Project: **Software Technologies Supporting CREAM, GT17-413SD**

ESA Contract No.: 4000134606/21/D/SR

Title: # Executive Summary

Date: 04/07/2023

	Name	Date	Signature
Prepared by:	Project Team		
Project Management:	Tilman Andriof	Tilman Andriof - SIG	Digital unterschrieben von Tilman Andriof - SIG Datum: 2023.07.04 11:04:36 +02'00'

This work has been created by Airbus Defence and Space GmbH on the order of the European Space Agency.
© Copyright European Space Agency, 2023

Executive Summary

Abbreviations

Abbreviation	Explanation
AITF	Automated Integration Test Framework
API	Application Programming Interface
CAM	Collision Avoidance Manoeuvre
CCSDS	Consultative Committee for Space Data Systems
CDM	Conjunction Data Message
CI/CD	Continuous Integration / Continuous Development
CREAM	Collision Risk Estimation and Automated. Mitigation
ESA	European Space Agency
GSTP	General Support Technology Programme
GUI	Graphical User Interface
HIE	High Interest Event
ODM	Orbit Data Messages
SSA	Space Situational Awareness
STCREAM	Software Technologies supporting CREAM
TCA	Time of closest approach
TDM	Tracking Data Message
TRACE	TRansparent Collision Avoidance CentrE
UML	Unified Modeling Language

This work has been created by Airbus Defence and Space GmbH on the order of the European Space Agency.
© Copyright European Space Agency, 2023

Executive Summary

1 Introduction

This executive summary of the GSTP activity "Software Technologies Supporting CREAM" describes the development of the *TRACE* (TRansparent Collision Avoidance CentrE) prototype, which aims to improve collision avoidance and coordination in spacecraft operations.

Collision avoidance becomes more and more crucial in space operations due to the limited orbital capacity and the increasing number of satellites. Close approaches and potential collisions are becoming more frequent, requiring efficient and standardized data exchange mechanisms among multiple actors. The CREAM cornerstone within ESA's Space Safety Programme focuses on enabling safe operation of space assets and reducing manpower efforts for collision avoidance. The *TRACE* prototype is developed as the central element of a potential CREAM system to facilitate streamlined interactions and data exchange among operators and service providers.

The scope of work for "Software Technologies Supporting CREAM" included enabling efficient stakeholder coordination through a secure communications architecture that considers data integrity, tamper resistance, trust, access control, confidentiality, and transparency. The architecture allows for different coordination protocols and supports data exchange among spacecraft operators and data providers. The project was executed by Airbus Defence and Space GmbH as prime, together with Deutsches Forschungszentrum für. Künstliche Intelligenz GmbH (DFKI) for supporting the Architecture Design and CGI Deutschland B.V. & Co. KG for Software Implementation, Verification and Validation. The kick-off was in June 2021 and final presentation in December 2022, followed by a 6-month warranty period. The work was conducted as agile software development process to provide the opportunity for iterative design and development activities by ensuring overlap between study and design, as well as design and implementation phases.

2 Stakeholder survey and literature review

The first task of the project covered the literature review, stakeholder survey, requirements baseline creation, initial architecture design activities, and the reference scenario creation for later validation.

The literature review explored topics related to space situational awareness (SSA) and software technologies that lead to the choices for prototype implementation (see chapters 3 and 4), including cryptography, authorization, and interfaces. Three design pillars, essential to acceptance of the final product, were identified (cf. Figure 2-1 with the related actual topics/properties of the software): Trust, Coordination, and Future-proof architecture. These pillars and related topics were used as guidance throughout the overall study. With respect to the *TRACE* prototype, most elements of the pillars deserved special attention and have been already considered for implementation, while others (operational aspects such as anomaly handling, system robustness, and actual validation interfaces to ensure data credibility) were agreed to be out of scope for a first prototype.

