Instructions for Running Systems Technology Interaction Model (STIM)

1. Folder Structure

Systems Technology Interaction Model-|

|-csv-|

| |- Automation_levels.csv

| |- Behavior_Experiment_Mapping.csv

| |- Behaviors.csv

| |- Experiment_ExtInfluence_Mapping.csv

| |- Experiments.csv

| |- External_Influences.csv

| |- ExtInfluence_IntInfluence_Mapping.csv

| |- Feature_Behavior_Mapping.csv

| |- Feature_Level_Mapping.csv

| |- Feature_Metrics.csv

| |- Feature_Metrics_Mapping.csv

| |- Feature_ODD_Mapping.csv

| |- Feature_SystemTech_Mapping.csv

| |- Features.csv

| |- Int_Influences.csv

| |- IntInfluence_SysTech_Mapping.csv

| |- ODD.csv

| |- ODD_Experiment_Mapping.csv

| |- ODD_OES_Mapping.csv

| |- OES.csv

| |- OES_Experiment_Mapping.csv

| |- System_Technologies.csv

| |- Usage.csv

| |- Usage_Experiment_Mapping.csv

|- cypher queries-|

| |-Neo4J_Queries.csv

|- import model-|

| |- demo_model.zip

| |- neo4j.dump

|- neo4j json model-|

| |- neo4j_importer_model.json

Description

- Csv: This folder contains csv files which have only data and node mapping information for STIM. To change or add new information, users can change these files and import updated data in the “import” tab in neo4j instance.
- Cypher queries: This folder contains a csv file containing a few cypher queries to filter required data. To use this, a user can import this csv file in the “query” tab in neo4j instance.
- Import model: This folder contains two files using which the user can upload model+data in one go. To do this, users can either upload the zip file in the “import” tab in neo4j instance or use the .dump file to upload a snapshot when running the instance.
- Neo4j json model: The json file contains the blueprint of the model. To view this, users can upload it on the neo4j instance.

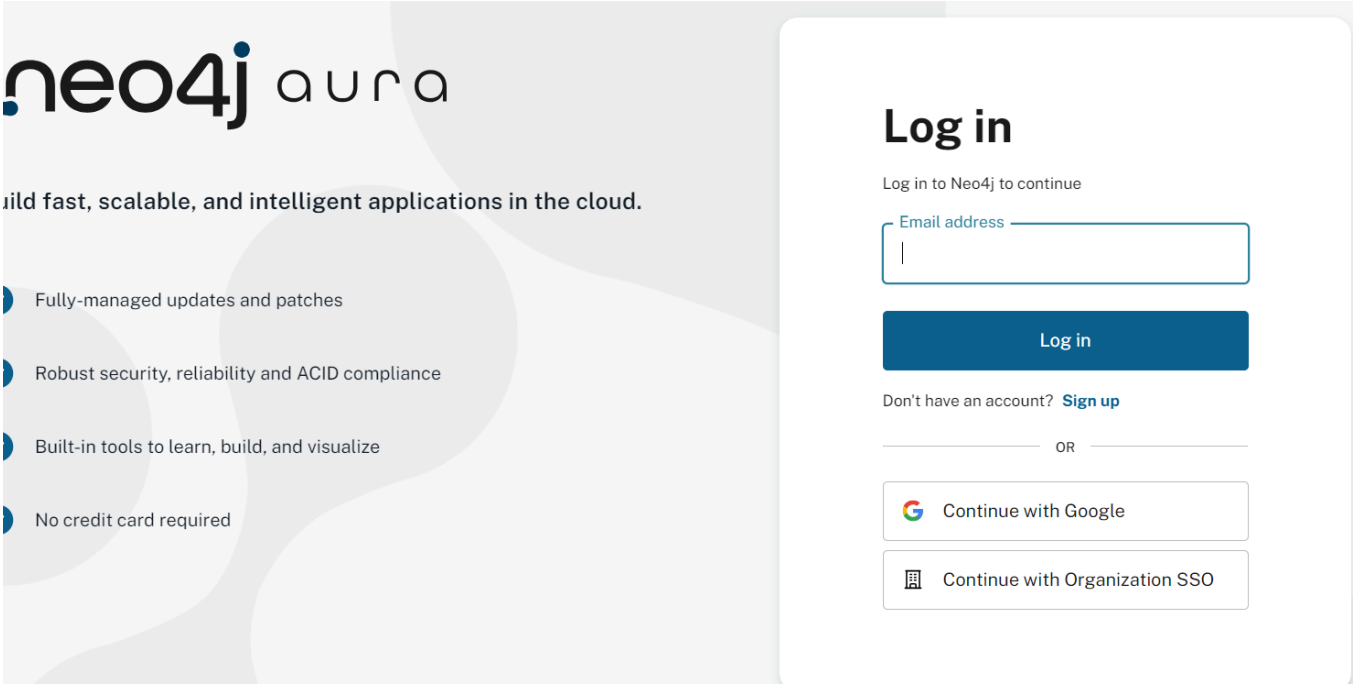
2. Definitions

S.No	Term	Definition
1.	Automation Level	These are levels 1-5 as described in SAE's J3016.
2.	Feature	According to SAE J3016, a feature is defined as the supporting functionalities in the vehicle and these features vary depending on the system's autonomy level within a particular ODD, if applicable. **E.g**. Adaptive cruise control is a feature of AV level 2 where the driver is performing the driving task while the vehicle controls lateral motion.
3.	Feature Metrics	A metric on a feature, or feature metric, is a specification or set of constraints on the performance of the Feature. **E.g.** If the Feature is Obstacle Collision Avoidance, one metric would be that the system maintains a distance of at least *d* meters from any obstacle. If the Feature is Adaptive Cruise Control (ACC), then one metric would be that the system accelerates and brakes as needed to maintain a set point and maintains a distance of 2 car lengths to the vehicle in front of it in the lane.
	ODD	Operational Design Domain(ODD) is a set of operating conditions for an autonomous vehicle. These conditions include environmental, geographical, time of the day constraints as well as traffic and roadway characteristics**.** **E.g.** Time of day, weather, geofencing, boundaries for operation etc.
4.	System Technology	System technologies are building blocks of AV(automated vehicle) features and other functions within AV. The focus remains on four pivotal system technologies in our work - Cybersecurity, Communication, AI and Perception.
5.	Internal Influence	It is an attribute of an interaction between system technology output that may adversely affect the performance of one or more other system technologies and this influence will affect the feature's performance as well. **E.g.** For Perception system technology, inaccuracy in sensor data is one of the internal influences that is outputted by that system technology, and may affect Feature's performance.

6.	External Influence	These are the external contributors that cause internal influences to take place. **E.g**. Possible external causes for inaccuracy in sensor data can be weather(rain, snow etc), speed of vehicle or physical damage to sensors.
7.	Behavior	Behavior is defined as a list of actions the AV performs in a given ODD for a feature during the dynamic driving task(DDT). **E.g.** An AV can have behaviors like maintaining a lane, change lane, responding to emergency vehicles etc.
10.	OES	An Operating Envelope Specification (OES) is a "structured description of the operating environment for driving, usable for formal reasoning (i.e., calculation based reasoning) about that environment in testing and certification applications and in real-time driving conditions". An instance of an OES comprises the dimensions of the operational state space (whether chosen by the manufacturer, developed from a relevant scenario set, or defined de novo) sufficient to enable formal reasoning about the state space. **E.g.** For ODD of divided access controlled highways, one of the information OES stores is lane dimensions and number of lanes etc.
8.	Scenario	In the context of AVs, a scenario is formally defined as a specific set of conditions and configurations under which an AV's performance and behavior are tested. This definition encompasses various elements such as the ODD and the OES. Scenario-based testing is crucial for validating and verifying AV safety and functionality by simulating real-world conditions and examining the behavior of an AV. **E.g.** ● Straight road on highway 495 where width is 3.2m (taken from OES) ● Prepares for exit 12 ● Takes exit 12 ● Detect obstacle in front at *X* meters ● Applies brake at *d* meters from *X* ● Stops at safe distance d' from obstacle ● End scenario This description of scenario can be further populated with specific ODD information including location of traffic signs, width of road, type of intersection etc.
9.	Usage Specification (SAE J3016)	The concept of Usage Specification is being developed by the Exploratory Working Group from SAE to capture vehicle automation usage or purpose, automation level, and operational constraints.

