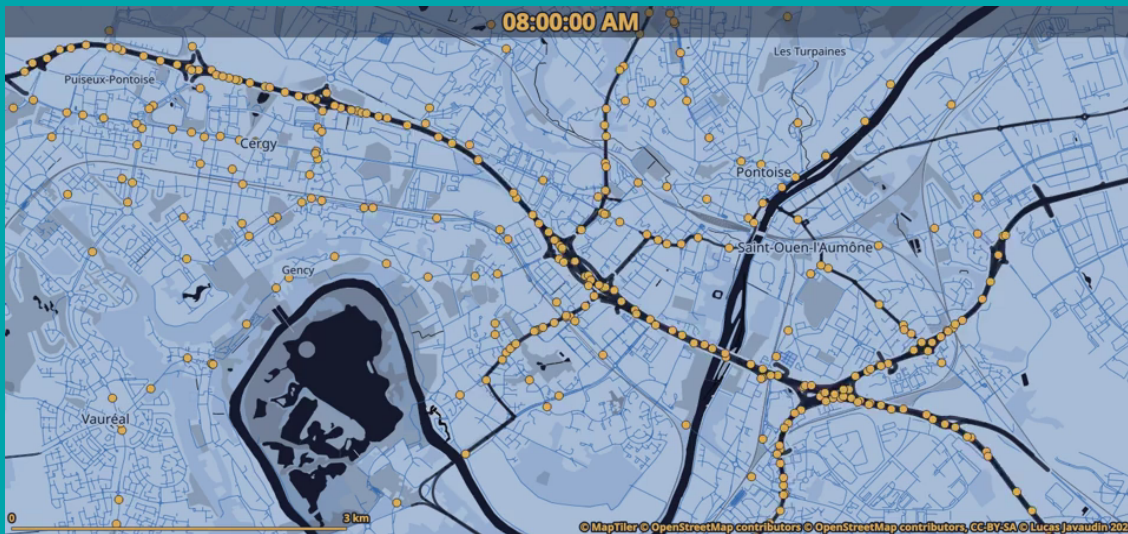


METROPOLIS2: a dynamic, mesoscopic, agent-based transport simulator

André de Palma, Lucas Javaudin
THEMA, CY Cergy Paris Université



November 2024

What is Metropolis?

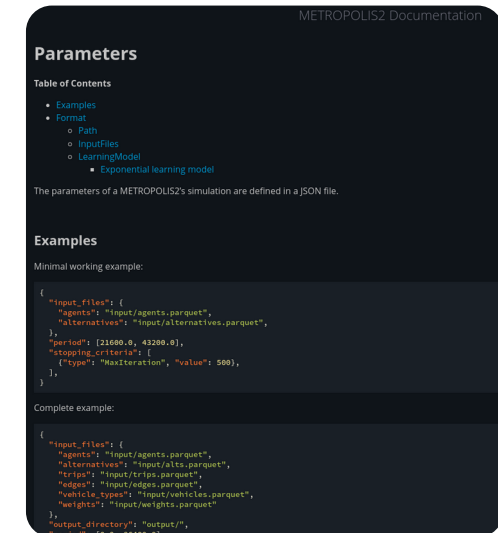
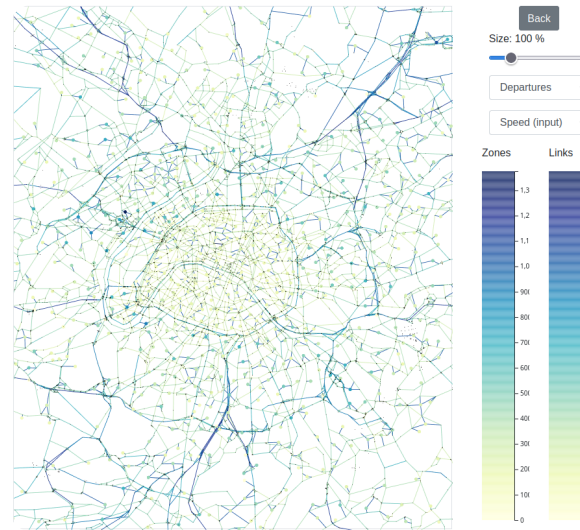
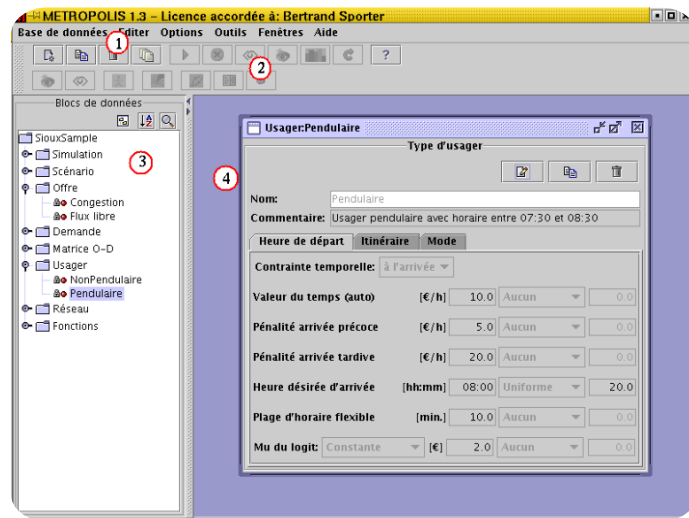
- METROPOLIS is dynamic, agent-based traffic simulator
- It is rooted in the theory of dynamic models (Vickey, 1969; de Palma, Ben-Akiva, et al, 1987; Arnott, de Palma and Lindsey, 1992)
- It combines:
 - A rigorous micro demand model (describing the choice tree)
 - Realistic congestion laws
 - Day-to-day iterative learning processes leading to a stationary state
- More suitable to evaluate short-, medium-, and (later) long-run policies



METROPOLIS: History

METROPOLIS History

- **1994-1995:** first simulator of METROPOLIS1 [C++] (A. de Palma, Y. Nesterov)
- **1996-2003:** simulator [C++] & Interface [Java] for METROPOLIS1 (A. de Palma, F. Marchal)
- **2004-2018:** applications, calibration, case studies, interface with VISUM
- **2018-2021:** Web interface for METROPOLIS1 [Python Django] (L. Javaudin)
- **2021-:** METROPOLIS2 [Rust] (A. de Palma, L. Javaudin), external modules [Python]



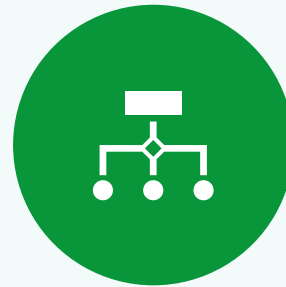


METROPOLIS in a nutshell

Demand side: Discrete-choice models with agents



Agents are making travel decisions (mode of transportation, departure time, route)



Decision-making process based on **utility maximization** using **discrete-choice models** (e.g., Multinomial Logit, Continuous Logit)



Utility depends on value of travel time, schedule delays, fuel cost, etc.