This work has been created by Airbus Defence and Space GmbH on the order of the European Space Agency.
© Copyright European Space Agency, 2023

Executive Summary

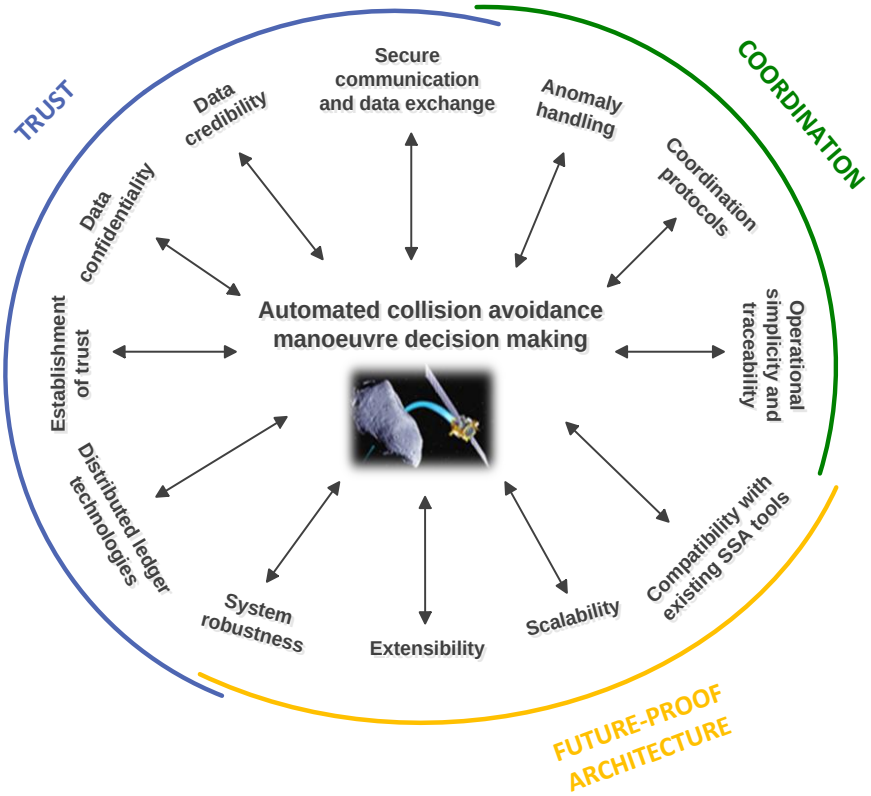

Figure 2-1: Design driver pillars for TRACE.

Several aspects of manoeuvre negotiation have been investigated. Interfaces to future sophisticated algorithms have been considered but it was agreed that the prototype development should only include a simple starting point. The key issue of conjunctions between multiple manoeuvrable spacecraft of different operations teams was found to be reaching a mutual decision on the way forward. It shall be agreed on whether (and who) to manoeuvre or not to manoeuvre, based on the individual risk assessments of the involved operators. Airbus has identified a “Double offer” strategy as best starting point for the prototype, which could be extended in future also with pre-defined rules. The overall idea is that operators state their preference for either executing a collision avoidance manoeuvre or for not executing one. Preferences are placed as “offers” with unequal strengths for both options. The scheme promotes manoeuvres, as it can be assumed that manoeuvring is more secure than not to manoeuvre (provided that a certain minimum level of data quality is established).

The stakeholder survey gathered information from satellite operators and conjunction analysis providers to understand their views and requirements for a conjunction coordination platform. In total, 56 companies, organisations and space agencies have been contacted. The response rates were 23% and 37% for operators and service providers, respectively. The survey indicated a very high interest in such a system, with nearly all responding stakeholders willing to participate. One aspect of the survey was to determine better estimates for expected communications traffic and storage capacity needs. It was derived that several hundreds of TB per year could be needed in future to store all CDMs and OEMs for

This work has been created by Airbus Defence and Space GmbH on the order of the European Space Agency.
© Copyright European Space Agency, 2023

Executive Summary

complete traceability. Regarding the user needs, it was found that both options of API- and web-access should support all core functionalities. CCSDS was confirmed as favourable baseline for the interface, including all CCSDS formats recreations (i.e. XML, KVN, and JSON). JSON will have a specific importance in the context of automation. Last but not least, a diverse understanding of the definition of a High Interest Event was shown. This will lead to different opinions in negotiations which need to be considered in the future system.

