Formal Design and Validation of an Automatic Train Operation Control System

Arturo Amendola[1], Lorenzo Barruffo[1], Marco Bozzano[2], Alessandro Cimatti[2], Salvatore De Simone[1], Eugenio Fedeli[1], Artem Gabbasov[2], Domenico Ernesto Garrubba[1], Massimiliano Girardi[2], Diana Serra[1], Roberto Tiella[2], and Gianni Zampedri[2]

[1] Rete Ferroviaria Italiana, Osmannoro, Italy
[2] Fondazione Bruno Kessler, Trento, Italy

Introduction …

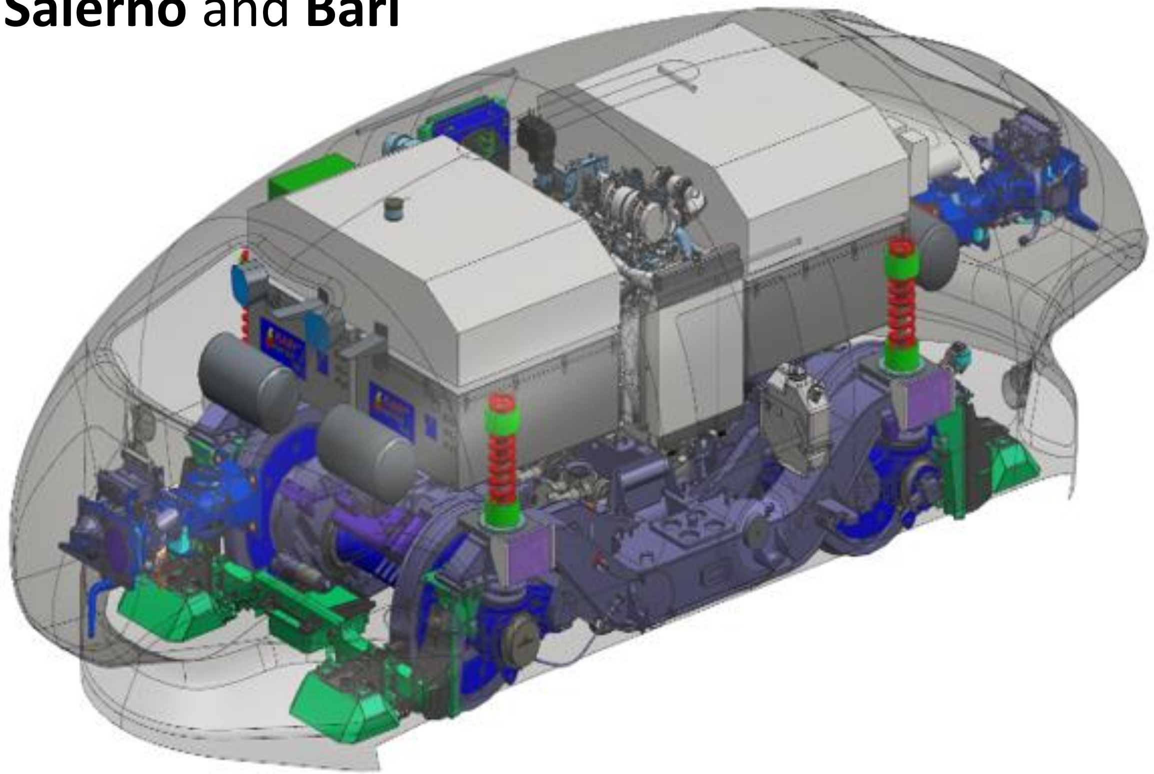
Introduction

- Automatic train operation (**ATO**)
- Motivation
  - Enhance the Grade of Automation (GoA) in train operations in high-speed lines
    - GoA0: absence of automation
    - …
    - **GoA4**: fully automated train control and management (no staff on board)
  - Optimize the driving performances
    - Energy consumption
    - Comfort
  - Applications
    - Infrastructure monitoring
    - Passenger transportation

ATO Project

- Industrial project
  - Led by **Rete Ferroviaria Italiana** (RFI)
  - Contractor **Fondazione Bruno Kessler** (**FBK**): design of ATO control system, implementation of a subset of ATO components, system integration
    - 4 persons, about 126MM total and >4 years timespan
  - Other contributors: **Universities of Naples**, **Salerno** and **Bari**

- Objective of the project
  - Develop a **GoA4 ATO** operating on a prototype light-vehicle, equipped for infrastructure monitoring, running on an ERTMS/ETCS Italian high-speed line



