DISH LMC Software
An Idea for Qualification Plan

Francesco Schillirò
INAF- OACT
LMC Harmonisation Workshop
Madrid 11-13 April 2016

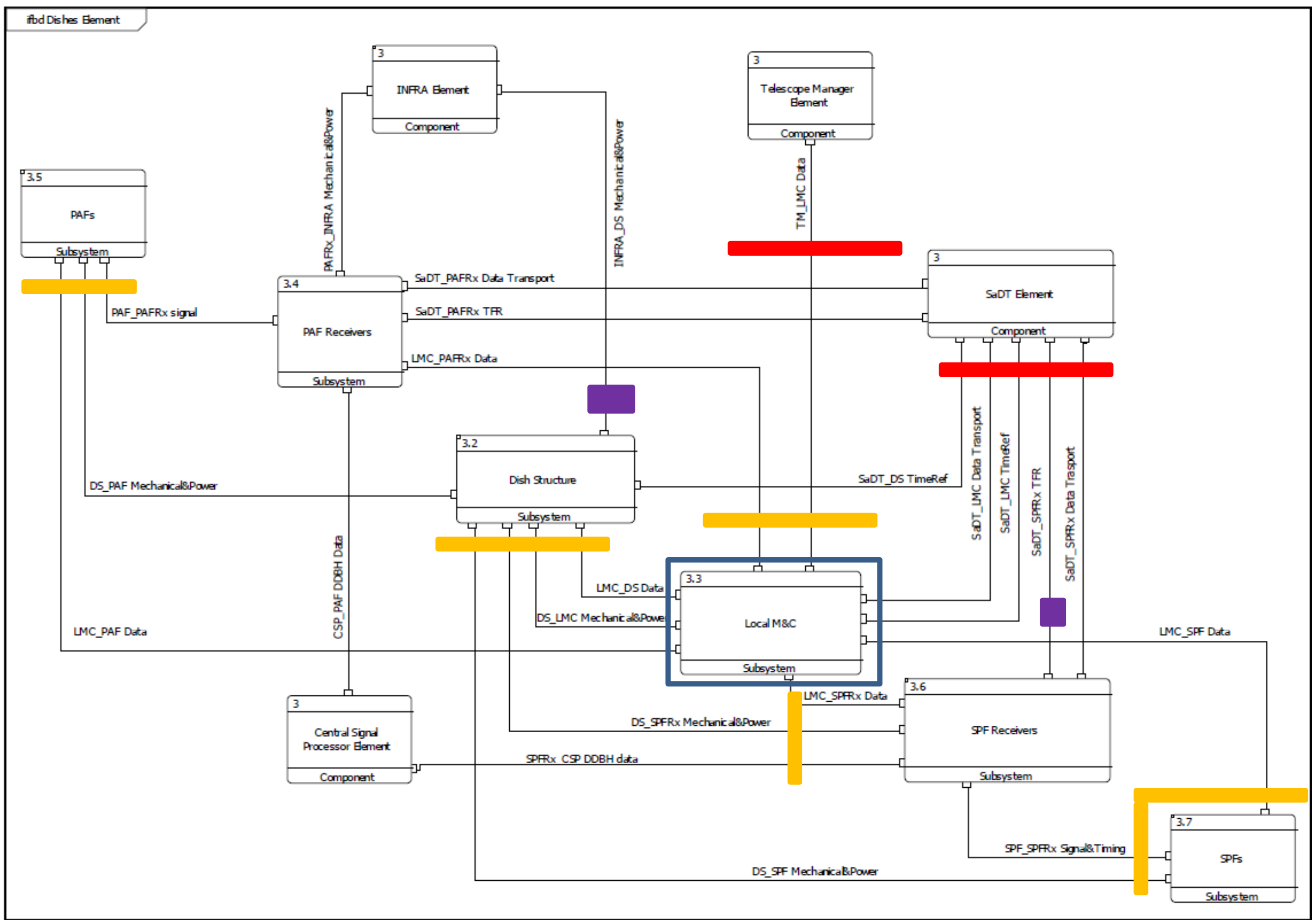
SKA
SQUARE KILOMETRE ARRAY
DISH

Functional and Fata Flow

SKA SQUARE KILOMETRE ARRAY

DISH

MID and Survey

