Coupling of models for fire dynamics and pedestrian dynamics to estimate the required safe egress time (RSET)

Fire and Evacuation Modeling Technical Conference 2024

Kristian Börger, Mohcine Chraibi

BERGISCHE UNIVERSITÄT WUPPERTAL

JÜLICH Forschungszentrum

Outline

1. Importance of Software in modern Research

2. Motivation

3. JuPedSim ─ Python package with a C++ core to simulate pedestrian dynamics

4. FDSReader ─ Fast and easy-to-use Python reader for FDS data

5. FDSVismap ─ Python tool for waypoint-based assessment of visibility in performance-based fire safety design

6. Application example

7. Outlook
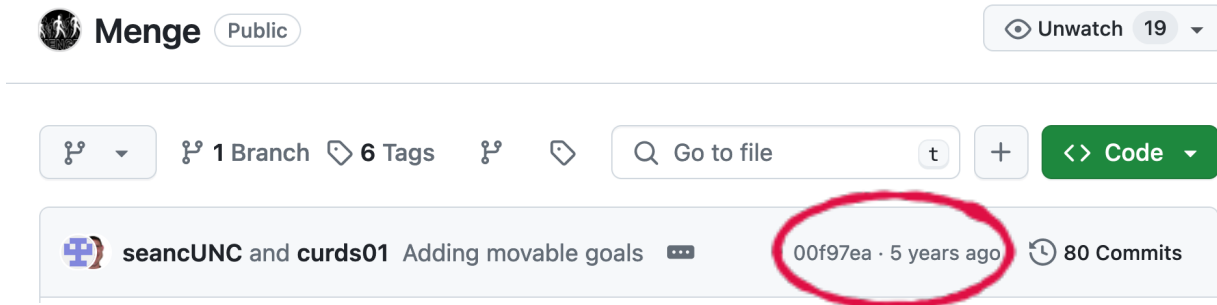
Importance of Software in modern Research

While commercial software often provides robust features, it comes with limitations that make it less suited for scientific pursuits.

- **High costs**

- **Closed-source nature**

- **Restrictive licensing**

**Open-source software offers several advantages crucial for researchers:**

- **Cost-Effectiveness**: Open-source software is typically free.

- **Transparency and Reproducibility:** Inspect, modify, and share.

- **Collaboration and Innovation:** Encourages collaboration, accelerating innovation and problem-solving.

- **Flexibility and Customization:** Adding new features or improving existing ones without waiting for commercial updates.

Motivation

Menge Public

Unwatch 19

1 Branch 6 Tags

Go to file t

Code

seancUNC and curds01 Adding movable goals ... 00f97ea · 5 years ago 80 Commits
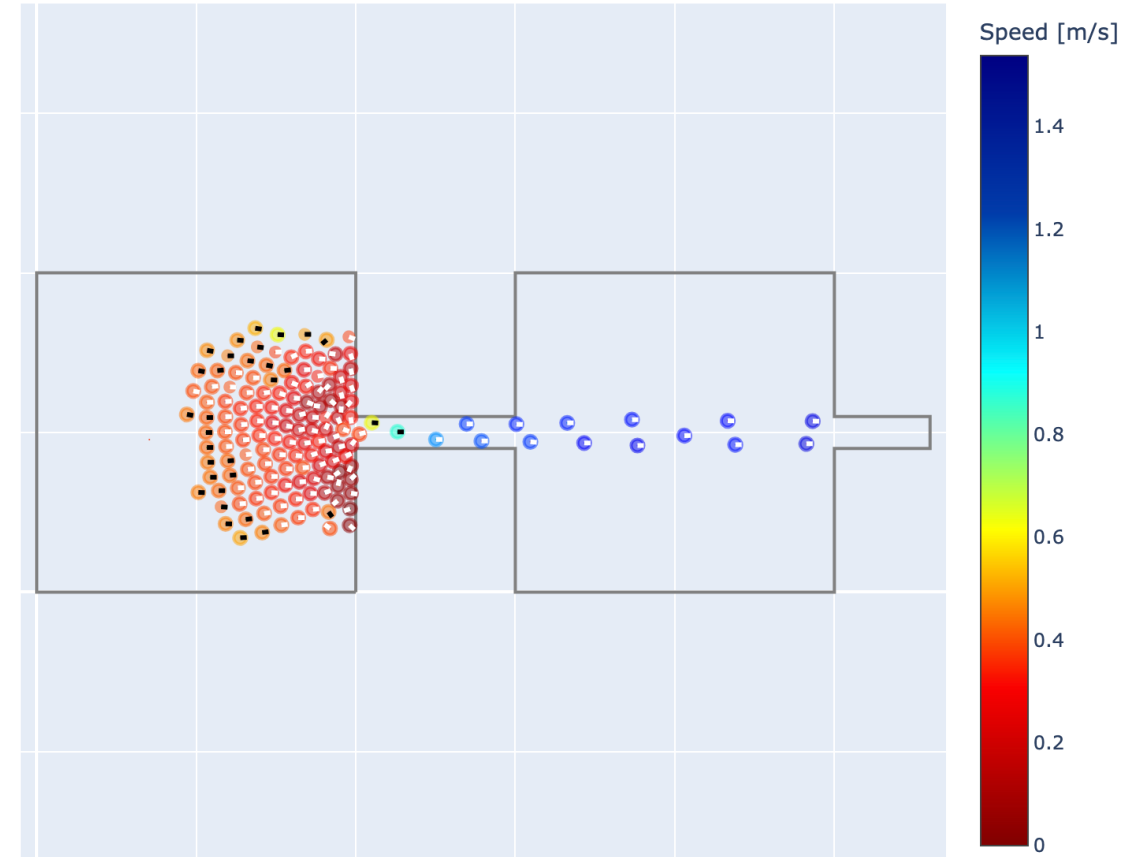
FDS+Evac no longer supported