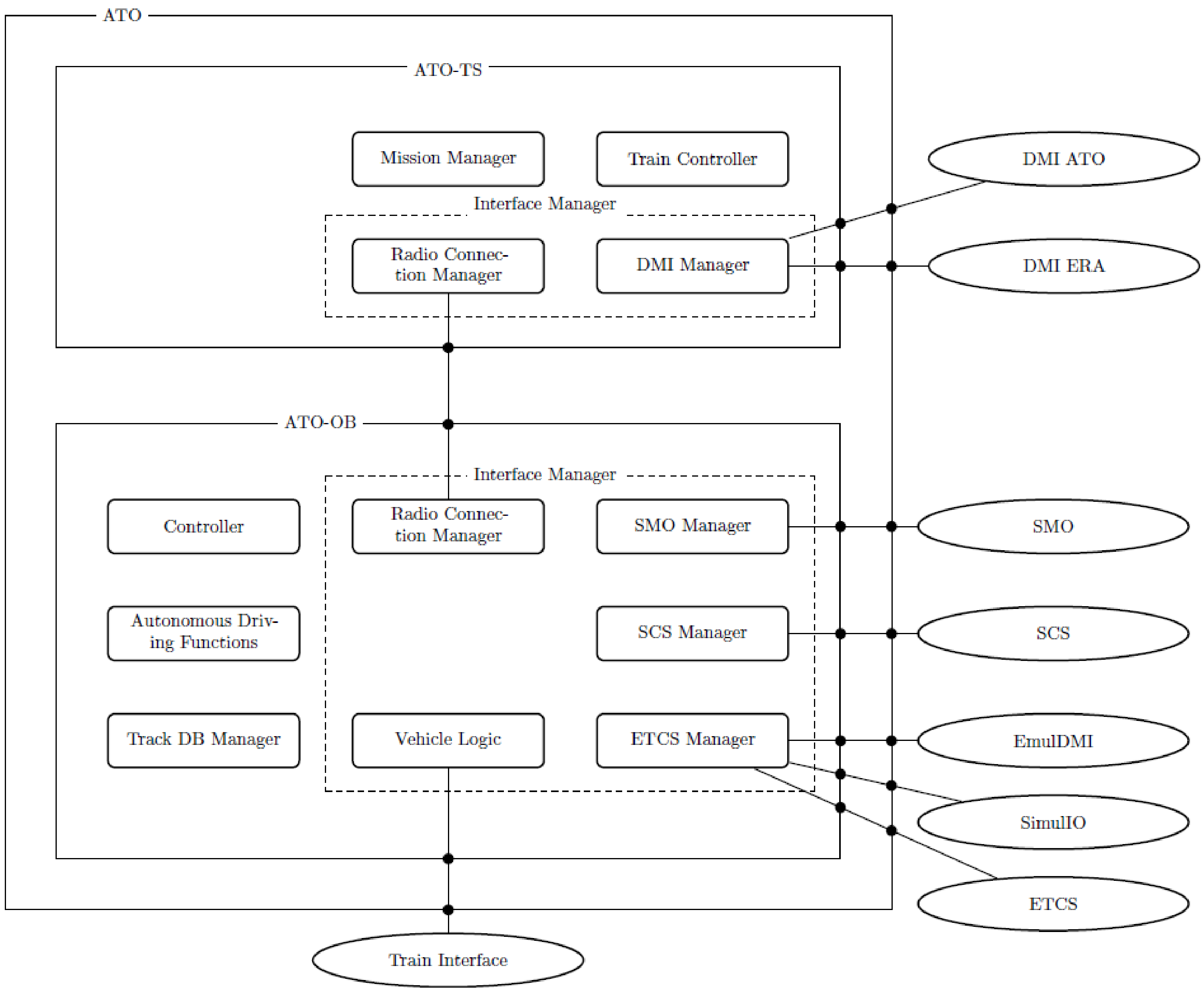
RFI RETE FERROVIARIA ITALIANA GRUPPO FERROVIE DELLO STATO ITALIANE

FONDAZIONE BRUNO KESSLER

The ATO System …

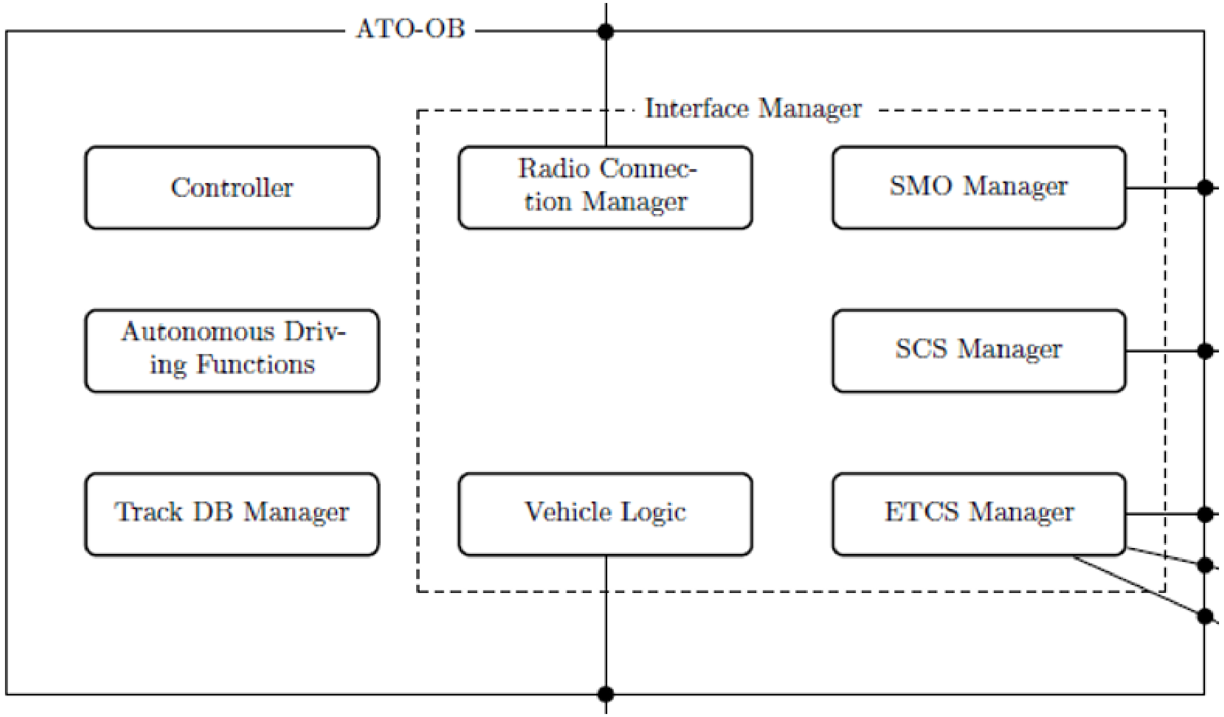
The ATO System

- Two cooperating sub-systems
  - ATO Track Side (**ATO-TS**)
    - Collects and forwards data on trains, tracks and timetables
  - ATO On Board (**ATO-OB**)
    - Controls and drives the train



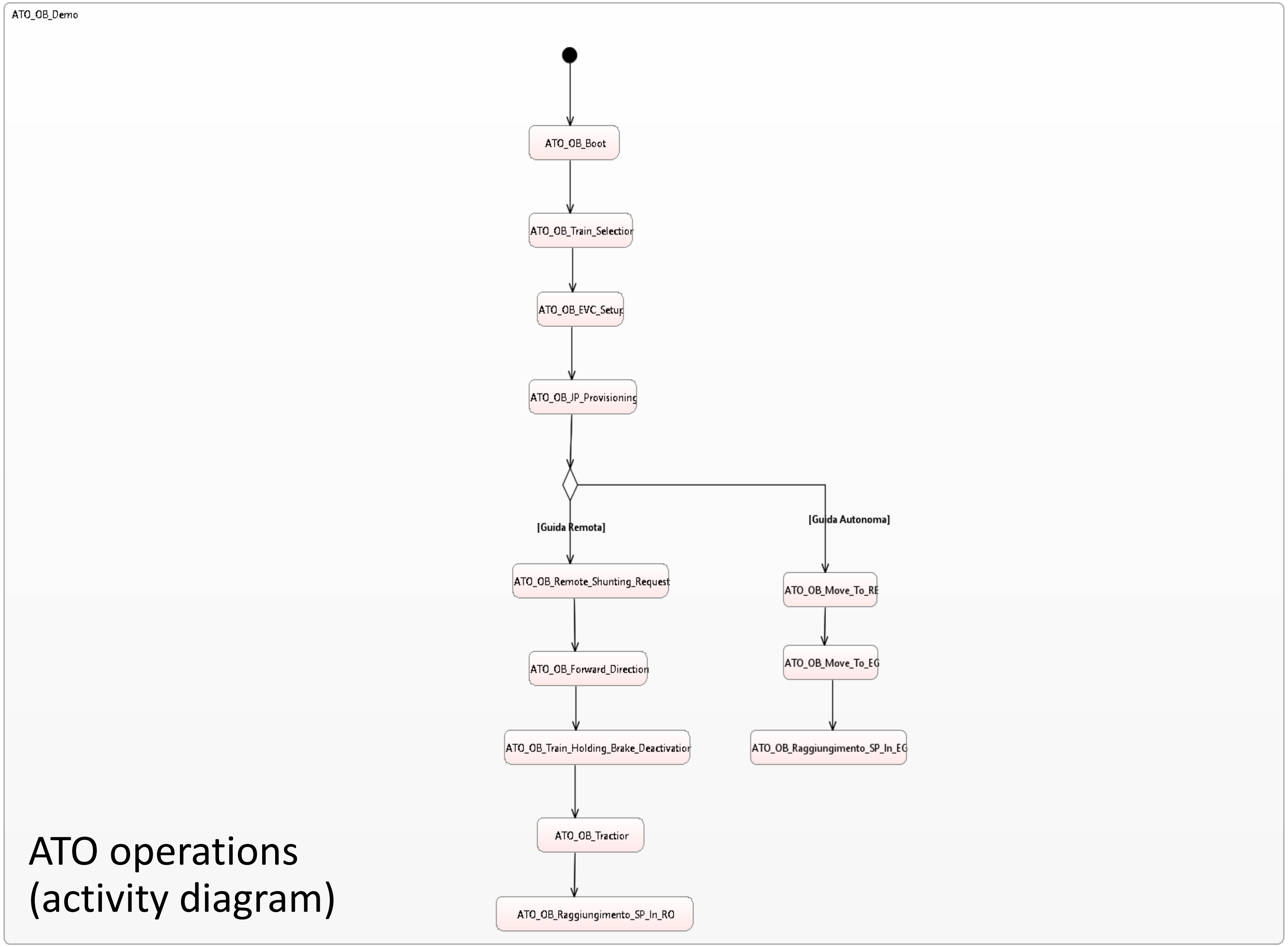
ATO-OB

- ATO-OB main components
  - **Controller**: implements ATO-OB functional state machine
  - **Interface Manager**: interfaces ATO with other modules
  - **SCS** (Supervision and Control System)
  - **SMO** (Speed Monitoring and Odometry)
  - **TIU** (Train Interface Unit)
  - **Track DB Manager**: train localization on the line, journey validation
  - **Autonomous Driving Functions** (**ADF**): based on track and journey profile data, generates an optimal speed profile, brake and traction commands



