Simulation layer: Documentation of the logical channels and their embedded topic structure

In the following, there is an overview of the topic structure that can i.a. be used to communicate with the simulation service.
“.” acts as a hierarchy level delimiter. Therefore the point cannot be used for names, attributes, etc.. The top 3 hierarchy levels are given below:

channel.component.scenarioRun.{channelSpecific}

(Initially it was ***component.channel.scenarioRun…*** **,** but since the structure might be used by more components, the switch might fit better)

Orchestration Channel

The only exception to the given structure comes with the **orchestration.simulation** topic, as it is the bootstrap topic. All further actions rely on a simulation request by publishing a sce-file here.

An exemplary base topic for a requested simulation run is for example
orchestration.simulation.simulationRun12312.{channelSpecific}

Channel specific part for orchestration
Contents of **{channelSpecific}** are further described in the following:

subtopic	**values**	**comment**
.sync	SyncMsg syncMsg	For syncing the logical clocks
.status	StatusMsg statusMsg	Request or post a status
.ctrl	CtrlMsg ctrlMsg	Requesting e.g., the termination of a run

Interaction Channel

The channel specific part of the topics in the interaction channel are further split into a general interaction and a layer specific part. The general part gives info about the domain and potentially, about the layer, and the instance.
channel.component.scenarioRun.domain.[layer.instance.]scope.[entity.attribute.]method

General interaction part for interactions with microscopic traffic simulations
It is not necessary to name the instance when a request is made. Let's say a traffic scenario is partially distributed to two sumo instances and we want to get the position of a vehicle with the id "veh42". We would not know if sumo0 or sumo1 is currently responsible for the vehicle. Therefore, the request would be published to:
interaction.simulation.scenario42.traffic.micro.vehicle.veh42.position.get

Layer specific part for interactions with microscopic traffic simulations
Will be implemented until end of Sept. 2020

scope	entity	attribute	method	values	comment
vehicle	{vehID}	acceleration	get		
			ret	double acceleration	
vehicle	{vehID}	angle	get		
			ret	double angle	
vehicle	{vehID}	edge	get		
			set	string edge	Map to moveto
			ret	string edge	
vehicle	{vehID}	emissions	get		
			ret	string emissions	
vehicle	{vehID}	lane	get		
			set	string lane	Map to moveto
			ret	string lane	
vehicle	{vehID}	position	get		
			set	string lane	Map to moveto
			ret	string lane	
vehicle	{vehID}	route	get		
			set	string route	
			ret	string route	
vehicle	{vehID}	speed	get		
			set	double speed	
			ret	double speed	
vehicle	{vehID}	speedMode	get		Too tool specific?
			set	int speedMode	
			ret	int speedMode	
vehicle	{vehID}	genericParameter	get		
			set	string params	
			ret	string params	
vehicle	{vehID}	--	moveTo	string lane, double pos	
			changeTarget	string destination	
			setStop	string edge, double pos, double time	Make a break, e.g. charging

scope	entity	attribute	method	values	comment
world	--	vehicleIDs	get		
			ret	string vehicleIDs	
world	--	boundaries	get	--	
			ret	double x1, double y1, double x2, double y2	
world	--	edges	get	--	
			ret	string edgeIDs	
world	--	--	addVehicle	micro fullDesc	
			removeVehicle	string vehicleID	
			exportSim	double time, String formats	Snapshort for UC3

scope	entity	attribute	method	values	comment
road	{roadID}	speed	get	--	
			ret	double avgSpeed	
road	{roadID}	density	get	--	
			ret	double avgSpeed	
road	{roadID}	flow	get	--	
			ret	double avgSpeed	
road	{roadID}	emissions	get	--	
			ret	string emissions	

Provision Channel

Currently, this is very poor specified. For testing purposes the following topics are supported. More will come and maybe there will be changes in the structure.

Sub Topics for **channel.component.scenarioRun...**

subtopic	**data type**	**comment**
.resource	Avro::ResourceFile rf	RoadMap, Traffic Input, ...
.scenario	Avro::Scenario scenario	Copy of requested scenario definition. Also used is ack for simulation request.
.results	ZIP-Archive of results that were requested with sce-file OR reference to the same zip that was already uploaded to the storage component	
.micro.vehicle.{vehID}.micro	Avro::Micro	Currently, these 6 attributes can be requested for the provisioning within the observer component. string return type for simple test → may be changed to more efficient avro types. In the future
.micro.vehicle.{vehID}.acceleration	string	
.micro.vehicle.{vehID}.edge	string	
.micro.vehicle.{vehID}.position	string	
.micro.vehicle.{vehID}.speed	string	
.micro.road.{roadID}.emission	string	

Possible error codes for integer and double return values

```
0xBAAAAAADDEADFEED   No Connection to Instance
0xBAAAAAAD00004B1D  Call Forbidden
0xBAAAAAAD8BADF00D   Input not valid
0xBAAAAAADDEADBABE   No memory left
0xBAAAAAADDABBAD00   Method not supported by Instance
0xBAAAAAAD0D15EA5E   --
0xBAAAAAADD0D0CACA   --
```