Understanding Cybersecurity Interface Agreements

Revision

Version 7
7/15/21 5:00 PM

Author

Charles Wilson

Abstract

This document explains the symbology and intent behind the **AVCDL Cybersecurity Interface Agreement**.

License

Overview

The **AVCDL Cybersecurity Interface Agreement** is intended to establish the following:

- identity of responsible individuals
- understanding of supplier capabilities
- responsibility of both consumer and supplier for the various AVCDL work products
- agree upon the confidentiality level for the various AVCDL work products
- relevant comments of both consumer and supplier

Note: As the **AVCDL Cybersecurity Interface Agreement** is a legal agreement, it is a tracked document.

Note: When completing the **AVCDL Cybersecurity Interface Agreement** replace all **[bracketed red text]** with the appropriate information.

Cover

The cover sheet specifies the supplier and supplied item which this agreement covers.

Cybersecurity Interface Agreement for [supplier company name]

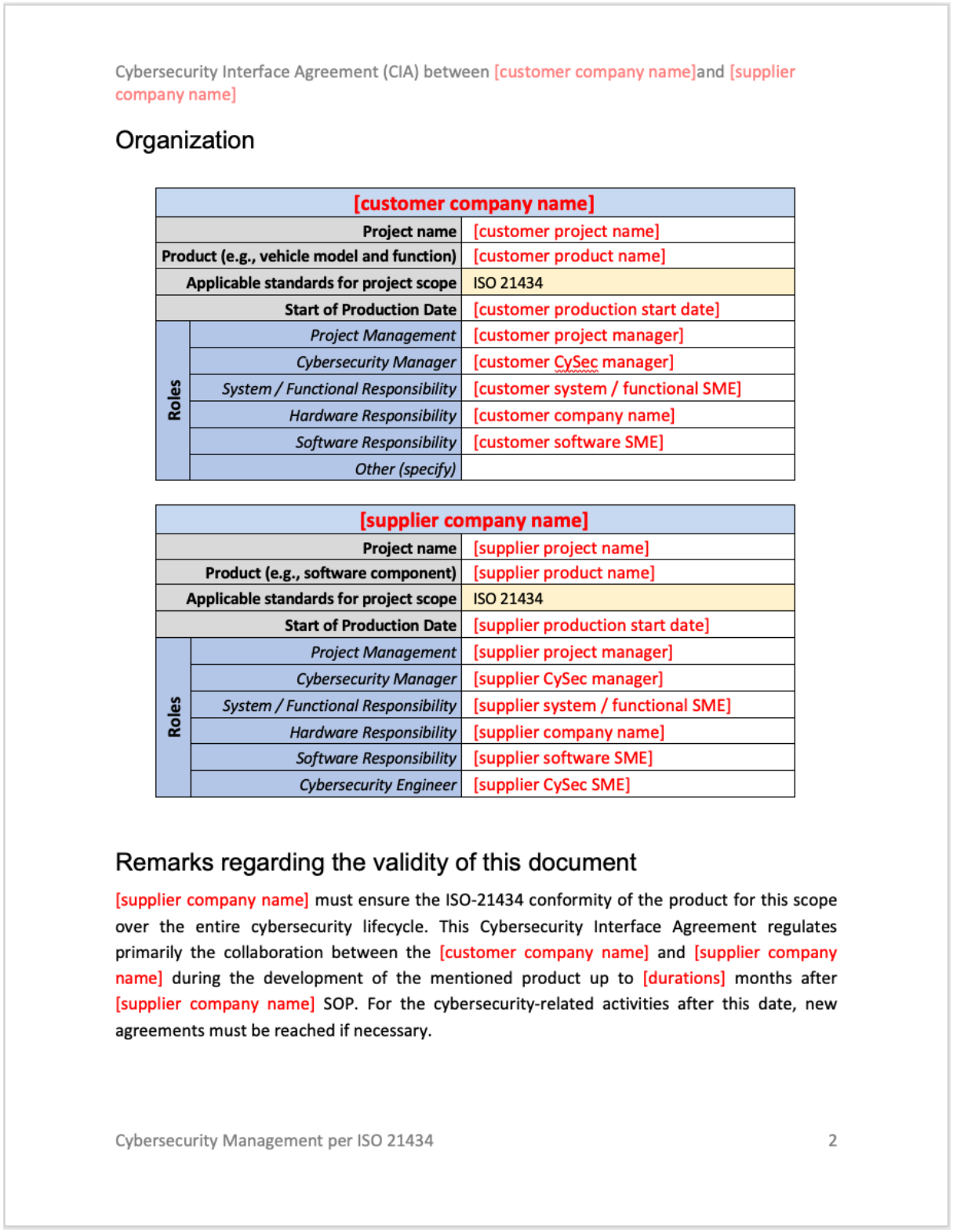
Object This document specifies the cybersecurity-related documents exchanged between [customer company name] and [supplier company name] for the development of [supplier product item].

