UK Civil Aviation Authority

Safety and Security of Cloud Computing

CAP 3254



CAP 3254

Published by the Civil Aviation Authority, 2026

Civil Aviation Authority
Aviation House
Beehive Ring Road
Crawley
West Sussex
RH6 0YR

You can copy and use this text but please ensure you always use the most up to date version and use it in context so as not to be misleading, and credit the CAA.

First published May 2026
Second edition October 2026

Enquiries regarding the content of this publication should be addressed to: cyber@caa.co.uk

The latest version of this document is available in electronic format at: www.caa.co.uk/

Revision History

Edition	Date	Revision
First edition	May 2026	Initial publication of CAP
Second edition	October 2026	Revised following industry and stakeholder feedback. Expanded guidance on scope and proportionality. Introduced Annex A to describe integrating of exiting security risk assessment with the safety and security co-engineering approach.

Contents

Scope **6**
Assumptions 7
Acronyms **8**
Introduction **9**
Cloud Architecture in Aerospace Applications **10**
Deployment Models 10
Core Cloud Technologies 11
Architectural overview 12
Cloud-Hosted Applications and Services 13
Architectural Principles 14
Safety and Security of Cloud Architecture **15**
Cloud layer 16
Communication layer 18
Edge Layer 19
Safety and Security Co-Engineering **20**
Analysis Techniques 20
Integrating the techniques 22
Mapping Co-Engineering to Cloud Layers 22
Co-engineering as a lifecycle activity 23
Co-engineering in Federated and Multi-Cloud Systems **27**
Annex A – Integration with Security Risk Management **29**
A.1 Purpose 29
A.2 Relationship with established security frameworks 29
A.3 Security risk management as an input to safety and security co-engineering 30
A.4 Safety analyses as an input to security risk management 31
A.5 Traceability and assurance evidence 31
A.6 Lifecycle integration 32
A.7 Application across organisational boundaries 32

Scope

This Civil Aviation Publication (CAP) provides guidance on the safety and security assurance of cloud-integrated architectures used within aerospace operational technology systems.

The guidance applies to civil aviation and space domain applications where cloud computing, virtualised infrastructure, or distributed digital services support functions that are safety-related or safety-influencing.

This publication is applicable to:

- Operators conducting Beyond Visual Line of Sight (BVLOS) or remotely piloted aircraft operations supported by cloud services.
- Air Traffic Management (ATM) and U-space service providers adopting virtualised or cloud-hosted systems.
- Organisations implementing cloud-based telemetry, tracking and control (TT&C) or mission data processing services within space operations.
- Hybrid, multi-cloud, and federated cloud environments where safety-relevant services are distributed across organisational or technical boundaries.
- Organisations providing Software-as-a-service or other cloud hosted applications that support, influence, or enable safety related aerospace operational functions, including mission planning, operational coordination, compliance management, telemetry processing or decision support services.

This publication:

- Describes architectural principles for integrating cloud infrastructure within safety-relevant systems.
- Provides guidance on the allocation of safety and security responsibilities between system operators and cloud service providers.
- Sets out analytical approaches, including Functional Hazard Assessment (FHA), Failure Modes and Effects Analysis (FMEA), and System-Theoretic Process Analysis (STPA and STPA-Sec), for the assessment of distributed cloud-enabled systems.
- Addresses the treatment of Service Level Agreements (SLAs) and Service Level Objectives (SLOs) as elements of the safety and security assurance framework.
- Guidance on integrating existing security risk management activities with the safety and security assurance approach described in this CAP is provided in Annex A

This publication does not:

- Replace, supersede, or modify existing certification requirements, regulatory obligations, or applicable airworthiness standards.
- Mandate specific technologies, vendors, or commercial cloud service models.
- Provide detailed cyber threat intelligence or standalone cybersecurity certification/assurance guidance.
- Provide specific guidance on AI or ML assurance. Where AI or ML capabilities are referenced, they are used solely to illustrate cloud-hosted services that may contribute to safety-related functions.

The guidance is intended for system designers, operators, service providers, cloud infrastructure providers supporting safety-relevant services, and regulatory personnel responsible for oversight of cloud-enabled aerospace systems. Furthermore, it is intended to be applied proportionately to the operational context and contribution of cloud services to operational safety.

Assumptions

This publication assumes that the reader:

- Is familiar with established aerospace safety assessment processes, including Functional Hazard Assessments (FHA), Failure Modes and Effects Analysis (FMEA).
- Possess working knowledge of cybersecurity principles relevant to operational technology environments.
- Understands basic cloud computing concepts including virtualisation, containerisation, service orchestration, and Service Level Agreements (SLAs)
- Is engaged in the design, assurance, operation, or oversight of safety-relevant aerospace systems.

This publication does not provide introductory training in safety engineering, cybersecurity, or cloud computing fundamentals.

Acronyms

Advanced Message Queuing Protocol: AMQP

Air Traffic Management: ATM

Application Programming Interface: API

Artificial Intelligence: AI

Beyond Visual Line of Sight: BVLOS

Detect and Avoid: DAA

Failure Mode and Effects Analysis: FMEA

Functional Hazard Assessment: FHA

Machine Learning: ML

Message Queuing Telemetry Protocol: MQTP

Quality of Service: QoS

Service Level Agreements: SLA

Service Level Objectives: SLO

Software Defined Networks: SDN

System-Theoretic Process Analysis: STPA

Telemetry, Tracking and Control: TT&C

Time to Live: TTL

Unmanned Air System: UAS

Introduction

The aerospace sector is undergoing a rapid digital transformation, driven by advances in connectivity, automation, and data analytics. Across both aviation and space domains, operations are becoming increasingly networked and data centric. Cloud computing now plays a vital role in this development by providing scalable computational resources, secure data storage, and globally accessible analytics for safety influencing/safety related systems.

Cloud-based architectures are being adopted for a wide range of aerospace applications. In the aviation domain, they are enabling Beyond Visual Line of Sight (BVLOS) operations, predictive maintenance through machine learning, and Detect and Avoid (DAA) systems for Unmanned Air System (UAS) traffic management. Conventional Air Traffic Management (ATM) systems are also increasingly adopting cloud and virtualised infrastructure to support modernisation, data distribution, operational resilience. In the space domain, they support satellite Telemetry, Tracking & Control (TT&C) processing, and mission data management, allowing ground operations to manage resources using globally distributed cloud infrastructure and ground stations.

