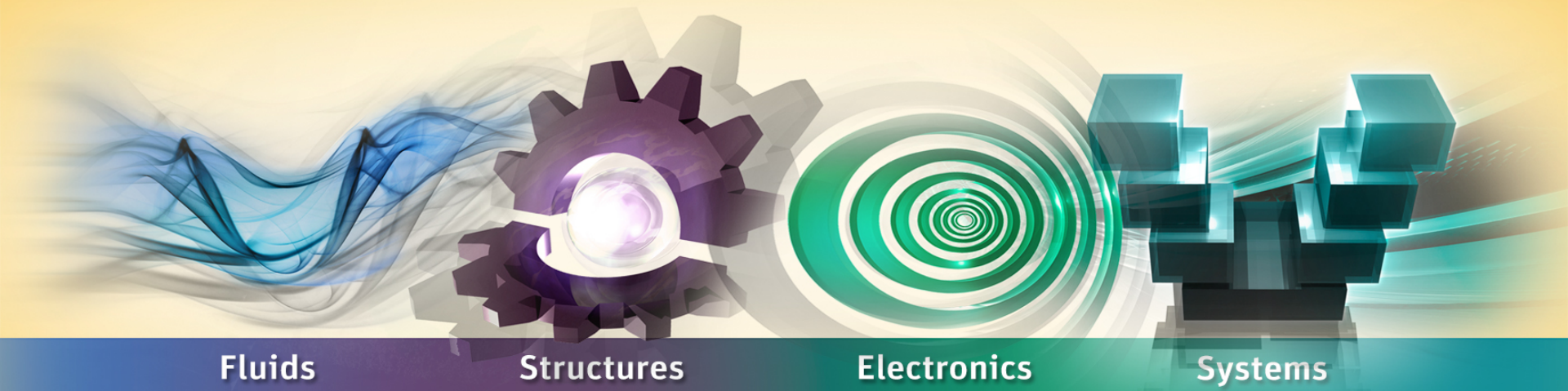


Model-Based Verification and Validation with SCADE Suite



SCADE Combined Testing Process with TE and TMC

- **SCADE Combined Testing Process (CTP)**
- **SCADE Test Environment (TE)**
- **SCADE Test Model Coverage (TMC)**

- **SCADE Combined Testing Process (CTP)**
- **SCADE Test Environment (TE)**
- **SCADE Test Model Coverage (TMC)**

The SCADE Combined Testing Process is a SCADE model-Based approach for combining efficient and rigorous testing activities

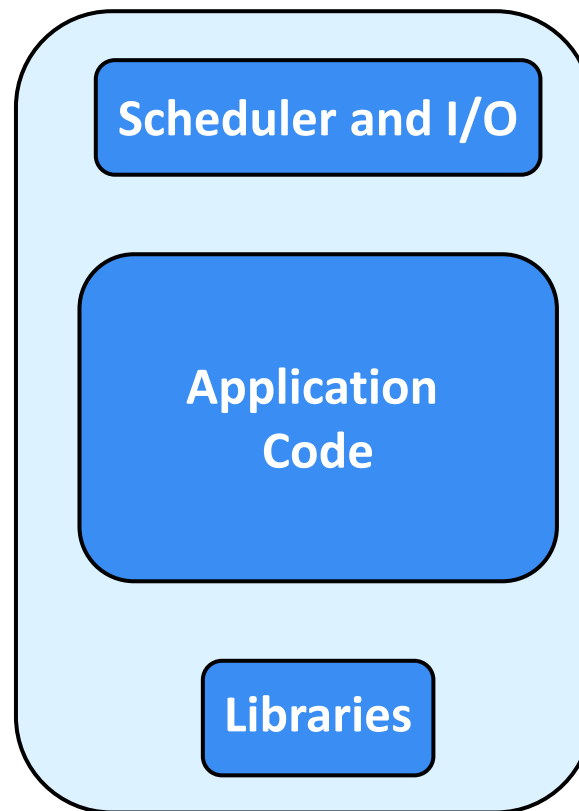
Objectives are the following:

- Optimize the testing effort
- Maximize the benefit from KCG qualification/certification

Combination is the following:

- Software requirements-based testing
- Low-level testing on a sample
- Testing on host and on target

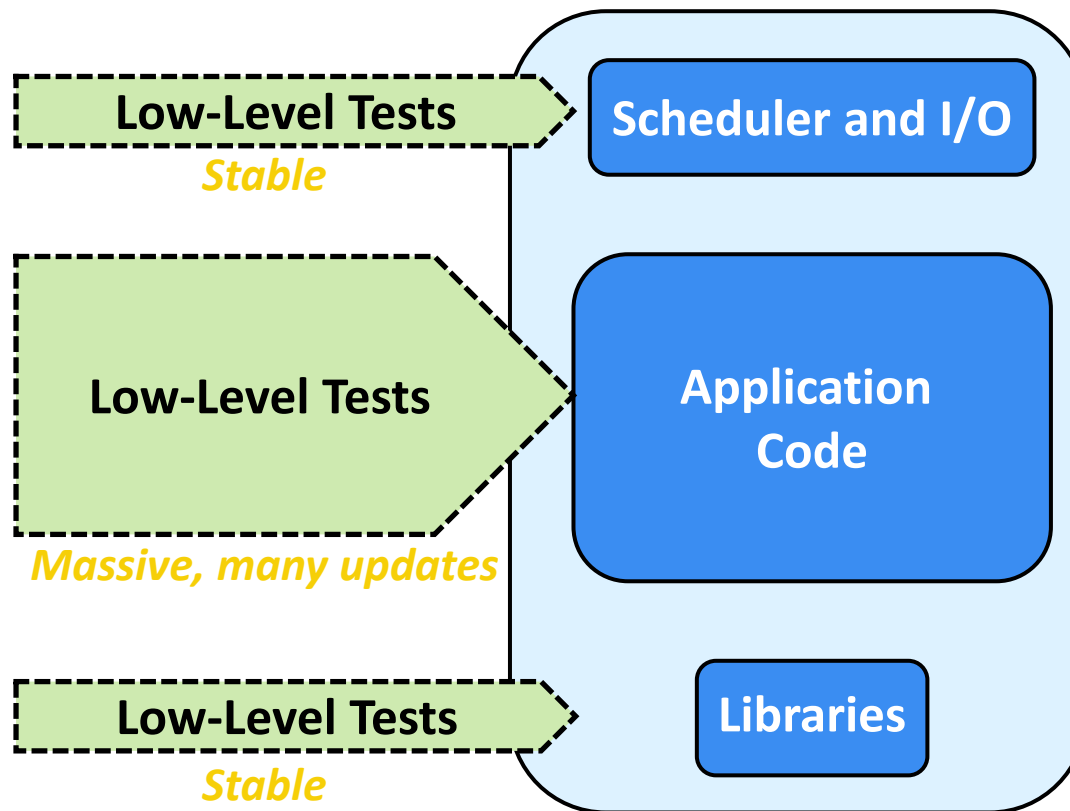
“Traditional” Testing Process



An embedded application is typically made of:

- A scheduler or an operating system + drivers for inputs and outputs
- The code of the application
- Libraries of some elementary operations (C code) that are repeatedly used in the application

