U.S. Department of Transportation
Office of the Assistant Secretary for
Research and Technology
Volpe Center

Voices
Virtual Open Innovation
Collaborative Environment for Safety

# VOICES System Integration for Eco Approach and Departure Use Case

## 1 Objective

The main objective of this paper is to lay out what is needed to test a sample use case (the EAD in this paper) in VOICES. The following are the main components of this layout:

- ✓ Use case concept of operation.
- ✓ Test setup.
- ✓ Architecture.
- ✓ Data flow.
- ✓ Test operational process.
- ✓ Test Scenario description.

This layout structure is aimed to drive the Community of Practice discussions, establish a reference architecture, and guide the VOICES developers in building the required TENA modules and Adaptors to construct the simulation data and its flow.

## 2 Cooperative Driving Automation (CDA) Classes of Cooperation

SAE J3216 standardizes how cooperation between vehicles is classified. The classes of cooperation address different capabilities of a vehicle equipped with a cooperative-automated driving system (C-ADS). Table 1 summarizes the SAE J3216 CDA cooperation classes across the different levels of automation defined in SAE J3016.

**Table 1. CDA Classes across Levels of Automation**

| **No Automation** | | **Partial Automation of DDT** | | | **Complete Automation of DDT** | | |
|---|---|---|---|---|---|---|---|
| | | Level 0:<br>No Driving Automation (human does all driving) | Level 1:<br>Driver Assistance (longitudinal or lateral vehicle motion control) | **Level 2:<br>Partial Driving Automation (longitudinal and lateral vehicle motion control)** | **Level 3:<br>Conditional Driving Automation** | **Level 4:<br>High Driving Automation** | **Level 5:<br>Full Driving Automation** |
| **No Cooperative Automation** | | E.g., signage, TCD | Relies on driver to complete DDT and supervise feature performance in real time | | Relies on ADS to complete DDT under defined conditions (fallback condition performance varies between levels) | | |
| **SAE class A: Status Sharing** | **Here I am and what I see** | E.g., brake lights, traffic signal | Potential for improved object and event detection[*] | | Potential for improved object and event detection[**] | | |
| **SAE class B: Intent Sharing** | **This is what I plan to do** | E.g., turn signal, merge | Potential for improved object and event prediction[*] | | Potential for improved object and event prediction[**] | | |
| **SAE class C: Agreement Seeking** | **Let's do this together** | E.g., hand signals, merge | N/A | | C–ADS designed to attain mutual goals through coordinated actions | | |
| **SAE class D: Prescriptive** | **I will do as directed** | E.g., hand signals, lane | N/A | | C–ADS designed to accept and adhere to a command | | |

| | | | | |
|---|---|---|---|---|
| | | assignment by officials | | |

DDT ≡ Dynamic Driving Task; TCD ≡ Traffic Control Device; ADS ≡ Automated Driving System; N/A ≡ Not Available

## 3 Echo Approach and Departure (EAD) System Overview

EAD is intended to optimize traffic throughput through intersections by using cooperative driving systems that integrate data from the host vehicle, other vehicles of interest, and the infrastructure. This is achieved by having different levels of capabilities at both the vehicle and infrastructure levels, and exchanging timely messages among vehicles and between vehicles and the infrastructure. Here are some forms of such a system:

- Systems focus on traffic signal timing optimization, where efficiency of vehicles passing through intersections is improved by adjusting traffic signal timing appropriately to reduce the amount of slow down or stopping at red lights.
- Systems focus on sharing the traffic signal timing with the vehicles so that vehicle speeds could be optimized to reduce the amount of slow down or stopping at red lights.
- Closed-loop systems where traffic signal timing is modified in relation to the traffic coming up to an intersection and calculated arrival time schedule. Both the updated traffic signal timing and arrival time schedule are shared with the vehicles so that vehicle speeds could be optimized to meet the real-time calculated schedule.

## 4 Concept of Operation of EAD System Under Test

The scenario of interest in this document addresses the optimization of traffic throughput at signalized intersections. The scenario is depicted in Figure 1 where a set of three vehicles are approaching a 4-way signalized intersection from south direction (traveling northbound), and another set of four vehicles are approaching the intersection from the east direction (traveling westbound). All vehicles are Class B or higher CDA vehicles. This means that, at a minimum, they can share their current status and their intent (future move). Some non-CDA vehicles can also be introduced in a similar scenario. Having mixed vehicles (CDA and non-CDA) will be useful to demonstrate how the EAD system deals with traffic streams mixed with CDA and non-CDA vehicles, and show the value of such a system in this mix environment.