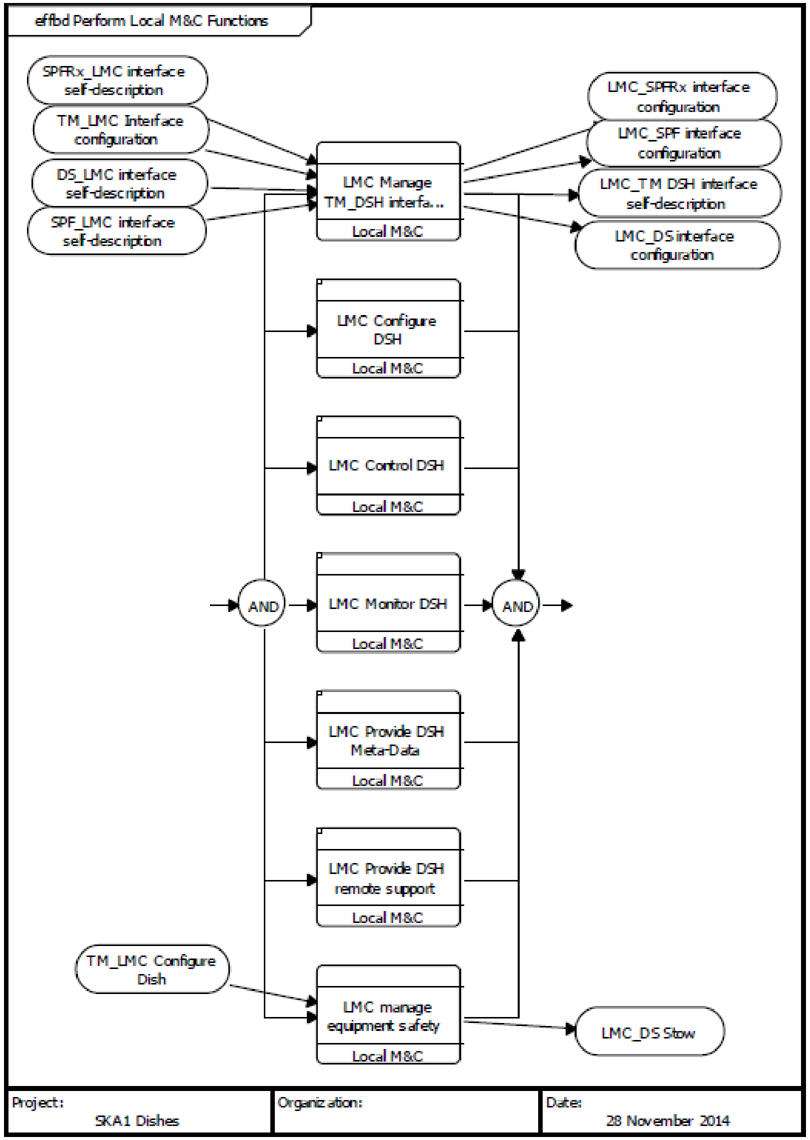


Internal Interface (sub-elements)

External Interface

Encapsulated Interface

Functional Breakdown

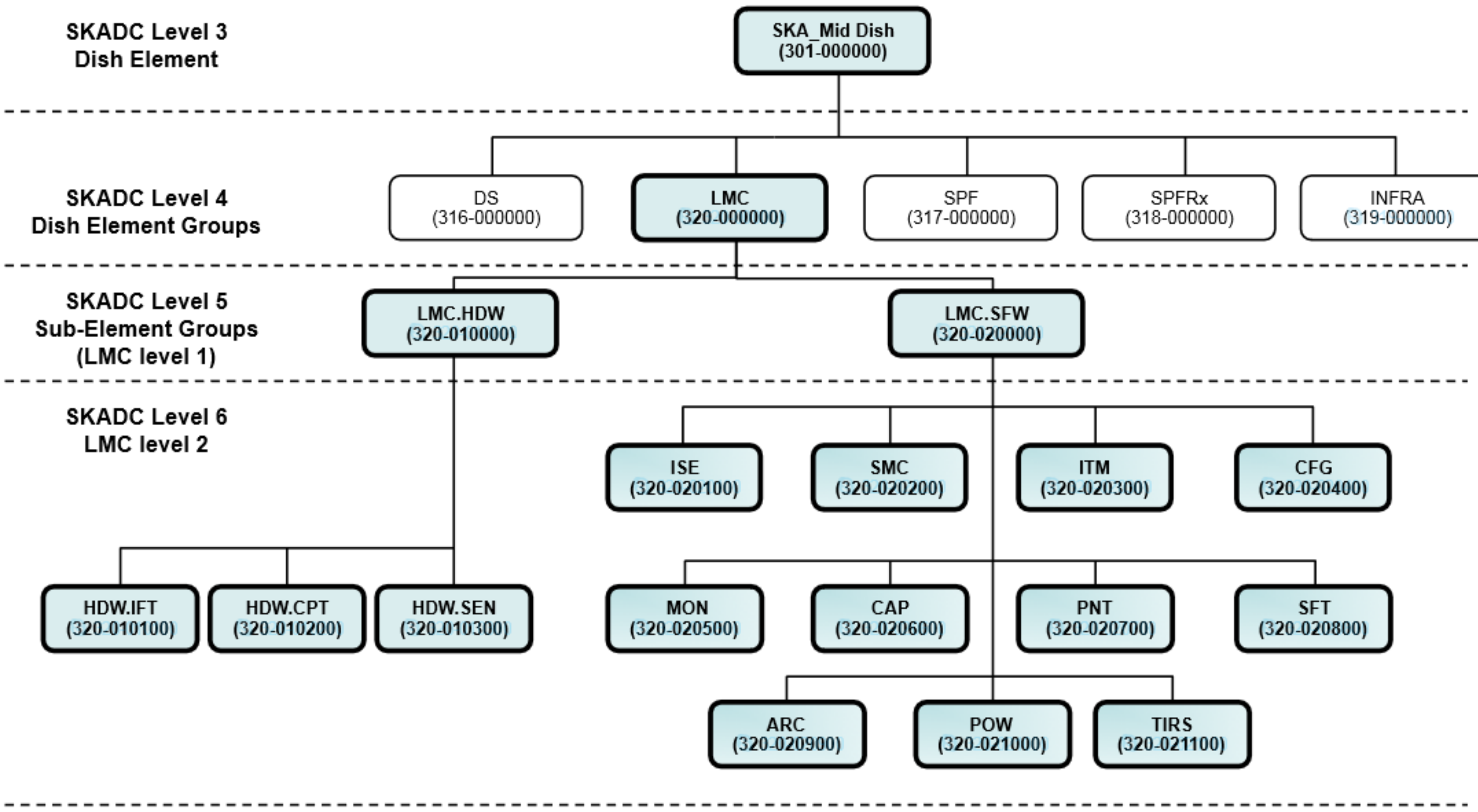


Functional breakdown

The following basic functions shall be implemented by the LMC:

- Managing the TM_LMC interface;
- Configuring all the components of the Dish in preparation for an observation;
- Real-time control of the Dish pointing and Beam forming during an observation;
- Monitoring of all Dish components and reporting of this monitoring information to the Telescope Manager;
- Sending meta-data to the TM that is required for the processing of signals;
- Providing functionality for the remote support of the Dish and all its sub-elements;
- Managing equipment safety;

Product Breakdown Structure



Component, Engine, leaf level

Breakdown Design Stage

PBS developement (leaf level)

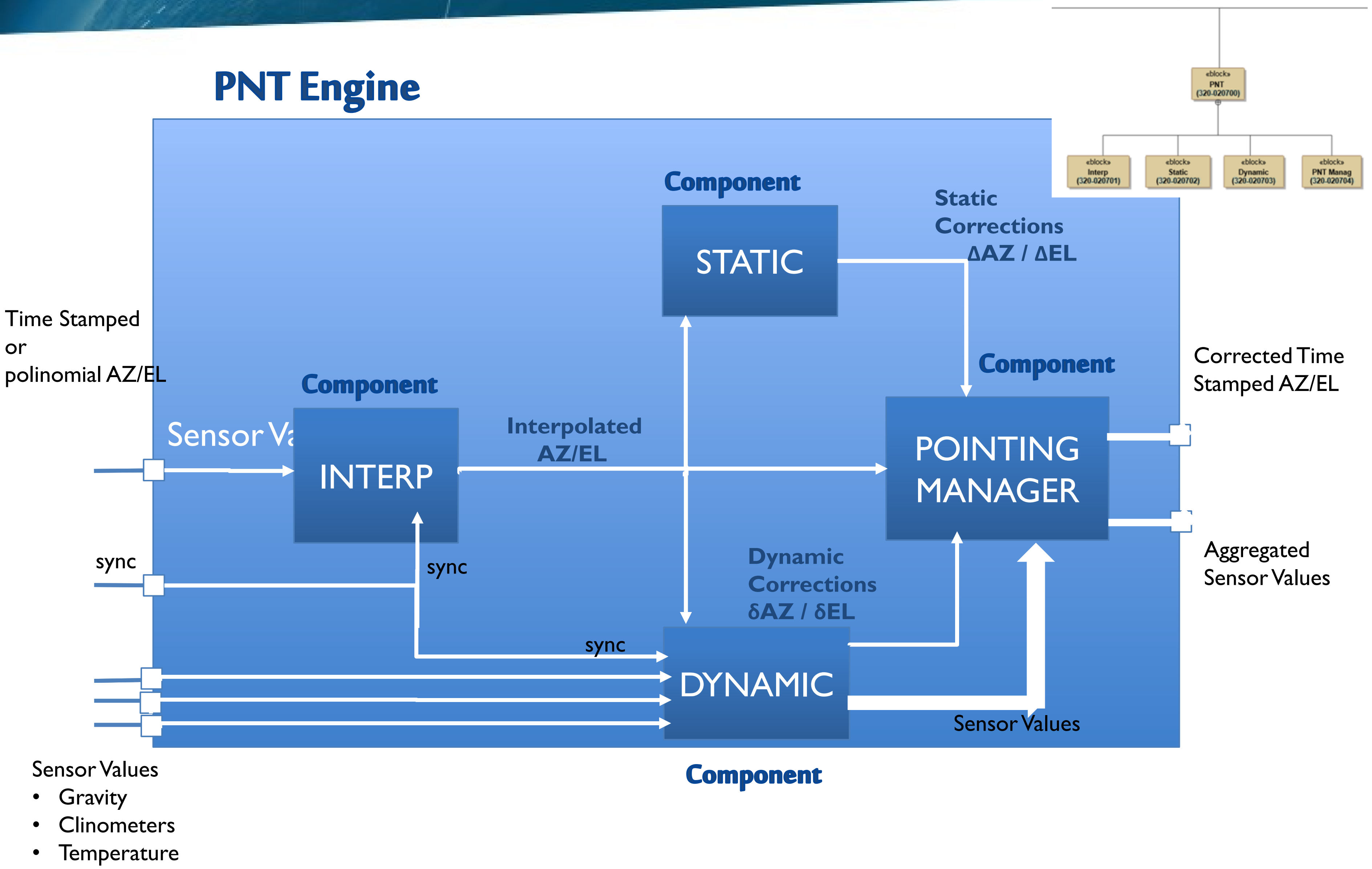
COMPONENT leaf level

-> simple and self standing functional unit;
-> from a point of view of design, component is an Object;
-> define Component member and functions;
-> define input and output members;
-> define sub-level design requiremements;

ENGINE

-> a functional aggregation of component ;
-> a higher level entity using components belonging to different PBS unit;
-> a high level input-output system ;
-> a functional entity ruled by high level requirements

