WP5 Demonstrator

Kick off

09/10/2013 - Braunschweig

ITEA 2

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

openETCS

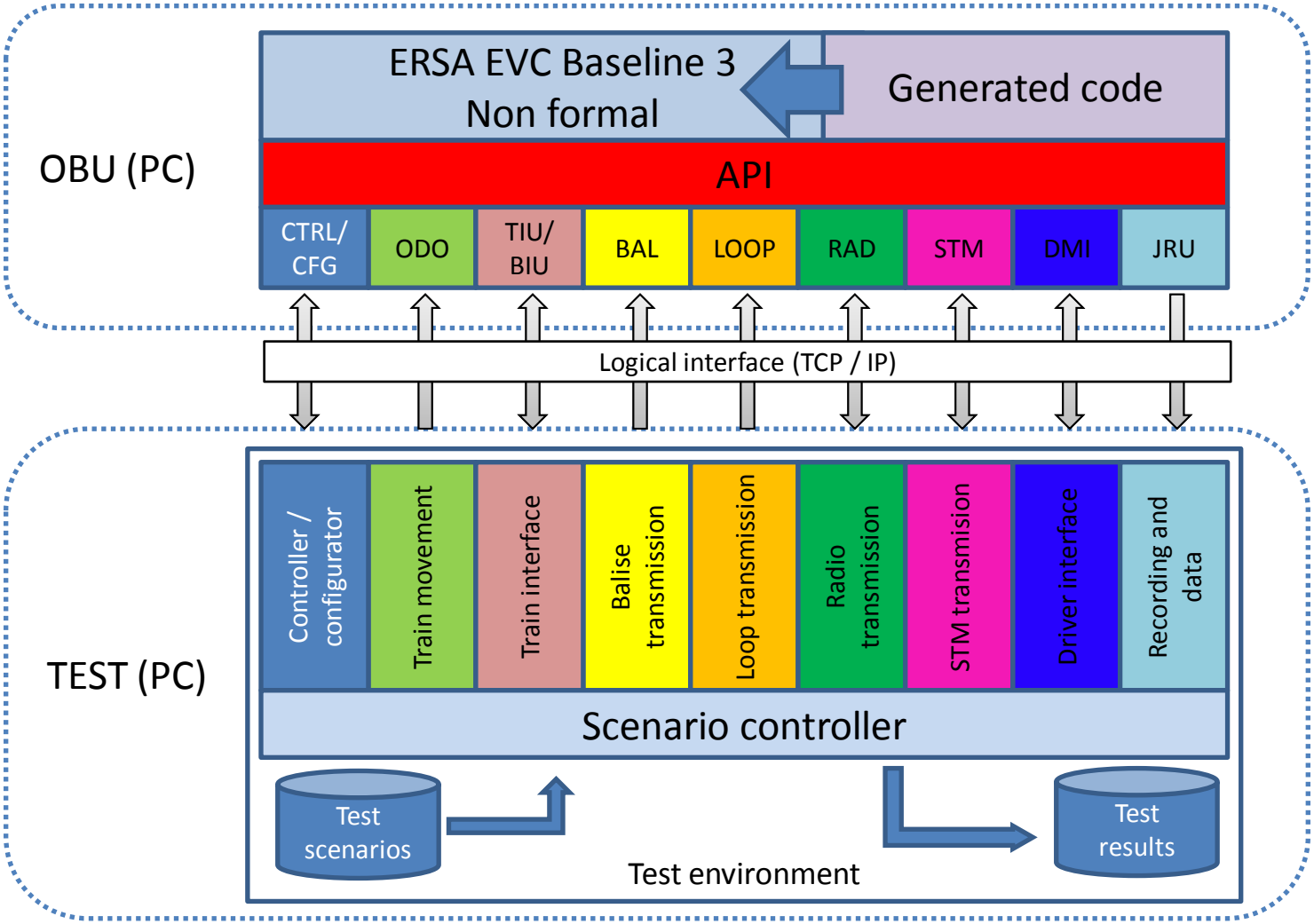
Overview

- Current status of WP 5
- Functional specifications
- Delivery of on board simulation and related issues
- Delivery of test environment and test material
- Industrial platform (introduction)
- Proposed planning
- Identification of possible tasks
- Further steps