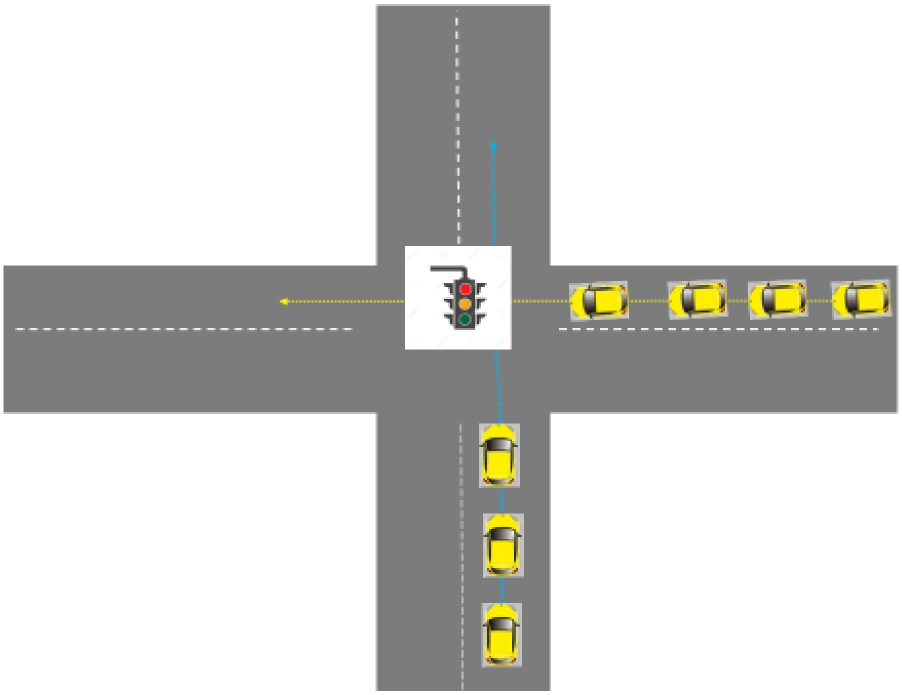
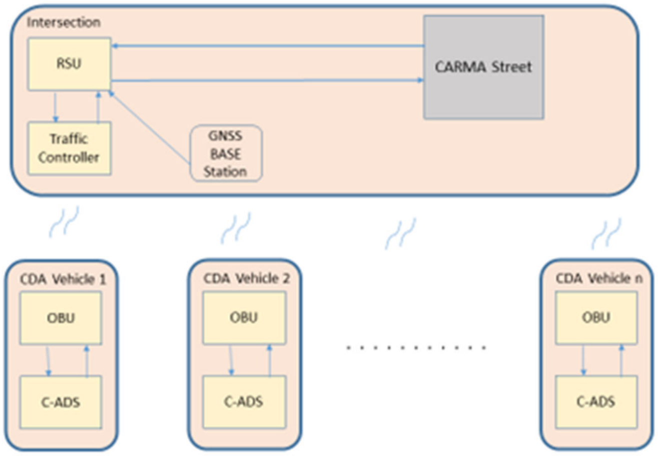


Figure 1. EAD Scenario of Interest

Figure 2 illustrate a simple EAD system architecture. The 4-way intersection is equipped with:

1. Traffic controller controlling signal heads to regulate the approaching traffic on lane level
2. CARMA Streets
3. Roadside Unit (RSU)

The communication between the CARMA Streets and the CDA vehicles are performed wirelessly via the RSU.



OBU ≡ On-Board Unit; GNSS ≡ Global Navigation Satellite System

Figure 2. EAD System Architecture

In this scenario, the EAD system under test performs the following sequence of tasks:

1. CDA vehicles are approaching the intersection and start receiving RSU broadcasted payload, including WSA, MAP, and signal phase and timing (SPAT) messages.
2. CDA vehicles decide if this RSU is of interest by some methods, such as checking if the vehicle position is map-matched on the MAP broadcasted by the RSU.
3. CDA vehicles determine if their driving direction indicates that they are approaching the intersection of interest.
4. If approaching the intersection, CDA vehicles broadcast two types of data: vehicle current status such as position, speed, heading, and vehicle parameters such as vehicle type and some other vehicle model parameters; and vehicle intention (next move), such as current and future driving lanes.
5. RSU receives vehicles messages/data and pass them to CARMA Streets (or any other platform).
6. RSU receives SPAT message/data and passes it to CARMA Streets.
7. RSU passes MAP message to CARMA Streets.
8. CARMA Streets receives three set of data from RSU: one set is originated from the vehicles (status and intention), one is originated from the traffic controller representing the SPAT, and one from the RSU itself (Map data).
9. CARMA Streets uses these three sets of data to perform the following:
    a. Classify vehicles based on their position relative to the intersection entry points.
    b. Calculate earliest entry time to the intersection for each vehicle that is located in a specified zone of intersection approaches.
    c. Calculate a SPAT plan.
10. CARMA Streets sends the SPAT plan to the RSU.
11. RSU passes the SPAT plan calculated by CARMA Streets to the traffic controller.
12. CARMA Streets broadcasts via RSU the updated received SPAT from the traffic controller.
13. CARMA Streets broadcasts via RSU the earliest entry time if the received SPAT is the same as what was originally requested.
14. CDA vehicles use the data received from CARMA Streets via RSU broadcast, along with data from other CDA vehicles, to calculate its predicted speed profile along the remaining distance to the intersection entry point.
15. CDA vehicles update its status, intent, and speed profile and then broadcast it.
16. CDA vehicles start vehicle control to achieve the desired speed profile.
17. Go Step 5.

## 5 Required Test Setup

Figure 3 shows a configuration of CDA vehicles as an example to run a simulation test. In this configuration, a live-virtual-constructive (LVC)[1] simulation environment is represented where yellow vehicles are *constructive* CDA vehicles, the blue vehicle is the *live* CDA vehicle, and the red vehicle is the *virtual* CDA vehicle.

