SSC24-P1-07
38th Annual Small Satellite Conference
August 3-8, 2024 | LOGAN, UT, USA

AUTOMATION CONCEPTS AND TECHNOLOGIES FOR FLIGHT DYNAMICS SYSTEMS

M. Rasotto, Flight Dynamics Engineer, Terma B.V., mras@terma.com
J. Cruz, Software Engineer, Terma B.V., jotc@terma.com

I. Introduction

Among the wide set of ground operations, Flight Dynamics (FD) operations can be automated very effectively and allow:

01 Higher efficiency, in case of multiple missions or satellite constellations

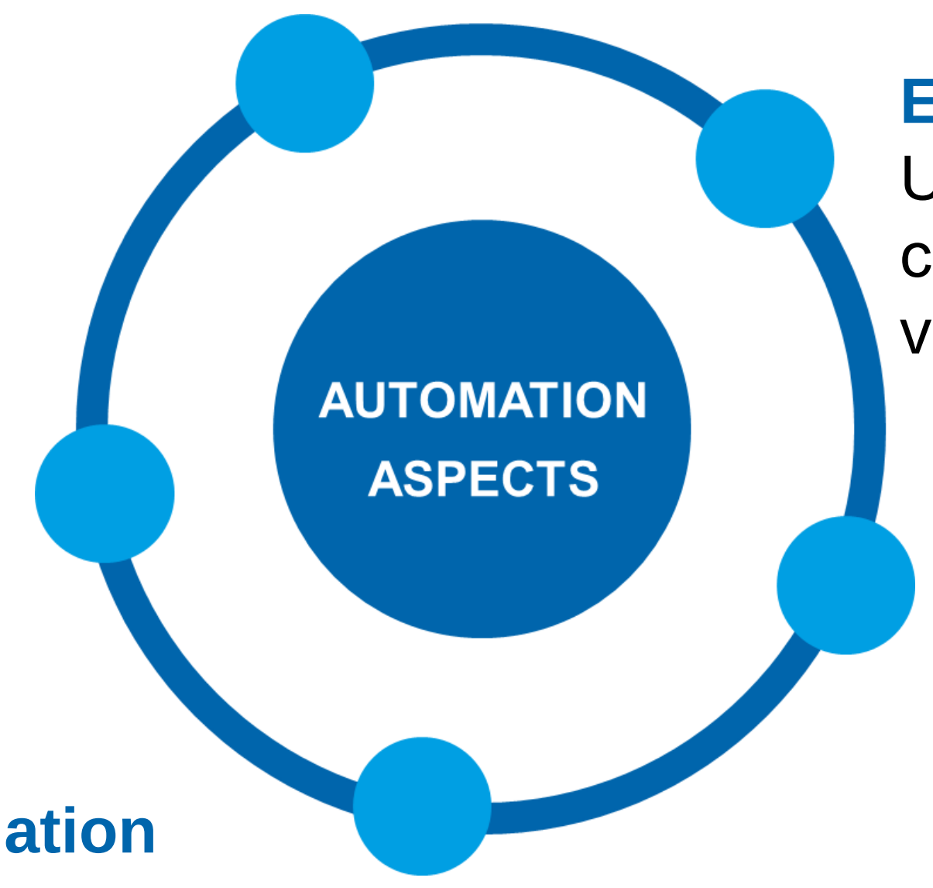
02 Execution of tasks at inconvenient times, e.g., during weekends and public holidays

03 Shorter reaction time in case of time critical or contingency operations (e.g., collision avoidance maneuver preparation)

04 Avoidance of human errors

II. Automation aspects

The main automation aspects applicable to on-ground FD operations can be summarized in the following categories:



Data handling and processing
Ingestion and analysis of input products as soon as they become available (e.g., raw TM, observation files, conjunction warnings, etc.)

Execution of astrodynamics algorithms
Use of the processed data as input for the computation of FD products (e.g., stations' visibilities, maneuver plan, etc.)

