https://doi.org/10.1109/DASC58513.2023.10311187

Cybersecurity Engineering: Bridging the Security Gaps in Avionics Architectures and DO-326A/ED-202A

Fahad Siddiqui*, Alexander Ahlbrecht†, Rafiullah Khan*, Sena Yengec Tasdemir*, Henry Hui*
Balmukund Sonigara*, Sakir Sezer*, Kieran McLaughlin*, Wanja Zaeske†, Umut Durak†
*Queen's University Belfast, United Kingdom
†German Aerospace Center (DLR), Institute of Flight Systems, Germany
*f.siddiqui, rafiullah.khan, s.yengectasdemir, h.hui, b.sonigara, s.sezer, kieran.mclaughlin@qub.ac.uk,
†alexander.ahlbrecht, wanja.zaeske, umut.durak@dlr.de

***Abstract*—*Urban Air Mobility* is envisioned as an on-demand, highly automated and autonomous air transportation modality. It requires the use of advanced sensing and data communication technologies to gather, process, and share flight-critical data. Where this sharing of mix-critical data brings opportunities, if compromised, presents serious cybersecurity threats and safety risks due to the cyber-physical nature of the airborne vehicles. Therefore the avionics system design approach of adhering to functional safety standards (DO-178C) alone is inadequate to protect the mission-critical avionics functions from cyber-attacks. To approach this challenge, the DO-326A/ED-202A standard provides a baseline to effectively manage cybersecurity risks and to ensure the airworthiness of airborne systems. In this regard, this paper pursues a holistic cybersecurity engineering and bridges the security gap by mapping the DO-326A/ED-202A system security risk assessment activities to the Threat Analysis and Risk Assessment process. It introduces *Resilient Avionics Architecture* as an experimental use case for *Urban Air Mobility* by apprehending the DO-326A/ED-202A standard guidelines. It also presents a comprehensive system security risk assessment of the use case and derives appropriate risk mitigation strategies. The presented work facilitates avionics system designers to identify, assess, protect, and manage the cybersecurity risks across the avionics system life cycle.**

***Index Terms*—Cybersecurity, Avionics, Threat modelling, Threat analysis, Risk assessment, Model-based system design, Cyber resilience, Safety-critical, DO-326A, ED-202A.**

I. Introduction

Urban Air Mobility (UAM) is defined as a safe and efficient way of transporting passengers, delivering goods, and emergency services within or traversing urban areas [1], [2]. It is envisioned as an on-demand, highly automated and potentially autonomous air transportation modality. By 2026, it is estimated that the 5G market targeting the aviation industry will grow to $3.9 billion [3]. Though, where the technological evolution of embedded systems [4] and adoption of advanced autonomous technologies bring benefits (e.g. broader connectivity), it equally opens doors to a wide range of vulnerabilities and attack vectors [5]. These advanced features can enable new opportunities for adversaries to remotely carry out sophisticated cyber-attacks [6]. In 2019, a short period of system errors across some *Automatic Dependent Surveillance Broadcast* (ADS–B) units caused about four hundred flights cancelled [7]. To date, the publicly reported aviation cyber-attacks are not cyber-physical and limited to data breaches. However, this fact may not remain valid in future, as publicly reported cyber attacks against critical infrastructure and the automotive industry have increased acutely [8]. The *Radio Technical Commission for Aeronautics* (RTCA) and the *European Organisation for Civil Aviation Equipment* (EUROCAE) have coordinated with the industry to develop standards such as DO-326A/ED-202A, to tackle the rising cybersecurity threats that may compromise the safety of airborne aviation technologies.

Concerning the recent cybersecurity efforts, the idea in this paper is to introduce an experimental application use case *Resilient Avionics Architecture* (RA2) for UAM that shall establish and maintain cyber resilience [10]. To achieve this, cybersecurity shall be systematically built into the system from the ground up. Therefore a *secure-by-design* method is targeted that improves on approaches where cybersecurity is considered as an afterthought or added into the system on an ad-hoc basis. There is a need for a holistic cybersecurity engineering process that systematically takes the recommendations of cybersecurity standards and maps them onto the existing traditional system design engineering processes [9]. This process shall enable system designers to systematically define, assess, design, implement, verify, and validate the system's safety and security requirements through the system design phases and operational life cycle. The RA2 shall establish resilience [4] by providing *prevention, detection, response and recovery* capabilities to ensure security and functional safety (fail-operational, fail-safe and fault-tolerant) [5], [11], [12] of UAM missions.

This paper extends [9] and complements previously published research works [12], [13], [14], [15]. In this paper, the *Resilient Avionics Architecture for Flight Assistance System* (RA2FAS) has been chosen as an experimental use case. It illustrates one of the possible ways to realise a resilient, secure-by-design avionics architecture. Section II presents the essential background on cybersecurity engineering, threat modelling and risk assessment processes. To establish the right context, Section IV defines the scope and presents