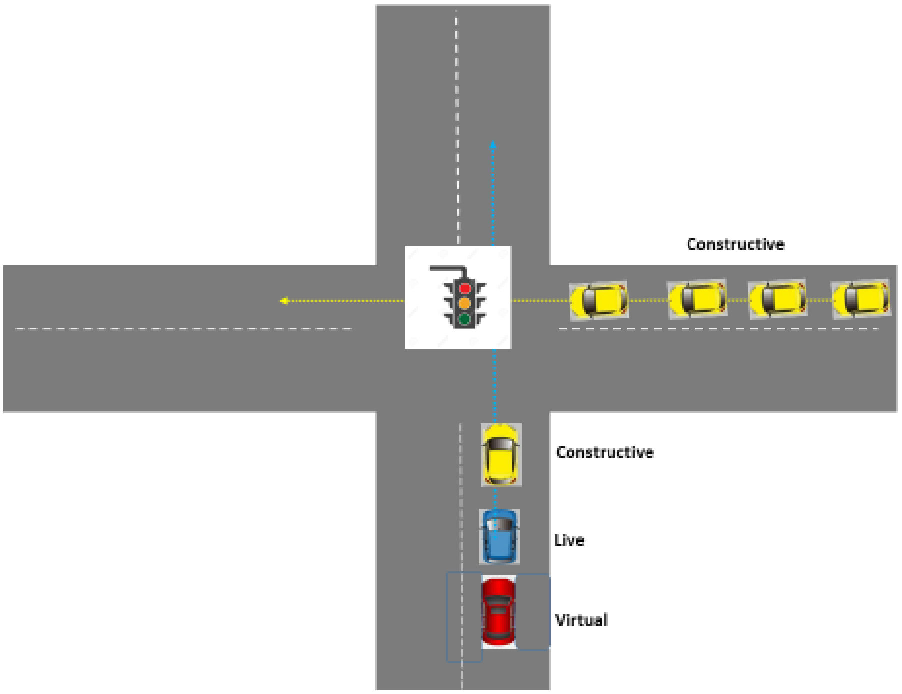


**Figure 3. Configuration of CDA Vehicles for a Simulation Test**

From Steps 1-17 in previous section, one can deduce the needed simulation test setup as follows:

1. One live CDA vehicle and one virtual CDA vehicle.
2. Multiple constructive CDA vehicles depending on the measures of effectiveness of the test case; e.g., percent improvement in cars passing through the intersection per some time duration (an hour for example).
3. Intersection approaches long enough to enable running the test per test design.
4. A map of the test area to place all vehicles on the same reference, and to associate the SPAT with every physical driving lane connected to the intersection.
5. A traffic signal controller and signal heads hardware as live elements of the test. The traffic controller will be generating the SPAT contents.

---

[1] Live refers to a real vehicle system, virtual is just like a virtual simulator with a person in it, and constructive is purely generated by software (created in a simulation package such as CARLA).

6. Traffic signal controller simulator may be used in the simulation experiment as constructive element of the test to drive a live or constructive signal heads.
7. Roadside equipment hardware as a live element in the test to communicate with both the traffic controller as well as with CDA live, constructive, and virtual vehicles.
8. Hardware platform to run CARMA Streets to connect to RSU and traffic signal controller.
9. Live and virtual CDA vehicles need to be equipped with OBUs to communicate with the RSU wirelessly/through network.
10. Live and virtual CDA vehicles need to be equipped with control hardware platform (C-ADS) to run CDA vehicle control algorithms (example CARMA platform), and to supply the OBU unit with the needed status and intent data. OBU and C-ADS can also be combined in one hardware platform.
11. Scenario definition that includes: scenery component such as road description, environment component such as weather, static elements such as traffic sign, dynamic components and their behavior such as vehicles and traffics, operating conditions (i.e., operational design domain), and metrics component including parameters to be calculated, parameters to be visualized, and data elements to be collected.
12. Scenario simulation configuration file to allocate where every actor or element (any actor in the scenario: vehicles, traffic controller, etc.) will be created, containing the simulation parameters such as simulation step time.
13. Base station for GNSS position correction to improve OBU GNSS positioning.
14. Required network connections.

## 6 VOICES System Architecture for EAD Demonstration

Figure 4 captures VOICES system architecture needed to execute the above described scenario.

