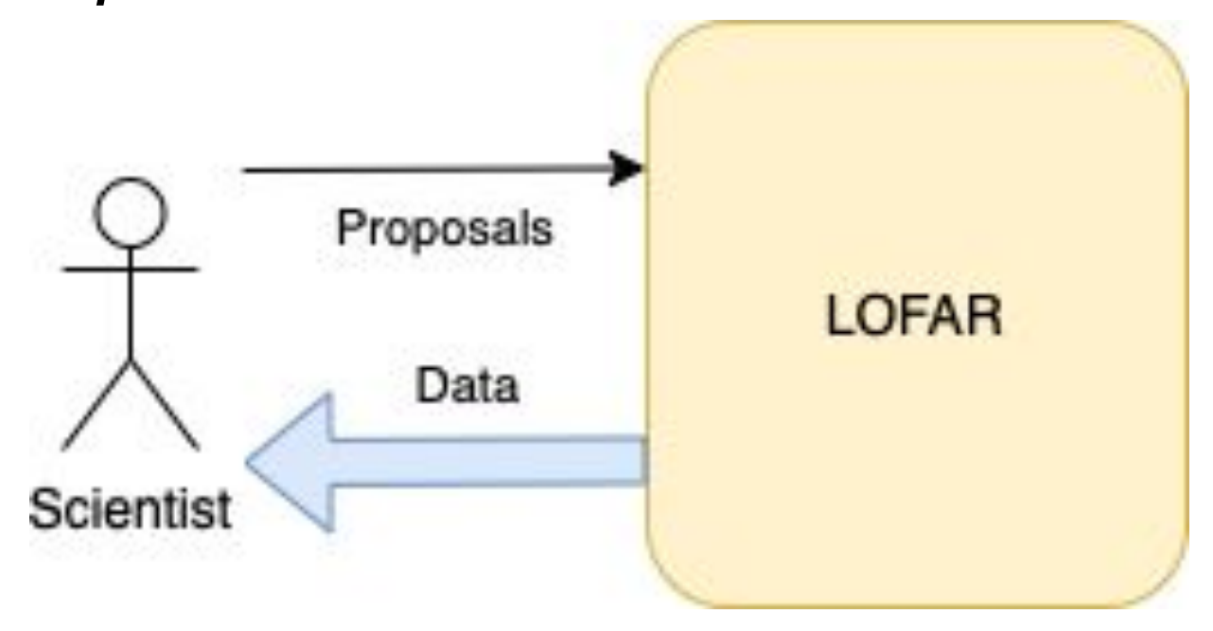
The TMSS project for LOFAR 2.0 operations

Mario Raciti (INAF - Osservatorio Astrofisico di Catania - mario.raciti@inaf.it),
F. Vitello, J. Schaap, R. Kumar, J. Künsemöller, M. Krishnan, A. Klazema, N. Santhanam,
R.d. Goei, H.D. Subbaramaiah, S.t. Veen, R. Pizzo, M. Iacobelli, J.D. Mol, T. Kamphuis

INTRODUCTION

The project LOFAR 2.0 has involved, among new changes and challenges, in the design and implementation of the **TMSS (Telescope Manager Specification System)** project, which supplies one interface and an integrated process for specification, scheduling and reporting, with data flow enhancements, that **improve the efficiency and automation of LOFAR operations.** TMSS can be seen as *a user who proposes specifications to LOFAR observations and receives results for such operations*.



SOFTWARE DESIGN

The architecture of TMSS (Fig. 1) is essentially composed of **data specification, dynamic scheduling, workflows** and **reporting**, and includes main features among which a new frontend interface for users and a new backend to interact and integrate within other LOFAR components.