Computation of **individual surplus** for policy evaluation

Demand side: Input data

List of agents with:

- List of **trips** to perform (with origin and destination, desired departure / arrival time)
- **Modes** available (road mode or constant-travel-time mode)
- **Preference parameters** (e.g., value of travel time)
- **Choice models** (how mode / departure time is chosen)



Supply side: mesoscopic & dynamic simulations



Road trips take place on a **road network** (directed graph)



Vehicles are simulated as autonomous entities



Mesoscopic: Congestion is aggregated at the road-segment level



Dynamic: All events are simulated in continuous time



Congestion appears from **bottlenecks**, with queue propagation (**spillback**)

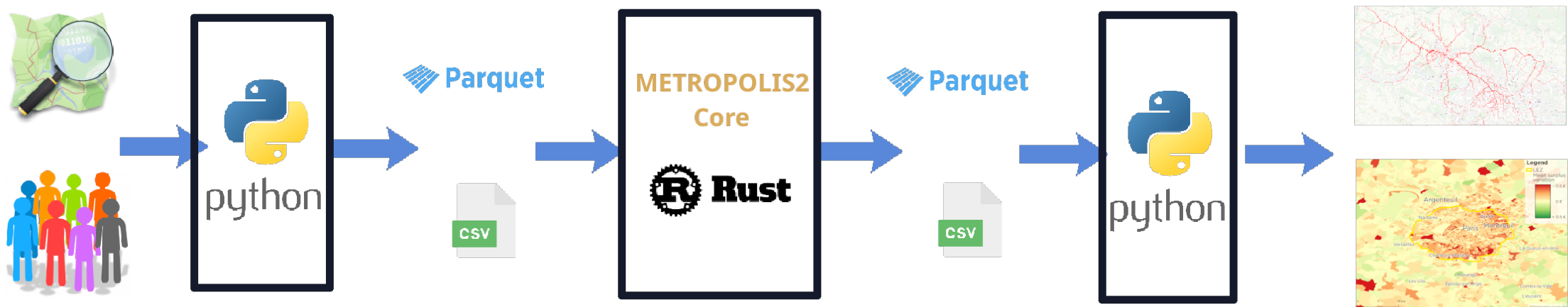
Supply side: Input data

List of **road segments** defining a directed graph with:

- Source / target node
- Length
- Number of lanes
- Free-flow speed
- Bottleneck capacity

List of **vehicle types** with:

- Headway length
- Passenger car equivalent
- Speed limit
- Road restrictions



Complete workflow

1. Python scripts are processing raw data to create the simulator's input data
2. The simulator reads input data and writes output data in Parquet or CSV format
3. Python scripts are processing the simulator's output data to create output tables / graphs / maps

Output data



Aggregate results (e.g., surplus, inequality measures, travel time, congestion, mode shares, vehicle kilometers)



Individual-level travel decisions (mode, departure time, route) and **output** (surplus, travel time, arrival time)



Road-level output (flows, travel-time function, local and global emissions)



METROPOLIS1: Past achievements

Bibliography on METROPOLIS

1. de Palma, A. & F. Marchal (1996), METROPOLIS: un outil de simulation du trafic urbain, *Revue Transports*, 378, 304-315.
2. de Palma, A. & R. Rochat (1997), Impact of Adverse Weather Conditions on Travel Decisions: Experience from a Behavioral Survey in Geneva, *International Journal of Transport Economics*, XXIV(2), 307-325.
3. Khattak, A. & A. de Palma (1997), Impact of Adverse Weather Conditions on the Propensity to Change Travel Decisions: A Survey of Brussels Commuters, *Transportation Research A*, 31(3), 181-203.
4. de Palma, A., F. Marchal & Yu. Nesterov (1997), METROPOLIS: Modular System for Dynamic Traffic Simulation, *Journal of the Transportation Research Board*, 1607, 178-184.
5. de Palma, A. & O. Sanchez (1998), Bilans socio-économiques des infrastructures de transport : pertinence des méthodes d'évaluation en Ile-de-France, *METROPOLIS*, 106-107, *Évaluer et décider dans les transports*, 51-56.
6. de Palma, A. & F. Marchal (1998), METROPOLIS - A Dynamic Simulation Model Designed for ATIS Applications, in *Traffic and Transportation Studies, Proceedings of ICTTS' 98*, Z. Yang K. Wang and B. Mao, (eds.), American Society of Civil Engineers, 770-781.

Bibliography on METROPOLIS

7. de Palma A. & F. Marchal (1998), METROPOLIS: From W. Vickrey to Large Scale Dynamic Traffic Models, in *PTRC Proceedings, Transportation Planning Methods*, I, 211-224.
8. de Palma, A. & F. Marchal (2002), Real Cases Applications of the Fully Dynamic METROPOLIS Tool-Box: an Advocacy for Global Large-scale Mesoscopic Transportation Systems, *Network and Spatial Economics*, Regional and Transportation, special issue on Micro-simulations, 347-369.
9. de Palma, A., M. Kilani & R. Lindsey (2003), Congestion Pricing on Urban Road Networks Using the Dynamic Traffic Simulator METROPOLIS, B.P.Y. Loo and S.W.K. Lam (eds.), *Proceedings of the 8th Conference of Hong Kong Society for Transportation Studies*, December 13, 2003, 462-471.
10. de Palma, A., M. Kilani & R. Lindsey (2005), Congestion Pricing on a Road Network: A Study Using the Dynamic Equilibrium Simulator METROPOLIS, *Transportation Research, A*, 39, 588-611.
11. Saifuzzaman, M. L. Engelson, I. Kristoffersson & A. de Palma (2016), Stockholm congestion charging: an assessment with METROPOLIS and SILVESTER, *Transportation Planning and Technology*, 39(7), 653-694.
12. Vosough; S, A. de Palma & R. Lindsey (2022). Pricing vehicle emissions and congestion externalities using a dynamic traffic network simulator, *Transportation Research A*, 161, 1-24.