Synthesis / Action for application

Revision 15

Organization

The **Organization** page contains information defining the scope and responsible parties.



Cybersecurity Interface Agreement (CIA) between [customer company name]and [supplier company name]

Organization

[customer company name]		
	Project name	[customer project name]
	Product (e.g., vehicle model and function)	[customer product name]
	Applicable standards for project scope	ISO 21434
	Start of Production Date	[customer production start date]
Roles	*Project Management*	[customer project manager]
	Cybersecurity Manager	[customer CySec manager]
	System / Functional Responsibility	[customer system / functional SME]
	Hardware Responsibility	[customer company name]
	Software Responsibility	[customer software SME]
	Other (specify)	

[supplier company name]		
	Project name	[supplier project name]
	Product (e.g., software component)	[supplier product name]
	Applicable standards for project scope	ISO 21434
	Start of Production Date	[supplier production start date]
Roles	*Project Management*	[supplier project manager]
	Cybersecurity Manager	[supplier CySec manager]
	System / Functional Responsibility	[supplier system / functional SME]
	Hardware Responsibility	[supplier company name]
	Software Responsibility	[supplier software SME]
	Cybersecurity Engineer	[supplier CySec SME]

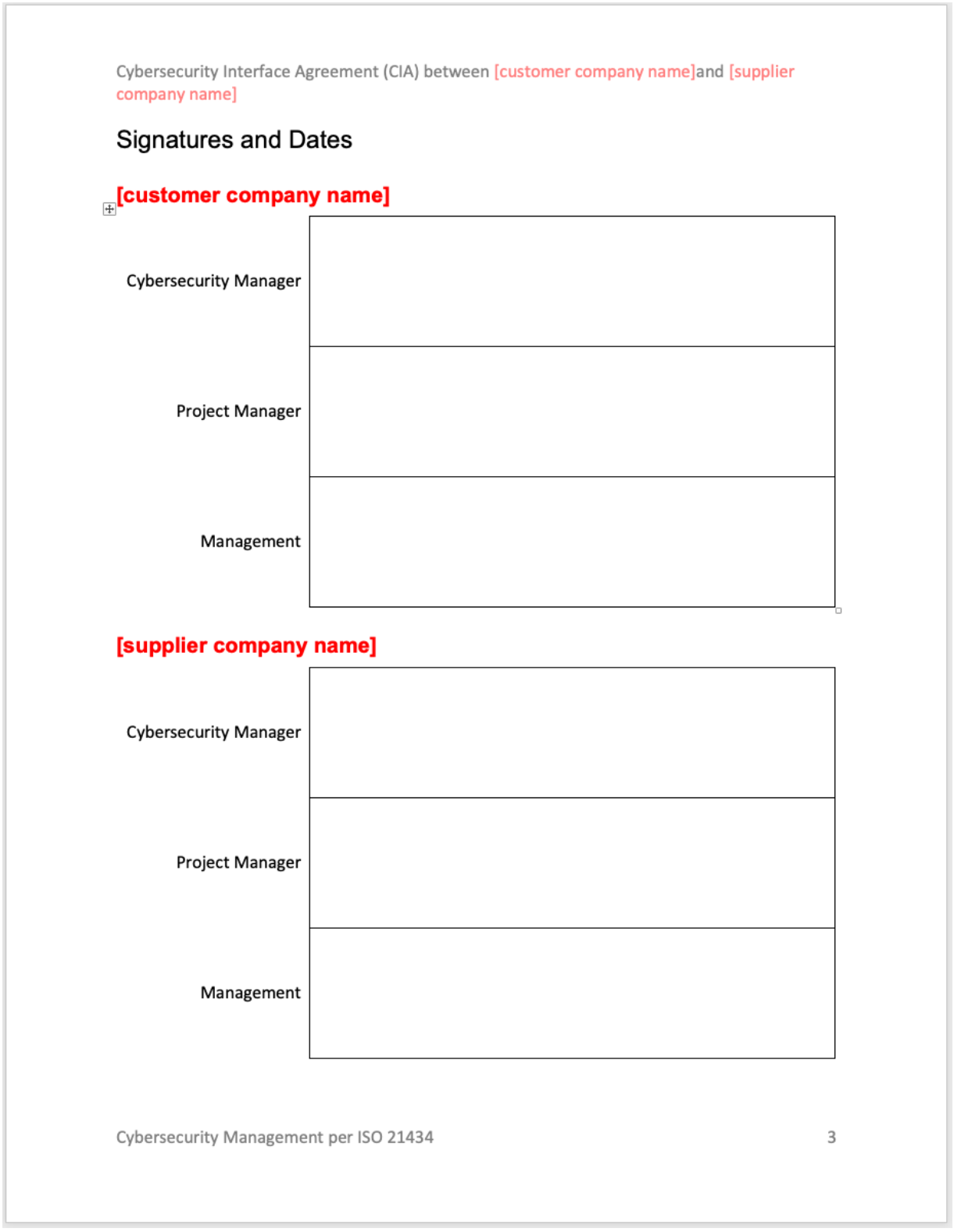
Remarks regarding the validity of this document

[supplier company name] must ensure the ISO-21434 conformity of the product for this scope over the entire cybersecurity lifecycle. This Cybersecurity Interface Agreement regulates primarily the collaboration between the [customer company name] and [supplier company name] during the development of the mentioned product up to [durations] months after [supplier company name] SOP. For the cybersecurity-related activities after this date, new agreements must be reached if necessary.

Cybersecurity Management per ISO 21434 2

Signatures and Dates

The **Signatures and Dates** page contains the signoffs for the document.



Cybersecurity Interface Agreement (CIA) between [customer company name]and [supplier company name]

Signatures and Dates

[customer company name]