These developments mark a significant shift from traditional architectures, which have historically been deterministic, tightly controlled, and physically isolated systems. Cloud environments are by contrast elastic, shared, and continuously developing, thus introducing new complexity and interdependence.

These developments raise several critical questions:

- How can cloud platforms be safely integrated into safety influencing/safety related aerospace systems?
- How can safety and security assurances be maintained across externally managed distributed infrastructures?
- What analytical methods can effectively capture the interactions between physical systems, software, and cloud services?

This CAP addresses these questions by linking cloud architectures with safety and security analysis frameworks. It demonstrates how both traditional reliability-based methods and system theoretical approaches can be applied to assess and manage the risks of cloud integrated systems. The focus will not be on describing generic cloud technologies but on understanding their implications for operational safety, security, and continued assurance in aerospace applications.

Cloud Architecture in Aerospace Applications

Cloud computing has become an enabler of modern aerospace operations, providing the computational elasticity, data integration, and global reach once limited by physically co-located systems. For safety influencing/safety related applications however, cloud architectures must be designed not only for performance, but also for predictability, verifiability, and controlled development.

Deployment Models

Different deployment models govern how control, visibility, and assurance responsibilities are distributed between the operating organisations and cloud providers. Each model presents a different balance between control, scalability, and transparency.

Public Cloud

Operated by third-party providers, shared across multiple tenants.

Potential Assurance Benefit: Elastic scalability, rapid access to advanced computing and Artificial Intelligence (AI) capabilities.

Challenges: Limited infrastructure transparency, varying data jurisdiction rules.

Private Cloud

Dedicated infrastructure operated by a single organisation, either on-premises or exclusive use through a third-party host.

Potential Assurance Benefit: Greater organisational control over configuration, resource allocation, and audit activities, together with the opportunity to engineer deterministic performance where required.

Challenges: Higher maintenance and lifecycle cost.

Hybrid Cloud

Combines Private and Public resources via secure interfaces.

Potential Assurance Benefit: Balances scalability and control; supports selective segregation of workloads.

Challenges: Requires synchronisation and unified policy enforcement.

Multi-Cloud

Uses multiple independent providers for resilience and flexibility.

Potential Assurance Benefit: Reduced dependence on a single provider and improved resilience where sufficient independence between cloud services, supporting infrastructure, and operational dependencies can be demonstrated.

Challenges: Complex identity and configuration management.

Federated Cloud

A cooperative network of autonomous clouds governed by shared trust, identity, and policy frameworks.

Potential Assurance Benefit: Facilitates collaboration while retaining domain autonomy.

Challenges: Requires consistent governance, and an understanding of shared organisational and technical dependencies between participating domains.

Edge and Fog integration

Extend cloud resources closer to data sources; airborne, ground, or orbital.

Potential Assurance Benefit: Reduced latency and continued operation during network disruption.

Challenges: Ensuring state synchronisation and coherent decision authority across distributed nodes.

The potential assurance benefits associated with each deployment model are not inherent characteristics of the deployment itself and should not be assumed. They are dependent on the architecture, implementation, configuration, governance, and operational management of the cloud environment.

Claims of independence between redundant services, deployment environments, or cloud providers should identify and assess shared dependencies, including identity and access management services, telecommunication, management planes, software supply chain, shared infrastructure, and external service providers that could result in common mode failures.

Core Cloud Technologies

Modern cloud ecosystems rely on a portfolio of technologies that together provide scalability, isolation, and verifiable control.

- **Virtualisation and Hypervisors**: provide logical separation of workloads and abstraction of computing resources enabling controlled resource allocation, and fault containment where appropriately designed and configured.
- **Containerisation and Microservices**: support module deployment of software components, enabling independent lifecycle management, fault isolation, and targeted verification of individual services within a distributed architecture.
- **Orchestration and Service Meshes**: automate deployment, scaling, and lifecycle management of distributed workloads whilst providing policy enforcement, service discovery, communication management, and operational observability.
- **Data Streaming and Message Queues**: facilitate the reliable exchange of

operational data between distributed system components. Where supporting safety related functions, message delivery characteristics such as ordering, acknowledgement, latency, duplicate handling, and recovery behaviour should be defined and demonstrated as part of the overall assurance activities.

- **Edge and Fog Computing**: provide distributed processing closer to the operational environment, supporting reduced communication latency, local resilience, and continued operation where connectivity to cloud service is degraded or unavailable.
- **Storage and Data Management**: provide persistent storage of operational information through controlled mechanism such as replication, versioning, immutability, and backup, support data integrity, traceability, and recovery.
- **Serverless and Event Driven Computing**: support execution of discrete operational function in response to defined events, enabling scalable processing of non-time critical activities such as monitoring, alerting, workflow automation, and log analysis.
- **Software Defined Networks and Network Function Virtualisation**: enable programmable network management and flexible deployment of communication services, supporting controlled routing, redundancy, segmentation, and operational resilience.
- **Security, Identity, and Key Management**: provide authentication, authorisation, cryptographic key management, and trust services that support secure access to distributed cloud resources and protection of safety relevant functions.

Where cloud technologies support safety related functions, the operational characteristics on which the safety argument depends should be identified and demonstrated through appropriate verification and validation activities.

Architectural overview

Across deployments and operational contexts, cloud systems in operational technology environments typically adopt a three-layer hybrid architecture, defining how computation, communication, and controls are distributed.

- **Edge layer:** onboard avionics, ground control stations, and mission computers executing time-critical functions such as stabilisation and collision avoidance.
- **Communication layer**: terrestrial and satellite networks providing redundant, encrypted, latency-controlled connectivity.
- **Cloud layer**: scalable computation, storage, and analytics platforms hosting mission data management, simulation, and coordination systems.

These layers form an integrated control loop. The edge generates the data and immediate control actions, the communication layer transmits and synchronises the information, and the cloud aggregates, analyses, and redistributes decisions or insights.

From a safety perspective, this structure provides fault containment and clear authority boundaries. However, control logic that spans layers, together with shared supporting services such as identity and access management, key and certificate management, time synchronisation, domain name services, monitoring platforms, management planes and telecommunications may introduce misconfiguration or timing faults or common dependencies that propagate across multiple layers. These interactions require holistic, cross layer assurance methods that consider behaviour of the integrated system rather than individual architectural layers in isolation.

Cloud-Hosted Applications and Services

Cloud infrastructure does not operate in isolation; it hosts a portfolio of applications and services that support, augment, or coordinate aerospace operations. These cloud-hosted applications vary significantly in their criticality, timing sensitivity, and assurance requirements, and must therefore be explicitly identified and assessed within the system architecture.