3. Setup Neo4J Instance

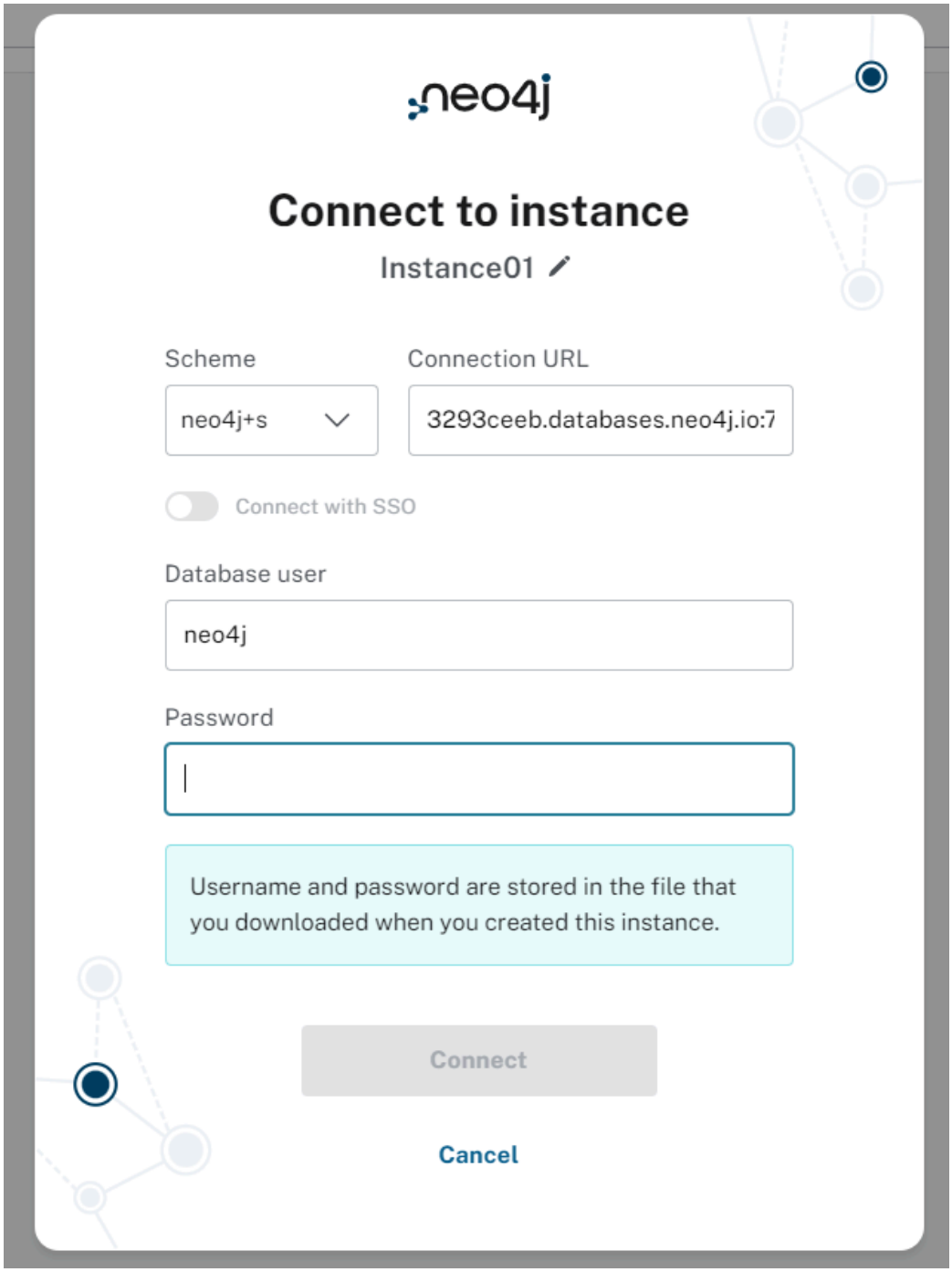
- **Open** Neo4J website → Link
- **Sign up** with work or personal email id. (**Login** if you already have an existing account.)



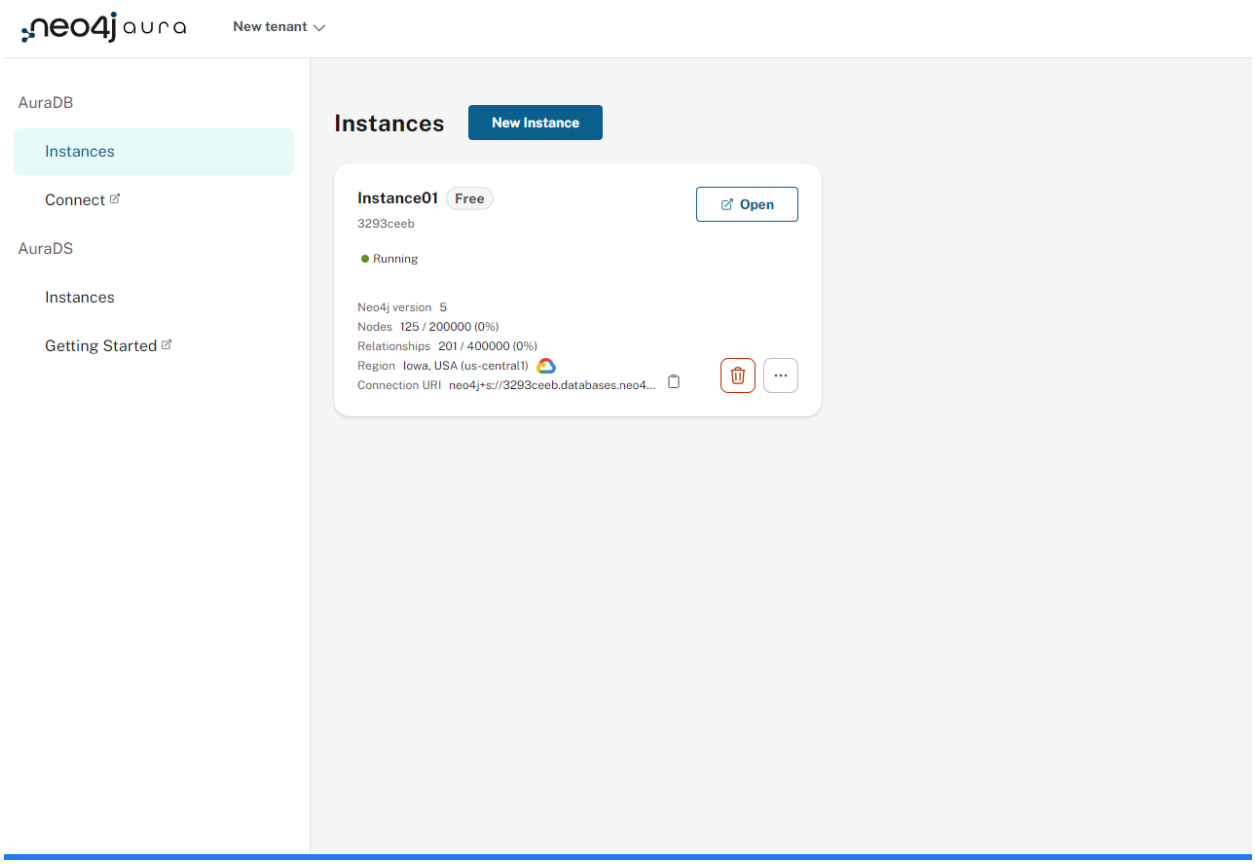
- Choose the "*create free instance"* option. It will ask you to **save the database password file** and automatically download a .txt file**.** Copy the **password** on a microsoft sticky note/somewhere accessible for ease.