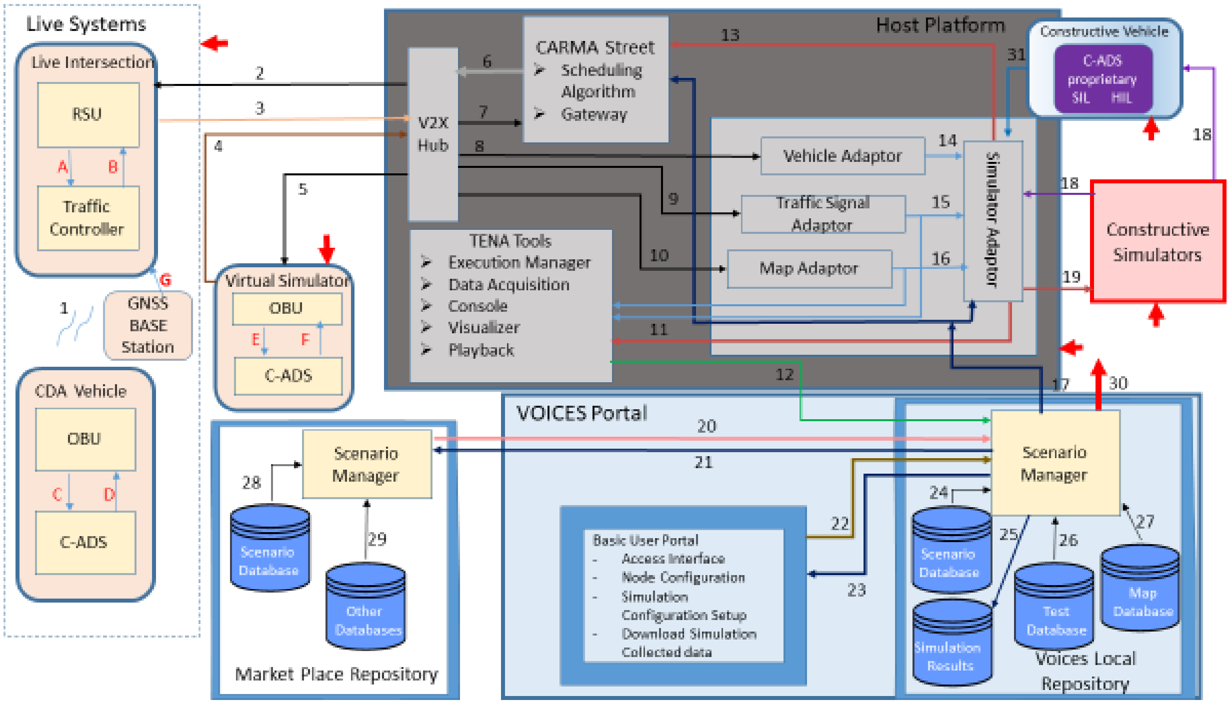


**Figure 4. VOICES System Architecture for Executing EAD Test Scenario**

### 6.1 Functional Block Descriptions

The following is functional description for the basic blocks in the above system architecture:

- **CDA Vehicle**: is live vehicle. In this intended test, it is the CARMA vehicle. It has two main components:
  - ➢ The OBU is the dedicated short-range communication (DSRC) on-board equipment responsible for constructing and transmitting 10 Hz basic safety messages (BSMs) that contain status and intent data of the CARMA vehicle. It also receives other DSRC messages broadcasted by the RSU.
  - ➢ C-ADS is basically performing the behavior calculation and the control execution. In this intended test, it calculates the optimized speed profile and the control algorithm and commands to achieve this desired speed profile.
- **RSU**: is live DSRC road side equipment deployed at the test intersection. It has the following functions:
  - ➢ Store, broadcast, and send MAP message.
  - ➢ Construct SPAT message.
  - ➢ Gateway for
    - ○ BSM messages

    - SPAT message
    - Arrival time schedule.
    - Possibly a gateway to initialize and start messages.
- **Traffic Controller**: is live traffic signal controller. It provides the content needed for the SPAT message.
- **Virtual Simulator**: is a virtual simulator in which its virtual movement follows a 6-dof vehicle dynamic realized model. It is very similar to the live CDA vehicle, except it drives on a virtual road. It has two main components:
    - ➢ The OBU is the DSRC on board equipment responsible for constructing and transmitting 10 Hz BSMs that contain status and intent data of the CARMA vehicle. It also receives other DSRC messages routed to it. The assumption here is that the OBU receives and sends BSMs and other DSRC messages through its wired ports.
    - ➢ C-ADS is basically performing the behavior calculation and the control execution. In this intended test, it calculates the optimized speed profile and the control algorithm and commands to achieve this desired speed profile.

The human in the loop may also be considered as a component in the virtual simulator.

- **GNSS Base Station**: provides Radio Technical Commission for Maritime Services (RTCM) position correction (differential GPS correction).
- **V2X Hub**: is a software module that encodes and decodes J2735 messages such as BSM, MAP, and SPAT and passes them to the intended recipients.
- **TENA Tools**: the list of TENA tools and their descriptions are provided below:
    - ➢ Data Acquisition and Playback: both represented by the TENA Data Collection System (TDCS) that provides both a data collection and data playback application. There are two potential tools for visualization: one is the TENA DataView that helps monitor the data being exchanged between the applications, and the other is the 3D visualization tool SIMIDS.
    - ➢ TENA Execution Manager: every TENA execution utilizes an Execution Manager process to manage group membership. When TENA applications attempt to join or resign from the execution, they are required to perform these requests with the Execution Manager. The Execution Manager is used to introduce publishing and subscribing applications to enable them to make direct connections when appropriate. Execution wide configuration parameters, such as multicast properties and object model definitions, are maintained by the Execution Manager. Multiple Execution Managers can be utilized to provide fault tolerance support.
    - ➢ TDCS: acts as a pure subscriber that stores received stateful distributed object (SDO) events (discovery, update, and destruction) and messages in a database. The tool subscribes to all SDOs and messages that were built for, but can be filtered by the operator before joining an execution. Best match filtering is always enabled so that the most-derived class available will be saved. The Playback tool publishes data stored in the database. By default, start and

  stop times are the minimum/maximum event times collected. Playback will default to all available SDOs and messages, but this may be limited too.
  - TENA DataViewer (TDV): displays Test and Training Enabling Architecture (TENA) data as it is received by the TDCS. TENA DataView relies on the TDCS that logs all transmitted data to an SQLite database and, in turn, TDV reads the database tables in near real-time and displays the data in a readable format. TDV provides real-time filtering capabilities and comma separated file (CSV) exporting of the data.
  - SIMDIS™: is a set of software tools that provide two- and three-dimensional interactive graphical and video display of live and post processed simulation, test, and operational data. SIMDIS has evolved from a Naval Research Laboratory display tool for the output of missile models to a premier government-off-the-shelf (GOTS) product for advanced situational awareness and visual analysis. The DOD's Test Resource Management Center (TRMC) maintains a SIMDIS plugin that subscribes to the specified TENA standard object models (configurable via the Plugin dialog) for display of entities in SIMDIS.
  - TENA Console: a graphical user interface-based event management tool used to evaluate and monitor TENA applications and the network. It provides application diagnostics, network monitoring, and alerts. It also includes the TENA Canary application useful for examining and diagnosing the multicast properties of a TENA execution, including network connectivity between a computer and the TENA Execution and to confirm multicast connectivity between all computers within a TENA Execution.
- CARMA Streets: is a software platform with three main functionalities:
  - Calculates the arrival time schedule for all the CDA vehicles in the test.
  - Calculates the optimized SPAT schedule.
  - Gateway for constructive as well as virtual CDA vehicles status and intent data.
- **Vehicle Adaptor**: converts BSM content data to TENA object (Common Vehicle).
- **Traffic Signal Adaptor**: converts SPAT message content data to TENA object (Common SPAT).
- **Map Adaptor**: converts MAP message content data to TENA object (Common Map).
- **Simulator Adaptor**: has two main functionalities:
  - Constructs required simulator input data from its TENA objects inputs.
  - Converts back simulator outputs to TENA objects.
- **Constructive Simulator**: runs the simulation for the intended test.
- **Constructive vehicle**: is equivalent to the CARMA platform in the live vehicle.
  - C-ADS software/hardware in the loop (SIL/HIL) is basically performing the behavior calculation and the control execution for the constructive CDA vehicle. In this intended test, it calculates the optimized speed profile and the control algorithm and commands to achieve this desired speed profile.
  - It is possible to have its own host vehicle engine/dynamic model and simulator.
- **Scenario Manager**: is responsible of making the test happens according to the simulation specification from the scenario and test files along with simulation setup/configuration. It is

