pFlow

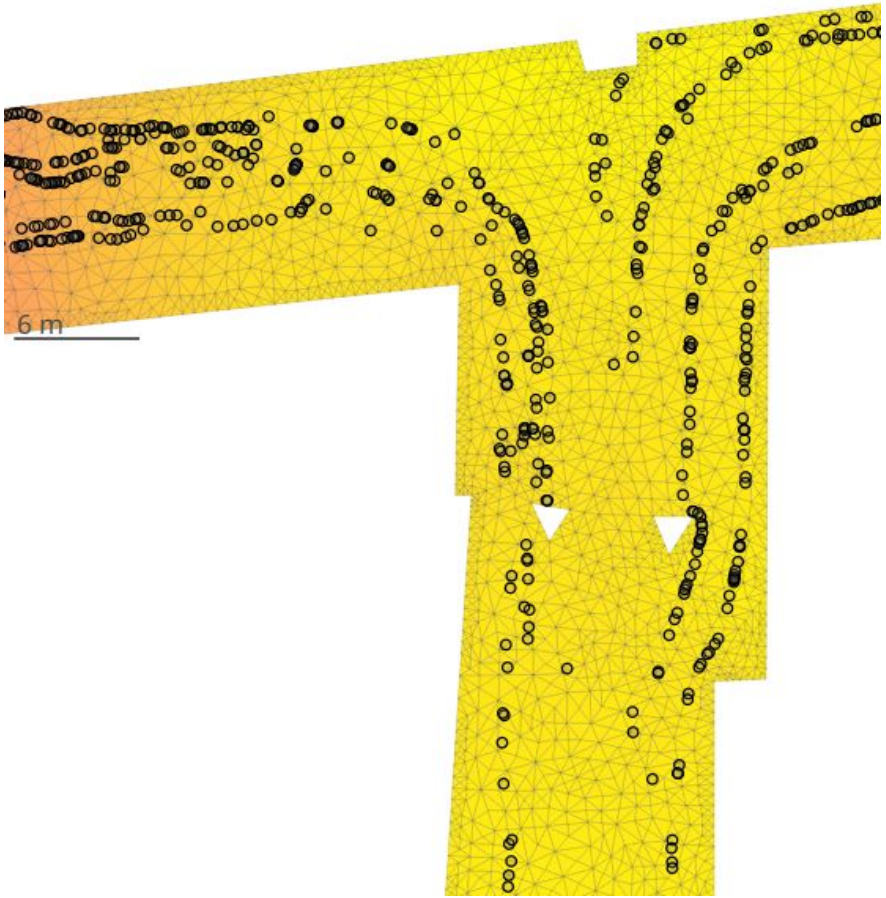
Barend, Dominik, Fabian, Lucas, Tobias

Inhaltsverzeichnis

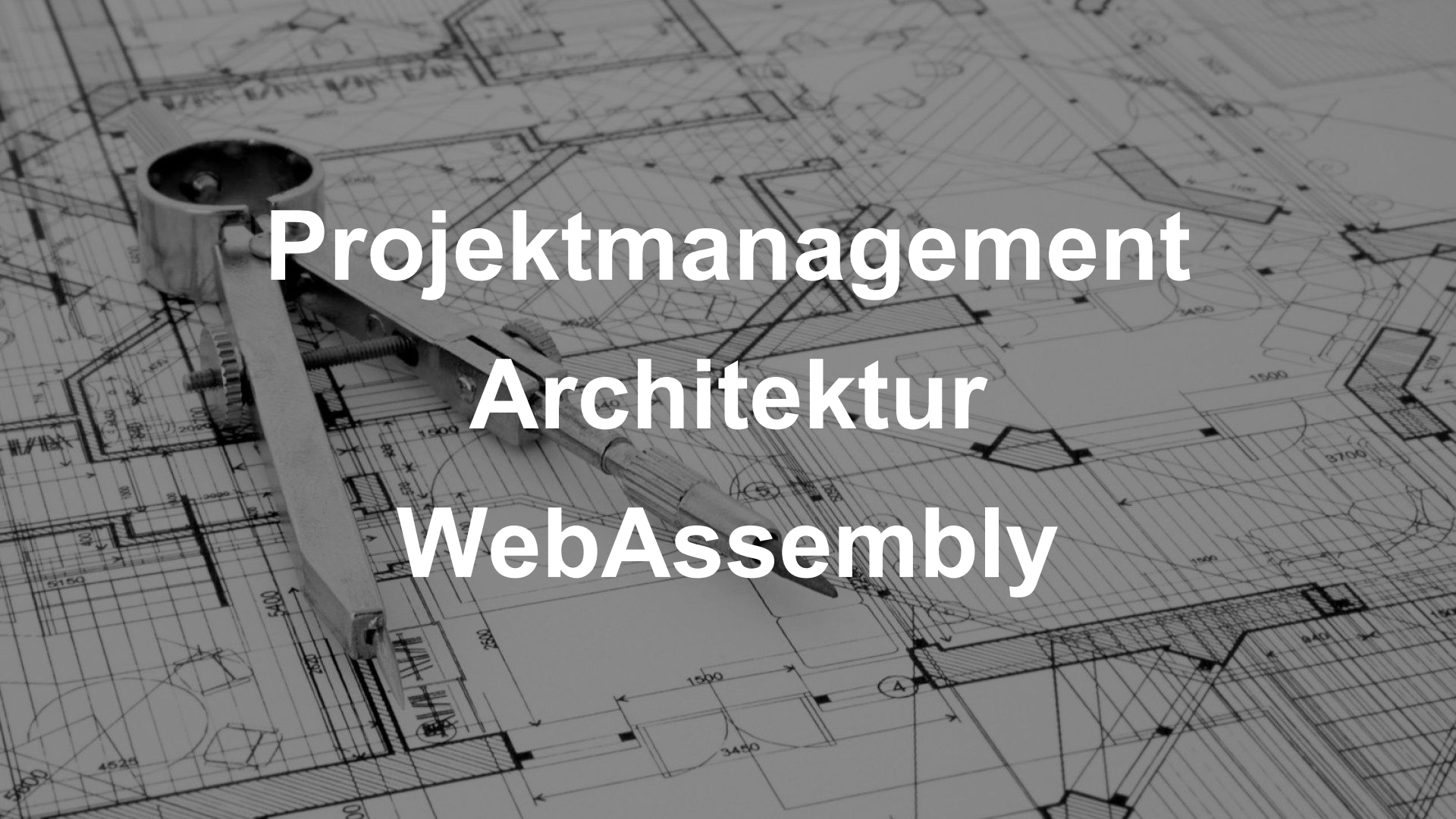
Was ist pFlow?