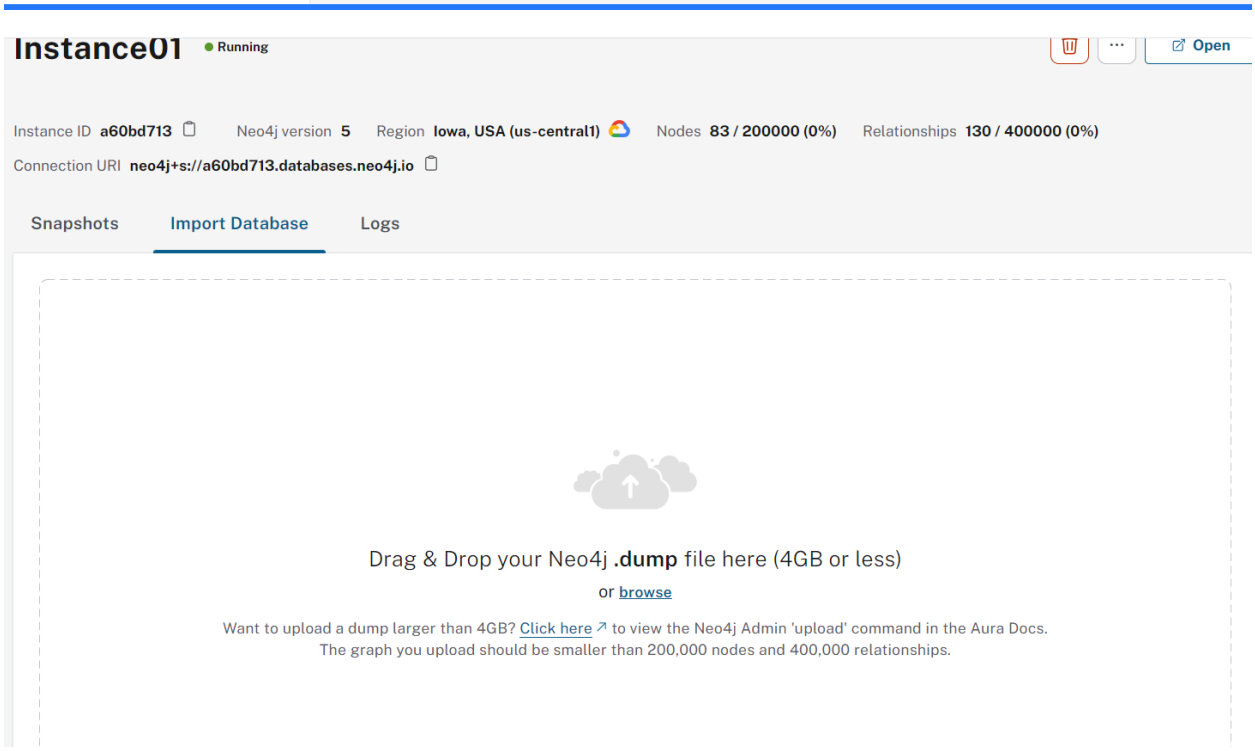
Tip: You can change this password to something simple once you are inside the instance.

- Enter the **database password** when the "*connect to instance*" window pops up.



- Once you open the instance, you can now load your Neo4J model.
- The window mentioned below shows the instance information and status(in this case, it's running). There are two ways to load the model+data from here: If the user has a .dump file, double click this instance and import the dump file in the “Import Database” tab.



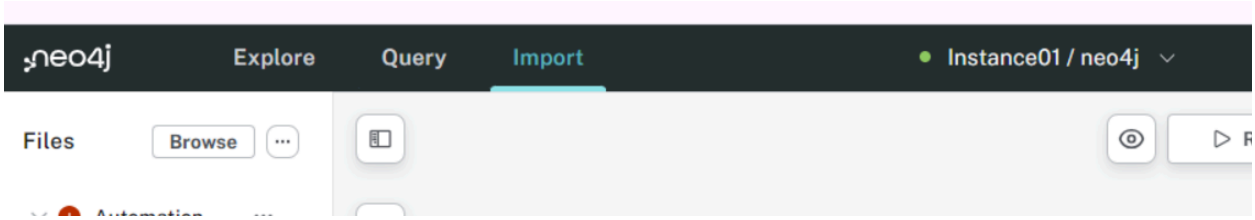


- The second way to upload model+data is to upload a .zip file in the instance.(see point 4)

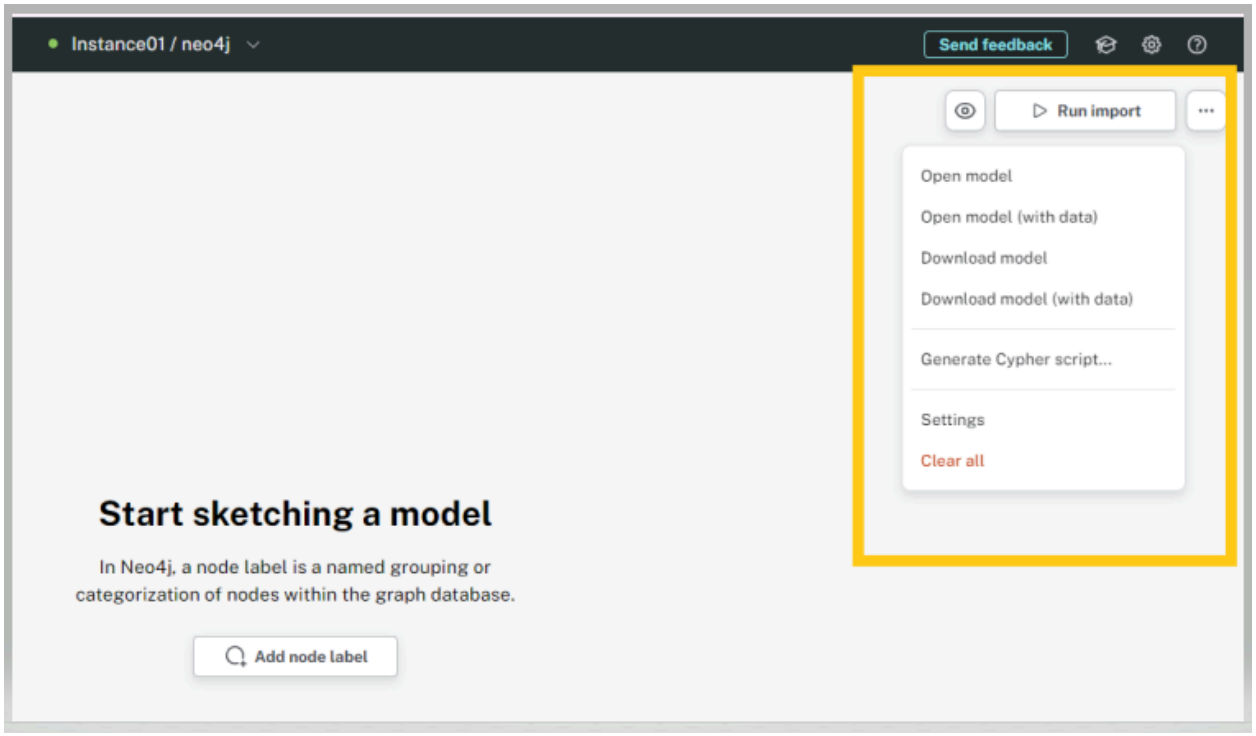
- Click "*open*" and connect to your instance and put the database password as mentioned in the previous step.
- You are now **ready to load the model** into Neo4j.