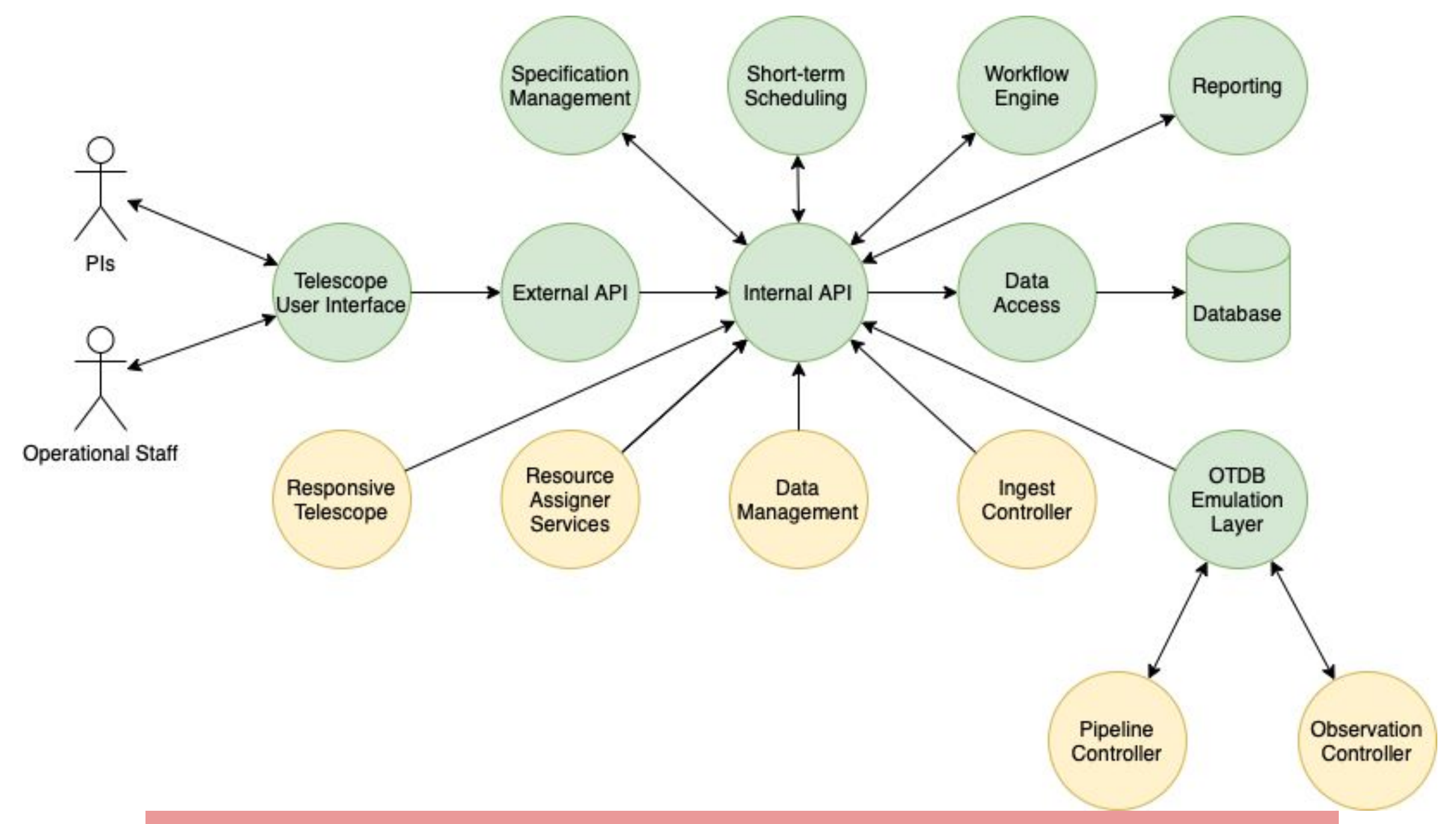


Fig. 1: System Architecture

State-of-the-art Web technologies (Fig. 2) are adopted for the implementation of the frontend (React) and backend (Django Rest Framework), including open source and modern de-facto global standards packages.