Support for FDS+Evac has been discontinued as of FDS 6.7.8 (the last working version is FDS 6.7.7). FDS+Evac is an evacuation simulation module for FDS developed and maintained by VTT, Finland. Visit the FDS+Evac website for documentation, examples, verification, and validation of FDS+Evac.
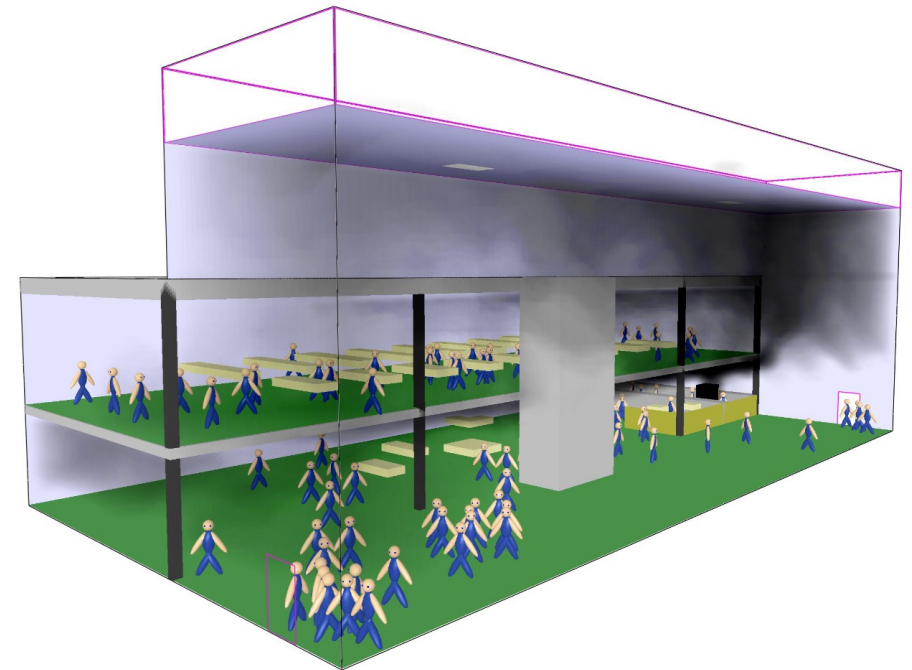
Source: Fire Dynamics Simulator with Evacuation: FDS+Evac Technical Reference and User's Guide

Slide 4

JuPedSim

- JuPedSim is a python library with a native core for performance relevant tasks.

- State of the simulation can be modified at every iteration, allowing to implement the **strategic** and **tactical** level in an ad-hoc fashion for quick experimentation.

- Basic elements to script behaviour are provided, i.e. way pointing, waiting at position, scripted queues, exits allowing to model more complex scenarios as before.