4. Load Model

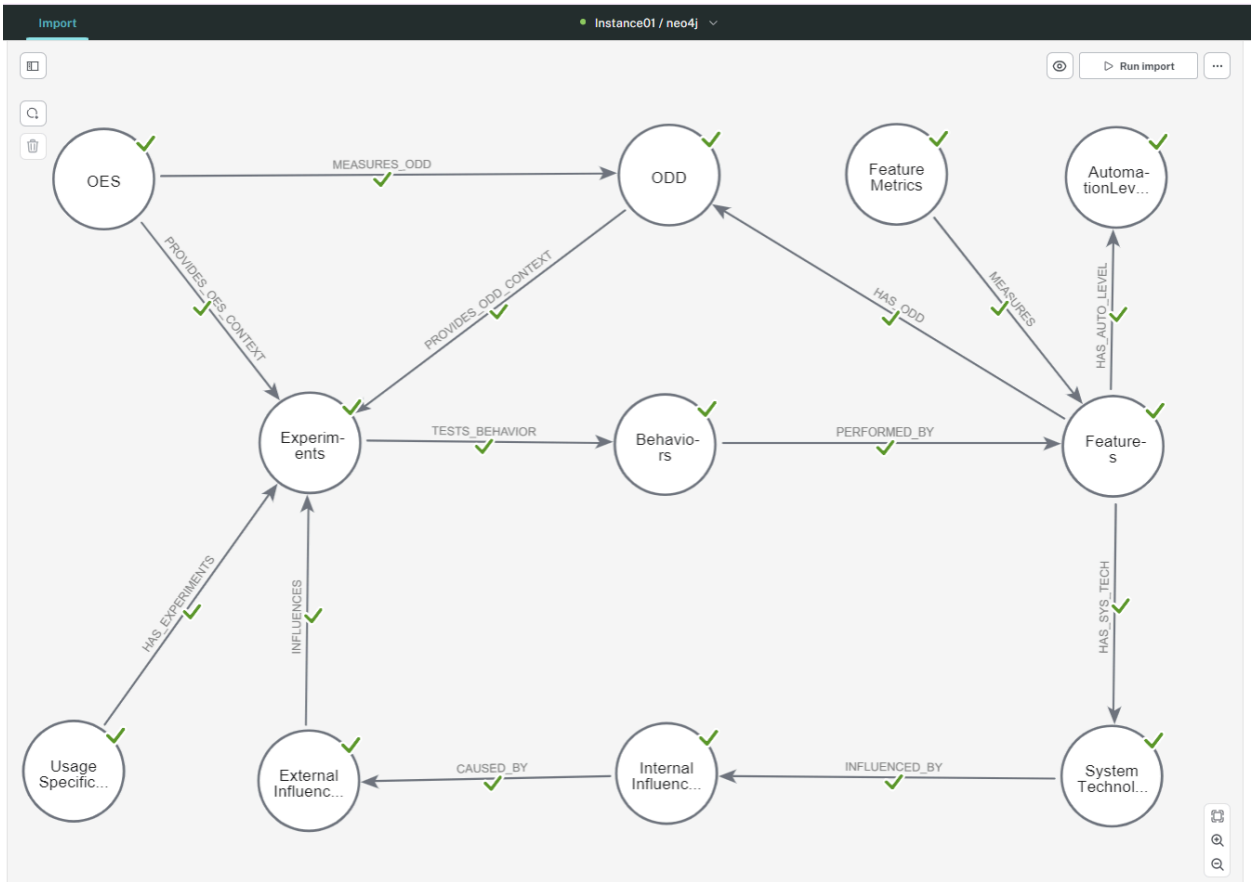
- To load the model, go to the "*import"* tab on top left. (Neo4j has 3 tabs: Explore, Query and Import)



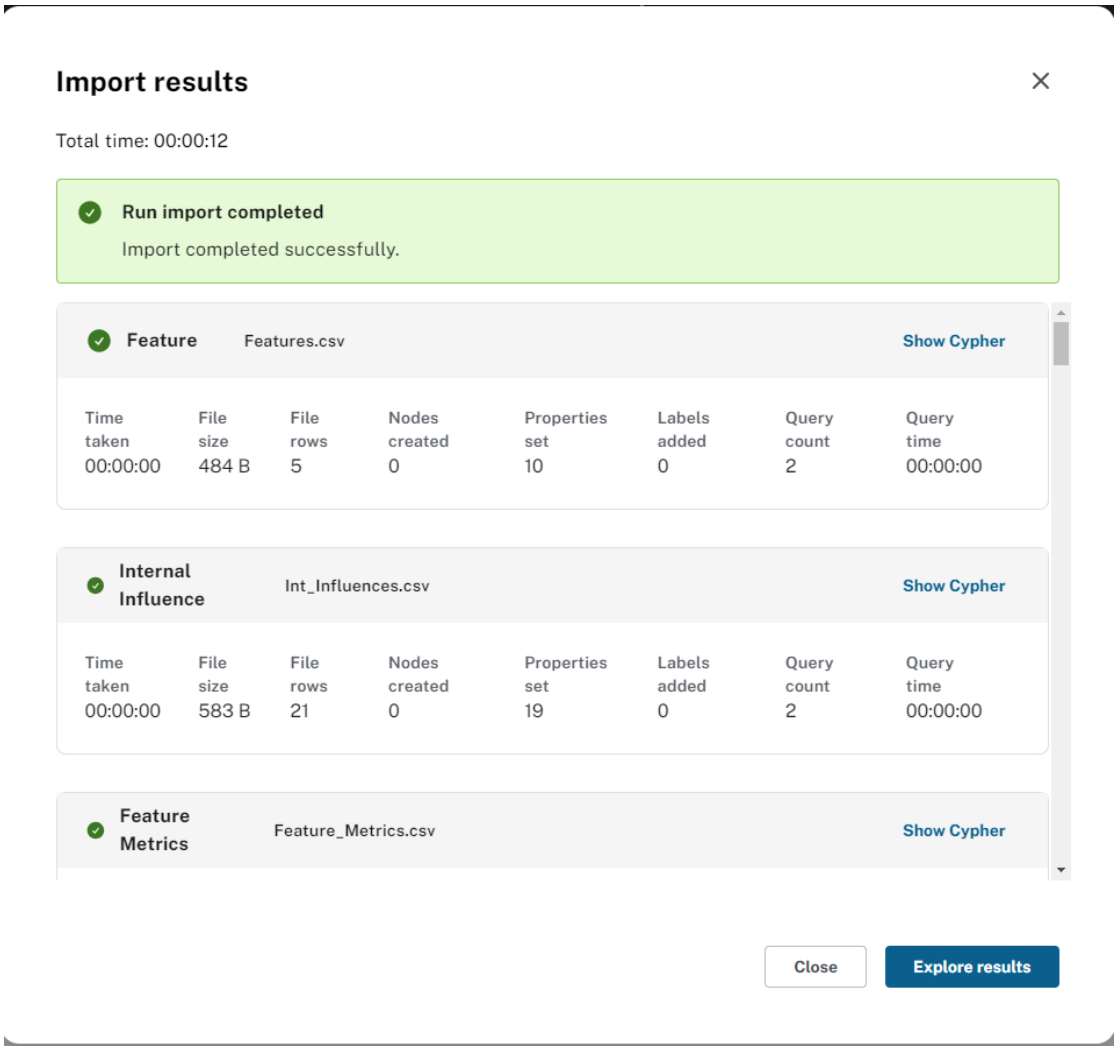
- Use zip file *"Systems Technology Interaction Model\import model\demo_model.zip"* which contains csv data files and a .json model file. **(No need to unzip the file).**
- Click the menu button(three dots on the right side of run import) and select option "*Open model(with data)*" from the dropdown and select zip file.



- You should now see a model as shown below and the csv files on the left.



- Click *“Run import”.* If the model is made correctly, and files are loaded properly, the run import should run and output something like below.

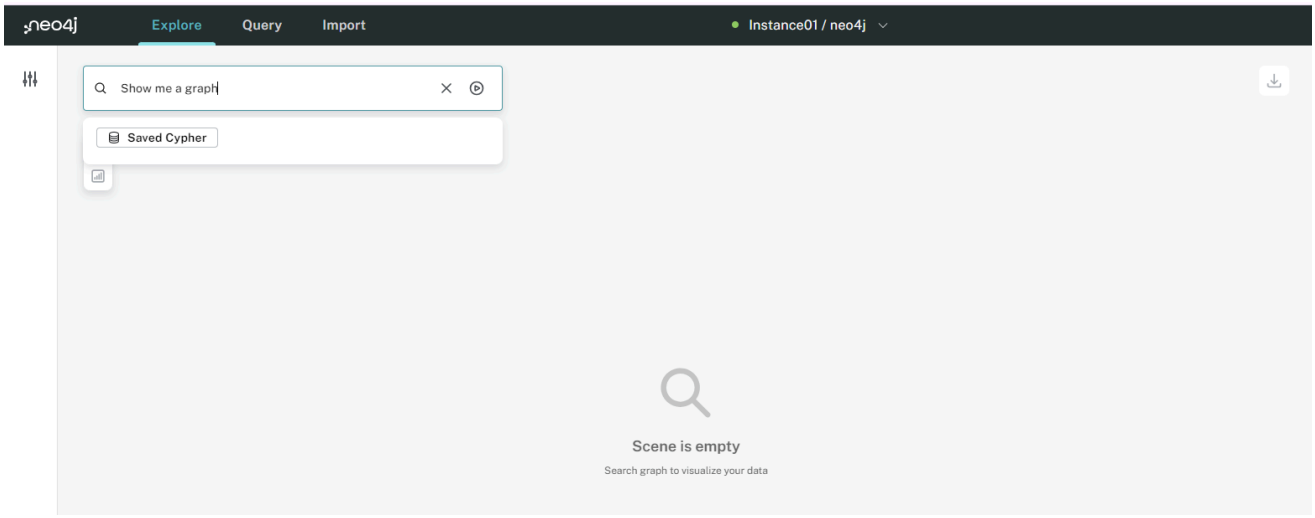


- Click *"Explore results"*. If you click *"Close"* accidentally, select the *"Explore"* tab on top left instead of *"Import".*
- **The model is now ready to explore.**