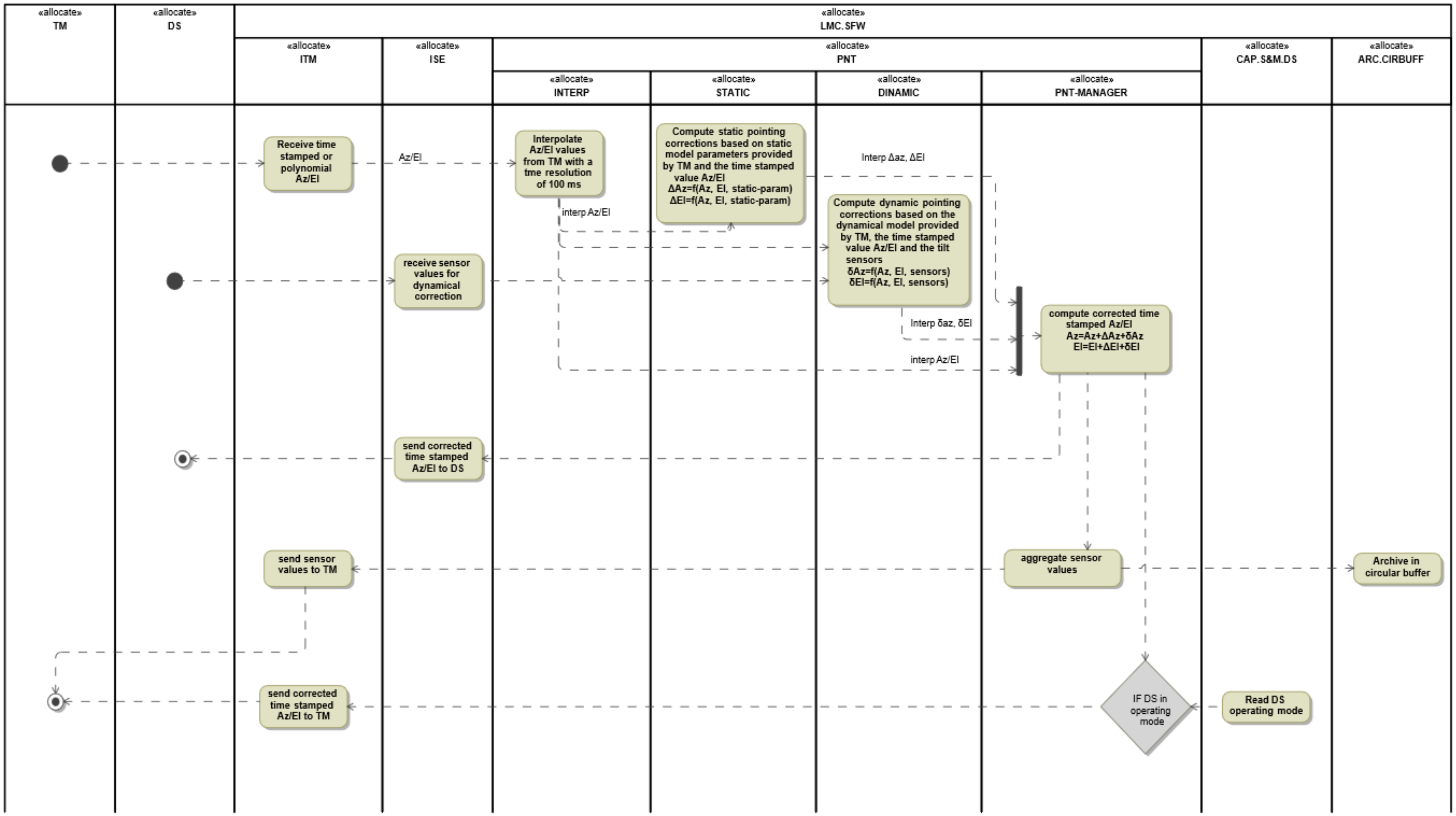
Component, Engine: example



Component, Engine: example

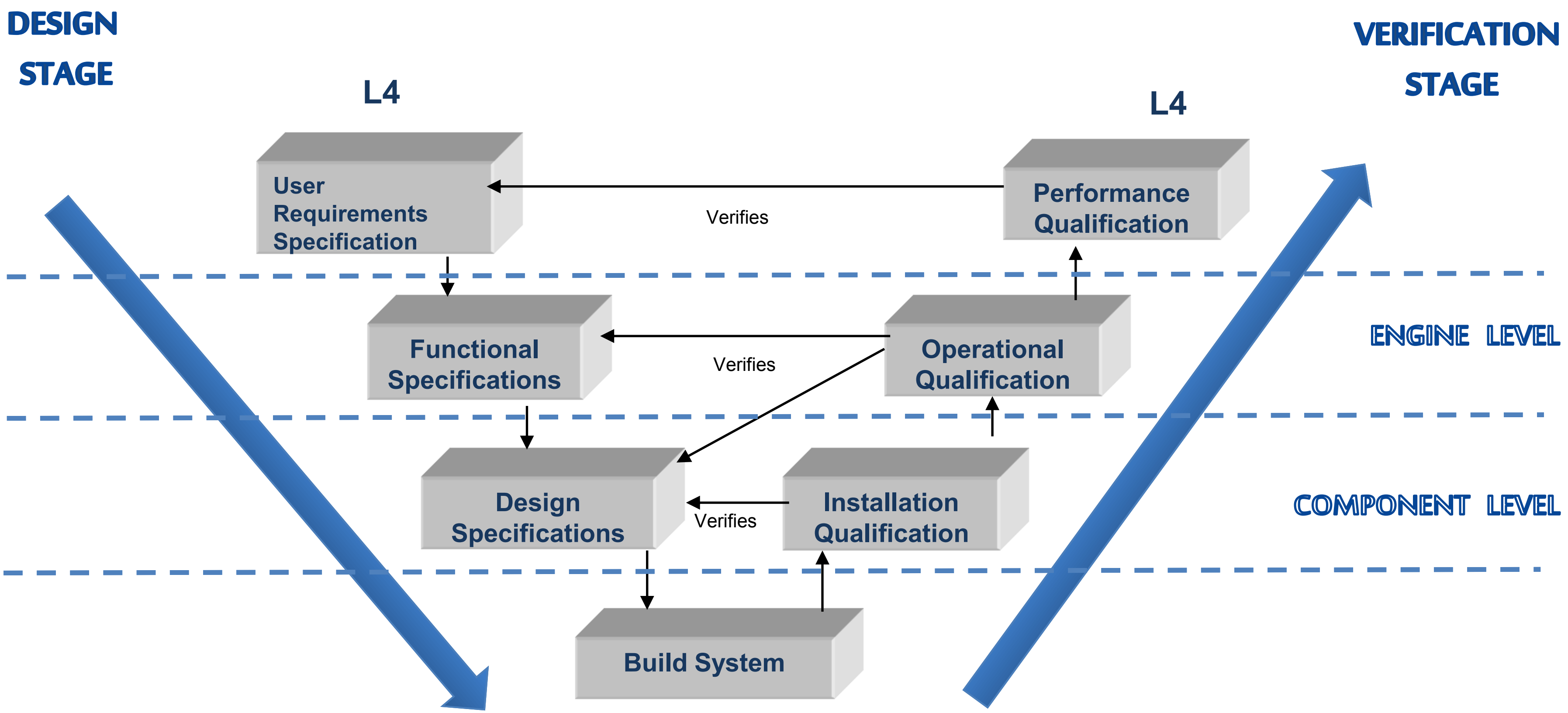