3 Architecture Design

The project involved developing several proof-of-concepts (PoCs) and designing the architecture for the backend system. The architecture includes components such as the Primary Log, Management Function, and a Negotiation Module. Several trade-offs have been conducted in technology selection, including the options for a centralized versus decentralized approach for the Audit Log.

Figure 3-1 shows a reference architecture for the CREAM/TRACE Backend.

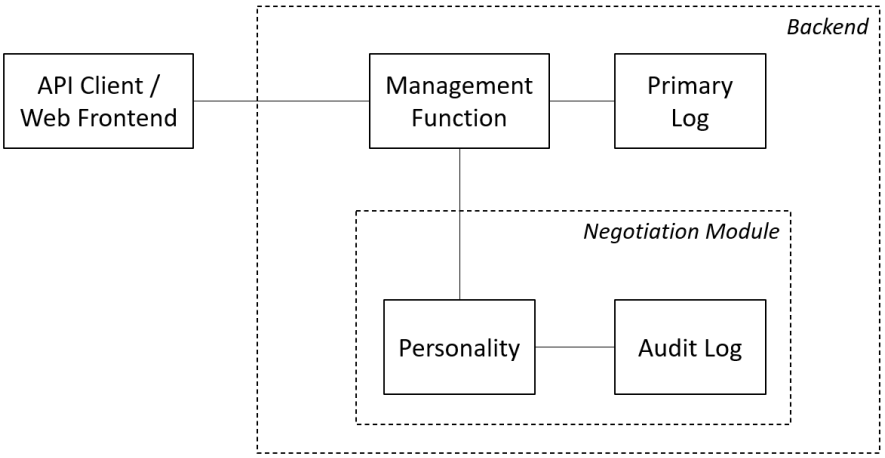

Figure 3-1: Overall Architecture Components of TRACE.

The Primary Log serves as the primary storage for all data uploaded by users. It uses a relational database to handle the high volume of orbit data. The Audit Trail data from the Primary Log is sent to the Negotiation Module, where it is assigned an HIE ID and stored securely. The Management Function acts as an intermediary between clients, the Primary Log, and the Negotiation Module, coordinating their interactions and handling data access restrictions.

The Negotiation Module provides services for audit trail abstraction, derivation of CAM (Collision Avoidance Manoeuvre) assignments, and a pluggable negotiation algorithm in terms of exchangeable algorithms for the actual negotiation of CAM responsibility. It defines the HIE (High Interest Event) procedure as a state machine. The module communicates with the Management Function and handles the data structures stored in the Audit Log. The architecture requires a message-based data storage due to the nature of the *TRACE* application and the immutable storage mechanism.

This work has been created by Airbus Defence and Space GmbH on the order of the European Space Agency.
© Copyright European Space Agency, 2023

Executive Summary

In the centralized approach, a single database is used for the Audit Log, ensuring data integrity and verifiability. The decentralized approach involves setting up a blockchain network with multiple nodes, distributing the responsibility for data handling and ensuring resilience.

The technology trade-offs are evaluated, considering various centralized[1] ("C") and decentralized[2] ("D") options such as *Trillian* (C), *ImmuDB* (C), *Hyperledger Sawtooth* (D), and *Parity Substrate* (D). The centralized architecture with *ImmuDB* is rated as the best option due to its maturity and support for a verifiable Audit Log. Although distributed ledgers benefit in resilience and auditability, they have disadvantages in operational simplicity, performance, and scalability. Additionally, a directly distributed execution of HIE logic on Blockchain nodes would violate certain data sharing and encryption requirements. As alternative Immutable Database *Trillian* was considered as well, but *ImmuDB* is seen as more mature technology with a rich tool set. However, it should be noted that due to the described level of abstraction the architecture supports all investigated Audit log technologies and the current *ImmuDB* implementation serves as a starting point.

4 Implementation

The *TRACE* software was developed as a prototype for cloud-native applications. The project was classified as TT-4, Software Development in Studies and Prototypes, and followed an agile and iterative development approach.

