Copyright © SYSGO - Embedding Innovations

SYSGO
EMBEDDING INNOVATIONS



Integrating Security into Safe Airborne Software Development

DO-326B and DO-178C as Two Sides of the Same Coin



SYSGO, 10-2026

Europe’s trusted No1 for safe & secure embedded Systems

- Leading European real-time OS vendor for **Safety- & Security-critical** systems
- Industrial-grade embedded Linux for long-term and secure systems
- Trusted by Aerospace, Defense, Automotive, Industrial, Rail, Medical
- Part of **Thales Group → strategic sovereignty & defense credibility**
- Embedded software experience since 1991
- Certification experts since 1995

PikeOS RTOS & Hypervisor

ELinOS Embedded Linux

PikeOS for MPU


SYSGO Headquarters Germany

Common Criteria

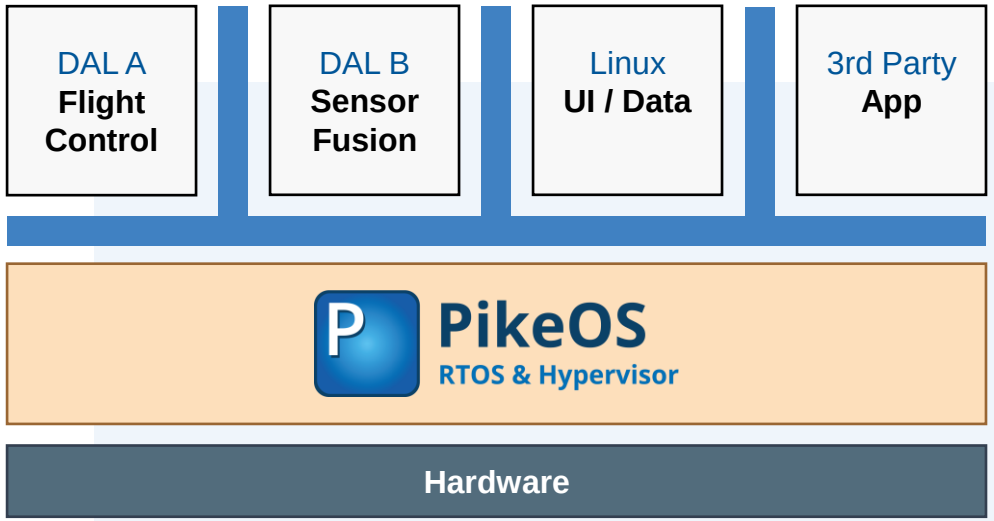
ITAR free

👉 **European sovereignty → ITAR-free, no export restrictions**

👉 **Security and Safety convergence leader**

PikeOS in Partitioned Avionics Architectures





Agenda



1. Airworthiness Development: The Safety Baseline
2. Airworthiness Development: Extending into Security
3. DO-326 / DO-356 vs. Common Criteria
4. Secure Software under Safety Restrictions

Interpretation Note

A practical view

Scope

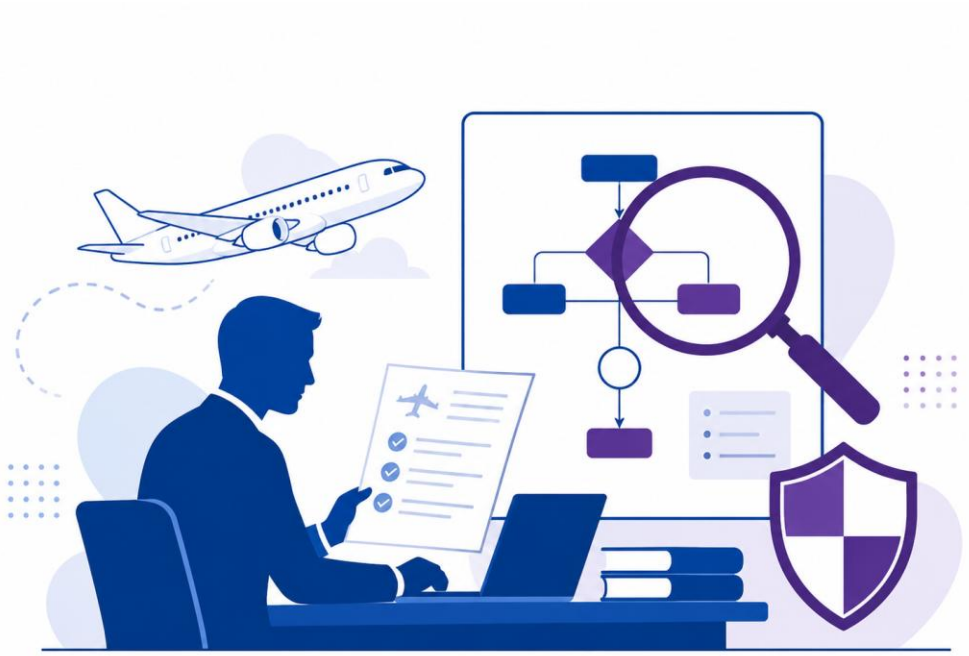
- Simplified, summarized and abstracted representation
- Not exhaustive; elements and relationships may be omitted or generalized

Position

- Content reflects interpretation for illustration purposes
- Non-normative position; no certification guidance

Details

- Some slides intentionally contain detailed material for later study

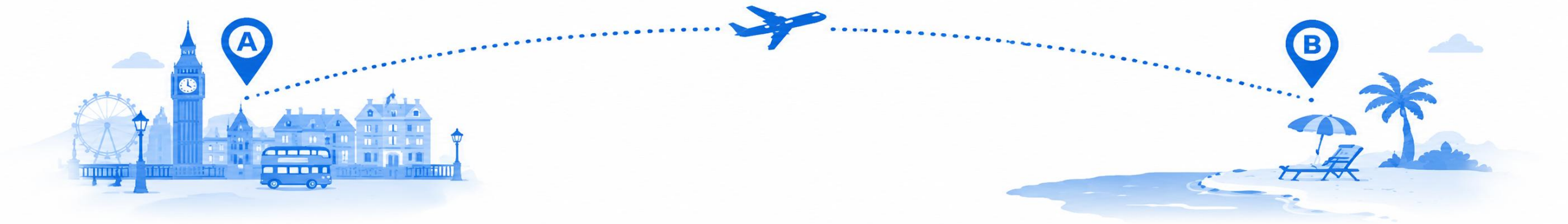




Airworthiness Development:

The Safety Baseline

Safety Objective: Controlling System-related Risk



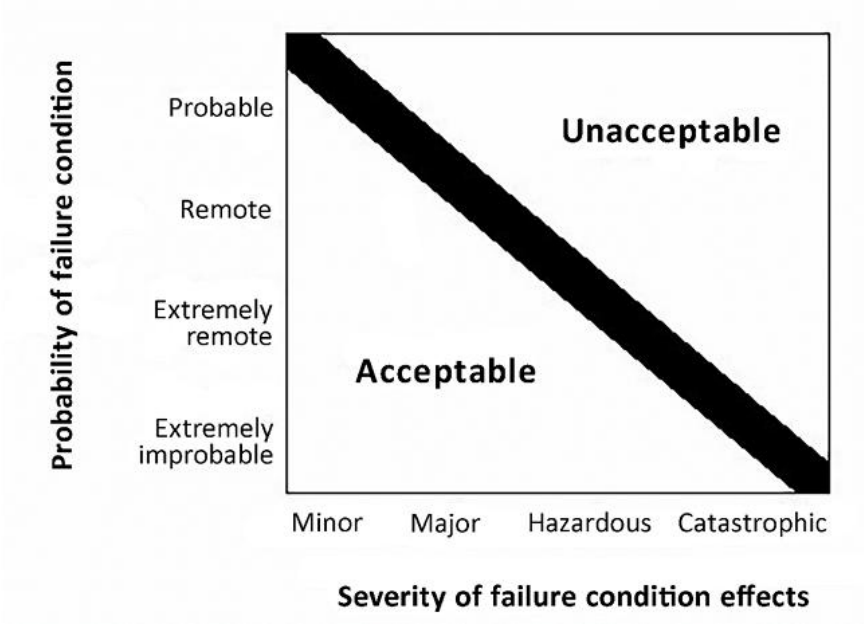
Goal: Safe transport from A to B

Reality:

- 100% failure free is not achievable
- Failures must be anticipated

Objective:

- Control likelihood and severity
- Prevent unsafe conditions



Source:
EASA, CS 25.1309, 2023

Analysis

What are the failure conditions, how likely are they and how severe are the consequences?

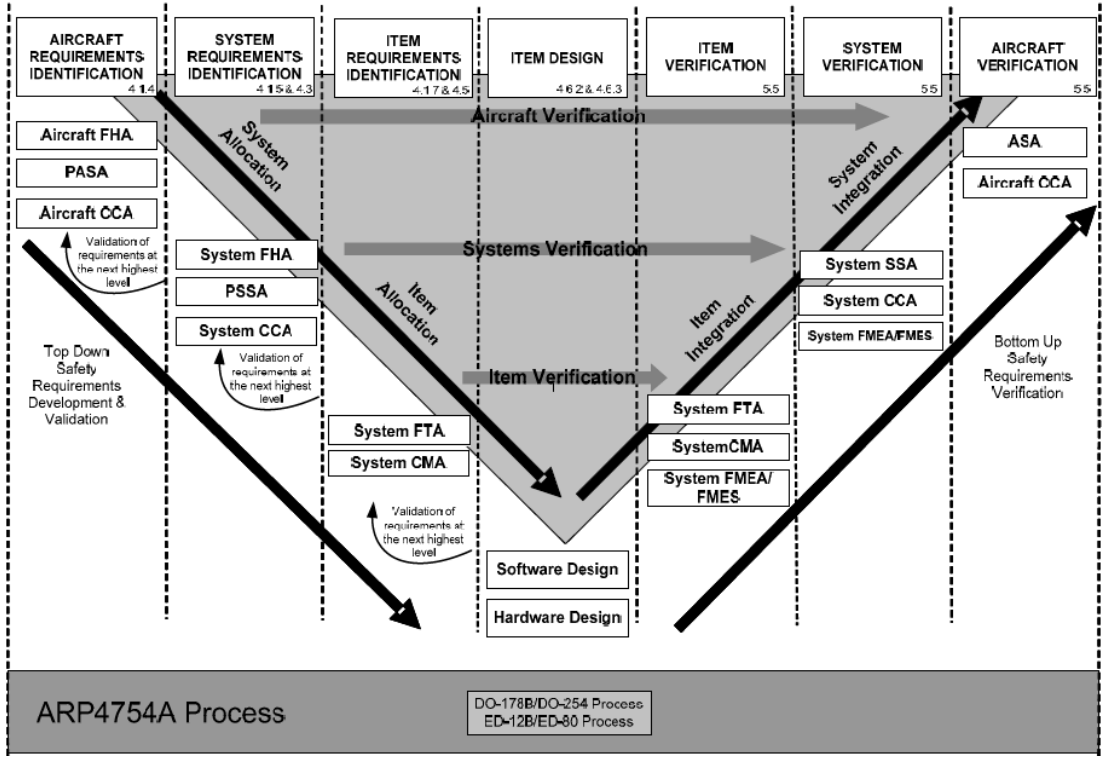
Functional Hazard Assessment (Aircraft level or system level)

- What is the function?
- What are the failure conditions?
- What are the effects?
- How severe are the effects?
- How likely is it / how unlikely must it be?
- What must the design assure?

Because systematic development errors cannot be quantified like random hardware failures, 5 “Development Assurance Levels” (DAL) are defined.

Analysis	What are the failure conditions, how likely are they and how severe are the consequences?
System Definition & Development	How are system requirements defined, allocated and structured satisfying safety objectives?