METROPOLIS2: Under the hood

Input: Synthetic population

- **Fictitious population** of households and persons generated to be representative of the actual population
- **Combination of data:** census, travel survey, firm & building database
- **Example** of household:
 - a. ID: 19
 - b. Number of cars: 1
 - c. Number of bicycles: 3
 - d. Monthly income: 11 865 €

Input: Synthetic
population

Persons

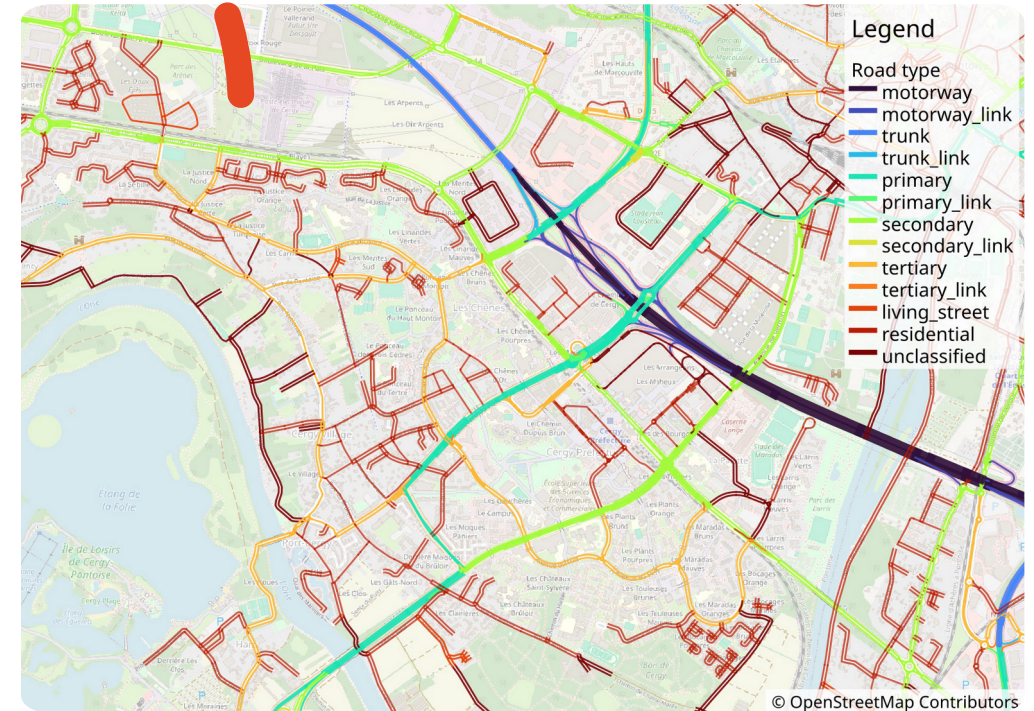
Id	Age	Sex	Socio-professional class
34	51	Man	Employee
35	48	Woman	Employee
36	17	Man	No professional activity
37	13	Woman	No professional activity

Trips

Id	Person id	Departure time	Arrival time	Preceding purpose	Following purpose
133	34	07:52:56	08:41:56	Home	Work
134	34	19:01:56	19:41:56	Work	Home
135	35	07:26:03	07:46:03	Home	Work
136	35	17:56:03	18:26:03	Work	Home
137	36	09:49:46	09:54:46	Home	Education
140	36	16:49:46	17:09:46	Education	Home
141	36	17:49:46	18:04:46	Home	Leisure
142	36	19:04:46	19:19:46	Leisure	Home
143	37	08:03:12	08:08:12	Home	Education
144	37	16:31:12	16:34:12	Education	Home

Input: OpenStreetMap & GTFS

- Road network data can be imported from **OpenStreetMap**
- Public-transit schedule can be read from **GTFS data**
- **OpenTripPlanner** is used to compute public-transit travel times



Calibration

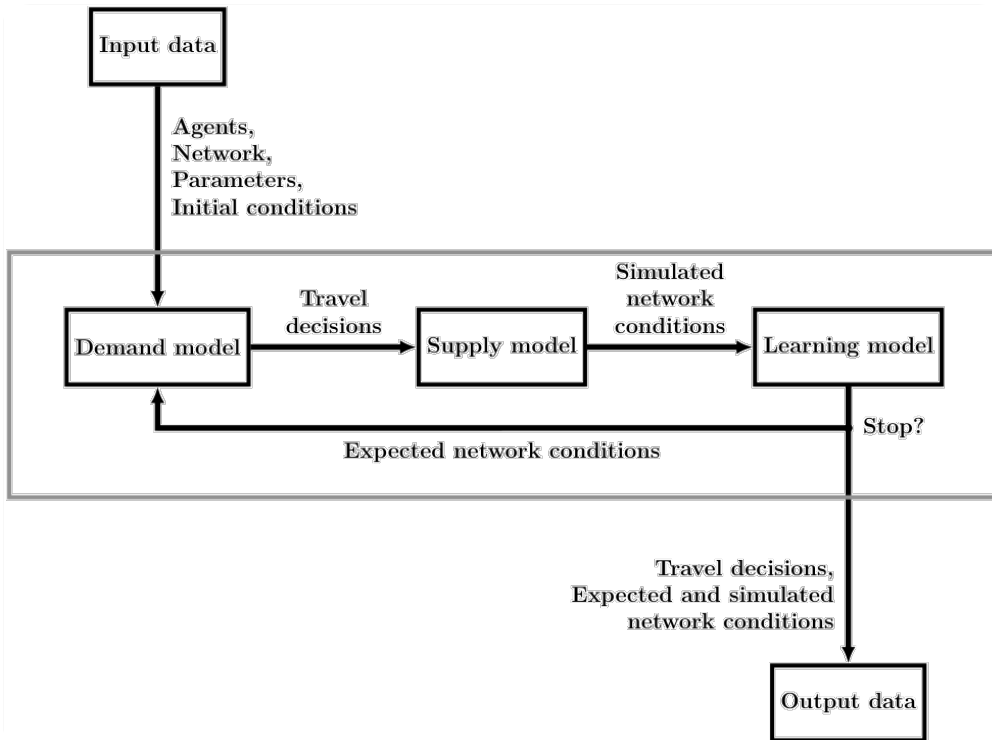


Many simulations parameters can be **automatically calibrated** to replicate **TomTom** and **travel survey** data

Step	Calibrated parameters	Target values	Target source	Methodology
1	Road constant penalties and free-flow speed	Free-flow travel times	TomTom API	Lasso regression
2	Road capacities	Time-dependent congested travel times	TomTom API	OLS regression
3	Schedule-delay penalties by purpose	Distribution of departure times by cluster	Travel survey	Bayesian Optimization with Gaussian Process
4	Utility parameters by socio-demographic characteristics	Mode shares by cluster	Travel survey	Random Forest regression

Iterative process

An iterative process is required to update agents' decisions until their congestion expectations match the simulated congestion