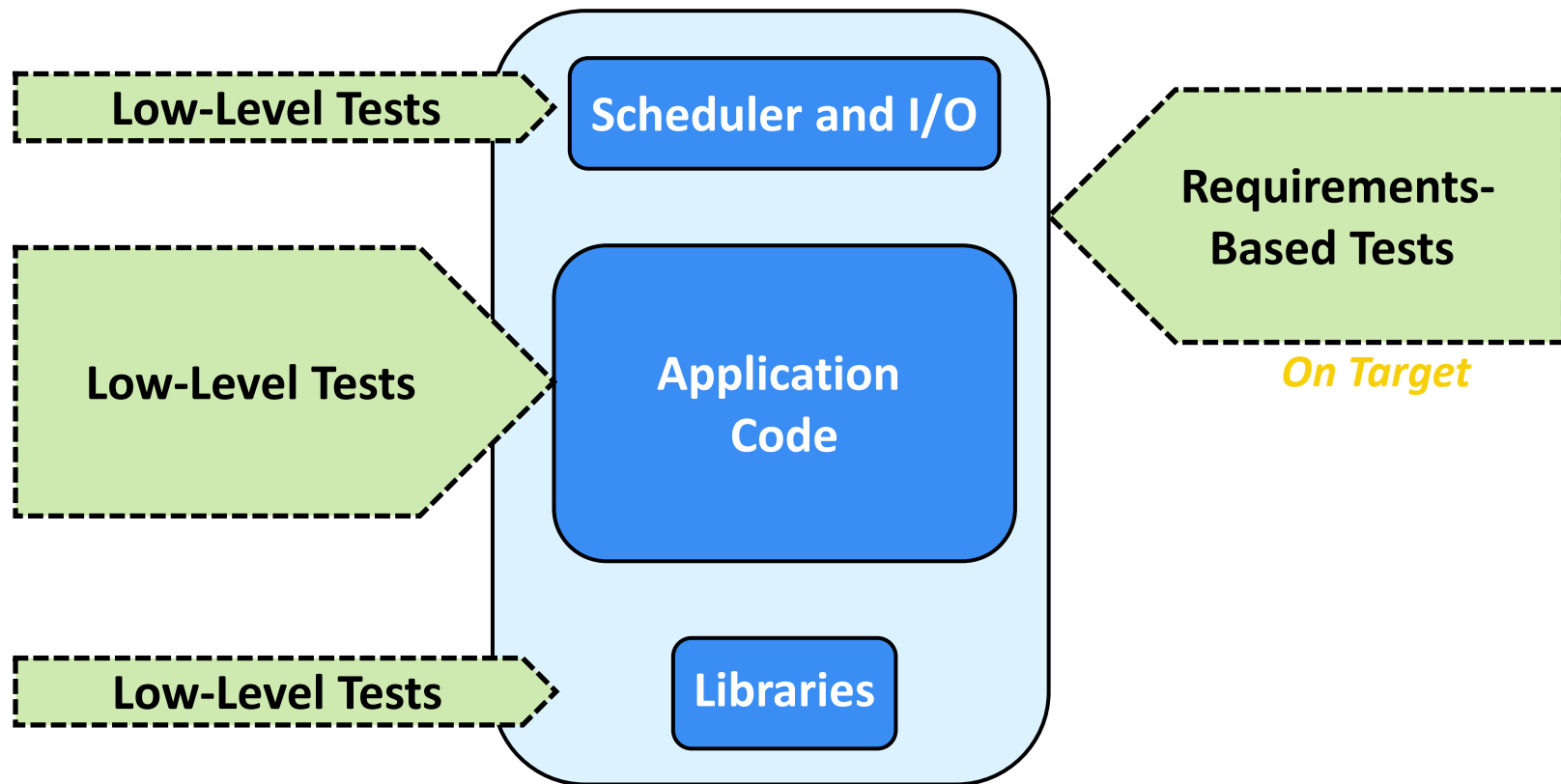
“Traditional” Testing Process



Low-Level Testing (a.k.a. Unit Testing) is applied to the 3 components of the application.

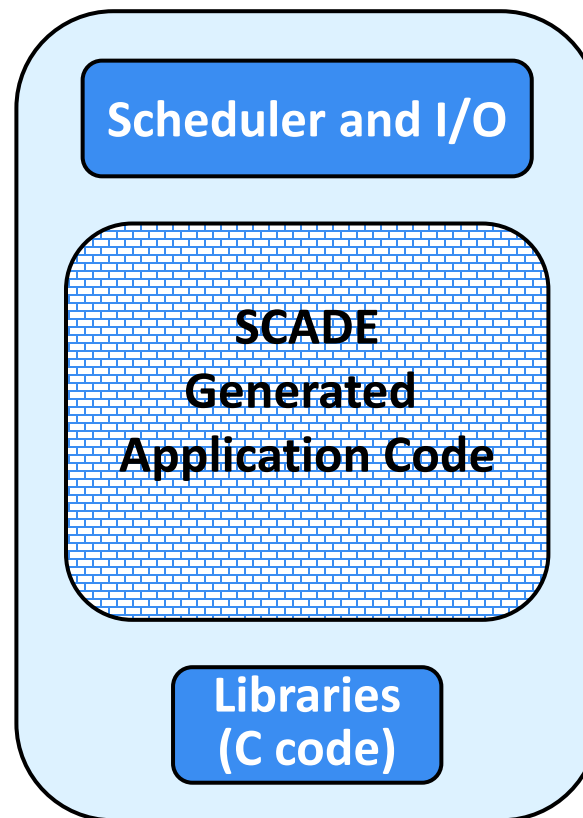
Low-level Testing of the Application Code is time consuming, requires updates each time there is code update (which is a standard situation), and the stopping criterion is not easy to establish.