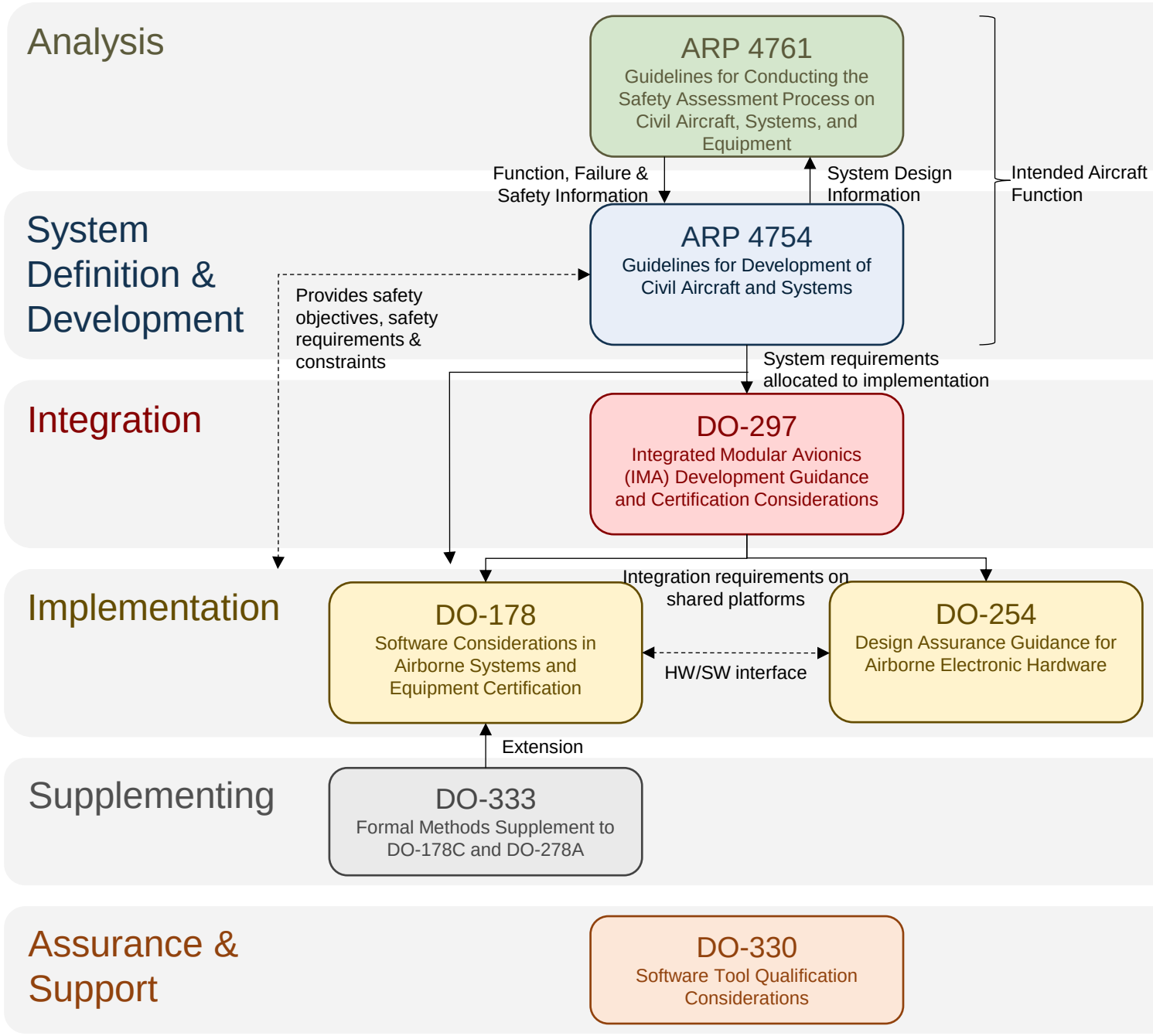
Intended Aircraft Function



Source:
SAE ARP4754A, 2010

Analysis	What are the failure conditions, how likely are they and how severe are the consequences?
System Definition & Development	How are system requirements defined, allocated and structured to satisfy safety objectives?
Integration	How are functions, data and resources integrated and segregated to ensure safety, independence and performance?
Implementation	Was the software / hardware implemented correctly to satisfy the allocated requirements?
Supplementing	What additional methods and guidance are applied to supplement the standards in practice?
Assurance & Support	Are the development process, tools and supporting elements capable and qualified to support certification?

Intended Aircraft Function


Analysis
ARP 4761
Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment
Function, Failure & Safety Information
System Design Information
Intended Aircraft Function
System Definition & Development
ARP 4754
Guidelines for Development of Civil Aircraft and Systems
Provides safety objectives, safety requirements & constraints
System requirements allocated to implementation
Integration
DO-297
Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations
Implementation
Integration requirements on shared platforms
DO-178
Software Considerations in Airborne Systems and Equipment Certification
HW/SW interface
DO-254
Design Assurance Guidance for Airborne Electronic Hardware
Extension
Supplementing
DO-333
Formal Methods Supplement to DO-178C and DO-278A
Assurance & Support
DO-330
Software Tool Qualification Considerations



Airworthiness Development

What was missing?

IUEI as Motivation for DO-326 and DO-356

Existing Safety Process



Focus

Prevent Failures from leading to unsafe aircraft states

- Systematic and random failures
- Safety objectives and DALs
- Requirements, verification, traceability

Intentional **U**nauthorized **E**lectronic **I**nteraction



IUEI

- Adversarial intent
- Connectivity and interface exposure
- Potential safety impact

