Following the first requirement refinement meeting from October 26th, the meetings from November 16th and 20th are documented in this protocol.

Based on the proposed concepts, a rough roadmap for the development has been discussed:

1. Agree on the architectural concept (Which interfaces are needed?)
2. Consolidation of interface drafts and development of "minimal" interface for an easy simulation
3. Implementation of logic components -> Ideally we can share the workload by splitting the implementation into work packages
4. Execution of an easy simulation
5. Iterative extension of the functionality

Discussion on current architecture draft:

- Not all scenario description languages need to use actions
- Structural units for the world: map, traffic, vehiclecontroller, ...
- Interfaces should not contain convenience functions (e.g. GetDistanceToObject is very specific and should be part of a map interface)
- Where will conditions be evaluated? ScenarioInterface allows a uniform computation across simulators. The simulation interface allows to account for different calculation methods
- Minimal set, needed for the a first working version: entities and road network

Demonstration of a C++ scenario editor by Mercedes: passive cut-in scenario.

Discussion on the shown “Abstract Environment API” consisting of:

- EnvironmentInterface
- ControllerInterface
- MapInterface
- SimulationInterface
- ScenarioInterface

Comparison to the Intech proposal of ScenarioDirectorInterce and SimulationInterface.

- The current architecture of openPASS can fulfill the requirements of the EnvironmentInterface as
well as of the MapInterface.

- The ControllerInterface does not completely comply with the current openPASS framework, as openPASS relies on a distributed system architecture.
- The two proposals are roughly in line. The practical difficulties should be analyzed in a prototypic implementation inclined towards enabling a passive cut-in use case.

BMW will internally clarify the license related topics and the possibility to organize the API development within the sim@openPASS project. Daimler will internally clarify how to possibly publish the already implemented first draft of the API as a basis for further developments.