“Traditional” Testing Process



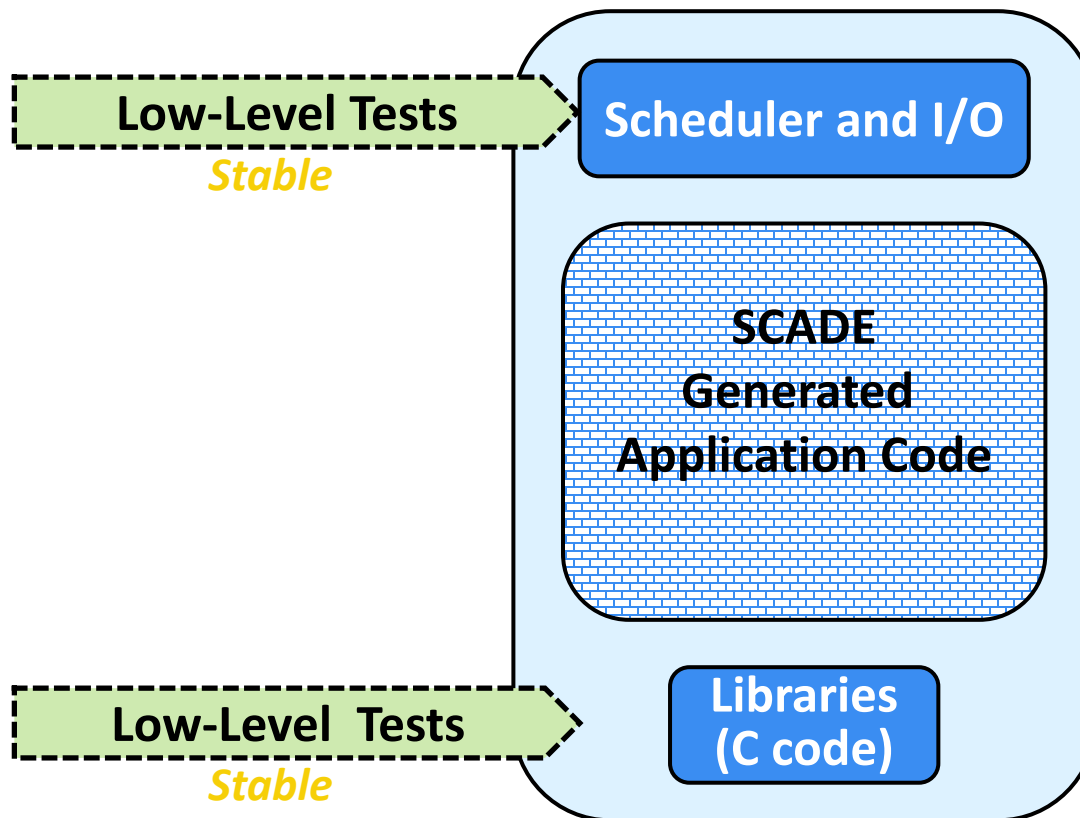
The requirements-based tests are coded in C, and/or in a specific target test tool format, and run on target; thus, they are debugged on target .

Target test tool structural coverage measurement can be the stopping criterion, generally mixed with low-level test coverage scores, sometimes not easy to justify.

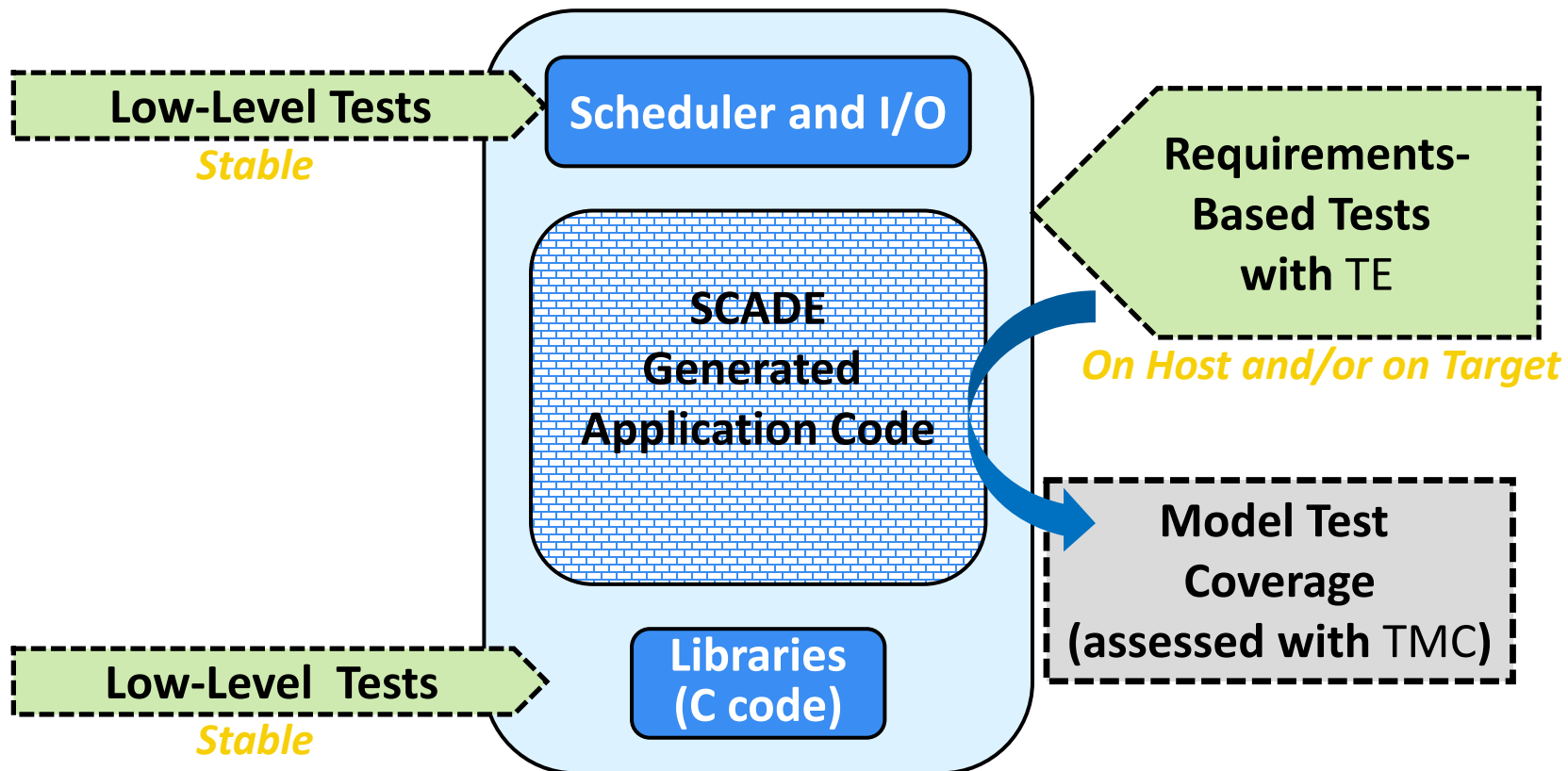


An embedded application designed with SCADE Suite is typically made of:

- A scheduler or an operating system + drivers for inputs and outputs
- The code of the application, **generated by SCADE Suite KCG**
- Libraries of some elementary imported operators (C code) that are repeatedly called in the SCADE model (SCADE libraries are part of the generated code).