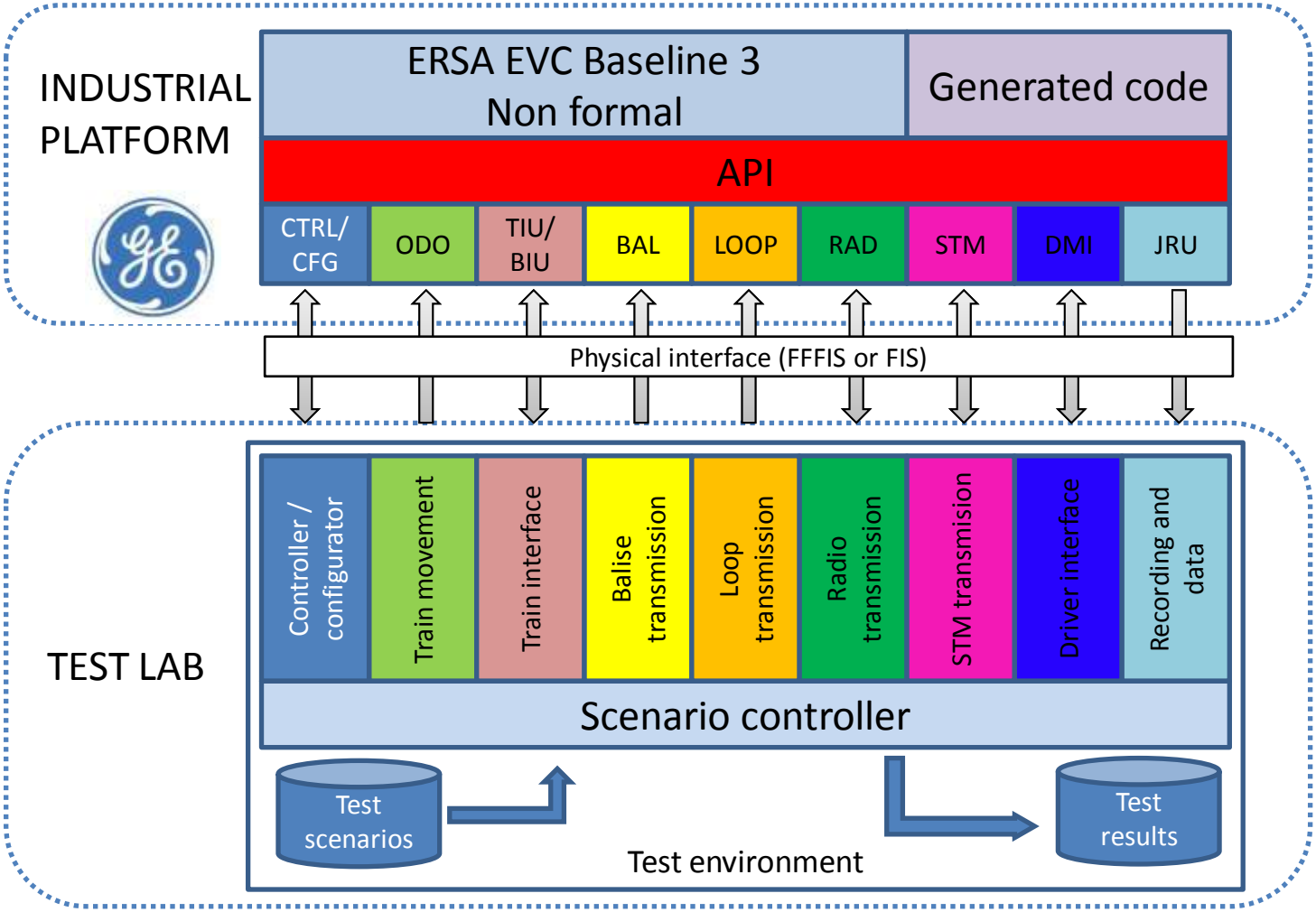
Main objectives

- Provide a Demonstrator to show the results of the openETCS project (EVC code mainly)
- Provide a Test Environment to prepare and run scenarios and produce test reports
- Provide a non vital reference unit

Demonstrator



Industrial Platform - Non vital application



openETCS

Main tasks (FPP)

- 5.1 Functional Specification
- 5.2 A Production of on ON BOARD Simulator Iteration 1
- 5.2 B Production of on ON BOARD Simulator Iteration 2
- 5.3 Production of Test Environment
- 5.4 Data preparation
- 5.5 Execution of scenarios
- 5.6 Exploitation of results
- 5.7 Extension of Test Environment with physical interfaces
- 5.8 Integration/Connection of Non Vital Platform
- 5.9 Execution of industrial tests

Project partners

	5.1	5.2	5.3	5.4	5.5	5.6	5.7	5.8	5.9
AEBT						X			
Alstom	X	X		X	X	X			
CEA		X							
DLR	X		X	***X***	X	***X***			
ERSA	***X***	***X***	***X***	X	***X***	X	***X***	***X***	***X***
Fraunhofer		X		X					
GE							X	X	X
Innovalia	X	X							
Institut Telecom	X	X	X		X				
Siemens	X	X		X	X	X			
SQS	X	X	X	X	X	X			
TWT									
University Rostock	X	X	X		X				

Initial planning from FPP

- T0: 2012-07-01: Start of the project
- T0+6: Functional Specifications
- T0+18: First iteration of OBU Simulator
- T0+21: Data for testing
- T0+24: Preliminary test Environment
- T0+27: Test Environment with physical interfaces
- T0+30: OBU Simulator Demonstrator
- T0+34: Main Tests Executed
- T0+36: Report on Test results