- Simulation von Personenströmen
  - Paniksituationen
  - Definierte Gebiete und Ausgänge
  - Personen Verhalten Modellierbar

- Aktives Forschungsprojekt
  - Frau Axthelm
  - Teamprojekte
  - HiWis



Live Demo



Projektmanagement
Architektur
WebAssembly

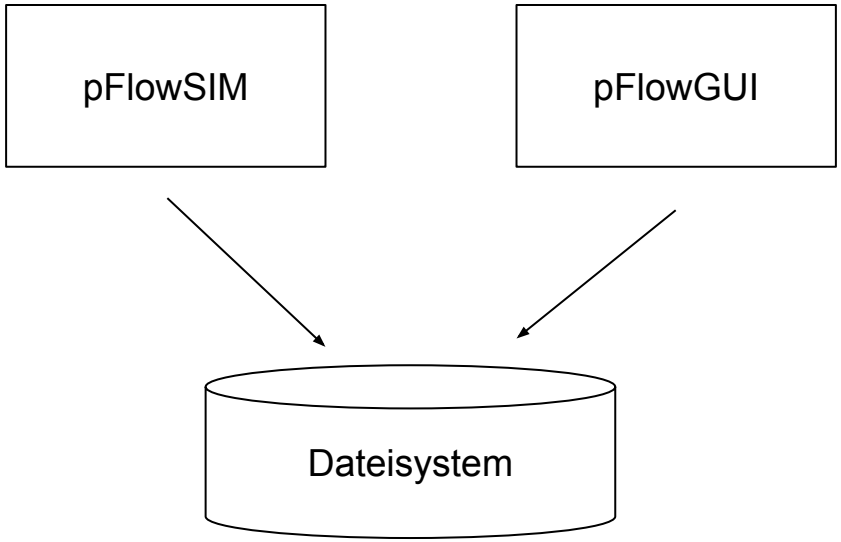
Projektmanagement

- Teamstruktur
  - 6 Personen
  - 1 Projektleiter
  - 4 Arbeitsgruppen
    - pFlow_sim
    - pFlow_server
    - pFlow_optimierung
    - pFlow_wui

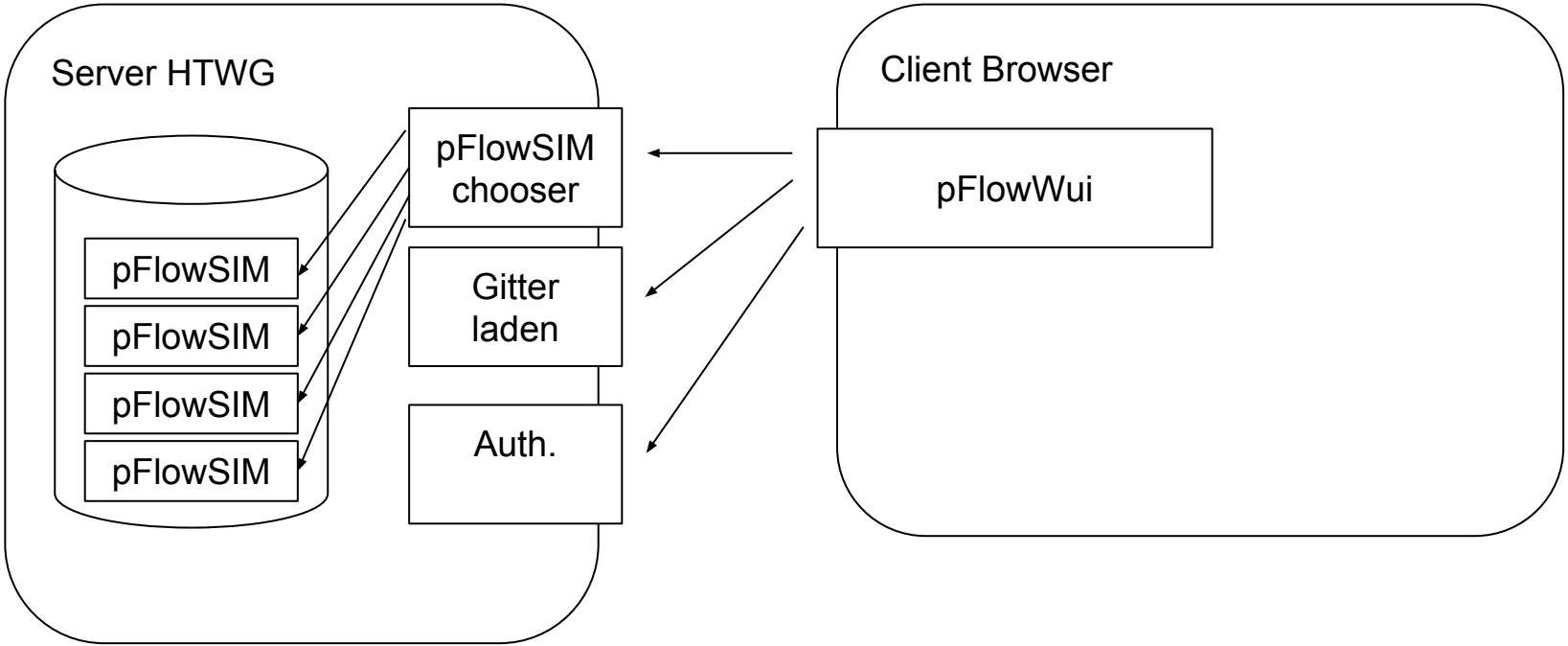
- Meetings:
  - 2 interne Treffen die Woche
  - Weekly Standup
  - regelmäsige absprach mit Project Owner