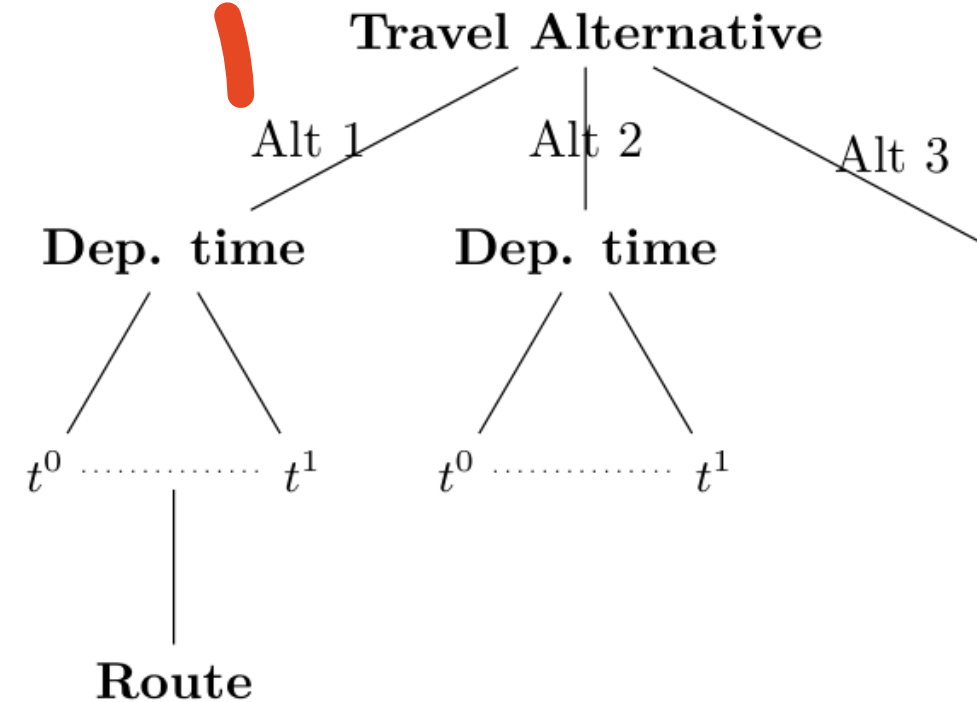


Demand model: Trips & Trip chaining

- Trip options:
 - **Road trips** (e.g., car driver, car passenger, truck, motorcycle)
 - **Constant-travel-time trips** (e.g., walking, bicycle, reliable public transit)
 - Public-transit trips (*work-in-progress*)
- **Trip chaining:** each agent perform an ordered sequence of an arbitrary number of trips
- **Intermodal trips:** Trip chains can combine different modes

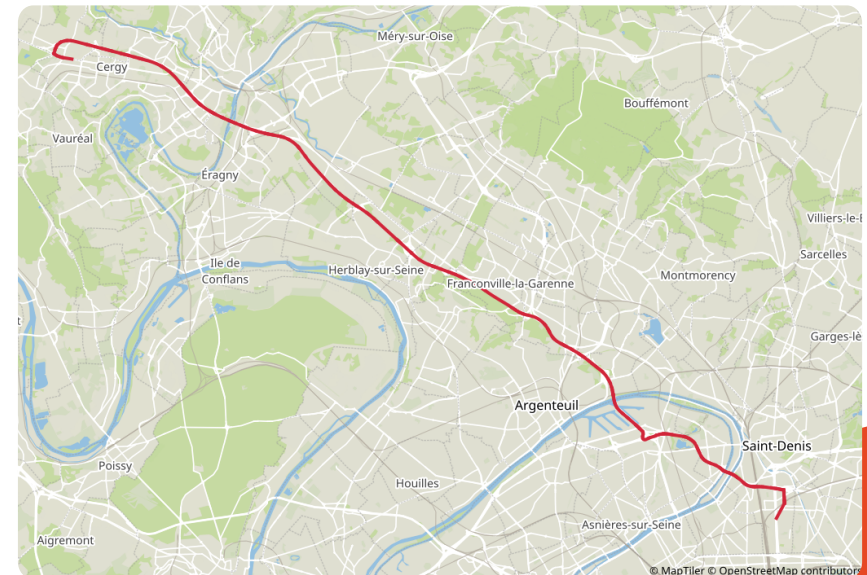
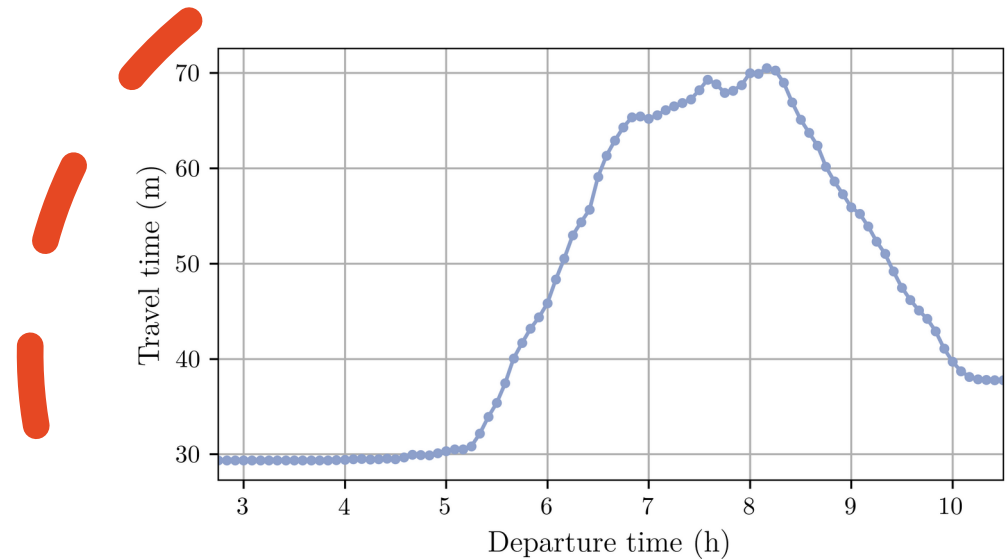
Demand model: Individual decision process

- Level 1: Agents choose an **alternative** (combination of mode and trip chain); *Multinomial Logit, Nested Logit, Deterministic model, etc.*
- Level 2: Agents choose a **departure time** for the first trip; *Continuous Logit, Multinomial Logit, exogenous.*
- Level 3: Agents choose the fastest **route** (for road trips)



Demand model: Routing

- **Routing algorithms** are required for the departure-time choice (travel time as a function of departure time for an origin-destination pair) and the route choice (fastest path from origin to destination given the departure time)
- METROPOLIS2 employs **time-dependent Contraction Hierarchies**, allowing for thousands of requests per second on large-scale networks



Supply model: Congestion modeling

- **Bottleneck** congestion: inflow of vehicles is limited by road capacity
- Queue propagation (**spillback**): inflow of vehicle is limited by road storage capacity (# vehicles that can spread on road length)
- **Speed-density functions**: vehicle speed is a function of road vehicle density