basically the interface between the management domain and the test domain in VOICES architecture. It has the following basic functionalities:

  - Consumes static scenario definition file from the scenario database in accordance with a file specification.
  - Instantiates distributed synthetic test environment (DSTE) images and initializes components.
  - Sets actor behavior from a scenario file.
  - Configures the nodes.
- Basic User Portal: enables:
  - Access interface
  - Node configuration
  - Simulation configuration setup
  - Download of simulation collected data
- **Marketplace Repository**: provides the building blocks of test scenarios for users to build the virtual environment for customized virtualized testing environment. The initial instance of the marketplace provides for the selection of preconfigured scenarios. The final goal will be the full modular marketplace for user customized testing.

More description of VOICES Portal is available in section 6: Operational Process Flow section of this paper.

## 6.2 Data Flow Description

Referring to Figure 4, the various links in the data flow are:

A. Optimized SPAT calculated Schedule Request. The request contains the new value of green, red, and yellow time for each traffic signal phase.
B. SPAT signal.
C.
- BSM content for live CDA vehicle (CARMA Vehicle) and other CDA vehicles created in virtual and constructive simulators. Part 2 of the BSMs is filled with intent data.
- Scheduling input (arrival to intersection schedule) originated in CARMA Streets.
- Updated SPAT originated from traffic controller and previously requested by CARMA Streets.
- MAP message either originated in the VOICES Portal or stored in the RSU message.
- Initialize and Start originated from the VOICES Portal (User Portal → Scenario Manager)

D. CARMA vehicle status and intent data.
E.
- BSM content for Virtual vehicle, Live CARMA vehicle, and other CDA vehicles created in constructive simulators. Part 2 of the BSMs is filled with intent data.
- Scheduling input originated from CARMA Streets.
- Updated SPAT originated from Traffic Controller and previously requested by CARMA Streets.

- MAP message either originated in the VOICES Portal or stored in the RSU message.
- Initialize and Start originated from the VOICES Portal (User Portal → Scenario Manager).

F. Virtual vehicle status and intent data.

G. RTCM correction data.

1.

- Over the air BSM’s messages or content for other CDA vehicles created in Virtual and Constructive Simulators. Part 2 of the BSM’s are filled with intent data. RSU→OBU
- Over the air scheduling input originated in CARMA Streets. RSU→OBU
- Over the air updated SPAT message originated from Traffic Controller and previously requested by CARMA Streets. RSU→OBU
- Over the air MAP message either originated in the VOICES Portal or stored in the RSU message. RSU→OBU
- Over the air Initialize and Start originated in the VOICES Portal (User Portal → Scenario Manager).
- Over the air RTCM position correction message originated in GNSS base station → RSU→OBU.
- Over the air BSM message of the live CDA vehicle (CARMA vehicle). Part 2 of the BSM’s are filled with intent data. OBU→RSU

2.

- Arrival to Intersection Schedule message.
- BSM’s messages for other CDA vehicles created in Virtual and Constructive Simulators. Part 2 of the BSM’s are filled with intent data.
- Requested SPAT schedule message content originated by CARMA Streets.
- Map message if originated by VOICES Portal.

3.

- Updated SPAT message originated from Traffic Controller and previously requested by CARMA Streets.
- MAP message either originated if stored in the RSU message.
- BSM of the live CDA vehicle (CARMA vehicle). Part 2 of the BSMs is filled with intent data.

4. BSM of the Virtual CDA vehicle. Part 2 of the BSMs is filled with intent data.

5.

- Arrival to Intersection Schedule message.
- BSM’s messages for live CDA vehicle (CARMA Vehicle) and other CDA vehicles created in Constructive Simulators. Part 2 of the BSM’s are filled with intent data.
- Optimized requested SPAT schedule originated by CARMA Streets.
- MAP message either originated in the VOICES Portal or stored in the RSU message

6.

- Arrival to Intersection Schedule message content.
- BSM’s messages content for CARMA Vehicle and other CDA vehicles created in Virtual and Constructive Simulators.

