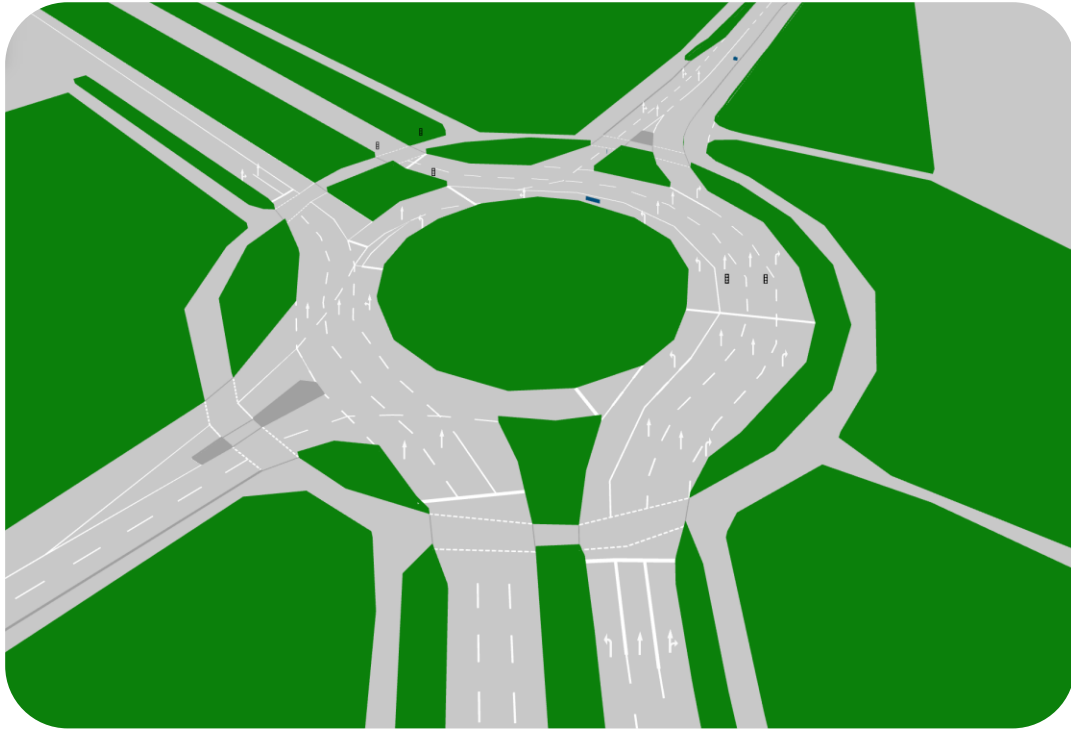


THE LANELET2 MAP FRAMEWORK

FABIAN POGGENHANS

FABIAN.POGGENHANS@MERCEDES-BENZ.COM



LANELET2 OVERVIEW



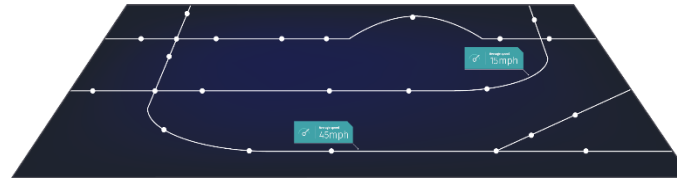
- A framework for working with **lane-level maps** in autonomous driving
- **BSD** license
- Available on [Github](#)
- Written in **C++14** with a **Python** API
- **State-of-the-art codebase**: Densely tested, statically checked, CI-verified and documented
- Part of **ROS 1+2**
- **Widely used**: AutonomouStuff, Autoware, Atlatec, Carma, Huawei, Parkopedia, Tier4, research institutes (Darmstadt, Hannover, Karlsruhe, Munich, Ohio, RWTH, Sydney, Ulm, UTC, Nagoja (Japan), Waterloo (Canada)), >500 Github users
- **Maintained by FZI** (Forschungszentrum Informatik in Karlsruhe), but further development has slowed down quite a bit

CORE DESIGN IDEAS

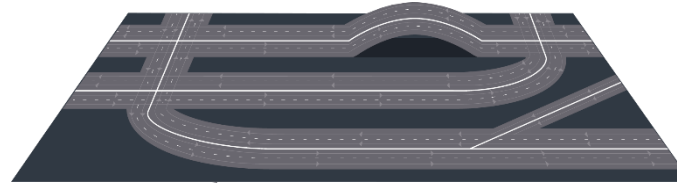


- **Completeness:** Offer a complete (3D) image of the world
- **Correctness:** Make it simple to create a correct map and hard to create an invalid one
- **Separation:** separate data *storage* from world *representation* from *interpretation* though traffic participants
- **Simplicity:** Describe the world with as few primitives as possible
- **Modularity:** Separate core components from utility and extension libraries
- **Flexibility:** Offer extension points for customization and internationalization
- **Compability:** Play nice with commonly used software (Boost, Eigen, ROS, OSM, ...)
- **Versatility:** Usable for e.g. routing, prediction, localization, scene understanding, map validation, emergency situations, simulation, behaviour generation, ...

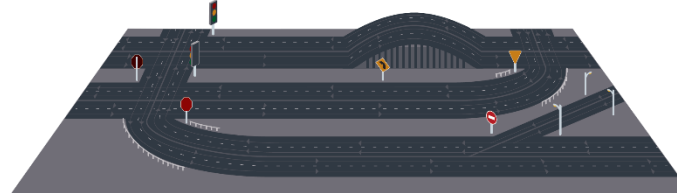
TRADITIONAL MAP APPROACH



Layer 1: Routing



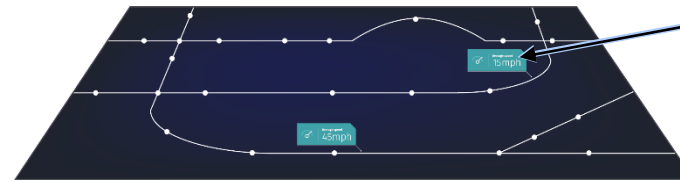
Layer 2: Lane Layer



Layer 3: Localization Layer

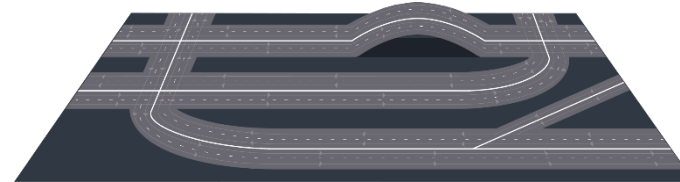
© HERE

TRADITIONAL MAP APPROACH

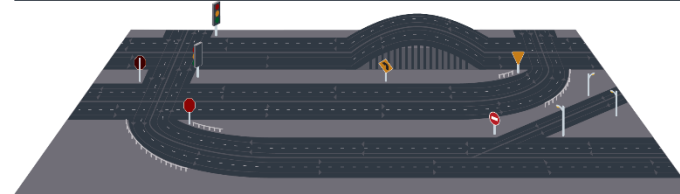


15mph: Where does this information come from?