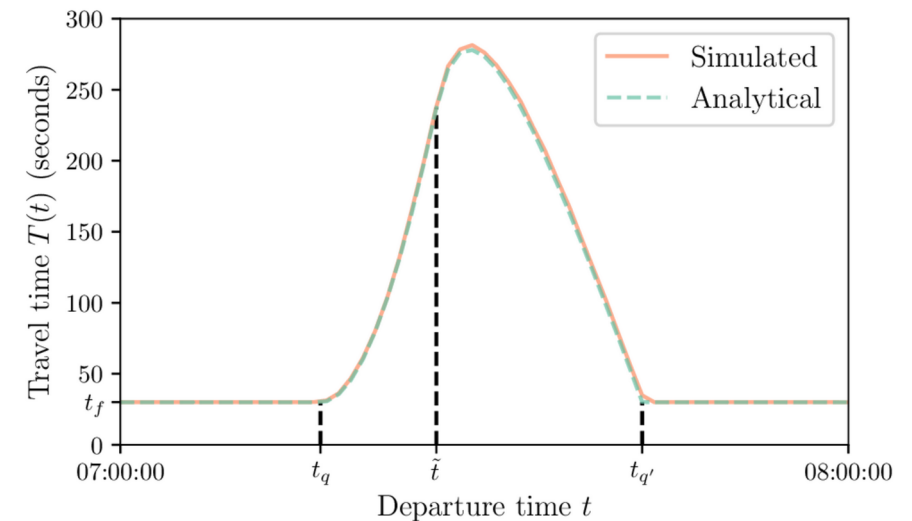
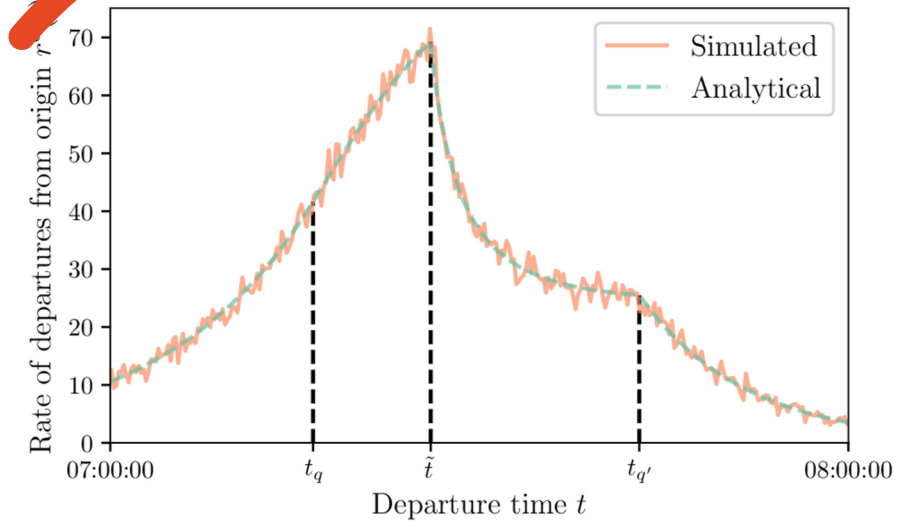


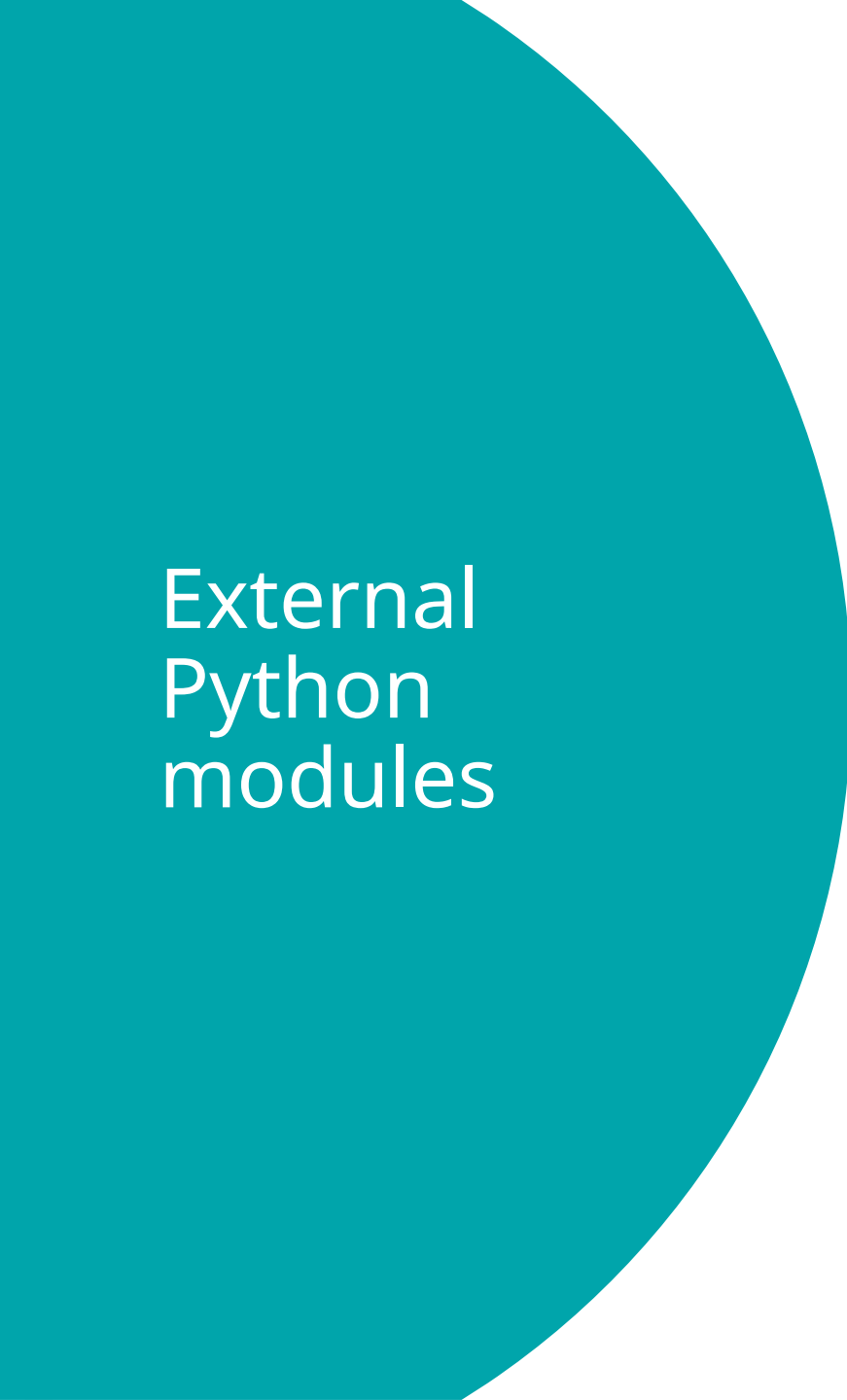
Running time

- Simulator running time depends on:
 - a. # agents
 - b. # alternatives
 - c. # trips
 - d. # road network size (# segments)
 - e. # unique origins / destinations
 - f. # OD pairs
- Exemple: runnning time 30 min. per iteration with 4M agents, 9M trips, 5 modes, 337k road segments, 117k unique origins / destinations, 4M unique OD pairs