Interaction with others WP

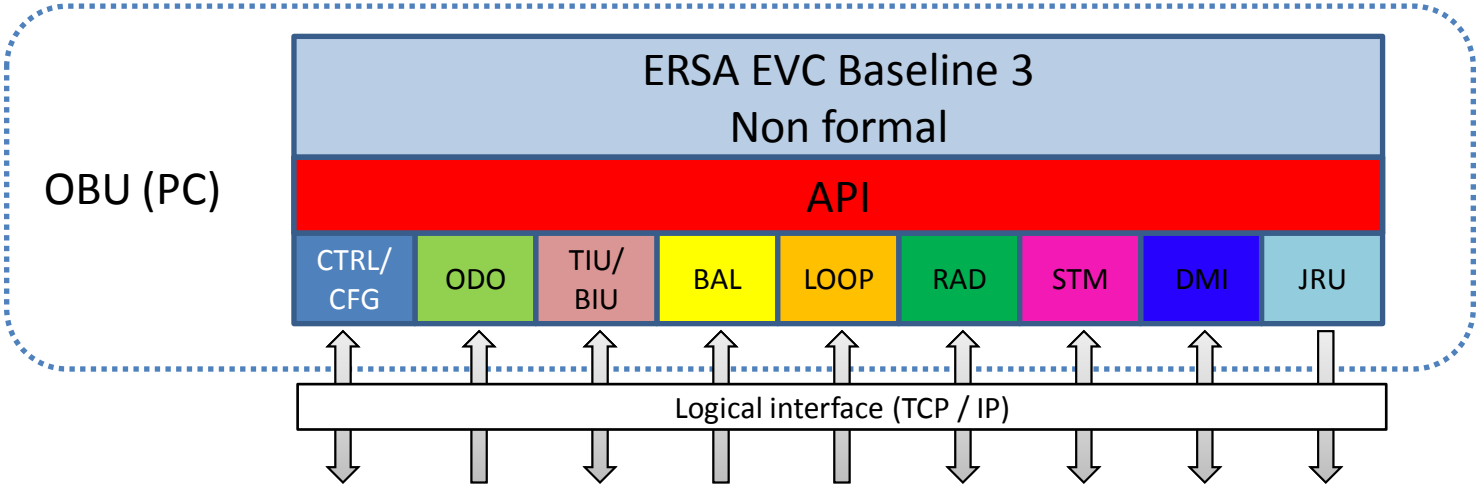
- WP2: T2.3 Requirements
  - **D2.6: Demonstrator (R-WP2/D2.6-02-086/090)**
  - D2.7: API definition (**draft exists**)
- WP3: T3.6 Functional model
  - D3.6: Definition of a functional model and of the corresponding functional API
  - Generated code from formal model
- SSRS:
  - API definition

T5.1 Functional specifications

- D5.1 is published on github : https://github.com/openETCS/demonstrator/tree/master/Documentation
- Working version (in progress)
- Test specification needs to be detailed
- Comments are welcome