- Zusammenarbeit
  - git-Lab
    - issues
    - git-wiki
  - Discord

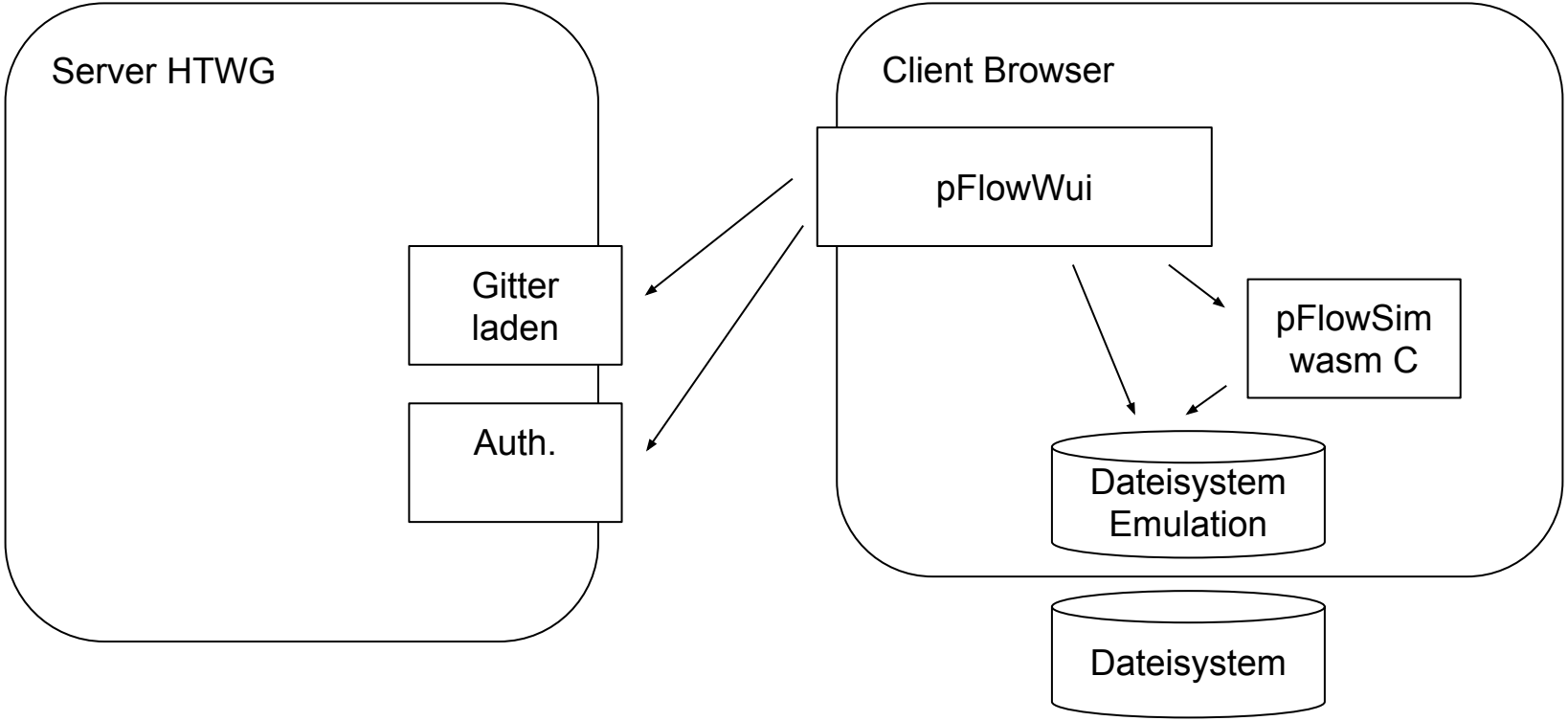
Architektur - 01.10.2018



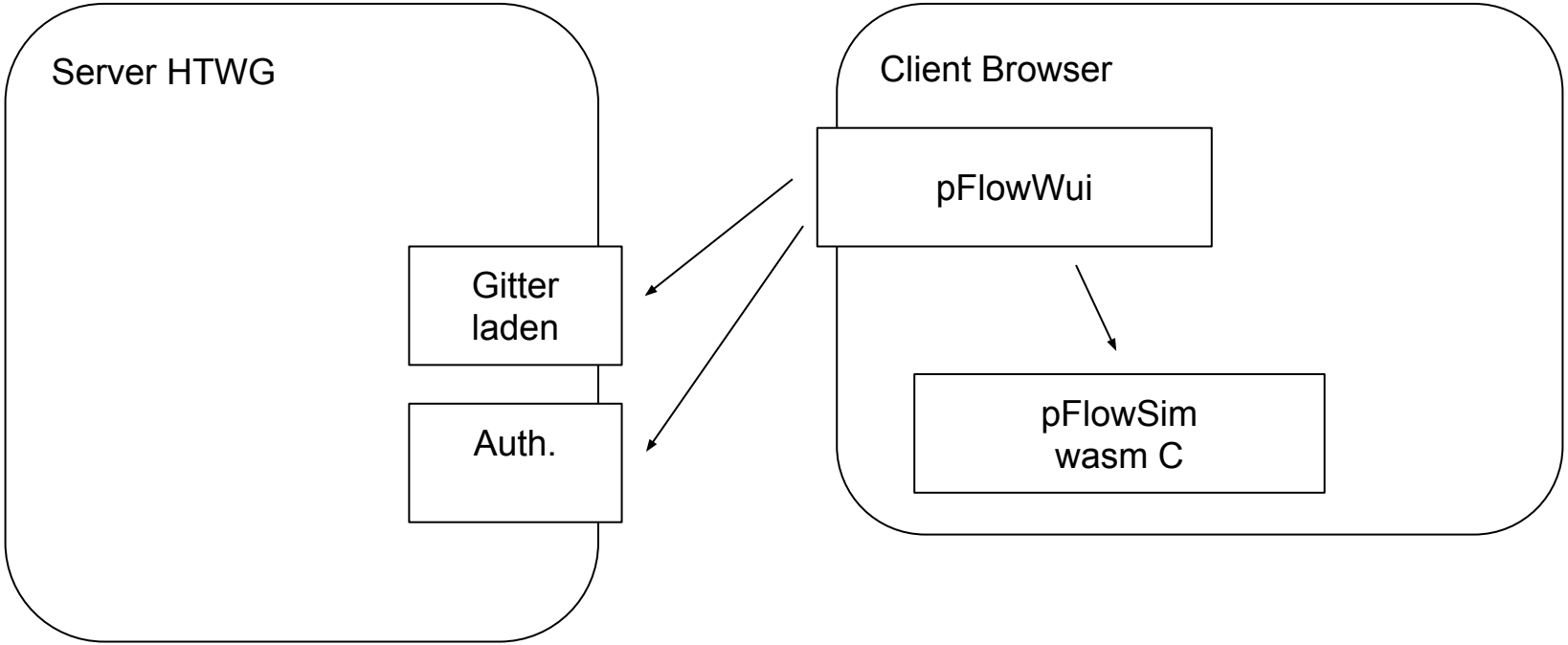
Architektur - Möglichkeit 1