Architecture of JuPedSim

Python-API: research

C-API:
connection to other
software like SUMO

C++-core: basic components
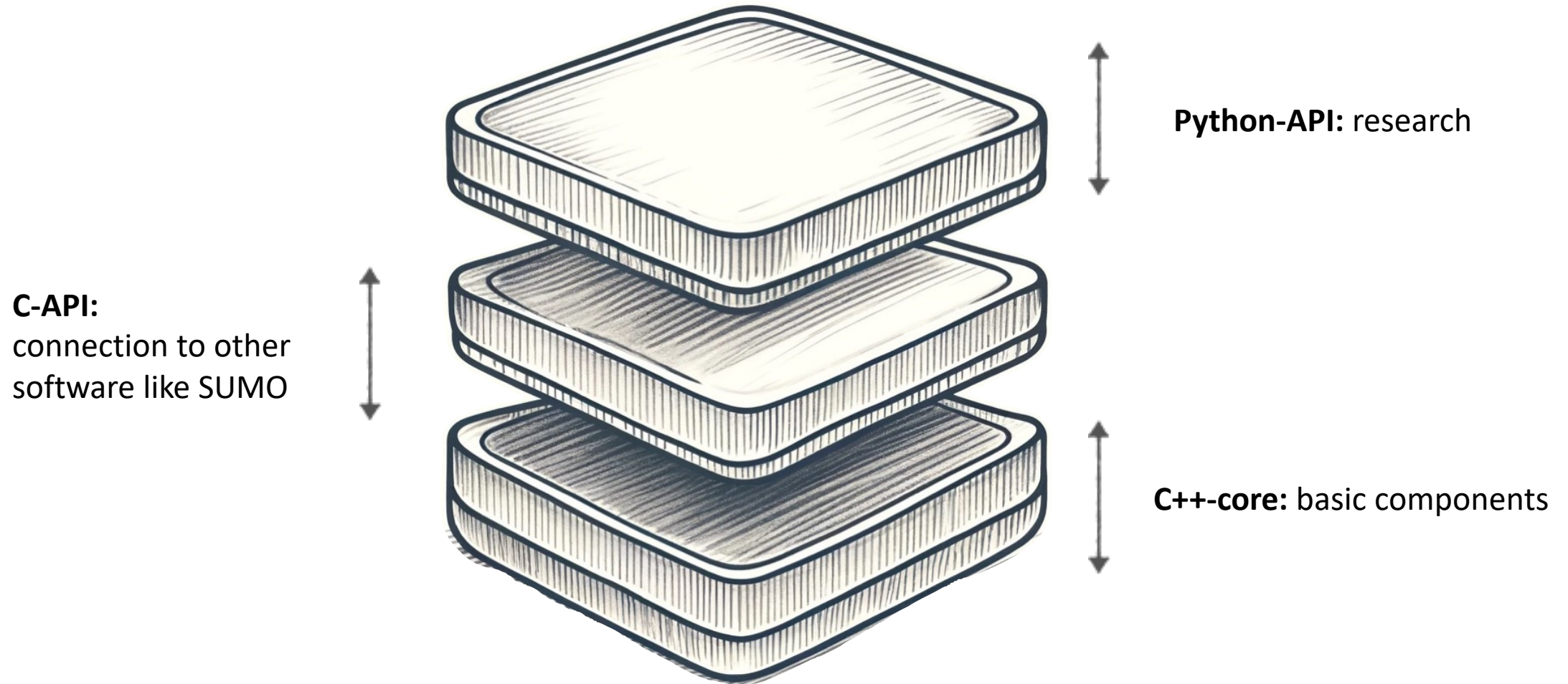
FDSReader

- FDSReader is a python library to read most of the FDS generated data outputs to map it to python data structures (numpy arrays).

- Data can be read from point devices and 2D or 3D slice files or boundary files.

- Metadata and obstructions geometry is also read by the module
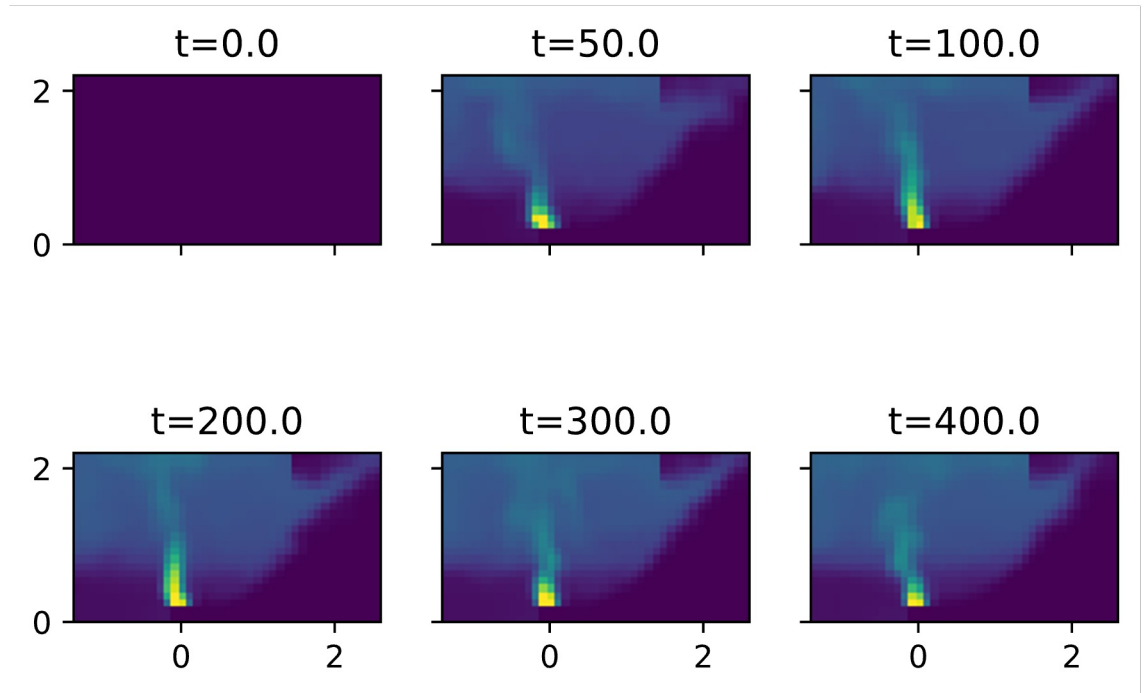
Visibility in Performance Based Design

- Visibility in performance based fire safety design is generally assessed by Jin's law

- Jin's law does not account for
  - Inhomogeneous smoke
  - Viewing angles
  - Obstructions in line of sight

- Visibility of 10 – 20 m is often considered a universal performance criterion
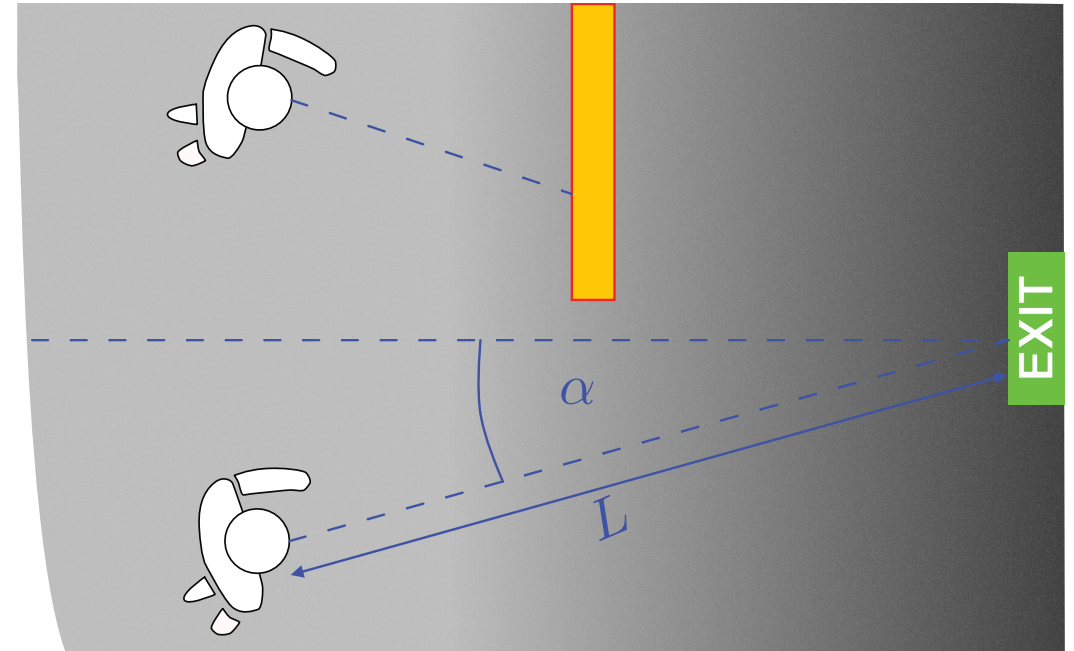
$$V = \frac{C}{\sigma}$$