T5.2a: ON BOARD Simulator iteration 1

- API implementation



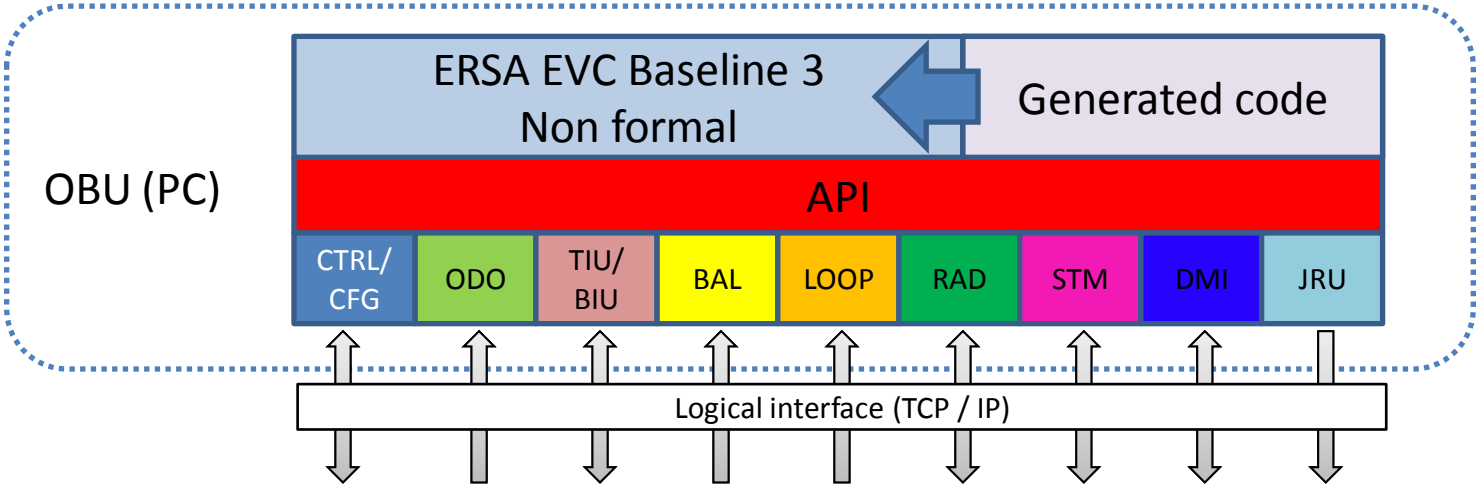
- Communication with peripherals

T5.2a: ON BOARD Simulator iteration 1

- Steps:
  - Prepare ERSA EVC code for publishing
    - Code cleanup
    - Remove unneeded or out of scope components
    - Ensure compliance with open source obligations
  - External API implementation “inspired” by WP3 and SSRS
  - Validation by ERSA tests
- Who:
  - Mostly ERSA
  - Partners involved with external API definition in WP3 / SSRS
- Issues:
  - Find EVC “scope” (what should or should not be implemented as openETCS project) is not always trivial
  - Make the ERSA EVC code useable outside ERSA environment
  - API definition that is well suited for ERSA existing code and for new code from the project

T5.2b: ON BOARD Simulator iteration 2

- Generated code integration

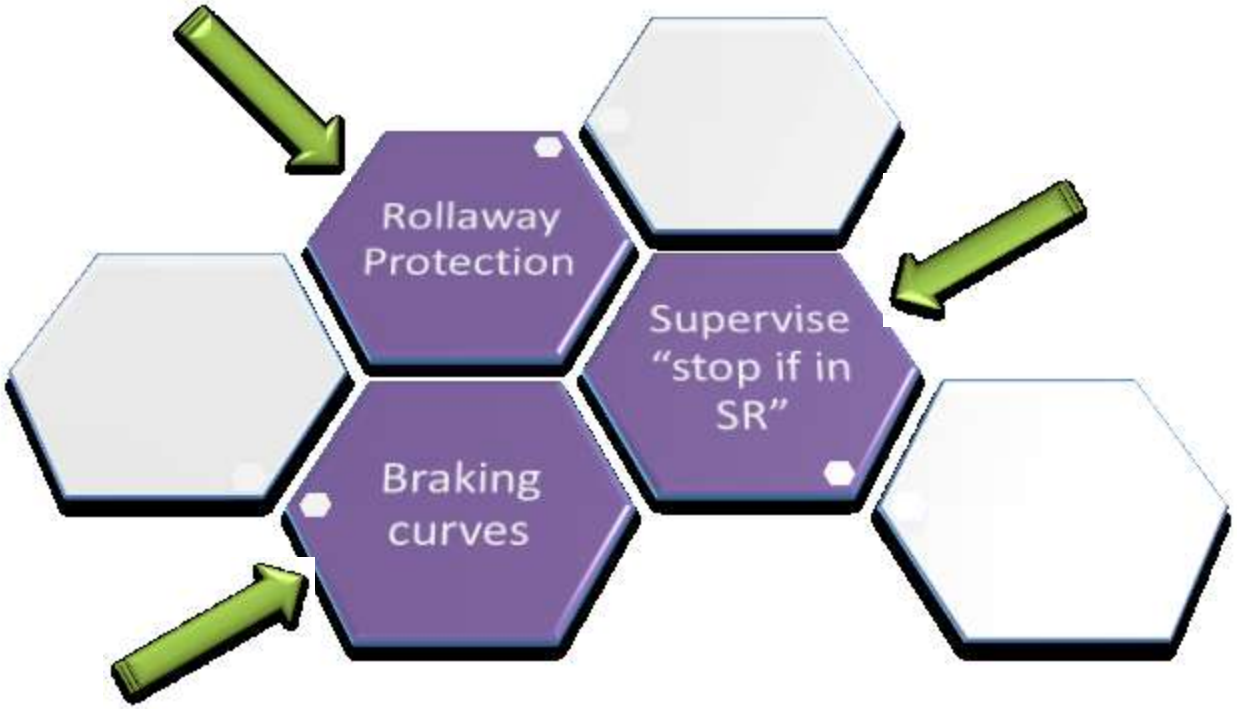


T5.2b: ON BOARD Simulator iteration 2

- Steps:
  - Prepare ERSA EVC code for generated code integration
    - Identify replaceable functions or blocks with WP3
    - Modularise EVC code to accept generated “formal” code
  - Integrate carefully selected generated blocks
    - Code refactoring / cleanup
  - Validation by ERSA tests
  - Loop while new replaceable code blocks are identified
- Who:
  - ERSA and others…
  - Partners involved with formal modeling (WP3)
- Issues:
  - Strategy for integration of generated code
  - Code integration into existing ERSA EVC code base as ERSA EVC internal architecture may be hard to change and adapt
  - ERSA is the only partner who really masters ERSA EVC code