Analytical validation

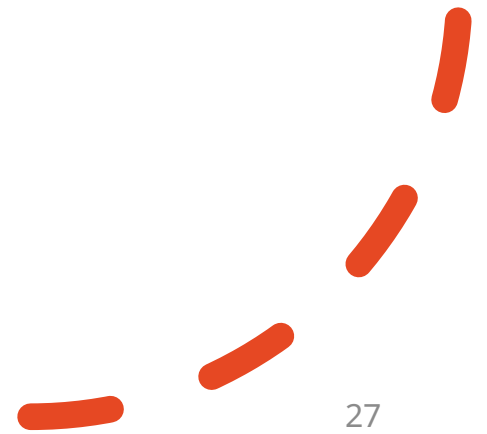
- Analytical results from de Palma et al. (1983) are well replicated with METROPOLIS2
- Model: single road, bottleneck, alpha-beta-gamma preferences, Continuous Logit departure-time choice



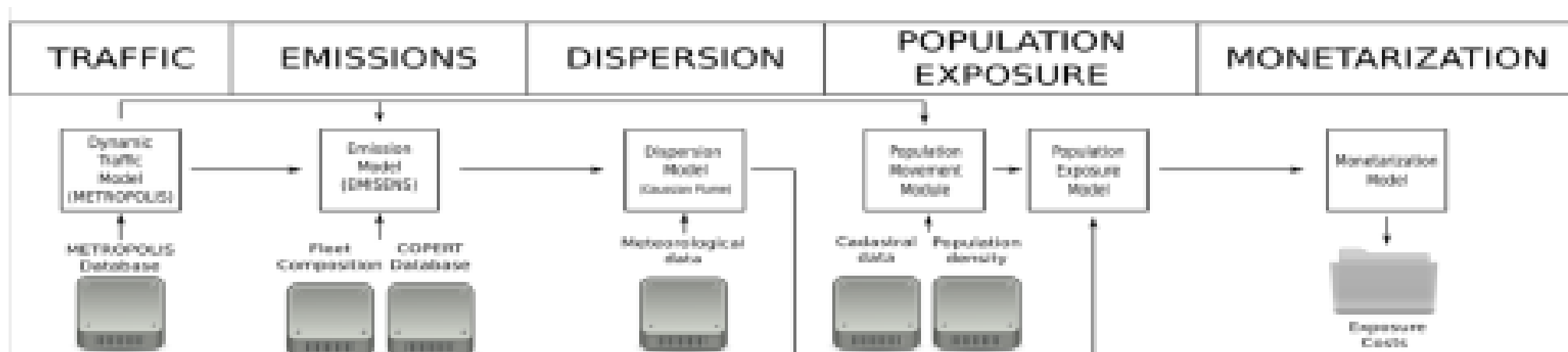
A large teal circle is positioned on the left side of the slide, partially cut off by the edge.

External Python modules

- METRO-MATCH
- METRO-TRACE (see next slides)
- METRO-NOISE (under construction)

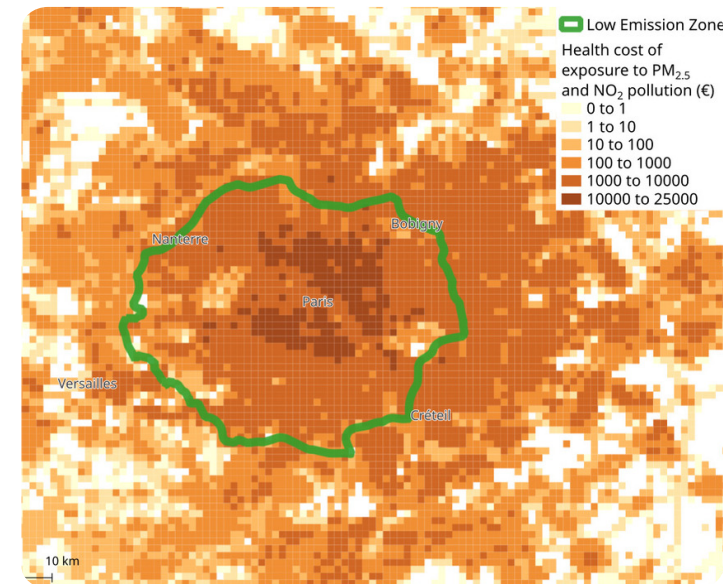
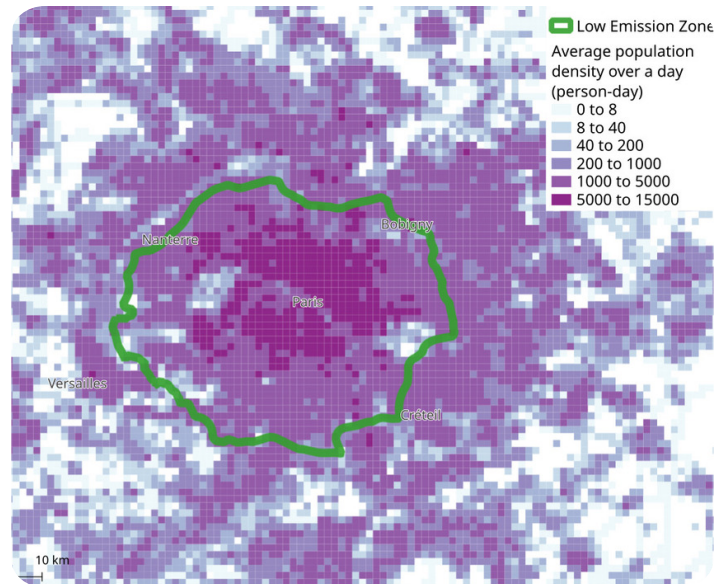
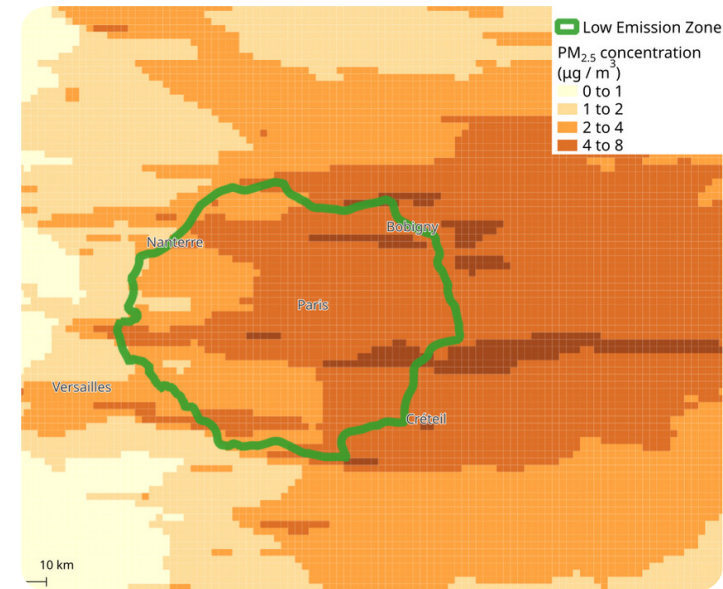
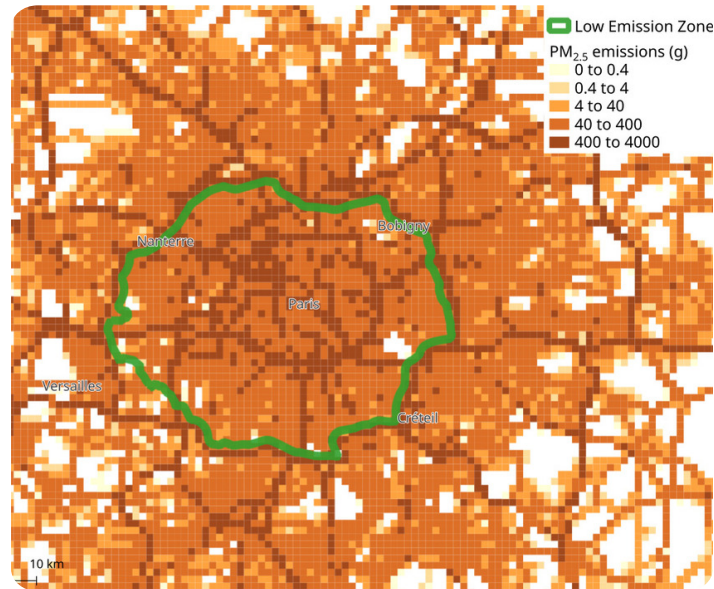