Layer 1: Routing



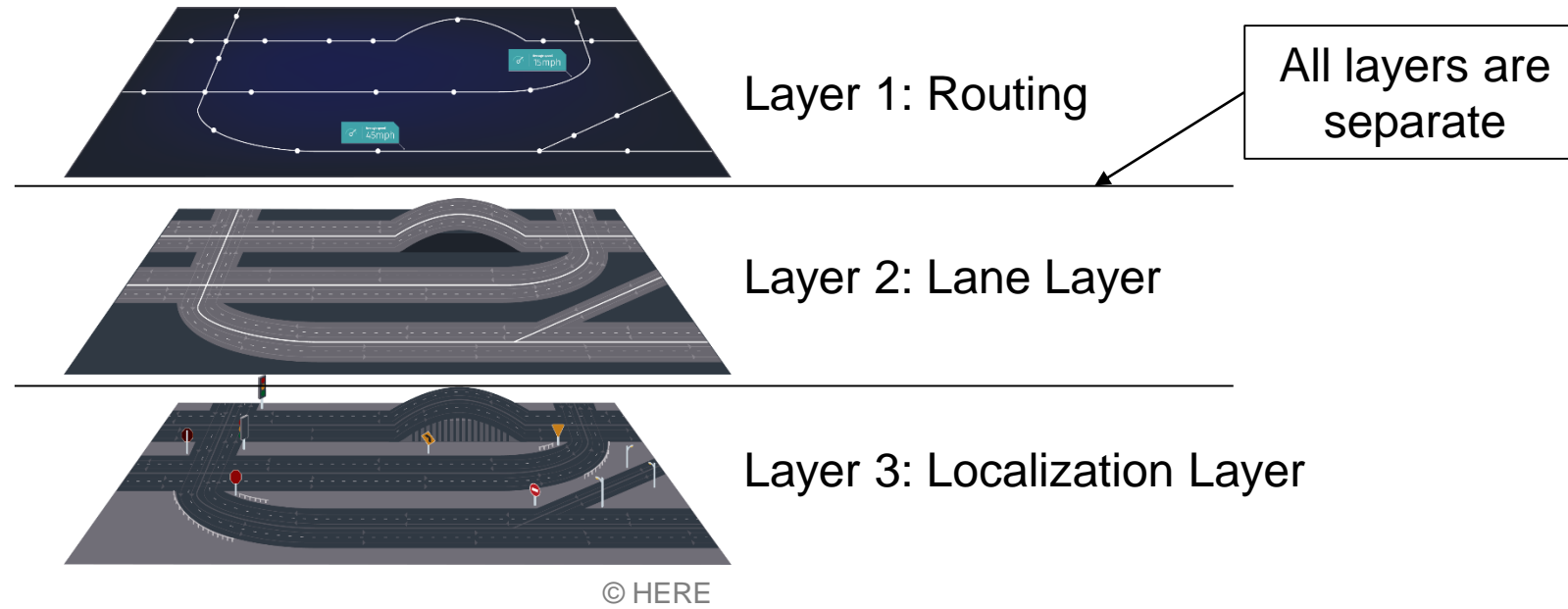
Layer 2: Lane Layer



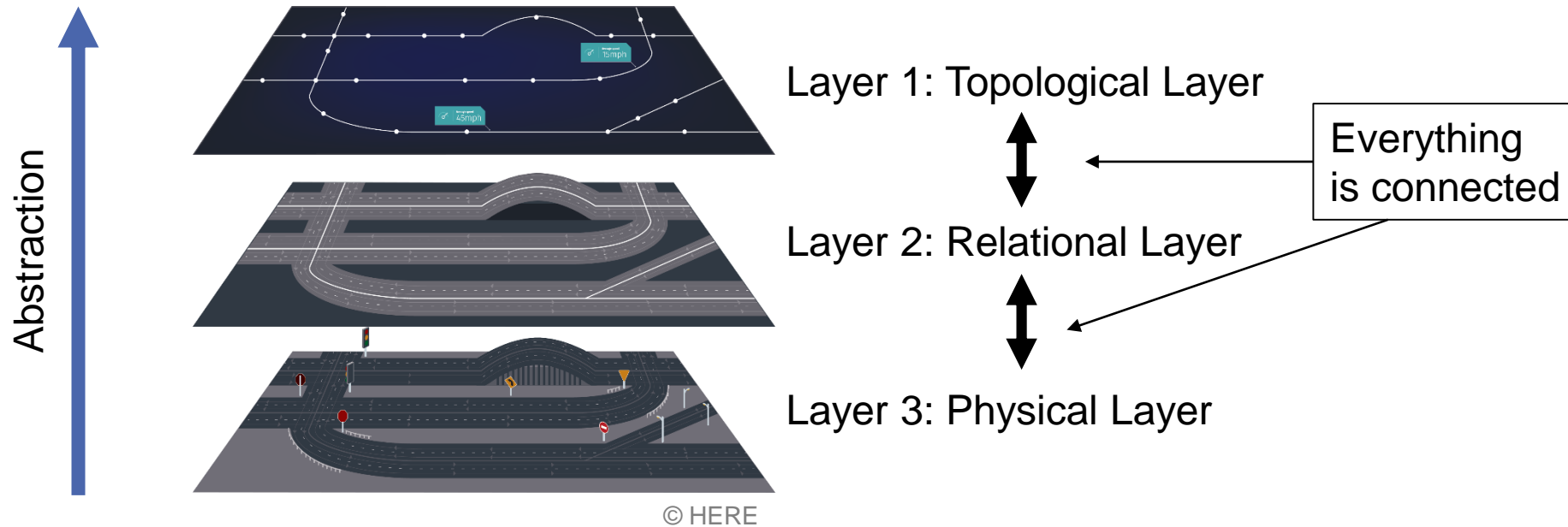
Layer 3: Localization Layer

© HERE

TRADITIONAL MAP APPROACH

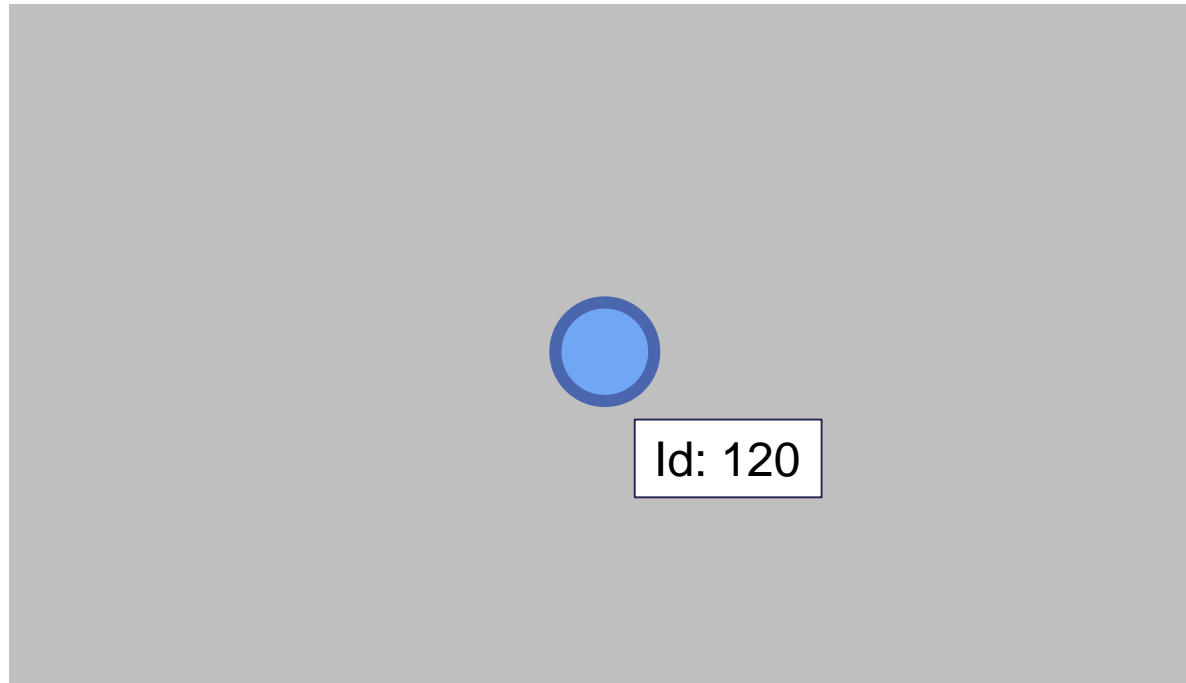


LANELET2 BOTTOM-UP-APPROACH



LANELET2 PRIMITIVE ZOO

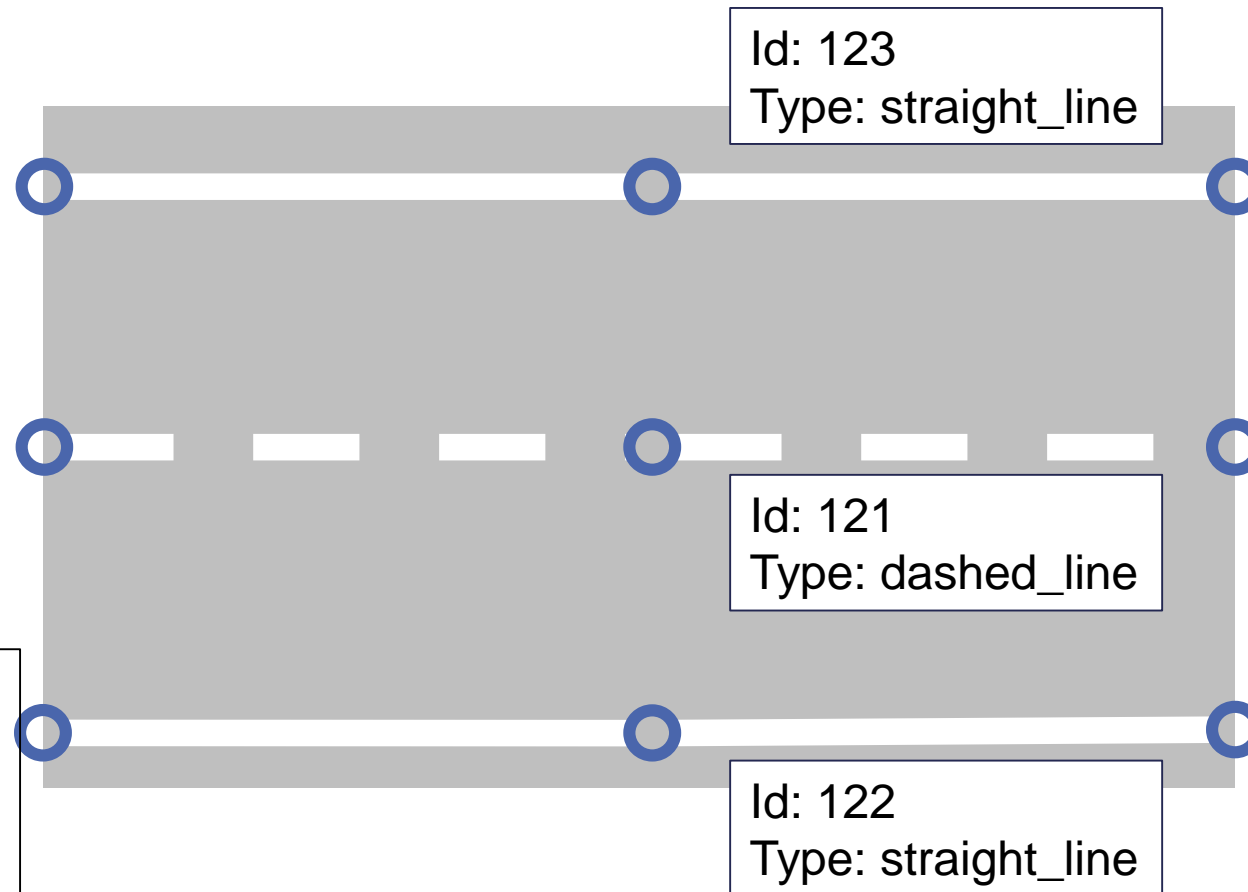
POINT



Points are:

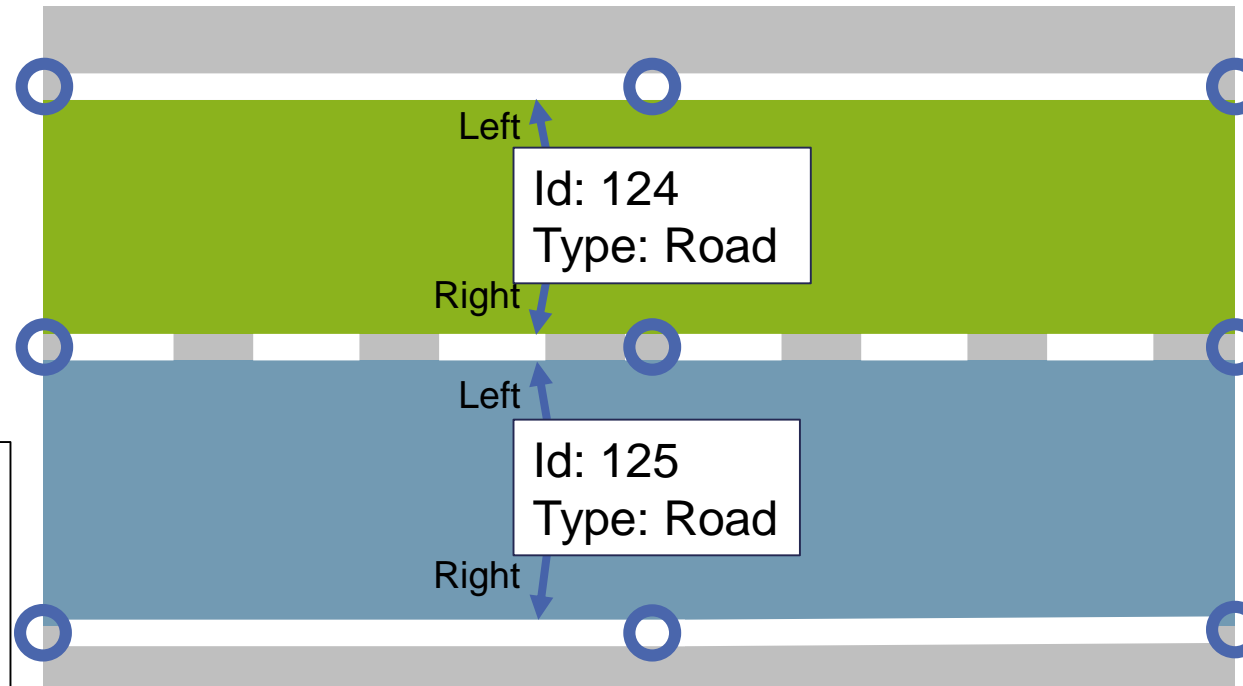
- 3D
- In cartesian space
- Unique

LINSTRING



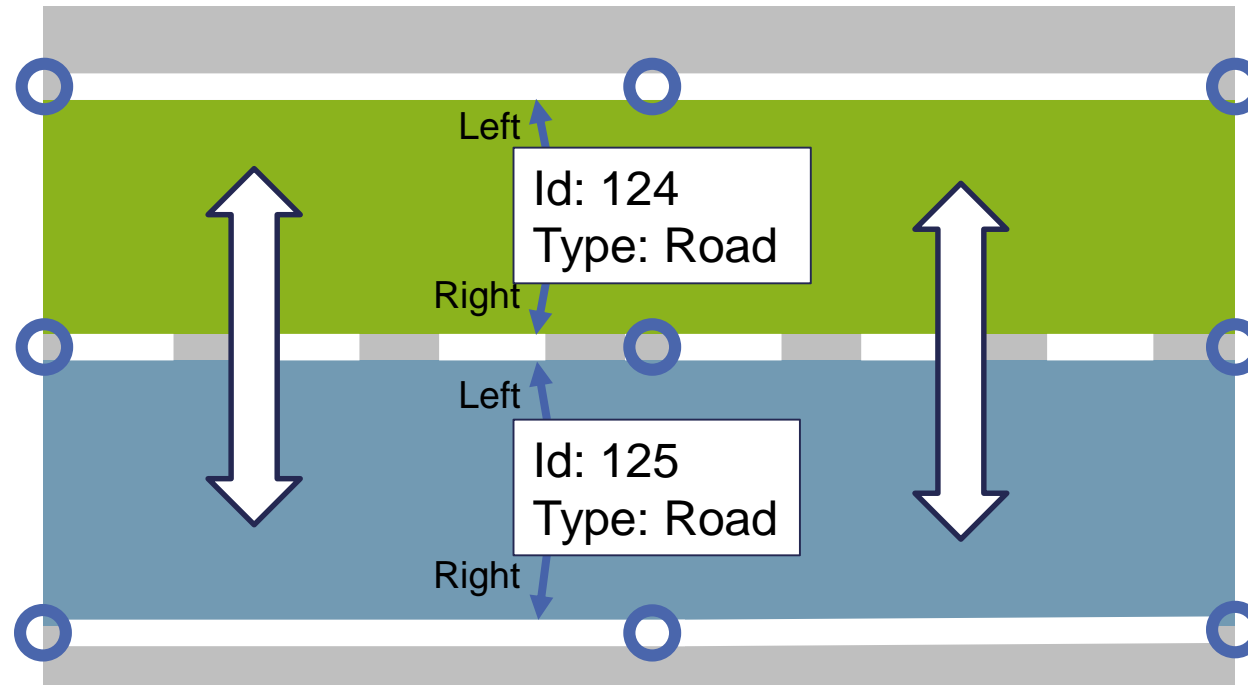
Linestrings are:

- no splines!
- sharing points
- a representation of physical objects

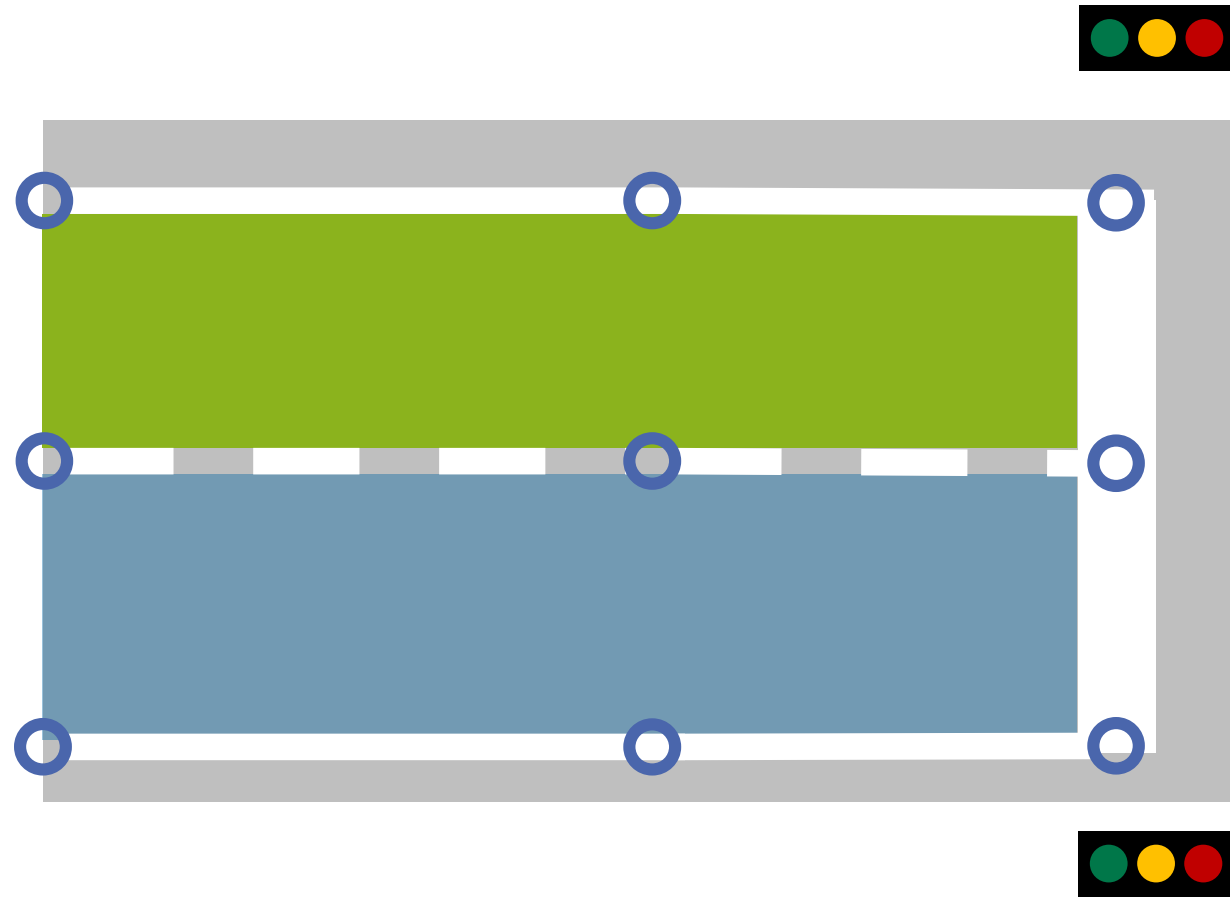


Lanelets are:

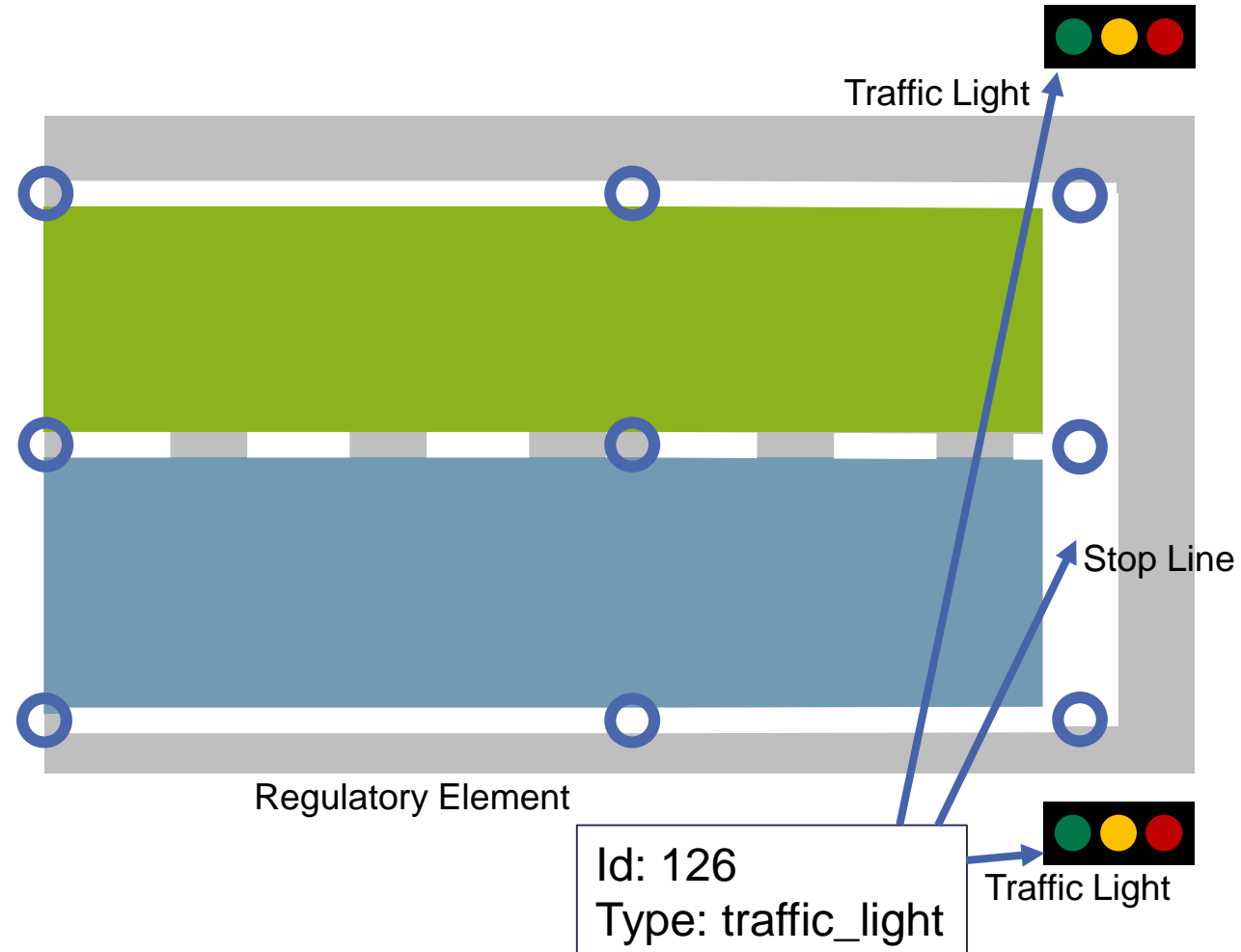
- the heart of Lanelet2
- sharing linestrings
- a part of a road
- also sidewalks, bikelanes, emergency lanes...