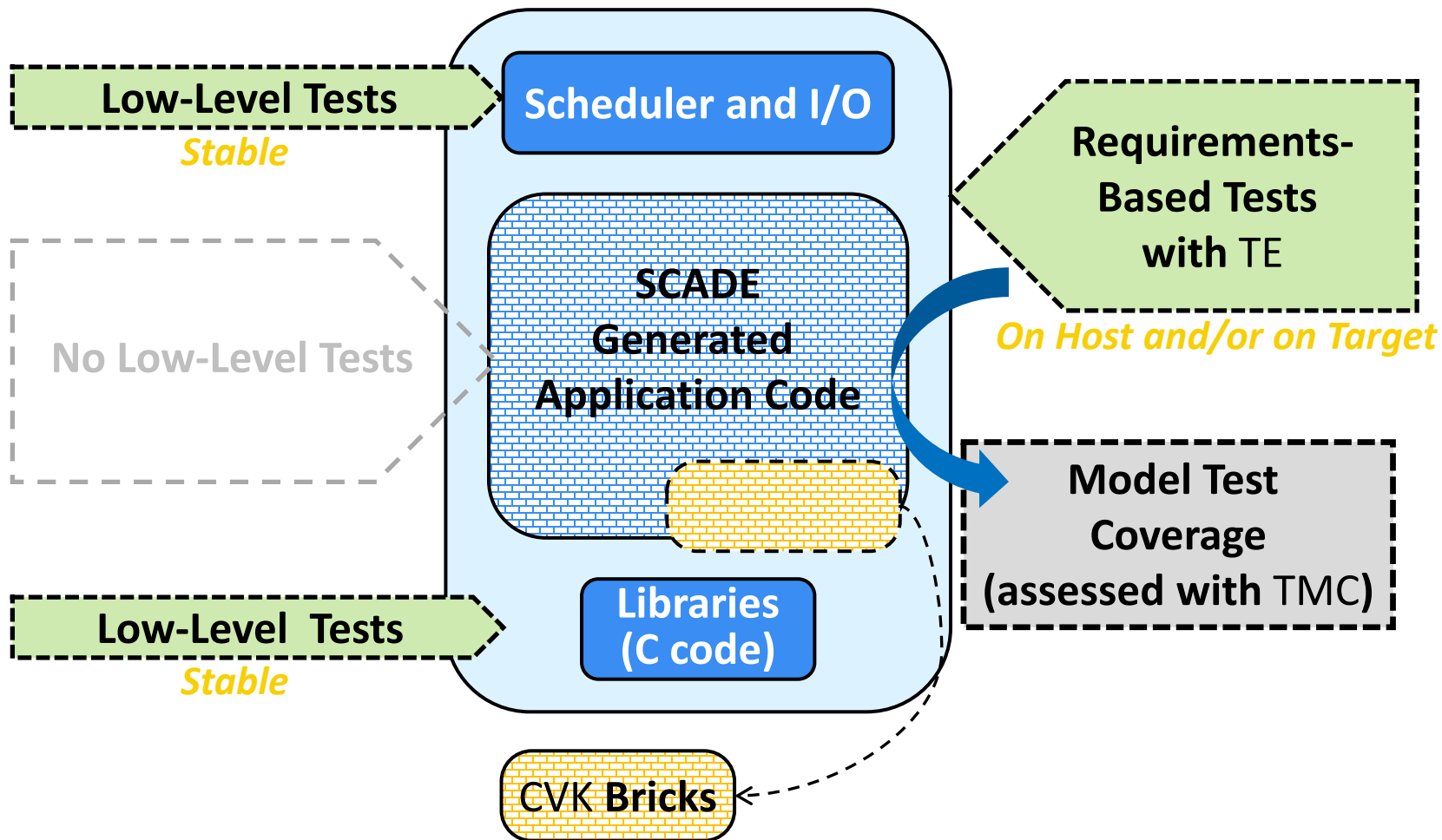


- No change in low-level testing against the scheduler and I/O (or OS) and against the libraries of C Code



The specification-based tests are created using SCADE TE and run on host (SCADE simulation) and/or on target (target testing) according to the test strategy that has been established.

SCADE Suite TMC is the unique means to monitor and stop the requirements-based testing activities.



SCADE Suite CVK is a low-level test suite used to perform unit testing on target of all the C constructs of the SCADE generated application code.

- **SCADE Combined Testing Process (CTP)**
- **SCADE Test Environment (TE)**
- **SCADE Test Model Coverage (TMC)**

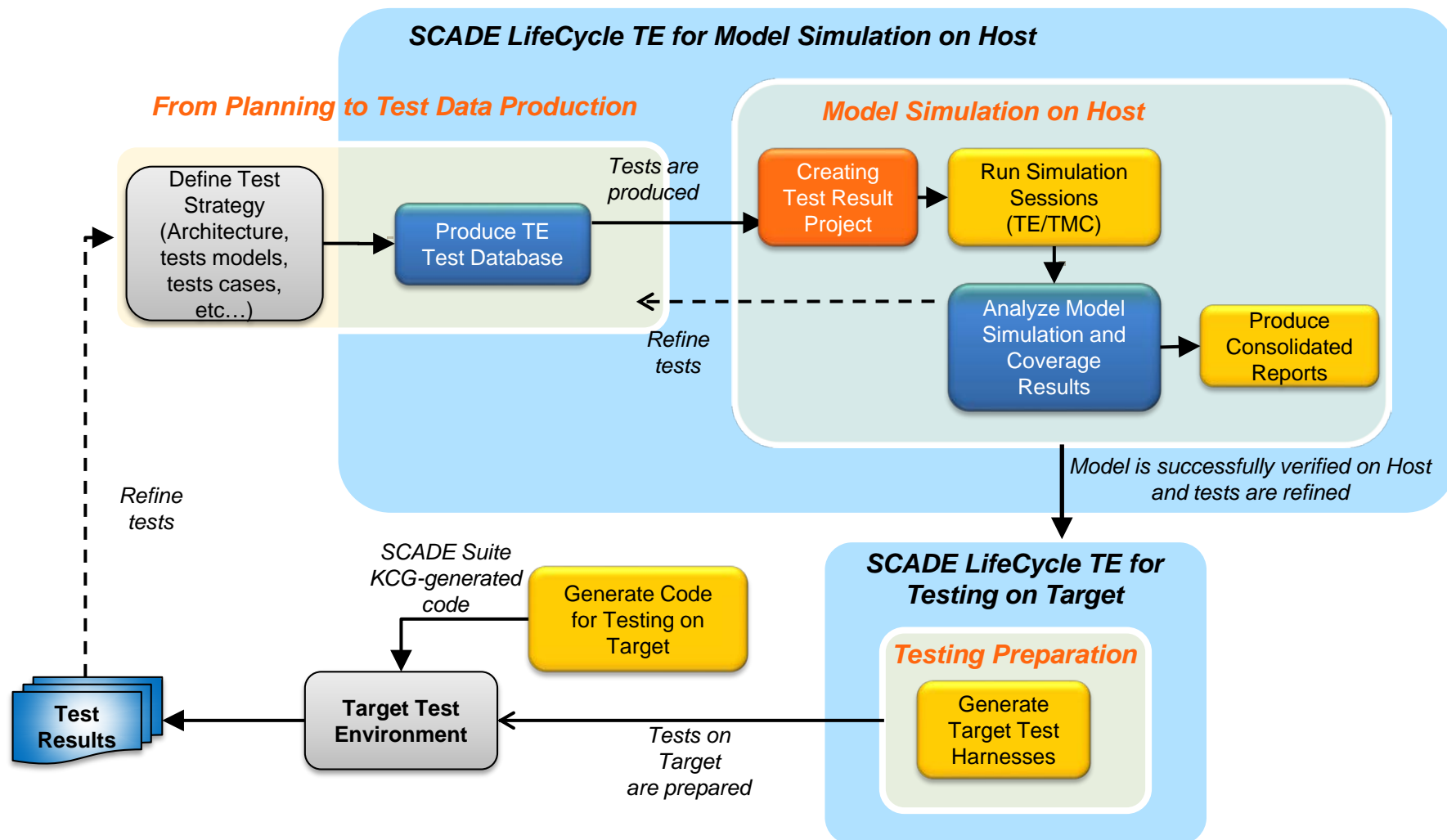
TE provides :