Anomalies and errors detection
Recognition of invalid inputs/outputs and/or errors during FD task execution with consequent trigger of appropriate actions

FD products validation
Validity check and assessment of the generated products

Internal/External interfaces
Reception and transmission of I/O with other ground segments components (e.g., Mission Control System, Mission Planning System), and/or external providers

III. Technologies and solution

Different software tools and technologies are required to address the above-mentioned automation aspects, including:

- **Monitoring services,** such as File Watcher in Java or watchdog module in Python
- **Configurable scheduling tools,** e.g., scheduling support in Spring or schedule/crontab modules in Python
- **Microservices architecture and APIs support,** such as REST, GraphQL, etc.
- **Validation criteria,** such as checksums, algorithm convergence and Out-Of-Limit checks
- **Messaging systems,** such as Apache Kafka, NATS or RabbitMQ

The proposed solution foresees the integration of these technologies in so-called **pipelines**, implemented as a set of classes that allow:

1) Configuring a set of **triggers**, to initiate FD tasks execution based on events (e.g., new TM data received) or predefined schedule
2) Defining a set of **agents**, to invoke available APIs for FD activities execution
3) Configuring a set of **controllers** to be invoked for validation and check of the generated products
4) Instantiating a set of **messengers**, used to send notifications regarding the results of the executed FD activities