DISH

PNT Activity Diagram



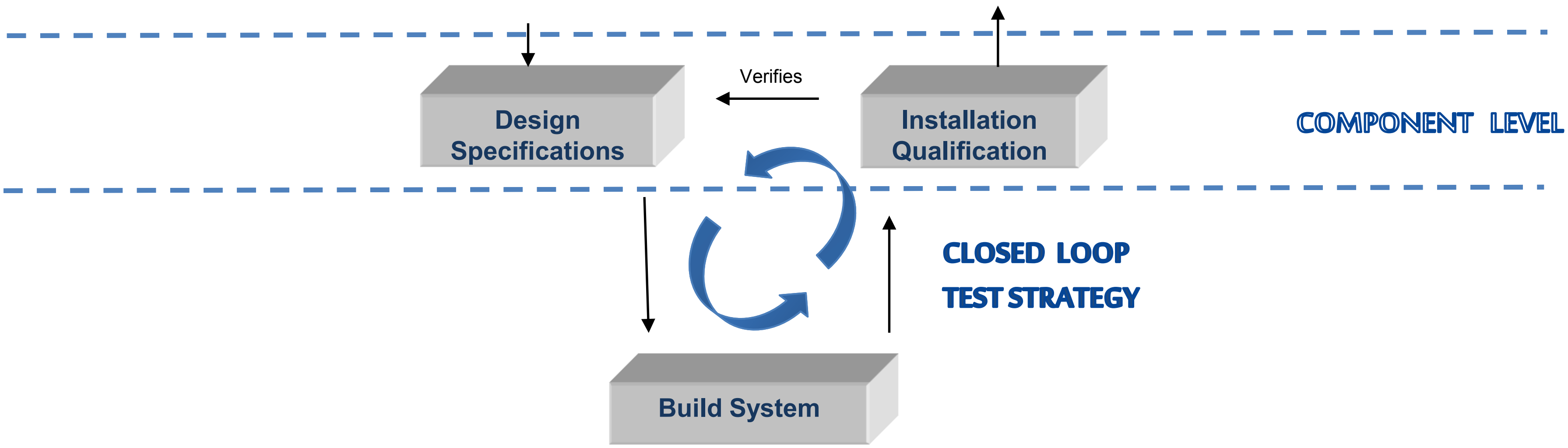
System Build and Verification Methodology