Cybersecurity Manager	
Project Manager	
Management	

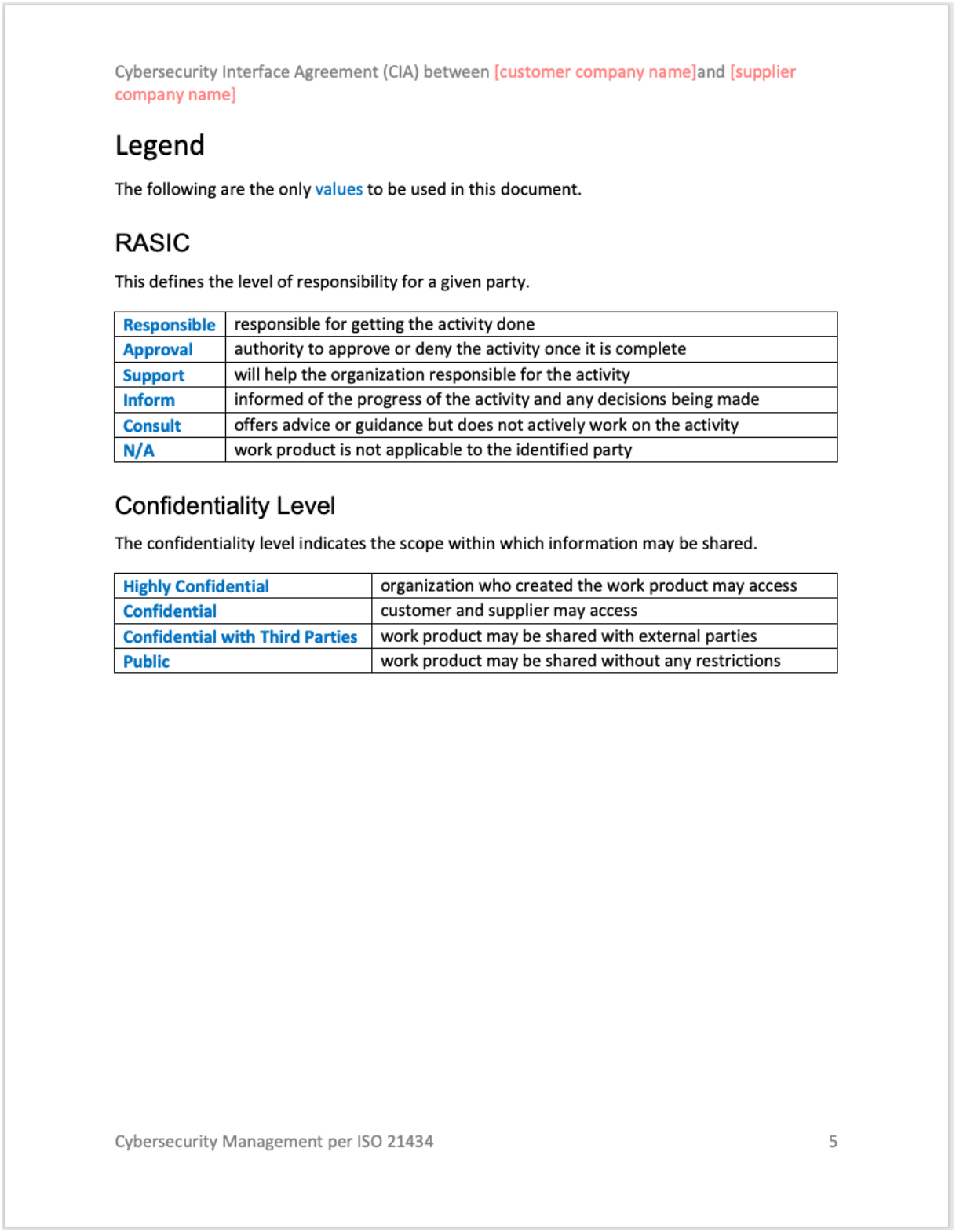
[supplier company name]

Cybersecurity Manager	
Project Manager	
Management	

Cybersecurity Management per ISO 21434 3

Legend

The **Legend** page contains information regarding selections appropriate to the document.



Cybersecurity Interface Agreement (CIA) between [customer company name]and [supplier company name]

Legend

The following are the only values to be used in this document.

RASIC

This defines the level of responsibility for a given party.

Responsible	responsible for getting the activity done
Approval	authority to approve or deny the activity once it is complete
Support	will help the organization responsible for the activity
Inform	informed of the progress of the activity and any decisions being made
Consult	offers advice or guidance but does not actively work on the activity
N/A	work product is not applicable to the identified party

Confidentiality Level

The confidentiality level indicates the scope within which information may be shared.