The safety significance of a cloud hosted application is determined by the operational function it performs and the extent to which the overall safety argument depends upon its behaviour, rather than by the technology through which the service is delivered. Applications that do not directly control safety related functions may nevertheless influence operational decision making and therefore contribute to hazardous system states if they fail, become unavailable, or provide incorrect information.

Typical cloud hosted applications in aerospace operations include:

- **Mission data management services,** including telemetry aggregation, storage, replay, and post-mission analysis.
- **Situational awareness and coordination services,** such as traffic information distribution, airspace status updates, and collaborative planning tools.
- **Predictive analytics and machine-learning services,** supporting maintenance forecasting, anomaly detection, or advisory decision support.
- **Simulation and digital-twin environments,** used for operational rehearsal, contingency evaluation, or performance monitoring.
- **Identity, authorisation, and trust services**, including credential management, certificate authorities, and policy enforcement points.
- **Operational monitoring and observability services**, aggregating logs, metrics, and security events across distributed components.

While many of these applications are not directly safety influencing/safety related in

isolation, they frequently influence safety decisions by providing advisories, constraints, or shared situational context. As a result, their failure, corruption, or delay may still contribute to hazardous system states.

Each cloud-hosted application must therefore be:

- Mapped to the operational functions it supports.
- Assessed to determine how degradation, corruption, delay or loss of the service could influence operational safety.
- Assigned an assurance approach proportionate to it contribution to the overall safety argument.
- Included within the appropriate safety and security analyses where the service contributes to operational decision making or safety related functions.
- Subject to architectural constraints that prevent non-critical services from adversely affecting safety related behaviour.

Where cloud hosted applications provide information that supports operational or safety related decisions, organisations should also assess the quality of the information upon which those services depend. This should include considerations of data provenance, integrity, accuracy, consistency, timeliness, and validity, together with the potential operational consequences should those characteristics no longer be satisfied.

Furthermore, if cloud hosted applications incorporate AI or ML models that support or influence safety related functions, assurance should also consider the configuration and version of the deployed model, the quality and relevance of data upon which its behaviour depends, known performance limitations, and the potential for performance degradation or drift within the operational environment. Changes to models, associated data, or their deployment configuration should be subject to appropriate verification, validation, change control, and rollback arrangements proportionate to their contribution to the overall safety argument.

This explicit identification of cloud applications prevents implicit reliance on services whose failure modes or assurance boundaries are otherwise obscured by the underlying infrastructure abstraction.

Architectural Principles

Cloud systems should apply redundancy, determinism, and segregation within a distributed digital system where required by the safety argument. The following principles form the foundation of safe and secure cloud architecture.

- **Domain Separation**: logical and physical isolation between mission critical and non-critical workloads prevents cascading faults and exposures to unverified components.
- **Redundancy**: redundant compute nodes, storage replicas, and communication

paths enable continued functions under component or link failures.

- **Data Integrity and Provenance**: each transaction, telemetry or control must be verifiable, timestamped, and auditable to reconstruct system state and decision.

- **Deterministic Performance within Elasticity**: resource scaling must maintain predictable timing behaviour. Reserved resources or edge co- processing mitigate the non-deterministic nature that is typical of shared clouds.

- **Secure by Design**: authentication, encryption, and access controls are embedded at every layer, ensuring that security mechanisms directly support safety objectives.

- **Continuous Observability**: unified monitoring of performance, integrity, and security provides early detection of anomalies and assurance continuity throughout the operation.

Safety and Security of Cloud Architecture

Digitisation of the aerospace ecosystem blurs the boundary between security and safety. As functions migrate into software, virtualised infrastructure, and cloud services both random, and malicious actions act on the same digital pathways; message buses, Application Programming Interfaces (APIs), orchestration layers, and identity systems. As a result, the mechanisms that produce safety hazards and security breaches increasingly overlap.

Therefore, cloud infrastructure depends upon four operational properties that are shared by both safety and security:

- **Integrity**: ensuring that data, software, and system behaviour remain accurate, complete, and free from unauthorised modification.
- **Availability**: ensuring that cloud services, communications, and computation resources remain accessible when required to support operational functions.
- **Authenticity**: ensuring that users, devices, services, and data services can be verified and trusted.
- **Timeliness:** ensuring that operational information and services are delivered within the temporal constraints assumed by the safety argument.

Loss of any of these properties whether by random failure or malicious actions can create hazardous states. In cloud-integrated systems, a loss of timeliness can have the same operational effect as loss of availability. Excessive latency, jitter, or delayed delivery of otherwise valid data or services can invalidate safety assumptions and lead to hazardous states, even when infrastructure components remain nominally operational. The maintenance and demonstration of these operational properties are supported by broader

assurance characteristics, including resilience, recoverability, accountability, and traceability. These characteristics enable organisations to withstand, recover from, investigate, and provide evidence following failures or security events. Confidentiality should also be considered where the unauthorised disclosure of information could contribute to a safety risk enabling subsequent compromise of the operational system.

These hazardous states arising from cloud integrated architecture should be managed within an organisations existing Safety Management System (SMS), recognising that cloud services may form part of the operational system and therefore contribute to operational hazards requiring identification, assessment, mitigation, and ongoing monitoring.

Building on these operational properties, the safety and security considerations for cloud architecture of an operational technology system can be examined across complementary layers:

Cloud layer

The cloud layer provides the computational, storage, orchestration, and coordination services that support the operation of cloud enabled aerospace systems. Where these services contribute to safety related functions, they should maintain the integrity, availability, authenticity, and timeliness required by the established safety argument. Safety related cloud services should operate within defined performance and operational constraints, with mechanisms to detect deviation and maintain the established safety objectives.

Data integrity and provenance of all data used in safety must be verifiable and traceable to its source. Configuration and change management should be adhered to across system configurations, with all software releases undergoing rigorous change control and impact assessment to verify that no update introduces unsafe or insecure conditions.

Continuous monitoring of cloud performance and early detection of faults or intrusions can help identify issues, such as latency drift or data corruption that could impact safety. Only authorised personnel and processes should be able to modify, access, or operate safety related components; the principle of least privilege, combined with authentication, prevents unauthorised modifications or intrusions.