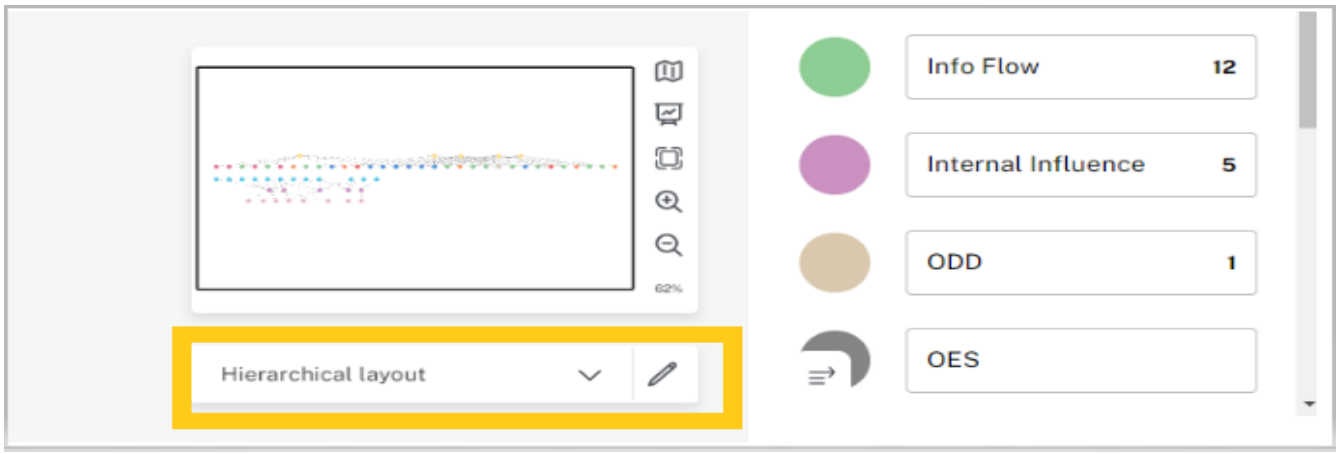
5. Explore Model

- In the search bar on the top left corner, enter "show me a graph". This will populate a random graph using the data in the model. Once nodes show up on the graph, right click in the free space on screen and select *"Clear Scene"* from the menu.
  Tip: You can skip this step if you wish to start with the examples directly, but it is recommended to run this as it initializes the model with data, hence the search becomes

faster for future.



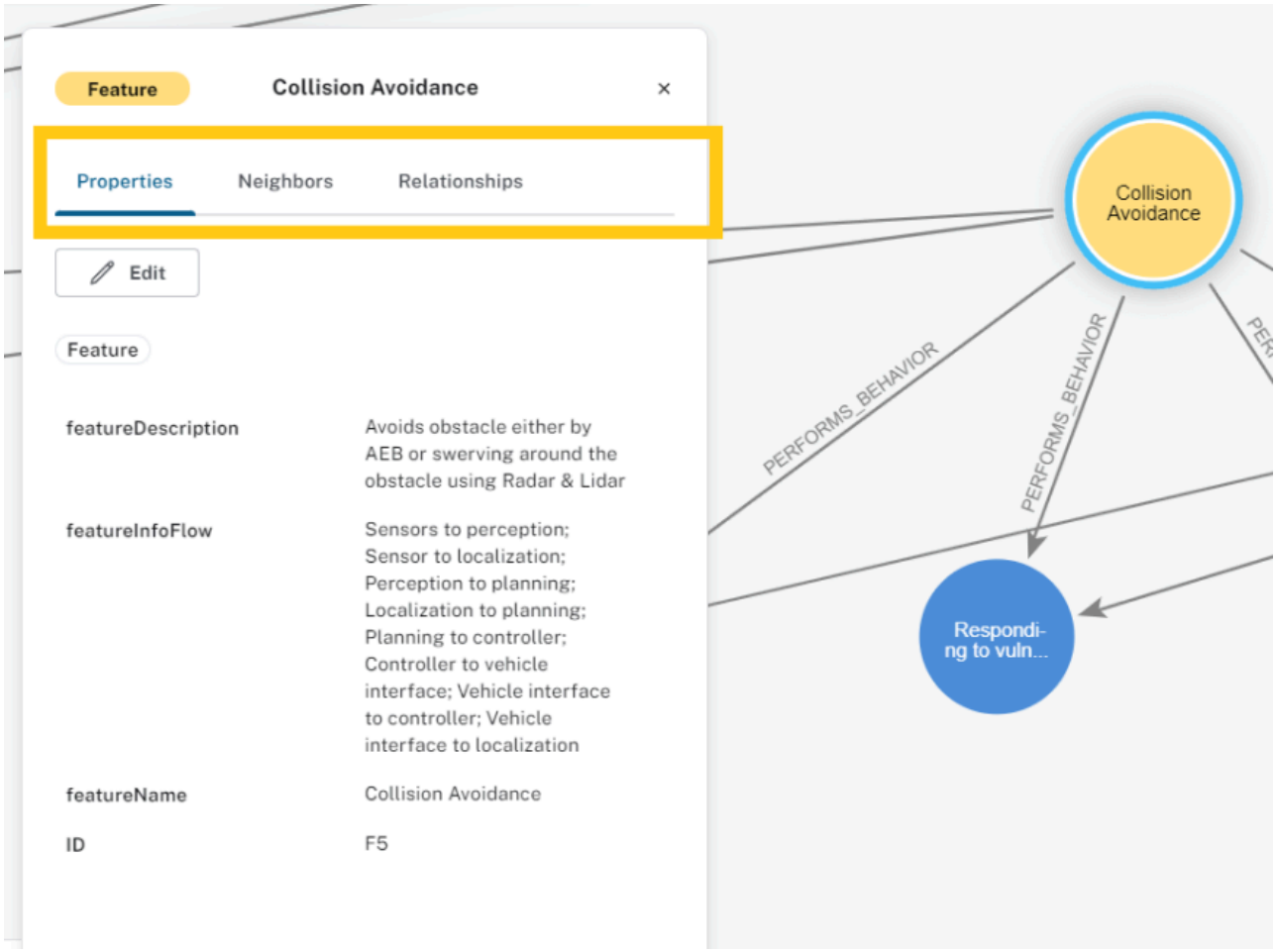
- Switch to *"Hierarchical layout"* from the drop down menu on bottom right to organize nodes.