C: dimensionless empirical factor,
   C = 8 for self illuminating signs,
   C = 3 for light reflecting signs
σ: Extinction coefficient [1/m]

FDSVismap - A Waypoint Based Approach to Visibility

Create visibility maps based on floorplans for performance based design:

- Compute visibility from any position x_i, y_i relative to different waypoints (exit signs) for inhomogeneous smoke

- Automatically import geometry and smoke data from CFD codes (E.g. FDS)

- Postprocessing of simulation results data with the *FDSVismap* Python package

Agent cell i, j
(x_i, y_j)

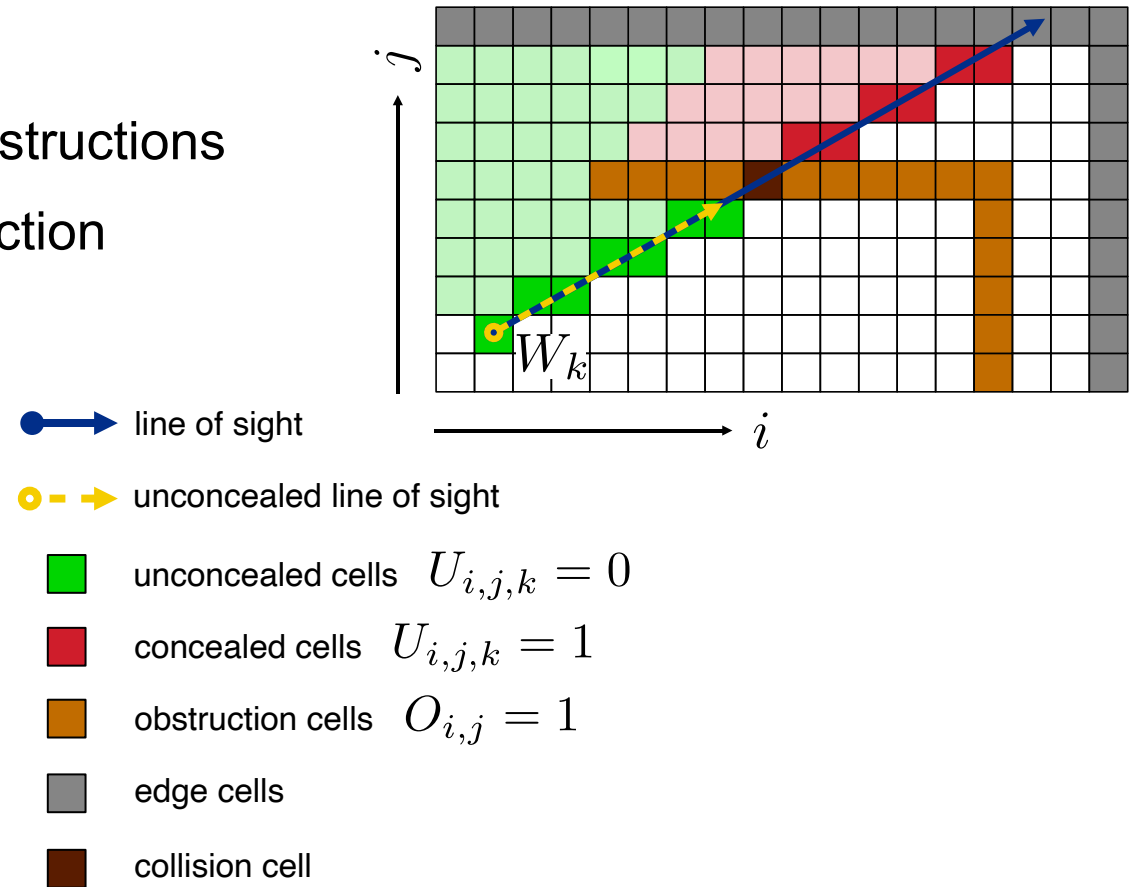$L_{i,j,k}$

θ

Exit W_k
(x_k, y_k)