Service Level Agreements and Assurance Responsibilities

The safety and security of cloud-integrated aerospace systems depend not only on technical architecture, but also on the explicit assurance commitments between cloud service providers and system operators. These commitments are typically formalised through Service Level Agreements (SLAs) and Service Level Objectives (SLOs), which define the expected performance, availability, and support characteristics of cloud services.

For safety-relevant applications, SLAs must be treated as assurance artefacts rather than commercial conveniences. Availability guarantees, latency bounds, recovery times, data

durability, and change notification obligations directly influence whether safety objectives can be met in operation. SLAs should explicitly address:

- **Availability and resilience**, including service availability, redundancy arrangements, recovery objectives, and the operational conditions under which degraded modes may be entered.
- **Performance and latency objectives**, including maximum latency, data freshness, error rates, throughput and other timing characteristics where these form part of the operational safety assumptions.
- **Change management and notification**, ensuring that infrastructure updates, maintenance activities, or configuration changes do not invalidate safety assumptions without prior assessment.
- **Incident response and escalation**, including fault reporting, notification times, escalation procedures, and support for operational decision making during degraded service conditions.
- **Auditability and transparency**, allowing operators and regulators to verify compliance with declared assurance claims and investigate operational events where necessary.

Where cloud services support safety related functions, the parameters defined in SLA and SLOs should be directly traceable to the operational assumptions established by the system safety assessment. Commercial service measures, such as monthly availability targets or service credits, do not by themselves demonstrate that operational safety objectives have been achieved. Safety related service parameters should therefore reflect the needs of the operational function and be supported by measurable evidence appropriate to the system under consideration.

The allocation of responsibility between cloud provider and system operator must be unambiguous and supported by a documented responsibility model. While the provider may be responsible for infrastructure availability and baseline security controls, the operator retains responsibility for demonstrating that the overall safety objectives of the operational system continue to be satisfied. The documented responsibility model should identify responsibilities for operational control, assurance evidence, monitoring, incident management, change notification, and supporting activities provided by subcontractors or managed service providers where applicable.

Where safety objectives depend on cloud service behaviour, those objectives must be traceable to defined and measurable service parameters. Where an SLA or SLO cannot demonstrate that the required operational assumptions can be maintained, the system architecture should incorporate appropriate compensating controls, such as local autonomy, buffering, degraded modes of operation, or other resilience mechanisms. The operational response to the loss of a safety relevant service assumption should be defined such that the established safety objective continues to be maintained.

Data Sovereignty and Cross-Border Assurance

Cloud enabled systems frequently distribute operational data across multiple geographical regions, cloud availability zones, or national jurisdictions to achieve resilience, scalability, and operational continuity. While these characteristics provide significant operational benefits, they may also introduce assurance considerations associated with data residency, legal jurisdiction, regulatory oversight, and cross-border access to operational information.

Organisations should understand where safety relevant operational data is stored, processed, replicated, and backed up, together with the legal and regulatory frameworks that govern those locations. Differences in national legislation, contractual arrangements, or regulatory requirements may affect incident response, auditability, regulatory access, or the continued availability of safety related services. Where cloud services form part of the operational system, data sovereignty consideration should be evaluated as part of the overall safety and security assurance activities. This includes assessing whether cross border data management could affect the integrity, availability, or timely accessibility of operational information required to support safe operation.

Architectural or operational mitigations may include defined data residency requirements, regional deployment strategies, resilience across trusted jurisdictions, contractual controls, and contingency arrangements to maintain operational continuity should access to cloud services become restricted or unavailable.

Communication layer

This layer connects the cloud services with edge systems and supports the exchange of operational data between distributed components. Communication services should support the integrity, availability, authenticity, and timeliness required to enable safe operation. Performance requirements relating to latency, freshness integrity, ordering, and availability should therefore be explicitly defined for safety relevant data flows. These requirements should be monitored throughout operation and supported by mechanisms that detect when established performance requirements are no longer satisfied and initiate an appropriate degraded or safe operational response.

The integrity of the communication system should be maintained by protecting against unauthorised command injection, spoofing, message alteration, and data corruption. This may be achieved through mechanisms such as digital signatures, mutual authentication, and integrity verification. Transmission errors should be detected and managed using appropriate techniques such as sequence validation, checksums, acknowledgements, and retransmission strategies suitable for the operational environment.

Availability should be supported through resilient communication architecture employing redundancy, path diversity, and independent communication technologies where appropriate. Communication failures and degraded communication performance should be considered within the system safety assessment and accommodated through appropriate architectural and operational mitigations. Where communication performance

no longer satisfies the established performance requirements, the operational system should maintain safety through predefined degraded modes, local capability, or other resilience mechanisms. Logical and physical separation of safety related, and non-safety related traffic should also be considered to minimise resource contention and reduce the potential for non-critical services to adversely affect safety related communications.

Timeliness should be achieved by ensuring that safety relevant information is delivered within the temporal constraints established by the system safety assessment. Mechanisms such as traffic prioritisation, Quality of Service (QoS), dedicated communication paths, or equivalent approaches may be used, where appropriate to support predictable communication performance. Time synchronisation should be maintained using trusted sources, and stale, delayed or replayed information should be identified and rejected where it could adversely affect operational decision making. Where latency, freshness, or ordering performance requirements can no longer be satisfied, the condition should be detected, and the operational system should transition to an assessed degraded or safe operational state consistent with the established safety objectives.

Edge Layer

The Edge layer interfaces directly with operational environment and typically comprises of onboard systems, sensors, actuators, ground based operational equipment, and local processing nodes. It supports the execution of time sensitive operational functions and provides the local capability necessary to maintain safe operation where communication cloud services are degraded, unavailable, or unable to satisfy the performance requirements established by the system safety assessment.

Where cloud services support safety related functions, the system architecture should ensure that the loss, degradation or corruption of those services does not result in an unassessed unsafe condition. This may be achieved through appropriate combinations of local processing, independent functionality, predefined degraded modes, procedural mitigations, or other resilience mechanisms proportionate to the contribution of the cloud services to the overall safety argument.

Safety relevant edge function should preserve the integrity, authenticity, and timeliness of operational data, detect failures or degradation affecting cloud services, and implement appropriate responses that maintain the established safety objectives until normal operation can be restored. Where synchronisation with cloud hosted services is required, mechanism should be provided to maintain data consistency and recover gracefully following communication loss or restoration. Recovery procedure should ensure that stale, incomplete or conflicting operational information does not adversely affect the continued safe operation of the system.