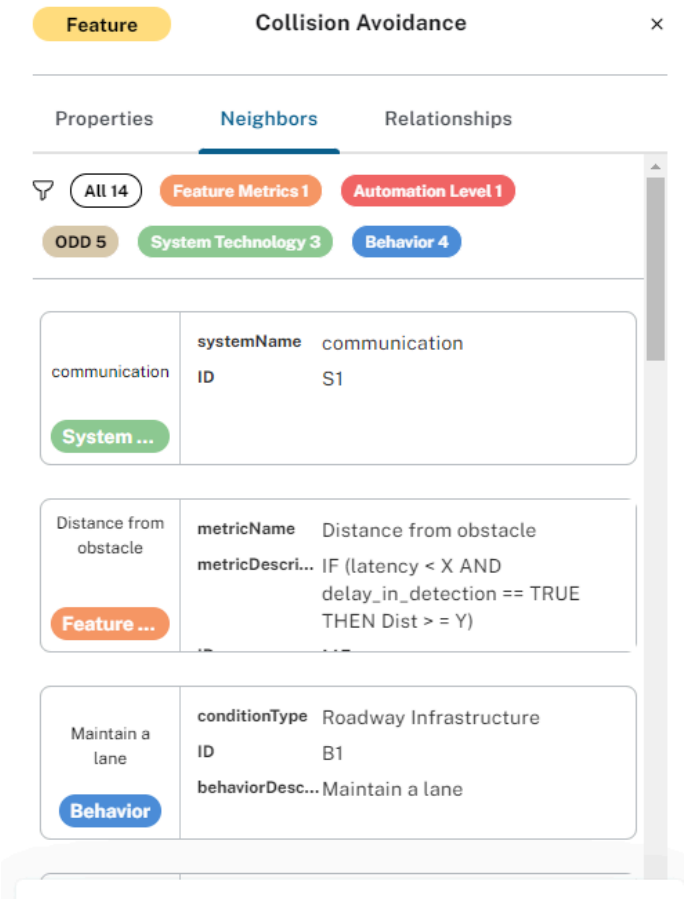


<u>Steps to navigate in model</u>

Note : Navigating in the model will help understand the nodes and their relationship with other nodes. It can be tricky as being a first time user as we might not know what neighboring nodes are or where to start. There are two ways to find neighboring nodes.

1. The best way to start is by looking at the model in the "*import"* tab on top left, where you see the model structure and the relationship between nodes. For your convenience, you can open a screenshot of model structure on a different tab to know where you want to explore next.
2. Another way to know where to go from a node is by double clicking on it. It has a relationship and neighbor for the node hence making it easier to navigate.


Feature
Collision Avoidance
Properties
Neighbors
Relationships
Edit
Feature
featureDescription
Avoids obstacle either by AEB or swerving around the obstacle using Radar & Lidar
featureInfoFlow
Sensors to perception; Sensor to localization; Perception to planning; Localization to planning; Planning to controller; Controller to vehicle interface; Vehicle interface to controller; Vehicle interface to localization
featureName
Collision Avoidance
ID
F5
Collision Avoidance
PERFORMS_BEHAVIOR
PERFORMS_BEHAVIOR
Respondi-ng to vuln...

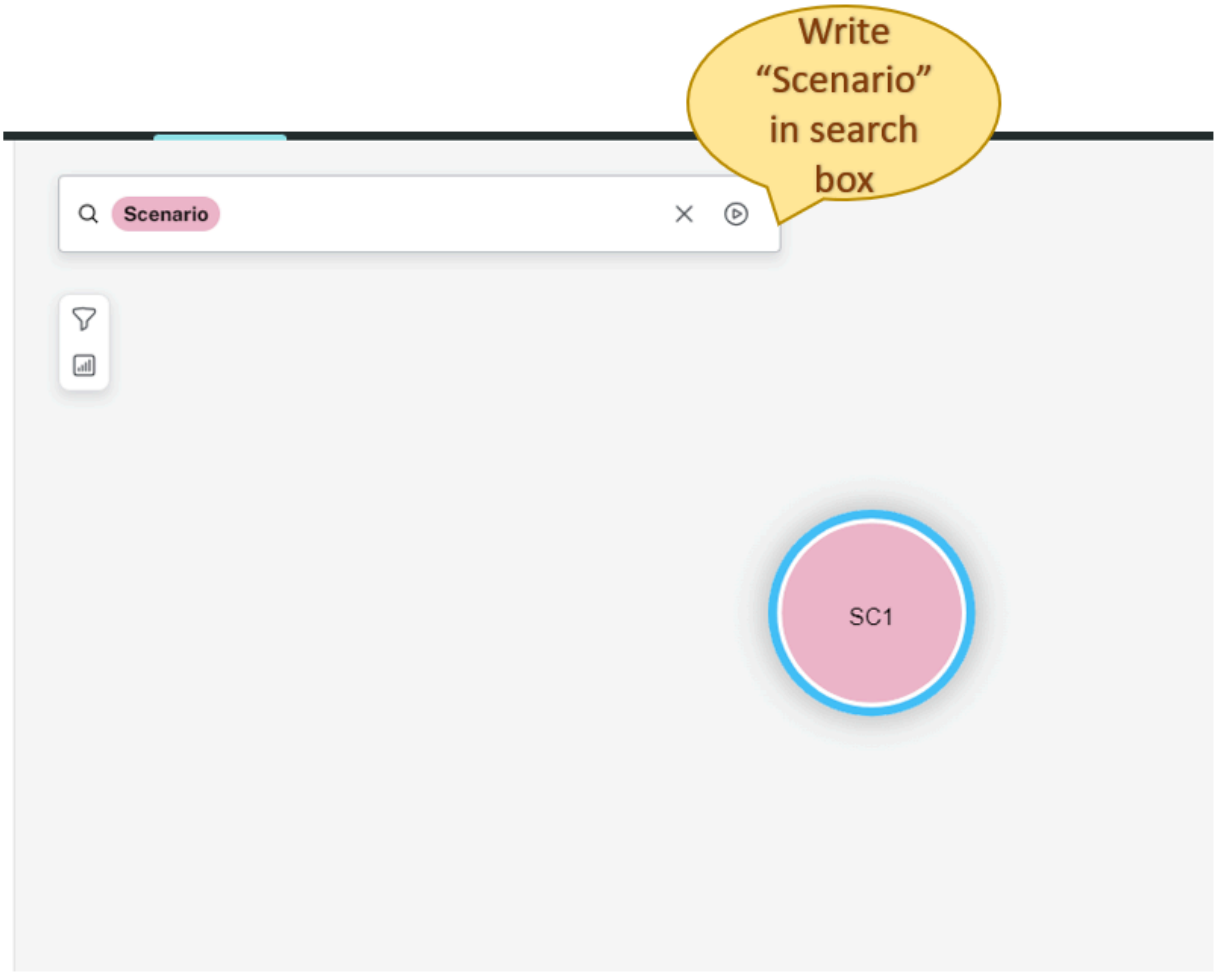


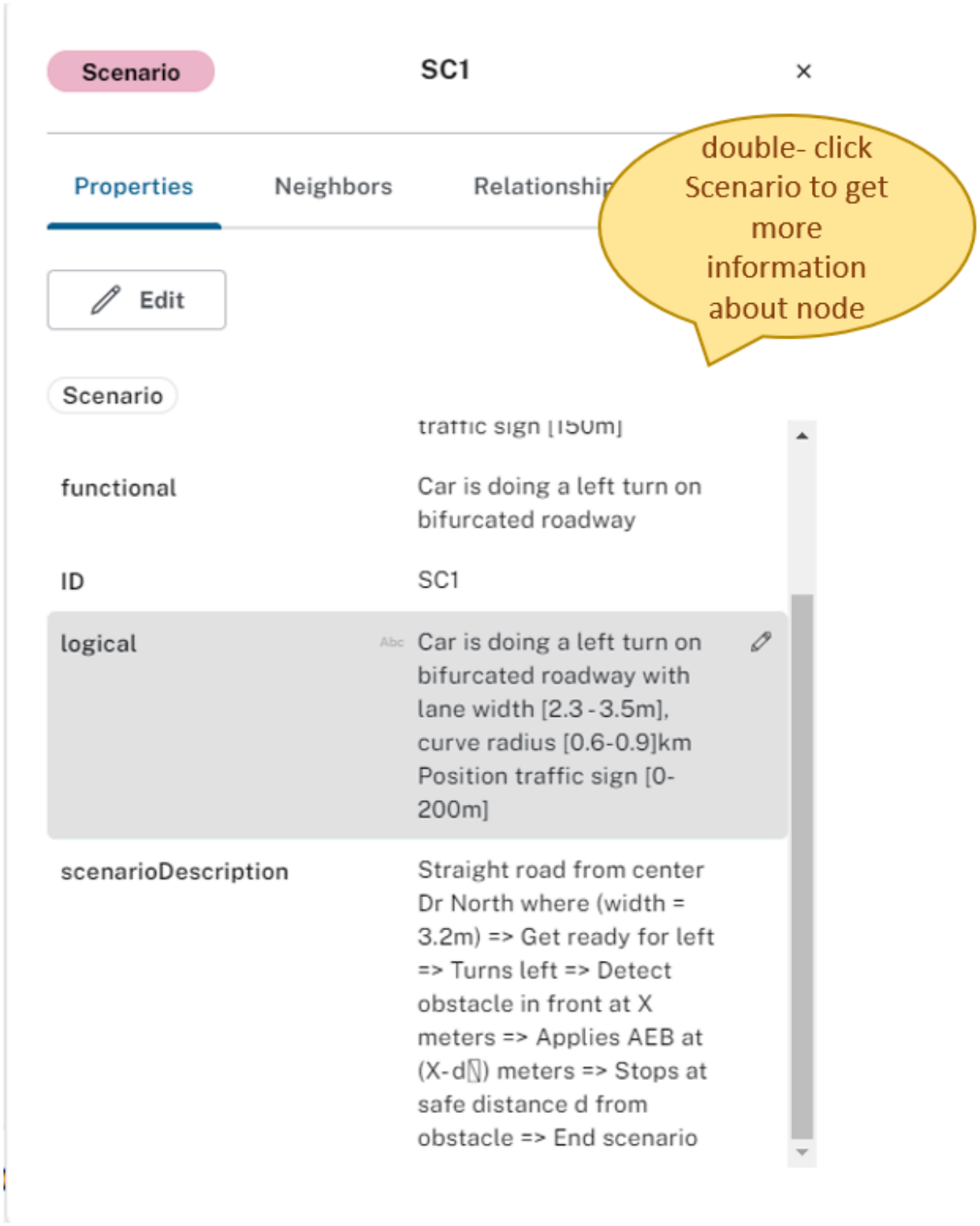
5. <u>Once you have the model opened, what to do with it?</u>