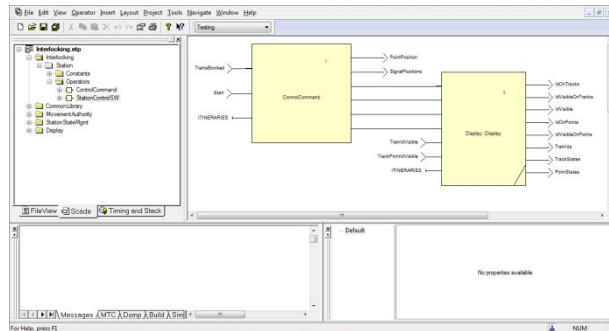
- Framework for **managing** the tests
 - Creation, organization, editing of tests
 - API, providing Test Data Access to scripts or third party tools
- **Automated test execution** on Host and on Target
 - Run of tests on host with the capability to get Model Test Coverage measurement
 - Production of test harness for the following list of COTS on-Target test execution tools:
 - IBM RTRT, LDRA TestBed, Vector Cast
- **Conformance Report Generator**, comparing actual results on host with expected results

TE is customizable for integration in any in-house or COTS Target test infrastructure

Certification Data:

- Qualified as a verification tool for DO-178C (TQL 5), Qualified Tool Class T2 for EN 50128,...





1. Create
Test Project

2. Edit Test
Project

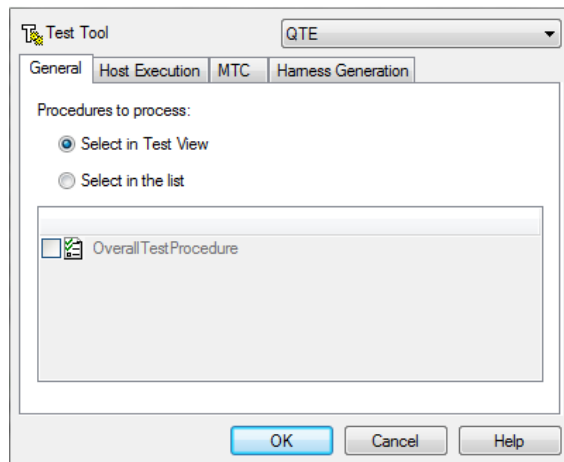


```
# Test Case Name: TC_N_01

#From line 2:
SSM::set Station::StationControlSW/TrackPointIdVisible false
SSM::set Station::StationControlSW/TrainIdVisible false
SSM::set Station::StationControlSW/TrainsBlocked (false,false)
SSM::set Station::StationControlSW/Start true
SSM::check Station::StationControlSW/TrackStates (CommonLibr
SSM::check Station::StationControlSW/PointPosition (CommonLi
SSM::check Station::StationControlSW/IdOnTracks (-1.0,-1.0,-
SSM::check Station::StationControlSW/IdVisibleOnTracks (false
SSM::check Station::StationControlSW/IdVisible false sustain
SSM::check Station::StationControlSW/IdOnPoints (-1.0,-1.0,-
SSM::check Station::StationControlSW/IdVisibleOnPoints (false
SSM::check Station::StationControlSW/SignalPositions (Common
SSM::check Station::StationControlSW/TrainIds (6083.0,3339.0
SSM::cycle

#From line 3:
SSM::set Station::StationControlSW/TrackPointIdVisible false
SSM::set Station::StationControlSW/TrainIdVisible false
```

3. Launch Test
Tools



4. Reports

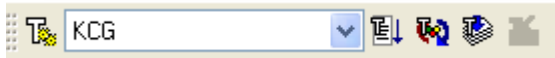
Qualified Test Conformance Report		
1. General Information		

Generated by	SCADE (TM) QTE Report R15.1 (build 130)	
Generation date	2014-08-04, 11:24:06	
Author		
Detail Level	full	
Exec. Directory	C:\Windows\system32	

2. Input Description		

ID	Filename	Project
Configuration		

	01 C:\Training\Base\SCADE Training Kit\trunk\	Q:\SCADE
Suite\Interlocking.etp	QTE	
	Process\	
	Realization of a Railway Application Compli	

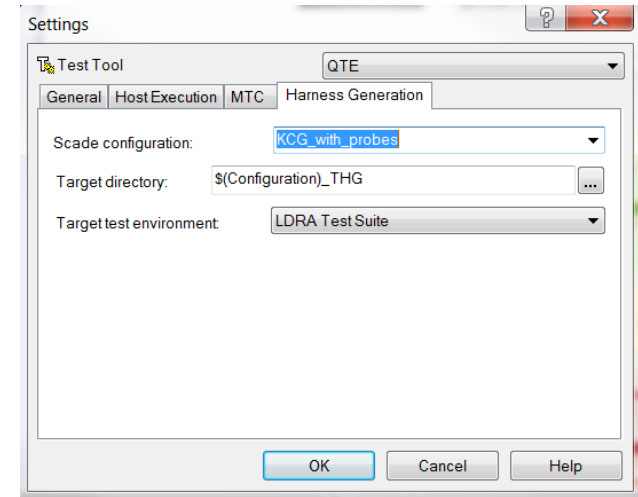


6. Generate Test Harness

```

-- Generated by: SCADE (TM) QTE Test Harness Generator R15.1 (build i30)
-- Date: Mon Aug 04 14:28:14 2014
--
-- Execution directory:
-- Q:\Tests\OverallTests\OverallTestResult
--
-- Invocation:
-- QTETHG.exe <options> Q:\SCADE Suite\Interlocking.etp
-- <options>: -target RTRT
--            -conf KCG
--            -test_file Q:\Tests\OverallTests\OverallTestProject\OverallTestProcedure.stp
--            -target_dir Q:\Tests\OverallTests\OverallTestResult\OverallTestResult\QTE_THG
--
-----
HEADER StationControlSW_Station, .
## include "kcg_consts.h"
## include "kcg_sensors.h"
## include "StationControlSW_Station.h"
# inC_StationControlSW_Station thg_inC;
# outC_StationControlSW_Station thg_outC;
BEGIN
SERVICE StationControlSW_Station
TEST TC_N_01
COMMENT Record1
# StationControlSW_init_Station(&thg_outC);
-- Record step 1
ELEMENT
-- input values
VAR thg_inC.TrackPointIdVisible, init = kcg_false, ev ==
VAR thg_inC.TrainIdVisible, init = kcg_false, ev ==
ARRAY thg_inC.TrainsBlocked, init = {kcg_false, kcg_false, kcg_false}, ev ==

```



Qualified Test Conformance Report			
1. General Information			

Generated by	SCADE (TM) QTE Report R15.1 (build i30)		
Generation date	2014-08-04, 11:24:06		
Author			
Detail Level	full		
Exec. Directory	C:\Windows\system32		

2. Input Description			

ID	Filename	Project	
Configuration			

01	C:\Training\Base\SCADE Training Kit\trunk\	Q:\SCADE	
Suite\Interlocking.etp	QTE		
	Process\		
	Realization of a Railway Application Compli		

5. Select Target:

- RTRT
- LDRA Tb
- ...