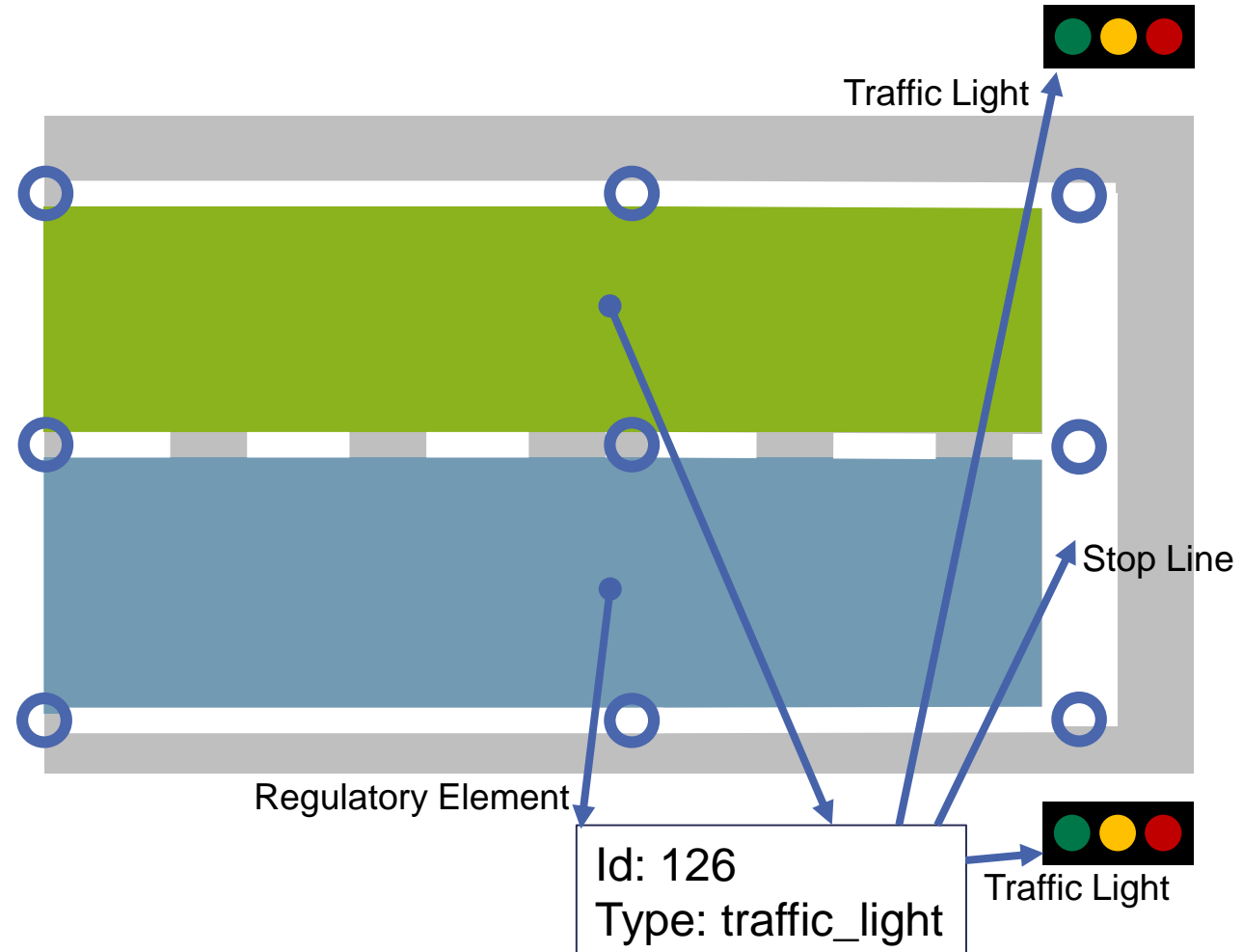
REGULATORY ELEMENT



REGULATORY ELEMENT



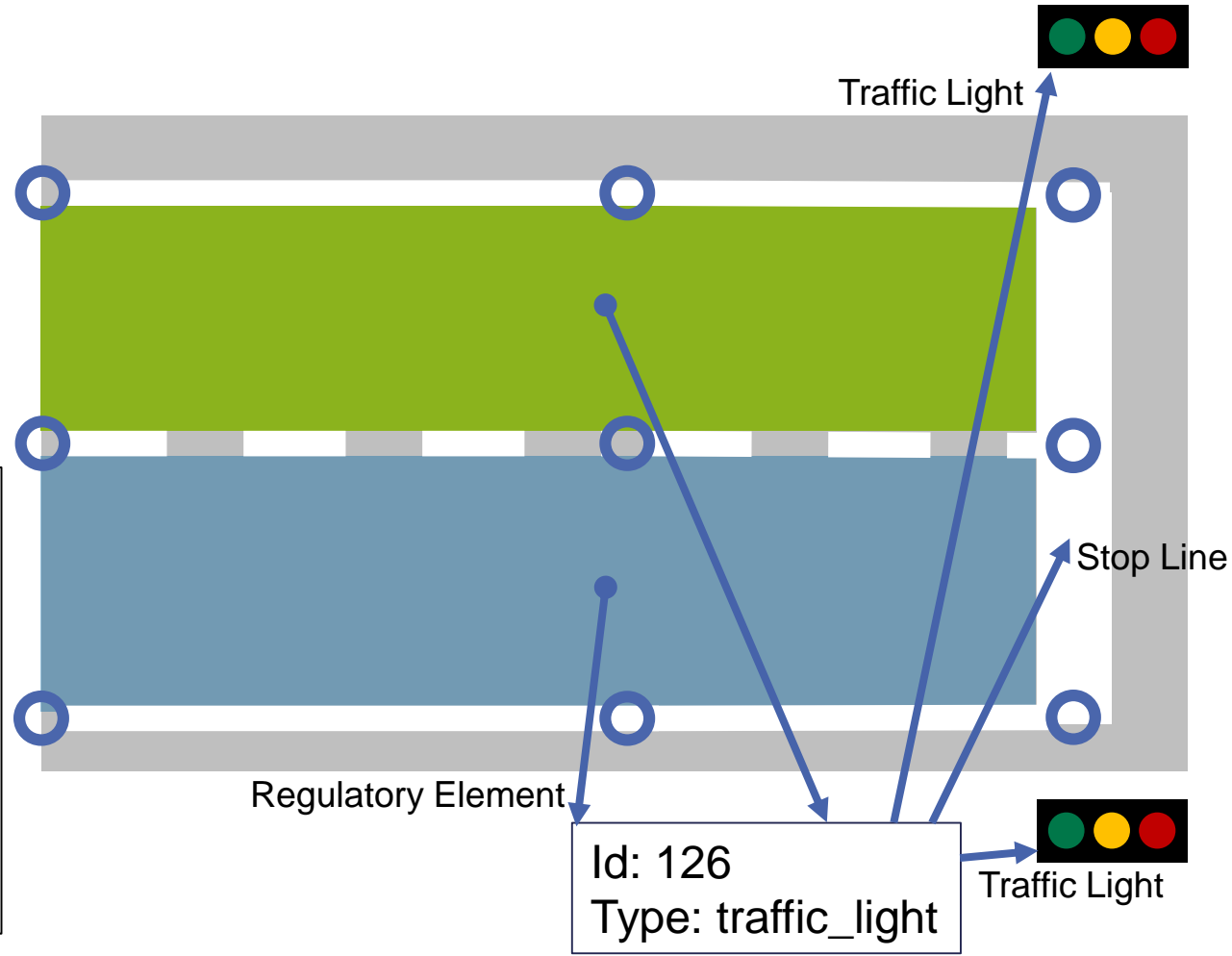
REGULATORY ELEMENT



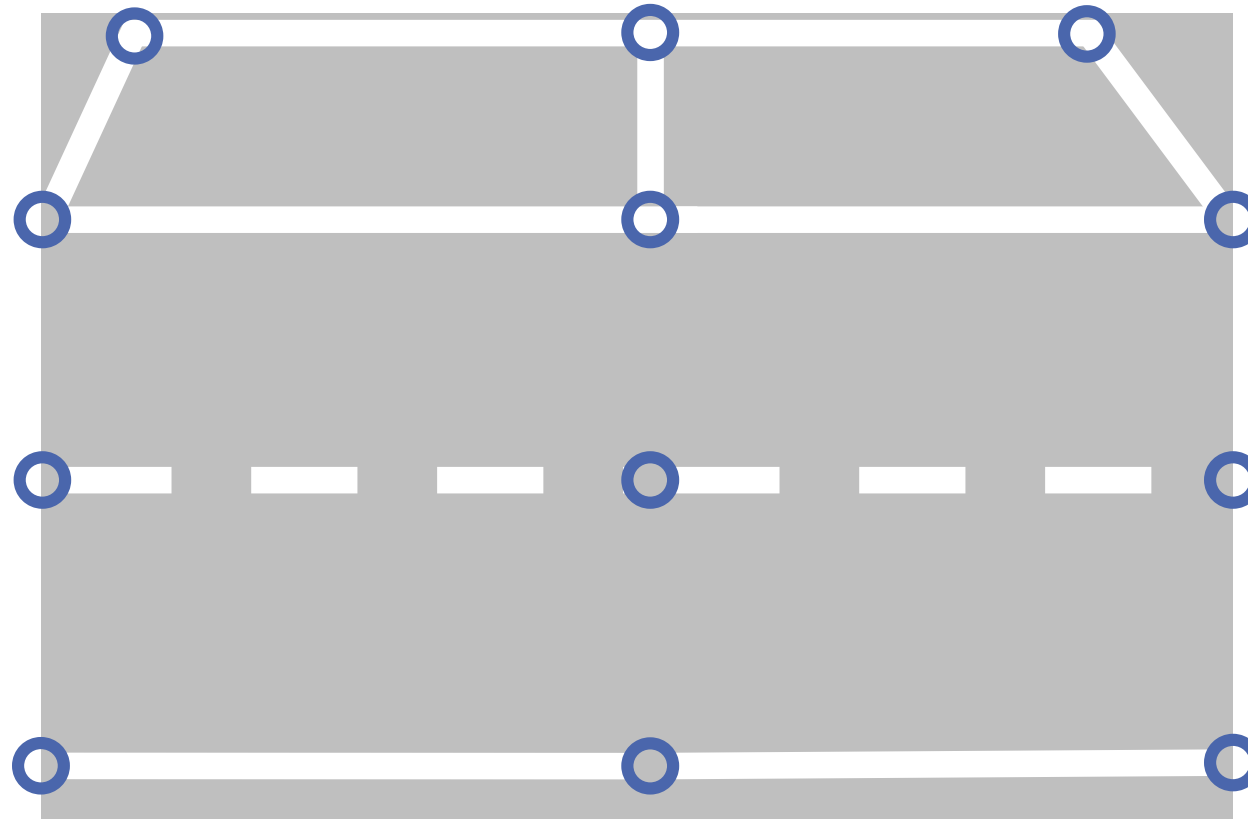
REGULATORY ELEMENT

Regulatory elements are:

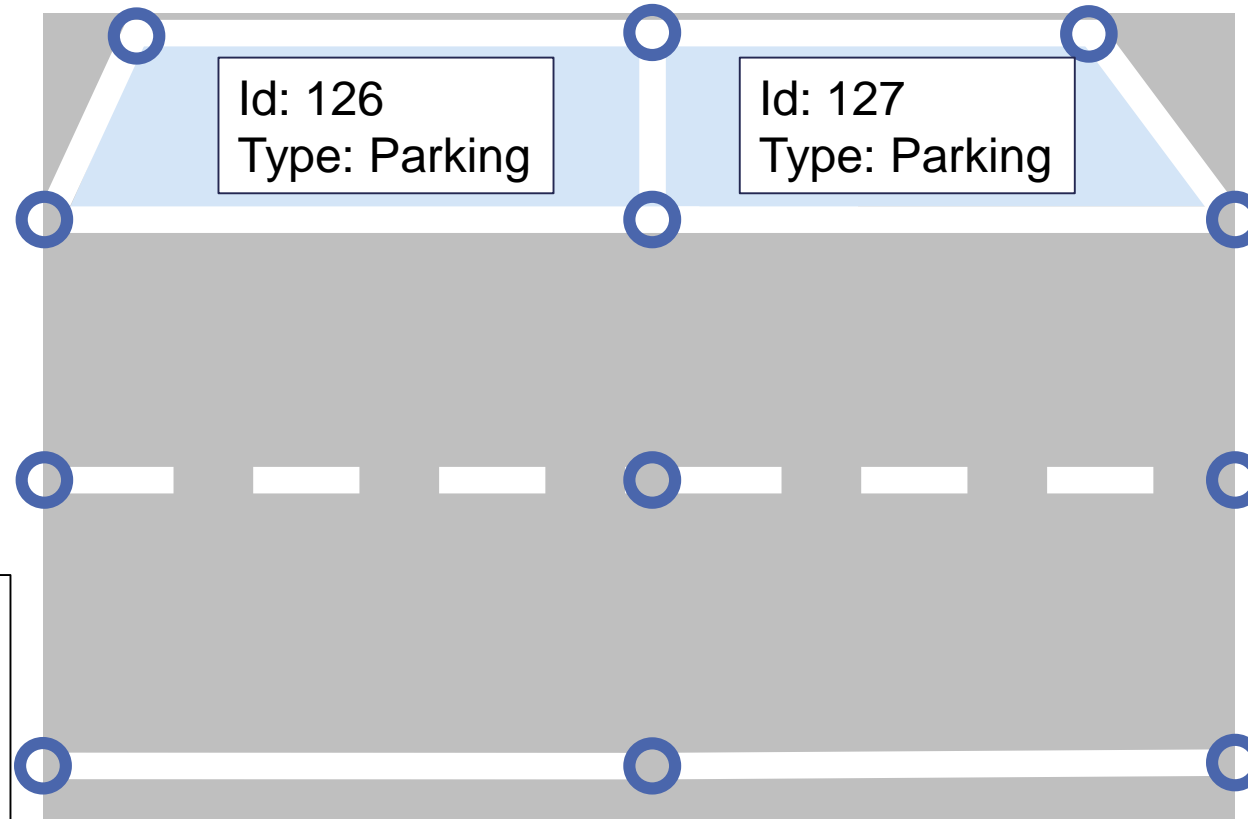
- Used to express relations between physical elements and lanelets
- E.g. traffic lights, speed limits, right of way, overtaking, ...