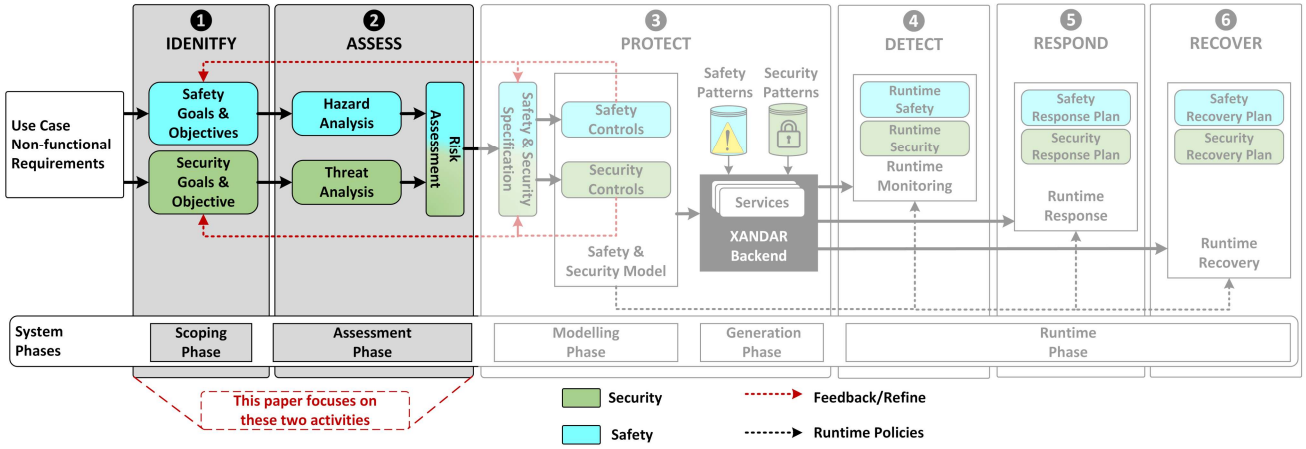


Fig. 1: Block diagram of cybersecurity engineering process detailing safety and security engineering activities required to establish resilience. Their required interactions across various phases of the system design and development life cycle [9].

the functional and software architecture of the experimental RA2FAS use case. The following are the paper's contributions:

- Identification of tasks required to conduct ***System Security Risk Assessment*** defined by DO-326A/ED202A, and map these activities on *Threat Analysis and Risk Assessment* (TARA) process presented in Section III.
- Defined ***Security Scope*** of a RA2FAS in line with DO-326A/ED202A in Section V.
- Conducted a comprehensive ***System Security Risk Assessment*** in line with DO-326A/ED202A of the experimental RA2FAS in Section VI.

II. Background & Related Work

A. Holistic Cybersecurity Engineering Process for Safety-Critical & Cyber-Physical Systems

To systematically approach the complex system design and cybersecurity challenges of safety-critical and cyber-physical systems, Siddiqui *et al.* have proposed a holistic cybersecurity engineering process [9] as shown in Fig. 1. This process is based on six core cybersecurity principles ①***Identify***, ②***Assess***, ③***Protect***, ④***Detect***, ⑤***Respond***, and ⑥***Recover***. These principles have been driven by and are in line with the international cybersecurity guidelines and frameworks [16]. This holistic cybersecurity process allows identification and assessment of both security risks, threats, and safety hazards to the safety-critical system [5], [8], [10]. During scoping and assessment phase of the system engineering, it facilitates both manufacturers and suppliers of safety-critical systems to effectively identify, assess, and manage cybersecurity risks across the entire life cycle of the system [17]. Hence, it helps to improve cybersecurity posture and maintain compliance of safety-critical systems with existing and evolving cybersecurity standards [18]. The presented work focuses on the ①***Identify*** and ②***Assess*** principles as highlighted in Fig. 1. The ④***Detect***, ⑤***Respond***, and ⑥***Recover*** principles will be discussed in future work.

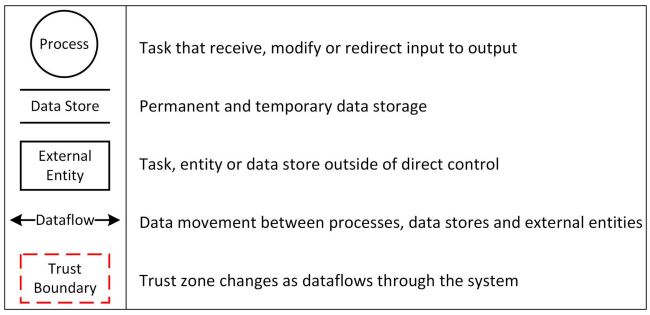


Fig. 2: Dataflow elements required for threat modelling.

B. Threat Modelling and Risk Assessment

Threat modelling and risk Assessment is defined as [19]:

> *"A structured approach of identifying and prioritising potential threats to a system and determining values of the risks posed by these threats and identify potential mitigation to reduce or neutralise those threats"*

Effective threat modelling and risk assessment requires systematic system modelling, threat analysis and risk assessment activities [20]. It is important to understand how the system works and interacts with other hardware/software components [21]. In this regard, the *European Union Aviation Safety Agency* (EASA) has provided guidance of adopting threat modelling and risk assessment methods to conduct trustworthiness analysis for avionics applications [17]. To methodologically model an embedded system, a *Dataflow Diagram* representation is a widely used approach by the research community as it depicts how the information is passed through different components of the system. Fig. 2 introduces the necessary dataflow elements to model system and the interactions among system components. There are various open-source and commercial threat modelling tools available that leverage known vulnerabilities databases to identify the potential threats to system assets such as Microsoft Threat