External modules: METRO-TRACE



External modules: METRO-TRACE

- Compute **road traffic emissions** given vehicle characteristics and road-level speeds (EMISENS model)
- **Disperse** emissions in the atmosphere on a grid (Gaussian plume dispersion model)
- Compute **population density** over the simulation period on a grid
- Compute **exposure** by combining pollutant concentration and population density
- Monetarize exposure to obtain **health costs**

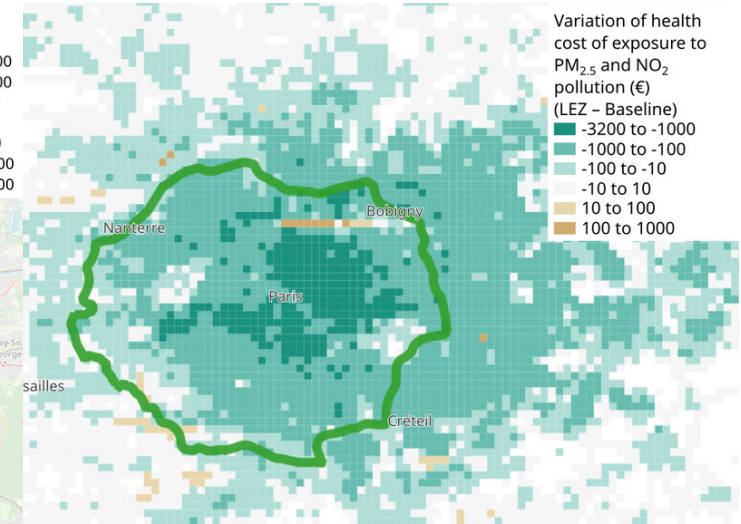
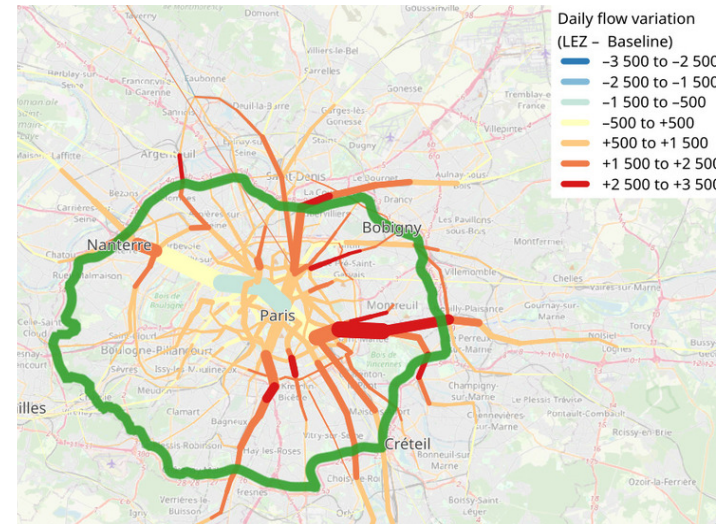
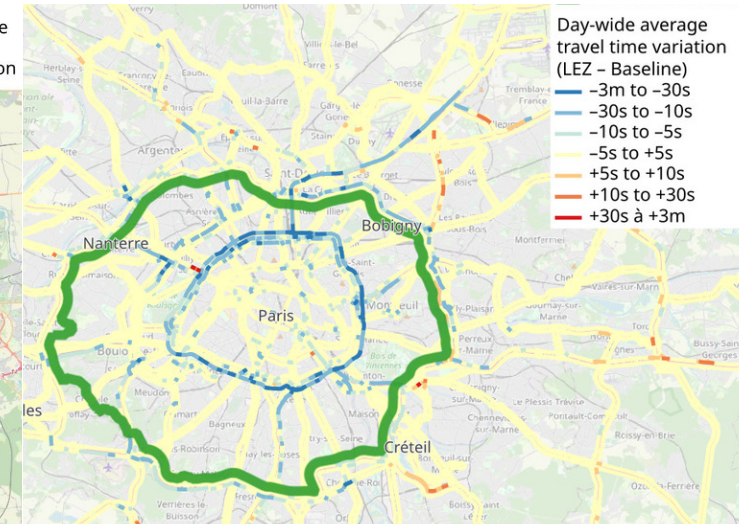
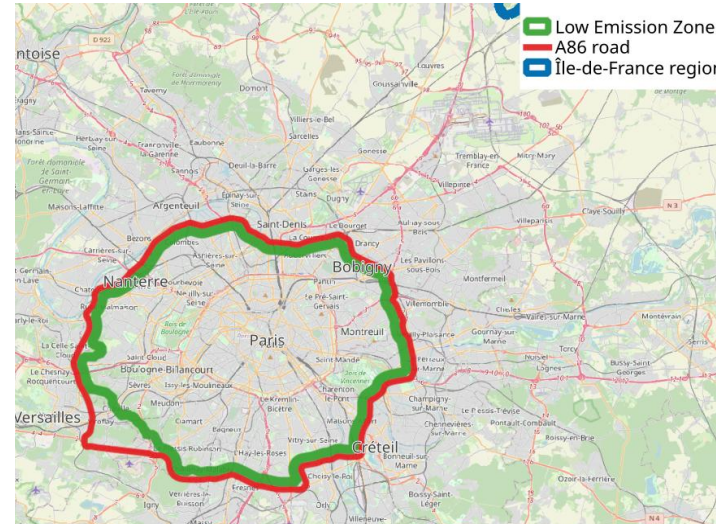