   - Requested SPAT schedule message content originated by CARMA Streets.
   - Map message content if originated by VOICES Portal.
7. 
   - Updated SPAT message content originated from Traffic Controller and previously requested by CARMA Streets.
   - MAP message content if stored in the RSU message.
   - BSM message content of the live CDA vehicle (CARMA vehicle).
   - BSM message content of the Virtual CDA vehicle.
8. 
   - BSM message content of the live CDA vehicle (CARMA vehicle).
   - BSM message content of the virtual CDA vehicle.
9. Updated SPAT message content originated from traffic controller and previously requested by CARMA Streets.
10. MAP message content if stored in the RSU message.
11. Common simulated vehicles (Live, Virtual and Constructive): TENA object.
12. Common simulation result: TENA object.
13. Common simulated vehicles (LVC): TENA object.
14. 
   - Common CDA vehicle (CARMA vehicle): TENA object
   - Common virtual CDA vehicle: TENA object.
15. Common SPAT: TENA object.
16. Common MAP: TENA object.
17. Common Scenario: TENA object.
18. Simulated vehicles (LVC) data.
19. 
   - Live and virtual vehicle data
   - SPAT data
   - Map data
   - Scenario data
20. Scenario data file or some other files originated from marketplace like Safety Pool.
21. Marketplace (such as Safety Pool) Scenario Manager login and launch.
22. Voices Scenario Manager launch, and test start after receiving test ready.
23. Test is ready.
24. Scenario description file.
25. Test results.
26. Test description file.
27. Map file.
28. Scenario description file from marketplace.
29. Test description file, Map file, or some other files.

30. Over the air Initialize and Start originated in the VOICES Portal (User Portal → Scenario Manager).
31. Over the air Initialize and Start originated in the VOICES Portal (User Portal → Scenario Manager).

## 7 Operational Process Flow Diagram

The VOICES Portal provides the ability for users to select virtualization scenarios from a marketplace repository. The repository will provide the building blocks of test scenarios for users to build the virtual environment for customized virtualized testing environment. The initial instance of the marketplace will provide for the selection of pre-configured scenarios. The final goal will be the full modular marketplace for user customized testing.

There will be three types of users in the marketplace, as shown in Figure **5**:

1. Test users who will have the ability to select components and configure variables within the components, or fully defined scenario. These users will also be able to run the simulation once all components are selected and configured.
2. Test contributors who will be able to submit new modules, scenarios, and test components for evaluation and inclusion in the marketplace. The modules and components will be evaluated by a board of administrators before inclusion for functionality and security prior to being available on the marketplace.

Administrators who will have the ability to add/remove/modify modules and components, participate in evaluation boards, remove components, and perform functions for adding, removing, elevating, or demoting privileges for other users and contributors.

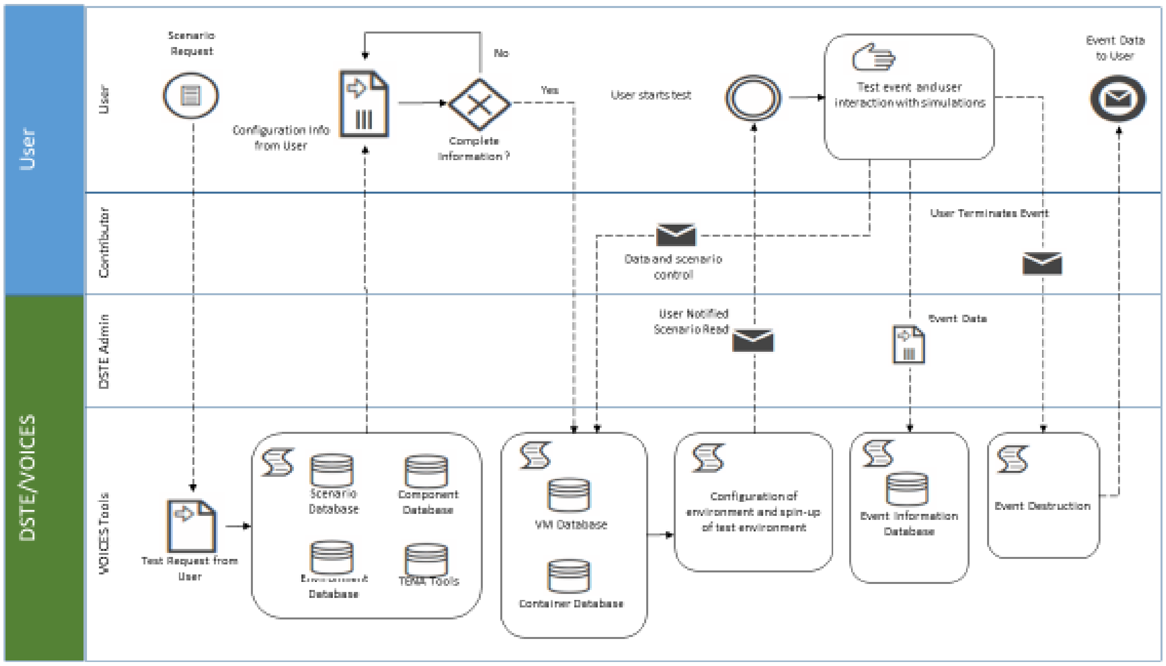


Figure 5. Marketplace Users

Figure 6 illustrates the operational process flow. The viewer walks through the User setup by being able to select the options and scenarios that are then passed to the scenario manager.

The scenario manager then passes the configuration information to the TENA Execution Manager for all the simulations, constructive components, and live components to subscribe and get the information. The User portal then provides the visualization of what is going on in the various simulations, data streams, and visualization tools. The Container environment holds all the Virtual Machines (VMs) and tools in a workspace dedicated to that one event, and is destroyed once the event/test is closed by the user and the scenario manager.