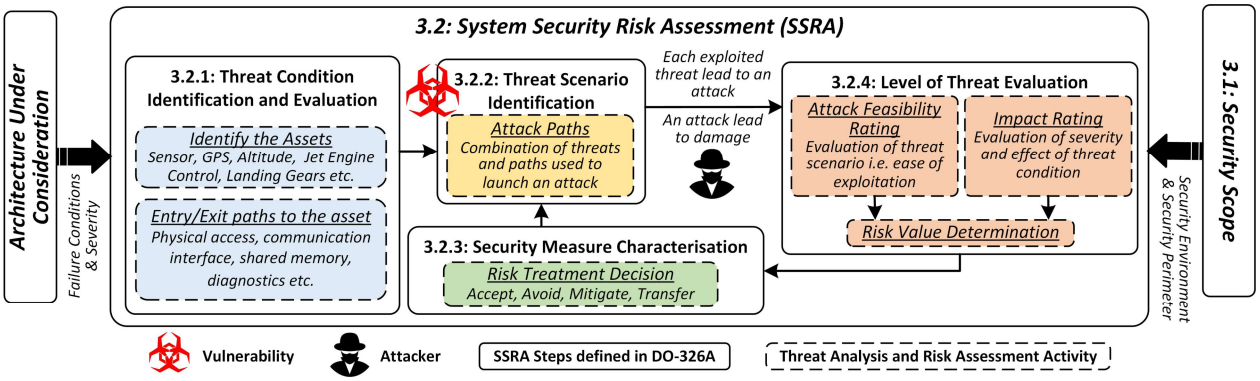


Fig. 3: The system security risk assessment activities defined in DO-326A/ED-202A and mapping of these activities on threat analysis and risk assessment process.

Modeling Tool [22], ThreatModeler [23] etc. These threat modelling tools provide means to methodologically model the use case and to shortlist a list of vulnerabilities and threats that are valid under the given context of the use case, as advocated by EASA [17]. This methodological approach significant decreases time, effort, and cost required to create, maintain system threat models and conduct threat analysis. Once the system's threat model is complete, a detailed *Threat Analysis and Risk Assessment* (TARA) of a given application use case can be conducted as shown in Fig. 1. To conduct TARA, it is essential to:

- Establish the context of the application use case.
- Identify the system's critical assets.
- Identify the threats to each asset.
- Assess potential risks to each asset by estimating the likelihood, that the identified threat will be materialised.
- Quantify the impact of each threat on the system's asset.
- Define and recommend a threat mitigation strategy to ensure the security of the system.

Kohnfelder et al. [24] developed the attack-based STRIDE and DREAD concepts of threat modelling at Microsoft. This work was an important milestone, differentiating between ad-hoc mitigations within the design and a formally produced threat analysis. Focusing on attack scenarios, STRIDE considers threat domains that attack consequences may fall under ***S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of service, and **E**levation of privilege*. The DREAD concept attempts to quantify the threats in terms of factors like criticality and likelihood of occurrence. The acronym represents the terms ***D**amage potential, **R**eproducibility, **E**xploitability, **A**ffected users, and **D**iscoverability*. Within the domain of cyber-physical system, Khan et al. [20] demonstrated the application of STRIDE to cyber-physical systems. Friedberg et al. [21] further described the combination of safety and security threat models within cyber-physical systems, known as STPA-SafeSec. The STRIDE and DREAD models will be considered for TARA of the RA2FAS in Section IV.

III. DO-326A/ED-202A: Airworthiness Security Risk Management Framework

DO-326A/ED-202A defines a seven steps security and risk management framework to ensure the airworthiness of the avionics systems. The steps two and three focus on the ***Security Scope Definition*** and the ***Security Risk Assessment*** processes, which complements the ***①Identify*** and ***②Assess*** principles of the holistic cybersecurity engineering process discussed in Section II-A. These two steps of security and risk management allow system designers and security architects to determine, assess and quantify the risks and their potential damage to the avionics system [18].

DO-326A/ED202A has defined a four-step (3.2.1 - 3.2.4) process to carry out *System Security Risk Assessment* (SSRA) of avionics system as shown in Fig. 3. The process involves threat condition identification and evaluation (3.2.1) by identifying the system's assets such as speed, altitude, and GPS data. It involves identifying entry/exit paths to these system's assets that can be exploited by the attackers which can be physical access to the data. For example: sniffing the data communication interface to extract confidential information, adversely modify the data communication interface, or modify/gain access to data inside the memory etc. A pair of asset and entry/exit paths allows threat scenario identification (3.2.2), where each exploited threat scenario can lead to an attack. Threat scenarios are categorised using a threat method such as STRIDE [20]. A level of threat evaluation activity (3.2.4) is carried out to determine the appropriate level of risk posed by each threat.

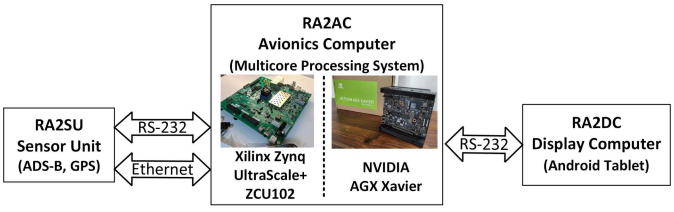


Fig. 4: Overview of Resilient Avionics Architecture for Flight Assistance System (RA2FAS).

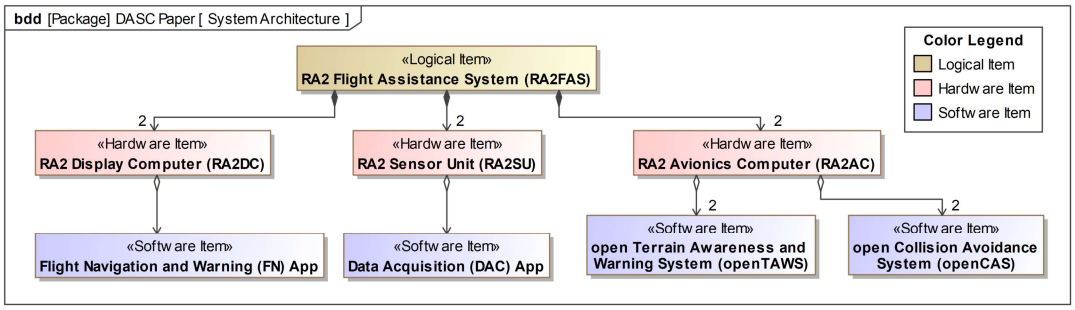


Fig. 5: Architecture diagram of the Resilient Avionics Architecture Flight Assistance System (RA2FAS).