- **SCADE Combined Testing Process (CTP)**
- **SCADE Test Environment (TE)**
- **SCADE Test Model Coverage (TMC)**

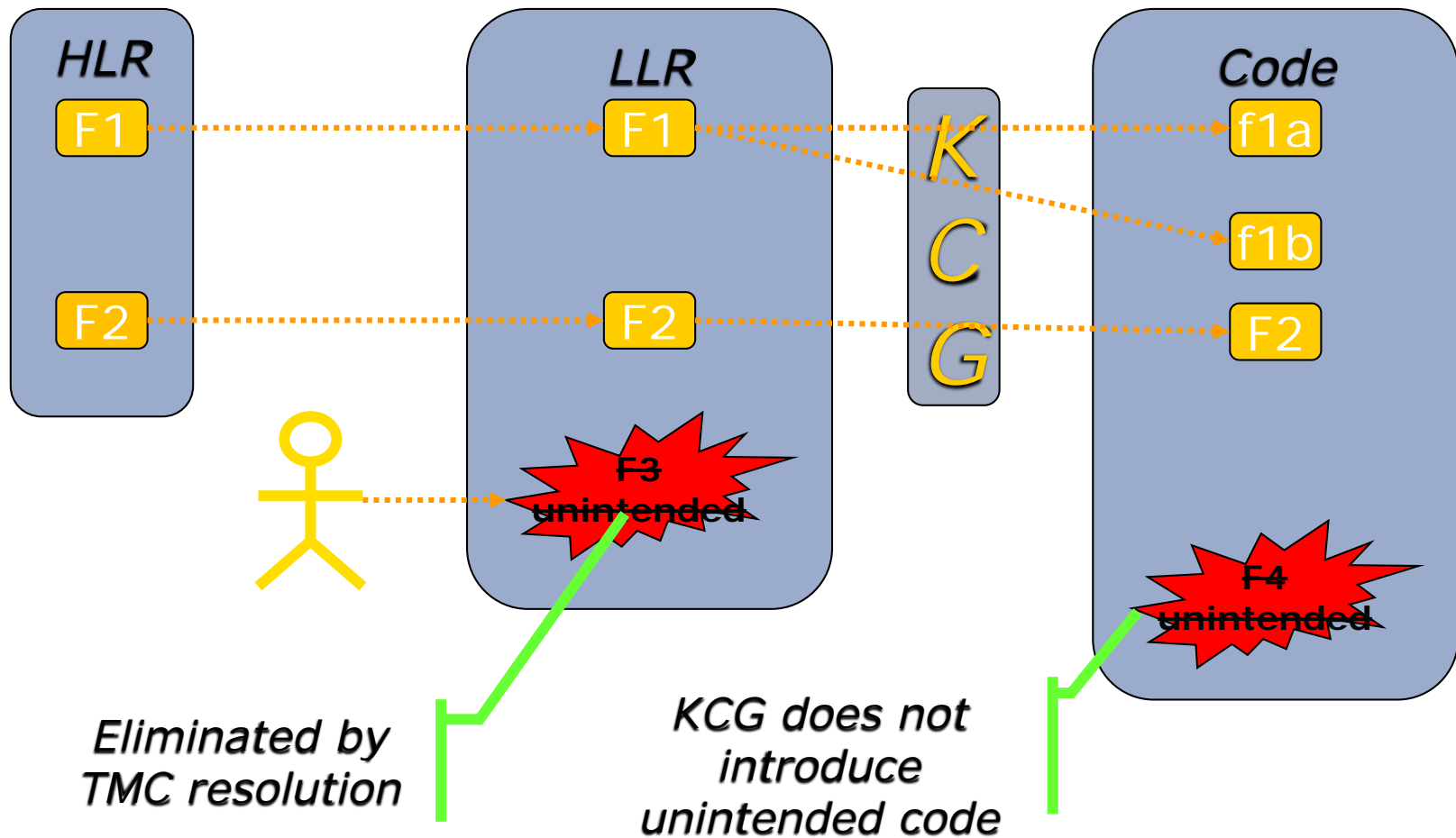
Model coverage analysis should be performed using high-level requirements-based test cases

- Coverage of the higher-level
 - Are **all** higher-level requirements correctly implemented in the SCADE Suite model?

Purpose of model coverage is to detect unintended functions in the model

- Coverage of the SCADE Suite model
 - Does the software implement **only** the higher-level requirements, i.e. does it contain no unintended functions?

TMC detects unintended functions



The purpose of SCADE Suite Test Model Coverage (TMC) is to assess how thoroughly a SCADE Suite model has been exercised: it measures LLR coverage

The Coverage Criteria are based on the observation of operator features activation

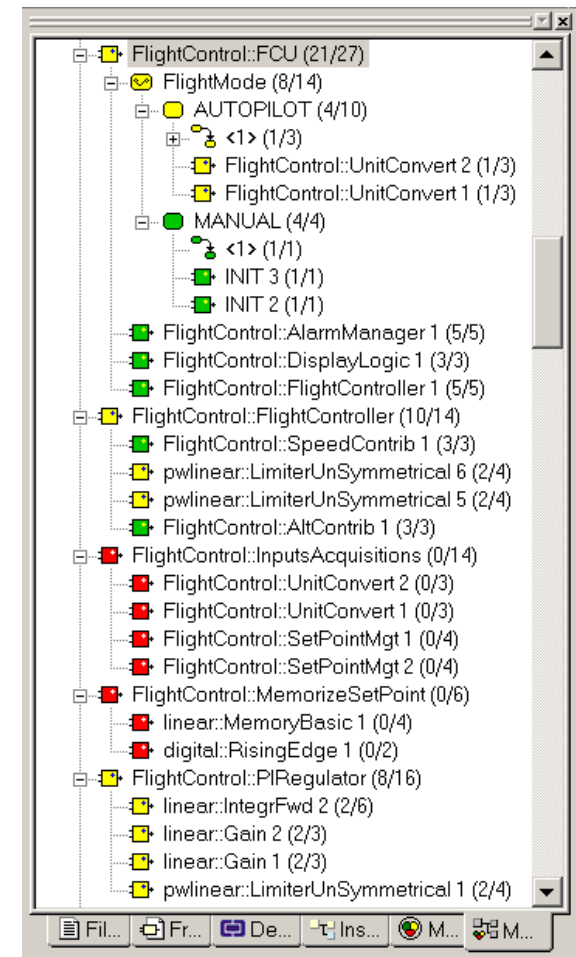
- **A SCADE Suite model is a network of operators**
- **Each operator has a set of characteristic ‘features’**
 - **Typically one feature per qualitatively different behaviors**

Coverage Analysis is performed at the level of each instance of an operator

Reveal portions of model that have not been activated

Identify deficiencies in SCADE Suite Design:

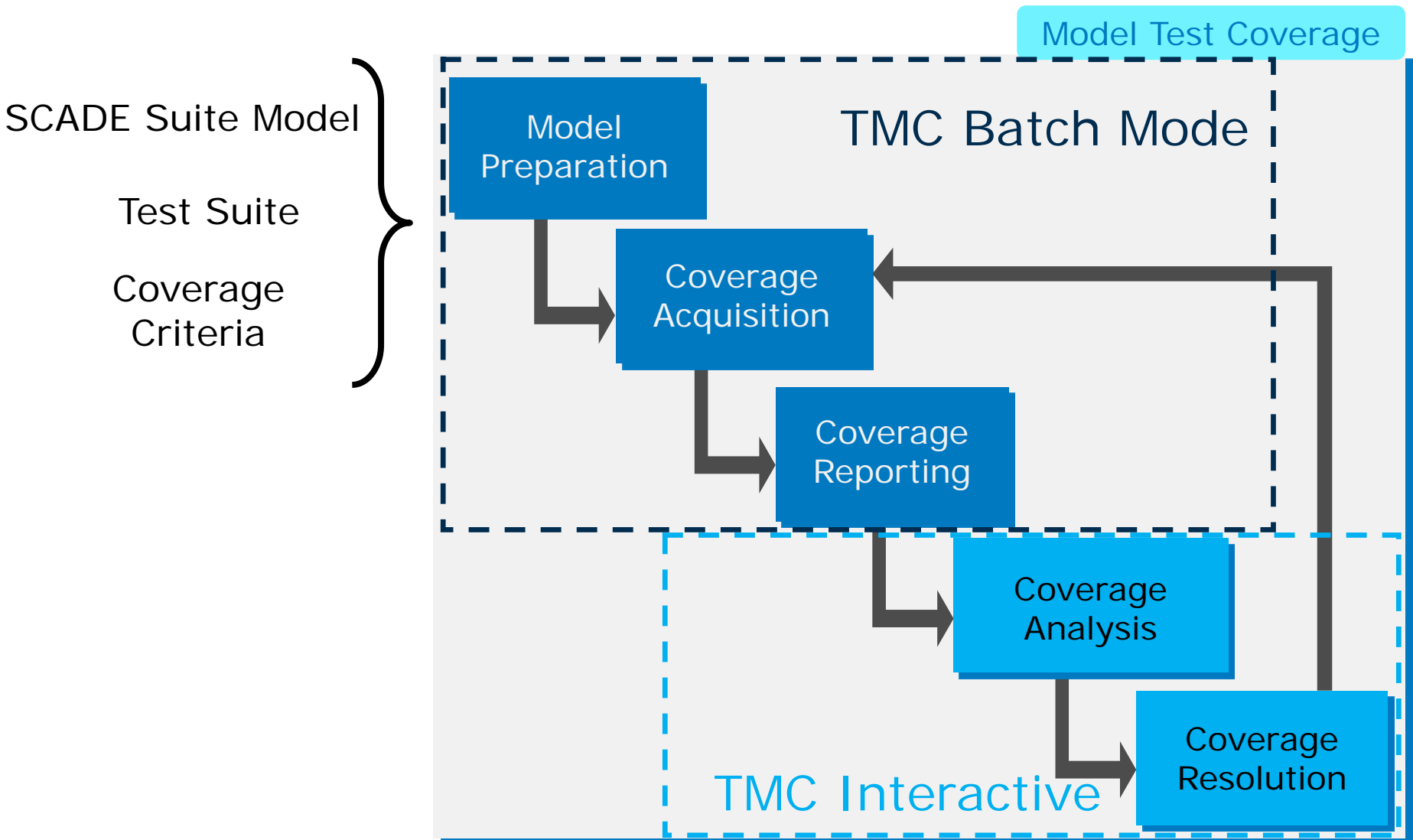
- Shortcomings in requirements-based verification cases or procedures
- Inadequacies or shortcomings from which the SCADE Suite model was developed
- Derived requirements expressed by the SCADE Suite model
- Deactivated functionality expressed by the SCADE Suite model
- Unintended functionality expressed by a SCADE Suite models



Acquisition Records	Nominal	Reseted	Over Saturated	Under Saturated
PI	X			X
PI_corrected	X		X	X
PI_justification		X		

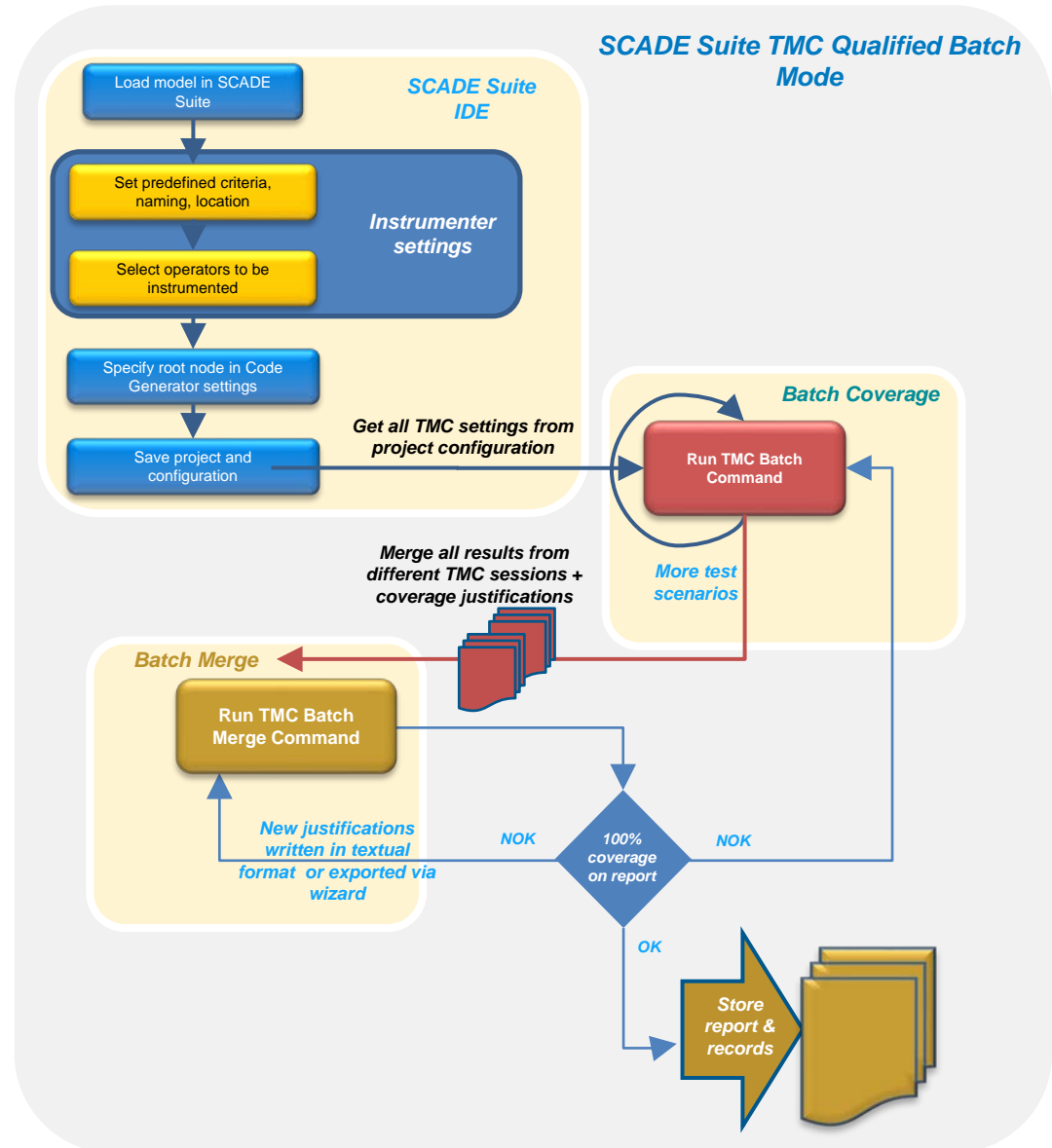
The SCADE Suite Model Test Coverage (TMC) tool set includes:

- A **qualified TMC Reporter** batch tool
 - Provides formal testing capabilities with Test results computation on host and model coverage measurement
- A **qualified TMC Merge** batch tool
 - Allows merging several intermediate coverage record files collected from different testing sessions, in particular made on different root operators
- A **TMC interactive** tool (not qualifiable)
 - Provides interactive white-box simulation capabilities with model coverage analysis in a user-friendly way, essential to be able to efficiently understand how to achieve the coverage

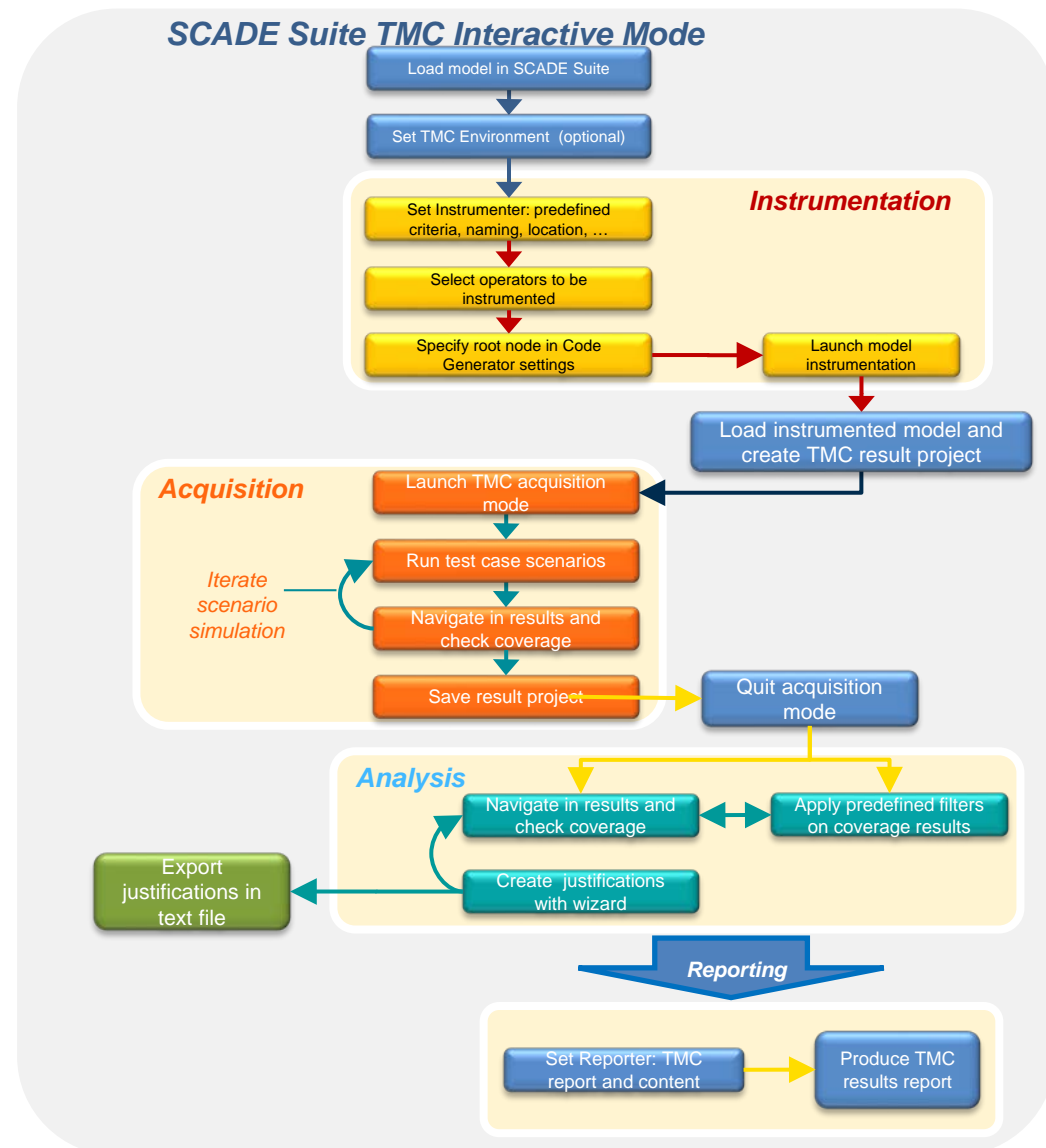


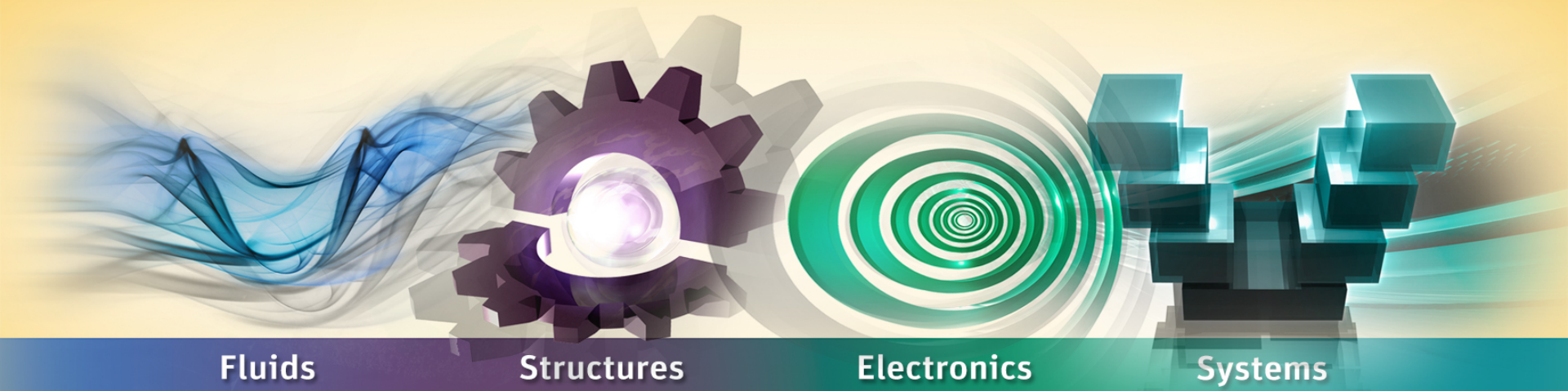
Setting Project

- Test scenarios, Configuration Settings
- Running Batch Coverage
 - Run several Coverage Acquisitions
- Running Batch Merge
 - Merging coverage results of several coverage acquisition
- Storing Batch output



- Instrumenting Project
- Launching Acquisition
 - Preparation, Execution and refinement of scenario test cases
- Performing coverage analysis
 - Resolution of coverage
- Reporting





Thank you for your attention