Safety and Security Co-Engineering

Cloud enabled aerospace systems are too dynamic to be able to treat safety and security as two separate sequential approvals. Services scale up and down, network paths change, data schemas evolve, Machine Learning (ML) models are retrained, yet the safety case must still hold. Integrated safety-security co-engineering means that safety and security are:

- Defined together – with the same function, same hazards.
- Implemented together – with the same architecture, same policies.
- Verified together – with the same monitoring, same evidence.

In classic aerospace systems, safety is about component reliability and fault tolerance, security is about keeping attackers out. In a cloud system:

- A security failure such as an unguarded API, expired certificate, or poisoned data stream can look like a safety failure such as a stale command, missing telemetry, or wrong advisory.
- A safety mechanism like the throttling of non-critical traffic can create security gaps like an operator becoming forced to bypass controls.
- A benign cloud behaviour like auto scale, node eviction, or route change can invalidate safety timing assumptions.

Integrated safety-security assurance for cloud-based aerospace systems relies on combining analytical methods that address various levels of system abstraction. No single technique can model both the functional purpose of the system and the dynamic, distributed interactions that are characteristic of modern cloud platforms.

Analysis Techniques

The three methods this section will address are Functional Hazard Assessment (FHA), Failure Modes and Effects Analysis (FMEA), and System-Theoretic Process Analysis (STPA and STPA-Security).

FHA is the natural starting point because cloud environments change what counts as a function in an aerospace system. Traditional avionics might treat flight control as a single function; however, a cloud integrated system could deconstruct it into on-board control, data relay, remote planning, and AI-based advisory services.

FHA forces explicit definition of which functions are safety influencing/safety related, what the operational consequences are in case of delay, corruption, etc and what safety objectives or integrity levels each function must satisfy. In the cloud context, this is vital because a function’s failure may result not from a physical fault but from virtualisation, network partitioning, or provider reconfiguration. Therefore, this technique anchors the entire assurance process in functional intent, defines the quantitative bounds for latency, freshness, and trust. This provides the first mapping of security events, such as credential

compromise which leads to safety outcomes like unauthorised control. While FHA describes the what and the so what, FMEA investigates the how. Cloud architecture contains many moving parts such as message brokers, orchestrators, virtual machines, APIs, etc; each with its own failure and recovery modes.

FMEA suits this environment because it manages a large number of components systematically. It links failure modes like pod evictions, queue overflow, token expiry, etc to system level effects defined in FHA. FMEA enables failure modes to be assessed in terms of their effects, severity and, where appropriate, likelihood or detectability, supporting the prioritisation of mitigations and translation of functional hazards into specific design or configuration controls.

FMEA also bridges the safety-security gap - many failures in cloud systems are non-malicious manifestations of security lapses like expired certificates or broken trust chains. By treating these within the same framework, we avoid duplication of analysis and ensure consistent risk scoring.

Even with FHA and FMEA, some failures will not be captured because they involve emergent behaviours rather than discrete component faults. STPA models the system as a hierarchy of control loops and examines how unsafe actions arise when control, feedback, or contextual information is missing, incorrect, delayed, or untrusted.

In the cloud system, controllers include autonomy modules, cloud planners, orchestration policies, and human operators. Feedback paths include telemetry streams, observability data, and security alerts. Unsafe actions may stem from delayed data, loss synchronisation, competing authority, or policy drift – none which will appear in conventional FMEA tables. STPA-Security (STPA-Sec) extends this reasoning to security triggered hazards such as issuing valid commands with forged credentials, misapplying safety logic due to tampered configuration, or losing safe control when trust or policy information becomes inconsistent across providers. In this system, these unsafe control actions rarely arise in isolation but are typically the result of identifiable adversarial behaviours acting on exposed digital control surfaces.

Because cloud infrastructure exposes safety functions to digital attack surfaces, the STPA-Sec activity should incorporate inputs from a focused cyber-threat assessment. The cyber-threat assessment may form part of an organisation’s existing security risk management activities and does not require a separate or prescribed security risk assessment methodology. Relevant outputs from those activities should be used as inputs to the co-engineering process where security conditions could contribute to hazardous system states. Annex A describes this interface in further detail. This allows malicious triggers such as API misuse, token theft, data poisoning, or cloud orchestrator manipulation to be treated as causal factors within the co-engineered FHA- FMEA-STPA workflow, rather than a separate security analysis.

Integrating the techniques

Each of these techniques feed into the next, ensuring that safety and security information remains consistent across layers and through time. FHA identifies critical functions and hazard severities whilst FMEA maps those functions to physical and virtual components, exploring how failures in hardware, software, or service configurations might produce functional loss. The severity classifications from FHA guide the weighting and prioritisation within FMEA. The high-risk failure modes identified in FMEA are inserted as disturbances in the STPA model to evaluate how the system control logic responds. For example, the message broker lag or orchestrator eviction from FMEA become contextual causes in STPA causal scenarios. This step tests whether the system's control structure has sufficient feedback, validation, and degraded mode behaviour to remain safe. The constraints derived from STPA are then formalised into design requirements, orchestration policies, or runtime checks. This closes the loop by feeding back into FHA/FMEA documentation as verified mitigations.

Where the analysis identifies security properties, dependencies, or controls necessary to maintain a safety objective, these should also inform the organisations security risk management activities so that the associated security risks and controls remain aligned with the safety argument. Further guidance on this relationship is provided in Annex A.

This allows safety and cybersecurity engineers to work on the same functional definitions rather than different taxonomies. Every runtime policy or configuration parameter maps back to a specific hazard or constraint.

Mapping Co-Engineering to Cloud Layers

Edge layer

If there is a vehicle in motion, the requisite safety need would be to keep the vehicle safe - even if everything else is on fire. The linking security need would therefore be to reject bad or forced instructions. Thus, the co-engineering rule would be to apply cloud/remote commands only if: the freshness is within the current threshold, the signature is valid, the source is authorised, and the timing budget has not been breached. If these bounds are not satisfied, then the system should stay in a local, conservative mode.

This would be an STPA-Sec constraint that has been derived from an FHA hazard around applying unsafe commands, and FMEA items such as broker lag and IdP outage. It involves both safety and security, do not act on old data and do not act on untrusted data.

To implement this, the cloud system would use controls such as signed containers, a local mission cache, or key storage in the hardware.