ATO-OB

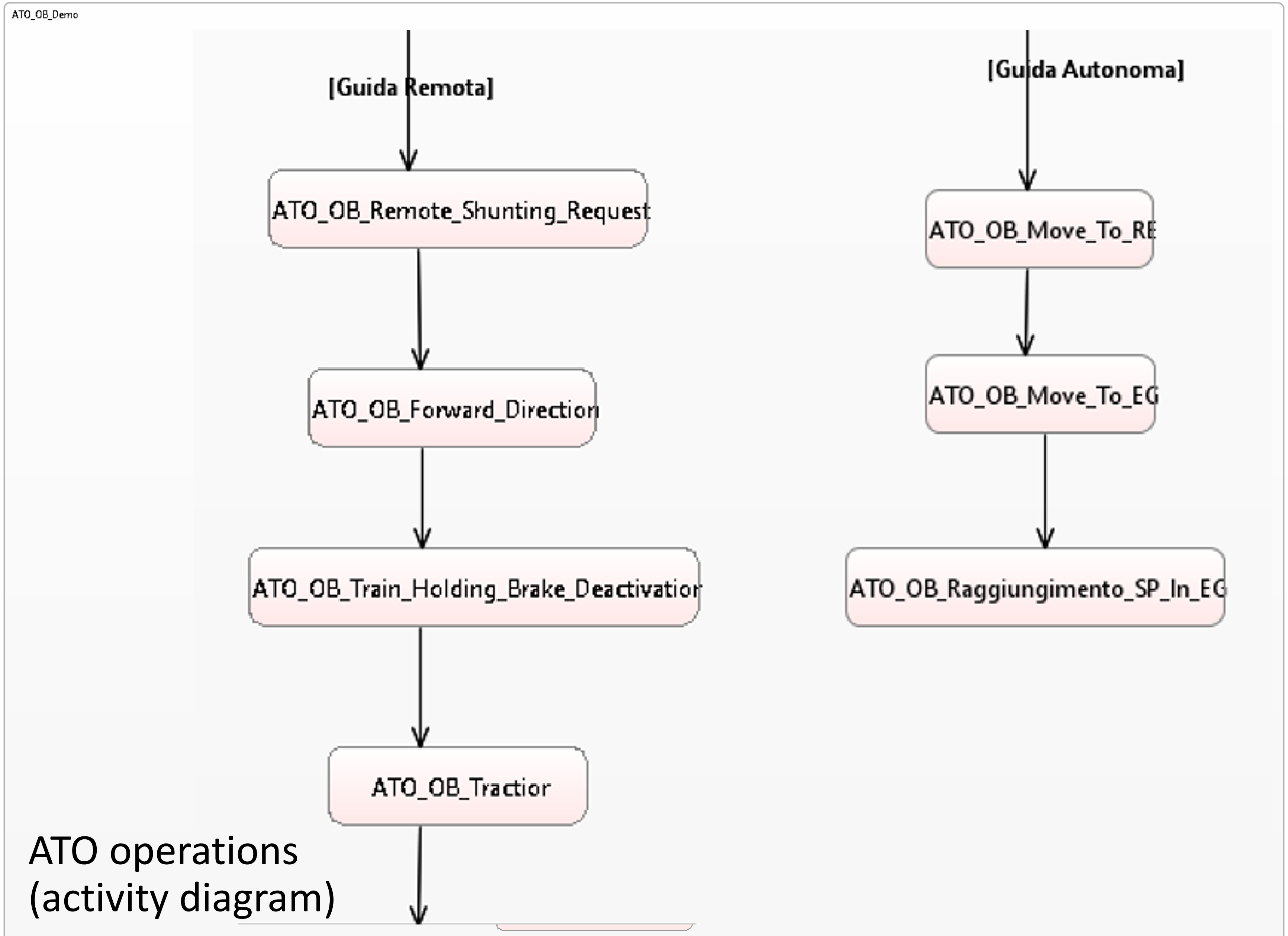
- Vehicle functionality
    - **remote driving**
    - **autonomous driving**
    - **vehicle rescue**
    - **…**



ATO operations
(activity diagram)

ATO-OB

- Vehicle functionality
  - **remote driving**
  - **autonomous driving**
  - **vehicle rescue**
  - **…**



ATO operations (activity diagram)

Challenges …

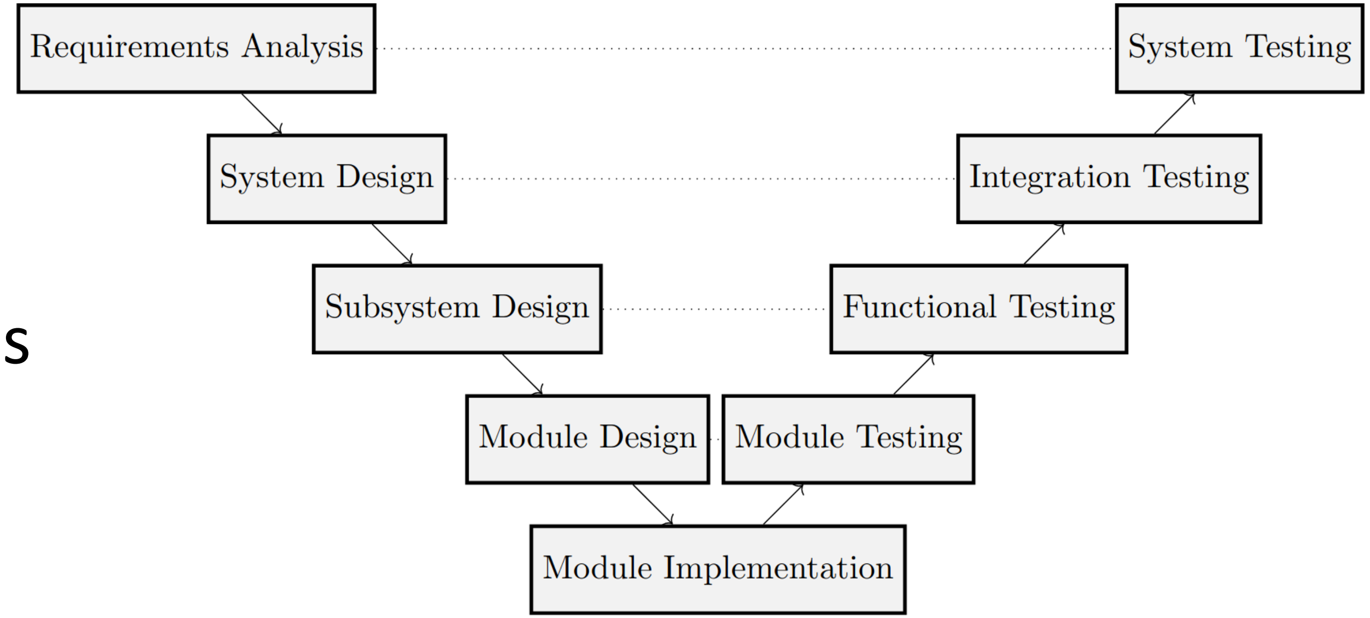
Challenges

- Complexity of the system and requirements
  - Distributed system, with several sub-components
  - Need to customize requirements w.r.t. to the standard UNISIG subset
  - Functional, safety and performance aspects
  - SIL3/4 requirements apply
- Need to support changes and evolution
  - Evolving set of requirements
- Heterogeneous system:
  - SW controllers
  - Components interacting with the HW
  - HW components with continuous dynamics
- Distributed development teams

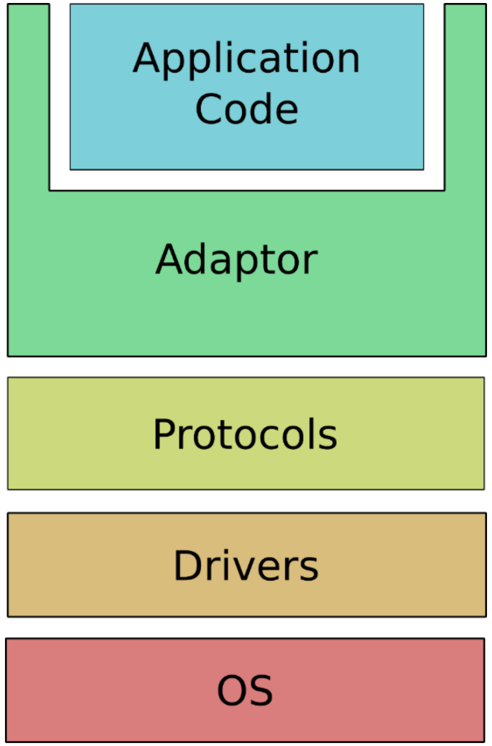
Formal Development Lifecycle …

Formal Development of ATO

- Model-based design
- Using heterogeneous design tools and languages
  - SCADE Suite and Architect, Simulink, C
  - SCADE offers certified code generation capabilities and compliance with SIL3/4
- V-model development process