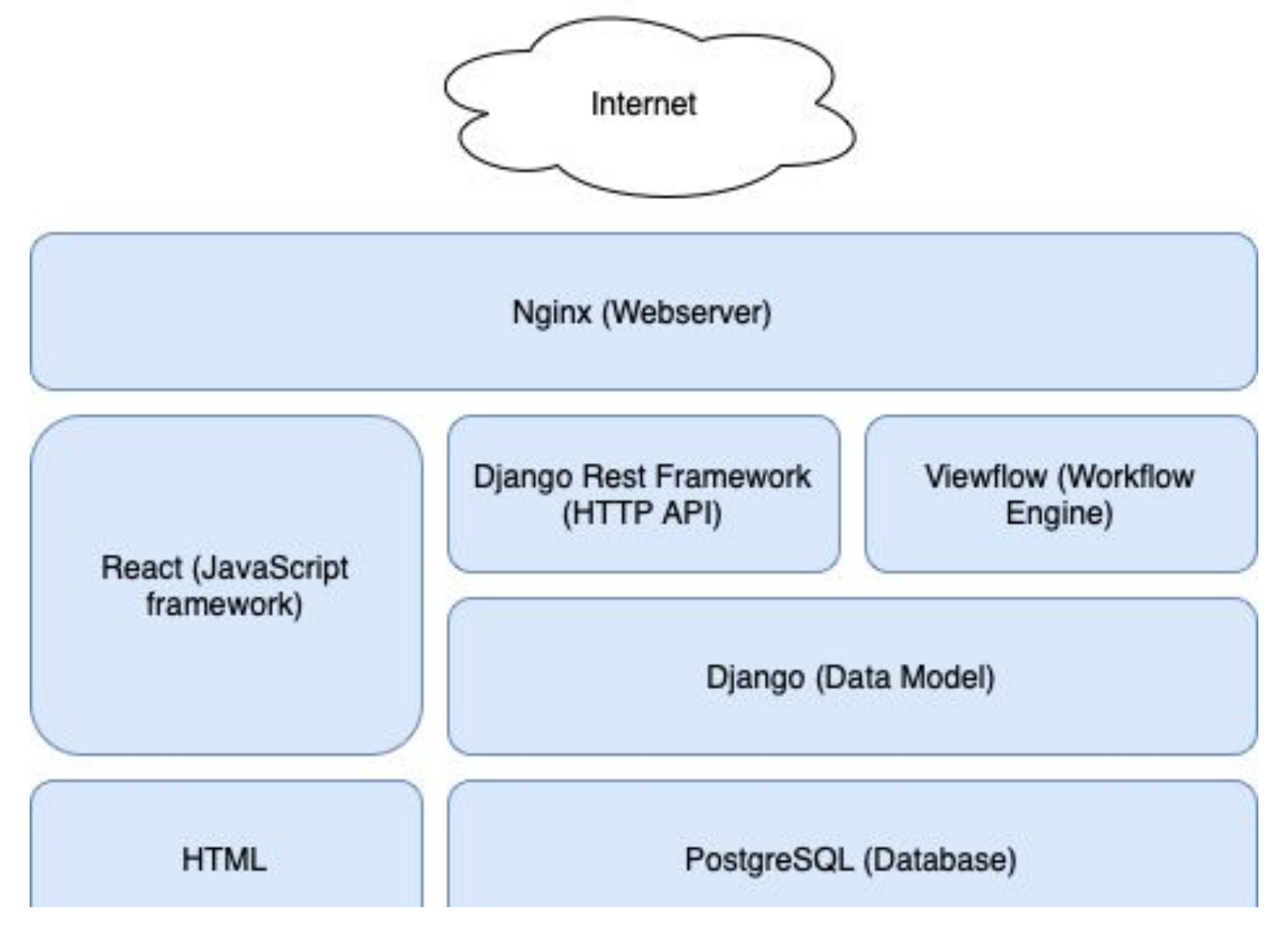

Internet
Nginx (Webserver)
React (JavaScript framework)
Django Rest Framework (HTTP API)
Viewflow (Workflow Engine)
Django (Data Model)
HTML
PostgreSQL (Database)

Fig. 2: Technologies Stack

DATA SPECIFICATION

The system supports a selection of standard observing strategies (JSON schemas), each requiring in itself only a minimal specification. We redesigned the functionality covered by *MoMDB/OTDB* to properly model both the capabilities of the telescope **(settings)** and to manage the work **(jobs)** required for operations. Thus, TMSS aims to replace MoM as an interface for checking specifications.