The backend services were implemented using the Go programming language, while the web frontend was developed using Typescript and the React framework. Automatic code generation was used for generating Application Programming Interface (API) stubs, API documentation, and mock objects for unit tests.

The *TRACE* prototype provides the following services:

- *CCSDS Data Sharing* – allows users to share CCSDS navigation data (e.g. CDM, ODM, and TDM). The CREAM platform provides strict data sharing policies for restricted data sharing. All data items uploaded to the system are cryptographically signed by the originator to allow other users to confirm origin and integrity of data items.
- *High-Interest Events Detection* – flags HIEs, close approaches with a high collision risk, to the operator/owner of the involved space objects.
- *Guided CAM Negotiation* – allows operators involved in a HIE to negotiate the responsibility to perform a collision avoidance manoeuvre using predefined coordination protocols.

[1] Both fully centralized versions and extensions as "transparent ledger" have been investigated.

[2] Different implementations with and without Smart Contracts have been investigated.

This work has been created by Airbus Defence and Space GmbH on the order of the European Space Agency.
© Copyright European Space Agency, 2023

Executive Summary

- *Tamper-proof audit trail* – for all negotiations the platform stores encrypted and signed messages, supporting data and the resulting manoeuvre agreements in a tamper-proof audit trail, using blockchain technology, to provide full traceability and non-repudiation to the involved parties.
- *Service Requests* – a marketplace for service or data requests and offers.

The system includes user roles such as operators/owners of space objects, conjunction analysis providers, and administrators. Particular attention was paid to a dedicated database schema and key handling mechanisms to ensure the security and encryption of private keys. The *TRACE* Prototype has been developed as a cloud-native application running within a *Kubernetes* cluster and using a *GitLab* CI/CD pipeline for continuous integration and deployment.

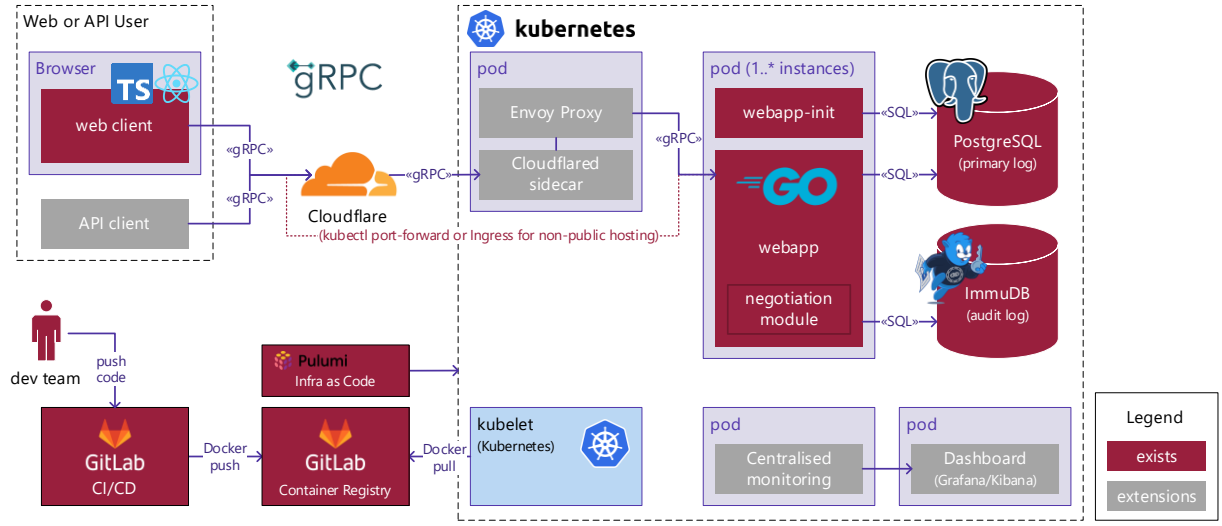

Figure 4-1: Overall architecture of the TRACE Prototype.