α
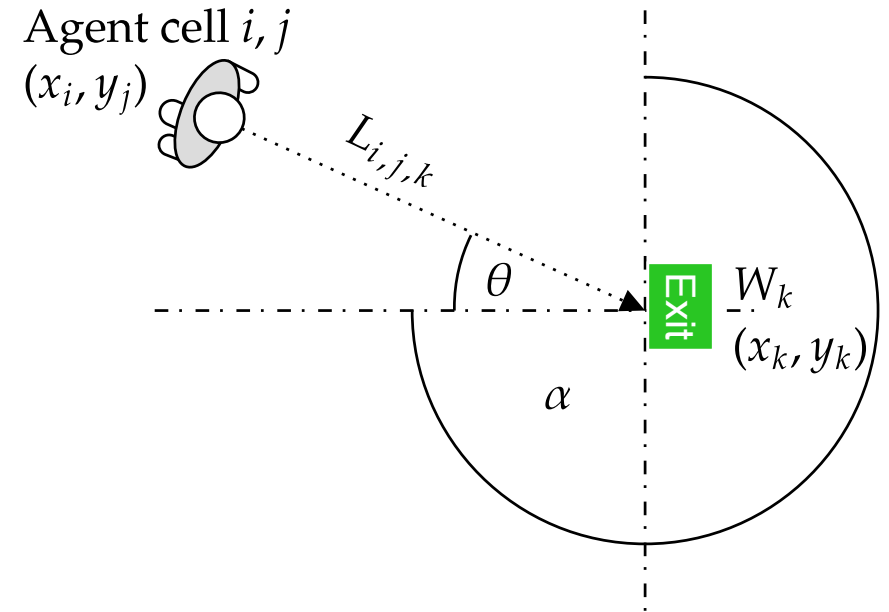
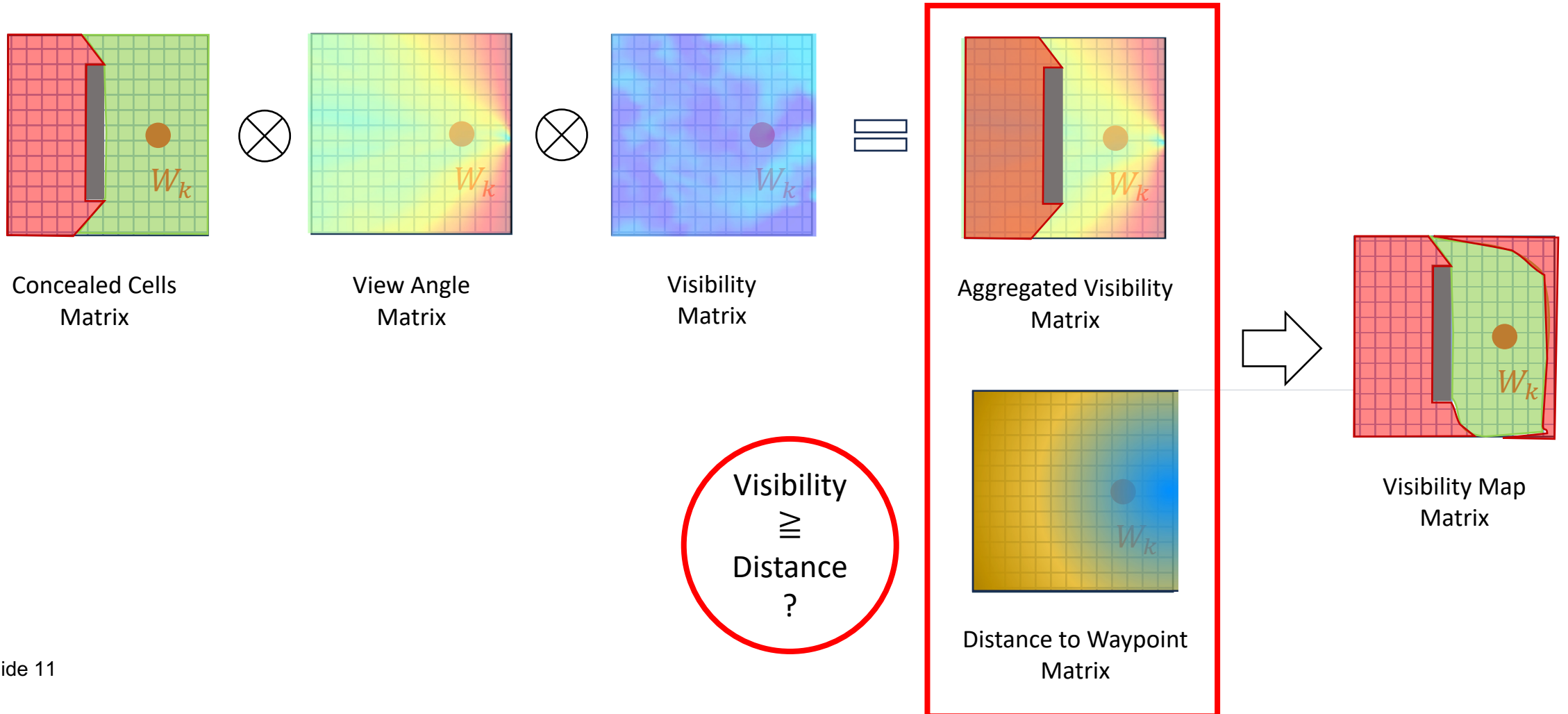# Ray Casting Algorithm

Ray casting is applied for

- Automated collision detection with FDS obstructions

- Computation of integrated / averaged extinction coefficient and visibility along line of sight

$$\bar{\sigma}_{i,j,k}^{t} = \frac{1}{|P_{\text{cells}}|} \cdot \sum_{p \in P_{\text{cells}}} \sigma_{k,p}^{t}$$

$$V_{i,j,k}^{t} = \min \left(U_{i,j,k} \cdot A_{i,j,k} \cdot \frac{C_k}{\bar{\sigma}_{i,j,k}^{t}}, V_{\max} \right)$$

line of sight

unconcealed line of sight

unconcealed cells $\quad U_{i,j,k} = 0$

concealed cells $\quad U_{i,j,k} = 1$

obstruction cells $\quad O_{i,j} = 1$

edge cells

collision cell

Creation of Visibility Maps (Boolean Maps)