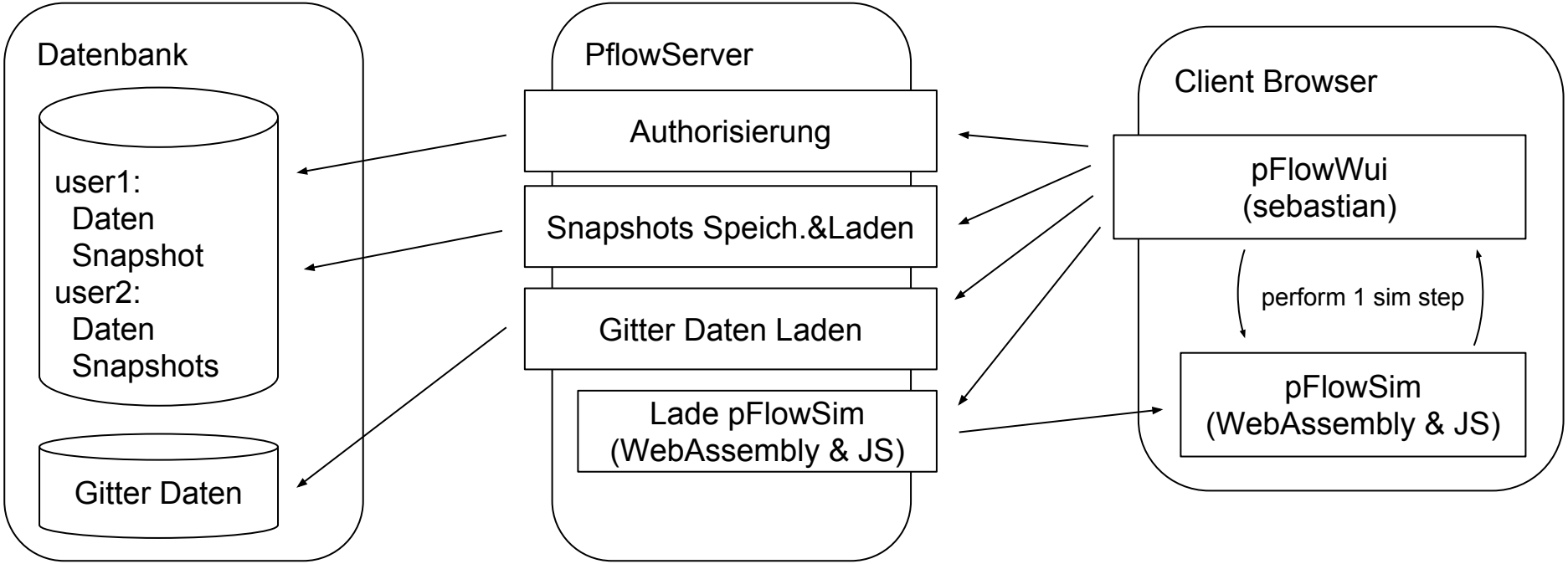
Architektur - Möglichkeit 2



Architektur - Möglichkeit 3



Architektur - das Ziel



Webassembly - Proof of Concept - Qt

- Assembly Format für Browser
- Qt -> Webassembly
- Automations Script:

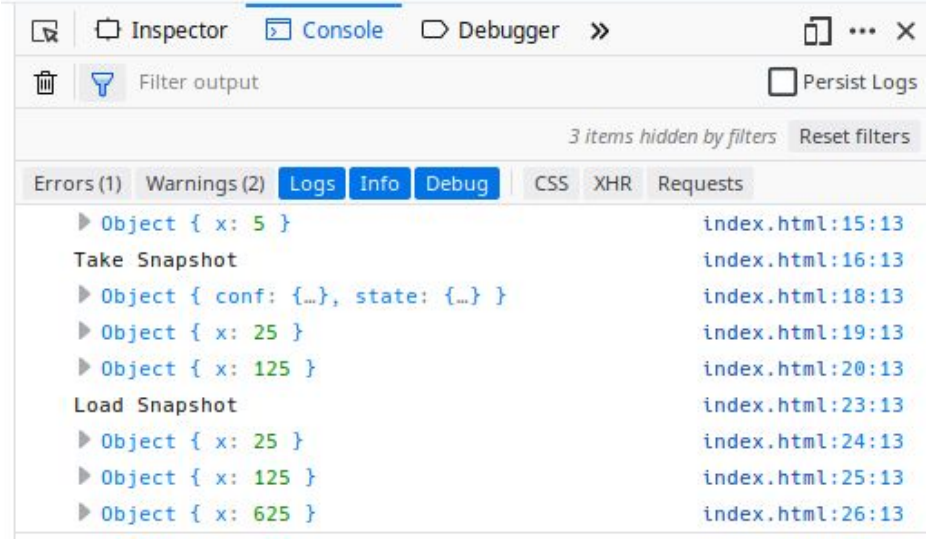
```
$ ./wasm_toolchain.sh help
wasm_toolchain.sh: Setup, Build and work with pflowGui & webassembly

USAGE:
  ./wasm_toolchain.sh [SUBCOMMAND]

SUBCOMMAND:
  setup           download emsdk, download qt, setup for webassembly, build qt
  build           build pflowGui to webassembly
  scratch         build pflowGui to webassembly from scratch
  open            open webassembly html File in Browser
  info            print info links to understand everything
```

Webassembly - Proof of Concept - MiniSim

```
1 class PFlowSim {
2   private:
3     State state;
4     Configuration conf;
5
6   public:
7     PFlowSim(Configuration conf)
8         : conf(conf)
9     {
10        State state;
11        state.x = 1;
12        this->state = state;
13    }
14 ...
15 }
16
17 struct State {
18   float x;
19 };
20
21 struct Configuration {
22   float constant;
23   bool useMultiply;
24 };
```

```
1 #include <emscripten/bind.h>
2 #include <emscripten/val.h>
3
4 using namespace emscripten;
5
6 // Binding code
7 EMSCRIPTEN_BINDINGS(my_class_example) {
8
9   value_object<Configuration>("Configuration")
10    .field("constant", &Configuration::constant)
11    .field("useMultiply", &Configuration::useMultiply)
12    ;
13
14  ...
15 }
```

Webassembly - Proof of Concept - MiniSim



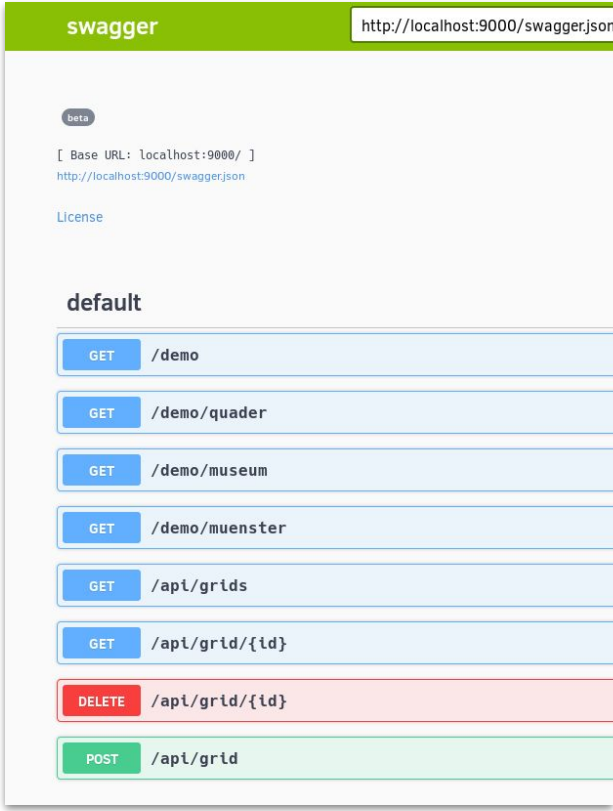

```
1 // Js Object will be automatically converted to a C struct
2 var configuration = {
3     constant: 5,
4     useMultiply: true
5 };
6
7 // Allocate a new Instanze of PflowSim
8 var pFlowSim = new Module.PFlowSim(configuration);
9 console.log(pFlowSim.perform_simulation_step());
10 console.log("Take Snapshot")
11 var snapshot1 = pFlowSim.take_snapshot()
12 console.log(snapshot1);
13 console.log(pFlowSim.perform_simulation_step());
14 console.log(pFlowSim.perform_simulation_step());
15
16 pFlowSim.load_snapshot(snapshot1);
17 console.log("Load Snapshot");
18 console.log(pFlowSim.perform_simulation_step());
19 console.log(pFlowSim.perform_simulation_step());
20 console.log(pFlowSim.perform_simulation_step());
21
22 // Don't forget to delete the Object (because C++ doesn't have GC)
23 pFlowSim.delete();
```