AREA



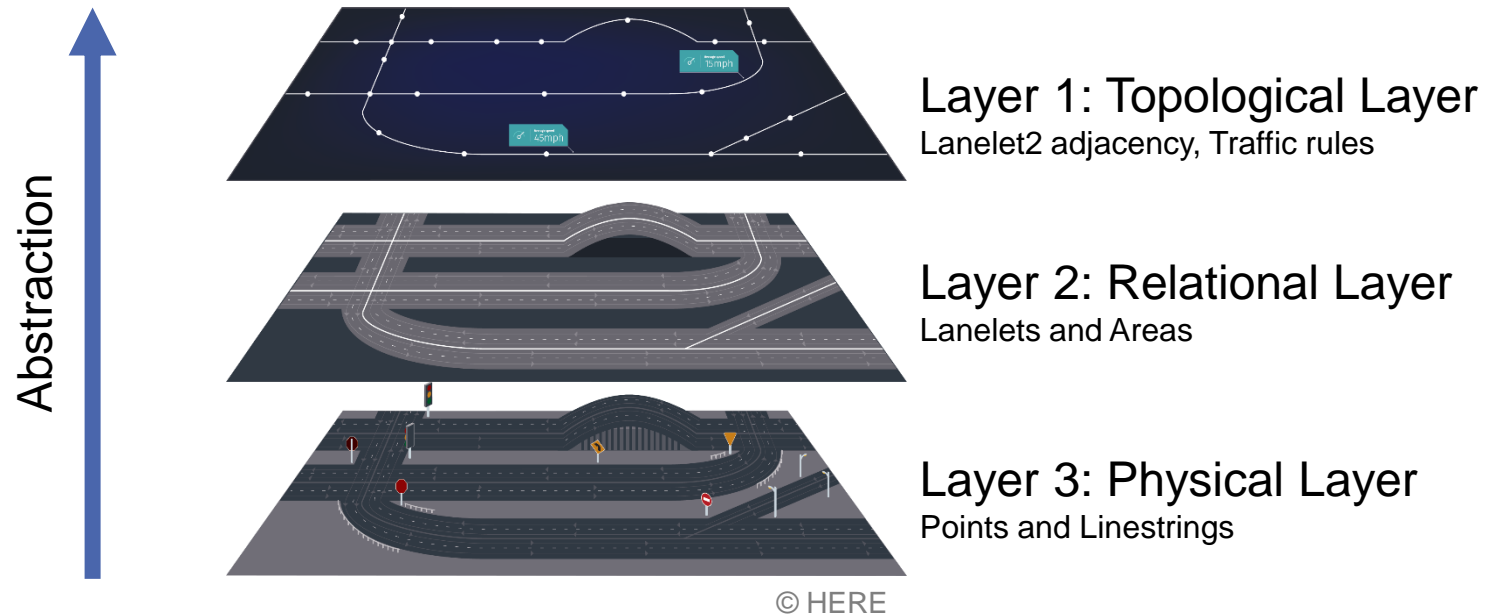
AREA



Areas are:

- For undirected traffic
- E.g. parking, open pedestrian spaces, hatched areas, ...