It involves quantifying the feasibility, likelihood, and ease of exploitation of each attack, and its impact on the system's security and functional operations using a risk assessment method such as DREAD. The outcome of this activity is the security measure characterisation (3.2.3) which facilitates security architects to assess and determine an appropriate risk treatment decision i.e. (*accept, avoid, mitigate, or transfer*) the risk as illustrated in Fig 3.

IV. Use Case: Flight Assistance System (RA2FAS)

RA2FAS is an experimental application use case for UAM. It consists of RA2 *Sensor Unit* (RA2SU), RA2 *Avionics Computer* (RA2AC) and RA2 *Display Computer* (RA2DC). The scope of the considered RA2FAS is to provide flight assistance functions such as flight management, flight warning, and risk mitigation. The RA2SU will collect data from the sensors such as an *Automatic Dependent Surveillance–Broadcast* (ADS–B) receiver, *Global Positioning System* (GPS) receiver or air data system as illustrated in Fig. 4. The RA2FAS will process data with a few applications running on a RA2AC to generate advisories, cautions, warnings, and alerts to the pilot and convey them using a serial interface to a ruggedized RA2DC. The XtratuM Next Generation (XNG) hypervisor will be used to ensure spatial and temporal partitioning among mix-critical services and resources of the RA2FAS and to detect policy violations. To conduct flight demonstrations, the RA2FAS is developed as an independent system (free of external dependencies except for electrical power) and targets ultralight aircraft platforms such as a gyrocopter. The scope of this paper is on the cybersecurity engineering process for the RA2FAS. Complementary, the safety engineering process of the RA2FAS is covered in [14], [15].

A. Functional Architecture

The functional architecture of RA2FAS consists of the two top-level functions: situation perception and pilot assistance. The top-level situation perception consists of position and altitude perception, traffic perception, terrain perception and health monitoring functions. The collected situational information facilitates the realisation of the top-level pilot assistance function. It includes multiple threat avoidance functions, terrain avoidance, traffic avoidance, flight warning, and waypoint navigation functions. The required pilot inputs and outputs to these functions are carried out with a *Human Machine Interface* (HMI) available on the RA2DC.

B. Software Architecture

The software architecture of RA2FAS system consists of *Data Acquisition* (DAC) App, open *Terrain Avoidance and Warning System* (openTAWS), open *Collision Avoidance System* (openCAS), *Flight Navigation and Warning* (FN) App as shown in Fig. 5. The DAC App collects data from the sensors which is consumed by DO-367 Class-C openTAWS and openCAS based on DO-385. The FN App guides the operating pilot across the planned route by incorporating advisories and assessments of openTAWS and openCAS. Therefore, it assists the operating pilot to maintain and ensure safe flight operations by mitigating possible risks and fatal accidents.

V. DO326A - Security Scope Definition of RA2FAS

To establish the security scope of the use case, the initial task is to gather the use case *Security Requirements* (SR). This enables system designers and security architects to realise the ①***Identify*** principle of the holistic cybersecurity engineering process (Fig. 1) discussed in Section II-A. It allows the identification of management policies that facilitates the derivation of security goals and objectives. The following are the two of the security requirements:

SR1: Data-in-Motion

- Data authenticity of RA2SU and RA2AC interface.
- Data integrity of RA2AC and RA2DC interface.

SR2: Secure Boot

- Firmware integrity of RA2DC and RA2AC.

TABLE I: Security goals and objectives of RA2FAS.

No.	Security Goal	Security Objective
SR1	The RA2FAS shall prevent unauthorised modification, manipulation, and tampering of the data communication interface between the RA2SU and the RA2AC.	The RA2FAS shall provide data security functions to detect and prevent unauthorised modification, manipulation, and tampering of the data communication interface between the RA2SU (which uses sensor information of ADS-B) and the RA2AC.
SR2	The RA2FAS shall prevent the execution of modified, manipulated, and tampered RA2DC firmware.	The RA2FAS shall provide data security functions to detect and prevent the execution of modified, manipulated, and tampered system code during the platform boot process.