Figure 4-1 displays the overall design of *TRACE*, containing the following components:

- **web client**: The web client is a TypeScript/React web application running within the web user's browser, which interfaces with the webapp server using the gRPC-Web variant of the gRPC protocol. For hosting with a public internet domain a *Cloudflare* proxy is foreseen by the design. For internal hosting without public domain this is not an option. For this scenario, a *Kubernetes* Ingress controller will be configured, or simple port-forwarding can be used for development.
- **webapp**: The webapp server is the central backend process which implements the application logic and provides the gRPC API for the web client and external API clients. Multiple instances of the webapp server are deployed as a load-balanced, scalable *Kubernetes* deployment.
- **webapp-init**: The webapp-init container executes the database schema migrations for the primary log as part of the webapp POD start-up. The init container updates the SQL database schema to match with the version expected by the webapp server.

This work has been created by Airbus Defence and Space GmbH on the order of the European Space Agency.
© Copyright European Space Agency, 2023

Executive Summary

- **negotiation module**: The negotiation module implements the HIE procedure logic for guided CAM negotiations with support for pluggable coordination protocols. The *TRACE* Prototype implements the Double Offer coordination protocol.
- **primary log**: The primary log is an SQL database providing persistent state for all application functionalities such CCSDS data sharing, user and session management, HIE processing, service requests, etc. An in-cluster deployment of *PostgreSQL* is provided as shown in the diagram, but an external *PostgreSQL* database can be used instead by re-configuring the database URL in the deployment configuration.
- **audit log**: The audit log provides a tamper-proof, immutable storage for the audit trail of HIEs. As for *PostgreSQL*, and in-cluster deployment of *ImmuDB* is provided or an external *ImmuDB* installation can be used by simple re-configuration.
- **GitLab CI/CD**: A *GitLab* CI/CD pipeline is used for fully automated continuous integration and deployment of the *TRACE* Prototype, based on the *Pulumi* Infrastructure as Code tool.
- **Container Registry**: A Container Registry is needed to store the Docker container images so that they can be deployed by the *Kubernetes* runtime. By default, the *GitLab* Container Registry is used for this purpose. Another container registry accessible by the *Kubernetes* cluster can be configured instead.

An external auditor service is not part of the *TRACE* Prototype implementation which uses the centralised approach for the audit trail instead of a decentralised, distributed ledger solution.

The web user interface of the *TRACE* system includes a dashboard and various elements for accessing different functionalities. Figure 4-2 shows the dashboard (overview of ongoing high-interest events and negotiations, open tasks as well as system news and information) and navigation bar.