Live Demo
Webassembly


Backend
Datenbank
Google
google.com/datacenters

Backend: Server

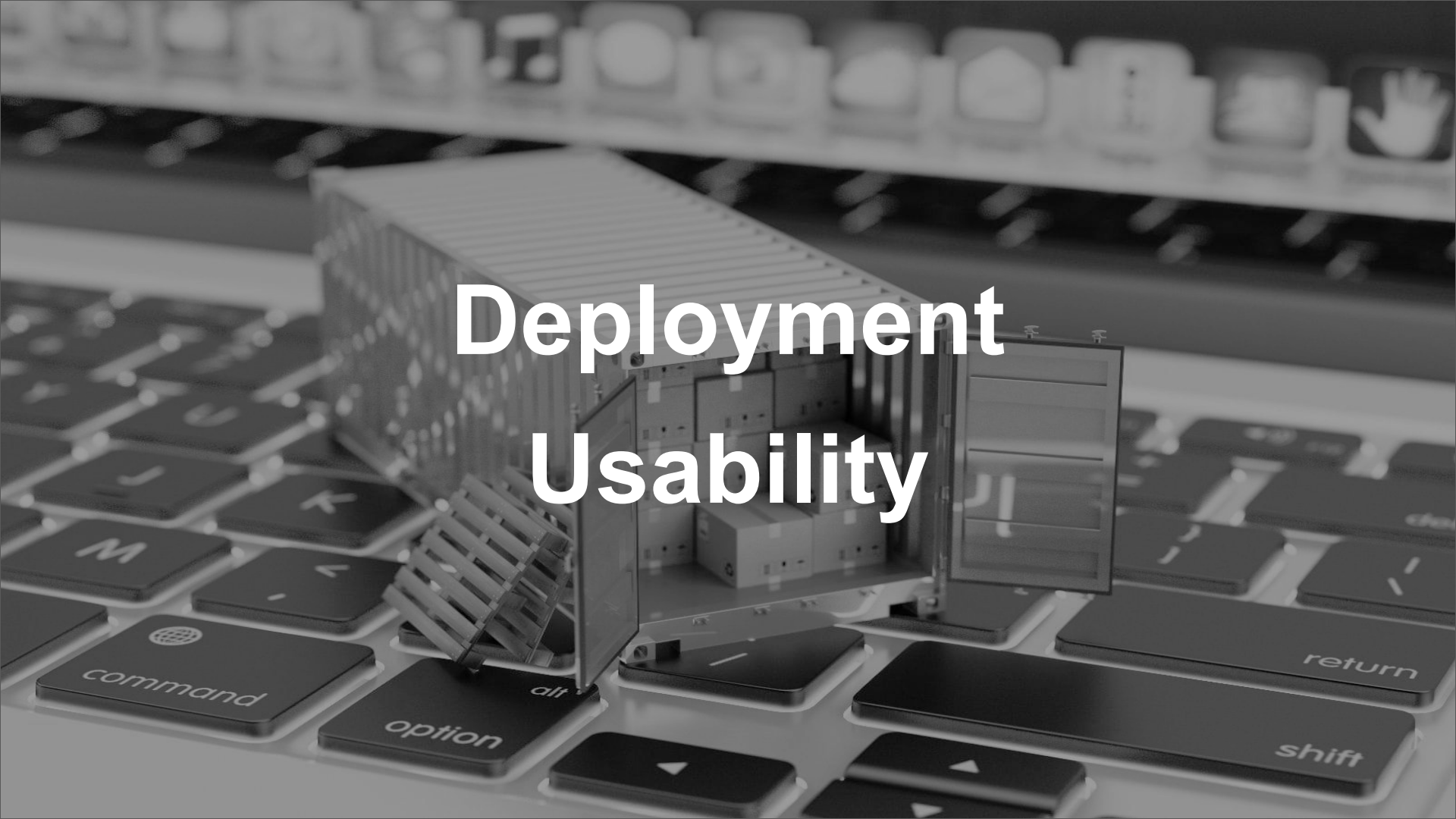
- Verwendung bekannter Techniken aus Web-Technologien
- Endpoints
  - Grid Endpoint
    - Erstellen eines Grids
    - Löschen eines Grids
    - Einzelne Grids einsehen
    - Mehrere Grids einsehen
    - Tests
  - Dokumentation der Endpoints über Swagger
  - Anbindung der Endpoints an eine MongoDB
- Deployment
  - Auf Frau Axthelms Rechner
  - Bei Heroku
- Dokumentation zum Server im GitLab-Wiki zu finden



Datenbank: MongoDB

- Vorteile:
  - Lernt man kennen bei Herrn Boger in Softwarearchitekturen
  - JSON Strukturen, praktisch im Web für Anbindung an JavaScript
  - Aufgesetzt über Docker
- Endpoints erstellen Einträge in die MongoDB
- Dokumentation zur MongoDB im GitLab-Wiki zu finden