DYNAMIC SCHEDULING

TMSS implements a **dynamic scheduling mechanism to to optimise the operational procedures and make operations more efficient**, as tasks (observations, pipelines, and ingests) need to be scheduled to run when sufficient resources (e.g., the sky, processing, storage, bandwidth, etc.) are available. The most critical constraints are predicting and planning the required sky conditions and station resources for the observation. **The process of scheduling all these tasks is (partially) automated** by handling priority classes to use the telescope as efficiently as possible and to reduce operational cost. The dynamic scheduler also takes care of resources reserved for maintenance windows, in which the resources under maintenance will be unavailable for production.

QA WORKFLOW

The **quality assessment workflow (Fig. 3)** in TMSS helps to streamline the reporting of observations to the project investigators within a comprehensive interface.

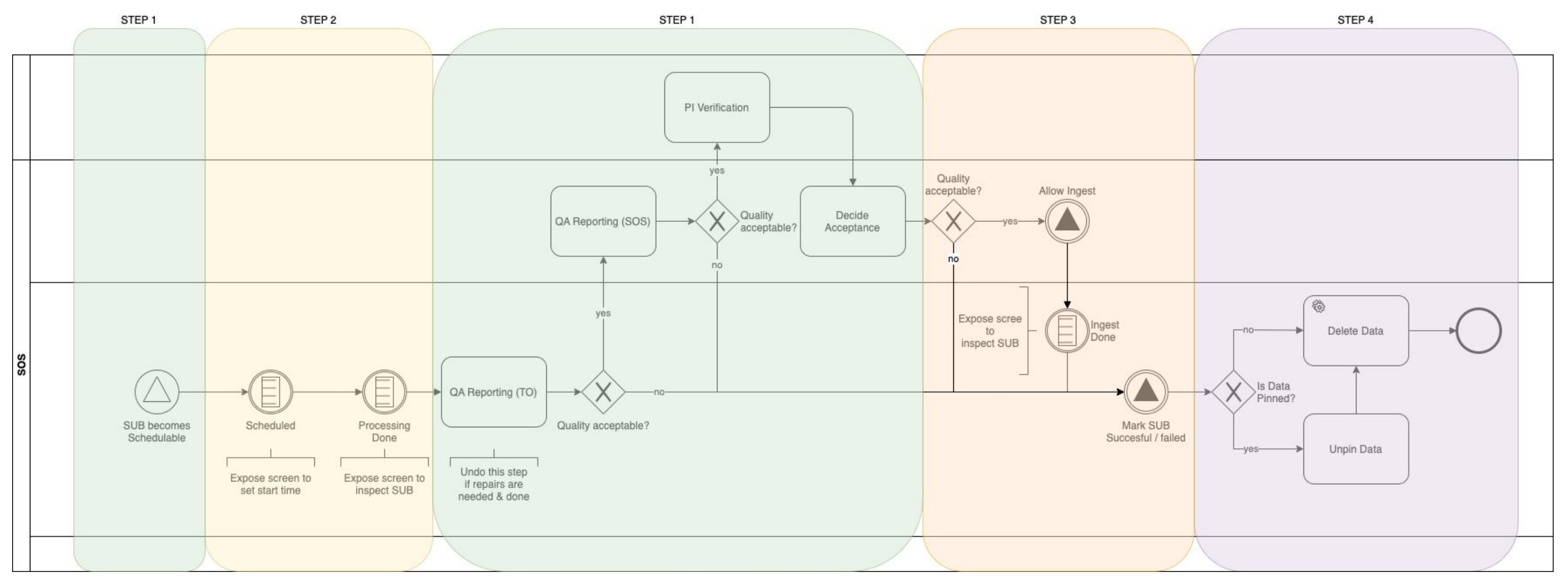


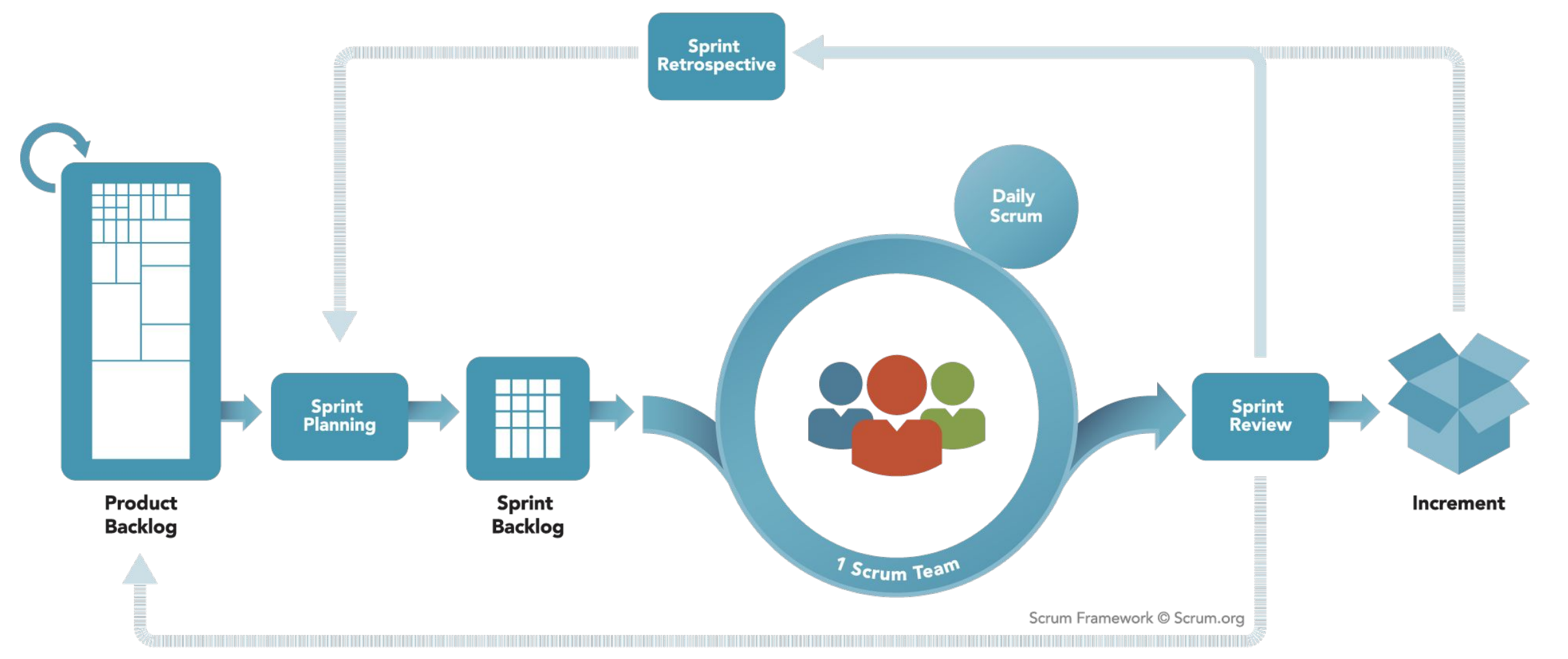
Fig. 3: QA Workflow Steps

REPORTING

TMSS includes a flexible engine that is able to generate reports tailored to the users' needs. Reports can be either directly generated within TMSS Web interface or by using third-party tools (e.g., Jupyter Notebooks) upon accessing the data through the exposed APIs.

DEVELOPMENT PROCESS

The TMSS project embraces the **Scrum AGILE methodology** in order to take advantage of benefits such as quickly adapting changes, including feedback from customers and stakeholders, and delivery throughout the development process. The project is divided in rapid blocks of work, namely *Sprints*, with a duration of 3 weeks and at the end of which an improvement of the software is gained. Issue tracking and AGILE project management are conducted with the employment of the Jira and Confluence softwares.



CI/CD & QA

The development process is supported by using **a Gitlab pipeline (Fig. 4) that ensures CI/CD practices and QA** thanks to a comprehensive test coverage, which is currently composed of unit tests, integration tests and regression tests.

