

# Eine webbasierte Anwendung zur Datenvisualisierung für MATSim

William Charlton, Technische Universität Berlin

---

## Einleitung

Derzeit gibt es keine webbasierten, interaktiven Tools zur Verbreitung von MATSim-Ergebnissen. Daraus entsteht eine Herausforderung für die Verwendung von MATSim in öffentlichen Richtlinieneinstellungen: Nur Personen mit umfassenden technischen Kenntnissen und Zugriff auf spezielle Software können die Ergebnisse sinnvoll untersuchen.

Unsere Forschung versucht, diese Lücke zu schließen, indem eine offene webbasierte Visualisierungsanwendung geschaffen wird, die speziell für MATSim entwickelt ist.

## Anforderung

Das Forschungsteam der TU Berlin hat mehrere offene ergebnisoffene Diskussionsrunden abgehalten, bevor Code geschrieben wurde. Die folgenden Anforderungen sind aus diesen Diskussionen herausgekommen.

- **Anforderung 1: Moderne webbrower-basierte Bedienung.** Verschiedene neue Webtechnologien ermöglichen diese Untersuchung, z.b. HTML 5, CSS 3, WebGL, ECMAScript 6 und “Web Workers”. Leider aktualisieren Endbenutzer\*Innen ihre Browser nicht häufig. Dies schafft eine “long-tailed” Technologie Anpassungs. Wir haben uns für die neuesten Web-Technologien entschieden, mit der Erwartung, dass diese in Zukunft weiter verbreitet werden.
- **Anforderung 2: Open Source.** Das gesamte Projekt ist vollständig Open Source und unter der GNU General Public License v3 lizenziert.
- **Anforderung 3: Verwenden guter Standardeinstellungen mit minimaler Konfiguration.** Webbenutzer\*Innen sind gewohnt, sich mit einer Site sofort vertraut zu machen, oft innerhalb von Sekunden nach ihrer ersten Interaktion. Es ist äußerst wichtig, dass diese Forschung den aktuellen Best Practices für Benutzeroberfläche und Benutzererfahrung folgt.
- **Anforderung 4: Eine erweiterbare Anwendung.** Jeder Anwendungsfall der Datenvisualisierung ist anders. Es werden grundlegende Funktionen und Vorlagen bereitgestellt. Eine Benutzer\*Innen mit Programmierkenntnissen sollte jedoch neue Visualisierungen erstellen können, die von den Autoren nicht erwartet worden.

## Erste Versuche

In ersten Experimenten wurden verschiedene Ansätze bewertet, bevor ein Front-End-Technologie-Stack eingesetzt wurde. Das Back-End ist in einem Artikel von J. Laudan beschrieben.

**Visualisation von Daten auf einer geografischen Karte.** Zwei beliebte webbasierte Javascript Bibliotheken wurden zur Anzeige geografischer Daten getestet, “Leaflet” und “Mapbox GL”. Mapbox GL hat große Datensätze viel besser als Leaflet gehandelt.

**Andere Daten.** Experimente mit vielen verschiedenen Grafik- und Diagrammbibliotheken führten zum Modul “Vega-Lite” (Satyanarayan, 2016). Vega-lite folgt einer sogenannten deklarativen Syntax “Grammatik der Grafiken”, wie sie von Wilkinson (2005) verbreitet wurde.

## Ergebnisse: Der aktuelle Status der Anwendung.

Eine Arbeitsinstanz der Anwendung ist jetzt online und unter <https://viz.vsp.tu-berlin.de> verfügbar. Beispieldatensätze wurden hochgeladen, und vorgefertigte Visualisierungen sind öffentlich zugänglich, um den aktuellen Status der Anwendung zu demonstrieren.

Derzeit sind verschiedene Arten von Aggregatvisualisierungen in Betrieb:

- Herkunft- / Zielflüsse zwischen aggregierten Gebieten
- Kantenflüsse nach Tageszeit
- ÖPNV-Angebote
- Sankey-Diagramme, die Flüsse zwischen mehreren Auswahlmöglichkeiten darstellen, z. B. Moduswechsel zwischen zwei Szenarien
- Emissionsniveaus auf geografischer Netzbasis

Auch ist eine disaggregierte Visualisierung verfügbar:

- Eine Fahrzeugflusssimulation, bei der Fahrzeuge in Echtzeit im Netzwerk fahren.

**Visualisierungs Plug-Ins.** Neue Visualisierungen können schnell erstellt und dem Framework hinzugefügt werden, um neue Funktionen zu generieren. Dies die Komponentenarchitektur der “Vue Single Page Application” ermöglicht. Um eine neue Visualisierung zu erstellen, kopiert man eine vorhandene leere Visualisierungsvorlage, gibt die erforderlichen Eingaben und Parameter an und ändert den Javascript-Code entsprechend der Anforderungen. Dies erfordert derzeit extensive Programmierkenntnisse in Javascript. Leider ist es kein System, das “Point and Click” ermöglicht.

## Leistung

Sogar mit moderner Hardware und den neuesten Browsern ist es schwierig, disaggregierten MATSim-Daten gut zu erreichen. Beispielsweise hängt die Fahrzeugflusssimulation stark vom Back-End-Server ab, um Daten für den Browser zu erzeugen.

Aggregierte Daten lassen sich viel einfacher visualisieren. Es müssen weitere Anstrengungen unternommen werden, um die jetzt verfügbaren Back-End-Serverfunktionen zu nutzen.

## Fazit

Mit den verschiedenen Technologien zu experimentieren und alles zusammenzubringen war eine große Aufgabe, aber die Anwendung ist jetzt ziemlich stabil.

Innerhalb weniger Tage können man eine neue Visualisierung erstellen. Es ist ermutigend zu sehen, wie Visualisierungen in so kurzer Zeit von der Idee zum groben Entwurf werden.

Die Welt der Datenvisualisierung steht nicht still. Vor kurzem wurden wichtige Datenvisualisierungsbemühungen von gut finanzierten Unternehmen wie Uber veröffentlicht. Es gibt berechnete Fragen, wie diese Arbeit von großen, professionellen Coding-Teams abgelöst werden könnte.

Trotz dieser Bedenken ist die MATSim Visualisierungsanwendung einsatzbereit. Dies ist ein gutes Vorzeichen für die weitere Entwicklung.

Der gesamte Code ist unter <https://github.com/matsim-org/viz> verfügbar.