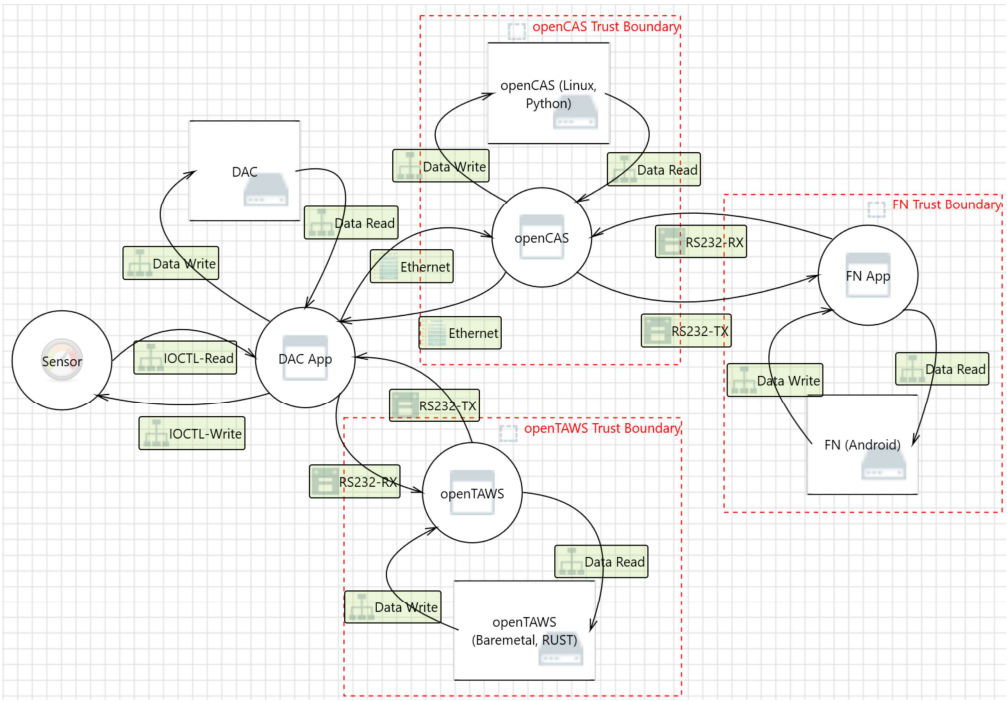


Fig. 6: Threat model of RA2FAS.

Based on SR1 and SR2, corresponding security goals and objectives are defined in Table I.

A. Threat Model

A threat model of RA2FAS is shown in Fig. 6. It is based on the software architecture presented in Fig. 5. The "Microsoft Threat Modelling Tool" [22] has been used for this purpose. The data processing pipeline of RA2FAS begins with the sensor node that captures and processes flight-critical data such as speed, altitude, GPS coordinates etc. This flight-critical information is relayed to the DAC App which reads the sensor data using a read/write data communication channel and stores it in the local memory of the DAC system. Furthermore, the DAC App passes this sensor data to openCAS and openTAWS processes using Ethernet and RS-232 communication interfaces respectively. The data-driven decisions made by the openCAS and openTAWS processes are then forwarded to the FN App using a serial interface, where the results are displayed at the RA2DC. These safety-critical actions are of great importance to ensure the safe and secure operation of the RA2FAS. Therefore, there is a critical need for a systematic TARA process that allows the definition and design of the security architecture of a RA2FAS.

B. Scope, Context and Assumptions

It is important to scope and establish the right context. For this purpose, several assumptions are identified. These assumptions will help security architects to set and define the scope for threat analysis. However, these assumptions are by no means exhaustive and do not encompass every aspect of system security.

- The embedded platform is not physically accessible to unauthorised users. Therefore, the offensive security and relevant attacks are considered out-of-scope.
- The communication interface between the software processes has no built-in security mechanism and is thus vulnerable to attacks.
- The data processing algorithms i.e., used by the software processes are considered intrinsically safe.
- The software process run in user mode.
- DAC App, openCAS, openTAWS software processes are not proprietary, and thus can be investigated and hardened from a security perspective.
- DAC App, openCAS, openTAWS are running in an isolated execution environment as a separate hypervisor partition [25]. Therefore, the attack scenarios such as privilege escalation, and spoofing caused due to inter-partition interference are not considered.
- The I/O communication between software processes (DAC App, openCAS, openTAWS) is managed by the XNG hypervisor [25].
- The software process implements input validation and data sanity checks.
- The sensor node and FN App is running proprietary firmware, hence treated as a black box.

Furthermore, the data interfaces highlighted in Fig. 7 are considered out-of-scope, as the XNG hypervisor memory management policies will enforce memory isolation between software processes. Technically, there is no such thing as a 100% safe and secure system. Given enough time and resources, any system can compromise. The motivation for conducting TARA

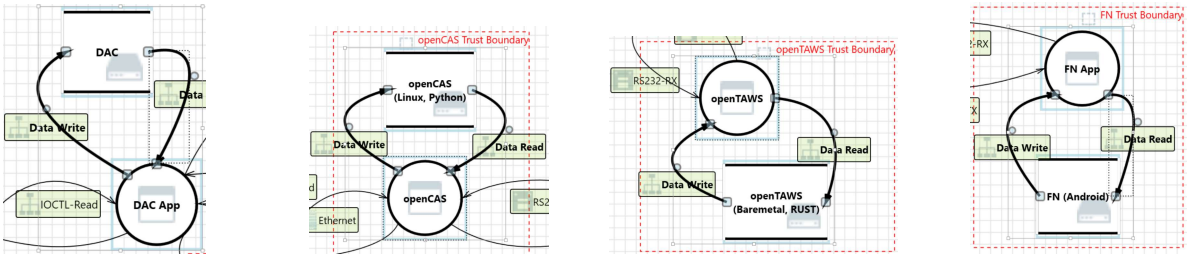
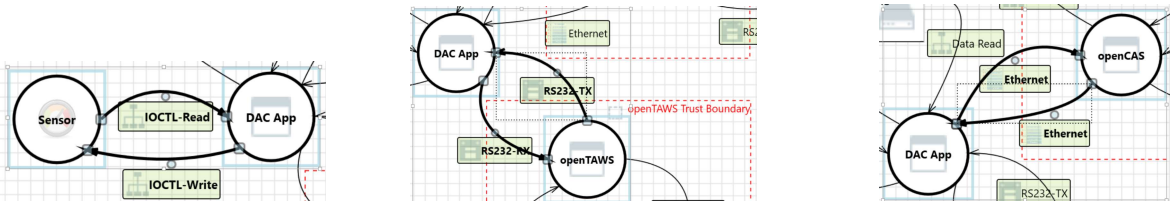


Fig. 7: Data interfaces (highlighted) that are out-of-scope for the system security risk assessment of RA2FAS.