Architecture of a generic ATO Process



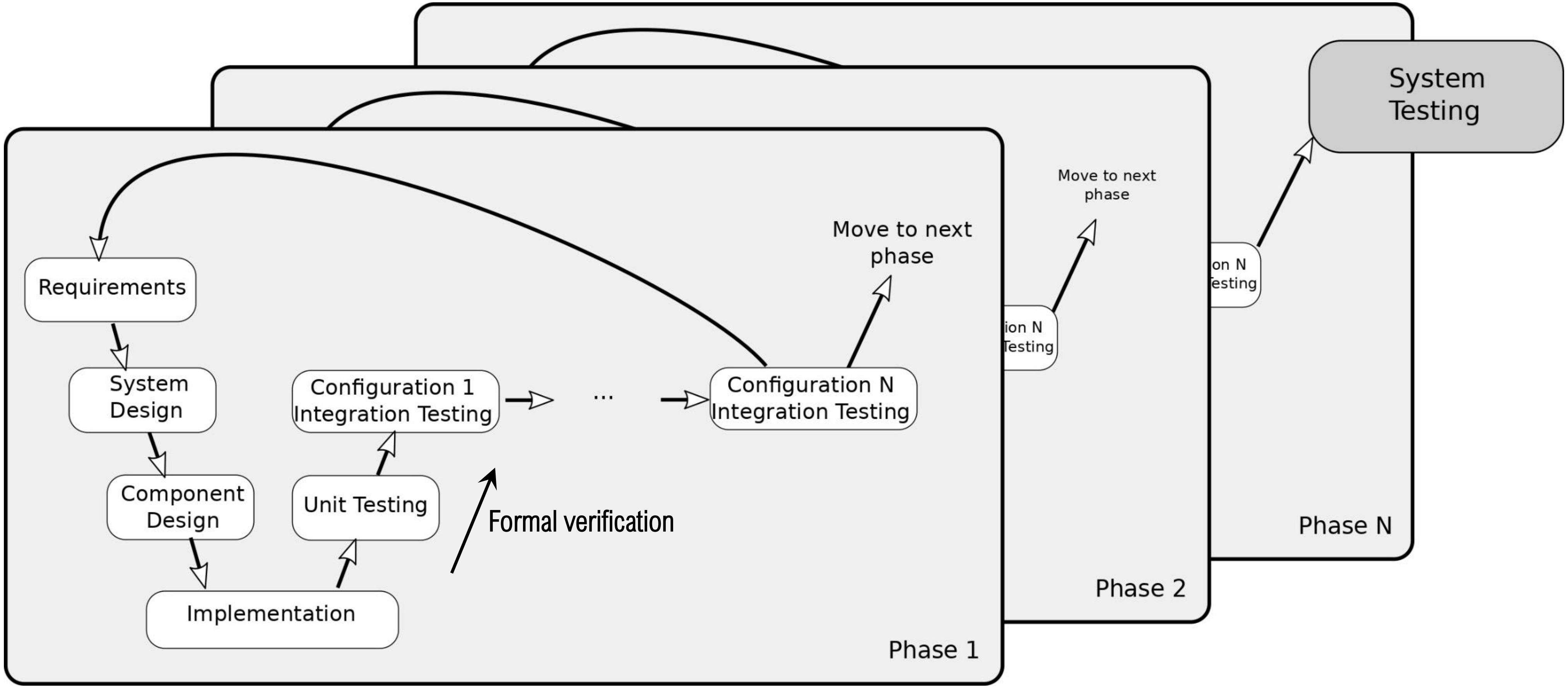
- Layered architecture
  - **Application Code** (pure C-function; protocol-independent abstract data)
  - **Adaptor**: routing data to the layers below
  - **Protocols**: encoding/decoding protocol data <-> abstract data
  - **Drivers**: handling communication with devices
  - **OS**: providing scheduling functionality and access to devices

Development Lifecycle

- Incremental development according to phases and configurations
- **Phases** describe functionality to be implemented
  - **Phase 1**: Remote Driving
  - **Phase 2**: Autonomous driving
  - **Phase 3**: Departure and return from/to the maintenance facility
- **Configurations** describe the deployment/ testing environment
  - **Configuration 1**: integration in SCADE development environment
    - Application code layer only
  - **Configuration 2**: integration on a Desktop Host PC
    - Adds adaptor and protocol layers
  - **Configuration 3**: integration on target HW (HIL: Hardware In the Loop)
    - Adds device and OS layers
  - **Configuration 4**: integration between target HW and external systems

Development Lifecycle

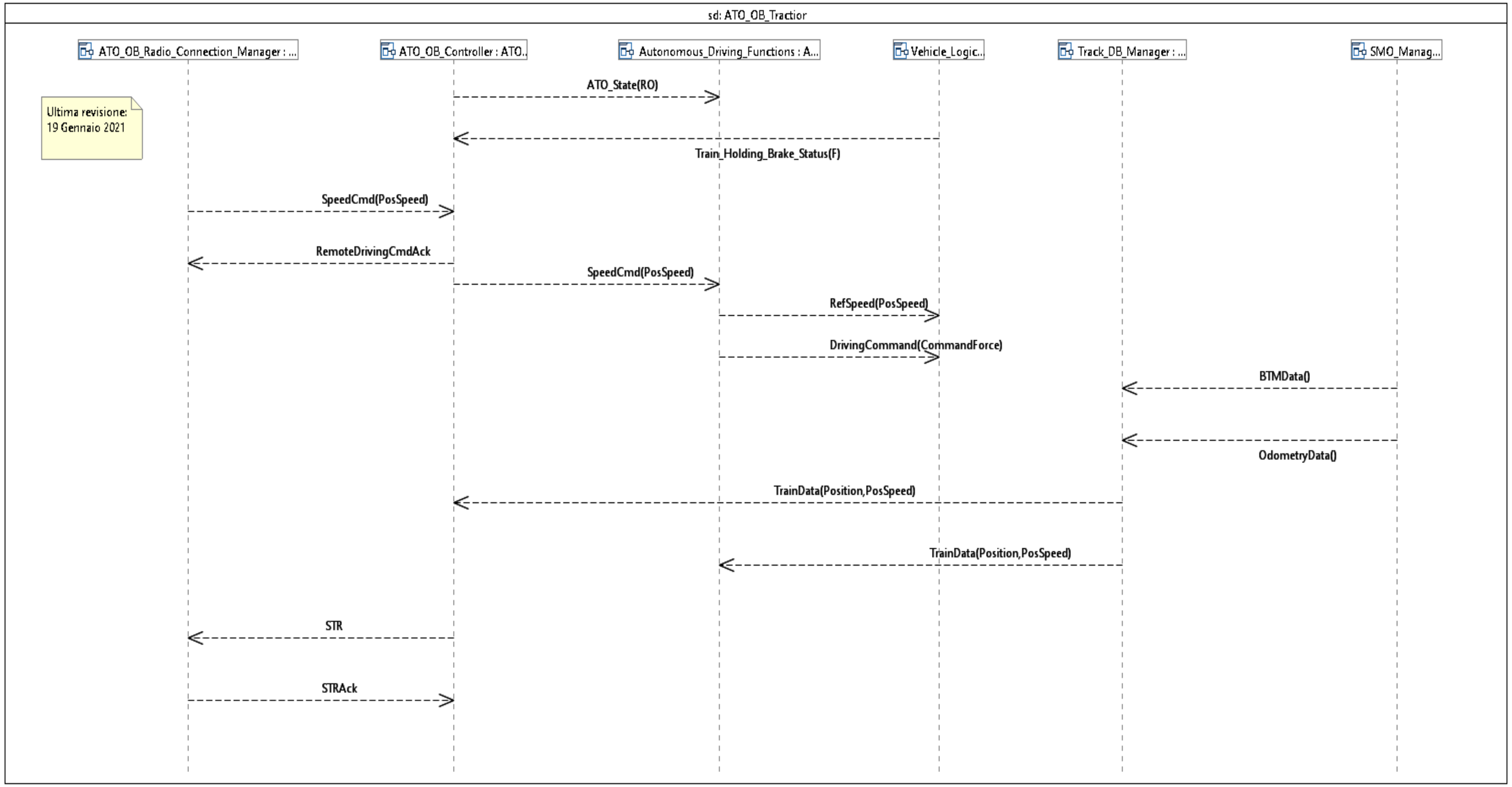
- **Phased V-model**: iterating through phases and configurations
  - Each phase iterates all the activities in the V-model



Formal Design …

Requirements and Architectural Design

- Requirements taken from UNISIG subset
  - Extended and customized for project specific goals
- Complemented by a set of operational **scenarios**
  - Specifying use cases
  - Formalized into sequence diagrams