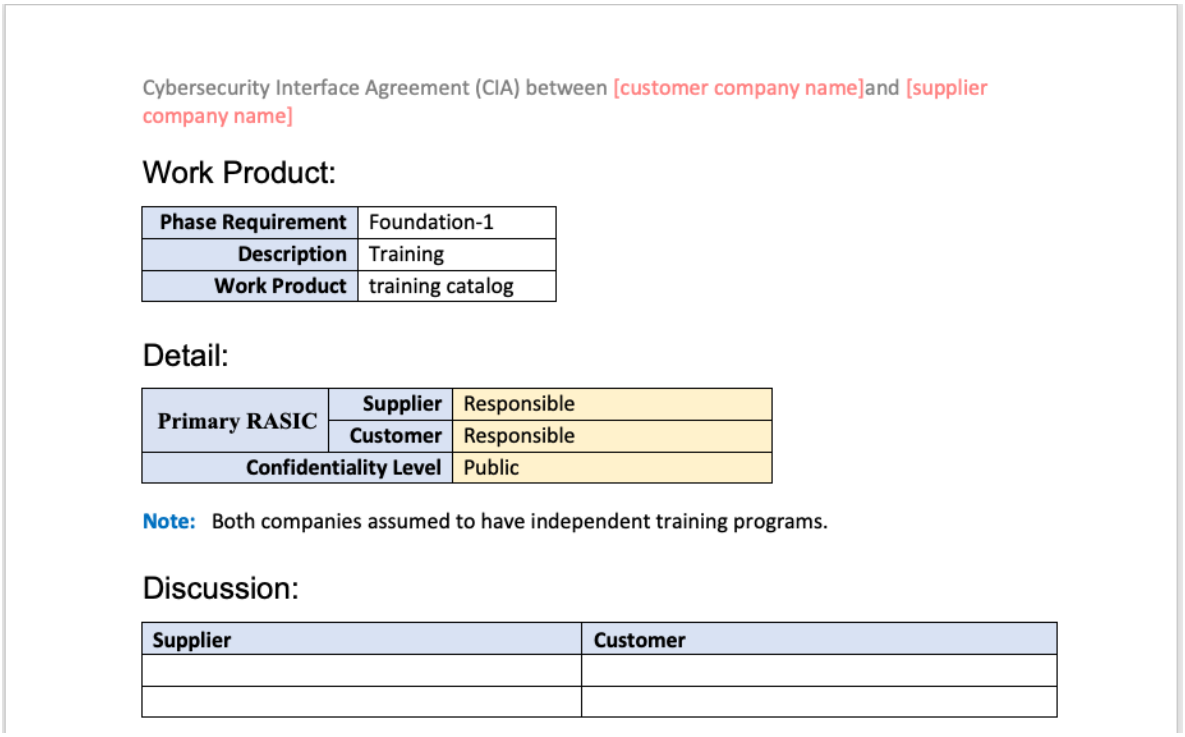
Highly Confidential	organization who created the work product may access
Confidential	customer and supplier may access
Confidential with Third Parties	work product may be shared with external parties
Public	work product may be shared without any restrictions

Cybersecurity Management per ISO 21434 5

Work Product

The **Work Product** pages contain the following (mentioned in the overview):

- responsibility of both consumer and supplier for the AVCDL work product
- confidentiality level for the AVCDL work product
- consumer and supplier comments



Cybersecurity Interface Agreement (CIA) between [customer company name]and [supplier company name]

Work Product:

Phase Requirement	Foundation-1
Description	Training
Work Product	training catalog

Detail:

Primary RASIC	**Supplier**	Responsible
	Customer	Responsible
Confidentiality Level		Public

Note: Both companies assumed to have independent training programs.

Discussion:

Supplier	**Customer**

Note: Highlighted cells indicate default (typical) values.