V Strategy



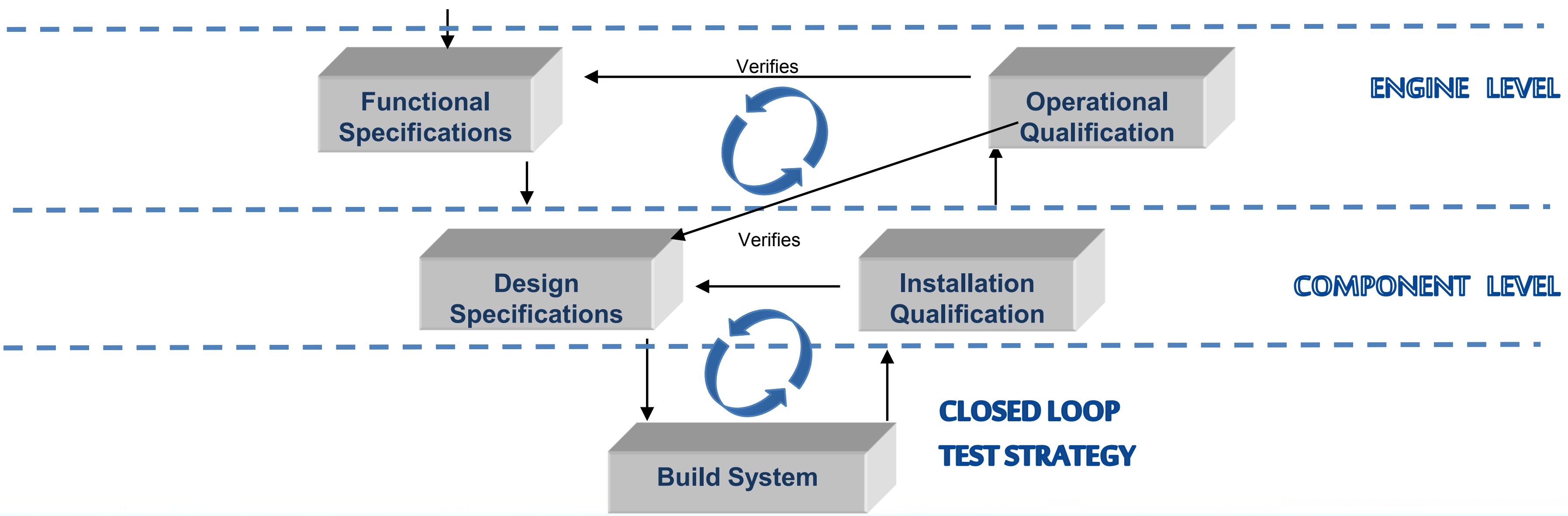
Bottom-up Verification Stage

- **Component verification (Installation)**
  - Input-Output analysis at Component
  - Closed loop Behavioural Test and debugging;
  - Component Resources Optimization (memory storage, RAM, algorithm)
  - Compliance with requirements at Component level



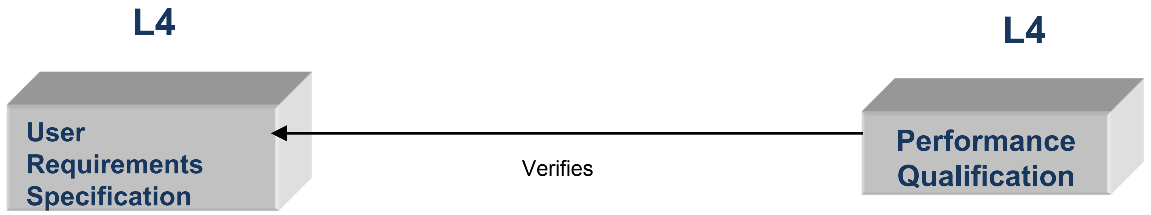
Bottom-up Verification Stage

- **Engine verification (Operational)**
  - Input-Output analysis at Engine
  - Double Closed loop Behavioural Test and debugging;
  - Engine Resources Optimization (memory storage, RAM, algorithm)
  - Compliance with requirements at Engine level



Bottom-up Verification Stage

- **Performance verification**
  - **Compliance with requirements at High Level (4 level , HDW+SFW)**
  - **LMC Resources Optimization (memory storage, RAM, algorithm)**
  - **External Interfaces verification (Sub-Elements, TM)**
  - **Performance Budget verification**
  - **Documentation**



Test Suite for Component and Engine

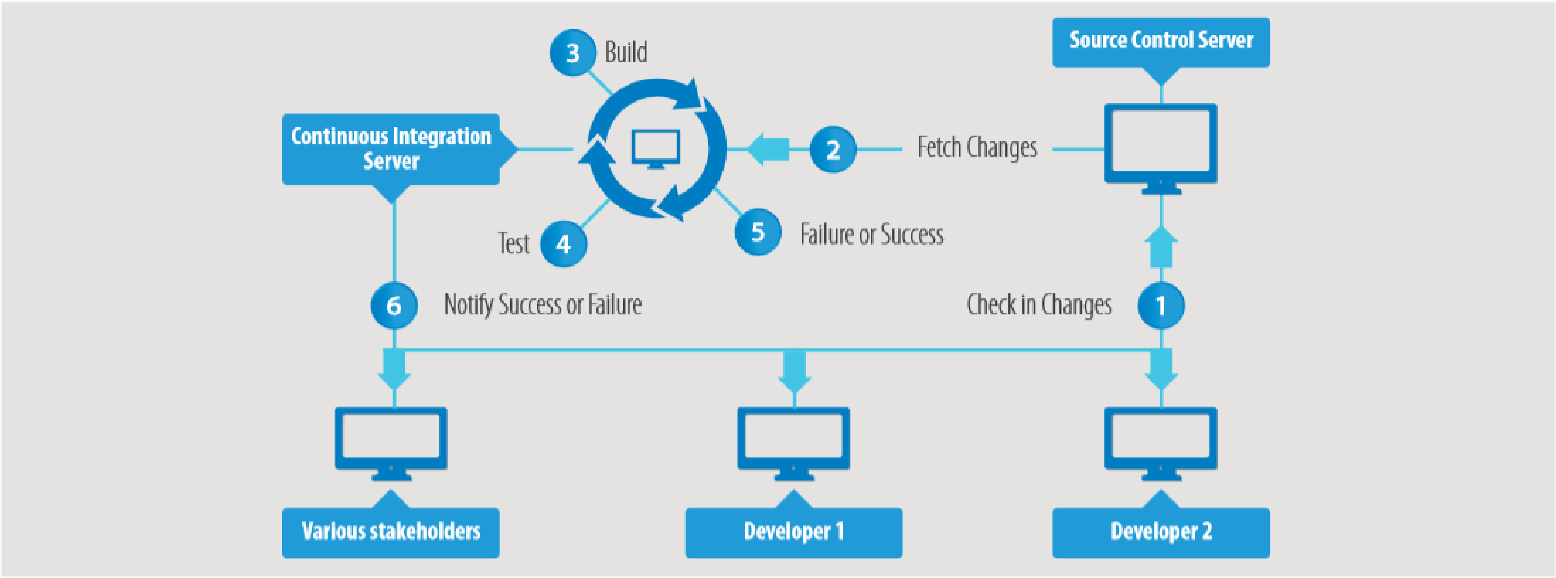
SKA SQUARE KILOMETRE ARRAY DISH

- **TM simulator suite**
- **SubElement Simulator suite**
- **Test bench for Components**
- **GUIs for testing Engine and Components**