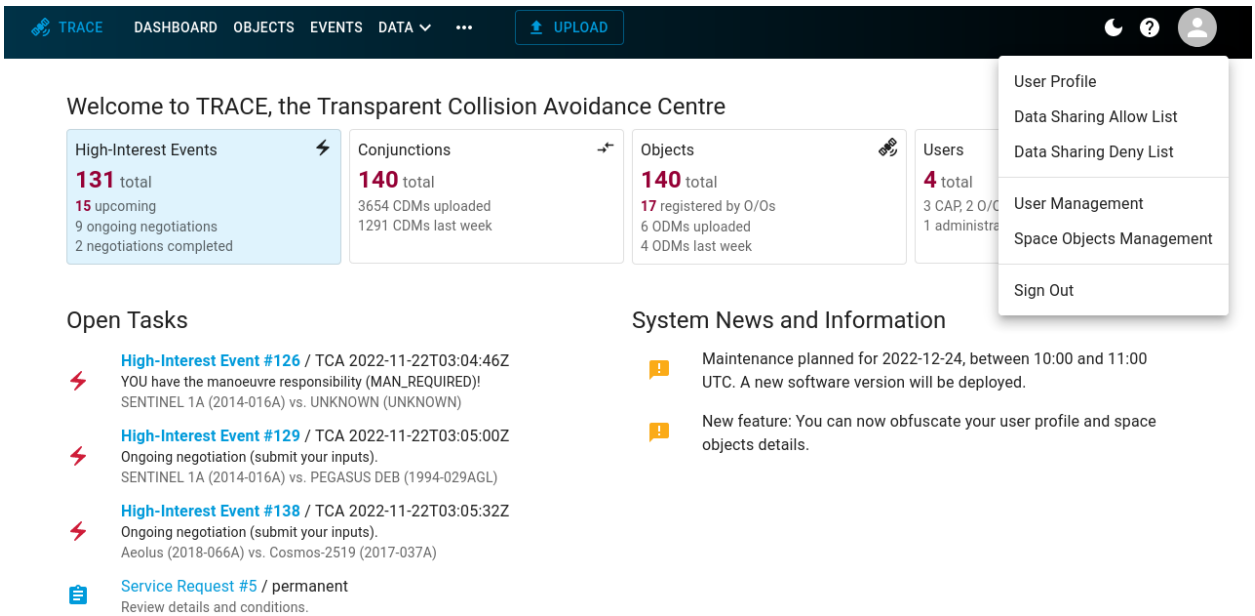

Figure 4-2: GUI Overview and navigation bar.

This work has been created by Airbus Defence and Space GmbH on the order of the European Space Agency.
© Copyright European Space Agency, 2023

Executive Summary

The state machine of the implemented HIE processing logic is shown in Figure 4-3 as Unified Modelling Language (UML) diagram. The transitions are annotated with the gRPC API messages which trigger the transition.

Uploads of new CDM files associated to a conjunction event do not affect the current state of the event apart from transitioning an event from MAN_UNINITIALISED to MAN_NEGOTIATE state when a conjunction event is first flagged as HIE. However, ongoing timers have to be re-scheduled when a shift in the TCA happens due to the newly uploaded CDM.