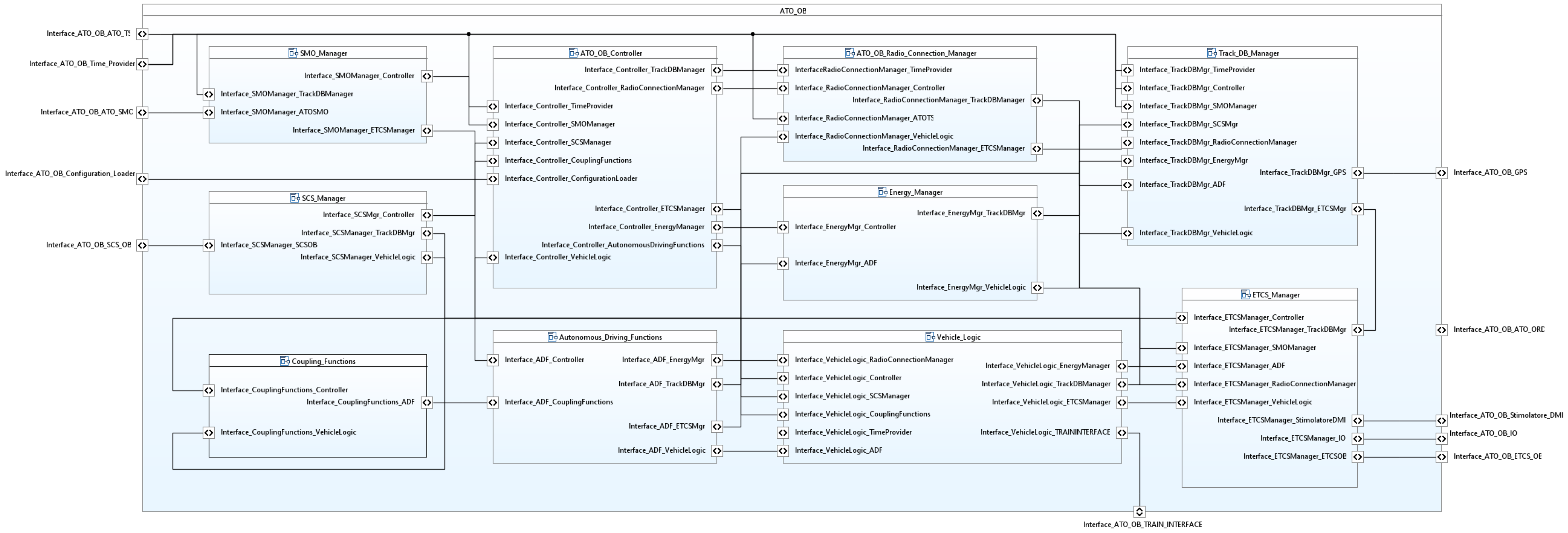
Example scenario: ATO_OB_Traction (sequence diagram)

Requirements and Architectural Design

- Requirements and scenarios guide the design of the logical architecture and hierarchical decomposition of ATO
  - Used to define the component interfaces
  - Guiding the implementation of the components
  - Used to derive test suites to perform unit and integration testing
- Architecture modeled in SCADE Architect
  - Block Definition Diagrams (BDDs) and Internal Block Diagrams (IBDs) in SysML
  - Diagrams specify the hierarchical decomposition of the system, connections, interfaces and data types

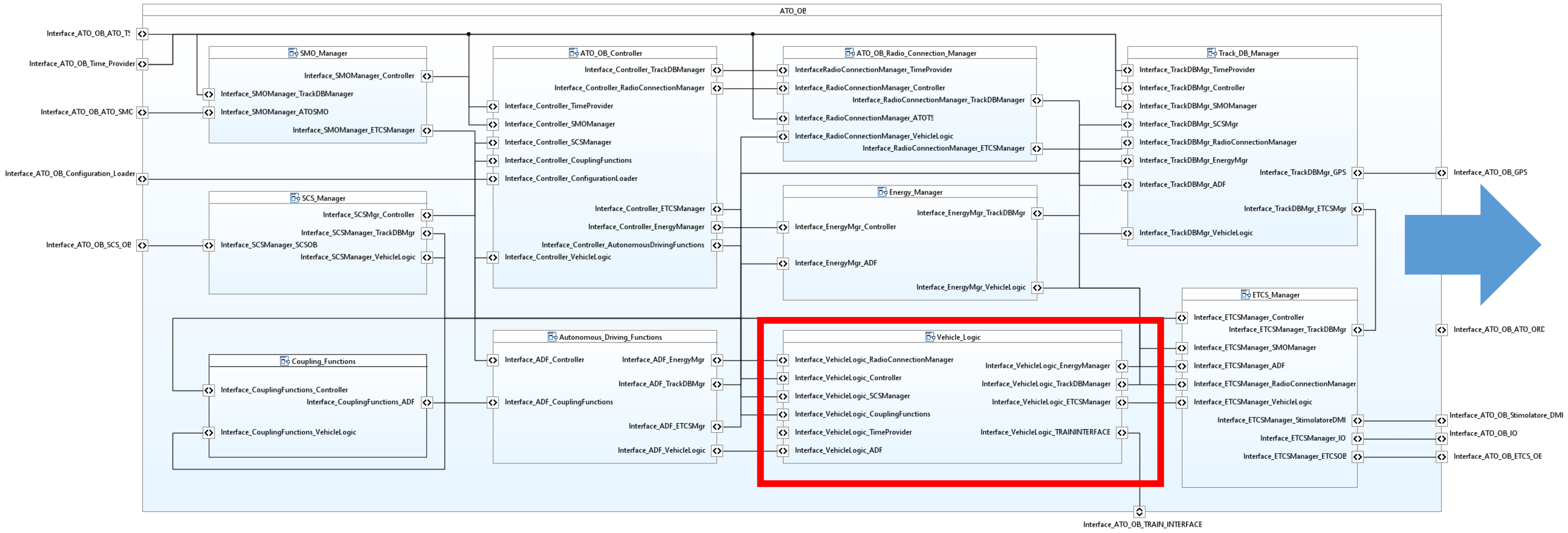
Architectural Design (ATO-OB)

- Architecture modeled in SCADE Architect



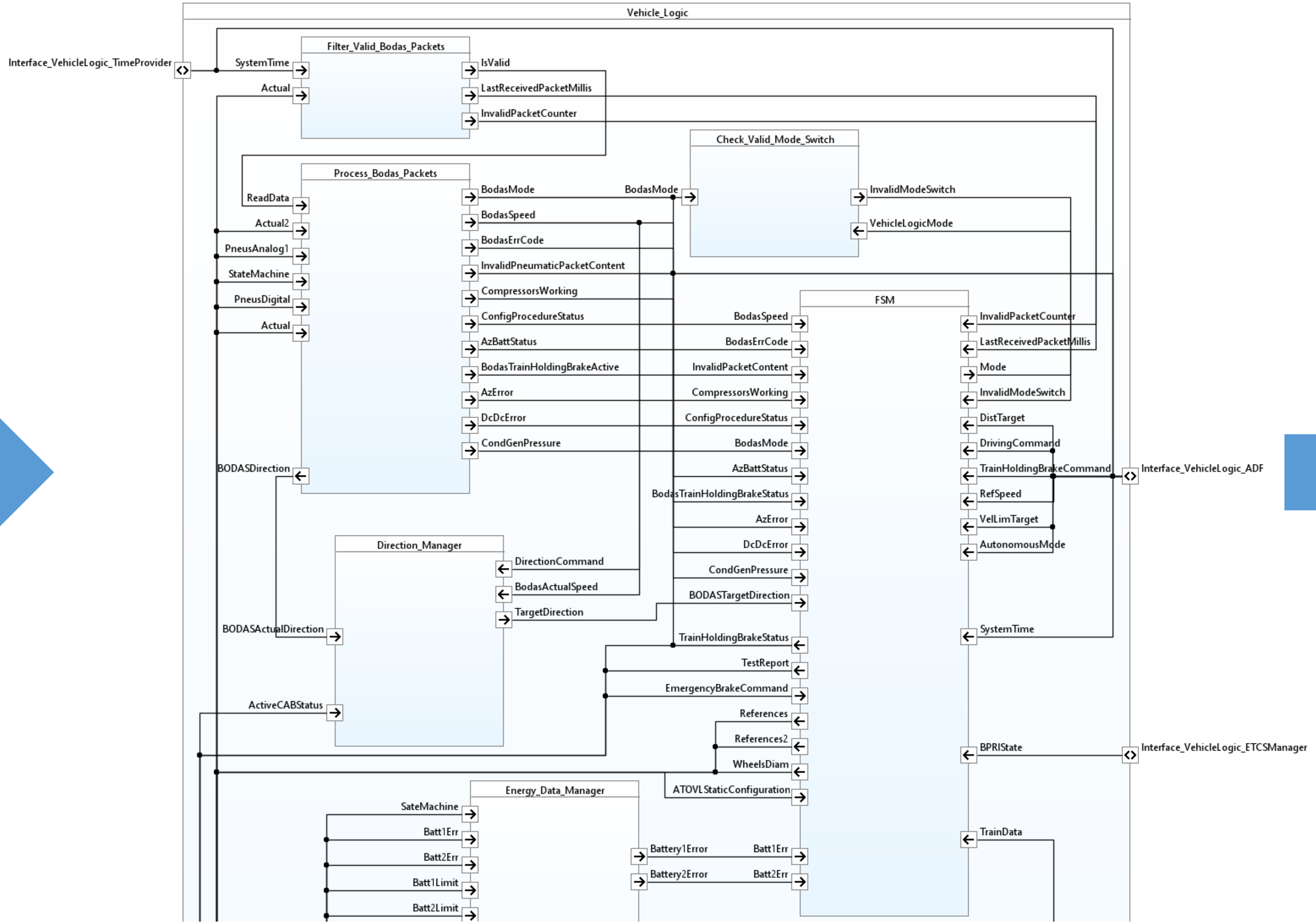
Architectural Design (ATO-OB)