Airworthiness Security



Focus

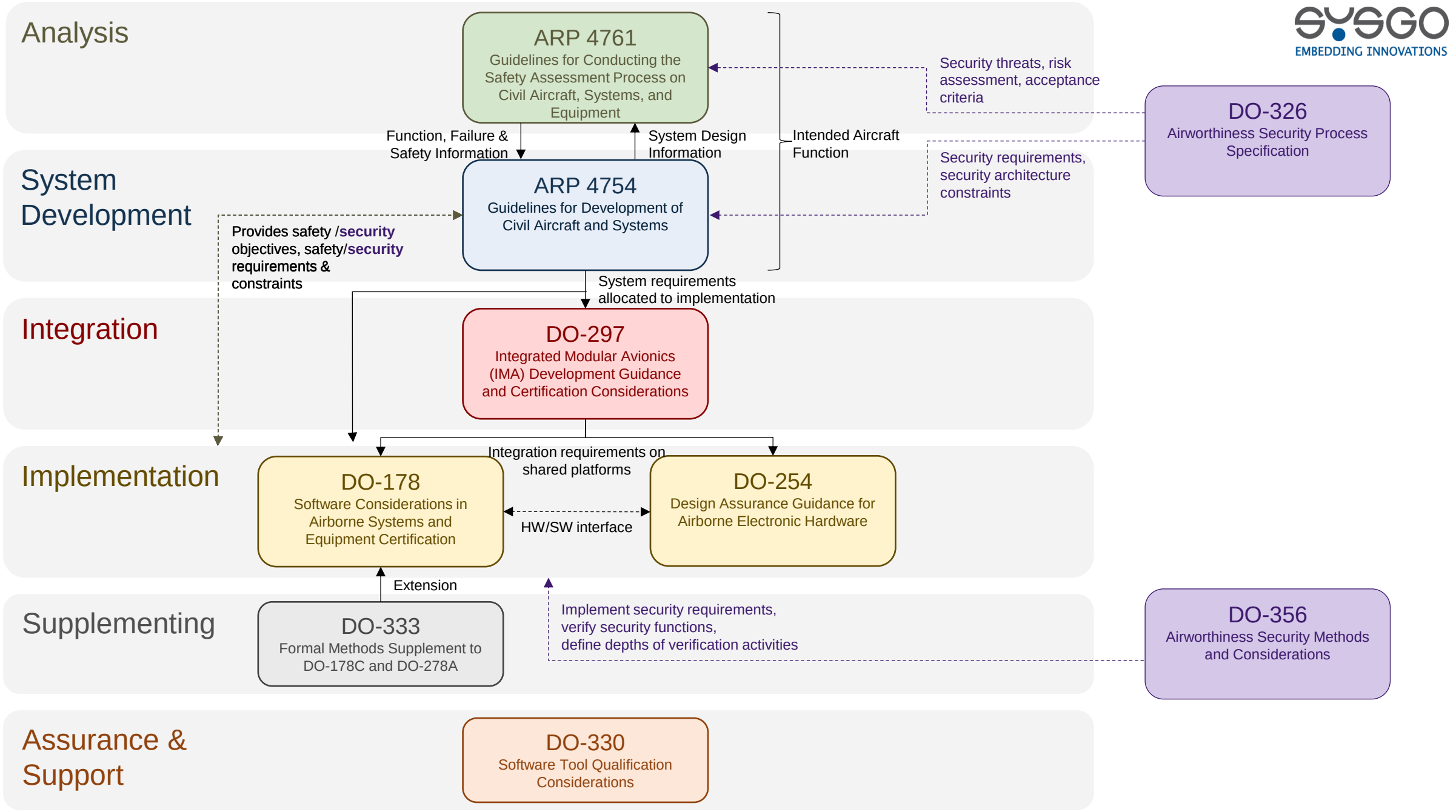
Security for Safety

→DO-326: Airworthiness Security Process Specification

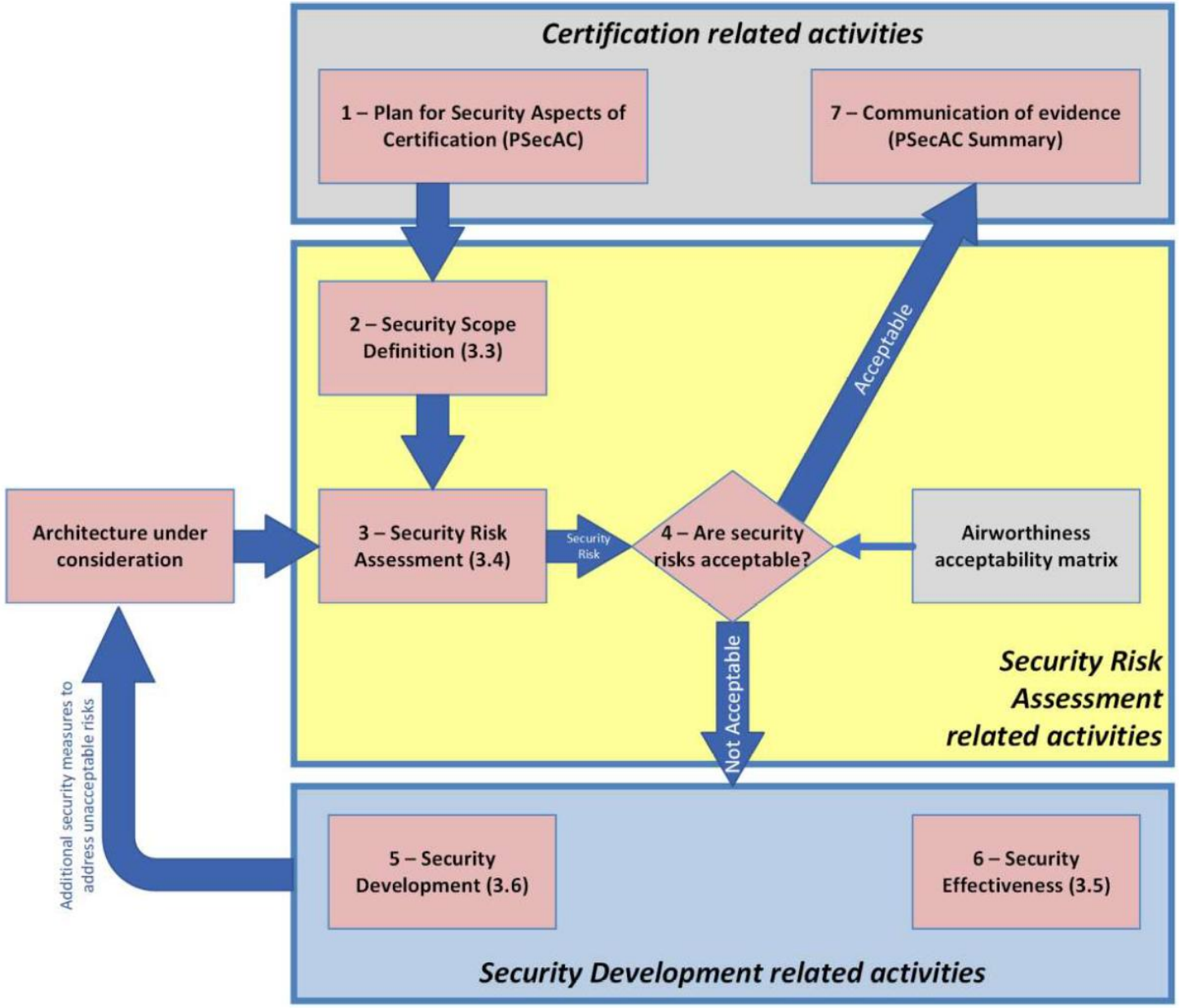
→DO-356: Practical implementation guide

- Threat-driven risk assessment, security requirements, effectiveness assurance
- Extension of the existing process

Analysis	What are the failure conditions, threats and vulnerabilities, how likely are they and how severe are the consequences?	Security Risk Assessment
System Definition & Development	How are system requirements defined, allocated and structured to satisfy safety and security objectives?	Risk-derived requirements
Integration	How are functions, data and resources integrated and segregated to ensure safety, security, independence and performance?	Interface risk analysis
Implementation	Was the software / hardware implemented correctly to satisfy the allocated requirements?	Robustness against misuse
Supplementing	What additional methods and guidance are applied to supplement the standards in practice?	
Assurance & Support	Are the development process, tools and supporting elements capable and qualified to support certification?	Process integrity assurance


Analysis
ARP 4761
Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment
Security threats, risk assessment, acceptance criteria
DO-326
Airworthiness Security Process Specification
Function, Failure & Safety Information
System Design Information
Intended Aircraft Function
System Development
ARP 4754
Guidelines for Development of Civil Aircraft and Systems
Security requirements, security architecture constraints
Provides safety /security objectives, safety/security requirements & constraints
System requirements allocated to implementation
Integration
DO-297
Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations
Integration requirements on shared platforms
Implementation
DO-178
Software Considerations in Airborne Systems and Equipment Certification
HW/SW interface
DO-254
Design Assurance Guidance for Airborne Electronic Hardware
Extension
Supplementing
DO-333
Formal Methods Supplement to DO-178C and DO-278A
Implement security requirements, verify security functions, define depths of verification activities
DO-356
Airworthiness Security Methods and Considerations
Assurance & Support
DO-330
Software Tool Qualification Considerations