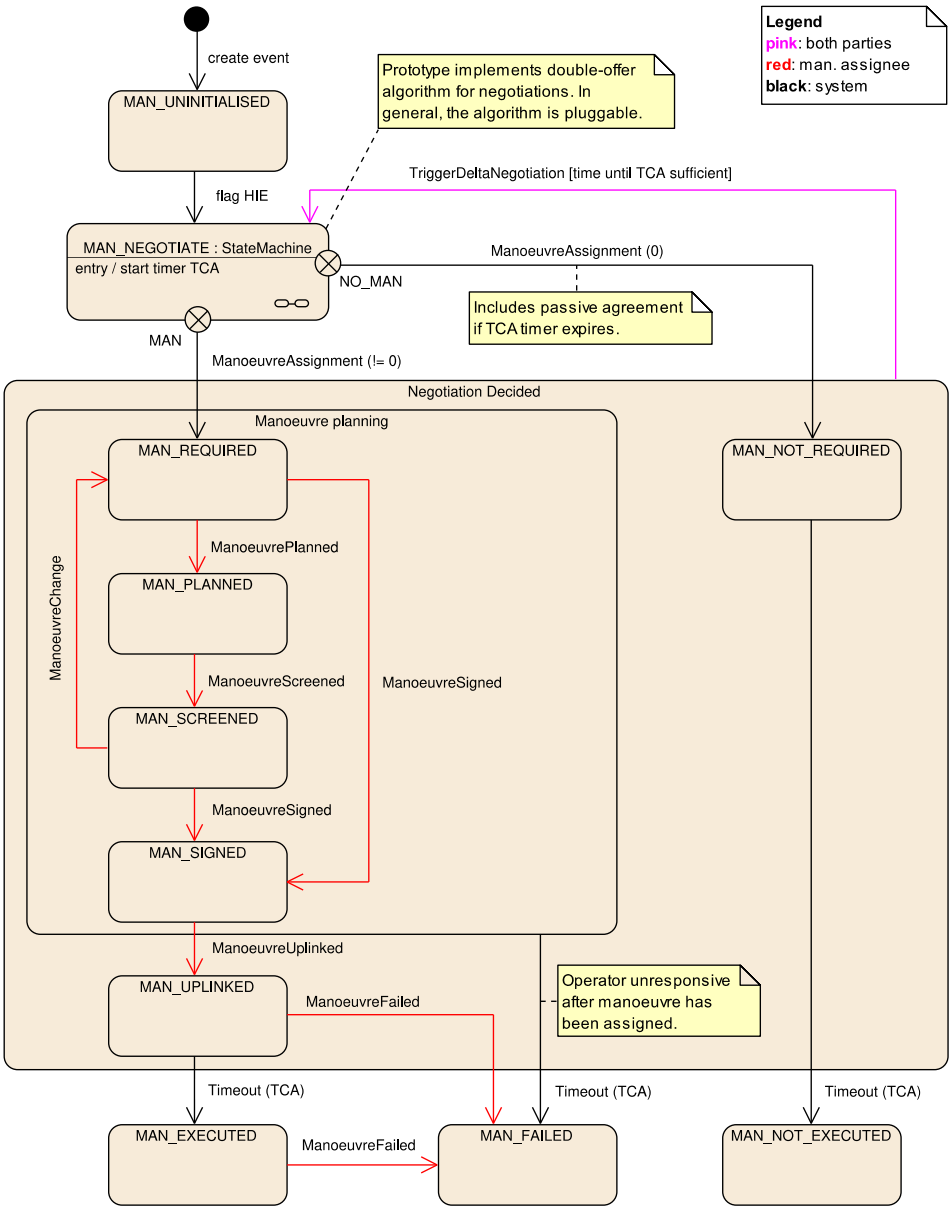

Figure 4-3: State Machine diagram of the HIE processing logic

This work has been created by Airbus Defence and Space GmbH on the order of the European Space Agency.
© Copyright European Space Agency, 2023

Executive Summary

The current coordination protocol implementation (“double-offer” algorithm) called in the MAN_NEGOTIATE state is shown in Figure 4-4, as UML state machine annotated with the gRPC messages that trigger the transition.

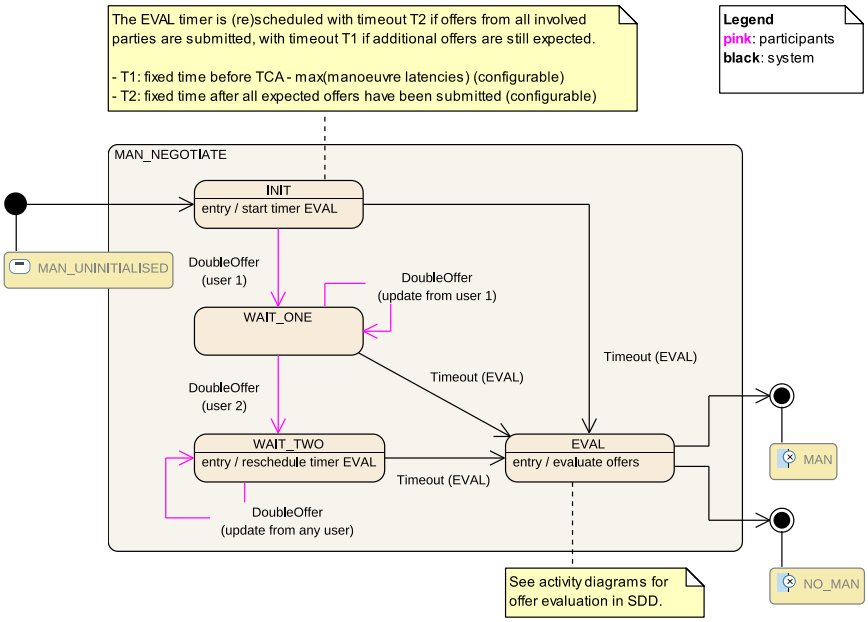

Figure 4-4: State Machine diagram of Double-Offer evaluation

5 Testing

The software validation and reference scenarios were performed using CGI’s Automated Integration Test Framework (AITF), which is a Ruby-based framework for automated testing. The AITF was used to conduct end-to-end verification tests of the *TRACE* system and validate different features and functionalities. This includes reference scenarios, based on historical conjunction data and CAM examples. The scenarios aimed to test the behaviour of *TRACE* in various communication flows, including nominal and non-nominal situations and sudden changes in decisions based on new data.