(a) Sensor and DAC App (b) DAC App and openTAWS (c) DAC App and openCAS

Fig. 8: Data interfaces (highlighted) that are in-scope for the system security risk assessment of RA2FAS.

of the RA2FAS application use case is to demonstrate how the discussed holistic cybersecurity engineering process (Fig. 1) facilitates and supports the realisation of a secure-by-design approach for safety-critical embedded systems.

VI. DO-326A - System Security Risk Assessment

An overview of the *system security risk assessment* process has been presented in Section III and the scope of the RA2FAS use case in Section V-B. To build upon, this section presents the detailed TARA of the RA2FAS threat model presented in Fig. 6. To systematically approach this activity, the simplified version of the threat models focusing on each software process is shown in Fig. 8a, Fig. 8b, Fig. 8c. The corresponding TARA is shown in Table II, Table IV, Table III respectively. As discussed in Section II-B, this section will focus on each software process based on the established context, lists down the identified threats, and categorises each threat using the STRIDE model. Furthermore, it assess the risks posed by each threat using the DREAD model and recommends a suitable mitigation strategy to minimise the impact of the threat on the RA2FAS. During TARA process, 62 threats have been identified, out of which 16 have been deemed valid based on the context of the considered RA2FAS use case.

A. Sensor and DAC App

Fig. 8a shows the entry/exit paths and interaction between Sensor and DAC App processes. The DAC App shall pass this digitally stored critical sensor data to openCAS and openTAWS processes which handle the safety-critical advisory generation. It is important to establish the right context to identify the assets and applicable threats relevant to control and data interfaces.

Establish the Context

The sensor node captures and converts the physical properties i.e. altitude, speed etc. into an equivalent digital representation and stores it inside a memory location. The data communication interface between the sensor node and DAC App provides a periodic stream of data from the sensor node. Accordingly, the sensor node passes the digitally stored sensor data to the DAC App.

Additional Assumptions

- Sensor firmware and DAC App execute on a physically separated platform.
- Sensor node does not run in isolated execution env.
- DAC App process does run in isolated execution env.

Identify the Assets

- The digitalised sensor data.
- The variable storing this data inside the sensor node.
- The variable storing this data inside the DAC App.

The detailed TARA of the Sensor and DAC App (Fig. 8a) is presented in Table II.

B. DAC App and openCAS

Fig. 8c shows the entry/exit paths and interaction between DAC App and the openCAS software processes. The openCAS process is responsible for collision avoidance to ensure the safe operation of the aircraft.

Establish the Context

The DAC App is expected to communicate the digital representation of altitude, speed etc. to the openCAS process. An Ethernet communication interface is used with no pre-configured security feature. Upon request, the DAC App shall pass the digitally stored sensor data to the openCAS process.

Additional Assumptions

- Ethernet communication interface is not secure.
- DAC App is not run in an isolated execution env.
- openCAS is run in an isolated execution env.

Identify the Assets

- DAC App – Memory variable storing the sensor data.

TABLE II: Threat Analysis and Risk Assessment of Sensor and DAC App.

Threat Analysis			Risk Assessment				Mitigation Strategy
ID	**Identified Threats**	**Category**	**DREAD**	**Total**	**Avg.**	**Priority**	
T1	Attacker may attempt to alter the sensor node firmware/configuration.	Tampering	8,6,6,4,8	32	6.4	High	Authenticity, Integrity
T2	Attacker may attempt to alter the sensor data at source.	Tampering	7,8,4,8,8	35	7.0	High	Authenticity, Integrity
T3	Attacker may attempt to alter the sensor data during communication.	Spoofing	9,7,8,4,8	36	7.2	High	Confidentiality, Integrity
T4	Attacker may gain access to the critical information, leverage it to reverse-engineer the sensor firmware to design and deploy an exploit.	Information Disclosure	8,5,6,8,3	32	6.4	High	Confidentiality

TABLE III: Threat Analysis and Risk Assessment of DAC App and openCAS.

Threat Analysis			Risk Assessment				Mitigation Strategy
ID	**Identified Threats**	**Category**	**DREAD**	**Total**	**Avg.**	**Priority**	
T5	Attacker may attempt to tamper the openCAS code.	Tampering	7,5,6,8,4	30	6.0	High	Authenticity, Integrity
T6	Attacker may attempt to alter raw Ethernet frames to tamper ($DACApp \Leftrightarrow openCAS$) data communication.	Tampering	7,4,6,5,3	25	5.0	Moderate	Authenticity, Integrity
T7	Attacker may extract classified information from the raw Ethernet packet and leverage it to design/develop/deploy an exploit.	Information Disclosure	6,5,5,5,6	27	5.4	Moderate	Confidentiality
T8	DAC App may be spoofed by an attacker, and this may lead to unauthorised access to openCAS.	Spoofing	5,4,5,6,3	23	4.6	Moderate	Authenticity
T9	openCAS may be spoofed by an attacker and this may lead to information disclosure by DAC App.	Spoofing	5,4,5,6,3	23	4.6	Moderate	Authenticity
T10	openCAS may crash, halt or stop due to wide range of reasons.	Denial-of-Service	8,4,4,8,5	29	5.8	Moderate	Time-bounded control, Runtime Monitoring

TABLE IV: Threat Analysis and Risk Assessment of DAC App and openTAWS.