- Architecture modeled in SCADE Architect



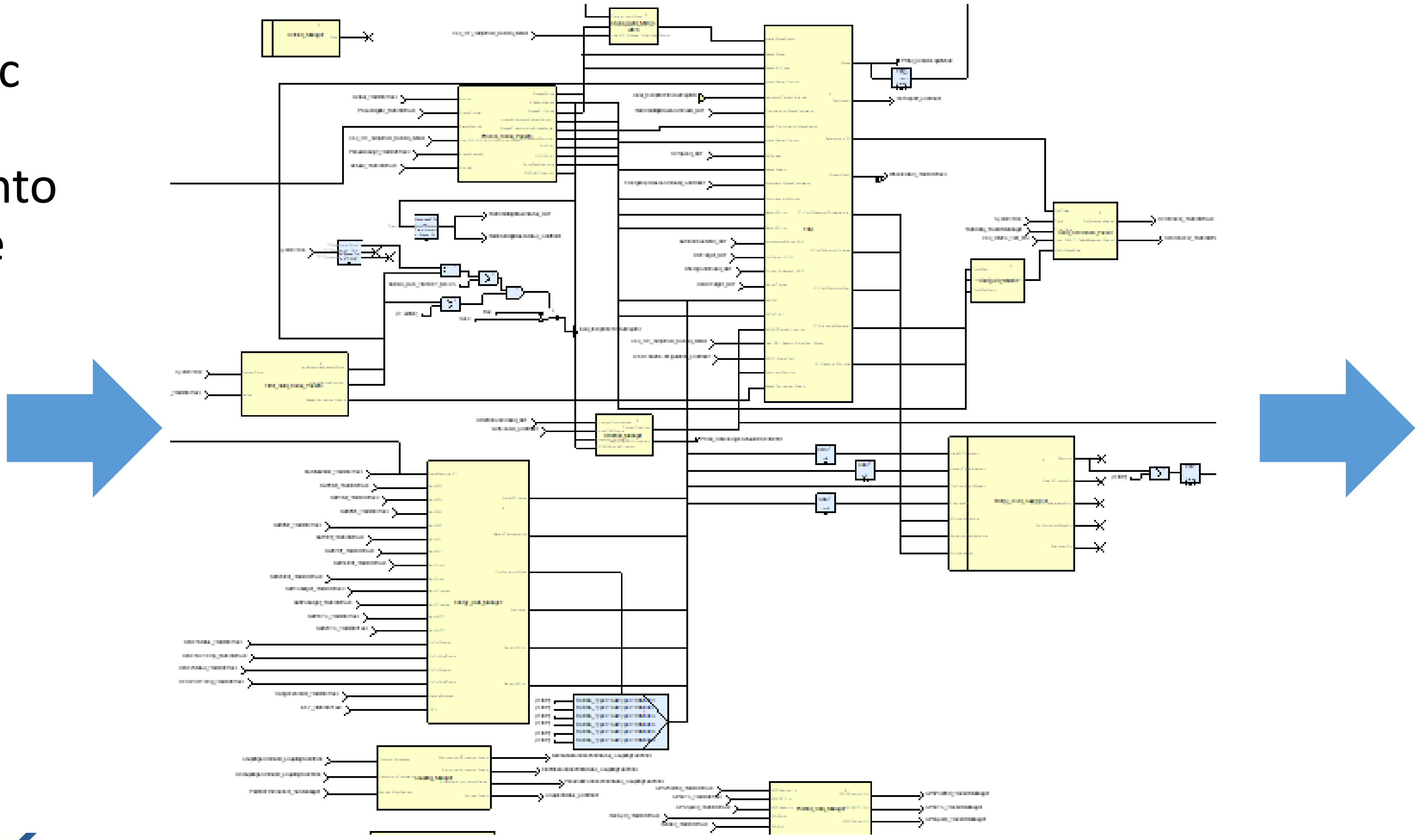
Architectural Design (ATO-OB)

- Architecture of Vehicle Logic component in SCADE Architect



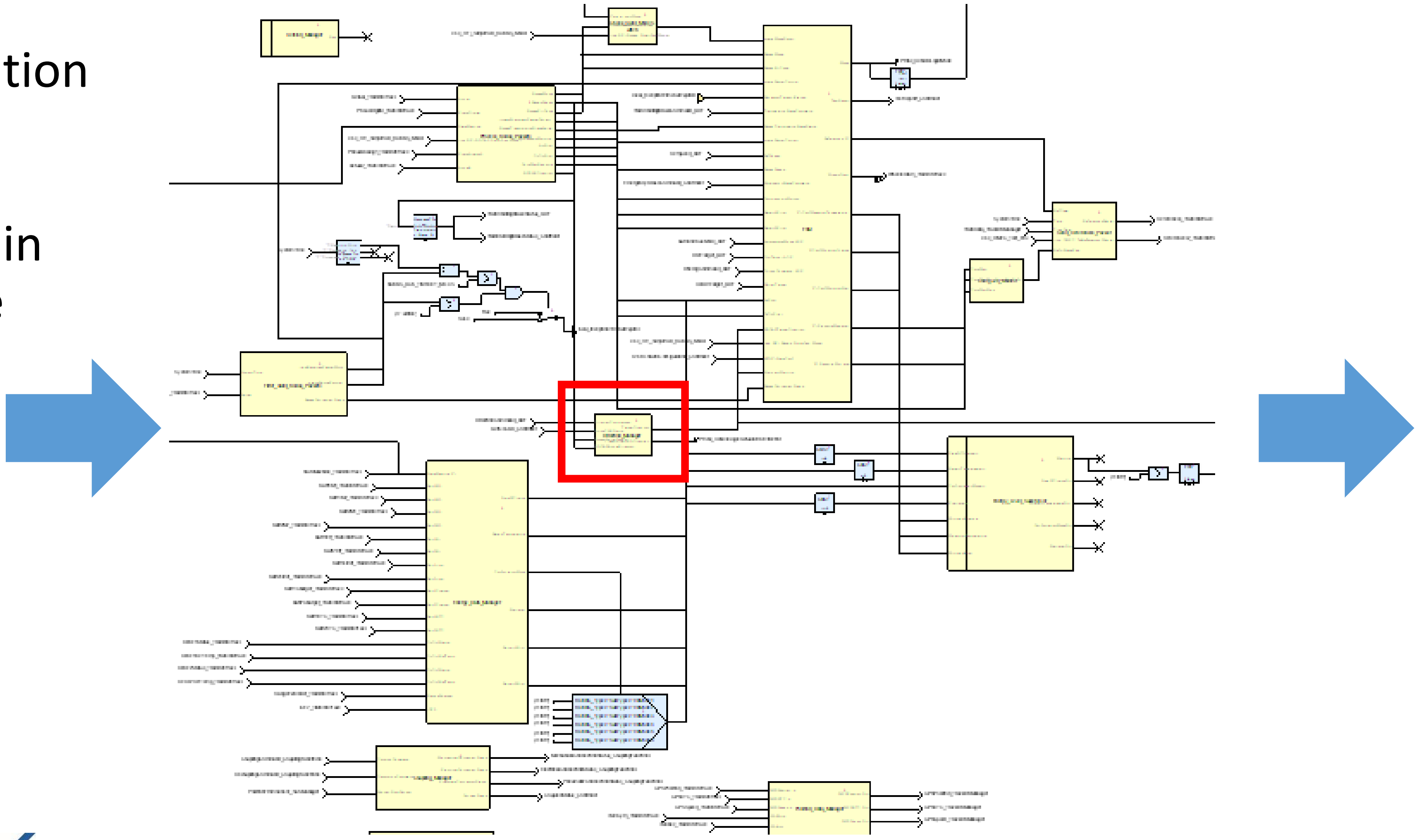
Component Design (ATO-OB)