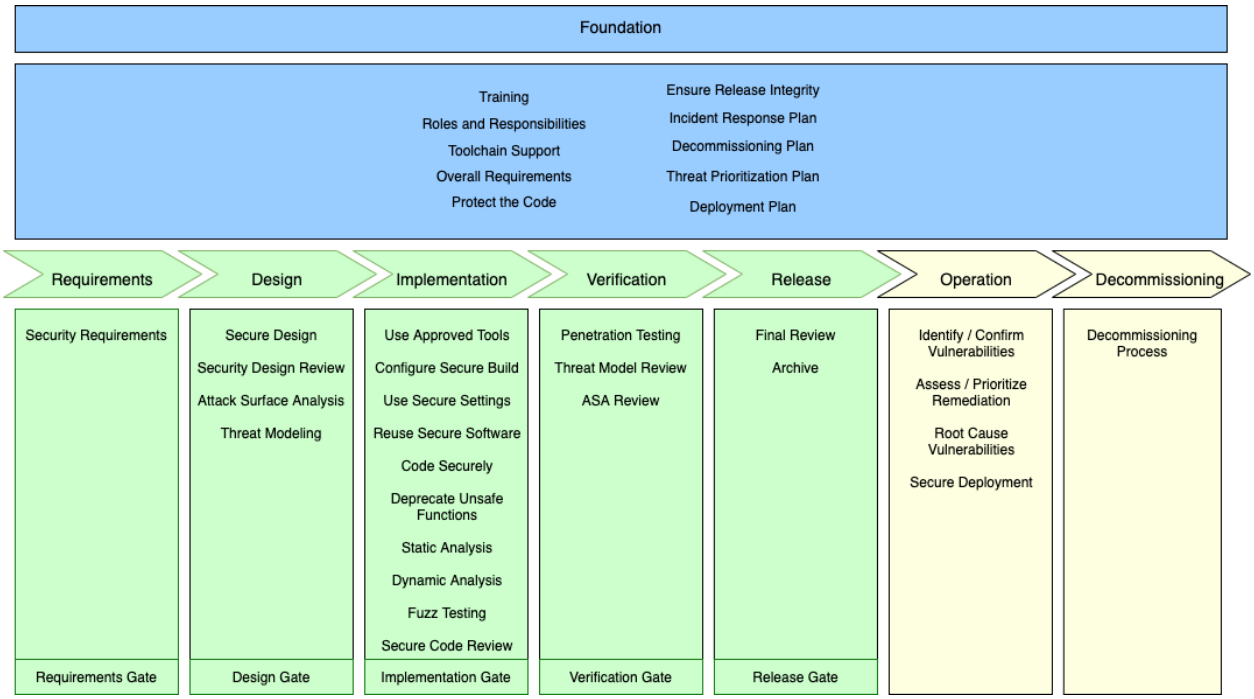
Note: Discussion entries should include a timestamp and be appended (not replaced).

Phase

The phase is the **AVCDL** (Autonomous Vehicle Cybersecurity Development Lifecycle) phase. There are eight phases:

Foundation	Activities performed at various points during the product lifecycle (may apply to multiple elements within the product and may carry over from previous cycles)
Requirements	Evaluation of functional requirements and tailoring of global cybersecurity requirements
Design	Evaluation of the design with respect to cybersecurity and review of design for compliance with cybersecurity requirements
Implementation	Implementation of the cybersecurity aspects of the product
Verification	Verification of the cybersecurity aspects of the product
Release	Validation, final review, archive and preparation for release of the product
Operation	Secure deployment and incident handling
Decommissioning	RMA and end-of-support/life handling

This can be visualized as follows:



Phase Requirement

The phase requirement is the specific requirement within a given AVCDL phase.

Description

This is the title of the phase requirement.

Work Product

This is a set of expected work products created to satisfy a given phase requirement.

Supplier

This is the primary RASIC value for the supplier.

Note: If this is changed from a default (shaded) value, the shading should be removed.

Customer

This is the primary RASIC value for the consumer.

Note: If this is changed from a default (shaded) value, the shading should be removed.

Confidentiality Level

This is the agreed upon level of confidentiality for material relating to a specific phase requirement's work product.

Note: If this is changed from a default (shaded) value, the shading should be removed.

Notes

These are notes about the work product and **should not be modified**.

Discussion

These are comments / questions from the supplier and consumer.

Phase Requirements and Work Products

This section will provide a description of the AVCDL phase requirements and their associated work products.

