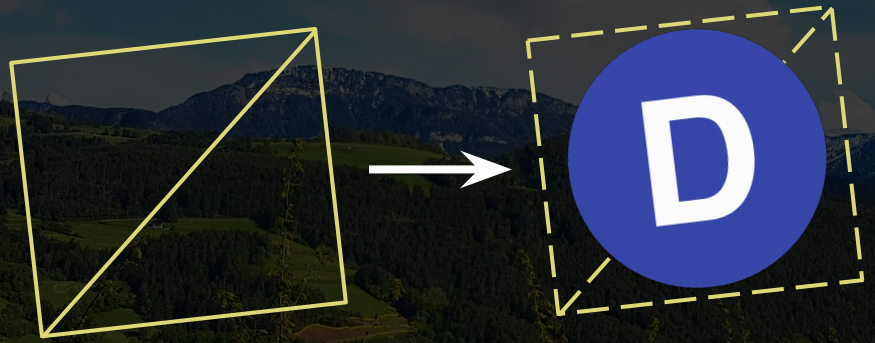
- Leaflet.js (https://github.com/Leaflet/Leaflet)
- Leaflet.PixiOverlay (https://github.com/manubb/Leaflet.PixiOverlay)
- Pixi.js (https://github.com/pixijs)
- Performante Eigenimplementierung (https://github.com/VerbaAlpina/Interactive-Map_Plugin)

WebGL Implementierung

- Triangulation von Polygonen und Linien-(Zügen)

WebGL Implementierung

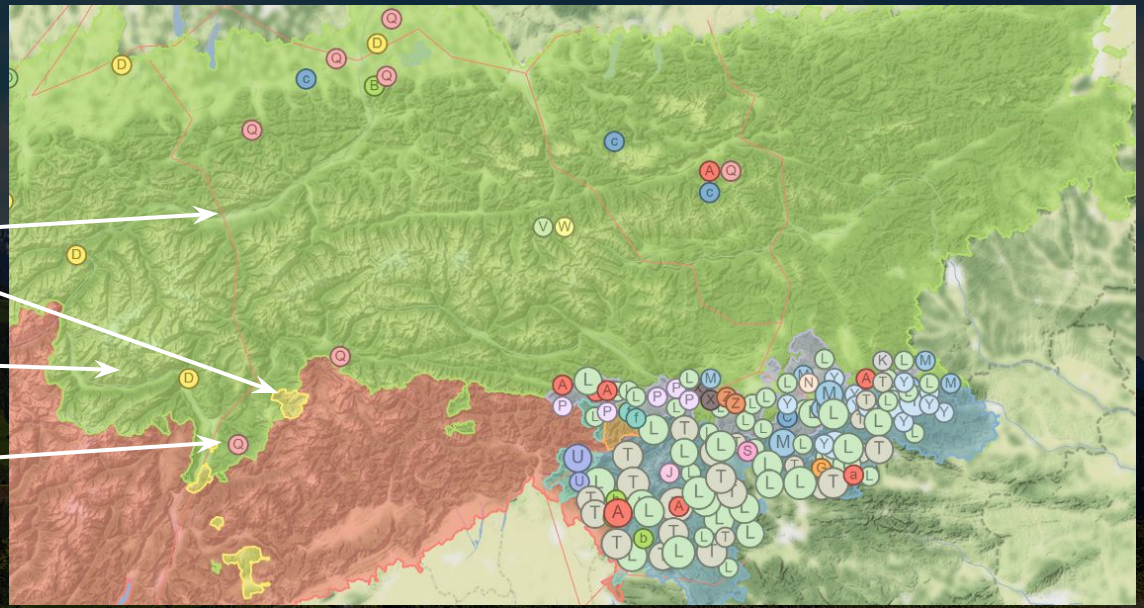
- Texturen und Inverse Skalierung



- Skalierung i.d.R auf alle Objekte angewendet => konstante Objekte invers skalieren (Shader)!
- Im Demo: Linien sowie Marker

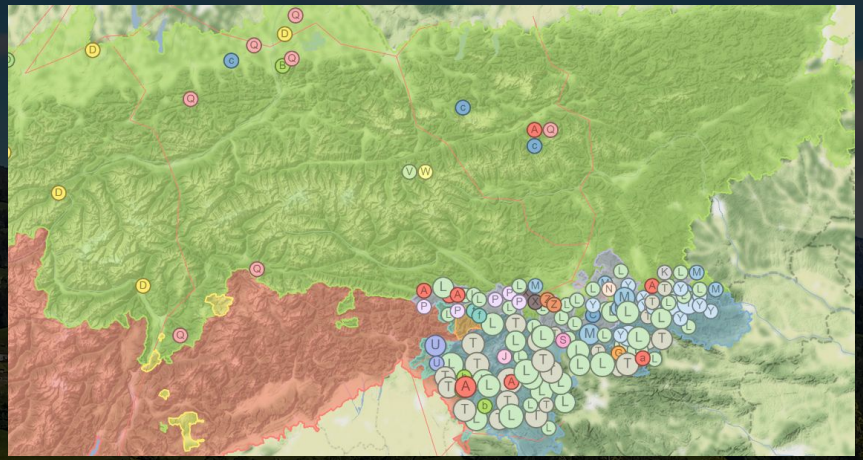
Klassische Kartenelemente

- Benötigte Grundelemente:
  - **Linien, Linienzüge**
  - **Polygone**
  - **Marker**



VerbaAlpina vs. T-Migrants

VerbaAlpina



- Interaktiv animiert
- Komplexe Marker
- Potentiell große Polygon-Anzahl
- Keine transparenten Linien

Anpassung der Implementierung

T-Migrants

- Permanent animiert
- Große Anzahl von interaktiven Markern
- Transparente Linien
- Keine Polygone

Demo:

Interaktive Karte (VerbaAlpina) vs. Interaktive Karte (T-Migrants)

Fazit & Diskussion

- Komplexe Implementierung => Immense Leistungsfähigkeit
- Grundlagen der Grafikprogrammierung sehr hilfreich + JavaScript Erfahrung
- **Uniforme Lösungen schwer umsetzbar für Karten-Tools wegen Anforderungs-Divergenzen**