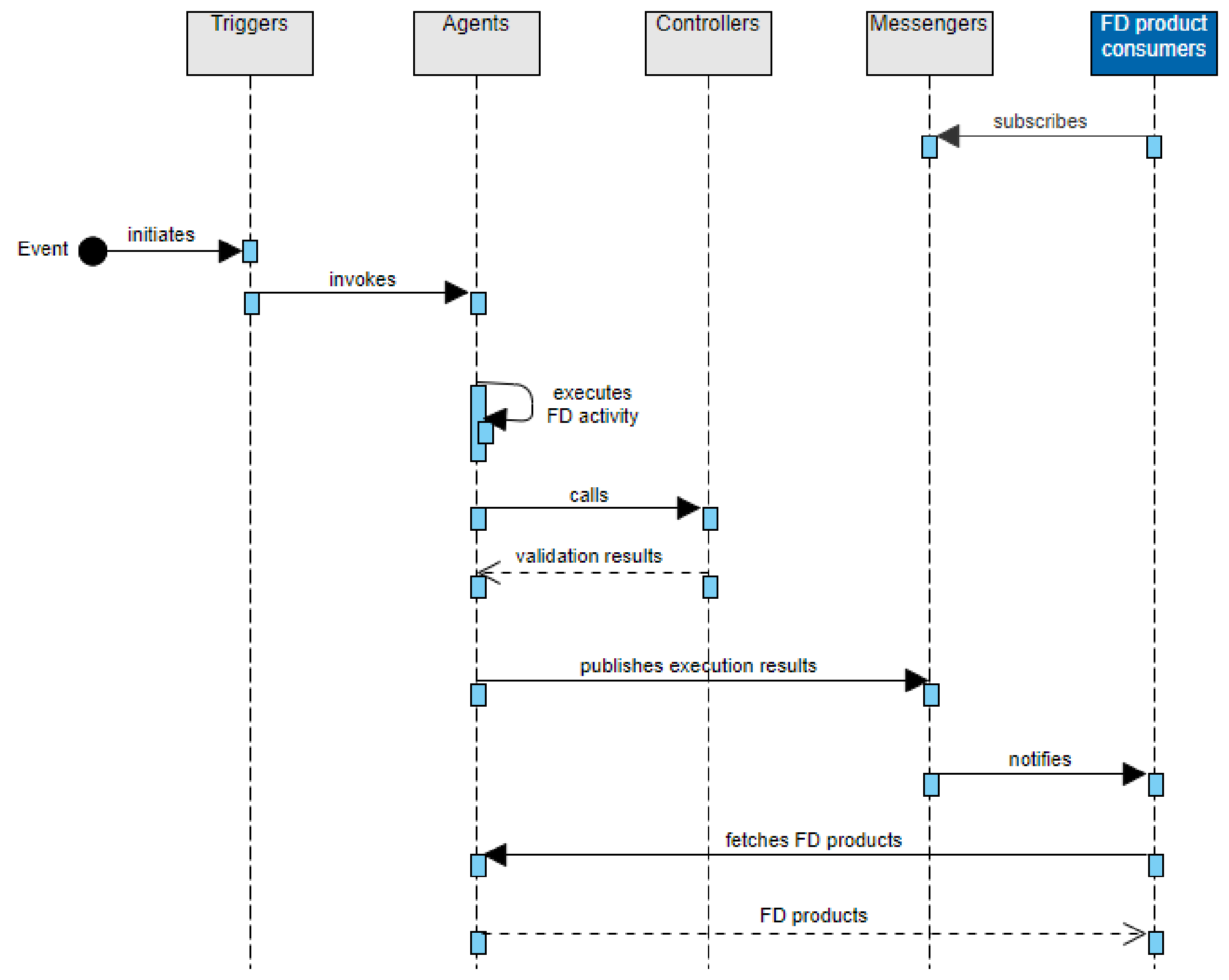
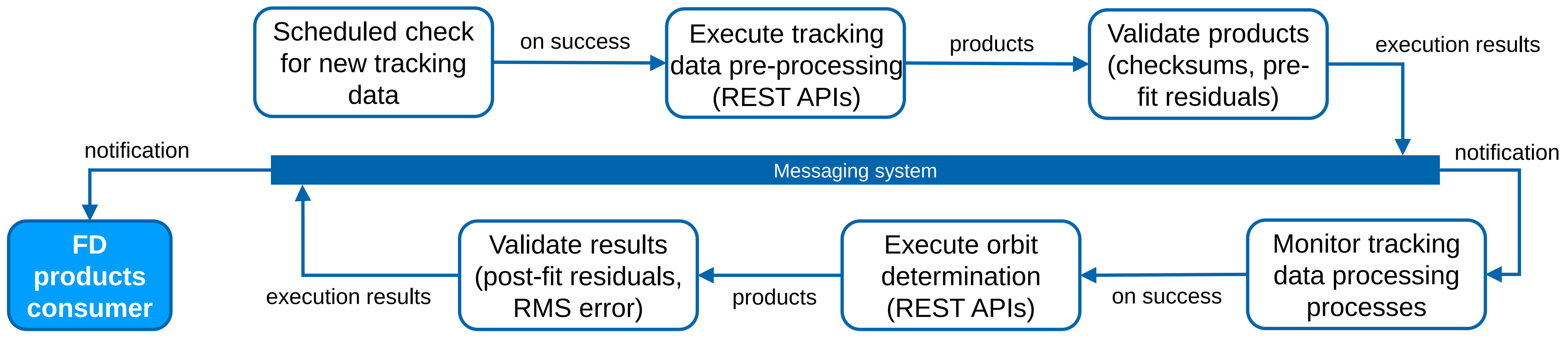


Figure 1 – Pipelines sequence diagram.