- Vehicle Logic component converted into SCADE Suite



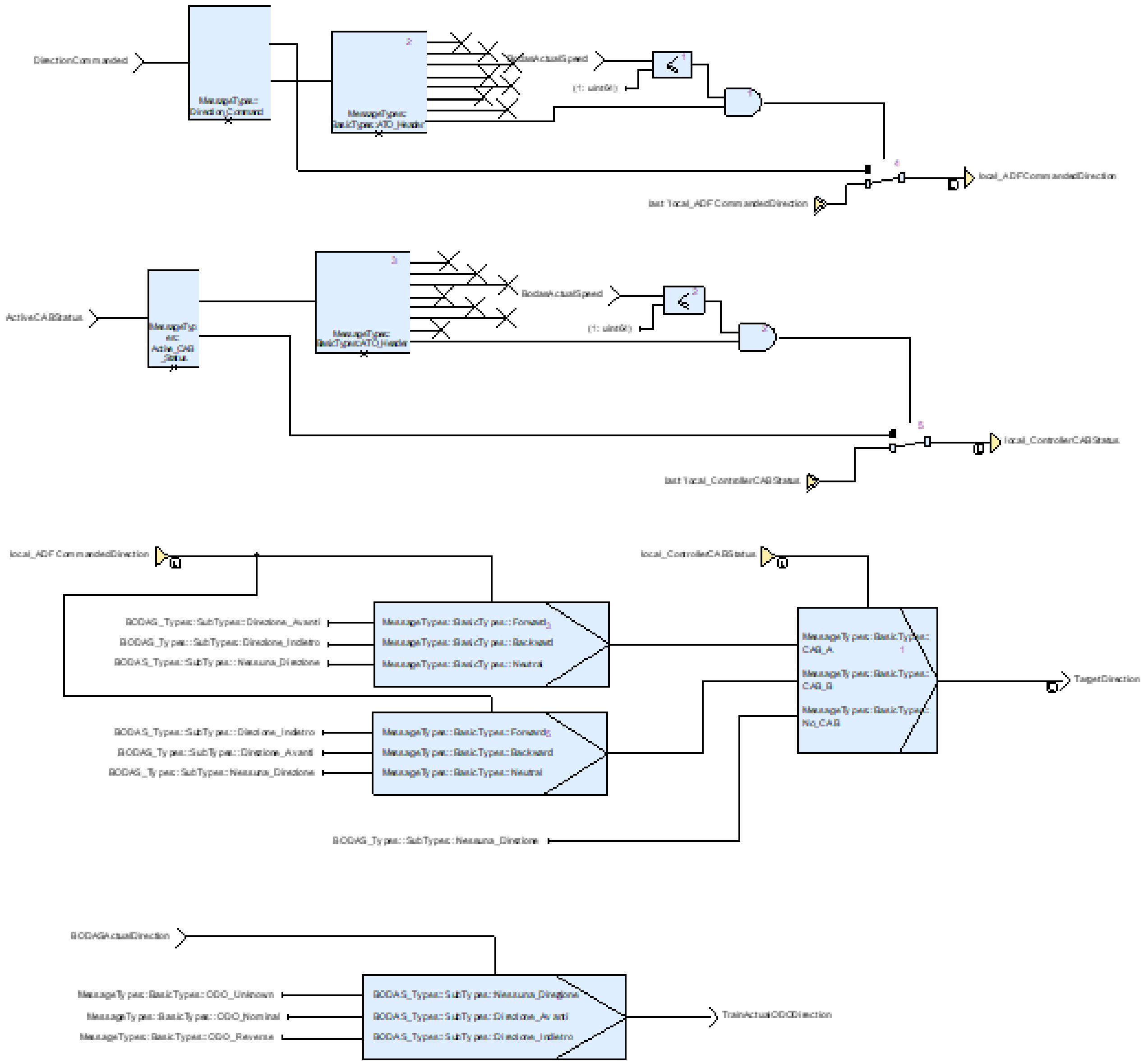
Component Design (ATO-OB)

- Implementation of Direction Manager component in SCADE Suite



Component Implementation (ATO-OB)

- Implementation of Direction Manager component in SCADE Suite



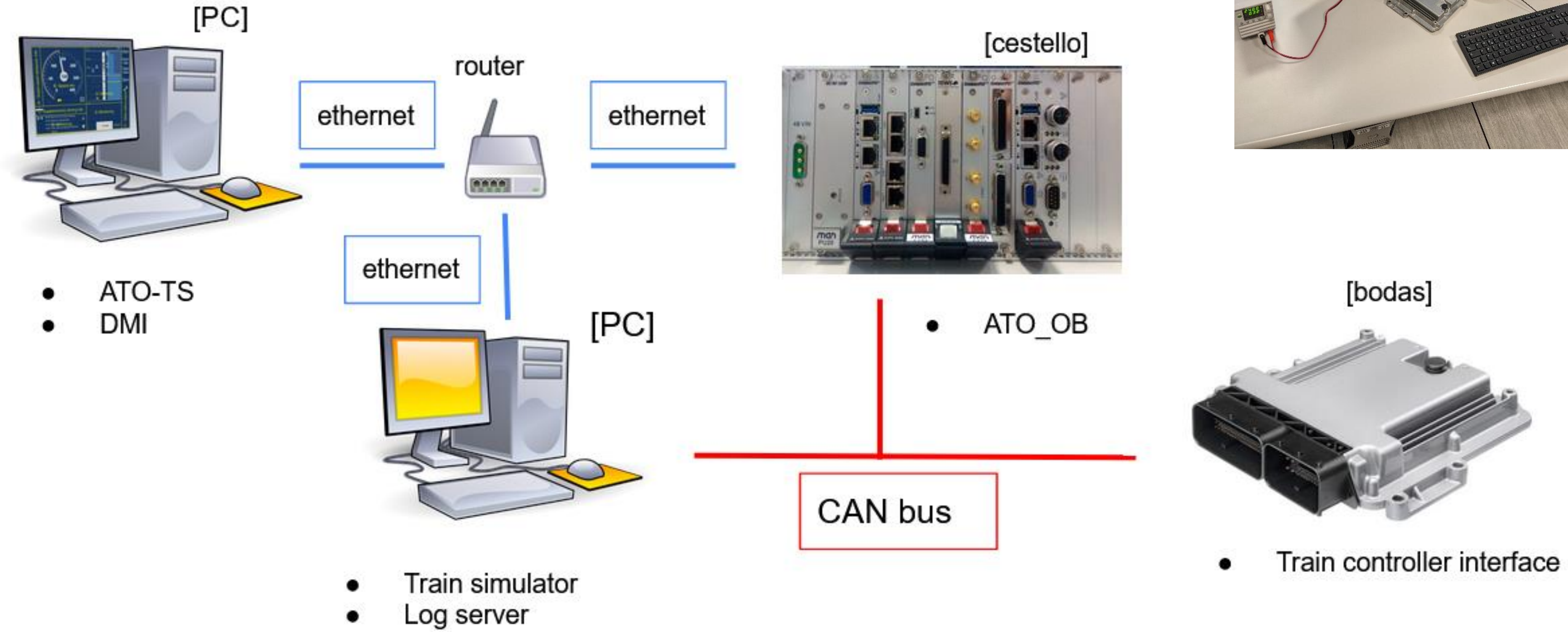
Component Design and Implementation

- Components implementation
  - Most components implemented in SCADE Suite language
  - One subsystem is implemented using Simulink (ADF)
  - One data-intensive component is manually written in C (TrackDB)
- Size of the design
  - Overall, ATO-OB Software contains about 75K lines of code
  - Each component has between 30 and 100 I/O ports
  - ATO-OB interface has more than 120 I/O ports
- Code generation
  - SCADE Suite comprises a code generator for translating models into C code
  - For ATO-OB, about 75% of the whole code was generated automatically

System Integration

- Challenging due to heterogeneous components (SCADE, Simulink, C)
- Testing design
  - A test suite is associated to each component
  - SCADE external operators used to link source code of external components
  - Operational scenarios are used to derive integration test cases, to test component interaction
- Testing strategy
  - Unit testing in Configuration 1 (SCADE)
  - Integration testing in Configurations 1—3 (SCADE/PC/HIL)
- Continuous-integration approach, based on git versioning control system, is used to prevent non-regression failures