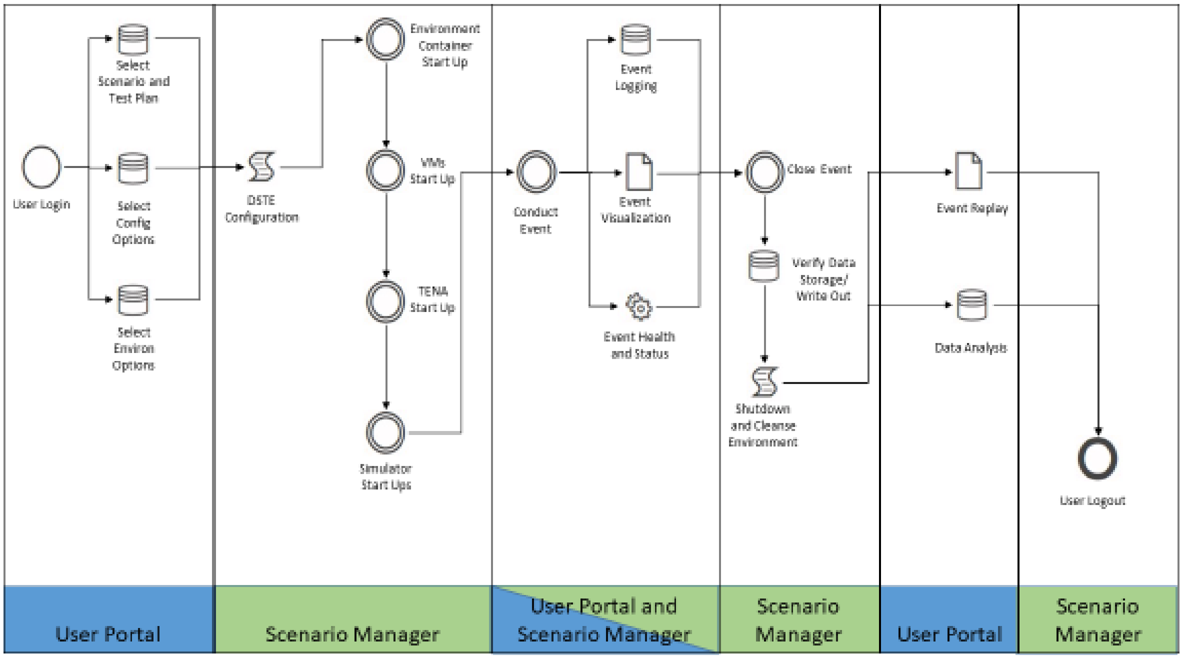


**Figure 6. Operational Process Flow**

## 8 VOICES Capabilities Demonstrated in this Use Case Scenario

The following capabilities (highlighted in yellow) from the VOICES Concept of Operations and Capabilities document are targeted to be demonstrated with this use case simulation:

1. Construct a virtual environment that resembles a real-world driving environment:
   a. Integrating multiple simulation platforms with different semantics, and regulating complex interactions between diverse simulation models.
   b. Simulation platforms shall be interoperable.
   c. Selected simulation tools shall be capable of creating (for example):
      i. On road static and moving on road objects such as vulnerable road users, vehicles, lane marking, traffic signs, traffic lights, stop lines, curbs, work zone structure, etc.
      ii. Different types of roads, road attributes and road conditions
      iii. Moving objects dynamic models
      iv. Different weather conditions
      v. Road surrounding building structure
      vi. Subject vehicle engine and dynamics
      vii. Event, etc.
2. Protect the intellectual property (black box of individual systems) and ensure data privacy and test confidentiality of individual or group of testers.
3. Be secure.

4. Be scalable.
5. Be a connected system
   a. Be accessed remotely and receive data wirelessly.
   b. Integrate cloud connectivity and V2X communication.
6. Include a replay capability.
7. Include a capability to visualize data of interest.
8. Include a mechanism to feed the scenario database with a new or modified scenario.
9. Combine LVC testing. It shall:
   a. Be able to run different simulation modules at different simulation time steps. This is a simulation only requirement.
   b. Synchronize the LVC data.
   c. Have the required interfaces to exchange data with different LVC modules.
   d. Any simulations that are needed for an LVC test shall be able to run one to one with wall clock time and must not fall behind in time.
10. Set up and execute integrated tests from different users. The architecture shall allow exchanging data between different testing nodes.
11. Implement a mechanism to time synchronize the data coming from different sources.
12. Run a stressed simulation scenario without any significant intra delay. This delay shall not scale up significantly as the number of nodes increases. Example of stressed simulation scenario will include different type of simulators, different users, complex test scenario (high number of actors in complex environment), and integration of different live elements.
13. Estimate with sufficient accuracy the latency in the data at different junctions in the data flow path.
14. Report the estimated latency to the SIL/HIL platform.
15. Accommodate throughput and processing to handle high data flow and multiple tests.
16. Host multiple tests at the same time. The architecture shall allow for executing multiple tests concurrently.
17. Execute stored and runtime scenarios from an integrated use case database.
   a. VOICES architecture shall enable simulated scenarios or some real-world scenarios to be specified or selected via user interface.
   b. Allow access to the database without any significant delay.
   c. Provide the scenario data to the proprietary model via API and without any significant delay.
   d. There may be room to change the motion data of other vehicles impacted by the disturbance of the HV motion.
18. Process raw and processed collected data. To handle the raw data, VOICES should have the needed framework to integrate a marketplace repository that includes a default AI-based perception module. The perception module is pluggable and therefore can be replaced with the user module if needed. The perception module shall be able to detect, track, classify,

and characterize different types of vehicles, pedestrians, cyclists, road attributes, traffic signs, road signs, traffic lights, free space, etc.