Threat Analysis			Risk Assessment				Mitigation Strategy
ID	**Identified Threats**	**Category**	**DREAD**	**Total**	**Avg.**	**Priority**	
T11	Attacker may attempt to tamper the openTAWS code.	Tampering	7,5,6,8,4	30	6.0	High	Authenticity, Integrity
T12	Attacker may sniff data ($DACApp \Leftrightarrow openTAWS$) via RS-232 interface and use it to attack other parts of the system OR disclosure of information leading to compliance violations.	Information Disclosure	6,5,5,5,6	27	5.4	Moderate	Confidentiality
T13	The DAC App may crash, halt, stop or run slowly due to wide range of reasons.	Denial-of-Service	8,4,4,8,5	29	5.8	Moderate	Time-bounded control, Runtime Monitoring
T14	Attacker may interrupt data flowing across a trust boundary in RS232-TX direction.	Denial-of-Service	8,3,4,7,4	26	5.2	Moderate	Time-bounded control, Runtime Monitoring
T15	Attacker may sniff data flowing across RS-232-RX. Depending on the type of data, it may be used to attack other parts of the system.	Information Disclosure	5,3,3,5,3	19	3.8	Low	Confidentiality
T16	openTAWS may crash, halt or stop due to wide range of reasons.	Denial-of-Service	8,4,4,8,5	29	5.8	Moderate	Time-bounded control, Runtime Monitoring

DREAD: D=*Damage* ; R=*Reproducibility* ; E=*Exploitability* ; A=*Affected Users* ; D=*Discoverability*.
DO-178C based Threat *(criticality)* Score: *A: Very High (8-9)* ; *B: High (6-7)* ; *C: Moderate (4-5)* ; *D: Low (2-3)* ; E: Extremely Low (0-1).
DO-178C based Risk Evaluation Score: *DAL-A: Catastrophic (8-9)* ; *DAL-B: Hazardous (6-7)* ; *DAL-C: Major (4-5)* ; *DAL-D: Minor (2-3)* ; *DAL-E: No Safety Effect (0-1).*

- openCAS – Memory variables storing the terrain data.
- openCAS – Memory variables storing the collision avoidance data and control.

The detailed TARA of the DAC App and openCAS (Fig. 8c) is presented in Table III.

C. DAC App and openTAWS

The Fig. 8b shows the entry/exit paths and interaction between DAC App and openTAWS software processes. The openTAWS process is responsible for creating terrain related avoidance advisories.

Establish the Context

The DAC App communicates digital representation of altitude, speed etc. to the openTAWS process. A serial communication interface is used for this purpose which has no built-in security features. The DAC App will periodically pass the digitally stored sensor data to openTAWS process which will be used for terrain warning generation.

Additional Assumptions

- openTAWS is implemented in Rust.
- RS-232 interface has no security and is thus vulnerable.
- openTAWS runs in a privileged context (baremetal).
- DAC App runs in an isolated execution env.
- openTAWS runs in an isolated execution env.

Identify the Assets

- DAC App – Memory/local variable storing the sensor data
- openTAWS – Memory variables storing the terrain data
- openTAWS – Memory variables storing the warning data

The detailed TARA of the DAC App and openTAWS (Fig. 8b) is presented in Table IV.

The outcome of the conducted SSRA activity is the list of threat mitigation strategies shown in the Table II, Table III, and Table IV for each considered threat. These threat mitigation strategies enable the system security architect to methodically define the avionics system security model [9] during the modelling phase by choosing appropriate security controls. These security controls minimise the probability and damage posed by a threat. An example of security control includes the use of data:

- Confidentiality methods to protect data at-rest and in-motion.
- Integrity methods to detect data tampering.
- Authenticity methods to establish/verify trustworthiness.

The implementation of these security controls could either be a manual (security extension) or an automated (security pattern [26]). The pattern based extension is part of the cybersecurity engineering process as shown in Fig. 1. During the runtime system phase, these fine-grained security patterns can then serve as building blocks to implement coarse-grained secure lifecycle management (e.g. secure onboarding/off-boarding and secure update) and runtime security monitoring activities. Therefore the realisation of the ④***Detect***, ⑤***Respond***, and ⑥***Recover*** principles is facilitated and the resilience of avionics architectures can be enhanced.

VII. Conclusion & Future Work

The use of advanced sensing, broader connectivity and autonomous technologies is expected to grow in next-generation avionics systems. As a result, it widens the attack surface and exposes them to a wide range of cyber-attacks. Therefore, the existing system design approach of adhering to functional safety standards alone is insufficient to protect advanced avionics systems from cyber-attacks. The ①***Identify*** and ②***Assess*** principles of the holistic cybersecurity engineering process are covered in this paper to bridge the security gap between traditional avionic system design activities and DO-326A/ED-202A risk management guidelines. A comprehensive system security and risk assessment of an experimental flight assistance system is presented and risk mitigation strategies are identified to exhibit realisation of a resilient, secure-by-design avionics architecture. The presented work shows avionics system designers how to effectively identify, assess, and manage cybersecurity risks both at system design and runtime phases. Thus enabling opportunities to improve cybersecurity posture and maintain compliance across the life cycle of the avionics system. In future, the presented work will be further extended by focusing on the ③***Protect*** principle of the cybersecurity engineering process, and cover the implementation of the identified mitigation strategies using pattern-based security.