Development tool:

- **TANGO**
- **LABVIEW**

Continuous Integration



Continuous Integration Setup

The below diagram illustrates the end to end Continuous Integration (CI) setup which is often used across projects. As seen in the figure, the main actors include the Development team, the Source Control Server and the Continuous Integration server.

Developers check-in the code into source control server which is integrated with CI server. For each build, CI server is configured to run functional test cases, code quality checks and provide notifications for any failure scenario which enables the development team to take immediate action. This continuous automation chain helps in reducing the overall defect density and thereby improving the code quality

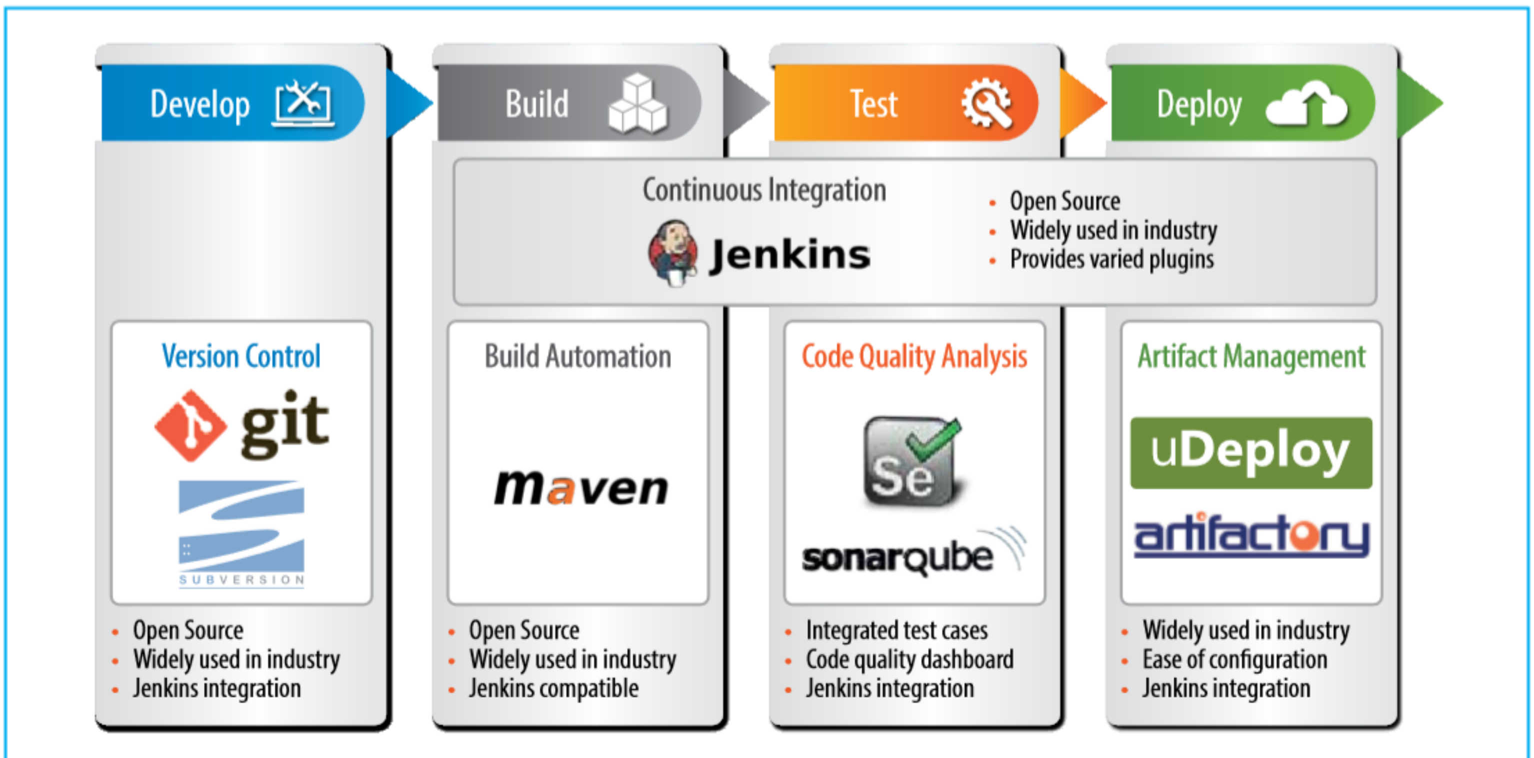
Continuous Integration : Tools

Tools Adoption

CI depends mostly on adopting the correct set of tools and their proper usage. The selection of tools is generally driven by various IT policies in the organization, existing technology landscape, current infrastructure setup, and other considerations. It is therefore recommended that every organization must do proper due diligence in evaluating different toolsets and choosing the appropriate ones suitable for their requirements.