19. Construct a sensing environment resembling sensors mounted on the vehicle and have pluggable sensor models for different types, including radar, lidar, and video camera.
20. Provide localization and electronic horizon (Look Ahead) data.
    a. Have access to different levels and types of map databases.
    b. Have models for different levels of GNSS receivers.
    c. Have access to point cloud maps for LIDAR localization.
21. Integrate the SIL and provide the needed API.
22. Run the test with system HIL and provide the needed interface and API.
23. Integrate and emulate the different V2X messages in SAE J2735.
24. Provide the needed interface for two-way data exchange with real devices such as traffic light controller, RSUs, and OBUs.
25. Store the testing results in a format configured by the tester.
26. VOICES test Environment should be controllable for test repeatability.
27. Integrate CARMA ecosystem capabilities into VOICES.

## 9 Architecture Flexibility

- The architecture is flexible to enable using different origins of the stored map database. Example: stored in RSU versus stored in Voices Marketplace Map database.
- The architecture is a framework for positioning correction if needed.
- Constructive simulators such as SUMO and CARLA can be hosted within the host platform or hosted within a remote connected node.
- Constructive vehicle SIL/HIL and host vehicle Engine/Dynamic model and Simulator can be hosted within the host platform or hosted within a remote connected node.
- The architecture is flexible to enable using VOICES local scenario database as well as using other scenario databases such as Safety Pool scenario database.
- The architecture enables hosting traffic signal/SPAT simulator within the host platform.
- The architecture allows testers to replace CARMA platform and/or CARMA Streets with their own platforms.

## 10 Scenario High-Level Description File Example

Here is a sample of Scenario Description following SDL Level 1 used in Safety Pool Scenario Database. This is meant to show how one can use an existing framework to describe a simulation scenario for this use case. Figure 7 is roughly a repeat of Figure 1. Level has more detailed description with parameters range values and event triggering. This will be discussed in more details in the Community of Practice (CoP) Working Group (WG) meetings.

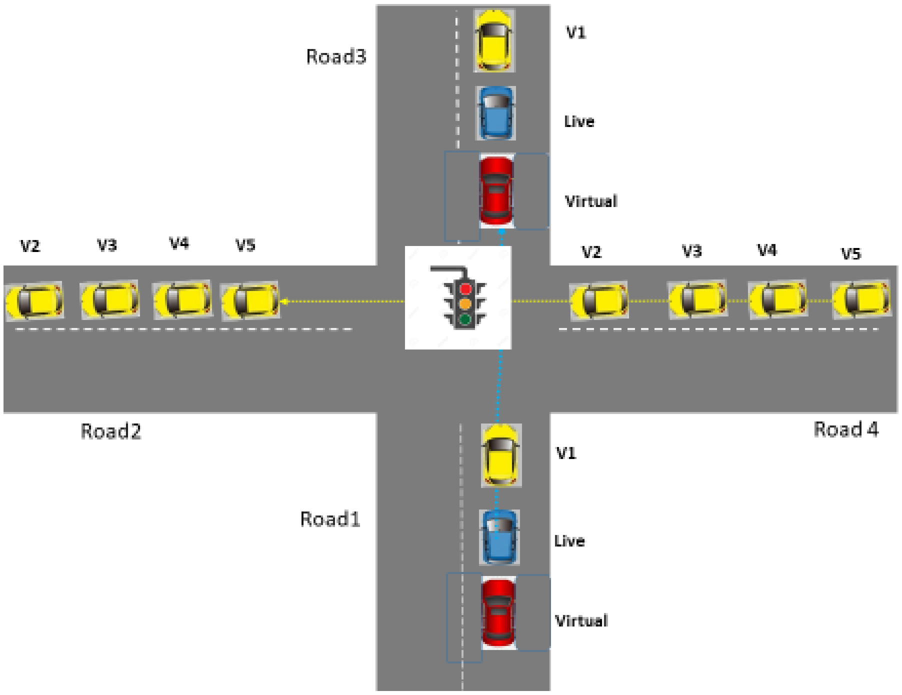


Figure 7. Scenario Depiction

**SCENERY ELEMENTS:**
**Junctions:**
[Junction 1] is a [4Way-Junction] with connections to [Road 1, Road 2, Road3, Road4]
AND with connection control [Traffic Light]
Connection from [Road 1] to [Road 3] is [Straight]
Connection from [Road 4] to [Road 2] is [Straight]
**Roads:**
[Road 2, Road 3, Road 4, Road 5, Road 6] are [Distributor roads] with [Straight] geometries
**DYNAMIC ELEMENTS:**
INITIAL: Vehicle [V1] on [Road 1] AND vehicle [Live] on [Road 1] AND vehicle [Virtual] on [Road 1] AND vehicle [V2] on [Road 4] AND vehicle [V3] on [Road 4] AND vehicle [V4] on [Road 4] AND vehicle [V5] on [Road 4]
WHEN [Live] is [Going ahead] on [Road 1] AND [far] from [Junction 1]:
[Virtual] [Drives towards] [Live] with [Accelerating] speed at [Rear] position
[V1] [Drives away] from [Live] with [Accelerating] speed at [front] position
[V2] [Drives toward] [Live] with [Accelerating] speed at [Front Right] position
[V3] [Drives towards] [Live] with [Accelerating] speed at [Front Right] position
[V4] [Drives towards] [Live] with [Accelerating] speed at [Front Right] position
[V5] [Drives towards] [Live] with [Accelerating] speed at [Front Right] position
END: [Vehicle [V1] on [Road 3] AND vehicle [Live] on [Road 3] AND vehicle [Virtual] on [Road 3] AND vehicle [V2] on [Road 2] AND vehicle [V3] on [Road 2] AND vehicle [V4] on [Road 2] AND vehicle [V5] on [Road 2]
**ENVIRONMENT ELEMENTS:**
DO: Environment [ENV1] as:
Wind [Moderate breeze]
Clouds [Light]
Particulate condition [None]
Rainfall [None]
Time [Day]
Illumination [Daylight]