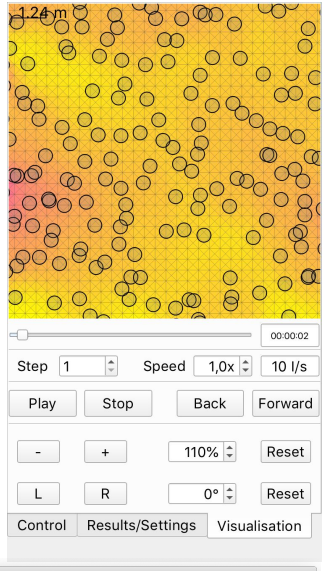
Docker ❤ MongoDB

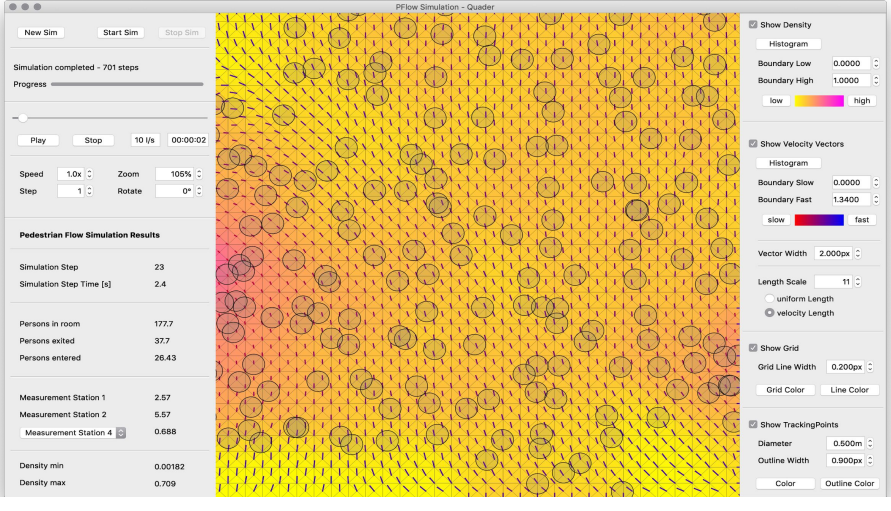


Deployment Usability

Deployment

- Integration von pFlowSim in Qt GUI Project
  - Entfernung von Dateisystemzugriffen in pFlowSim für GUI
  - Verwendung von pFlowSim ohne Qt GUI weiterhin möglich
- Static Build und Deployment der Qt GUI Application
  - funktioniert für
    - macOS
    - Webassembly
  - Probleme bei
    - Windows (8 Stunden)
    - Linux (keine Fonts)
- Proof of Concept Qt Smartphone/Tablet App
  - iOS
  - Android





Usability

- Speichern/Laden der aktuellen Simulationskonfiguration im JSON Format
- Simulationsbeispiele (Konfiguration und Gitter) built in
- Anzeige und Plot der Monitoring Daten in der GUI
- File Export
  - Simulationsergebnisse im CSV Format
  - Visualisierung als png/jpg Image
- Verbesserung der Usability der GUI
  - Shortcuts
  - Step, Back/Forward
  - Visualisierung
    - Full Screen
    - Zoom, Rotate
    - Antialiasing
    - Konfiguration Color/Width

Dokumentation

- Dokumentation im Markdown Format im Gitlab Wiki
- User
  - Verwendung der Qt PFlowSim GUI
  - Erstellen eines Gitters mit Gmsh Gittergenerator (updated)
  - Erstellen einer Gitterdatei (updated)
  - Erstellen einer Simulationskonfiguration im JSON Format
- Developer
  - Build von PFlowSim mit/ohne GUI
    - macOS, Linux, Windows
    - iOS, iOS Simulator, Android
  - Debugging mit Qt Creator, Xcode
  - Dokumentation der JSON Simulationskonfigurationsdatei als JSON Schema
  - Architektur, Webassembly, Backend

Trackpoint Collision

Collision Detection- proof of concept

- Benutzt Quadratur wie TRI
- Fertig:
  - TrackPoints in QUADRA struktur allokiert und zugewiesen im init schritt
  - Code zum Einfügen von TrackPoints in jedem simulationsschritt (muss noch in tracking eingeordnet werden)
  - Vergleich Trackpoints distanzen in nebenliegenden QUADRAs:
    - zeitmessung
- Noch Fehlend:
  - Eigentliche Detection und Vermeidungsstrategie

Performanz?

Dieser vergleich ist aufwendig, also kommt in frage, ob Rechenzeit wert.

Test: 10000 Personen im Raum

Längste Messung: 98 600 636 ns

Paarweiser vergleich ohne Quadratur wäre für N Personen N! vergleiche. Mit Quadratur:

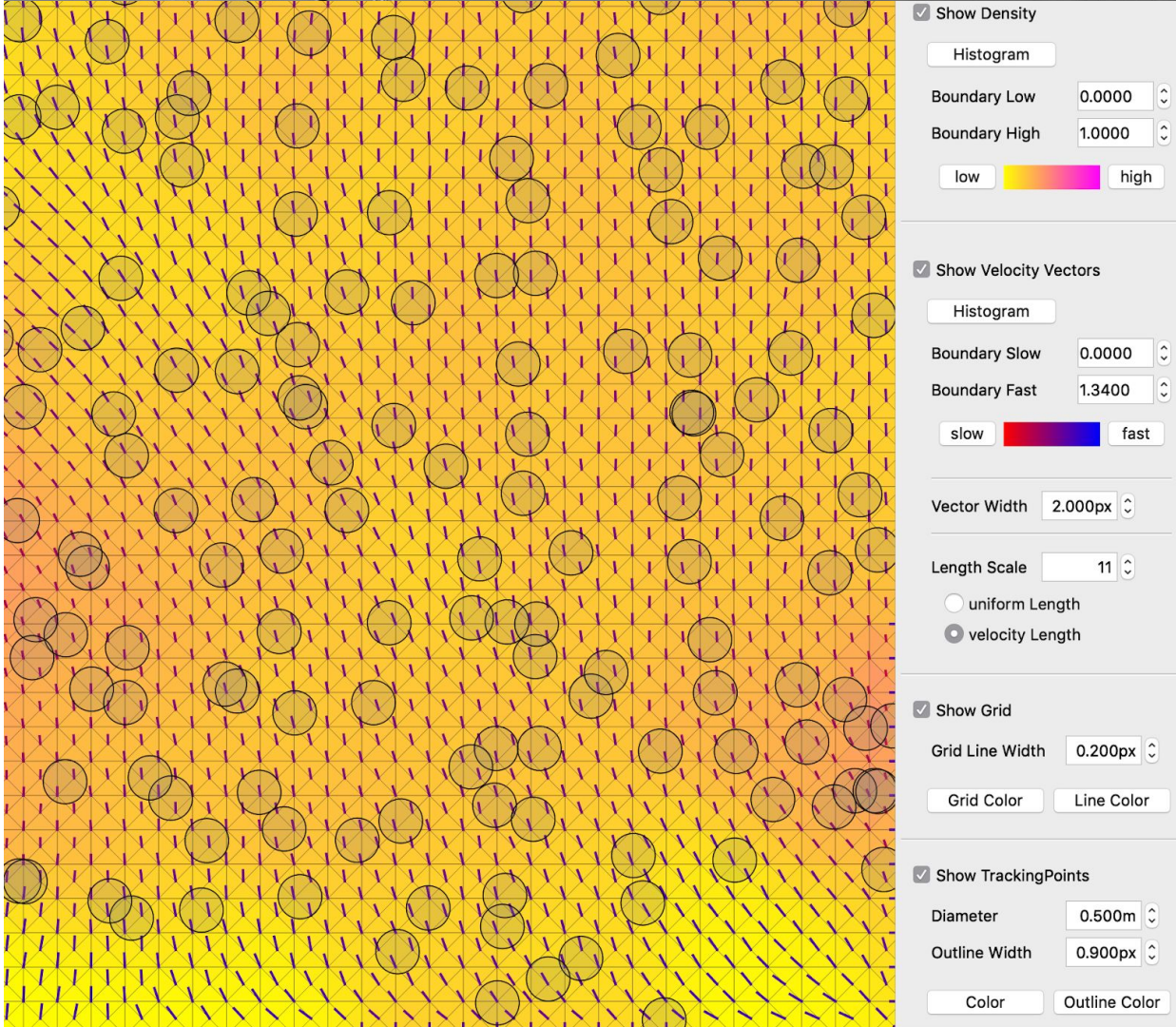
$$(N/QuadM*QuadN)!*QuadM*QuadN$$
$$+8*(N/QuadM*QuadN)^2*QuadM*QuadN$$

QuadM = 37; QuadN = 29; N = 10000

```
for (i=0 ; i<=QuadN ; ++i) {
  for (j=0 ; j<=QuadM ; ++j) {
    ij = i * QuadN + j;
    imin = max(0, i - 1);
    jmin = max(0, j - 1);
    imax = min(QuadN, i + 1);
    jmax = min(QuadN, j + 1);
    for (p_i = 0; p_i < quad[ij].Pnum - 1; ++p_i) {
      for (p_j = p_i + 1; p_j < quad[ij].Pnum; ++p_j) {
        min_dist = min_REAL(min_dist,
          sqrt(pow(video.TrackXYPosition[0][quad[ij].P[p_i]]
          - video.TrackXYPosition[0][quad[ij].P[p_j]], 2.0)
          + pow(video.TrackXYPosition[1][quad[ij].P[p_i]]
          - video.TrackXYPosition[1][quad[ij].P[p_j]],2.0)));
      }
      for (l = imin; l <= imax; ++l) {
        for (ll = jmin; ll <= jmax; ++ll) {
          ij0 = l * QuadN + ll;
          for (p_l = 0; p_l < quad[ij0].Pnum; ++p_l) {
            min_dist = min_REAL(min_dist,
              sqrt(pow(video.TrackXYPosition[0][quad[ij].P[p_i]]
              - video.TrackXYPosition[0][quad[ij0].P[p_l]], 2.0)
              + pow(video.TrackXYPosition[1][quad[ij].P[p_i]]
              - video.TrackXYPosition[1][quad[ij0].P[p_l]], 2.0)));
          }
        }
      }
    }
  }
}
```

Vielen Dank

Gibt es noch Fragen?



Quellen

- https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.contguard.com%2F%2Ftemplates%2Fcontguard%2Fuser_files%2Fimage%2FiStock-501820982.jpg&imgrefurl=https%3A%2F%2Fwww.contguard.com%2Fmain.asp%3FWebLang%3DEN%26article%3D32788&docid=lv-29GUkCgROKM&tbnid=q0r3JYSHK03ypM%3A&vet=10ahUKEwiup7jytq7gAhWCYIAKHUIaDZIQMwhOKBEwEQ..i&w=1600&h=1067&client=firefox-b-d&bih=925&biw=1920&q=cargo&ved=0ahUKEwiup7jytq7gAhWCYIAKHUIaDZIQMwhOKBEwEQ&iact=mrc&uact=8
- https://fortunedotcom.files.wordpress.com/2015/06/screen-shot-2015-06-24-at-11-54-41-am.png
- https://i.ytimg.com/vi/lnTlYZ1JhM8/maxresdefault.jpg
- https://www.citynews-koeln.de/wp-content/uploads/2018/12/konzert-highlight-koeln-citynews.jpg
- https://miro.medium.com/max/960/1*sHBjNVI_WAIqNNLsZNzkFQ.png