[Foundation-1] Training

This training ensures that the cybersecurity lifecycle and its requirements are understood by those interacting with it.

Training Catalog

The training catalog provides information as to the cybersecurity offerings available to ensure that those involved with cybersecurity functions are properly trained. The offerings should cover all functions including those performed by management, devops, development, and cybersecurity.

System to Track Training Participation

The training participation tracking system provides for audit and progress information regarding cybersecurity training. It is presumed that this will be supported by an existing company system.

[Foundation-2] Roles and Responsibilities

It is critical to the success of any cybersecurity relevant project that the roles and responsibilities be defined and assigned prior to the phase to which they apply. These individuals serve as gatekeepers of security issues at the various phase gates.

Roles and Requirements Document

The roles and responsibilities document identifies the responsible parties and technical roles required to carry out all identified cybersecurity functions. Description of the roles can be sourced from the **NIST NCWF** [5].

[Foundation-3] Toolchain Support

Software supporting secure development must be evaluated, installed, and trained for. Having this information is key in supporting later phase requirements.

List of Approved Tools and Components

The list of approved tools and components is best implemented as a database. Items include applications used to create the product (compilers, build system, analysis tools, …) and components sourced externally (open source and third-party libraries). Use of a database allows for better data extraction in later phase requirement operations. The recommended metadata formats for these are **SWID** [3, 4] and **SPDX** [2].

[Foundation-4] Definition of Security Requirements

Before designing a secure system, it is necessary to have a clear and coherent set of security requirements. These are the global security requirements as opposed to the more fine-grained requirements called out during the requirements phase.

Global Security Goals

The global security goals are a high-level set of objectives allowing for implementation of the company's security policy. An example basis would be the **UNECE WP.29** [1].

Global Security Requirements

The global security requirements are a reusable set of [non-functional] cybersecurity requirements which may be applicable to any product.

[Foundation-5] Protect the Code

The code storage and access should be set up in such a way as to prevent inadvertent or intentional unauthorized changes, inappropriate access, or theft.

Code Protection Plan

The code protection plan should describe how the following practices are supported:

- Secure code repository
- Code signing
- Software Bill of Materials (**SBOM**) [8]
- Source code archive

[Foundation-6] Ensure Release Integrity

To some extent, this could be considered part of toolchain support. Operations such as code signing, and root-of-trust fall into this process element.

Release Integrity Plan

The release integrity plan should describe how the following practices are supported:

- Code signing
- Hash tracking
- Credential management
- Root-of-trust implementation
- Secure deployment support
- Unauthorized alteration countermeasures

[Foundation-7] Incident Response Plan

The incident response plan covers the mechanisms needed for dealing with both internal and externally discovered security issues.

Incident Response Plan

The incident response plan details how the following activities are performed:

- Incident monitoring
- Vulnerability confirmation
- Assessment and prioritization
- External notification
- Root causing
- External notification updating
- Threat remediation
- Threat remediation notification

Continuous Monitoring Plan

The continuous monitoring plan details the mechanisms implemented to gather and process cybersecurity information from external sources.

[Foundation-8] Decommissioning Plan

A framework must be in place for the eventual removal from service of an in-use system. This should cover the proper handling for any sensitive data embodied in the system.

Decommissioning Plan

The decommissioning plan needs to address the cybersecurity issues related to the removal from service of the product. This may be for repair (RMA) or end-of-life. Elements of cybersecurity with need to be addressed include:

- Credentials
- Certificates
- PII
- Logs
- Location history
- Previous destinations
- Call history
- Stored contacts

[Foundation-9] Threat Prioritization Plan

A mechanism for quantifying potential risks and prioritizing their disposition must be established. This may take the form ranging from a gross-level quantization (bug bar) to a formal methodology (TARA).

Threat Prioritization Plan

The threat prioritization plan details the process of establishing the impact of a cybersecurity threat and determining its disposition.

