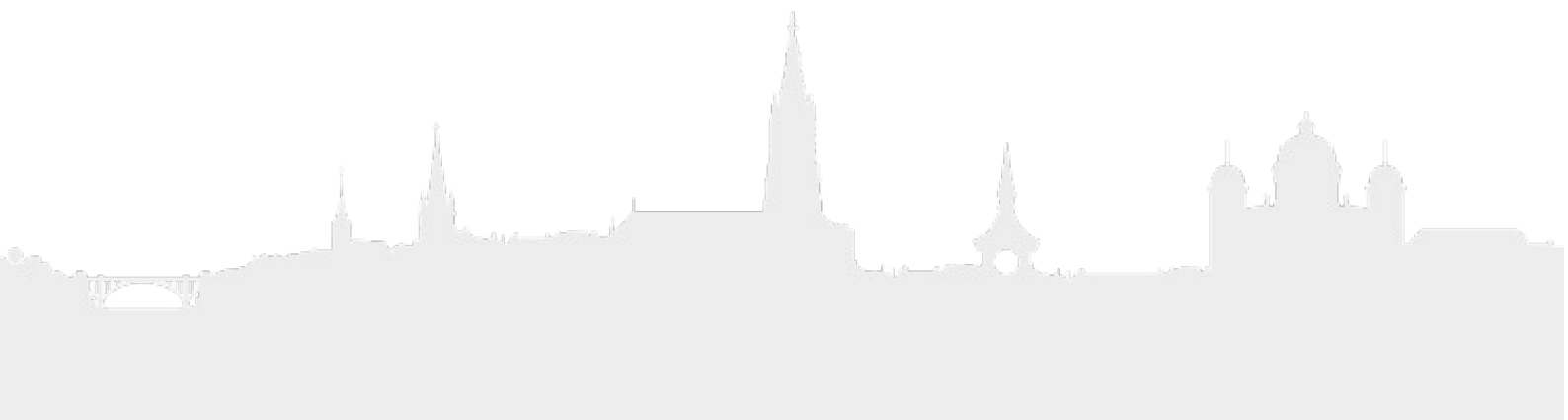




# Aareschwümmendi: Sunne oder Schatteplätzli?

---

Technische Informationen für die Jury



## Technische Informationen für die Jury

Aktueller Stand des Sourcecodes

- <https://github.com/cblaettl/sol-umbra>

### Ausgangslage

- Worauf habt ihr euch fokussiert?  
Die App soll möglichst selbsterklärend und leicht zu bedienen sein. Durch die einfache Bedienung wird die technische Komplexität verborgen und ermöglicht eine spielerische Interaktion mit der Applikation.
- Welche technischen Grundsatzentscheide habt ihr gefällt?  
Für die gegebene Problemstellung wurde eine reine Frontend-Lösung entwickelt. So wird die Rechenleistung für die Darstellung des Schattenwurfs von der Client-Maschine, respektive dessen Browser, bereitgestellt. So müssen keine teuren, leistungsstarken Rechner zur Verfügung gestellt werden.

### Technischer Aufbau

- Welche Komponenten und Frameworks habt ihr verwendet?
- Folgende Bibliotheken wurden verwendet:
  - o vue.js: Framework für das rendern der Webapplikation
  - o three.js: Engine für das 3D-Rendering im Browser
  - o ifc.js: Laden des Stadtmodells (ifc-File) für die Darstellung im 3D-Modell mittels Web-Assembly
  - o vanilla-js: Glue-Code und Anbindung diverser API's
  - o HTML/CSS: Darstellung/Visualisierung
- Wozu und wie werden diese eingesetzt?  
Siehe oben.

### Implementation

- Gibt es etwas Spezielles, was ihr zur Implementation erwähnen wollt?  
Das Potenzial von Web-Assembly ist enorm. Die Fähigkeit, rechenintensive Aufgaben wie diese im Browser auszuführen, eröffnet eine Vielzahl neuer Anwendungsmöglichkeiten.
- Was ist aus technischer Sicht besonders cool an eurer Lösung?  
Das Live-Rendering der Schatten-/Sonnen-Simulation finden wir besonders ansprechend. Die technische Komplexität (Sonnenstand, Uhrzeit, Datum) wird so fast zum Game.

### Abgrenzung / Offene Punkte

- Welche Abgrenzungen habt ihr bewusst vorgenommen und damit nicht implementiert? Weshalb?  
Es wurde kein Backend-Implementiert, weil dies für einen POC zu weit gegangen wäre. Selbst für einen ersten Release, um zu sehen ob die Anwendung Verwendung findet, wäre dies nicht notwendig. Falls dann weiterführende Funktionen wie z.B. Updaten der Meta-Informationen via Kunden-Feedback oder Hervorheben von Events/Lokalen/Firmen oder ähnlichem, müsste dieser Entscheid nochmals überdenkt werden.