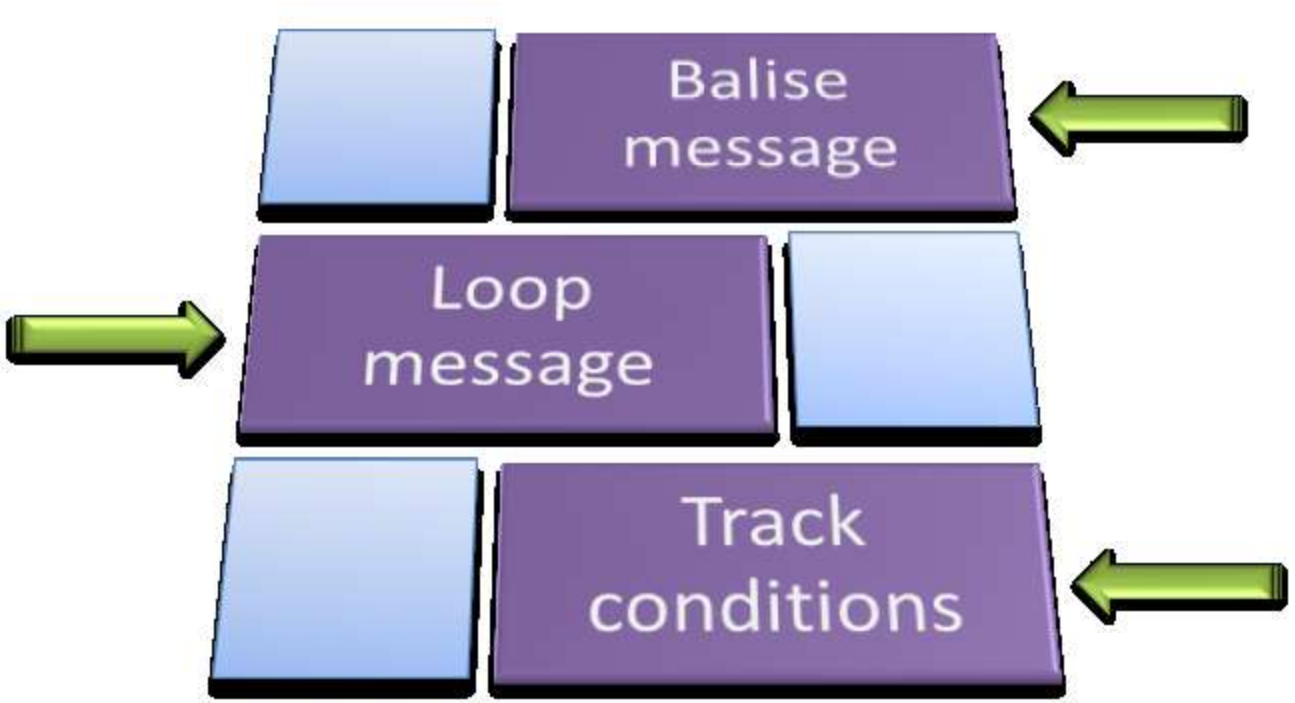
T5.2b: Integration by functions

- Replace only small self-contained code with simple and well defined input/output