Airworthiness Security expands the Safety Process



Sources:
RTCA DO-326B, 2024

Airworthiness Security expands the Safety Process

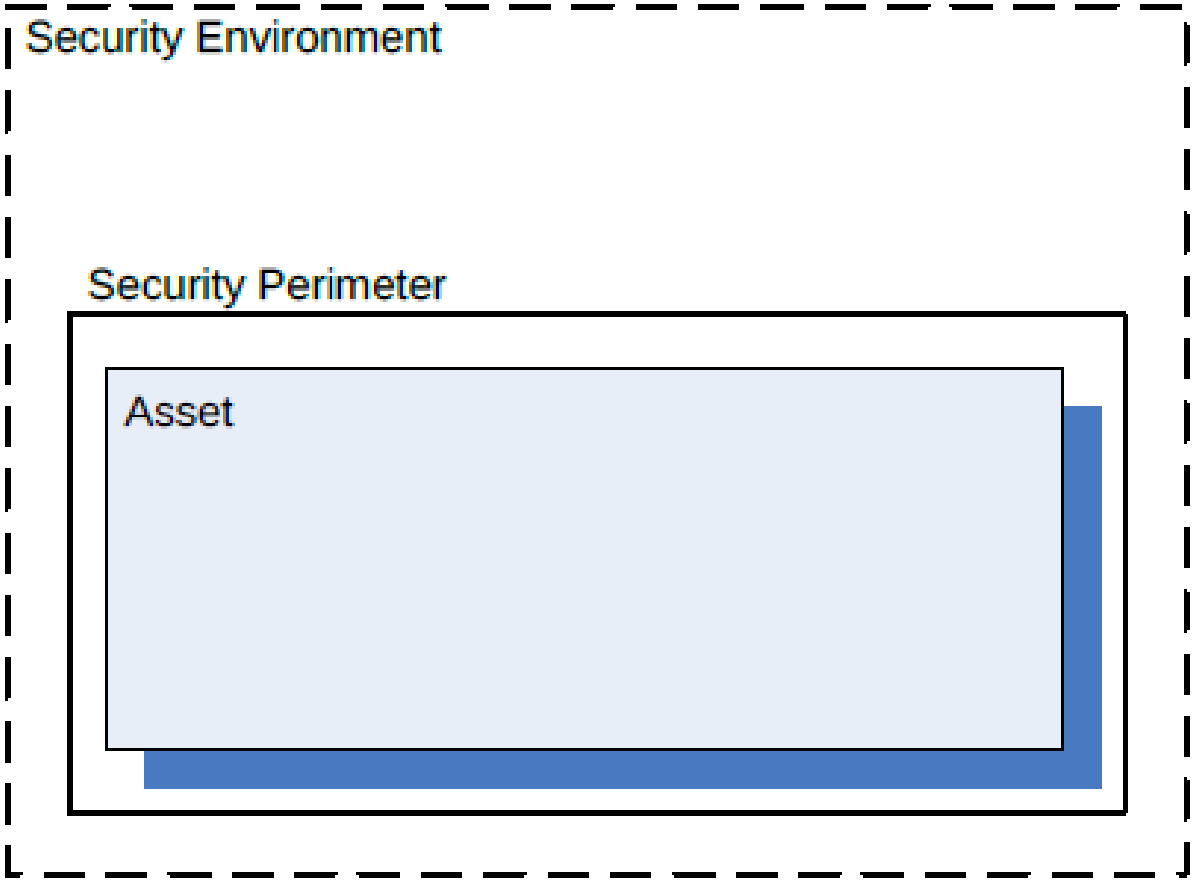
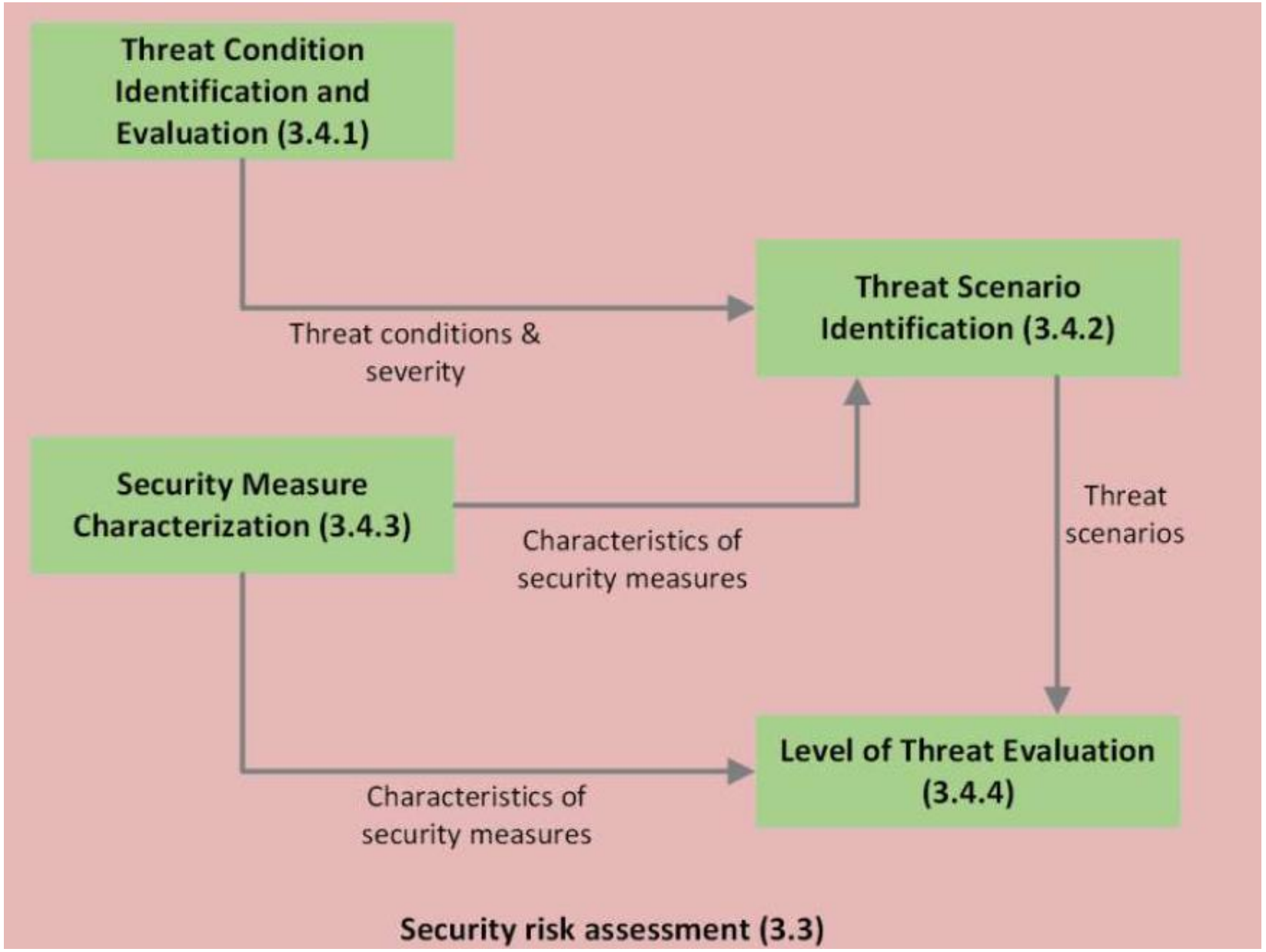