METROPOLIS2: Case studies

Low Emission Zone

- Starting in January 2025, the most polluting vehicles (about 20% of the vehicle fleet) will be restricted from entering a 367km² area around Paris
- Predicted impacts:
 - Congestion reduction** on the *Boulevard Périphérique* (top right)
 - Increase in passenger flows** for lines connecting Paris to the suburbs (bottom left)
 - Decreased health cost** from exposure to PM_{2.5} and NO₂ emissions around Paris (bottom right)



Filtering Process and Parameter Values

Matching: Filtering Process

40 % (82,000) car travelers are selected randomly to participate in ridesharing.

① Distance filter:

- Origins within 5 km.
- Destinations within 5 km.

② Time filter:

- Arrival times within 20 minutes ($t_a \pm 20$).

Parameter Values

- $\alpha = 10 \text{ €/hr}$, $\beta = 5 \text{ €/hr}$, $\gamma = 5 \text{ €/hr}$
- Vehicle mileage = 15 km/litre
- Fuel cost = 1.8 €/litre
- Emission cost = 200 €/tonne CO_2
- Walking speed = 4 km/hr
- Walking value of time = 10 €/hr

Carpooling (with Samarth Ghoslya)

Change in results with and without ridesharing scenario for morning peak:

	Before ridesharing	After ridesharing	Change
Total cars	2.06 million	1.86 million	−195 420 (9.45%)
Congestion	43.78%	41.02%	−2.76%
Fuel consumption	1.97 million litres	1.86 million litres	−5.48%
CO ₂ emission	625 tonnes	591 tonnes	−34 tonnes
Mean departure time	8:08 a.m.	8:09 a.m.	+1'
Mean arrival time	8:27 a.m.	8:26 a.m.	−1' 3''
Mean travel time	18' 33''	17' 02''	−1' 31''
Total distance	30.70 million km	29.25 million km	−1.45 million km (4.72%)

Speed limits (with Romuald Le Frioux)

Table 2.9: Total air pollutant traffic costs for CO_2 , NO_2 , O_3 , and $PM_{2.5}$ computed over Île-de-France from 3 to 10 a.m, while adding restricting the speed limitations by 20 km.h^{-1} with a minimum of 70 km.h^{-1} , while allowing or not for modal changes in the model

Pollutant	Total (k€)	Per driver (€)	Without mode choice		
			Per inhabitants (€)	Per kilometer (€)	Per emissions (€. kg^{-1})
CO_2	569 (-0.29%)	0.281 (+3.4%)	0.048 (-0.29%)	0.018 (+0.27%)	0.1 (0.0%)
NO_2	781 (-0.44%)	0.386 (+3.25%)	0.067 (-0.44%)	0.024 (+0.11%)	70.58 (-0.08%)
O_3	-639 (+0.09%)	-0.316 (+3.8%)	-0.054 (+0.09%)	-0.02 (+0.65%)	-
$PM_{2.5}$	798 (-1.51%)	0.395 (+2.14%)	0.068 (-1.51%)	0.025 (-0.96%)	891.39 (-0.82%)
Total	1509 (-1.17%)	0.746 (+2.49%)	0.128 (-1.17%)	0.047 (-0.62%)	-
Pollutant	Total (k€)	Per driver (€)	With mode choice		
			Per inhabitants (€)	Per kilometer (€)	Per emission (€)
CO_2	542 (-5.06%)	0.261 (-4.17%)	0.046 (-5.06%)	0.017 (-1.39%)	0.1 (0.0%)
NO_2	751 (-4.29%)	0.362 (-3.4%)	0.064 (-4.29%)	0.024 (-0.59%)	73.96 (+4.71%)
O_3	-626 (-1.92%)	-0.301 (-1%)	-0.053 (-1.92%)	-0.02 (+1.87%)	-
$PM_{2.5}$	797 (-1.58%)	0.384 (-0.66%)	0.068 (-1.58%)	0.025 (+2.22%)	878.82 (-2.22%)
Total	1464 (-4.13%)	0.705 (-3.24%)	0.125 (-4.13%)	0.047 (-0.43%)	-

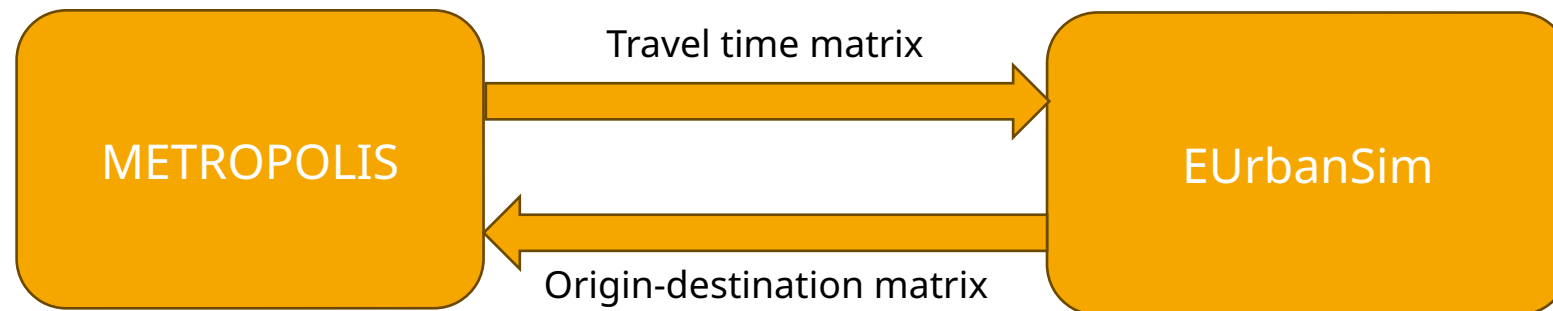


METROPOLIS2: Interfaces with other tools

LUTI (Land Use and Transport Interaction)

LUTI models allow to study the long-term consequences of policies (relocation of households and firms).

Here twin between EUrbanSim and METROPOLIS



A large teal circle is positioned on the left side of the slide, partially cut off by the edge.

Micro- simulators

Micro-simulators $\leftrightarrow$ mesoscopic
models

- Boundary condition for local traffic analysis
 - (a) detailed description of local situations (e.g. intersections); (b) analysis of aggregation and consistency
- 
- A series of four orange dashed line segments are arranged in a curved, upward-sloping pattern in the bottom right corner of the slide.