HIL Integration Setup



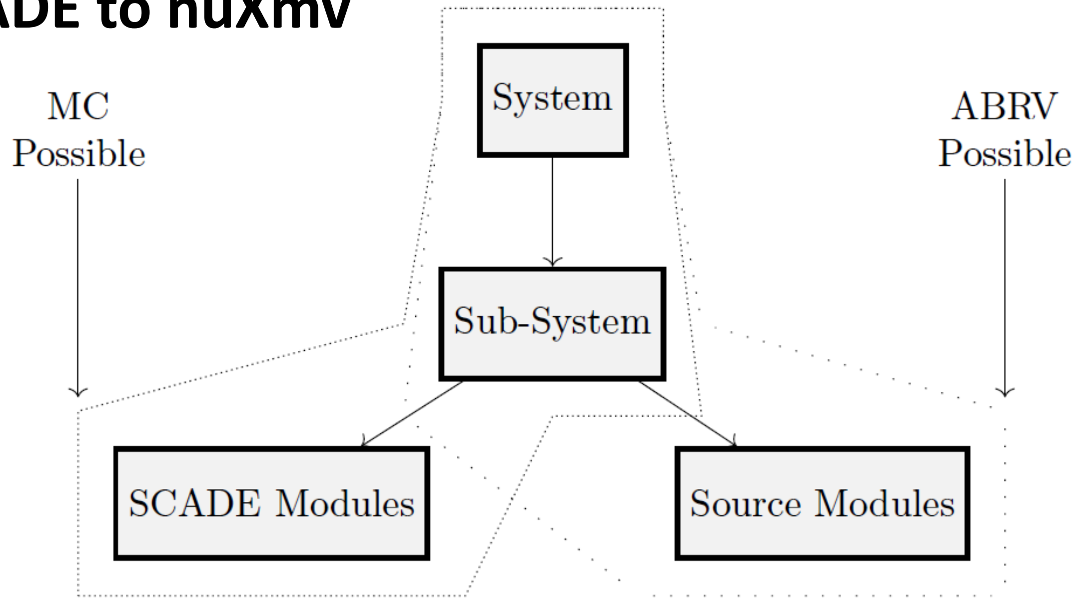
Verification and Validation …

Verification and validation

- Functionality offered by SCADE Architect and Suite
  - Validation tools for early identification of flaws (check compatibility of interfaces, consistency of sequence diagrams and data)
  - Scenario validation to design, simulate and test the system
  - Model coverage feature, to pinpoint paths of the model that are not stimulated by tests
  - Logging enables visualization and verification of ATO outcomes after executing a scenario
- Complemented by formal V&V functionality offered by FBK tool chain
  - Based on model checking and runtime verification

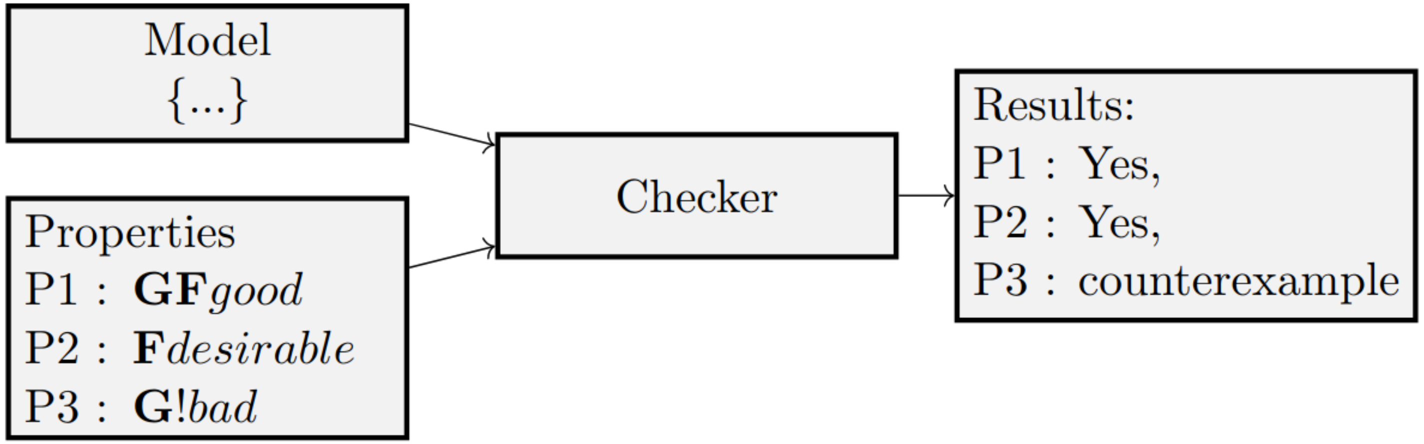
Verification and validation

- In-house formal V&V functionality
  - **Based on the nuXmv model checker and NuRV**, an extension of nuXmv for runtime verification
  - **Based on a (in-house) translation from SCADE to nuXmv**
- Two complementary approaches
  - **Model checking (MC)**
    - Applies to SCADE modules
  - **Assumption Based Runtime Verification (ABRV)**
    - Applies to generic components



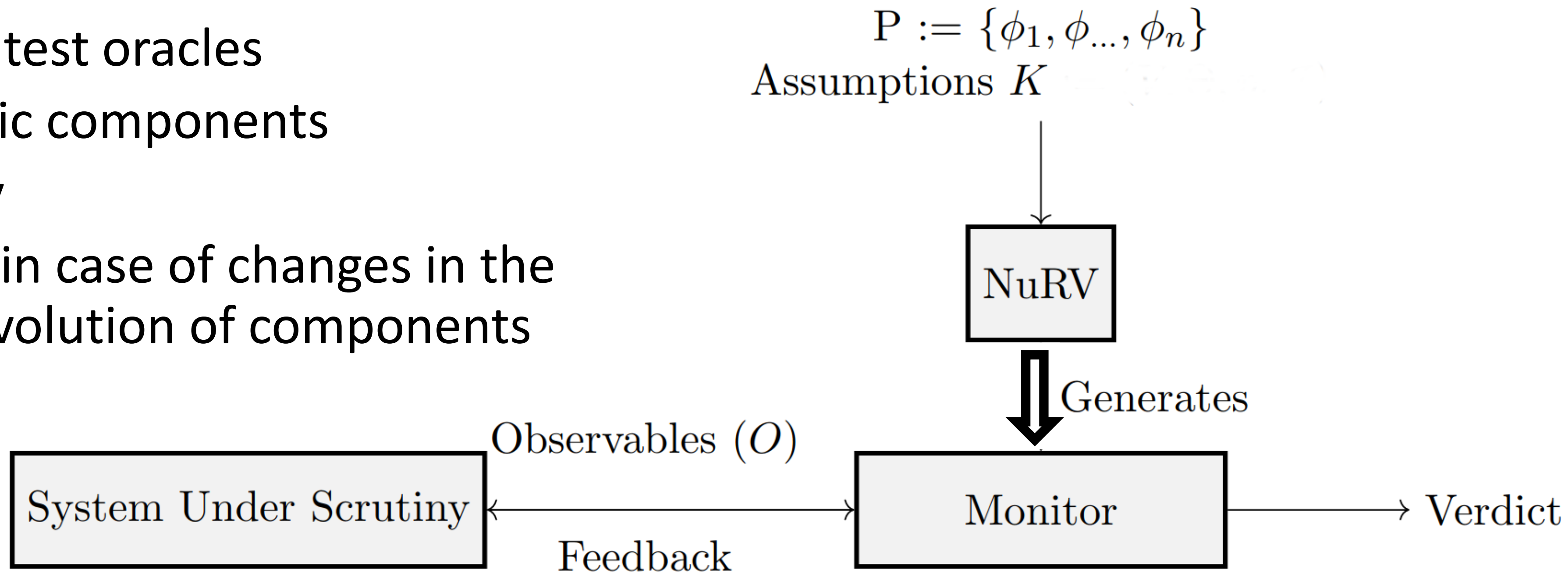
Model Checking

- Based on the nuXmv model checker
- Performs system-level property-based verification
- Features
  - Applies to SCADE components
  - Scalability issues: verification may run out of time



Assumption Based Runtime Verification

- Based on NuRV, an extension of nuXmv for runtime verification
- Automatically generate monitors from properties
- Features
  - Use monitors as test oracles
  - Applies to generic components
  - Better scalability
  - Easy refactoring in case of changes in the interfaces and evolution of components



Lesson Learned and Conclusions …

Lessons Learned: Challenges and solutions

- Distribution of work and responsibilities among teams
  - → Continuous integration
  - → Ad-hoc strategies to support system evolution
- Complexity of the design
  - → Progressive design and implementation using a phased V-model
  - → Concentrating on one scenario at a time
  - → Test the integrated system on different deployment configurations
  - → Progressive release of the system on different targets
- Complexity of the V&V activities
  - → Mix of strategies: static checks, simulation, proprietary tool chain for formal verification

Conclusions and Future Work

- Effectiveness of Formal approach
  - Most of the flaws encountered during system integration were located in outsourced components (designed and tested using traditional methodologies)
  - Reduction in development costs and expected reduction in certification activities
    - Estimated in the order of 50%
- Status of the ATO development
  - Prototype single-unit unmanned light vehicle
  - Currently being tested in laboratory
  - Field tests by the end of 2022 on the Bologna San Donato railway test circuit, the first fully equipped laboratory in the field throughout Europe
- Future developments
  - ATO vehicle able to control and drive a multiple-unit high-speed train, with passengers on board