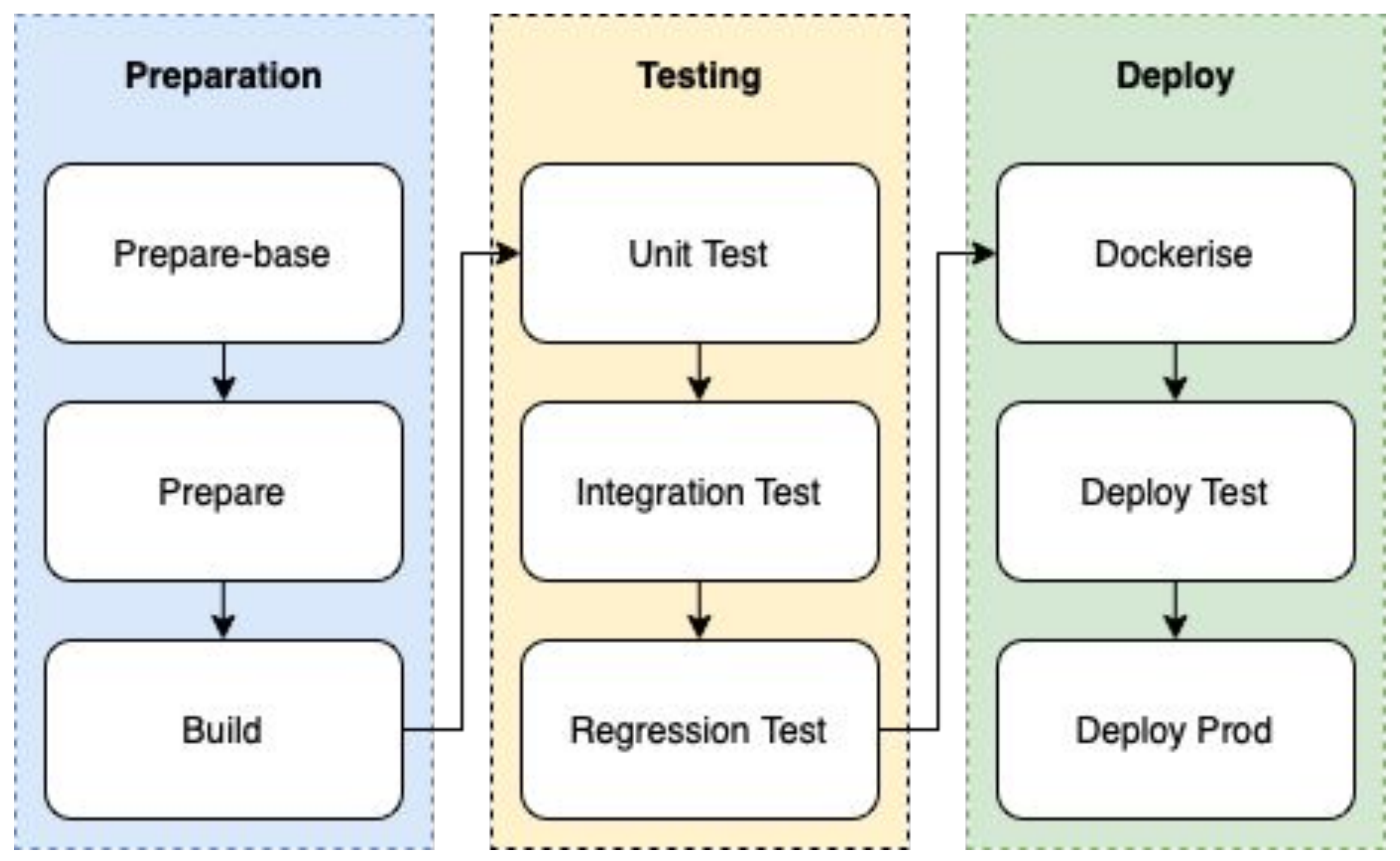


Fig. 4: Gitlab Pipeline for CI/CD & QA

Conclusions

TMSS development started in January 2020, it successfully performed its first production observation in April 2022, and further work is planned to support additional observing strategies and campaigns until the end of October 2022.

REFERENCES

1. https://tmss.lofar.eu/
2. https://git.astron.nl/ro/lofar

This work has been supported by the International LOFAR Telescope (ILT) foundation and the Italian National Institute for Astrophysics (INAF) and led by the Netherlands Institute for Radio Astronomy (ASTRON).