Concealed Cells
Matrix

$\bigotimes$

View Angle
Matrix

$\bigotimes$

Visibility
Matrix

$=$

Aggregated Visibility
Matrix

Distance to Waypoint
Matrix
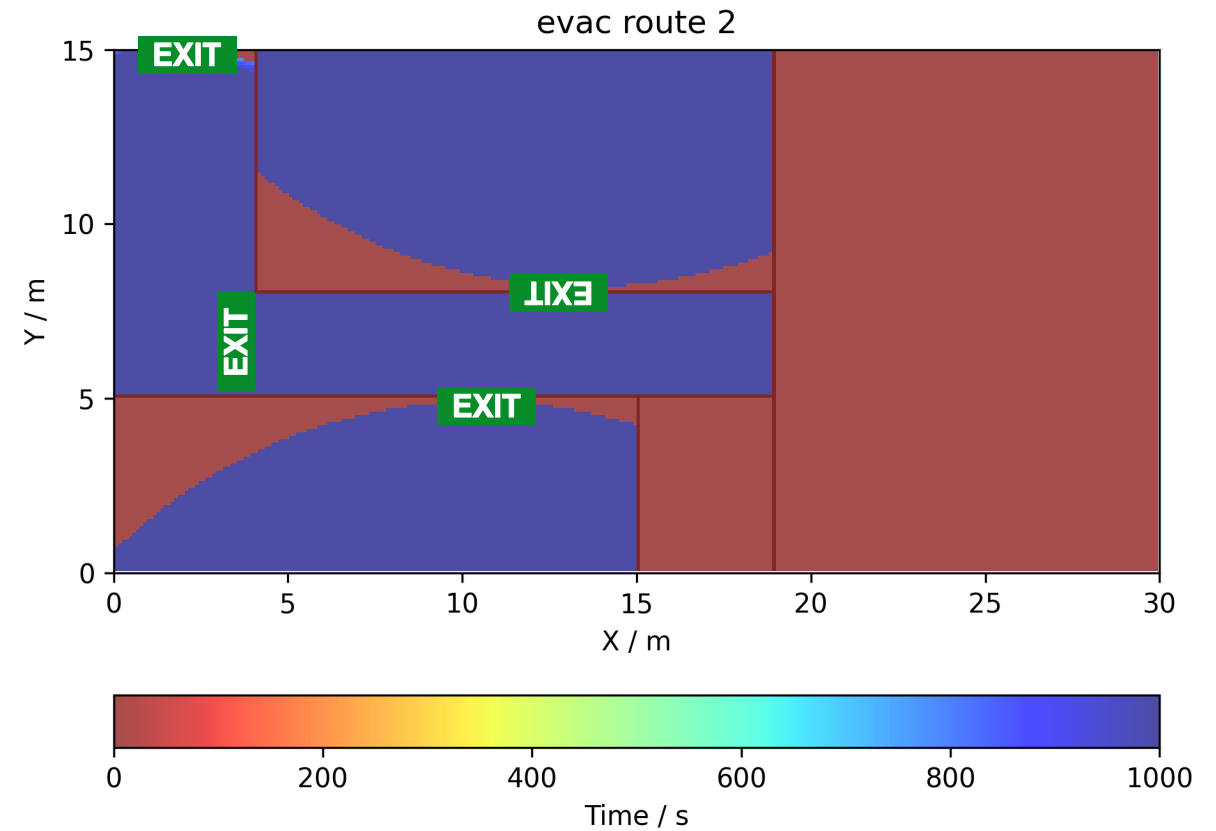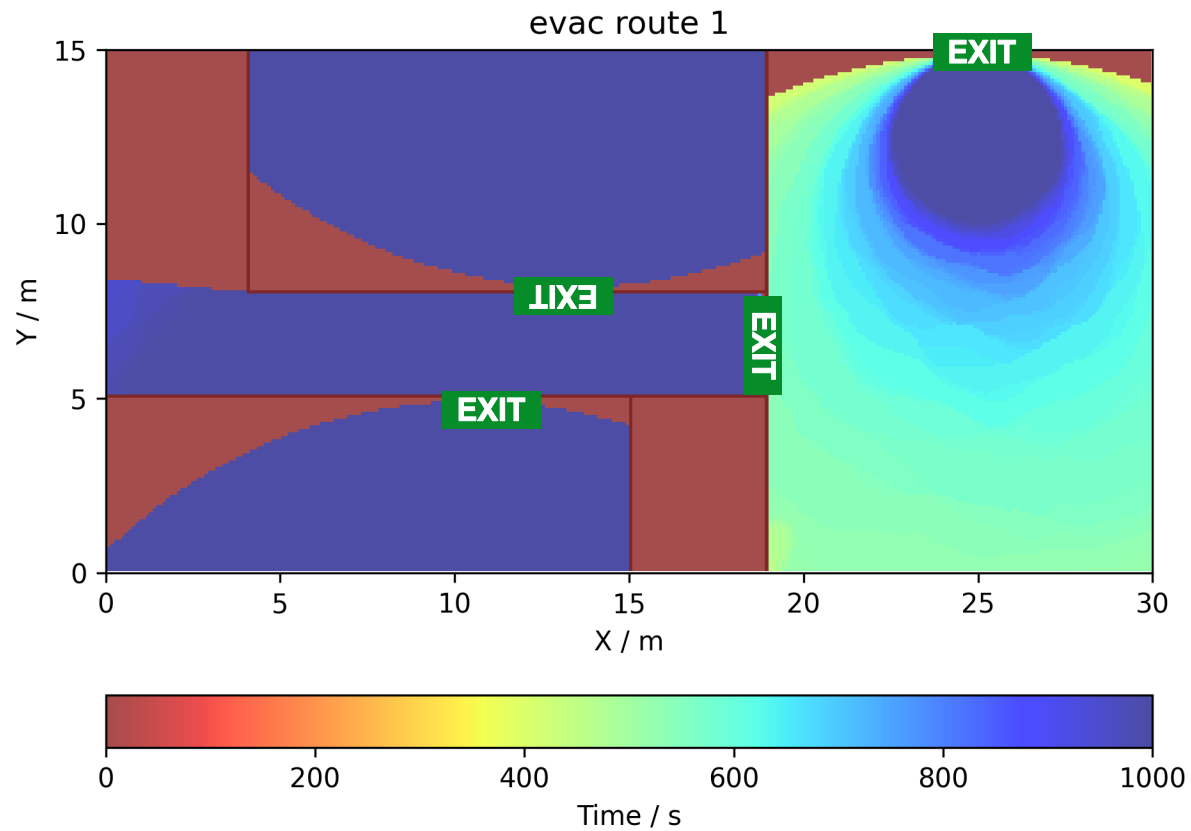
Visibility Map
Matrix