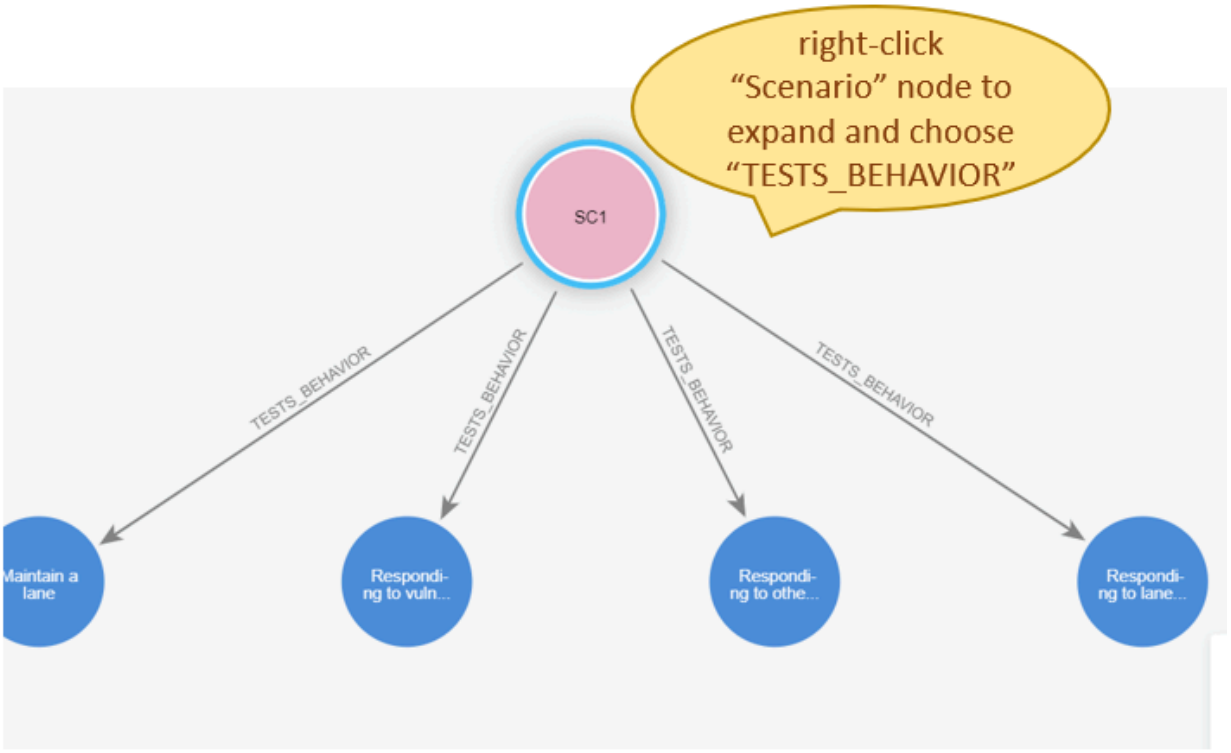
This model can help you answer some of the questions like:

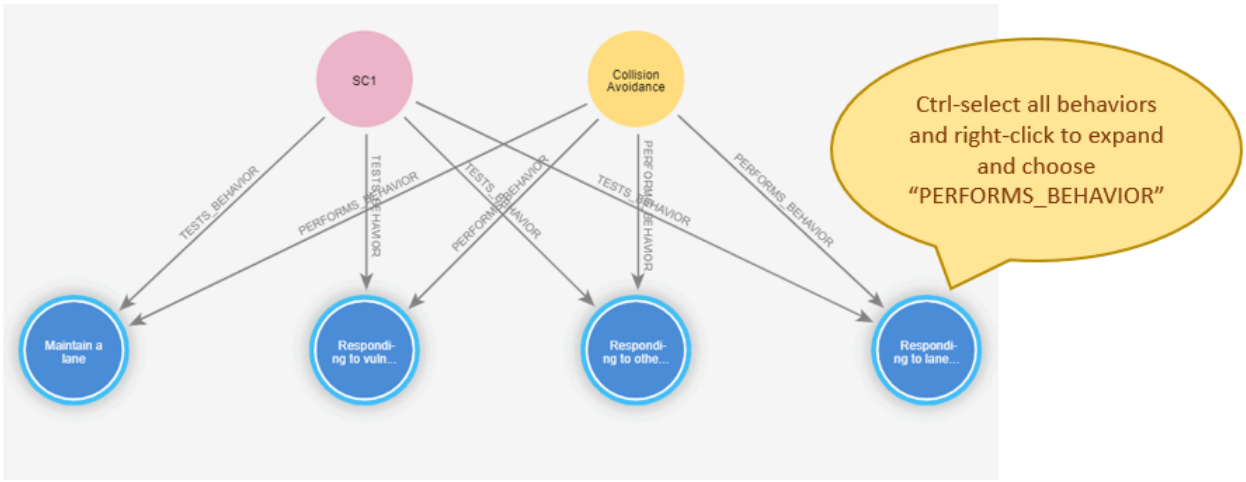
1. **What can influence my scenario?**

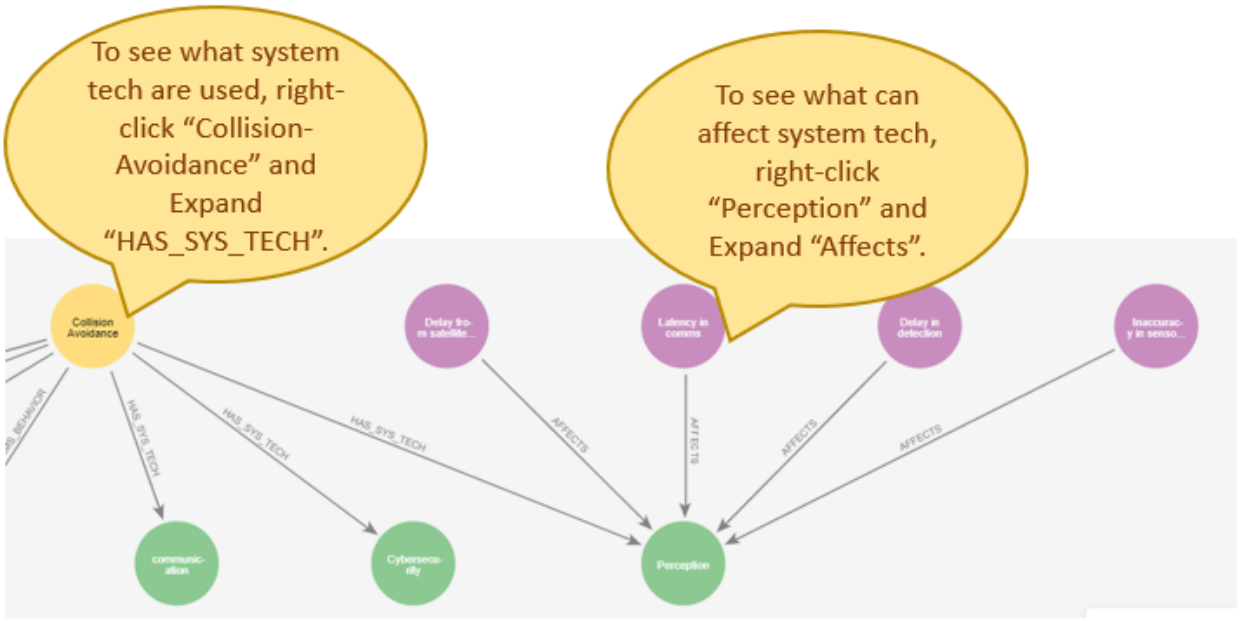
- To answer this question, we start with a scenario.

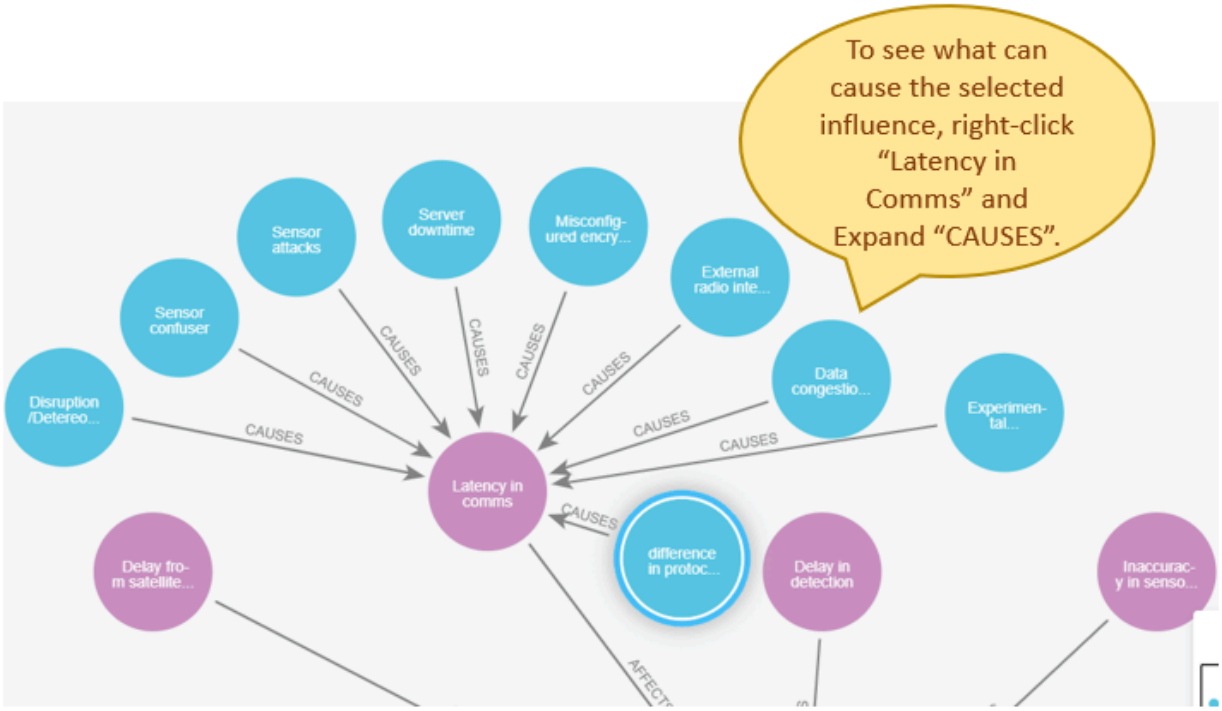

Write “Scenario” in search box
Scenario
SC1


Scenario
SC1
Properties
Neighbors
Relationship
double- click Scenario to get more information about node
Edit
Scenario
traffic sign [150m]
functional
Car is doing a left turn on bifurcated roadway
ID
SC1
logical
Abc
Car is doing a left turn on bifurcated roadway with lane width [2.3 - 3.5m], curve radius [0.6-0.9]km Position traffic sign [0-200m]
scenarioDescription
Straight road from center Dr North where (width = 3.2m) => Get ready for left => Turns left => Detect obstacle in front at X meters => Applies AEB at (X-d) meters => Stops at safe distance d from obstacle => End scenario


right-click
“Scenario” node to
expand and choose
“TESTS_BEHAVIOR”
SC1
TESTS_BEHAVIOR
TESTS_BEHAVIOR
TESTS_BEHAVIOR
TESTS_BEHAVIOR
Maintain a lane
Respondi-ng to vuln...
Respondi-ng to othe...
Respondi-ng to lane...


SC1
Collision Avoidance
Ctrl-select all behaviors
and right-click to expand
and choose
“PERFORMS_BEHAVIOR”
TESTS_BEHAVIOR
PERFORMS_BEHAVIOR
Maintain a lane
Respondi-ng to vuln...
Respondi-ng to othe...
Respondi-ng to lane...





Hence, Latency in comms due to Sensor confusers, disruption/Deterioration in Channels, External radio interference etc are some of the influencing factors that can affect your scenario SC1.

2. What can affect my system technology?(AI, Perception, Comms or Cybersec)

3. What feature is used in the automated vehicle and what system technologies it uses?
4. What other features in the car use Comms? If part of Comms is affected for one feature, will it affect other features?(You can replace Comms with your system technology)
5. What tests can I do to measure the feature metrics?
6. What internal influence can impact performance of my feature, and what system technology embodies that vulnerability?
7. Which features can be impacted, depending on their implementation using system technologies, by a given influence?
8. Which influences can relate to my usage specification?

Below mentioned are some examples for context while exploring the model.

Example 1: Reveal Scenario

Description: One of the tests we are trying to run is the vehicle's performance in a reveal scenario where we have three vehicles(front, middle and ego vehicle) traveling on a highway in a straight line in the same lane, and the front most vehicle suddenly stops. The middle car(non-autonomous) merges in the right lane.

Q. Design an experiment to test **what** affects distance from obstacles.
A. Possible ways to achieve this is looking at internal influences for a system technology that the feature uses(in this case, obstacle avoidance uses comms) like **Latency in Comms** which when increased by simulation to see how distance from the front vehicle is affected.

Example 2: Obstacle Collision Avoidance

Description: We are trying to test the vehicle's performance in a collision avoidance scenario when the ego vehicle encounters a car in the front, and the feature allows the vehicle to apply AEB to avoid collision. **It does not have the capability to swerve around the obstacle yet**. We wish for an ego vehicle to stop at a safe distance from the front vehicle. What are the possible tests one can do to measure this AEB's performance?

Q. Design an experiment to test how **sensor data(Perception)** can affect distance from obstacles.
A. Possible ways to achieve that is looking how inaccurate sensor data affects object detection and eventually affects stopping distance from
Front vehicle.

6. Why Neo4J?

- Neo4J is a graph based database that stores information in **relationships** and **nodes**.Users can load and focus their work in the model, and **look at nodes they are interested in or working with**. E.g. Features, System Technology, ODD etc.

- Neo4J gives a common reference framework for multiple stakeholders like AV developers for answering questions like what experiments are possible to study performance of an AV Feature or safety regulators to access or add their perspective on infrastructure that can affect the AV Feature.
- It makes answering system interaction questions easier as the user can visualize and explore the model easily by a simple right-click and expand. Moreover, it makes querying data easier and faster as compared to classic SQL databases.
- One can explore the node they are interested in and see its relation with any other node present in the model. E.g. **Influences node** and **Usage specification** node might not have direct link but one definitely affects another. E.g. Explore influences for how I build my robo-taxi(usage specification).