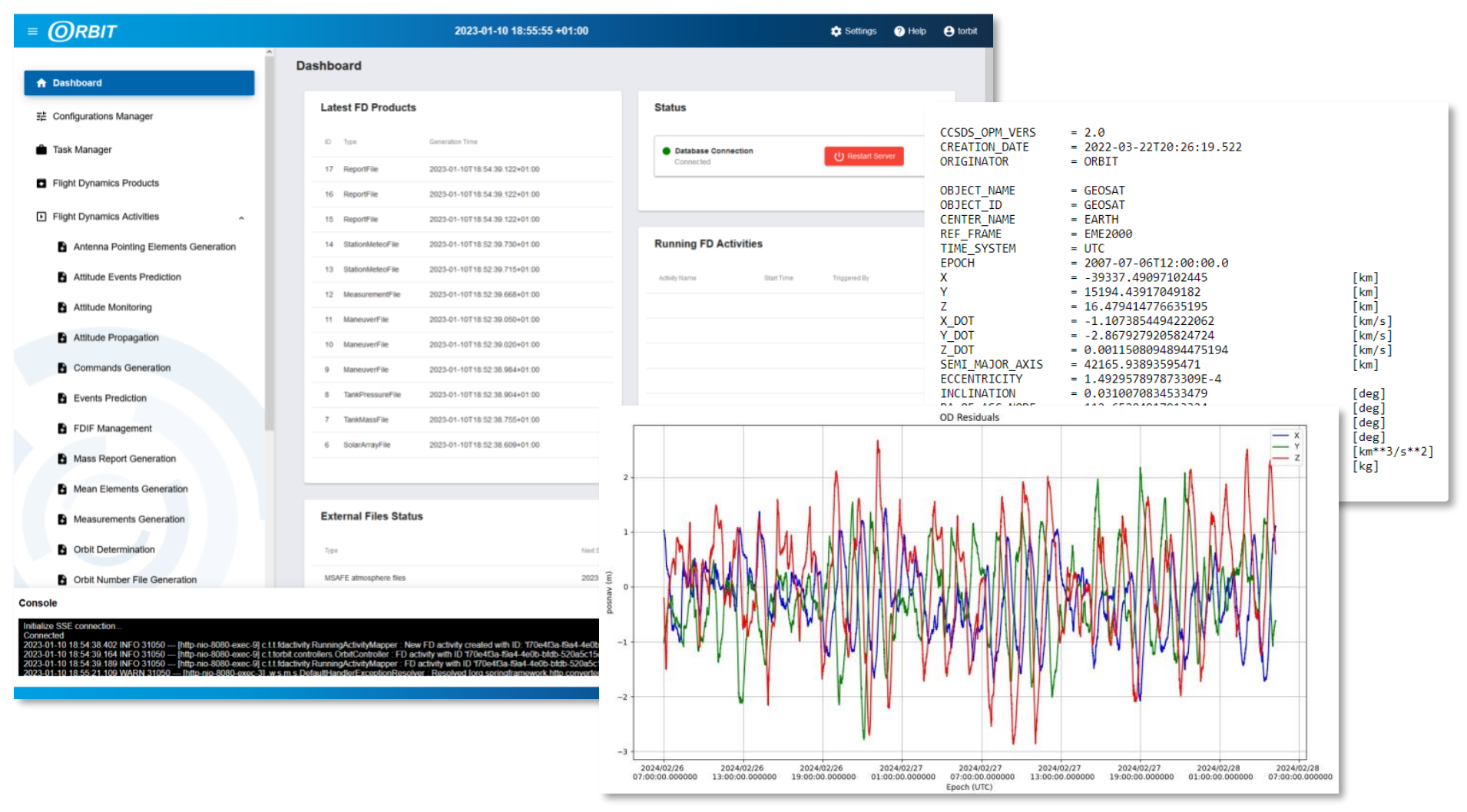
IV. Example: Tracking Data Pre-processing and Orbit Determination


Scheduled check for new tracking data
on success
Execute tracking data pre-processing (REST APIs)
products
Validate products (checksums, pre-fit residuals)
execution results
notification
Messaging system
notification
FD products consumer
execution results
Validate results (post-fit residuals, RMS error)
products
Execute orbit determination (REST APIs)
on success
Monitor tracking data processing processes

By combining different triggers, agents, controllers and messengers, a wide variety of pipelines can be obtained, covering different end-to-end use cases.

V. Conclusions

These concepts and solutions have successfully been implemented by Terma in the new **Flight Dynamics System, ORBIT**, part of the **Terma Ground Segment Suite (TGSS)**.



Find out more…