Figure 3-1 : Security Scope

Sources:
RTCA DO-326B, 2024

Airworthiness Security expands the Safety Process



Security Assurance Level depending on Risk Assessment

Sources:
RTCA DO-326B, 2024

Airworthiness Security expands the Safety Process

Table 2-2: Airworthiness Risk Acceptability Matrix

	Severity of the Threat Condition Effect				
Level of Threat	**No Safety Effect**	**Minor**	**Major**	**Hazardous**	**Catastrophic**
Very High	Acceptable	Acceptable	Unacceptable	Unacceptable	Unacceptable
High	Acceptable	Acceptable	Unacceptable	Unacceptable	Unacceptable
Moderate	Acceptable	Acceptable	Acceptable	Unacceptable	Unacceptable
Low	Acceptable	Acceptable	Acceptable	Acceptable	Unacceptable
Extremely Low	Acceptable	Acceptable	Acceptable	Acceptable	Acceptable*

Sources:
RTCA DO-356A, 2018

Airworthiness Security expands the Safety Process

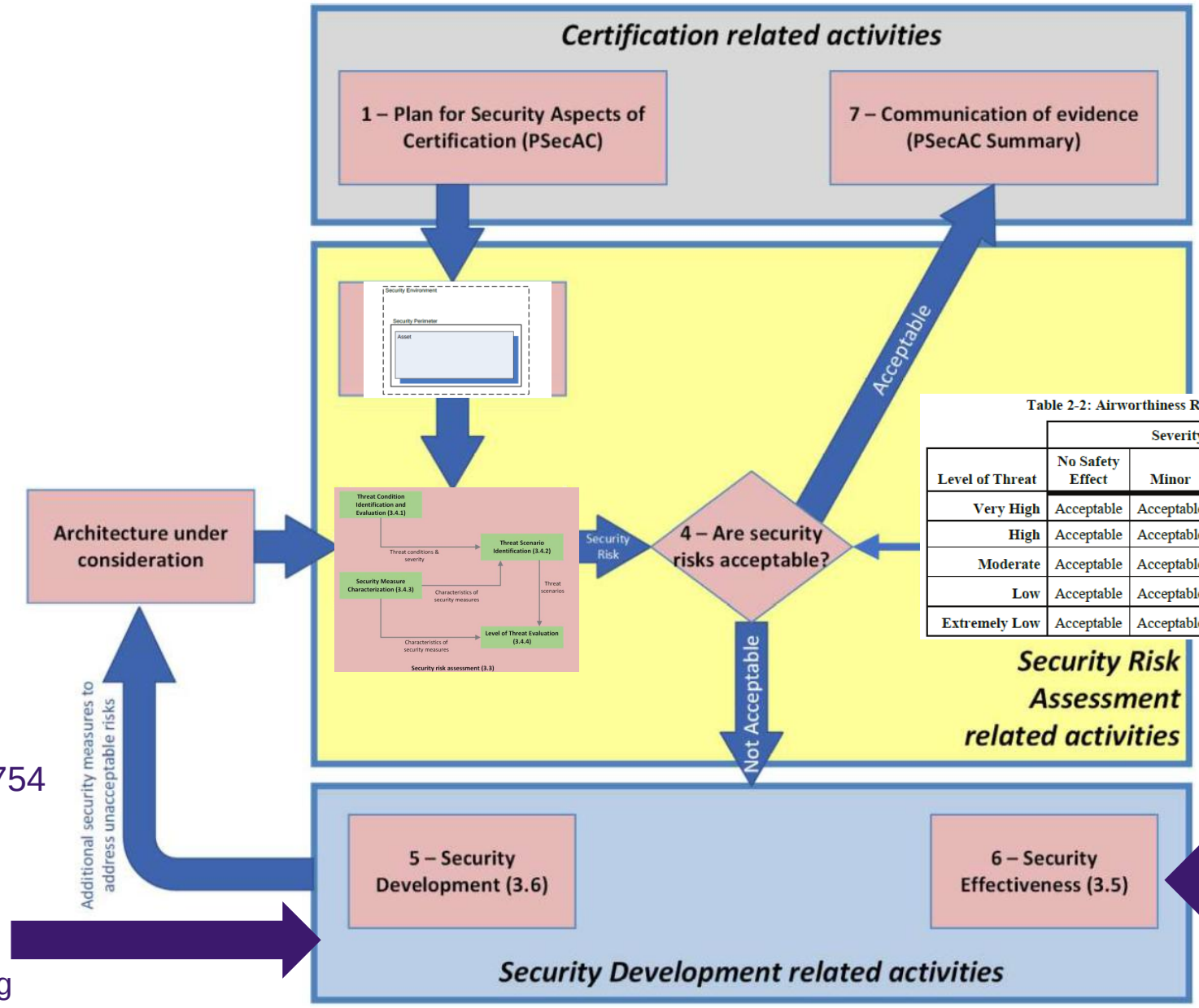


Table 2-2: Airworthiness Risk Acceptability Matrix

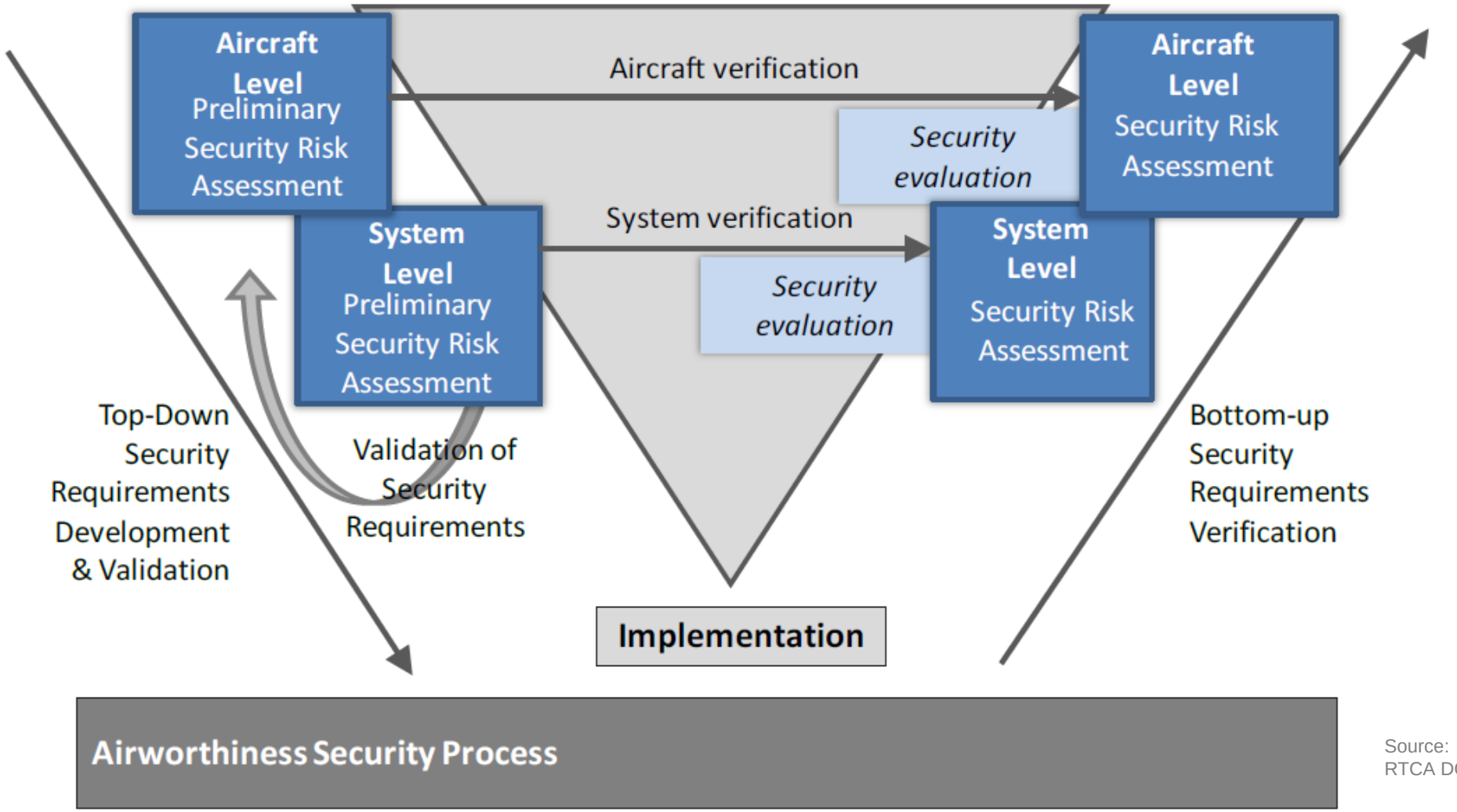
	Severity of the Threat Condition Effect				
Level of Threat	No Safety Effect	Minor	Major	Hazardous	Catastrophic