Communication layer

The communication system has a safety need of predictable, ordered, bounded latency delivered for control. The security need that is linked to this would be that data delivered on the network is authenticated, the system is replay-protected and has path-controlled delivery. Therefore, the co-engineering rule would be that the control flow should operate

within the Quality of Service (QoS) and performance requirements established by the safety assessment and use mutually authenticated channels; where those requirements cannot be maintained, the system should transition to an assessed degraded or safe operational state.

The co-engineering rationale here would start with a safety argument that states that if the latency is greater than the latency threshold, then enter safe mode; the security argument would be that if the certificate is expired, drop the connection.

To implement this, Software Defined Networks (SDN) intent needs to be defined, define multipath, control the Time to Live (TTL) for a topic, and network Service Level Objective (SLO) monitors tied to degraded-mode triggers at the edge.

Cloud layer

A safety need in the cloud layer would be orchestration, and analytics must not starve, evict, or reorder critical services. The linked security need would be that only verified code, and identities are able to affect critical services. Therefore, the co-engineering rule would be that no change to safety classified services may occur unless the change is policy validated, the identity has been authenticated, and the change preserves minimum capacity and timing SLO.

This was derived from the STPA unsafe action; the orchestrator scales down safety service below minimum, and the FMEA item level failure would be autoscaler misconfiguration and node pressure.

To implement the safety and security need, the system may include admission controllers like Open-Policy-Agent/Gatekeeper, PodDistributionBudgets, resource reservations, signed image, pre-service identities, and audit logs.

Co-engineering as a lifecycle activity

Traditional safety and security assurance activities have often been performed as discrete engineering tasks at defined points within the system lifecycle. In cloud enabled aerospace systems, the dynamic nature of cloud infrastructure means that assurance must continue throughout system operation. Safety and security become integrated lifecycle activities, where the output of each phase provides the assurance evidence and engineering inputs for the next, creating a continuous feedback loop between design, deployment, operation, and reassessment that enable the assurance argument to evolve alongside the operational system.

During the design and analysis phase, the operational function, system architecture, and safety objectives are established. FHA identifies the safety objectives associated with system functions, while FMEA identifies component level failures that could prevent those objectives from being achieved. STPA and STPA-Sec complement these analyses by identifying unsafe control actions, interactions, and security related causal factors that may not be adequately represented through traditional failure-based approaches.

Together, these activities establish the safety and security constraints that the system

architecture must satisfy and define the assurance evidence required to demonstrate that those constraints have been achieved prior to operational deployment.

In the build and deploy phase, the constraints established during design are implemented through cloud architecture, communication infrastructure, edge systems, and supporting operational controls. Technologies such as infrastructure-as-code and policy-as-code may be used to implement safety and security controls consistently, reducing configuration errors and supporting repeatable deployments.

Verification and validation activities provide evidence that the implemented system satisfies the established safety and security constraints before assurance acceptance and entry into operational service. The resulting assurance evidence forms the baseline against which subsequent operational monitoring and future changes can be assessed.

Where resilience and recovery mechanisms contribute to the safety argument, verification and validation should include planned and repeatable testing of representative failure and degradation conditions identified through the safety and security assessment. This may include, where applicable, loss or degradation of cloud services, communication disruption, failure of supporting services such as identity or key management, resource exhaustion, stale or unavailable data, and failure of redundancy, backup or recovery mechanisms. Testing should demonstrate that the established safety objectives are maintained during the failure, transition to degraded operation, and subsequent recovery, rather than demonstrating only the restoration of the affected cloud service.

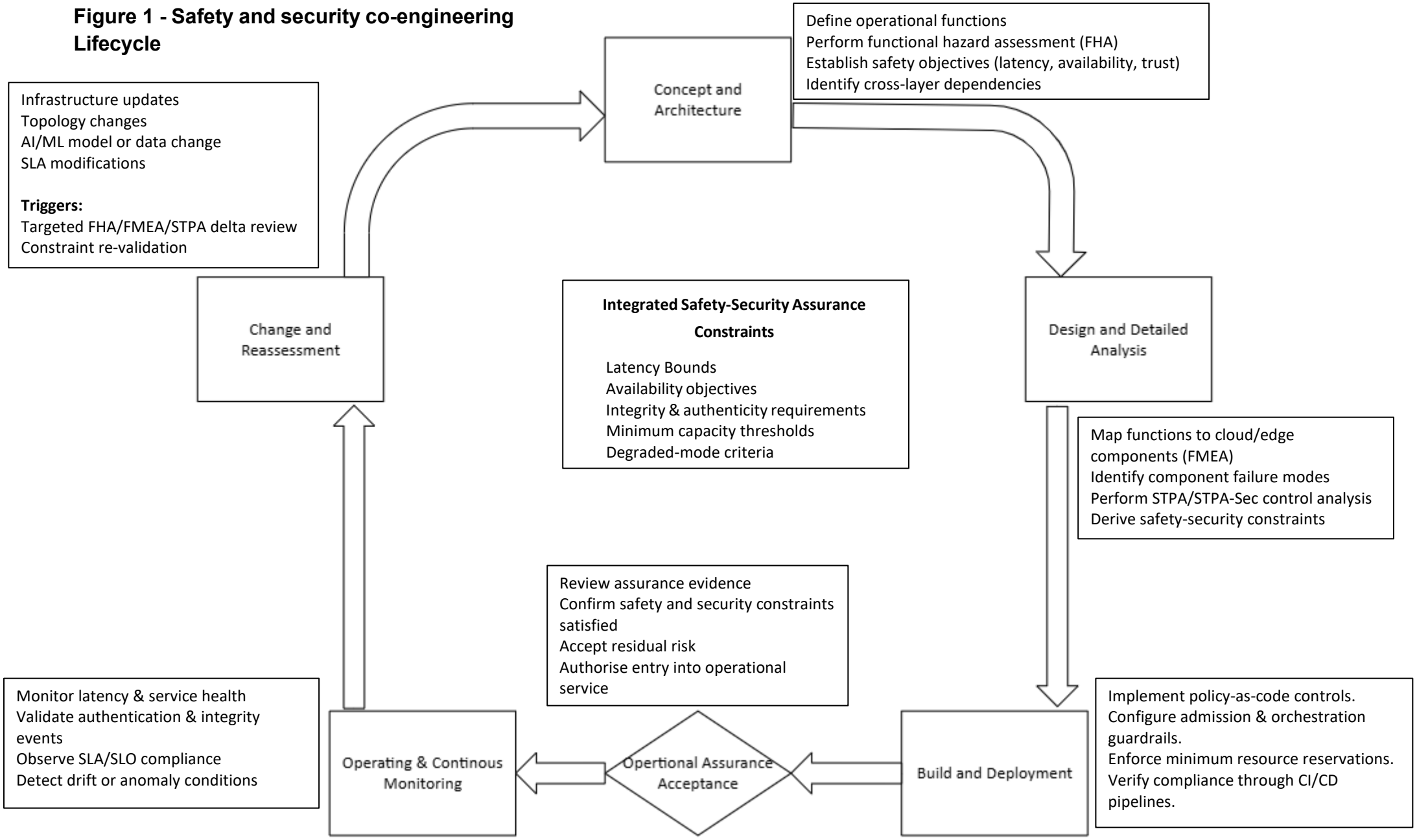
In the operate and monitor phase, continuous monitoring and operational observability provides evidence that the system continues to operate within safety and security constraints established during design. Operational metrics, infrastructure telemetry, security monitoring, and system events support the detection of configuration drift, degraded performance, unexpected system behaviour, and emerging operational risks that could affect the established safety objectives. Continuous monitoring and automated assurance activities should complement, rather than replace established engineering assurance processes. Independent review, periodic assurance activities, verification, validation, and regulatory oversight, where applicable, remain essential to demonstrating the continued acceptability of cloud enabled operational systems. Where continued safety depends upon resilience or recovery mechanisms, these mechanisms should be periodically reverified at an interval proportionate to the operational risk and following significant changes that could affect their effectiveness. Operational experience, incidents, recovery activities, and observed deviations provide valuable inputs into the ongoing assurance of the system. Significant operational events should be investigated to determine whether the existing safety and security arguments remain valid and whether the reassessment activities are required.