Visibility
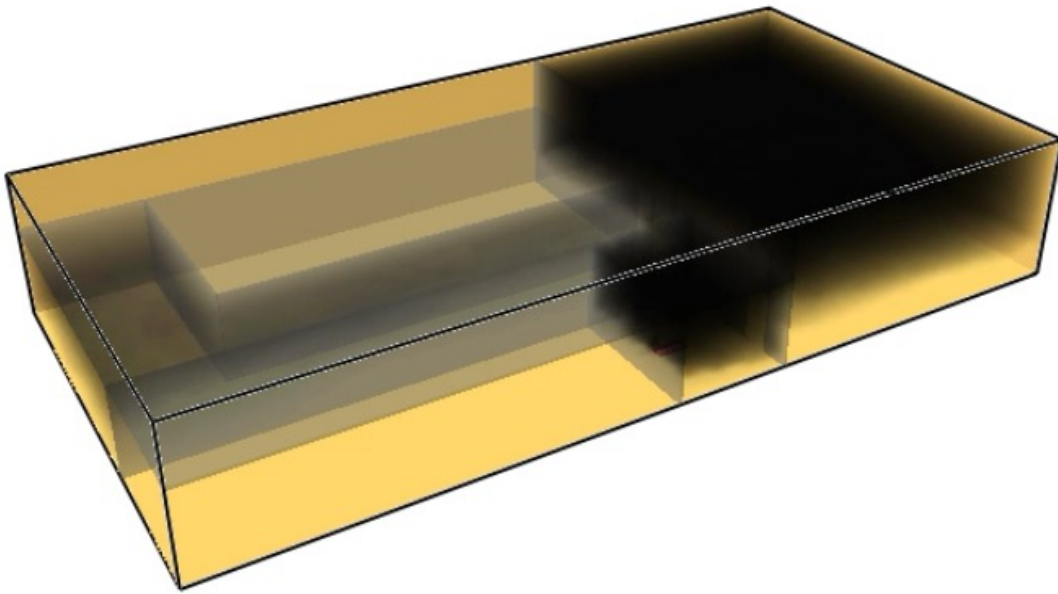$\geq$
Distance
?

Application Example

- Evac Route 1 is considered as default.

- Route decision is made based on visibility relative to the exit signs.

- Walking speed is changed based on visibility.