LANELET2 BOTTOM-UP APPROACH



LANELET2 ARCHITECTURE

ARCHITECTURE

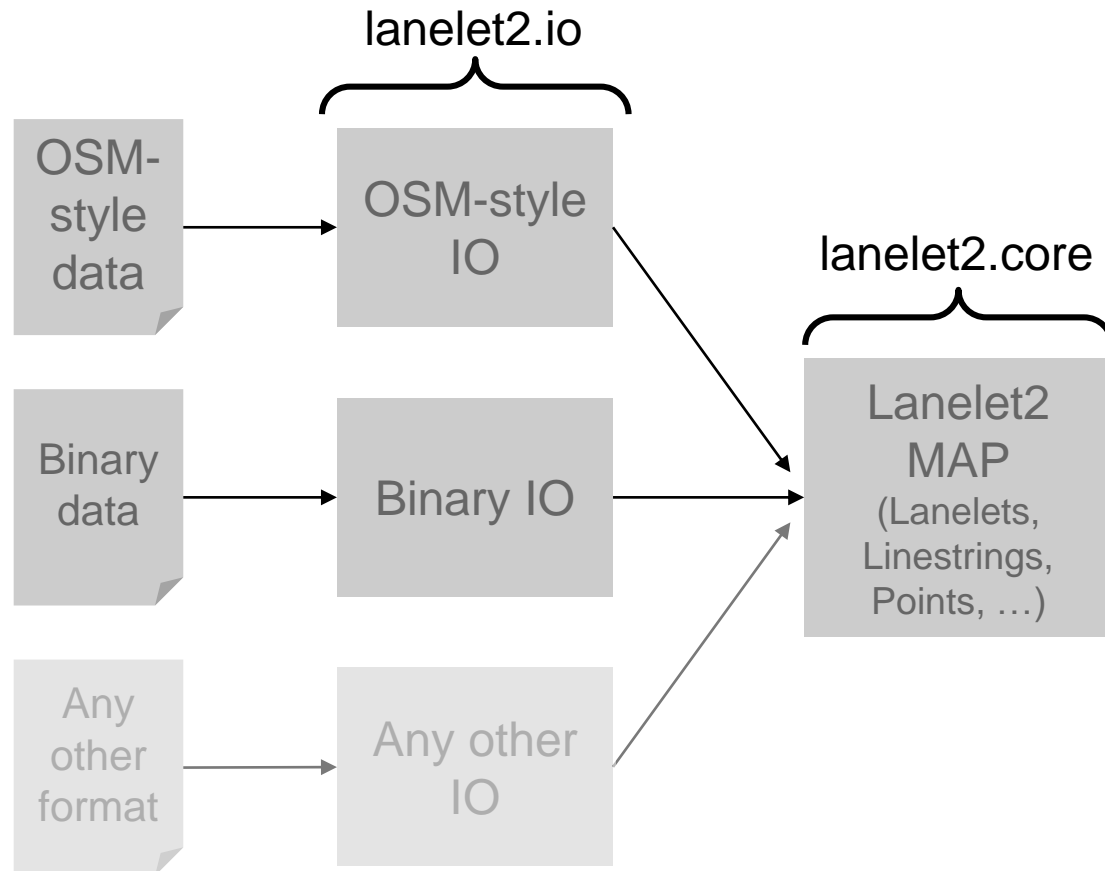


OSM-
style
data

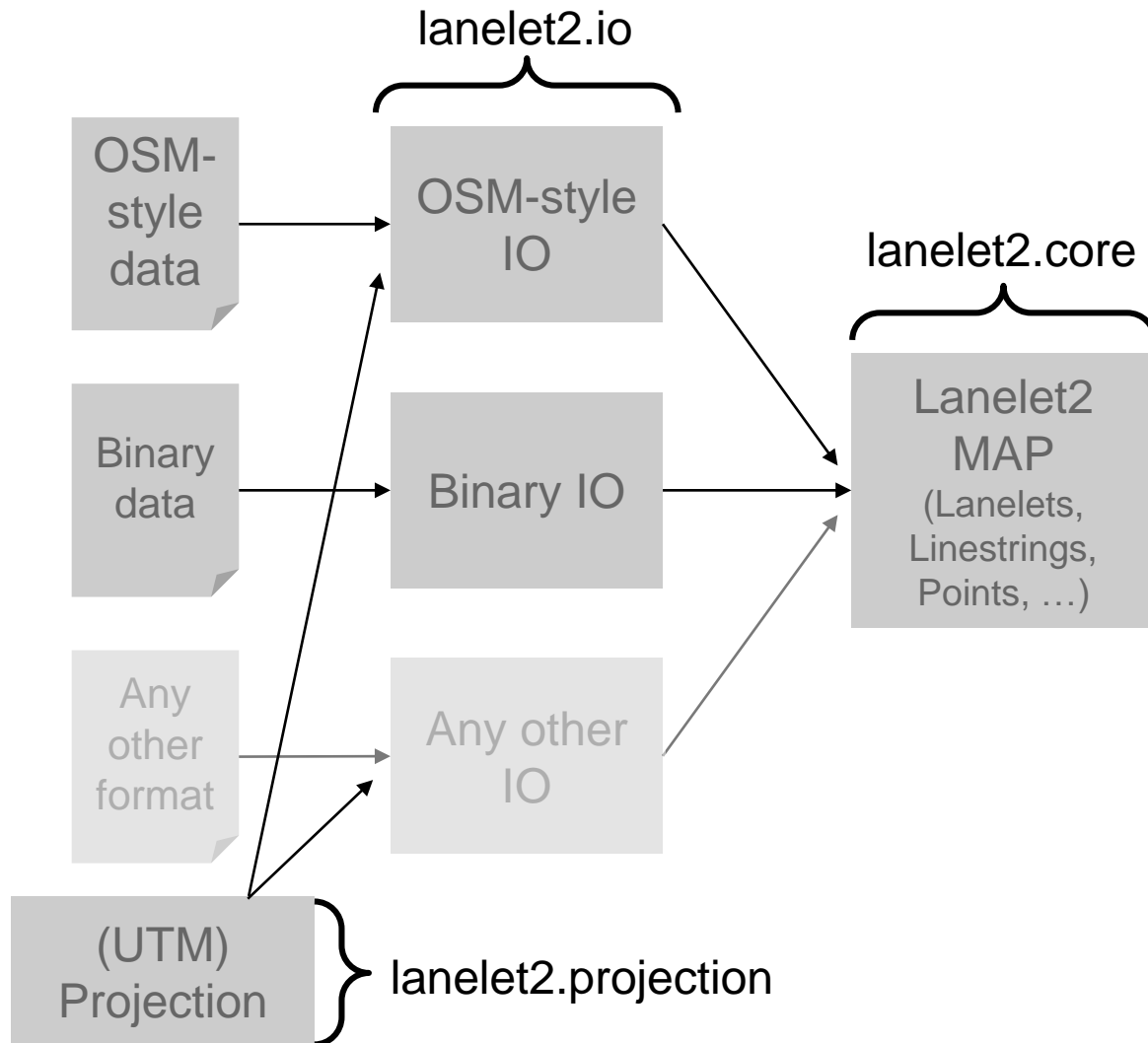
Binary
data

Any
other
format

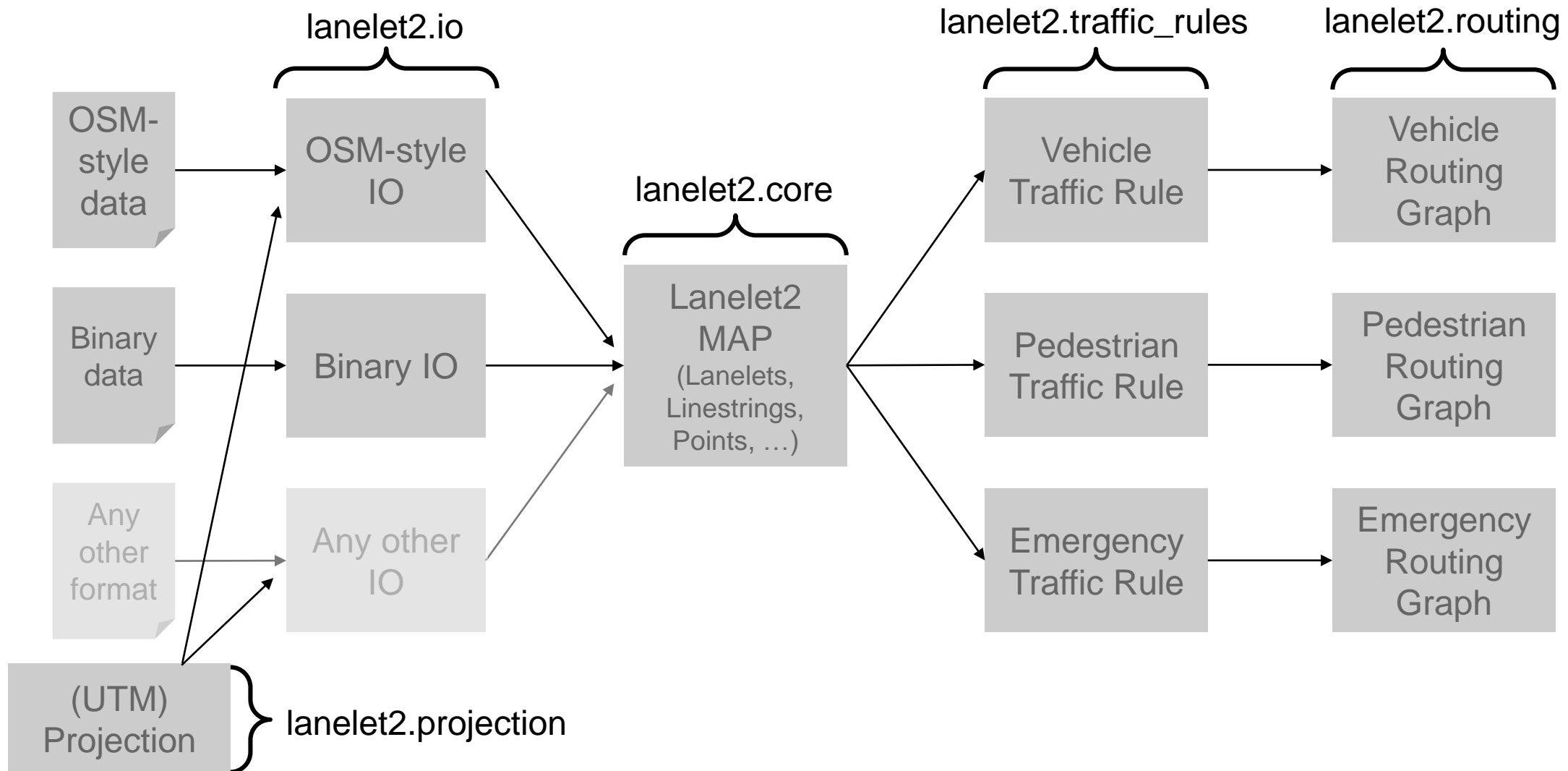
ARCHITECTURE



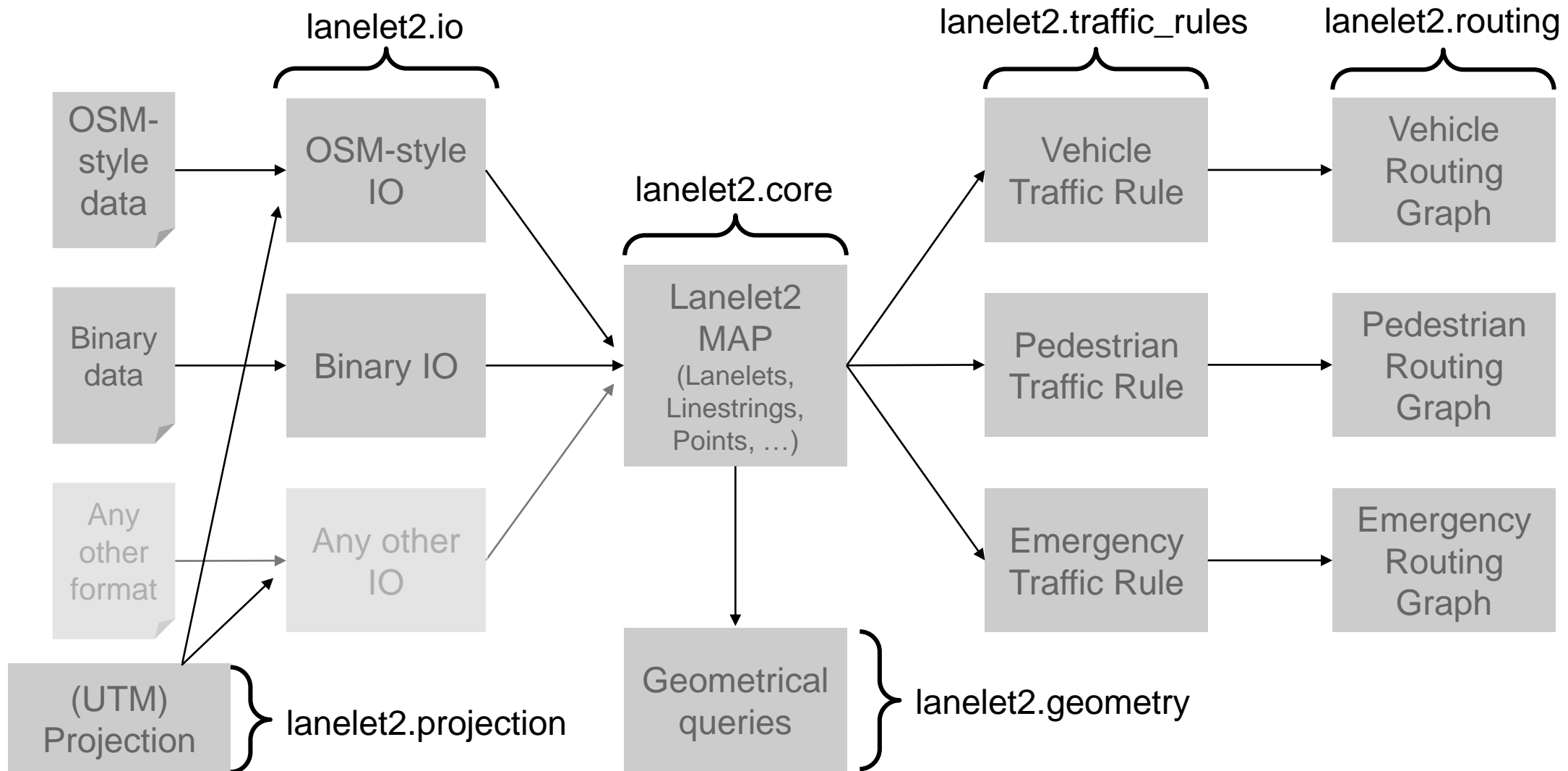
ARCHITECTURE



ARCHITECTURE

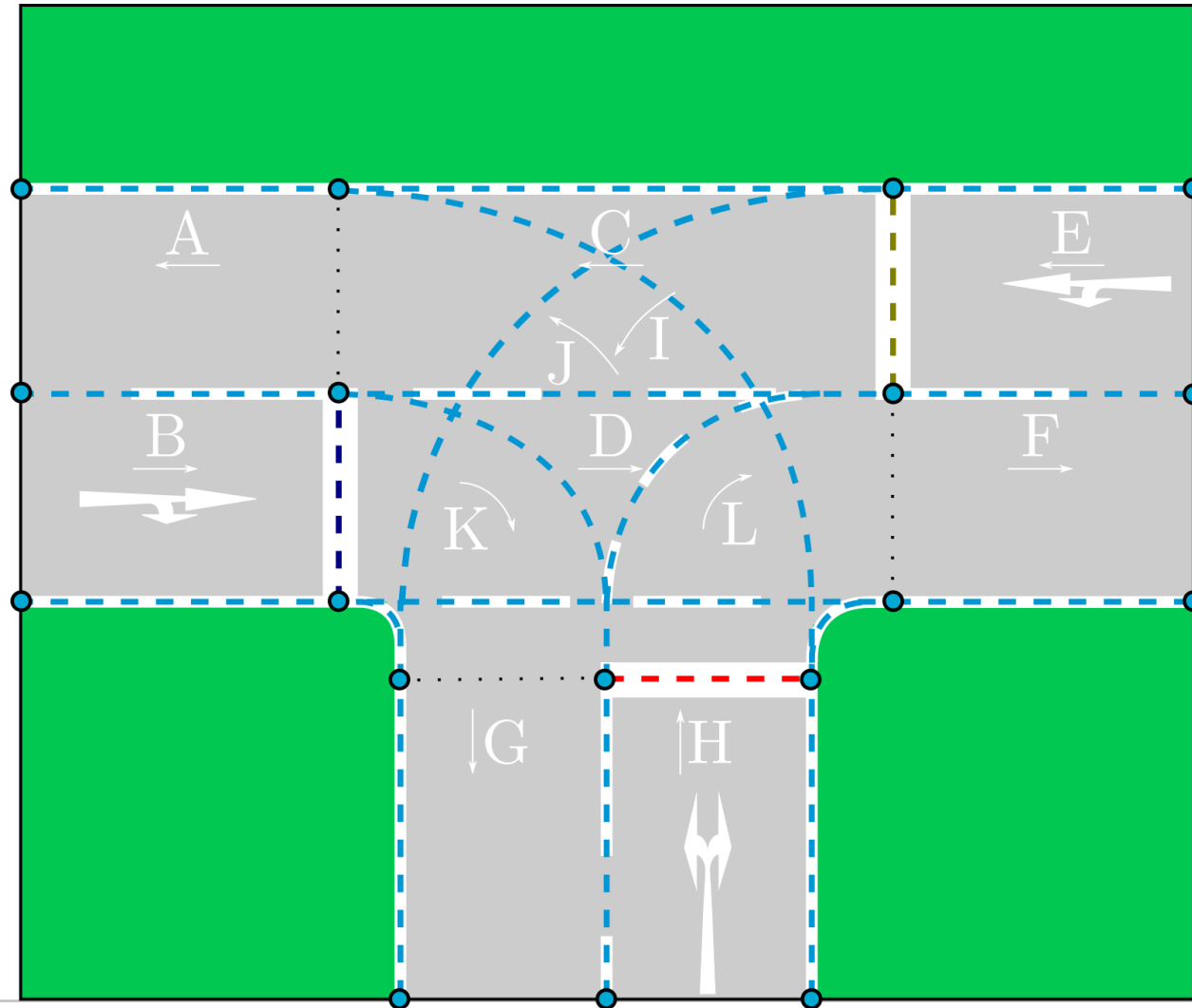


ARCHITECTURE