The software performance assessment measured the processing performance of the *TRACE* prototype for handling CDMs and OEMs. The performance benchmark showed the system's capability to process a certain number of file uploads and suggests that the targeted performance requirements from the requirement baseline for an operational system (e.g. processing of 100 million file uploads per day, including 30 million CDMs and 80.000 OEMs) are likely to be achievable for a production system. However, further optimizations and a more representative benchmark would be required to accurately assess scalability and estimate hardware requirements.

This work has been created by Airbus Defence and Space GmbH on the order of the European Space Agency.
© Copyright European Space Agency, 2023

Executive Summary

6 Conclusion and Outlook

The *TRACE* prototype was developed as part of the GSTP project “Software Technologies supporting CREAM”.

A requirements baseline was specified as first step and was the foundation for the development activities within the subsequent project phases. Trust, Coordination, and Future-proof architecture have been identified as design pillars, essential to acceptance of the final product and considered for the implementation of *TRACE*. A “Double offer” strategy was chosen as best starting point as manoeuvre negotiation algorithm, which could be extended in future also with pre-defined rules. A stakeholder survey gathered information from satellite operators and conjunction analysis providers to understand their views and requirements for a conjunction coordination platform. The survey indicated a very high interest in such a system. It was found that both options of API- and web-access should support all core functionalities. In addition, several hundreds of TB per year could be needed in future to store all CDMs and OEMs, which motivated to separate a main CCSDS data base from the actual audit trail.

A reference architecture was defined already early in the project. It is based on a centralized database for CCSDS data and an audit trail for HIE-related negotiations. The latter part can be described as an immutable database with a negotiation module. This negotiation module was designed with “pluggable” algorithms and defined common interfaces. The architecture design considered the development of different Proof of Concepts and technology trade-offs., especially for the audit trail. The audit trail traces the negotiation procedure between satellite operators for all HIEs in verifiable and immutable storage, to enable automated and semi-automated manoeuvre negotiation, and to ensure data sovereignty as well as security, integrity, non-repudiation. The architecture supports multiple Audit log technologies - both distributed ledgers (i.e. Blockchains) and Immutable Databases have been considered. Eventually, it was decided to use a central entity of *ImmuDB* for the *TRACE* prototype.

TRACE provides, first, the service of CCSDS Data Sharing with strict data sharing policies and cryptographic signing; secondly, a High-Interest Events Detection to flag HIEs to the operator/owner of the involved space objects; thirdly, a guided Negotiation of the responsibility to perform a CAM (Collision Avoidance Manoeuvre) using predefined coordination protocols; and last but not least, a tamper-proof audit trail for HIEs to provide full traceability and non-repudiation to the involved parties. In addition, a basic implementation of Service Requests is included for service or data requests and offers. The performance benchmark suggests that the targeted performance requirements from the requirement baseline for an operational system are likely to be achievable for a production system. A major part of the performance gain needed for the initial operational system may already be achievable by use of appropriate server hardware and only small amounts of optimisations.

For an operational software, however, there are also additional functionalities required that have been de-scoped for the prototype. It is recommended that this advancement will be pushed forward as part of bringing the system to actual users and should include the aspects of data validation and complete anomaly handling (e.g. defined processes for “non-nominal” actions of users). Data Validation includes, on the one hand, specific validation tasks in terms of credibility/plausibility by other users of data being shared within the system and, one the other hand, means for audit log verification by users via the

This work has been created by Airbus Defence and Space GmbH on the order of the European Space Agency.
© Copyright European Space Agency, 2023

Executive Summary

cryptographic proofs principally supported by *ImmuDB*. In addition, the overall evolution of all points that were only seen as a starting point from the beginning need to be continued: The exchangeable algorithms for manoeuvre negotiation should be finalized (e.g. combined with pre-defined rules), a kind of market place for third party services could be established (e.g. for the provision of trajectory screening or tracking data), and, in the end, also "intelligent" functionalities could be included directly in *TRACE* in the context of data fusion, specific risk assessment and manoeuvre optimisation. But independently of which services will remain outside of *TRACE*, also the overall inter-operability with alternative coordination systems should be kept in mind.

This work has been created by Airbus Defence and Space GmbH on the order of the European Space Agency.
© Copyright European Space Agency, 2023