Cloud enabled systems evolve continuously through software updates, infrastructure modification, configuration changes, scaling activities, AI/ML model or data updates, changes to supporting services, and operational improvements. Consequently, changes that could affect safety related functions should be assessed to determine whether the

existing safety and security arguments remain valid. The extent of the assessment should be proportionate to the significance and potential safety impact of the proposed change. Automated tools may assist by identifying affected requirements, constraints, controls, or assurance artefacts that require further assessment; however, they should support rather than replace engineering judgement. If necessary, the relevant FHA, FMEA, STPA or STPA-Sec analyses should be updated and supported by appropriate verification, validation, evidence retention, and approval activities before the modified system enters, or continues, operational service.

Throughout the lifecycle, assurance evidence is progressively developed, verified, maintained, and refined to support the continued operation of the system. Operational monitoring, incidents, recovery activities, and reassessment provide continuous feedback into the lifecycle, ensuring that the assurance argument evolves alongside the operational system while maintaining traceability between identified hazards, implemented controls, assurance evidence, operational experience, and subsequent changes.

Figure 1 - Safety and security co-engineering Lifecycle



Co-engineering in Federated and Multi-Cloud Systems

Modern aerospace and UAS rarely operate within a single technical or organisational boundary. Airspace management, cloud coordination, and communication services are increasingly spread across multiple providers, operators, and national infrastructures. In such an environment, the assurance problem extends beyond technology as it becomes an issue of inter-domain trust and accountability.

Integrated safety-security co-engineering approaches this challenge by treating each participating domain whether an operator, a U-space service provider, a telecommunications carrier, or a cloud host as a semi-autonomous control entity with its own hazards, failure modes, and responsibilities. The process begins by performing an FHA within each domain, identifying the functions that directly affect cross domain safety (for instance, time synchronisation, identity management, or telemetry distribution). These analyses are then connected through an inter-domain STPA-Sec exercise that focuses specifically on the trust boundaries and the points where data, commands, or assurances cross from one organisation's control plane into another's.

In federated and multi-cloud environments, organisations may have limited visibility into the infrastructure, operational controls, and supporting services provided by external organisations. Assurance activities should therefore establish sufficient confidence through appropriate assurance evidence, clearly defined interfaces, documented responsibilities, and effective coordination arrangements to demonstrate that externally provided services continue to support the established safety and security objectives. Further guidance on the use of security risk information and assurance evidence across organisational boundaries is provided in Annex A.

This approach acknowledges that no single domain can guarantee end-to-end safety on its own. Instead, safety emerges from the explicit contracts of assurance that exist between domains. For example, a U-space coordination service may trust positional data from an operator's cloud only if it is signed, timestamped, and verified within a known freshness window. Similarly, a ground operator's system may treat all external advisories as low-trust inputs until validated locally.

Through this mechanism, co-engineering creates a federated assurance architecture; each domain enforces local safety and security controls, but shared constraints such as freshness, authenticity, minimum capacity, and bounded latency remain consistent across the system-of-systems. Where resilience or independence is claimed between participating domains or cloud services, organisations should also consider shared dependencies and potential common cause failures that could affect multiple providers or service simultaneously. If any participating domain violates those constraints, others can immediately identify degraded trust and switch to safe local modes.

The result is a system that preserves the continuity of assurance even when no single organisation or infrastructure is in full control. This is particularly relevant for urban BVLOS operations, where command, control, and situational awareness depend simultaneously on mobile networks, cloud services, and shared airspace information exchanges. Where safety related functions depend upon external cloud services, organisations should also consider continuity arrangements appropriate to the

operational system, including service portability, continuity arrangements, or other mitigations that preserve established safety objectives throughout the operational lifecycle. Consideration should also be given to cross border data management and data sovereignty where these could influence the availability, integrity, authenticity, or timely accessibility of safety relevant information. By applying co-engineering across organisational and technical boundaries, aerospace operators can maintain a coherent safety case that is both distributed and resilient; a necessary property for the next generation of cloud-enabled aviation ecosystems.

Annex A – Integration with Security Risk Management

A.1 Purpose

Cloud enabled aerospace systems are subject to security risk management activities undertaken as part of an organisations existing governance, engineering, operational, or regulatory process. These activities may identify threats, vulnerabilities, attack paths, security dependencies, and weakness in technical or organisational controls that could affect cloud hosted or cloud supported operational functions.

The safety and security co-engineering approach described in this CAP does not replace these activities or prescribe a specific security risk assessment methodology. Instead, existing security risk assessment should provide relevant inputs to the safety and security assurance process where security threats, vulnerabilities, or failure of security controls could contribute to hazardous system states or affect the assumption upon which the safety argument depends.

Similarly, where safety analysis identifies security properties or controls necessary to maintain a safety objective, these should inform the security risk management process. The relationship between the two activities should therefore be iterative, ensuring that safety and security assessments remain aligned throughout the operational lifecycle.