Acknowledgement

This research work was funded by the European Union's Horizon 2020 Research and Innovation Programme under Grant 957210 (XANDAR).

References

[1] A. P. Cohen, S. A. Shaheen *et al.*, "Urban Air Mobility: History, Ecosystem, Market Potential, and Challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 6074–6087, 2021.

[2] A. Straubinger, R. Rothfeld *et al.*, "An overview of current research and developments in urban air mobility – Setting the scene for UAM introduction," *Journal of Air Transport Management*, vol. 87, p. 101852, 2020.

[3] (2020) 5G market in Aviation by End Use. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/5g-market-aviation-152979610.html

[4] F. Siddiqui and S. Sezer, "Evolution of Embedded Platform Security Technologies: Past, Present & Future Challenges," in *Proc. 33rd IEEE International System-on-Chip Conference (SOCC)*, Las Vegas, USA, 2020, pp. 13–18.

[5] K. Freeman and S. Garcia, "A survey of cyber threats and security controls analysis for urban air mobility environments," in *AIAA Scitech 2021 Forum*, 2021, p. 0660.

[6] (2022) Aviation is facing a rising wave of cyber-attacks in the wake of COVID. [Online]. Available: https://www.shlegal.com/insights/aviation-is-facing-a-rising-wave-of-cyber-attacks-in-the-wake-of-covid

[7] (2019) U.S. Flights Canceled as FAA Looks into GPS, ADS-B System Errors. [Online]. Available: https://insidegnss.com/u-s-flights-canceled-as-faa-looks-into-gps-ads-b-system-errors/

[8] F. Siddiqui, R. Khan *et al.*, "Bird's-eye view on the Automotive Cybersecurity Landscape & Challenges in adopting AI/ML," in *Proc. 6th IEEE International Conference on Fog and Mobile Edge Computing (FMEC)*, Gandia, Spain, 2021, pp. 1–6.

[9] F. Siddiqui, R. Khan *et al.*, "XANDAR: A holistic Cybersecurity Engineering Process for Safety-critical and Cyber-physical Systems," in *Proc. 95th IEEE Vehicular Technology Conference (VTC2022-Spring)*, Helsinki, Finland, 2022, pp. 1–5.

[10] F. Siddiqui, M. Hagan *et al.*, "Establishing Cyber Resilience in Embedded Systems for Securing Next-Generation Critical Infrastructure," in *Proc. 32nd IEEE International Conference on System-on-Chip Conference (SOCC)*, Singapore, 2019, pp. 218–223.

[11] C. Torens, A. Volkert *et al.*, "HorizonUAM: Safety and Security Considerations for Urban Air Mobility," in *AIAA Aviation 2021 Forum*, 2021.

[12] J. Becker, L. Masing *et al.*, "XANDAR: X-by-Construction Design framework for Engineering Autonomous & Distributed Real-time Embedded Software Systems," in *Proc. IEEE International Conference on Field-Programmable Logic and Applications (FPL)*, Dresden, Germany, 2021, pp. 382–383.

[13] T. Dörr, F. Schade *et al.*, "Safety by Construction: Pattern-Based Application of Safety Mechanisms in XANDAR," in *Proc. IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Nicosia, Cyprus, 2022, pp. 369–370.

[14] A. Ahlbrecht and U. Durak, "Integrating safety into mbse processes with formal methods," in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, 2021, pp. 1–9.

[15] A. Ahlbrecht and U. Durak, "Model-Based STPA: Enabling Safety Analysis Coverage Assessment with Formalization," in *Proc. 41st IEEE/AIAA Digital Avionics Systems Conference (DASC)*, Portsmouth, US, 2022, pp. 1–10.

[16] "Framework for Improving Critical Infrastructure Cybersecurity," National Institute of Standards and Technology (NIST), Tech. Rep., Apr. 2018. [Online]. Available: https://www.nist.gov/cyberframework/framework

[17] "EASA Concept Paper: guidance for Level 1 & 2 machine learning applications," European Union Aviation Safety Agency (EASA), Tech. Rep., 2023. [Online]. Available: https://www.easa.europa.eu/en/document-library/general-publications/easa-artificial-intelligence-concept-paper-proposed-issue-2

[18] C. Torens, "Safety versus security in aviation, comparing DO-178C with security standards," in *AIAA Scitech 2020 Forum*, 2020, p. 0242.

[19] (2023) Threat Modelling Cheat Sheet. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Threat-Modeling-Cheat-Sheet

[20] R. Khan, K. McLaughlin *et al.*, "STRIDE-based threat modeling for cyber-physical systems," in *Proc. IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, Turin, Italy, Sep. 2017, pp. 1–6.

[21] I. Friedberg, K. McLaughlin *et al.*, "STPA-SafeSec: Safety and security analysis for cyber-physical systems," *Journal of Information Security and Applications*, vol. 34, pp. 183–196, 2017.

[22] (2023) Microsoft Threat Modeling Tool. [Online]. Available: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool

[23] (2023) ThreatModeller - Automated Threat Modelling Solution. [Online]. Available: https://threatmodeler.com/

[24] L. Kohnfelder, *Designing Secure Software: A Guide for Developers*. No Starch Press, 2021.

[25] (2023) XtratuM Hypervisor - Bring the power of virtualisation. [Online]. Available: https://www.fentiss.com/xtratum-h/

[26] H. Martin, Z. Ma *et al.*, "Combined automotive safety and security pattern engineering approach," *Reliability Engineering & System Safety*, vol. 198, p. 106773, 2020.