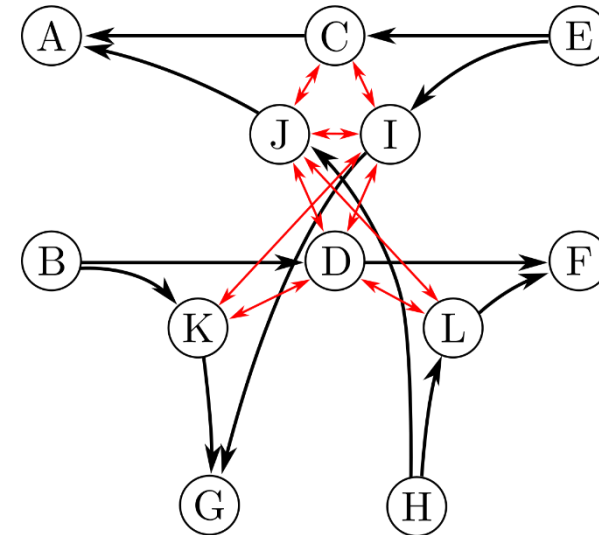
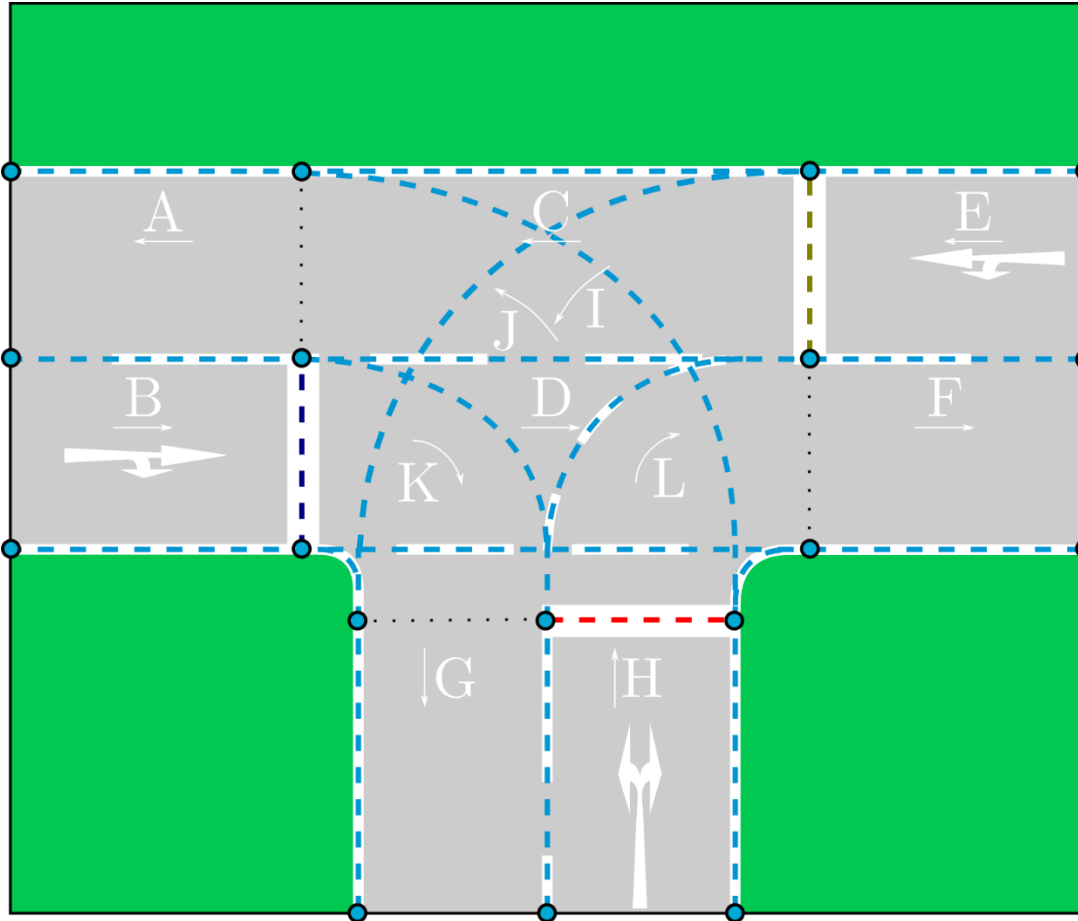


LANELET2 ROUTING+TRAFFIC RULES

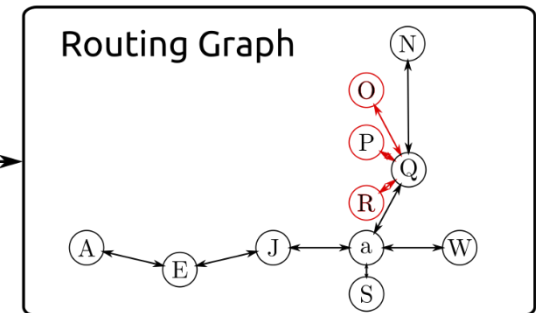
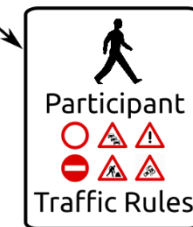
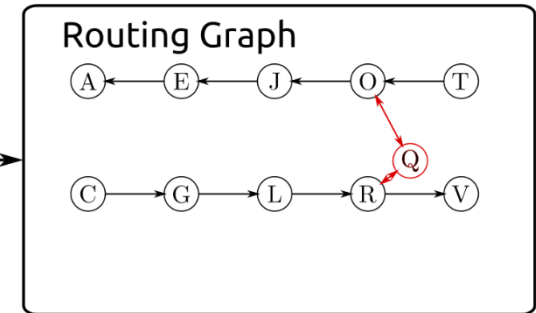
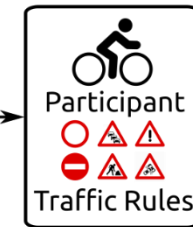
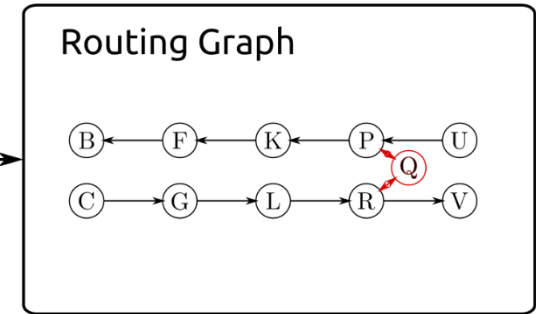
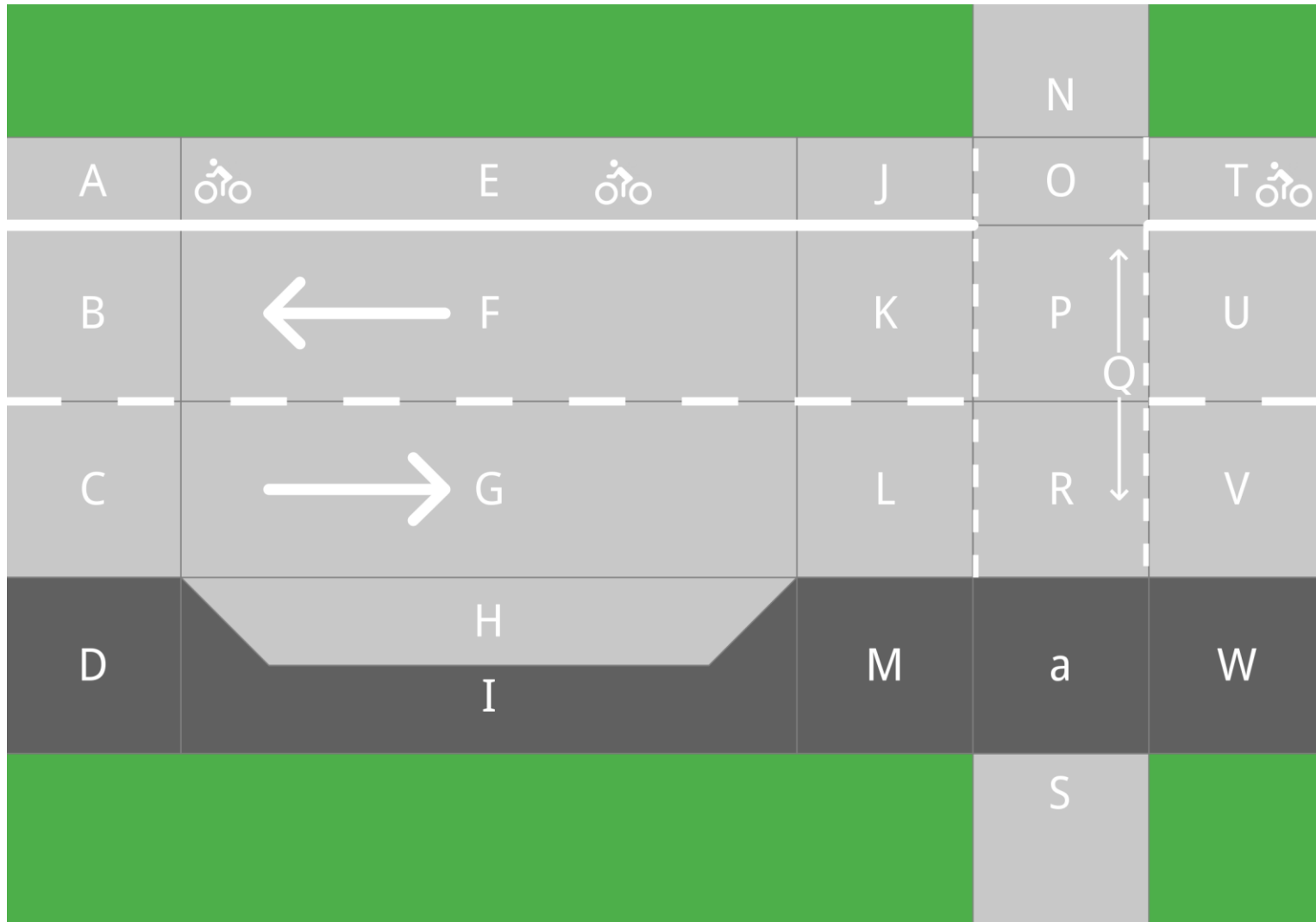
ROUTING