A.2 Relationship with established security frameworks

Organisations may already manage cybersecurity and cloud security risk through established standards, frameworks, regulatory requirements, or organisational security management processes. These may include, where applicable, ISO 27001 and associated 2700 series standards, the NIST Cybersecurity Framework, the NCSC Cyber Assessment Framework, and recognised cloud security guidance and assurance frameworks.

This CAP does not replace, duplicate, or prescribe the use of these frameworks. Conformance or certification against a particular cybersecurity or cloud security framework should not, by itself, be considered evidence that the safety implications of cloud related security risks have been adequately addressed.

Instead, relevant outputs and assurance evidence generated through existing security risk management activities should be used as inputs to the safety and security co-engineering approach described in this CAP where identified threats, vulnerabilities, dependencies, or security control failures could affect safety related or safety influencing functions. Conversely, safety analysis may identify security properties, controls, or dependencies whose importance should be reflected within the organisation’s security risk management activities.

Existing security assurance evidence should therefore be reused where it is relevant and sufficient to support the safety and security claims being made. This may include evidence arising from security risk assessments, control assessments, security architecture reviews, vulnerability assessments, security testing, independent assurance activities, supplier assurance, cloud provider assurance, operational

monitoring, or incident investigation. The relevance of such evidence should be determined by the claim it supports and the operational context in which the cloud service is used.

Cloud provider certifications, independent assurance reports, or evidence of conformity with recognised security standards may therefore contribute to the overall assurance argument. However, such evidence should not be assumed to demonstrate that a particular safety objective has been achieved without considering its scope, applicability, limitations, and relationship to the safety-related function or dependency concerned.

A.3 Security risk management as an input to safety and security co-engineering

Security risk management activities may produce information relevant to the FHA, FMEA, STPA and STPA-Sec activities described in this CAP. Relevant inputs may include, where applicable;

- identified threats and credible attack paths affecting safety related or safety influencing functions;
- vulnerabilities affecting cloud services, applications, communications, edge systems, interfaces, or supporting services;
- compromise or loss of identity, authentication, authorisation, certificate, or key-management services;
- risks arising from privileged access, administrative interfaces, APIs, management planes, or orchestration services;
- risks associated with data corruption, manipulation, poisoning, replay, delay, or loss;
- dependencies upon third-party providers, subcontractors, telecommunications, software supply chains, or other externally managed services;
- findings from security testing, vulnerability assessment, penetration testing, incident investigation, or operational monitoring; and
- changes in the threat environment or security posture that could affect assumptions within the existing safety and security argument.

These inputs should be considered where they could initiate, contribute to, or prevent the detection or mitigation of a hazardous system state.

For example, compromise of a cloud identity service may initially be identified through security risk assessment as a threat to authentication or access control. Where the same identity service authorises changes to a safety-related cloud service, compromise of that service may also enable an unsafe configuration or unauthorised control action. The security condition should therefore be considered within the relevant safety and security analysis rather than treated solely as an information-security concern.

This does not require every identified security risk to be incorporated into the safety

analysis. The level of integration should be proportionate to the potential for the security risk to affect safety-related or safety-influencing functions or the assumptions upon which their assurance depends.

A.4 Safety analyses as an input to security risk management

The exchange of information should also operate in the opposite direction.

FHA, FMEA, STPA and STPA-Sec may identify system functions, dependencies, failure conditions, unsafe control actions, or safety constraints whose achievement depends upon particular security properties or controls. Where this occurs, the associated security dependency should be communicated to the security risk management process.

For example, the safety assessment may determine that operational data must remain authentic and within a defined freshness threshold before it can be used by an operational function. Authentication, key management, protection against replay, and integrity verification may consequently become necessary contributors to the safety argument. Failure or compromise of those mechanisms should then be recognised within security risk management in the context of their safety significance.

This relationship enables security controls that contribute to safety to be identified explicitly and provides a basis for determining the assurance, monitoring, change management, and verification activities appropriate to those controls.

Security controls upon which safety objectives depend should not be treated solely as general information-security controls. Their implementation and continued effectiveness should be supported by assurance evidence proportionate to their contribution to the safety argument.

A.5 Traceability and assurance evidence

Organisations should maintain sufficient traceability between relevant security risks, safety hazards, system constraints, implemented controls, and assurance evidence to demonstrate how security conditions that could affect safety have been addressed.

This does not require duplication of existing security documentation. Existing organisational processes and assurance artefacts may continue to be used, provided that the relevant interfaces between safety and security activities are identifiable and the information necessary to support the overall assurance argument can be demonstrated.

Where security controls contribute to a safety objective, the adequacy of the associated assurance evidence should be determined by its relevance to the safety and security claim being made rather than solely by evidence of compliance with a particular security framework or organisational security process.

The level of assurance should be proportionate to the contribution of the security control, service, or dependency to the overall safety argument.

A.6 Lifecycle integration

The relationship between security risk management and safety assurance should continue throughout the operational lifecycle.

Changes to cloud infrastructure, software configuration, interfaces, external dependencies, security controls, AI/ML models, or data, or the other threat environment may alter the security conditions upon which the safety argument depends. Similarly, operational incidents, newly identified vulnerabilities, security monitoring, or changes to provider services may reveal previously unidentified interactions between security risks and safety hazards.

Where such changes could affect established safety objectives or constraints, the relevant safety and security analyses should be reviewed and reassessed proportionately. Conversely, changes to the safety architecture or operational assumptions should be considered within the security risk management where they alter the systems exposure, dependencies, or required security controls.

This lifecycle interaction should support continued confidence that the security assumptions relied upon by the safety argument remain valid as the cloud enabled system and its operational environment evolve.

A.7 Application across organisational boundaries

In federated, multi cloud, or external provided service environments, security risk information and assurance evidence may be distributed across several organisations. Operators may not have direct visibility of all provider infrastructure, vulnerabilities, threat assessment, or internal security controls.

Where externally managed security controls or services contribute to safety objectives, organisations should establish sufficient assurance, through appropriate evidence, defined interfaces, contractual and governance arrangements, monitoring, incident notifications, and change management processes.

The absence of detailed provider information does not remove the need to understand the safety consequences of security failures. Where sufficient assurance cannot be obtained for dependency upon which safety relies, the system architecture or operational arrangements should provide appropriate compensating controls proportionate to the contribution of that dependency to the overall safety argument.