Very High	Acceptable	Acceptable	Unacceptable	Unacceptable	Unacceptable
High	Acceptable	Acceptable	Unacceptable	Unacceptable	Unacceptable
Moderate	Acceptable	Acceptable	Acceptable	Unacceptable	Unacceptable
Low	Acceptable	Acceptable	Acceptable	Acceptable	Unacceptable
Extremely Low	Acceptable	Acceptable	Acceptable	Acceptable	Acceptable*

Sources:
RTCA DO-326B, 2024
RTCA DO-356A, 2018

All as defined in ARP 4754
+ Security Architecture
+ Security Measures
+ Security Guidance
+ Security Verification
  + Vulnerability Testing

Provides assurance that no known unacceptable vulnerabilities remain

Security Risk Assessment is performed top-down and verified bottom-up across the V-Model



Source:
RTCA DO-326B, 2024



Airworthiness Security vs. Common Criteria

Common Criteria: What Is It?

Purpose

- Goal:
  Comparable Product Security Claims

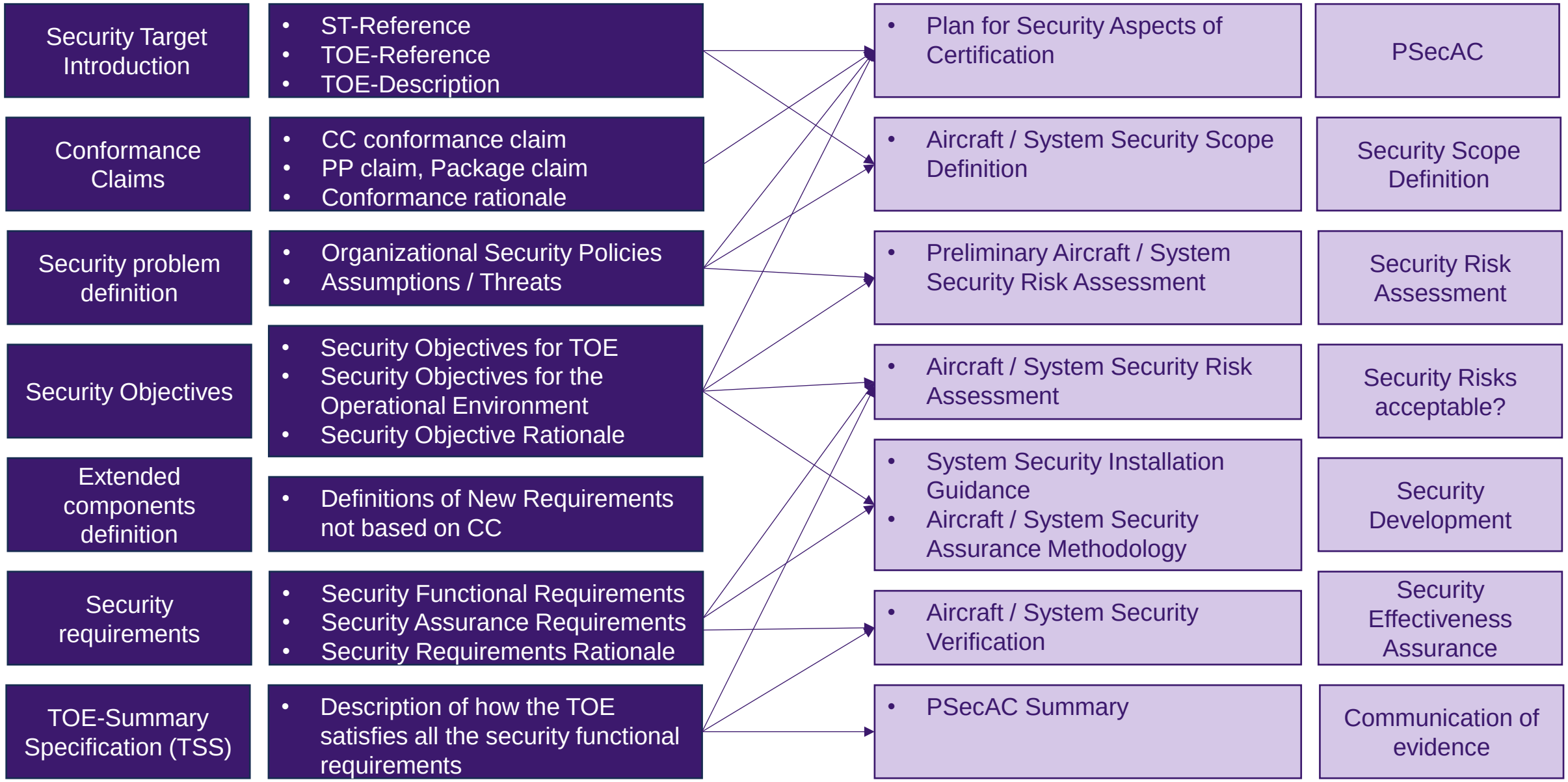
- Universal IT Security Product Standard
  - Not industry-specific
  - Standardized Framework for Specification, Implementation and Independent Evaluation