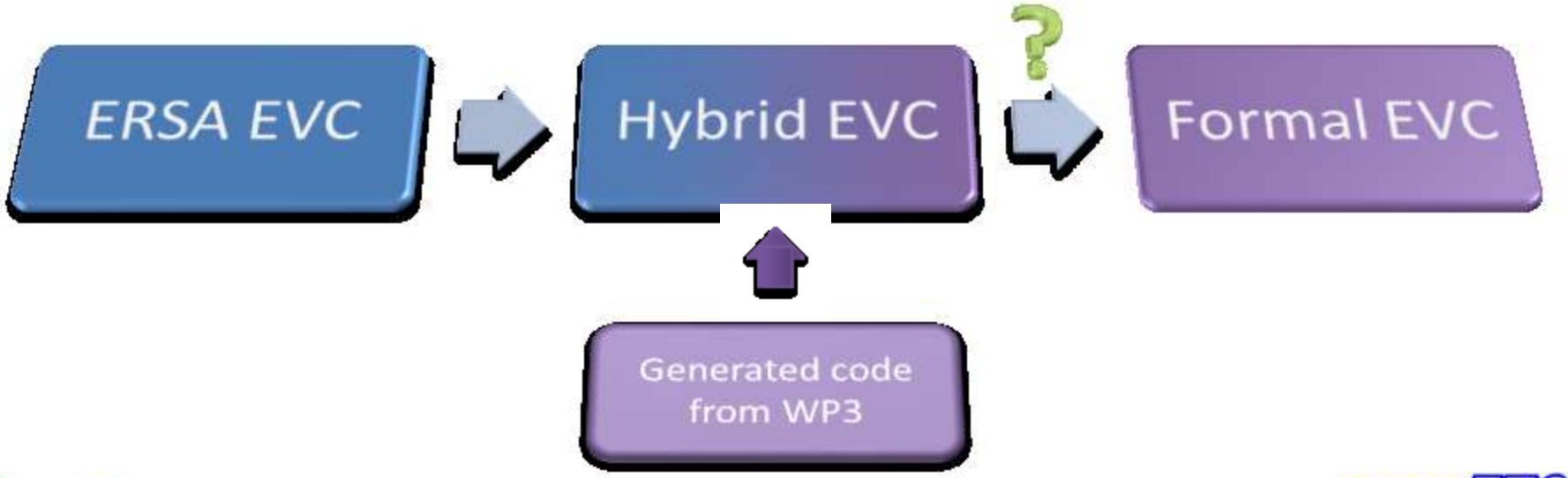
T5.2b: Integration by blocks

- Replace bigger blocks of code but still with limited interactions



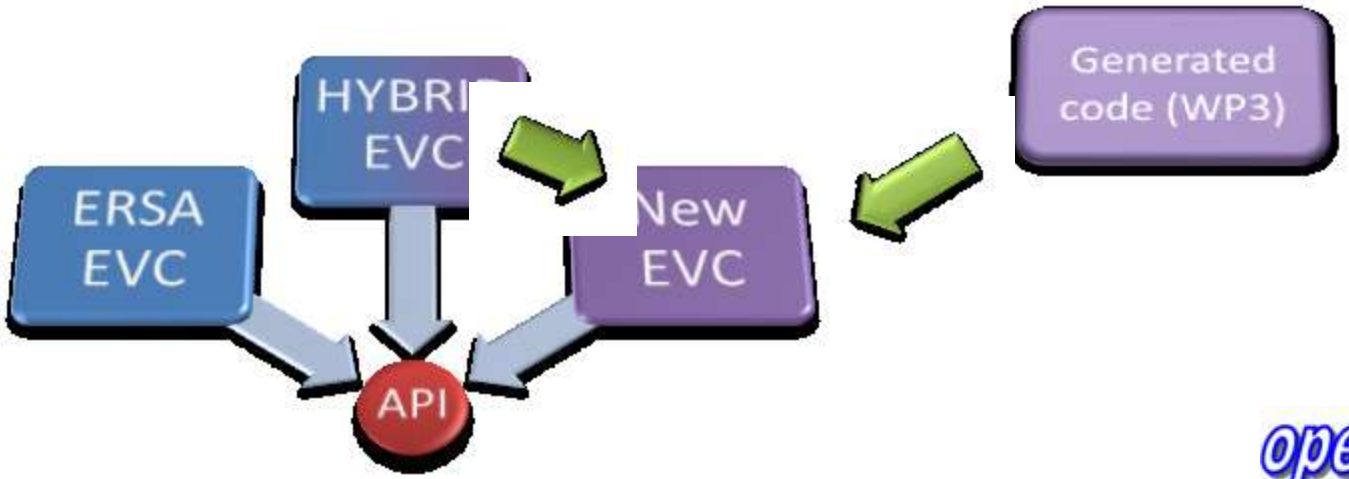
T5.2b: New code base strategy

- Integration of big blocks of generated code will be harder if not unrealistic because of ERSA architecture legacy
- We will reach a technical limit sooner or later