The diagram below shows the toolsets (phase wise) which we have been using successfully across

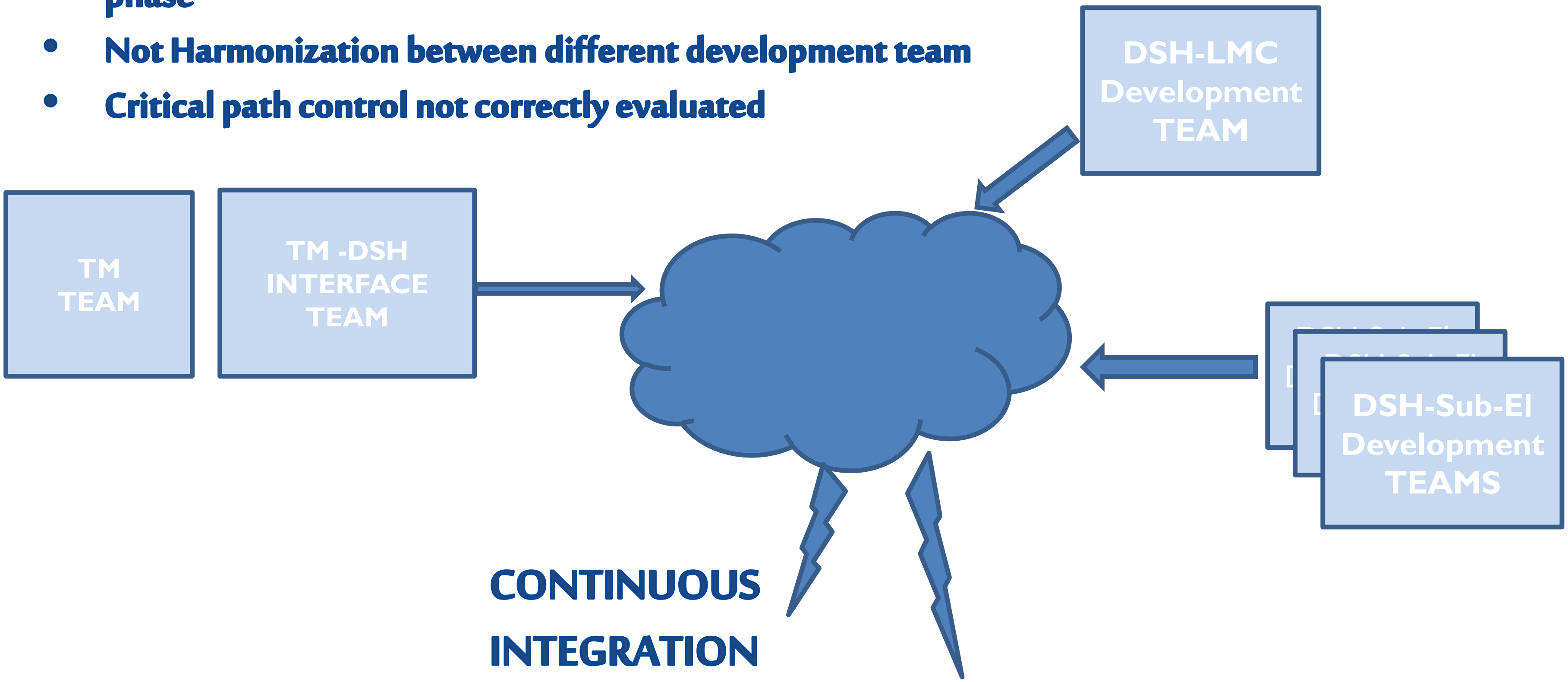
enablement. As seen, we have adopted Jenkins as the continuous integration platform but there are other CI platforms (Bamboo, TeamCity etc.) to choose from. The same case holds good for other toolsets also.



Continuous Integration : Risk Mitigation

RISKs:

- **Risk of mismatching interfaces parameters or procedure;**
- **Risk of temporal syncrhonization for different development times**
- **Long times for decision could be dangerous for implemetation plan,**
- **Change proposal at CDR phase for component design reflect dangerously into integration phase**
- **Not Harmonization between different development team**
- **Critical path control not correctly evaluated**



Verification Plan Document

DISH

LMC Qualification Plan: proposed structure & content			
1	Approach to detail design & implementation		- How will the software components be designed, specified, developed, integrated and tested - what is the methodology? (e.g. Unit testing, Component testing, Integrated lab testing, early integration testing with other sub-Elements, on-site set to work) - Release procedures - Version control and branching
2	Software development, integration & verification framework		Describe the build and test suite
			Describe the build and test methodology
			How will early integration testing with other Sub-elements be done?
			Will there be an engineering GUI to test the integration of the DSH and basic DSH control & monitoring?
3	Verification Requirements	(Test, Analysis, Demo, Inspection) traceability to all LMC requirements.	This will be a column in the Compliance Matrix and does not need to be repeated here, just reference to the compliance matrix.
4	Major Qualification events	Functional testing	Define: - Where the final formal qualification testing will be done (probably in a lab in Italy) - Test configuration ((e.g. in lab with test suite and simulators for external interfaces.) - High level description of tests to be done
		Environmental qualification	Define: - Where this will be done? - What facilities to be used? - Test configurations - High level description of test procedure. Refer to ETSI and applicable IEC standards
		RFI	Define: - Where this will be done? - What facilities will be used for RFI testing? - Test configuration - High level description of test procedure. Use EMI test plan as a framework.
5	Qualification Project Plan		Timeline showing the above test events.
6	Identification of key integration and verification risks & mitigation strategies		

THANKS!