Components

- Protection Profile (PP)
  - Implementation-independent security requirements for a product class
  - Can be used in normative documents

- Security Target (ST)
  - Product-specific security claims for the Target of Evaluation
  - Most Security Targets are published on commoncriteriaportal.org, e.g. SYSGO's PikeOS
- Target of Evaluation (TOE)
  - Product
- Certificate
  - Confirms TOE satisfies the ST claims

Mapping Common Criteria ST to DO-326 Activities



Copyright © SYSGO | High Integrity Software Conference

Leverage Existing Assurance Evidences for DO-356

SYSGO EMBEDDING INNOVATIONS

DO-178 Evidence

- Traceability
- Verification
- Configuration management
- Tool qualification

CC Evidence

- Security Target
- SFRs / SARs
- TSS



DO-356 Objectives satisfied

- Evidence reused
- Additional actions completed
- Airworthiness-security argument established



Secure Software under Safety Restrictions

Security Mechanisms can conflict with Certification Constraints

Security Expectations

Resist Attacks
Reduce exploitability and increase attacker's cost

Unpredictability
Use randomness, diversity and non-determinism

Detect & Contain
Detect abnormal behavior and limit impact

Evolve & Update
Enable continuous updates based on changing threats

Coexistence of Safety and Security

- Security must not compromise Safety
- Security mechanisms must be engineered as deterministic, analyzable and verifiable behavior
- Complexity affects assurance



Safety Expectations

Determinism
Predictable execution, timing and resource usage

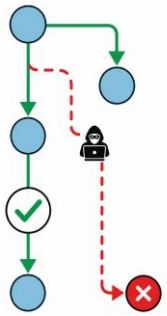
Analyzability
Static behavior, bounded complexity, no hidden state

Reliability
High assurance, freedom from unintended behavior

Configuration Control
Stable, verifiable baseline through the life cycle

Control Flow Integrity (CFI)

Threat and Security Architecture



Threats:

- Manipulate Return Addresses, Function Pointers or vtables

→ Unintended Code Execution

Possible Security Mitigation:

- Hardware-assisted branch-target enforcement, where available (e.g. IBT)
- Build allowed Control Flow Graph at compile time
- Check branch targets at runtime → only approved branch targets are accepted
- Invalid target → fail-secure handling

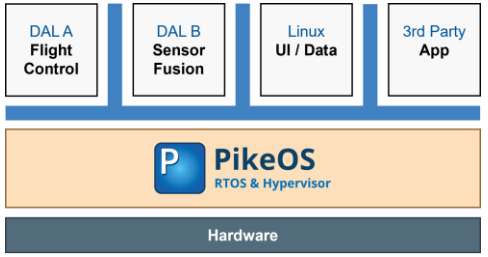
Airworthiness Security RTOS

Possible Safety Conflicts:

- Compiler-inserted checks become part of the certified software behavior, which might not be traceable
- Additional runtime checks affect timing analysis
- CFI metadata and handlers must preserve partition isolation

Other Security Architecture Mitigations:

- Using minimalistic microkernel
  - Deeply analyzed
  - Entering kernel with system call
    → Checking parameters and state



- Using ARINC-653 compliant operating system isolating potential threats to particular application

Stack Canaries

Threat and Security Architecture

- **Threat**:
  - Stack buffer overflow corrupts data
- **Typical Security Architecture:**
  - Guard value is inserted into the stack frame by compiler
  - Guard value checked before function return
  - If value changed, stack corruption detected
  - Entropy source needed to set random guard values

Airworthiness Security RTOS

Possible Safety Conflicts:

- Compiler-inserted checks become part of certified object code
- Additional runtime checks affect timing analysis
- Runtime overhead decreases performance
- Canary value initialization must be deterministic or justified

Other Security Architecture Mitigations:

- Guard pages enabled around stack
  - Memory protection exception → deterministic fail-safe handling
- Safe copy functions with fixed buffer-sizes
  - Buffer sizes do not depend on user-argument

Address Space Layout Randomization (ASLR)

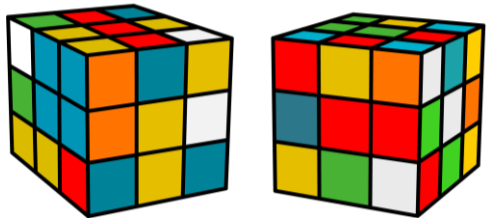
Threat and Security Architecture

Threat:

- Predictable target addresses make exploits possible

Typical Security Architecture:

- Randomize base addresses at load or startup
- Requires relocation support and entropy source



Airworthiness Security RTOS

Possible Safety Conflicts:

- Runtime randomization conflicts with reproducible certification baselines
- Memory layout changes may affect timing, cache and WCET analysis
- Debugging, traceability and failure analysis become complex

Get connected with us!

SYSGO GmbH
Am Pfaffenstein 8
55270 Klein-Winternheim
Germany

+49 6136 99480
info@sysgo.com

Sales Contact
sales@sysgo.com

Follow us, Like and Share



www.sysgo.com/news

www.sysgo.com