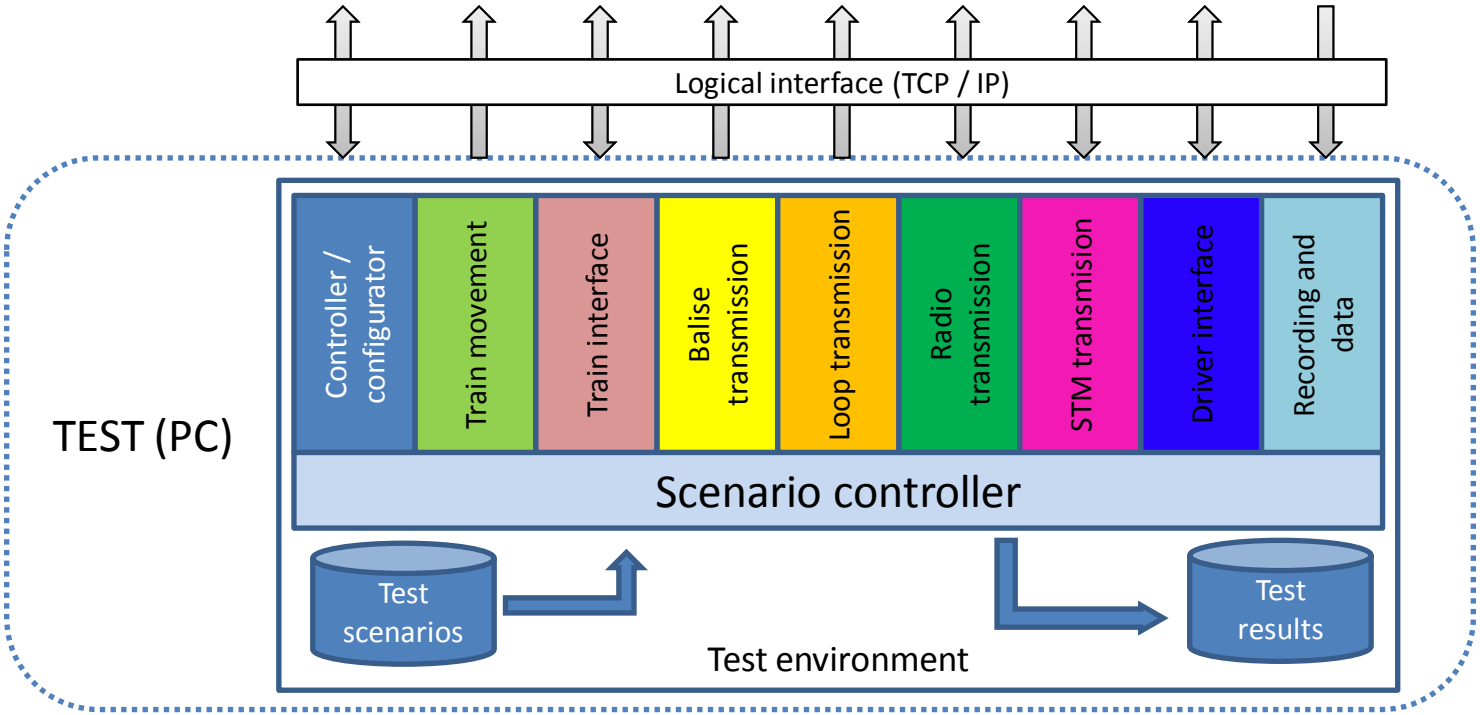


T5.2b: New code base strategy

- Capitalise on successful generated code from hybrid EVC to a new code base
- Common external API ensure EVC interoperability
- ERSA EVC or hybrid EVC can still be used as comparison basis



T5.3 – T5.6 Test environment



T5.3: Tests environment

- Steps:
  - Prepare ERSA test environment code for publishing
    - Code cleanup
    - Implement TCP/IP capabilities
    - Remove unneeded or out of scope components
    - Ensure compliance with open source obligations
- Who:
  - Mostly ERSA for software development
  - Other partners
- Issues:
  - Make the ERSA test suite useable outside ERSA environment

T5.4: Scenarios preparation

- DLR is the task leader
- Input data need to be collected
- Two kinds of tests:
  - Simple “unit tests”, not linked to any specific track and used for verification
  - Complex real life scenarios, used for validation
    - Pilot scenarios based on a high speed line
    - Main demonstration scenarios based on a corridor cargo line
- Taking up tests coming from verification/ validation (WP4)
- Tools to allow easy scenarios editing should be developed by partners
- Continuous test runner improvement to ensure proper execution

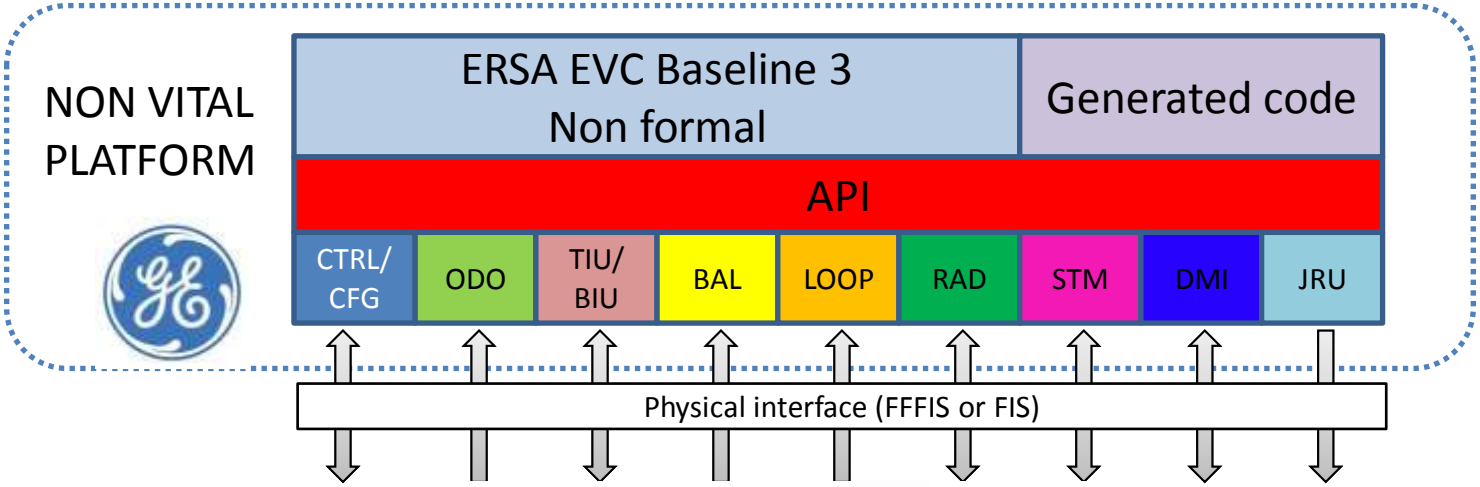
T5.5: Scenarios execution

- ERSA is the task leader
- Main goal: run tests and collect test results
- Scenarios execution should be automatic but a manual mode would be useful as well
- Test result and anomaly report templates should be defined by partners
- Feedback about inconsistencies or errors should be reported to WP3 as early as possible

T5.6: Result exploitation

- DLR is the task leader
- Log analysis
  - Conformity to unit tests
  - First evaluation on scenarios, deeper analysis done in WP4 Safety/ Assessment
  - Coverage
  - Impact and/or benefits of formal code
- Anomalies report to WP2/3