ROUTING



TRAFFIC RULES



LANELET2 IN THE REAL WORLD

ROUTING



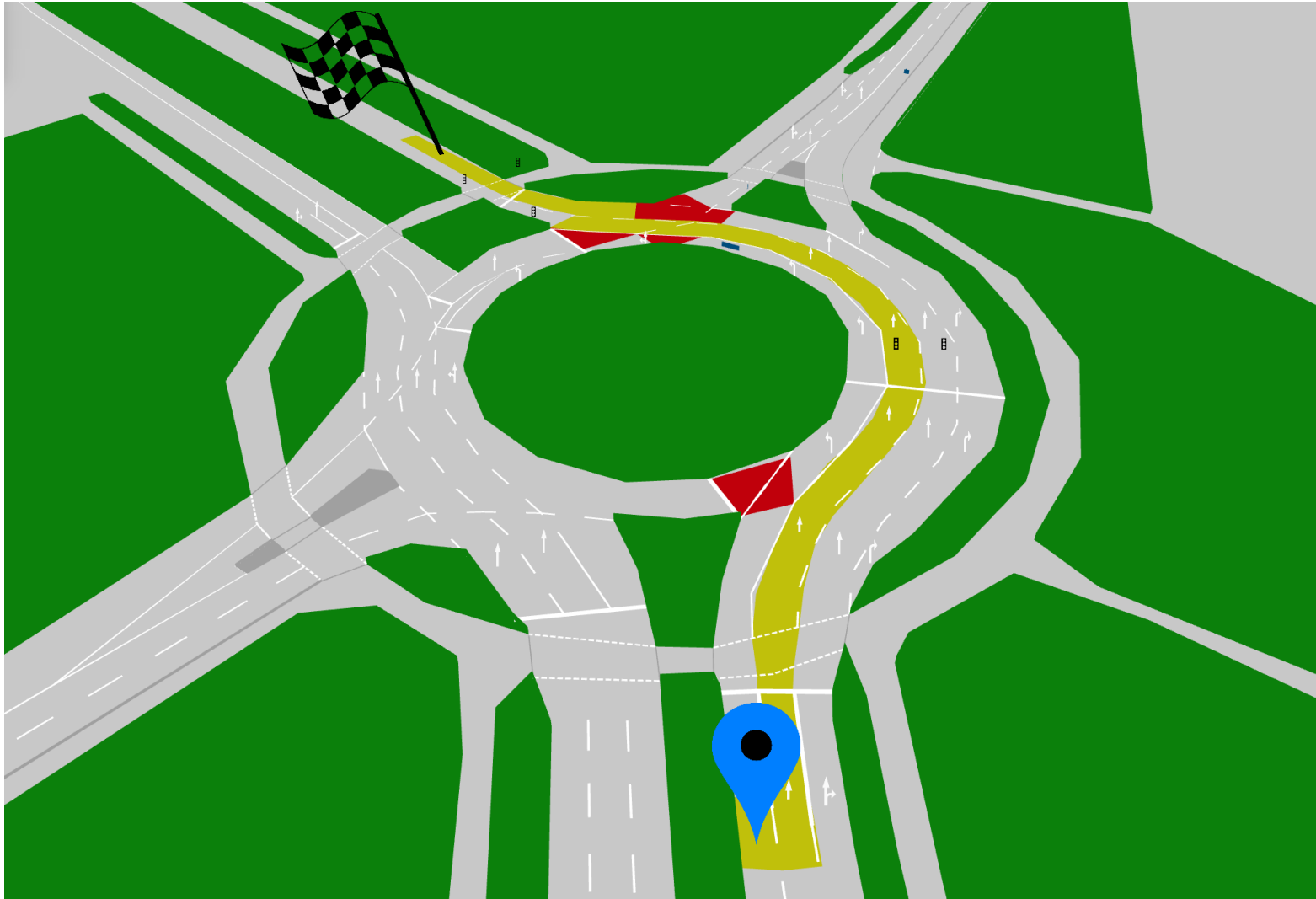
ROUTING



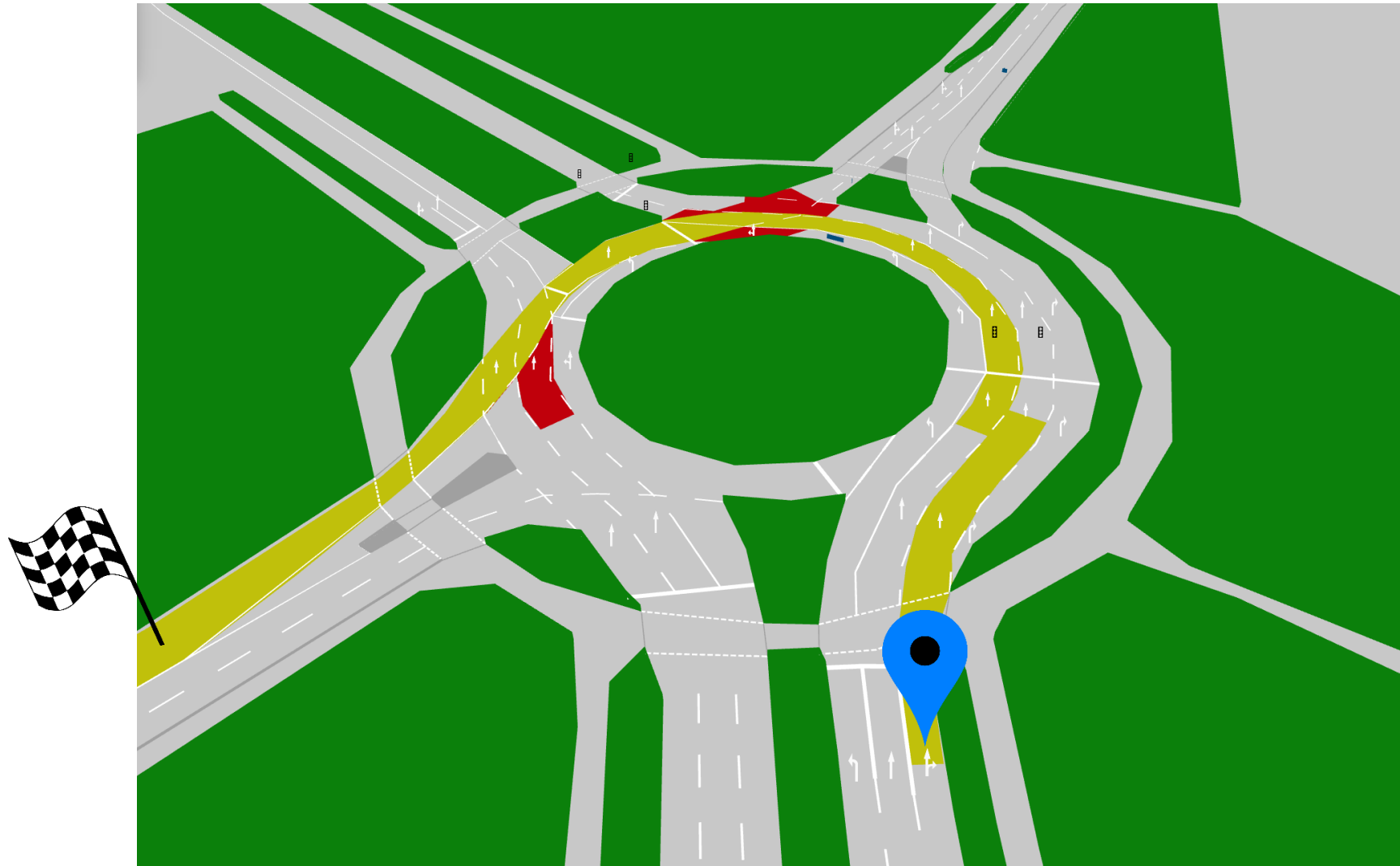
ROUTING



ROUTING



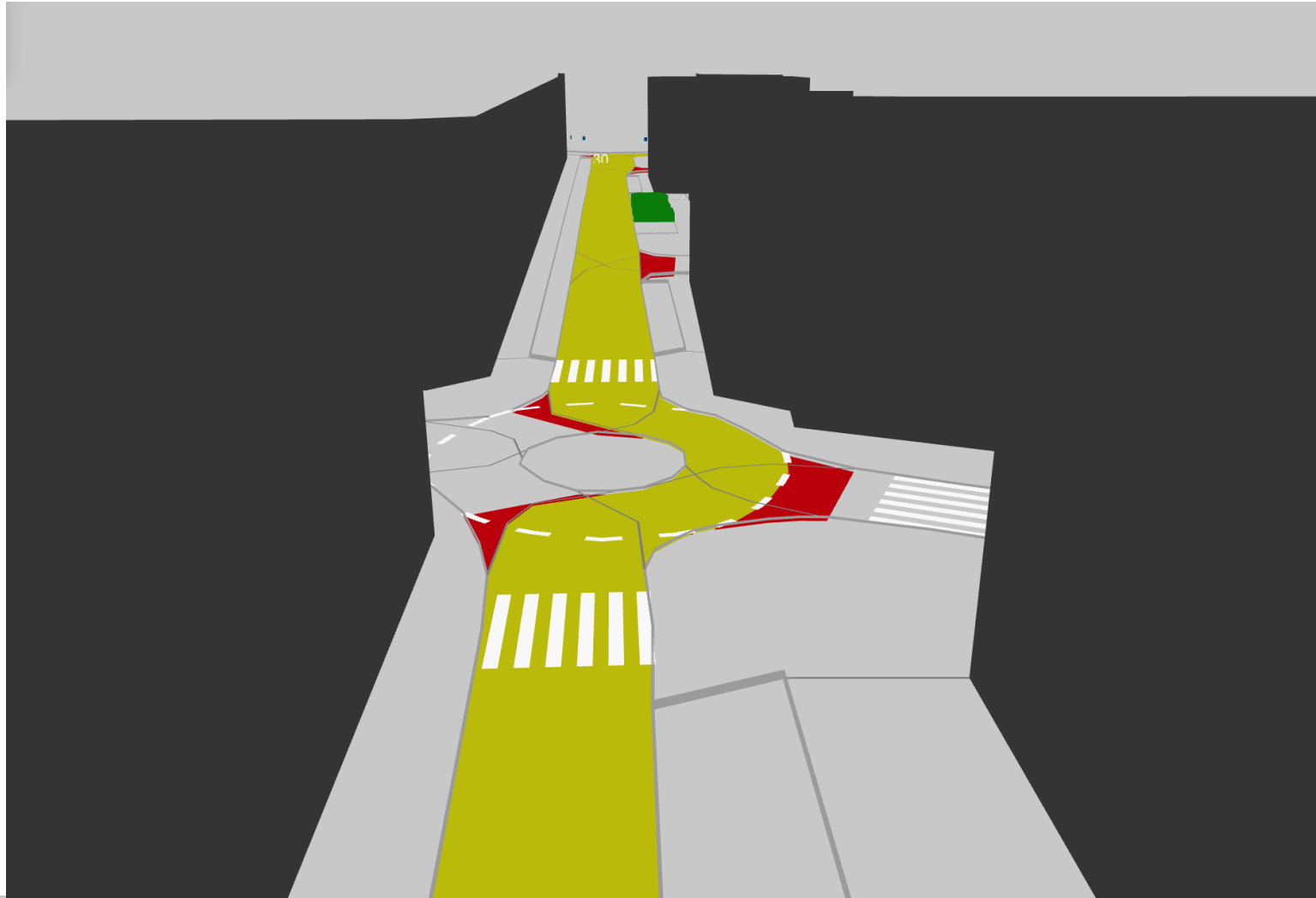
ROUTING



ROUTING



ROUTING



(CLEARLY BIASED) PRO AND CONS



Pros:

- Basically unrivalled open-source map framework
- Permissive license
- Widely used
- Allows for very fine-granular maps
- Modular, customizable and extensible
- Tested and used in lots of different contexts
- Open-source tooling for map creation and visualization available (based on OSM and Rviz)

Cons:

- Lack of publicly available maps/datasets
- Unclear future