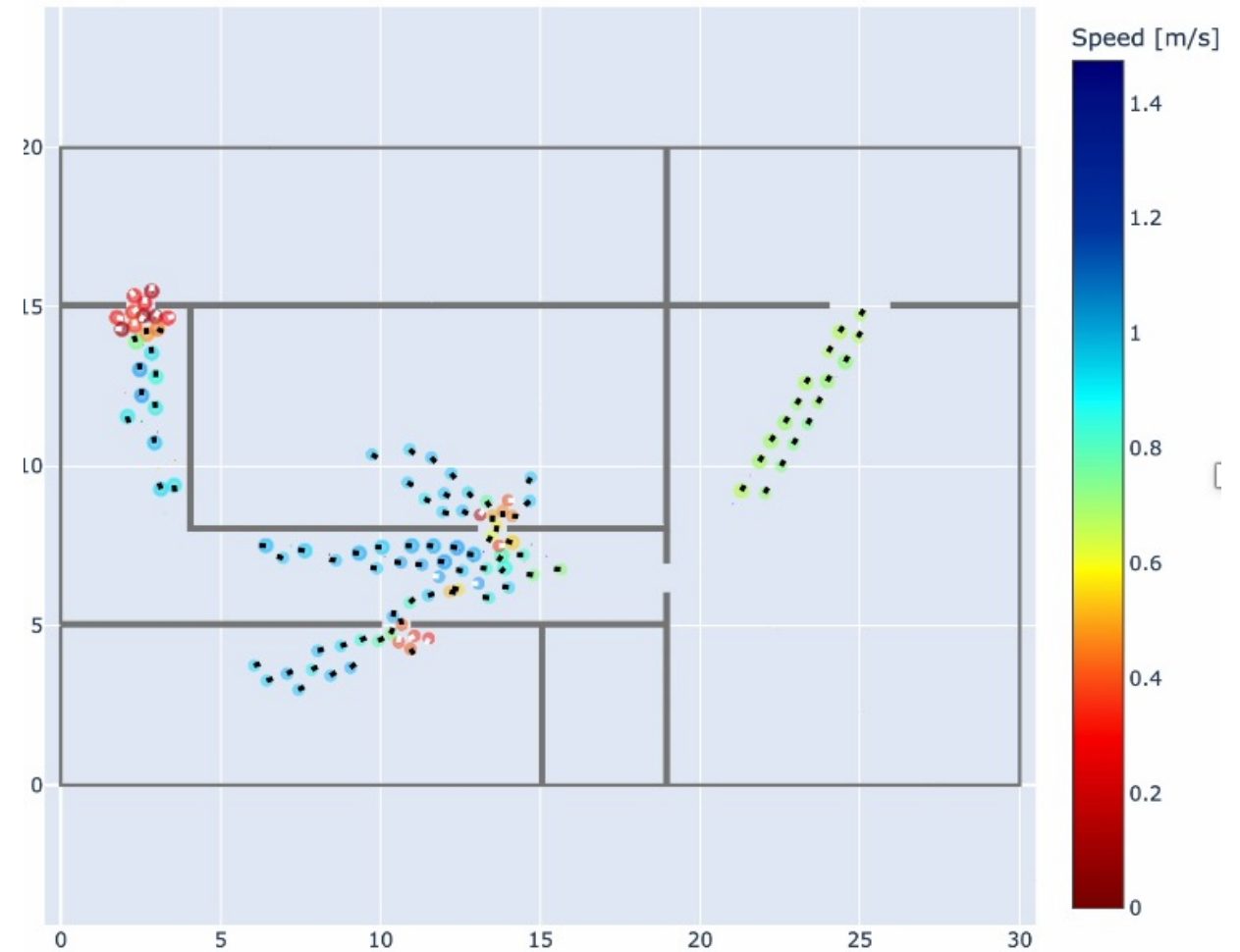- Agent attributes are updated at every timestep.
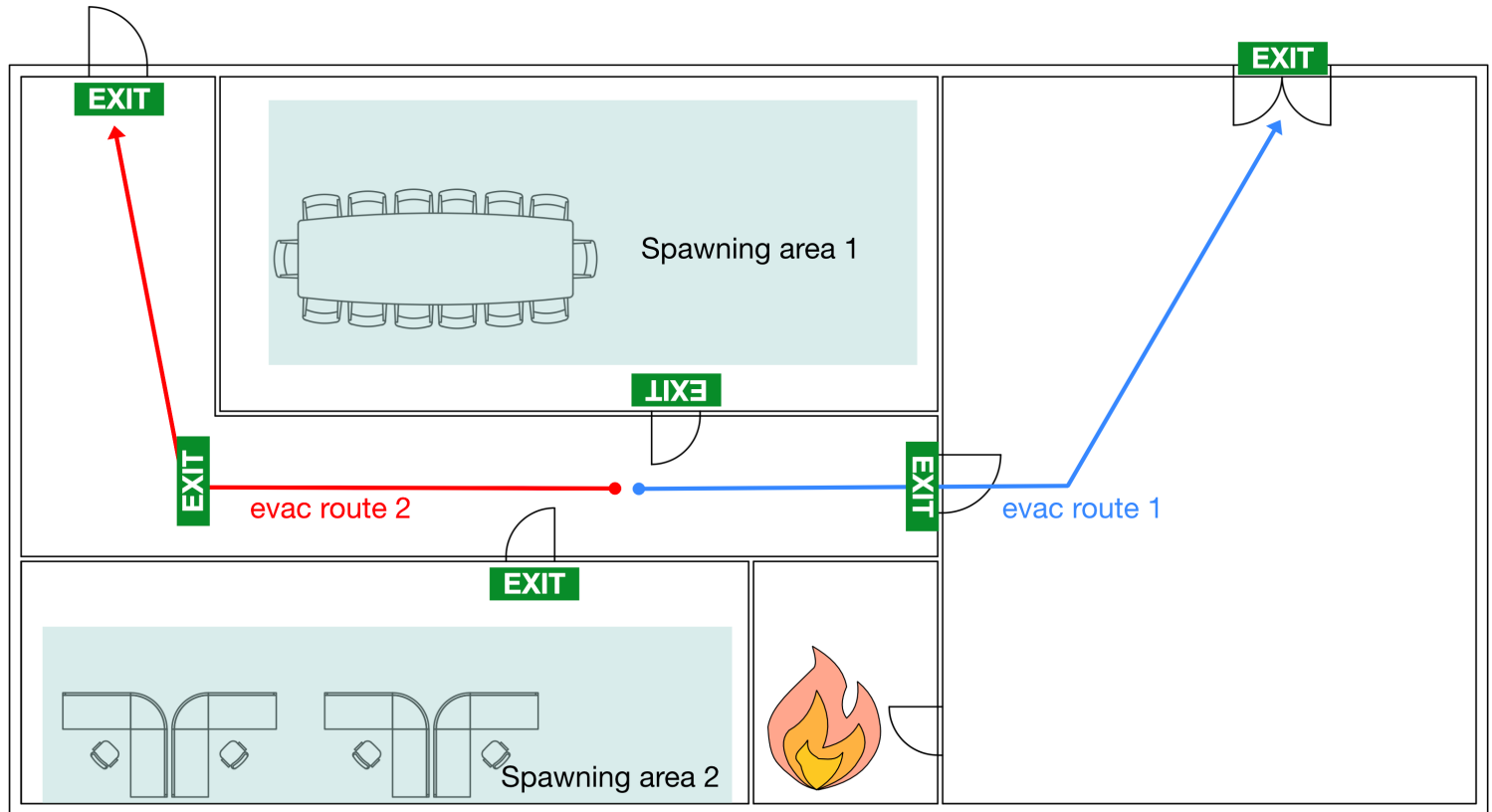
Creation of Visibility Maps (ASET Maps)

Application Example

Outlook

- More intelligent route decisions of agents based on knowledge about building geometry.

- Communication of agents may be integrated to avoid blocking of routes.

- FED to be considered as additional ASET criteria.

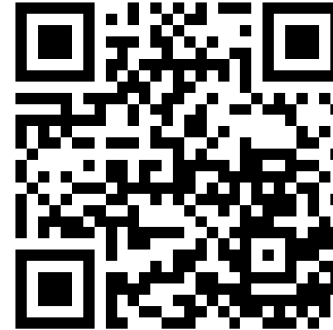- Two way coupling may allow the agent's actions to affect smoke distribution.

Thank you!

Coupling
FDSVismap
JuPedSim
on github

FDSVismap
on github

JuPedSim
on github

BERGISCHE
UNIVERSITÄT
WUPPERTAL

JÜLICH
Forschungszentrum