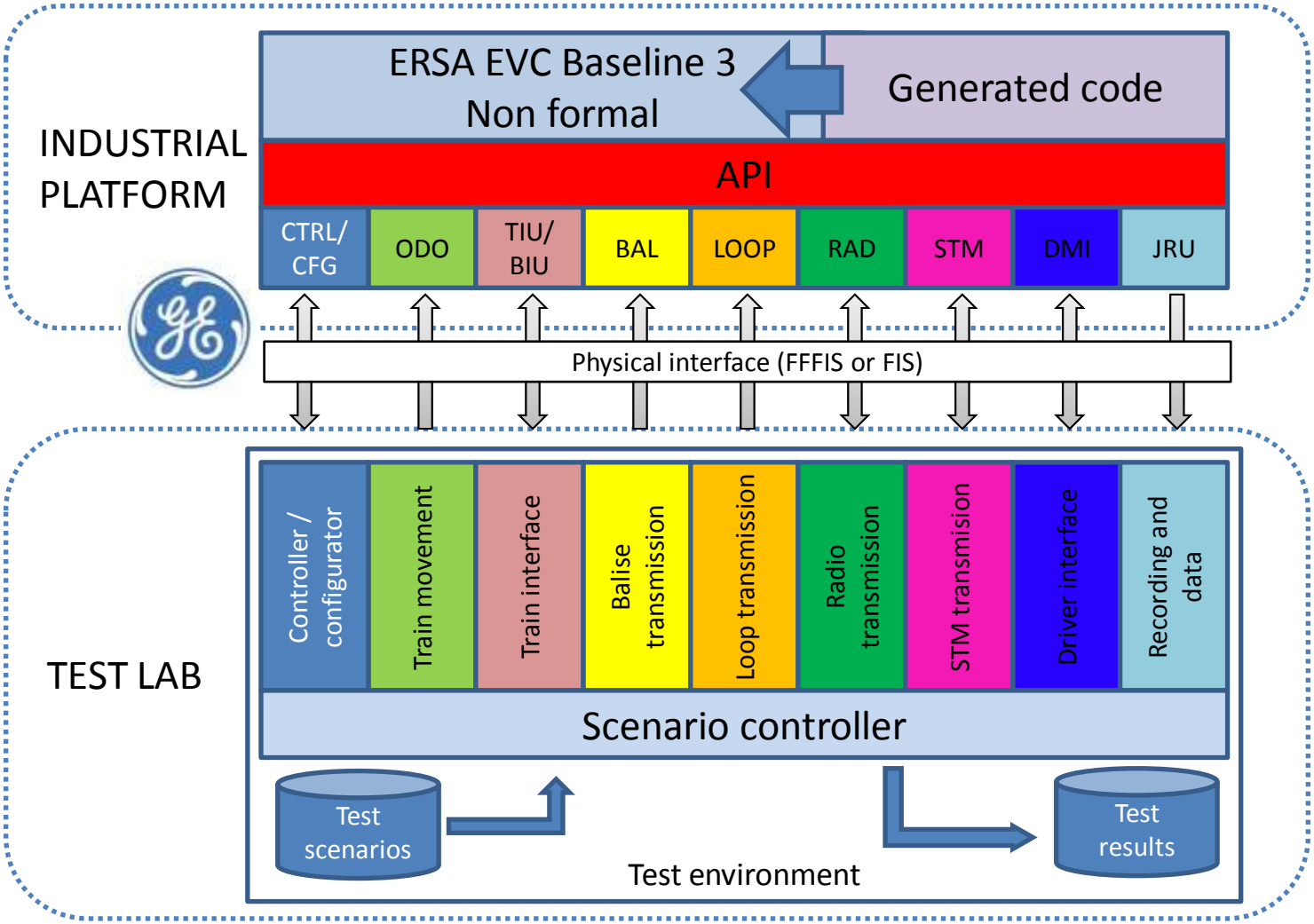
T5.7 – T5.9: Industrial platform – non vital application



T5.7 – T5.9: Non vital application

- Replace emulated OBU sub-systems by their equivalence from the reference non vital platform
- Prerequisites:
  - Hardware platform availability
  - Easy access to the peripherals proprietary software
  - Programming interface available in C language
- Who:
  - ERSA, General Electric
- Issues:
  - Sub-systems interface must match the external API defined in WP2

T5.7 – T5.9: Non vital application



Planning proposal

- November 2013: Functional Specifications
- March 2014: First iteration of OBU Simulator
- May 2014: Input scenarios
- July 2014: Preliminary test Environment
- December 2014: OBU Simulator Demonstrator (hybrid)
- January 2015: Main tests can start
- April 2015: Report on Test results

Identification of possible tasks

- Priority list of EVC functions to be integrated
- Detailed feasibility study of the integration methodology
- Data collection for testing
- Template for test scenario
- Criteria for evaluation
- Template for test report
- Review of documents
- Review of tests scenarios and test results
- Quality tasks
- Other tasks?

Next steps

- Comments to the Functional Specification
- Refinement of the Functional Specification
- Consolidation of the Functional Specification
- Consolidation of task list and feedback from WP partners
- Consolidation of updated planning