[Foundation-10] Deployment Plan

A framework must be in place for the loading of software onto the system. This should include both the initial loading / configuration and updating. This should cover the proper handling for any sensitive data to be embodied in the system.

Deployment Plan

The deployment plan details how the following activities are performed:

- Secure software deployment
- Initial provisioning and update scenarios
- Deployment failure handling

Handling of the following sensitive material should be addressed:

- Credentials
- Certificates
- PII
- logs

[Requirements-1] Security Requirements Definition

The requirements are created with consideration of the global security requirements. They provide constraints specific to the work under consideration.

Product-level Security Goals

The product-level security goals are a refinement of the global-level security goals constrained by the needs of the product.

Product-level Security Requirements

The product-level security requirements are created as a tailoring of the global-level security requirements based on the product's functional requirements. These non-functional requirements will be applied to the functional ones during the design phase.

[Requirements-2] Requirements Gate

Requirements phase exit is conditional (formally gated) on completion of all AVCDL phase requirements and work products for this phase.

Formal Gate Signoff

The formal gate signoff details and documents how the phase's work products have been completed.

Phase Product Linking

When the product is being created to a particular certification, the work products on the phase need to be mapped to the requirements of that certification.

[Design-1] Apply Security Requirements and Risk Information to Design

The design should take into consideration established security requirements and risk information.

Design Showing Security Considerations

The design showing security considerations is the process of analyzing the product element's design with respect to cybersecurity. This results in updates to the design and identification of applicable cybersecurity requirements. The applicable requirements are attached to the element's functional requirements as dependencies.

[Design-2] Security Design Review

Help ensure the software will meet the security requirements and satisfactorily address the identified risk information.

Security Design Review Report

The security design review report captures the security deficiencies discovered during the security design review process. This report is used to ensure that those discoveries are properly disposed.

This may be done either by alteration of the design or application of security requirement dependencies.

[Design-3] Attack Surface Reduction

Attack surface analysis guides the disabling or access restricting of system services based on an analysis of the APIs presented by the operation environment (operation system). It applies the principles of least privilege and layered defense.

Attack Surface Analysis Report

The attack surface analysis report captures the security deficiencies discovered during the attack surface analysis process. This report is used to ensure that those discoveries are properly disposed. This may be done either by alteration of the design or application of security requirement dependencies.

[Design-4] Threat Modeling

Threat modeling realizes an abstraction of the system as a set of interacting processes managing resources passing data between them. It is on these data flows that automated threat modeling tools reason.

Threat Modeling Report

The threat modeling report captures the security deficiencies discovered during the threat modeling process. This report is used to ensure that those discoveries are properly disposed. This may be done either by alteration of the design or application of security requirement dependencies.

Ranked / Risked Threat Report

The ranked / risked threat report summarizes the analysis performed on the output of the threat modeling activity which provides input into the threat candidate slicing activity. As indicated by the title, there are two discrete operations which take place. The first is the ranking of threat candidates to determine their relative severity. The second is the risking of threats to determine their potential harm.

Threat Report

The information provided by the ranked / risked threat report is taken and a threshold is established and used to decide whether any given candidate is considered controlled or uncontrolled. Controlled candidates are dismissed (with appropriate justification) and uncontrolled candidates are considered accepted threats. The threat report provides a summary of these considerations.

[Design-5] Design Gate

Design phase exit is conditional (formally gated) on completion of all AVCDL phase requirements and work products for this phase.

Formal Gate Signoff

The formal gate signoff details and documents how the phase's work products have been completed.

Phase Product Linking

When the product is being created to a particular certification, the work products on the phase need to be mapped to the requirements of that certification.

[Implementation-1] Use Approved Tools

Development teams should strive to use the latest approved version of approved tools and components to take advantage of new security analysis functionality and protections.

List of Tools and Components Used

In order to ensure the security of the software to be produced, it is necessary to validate and enumerate the software used in its creation. The process of selection and validation of the tools and components used is documented in the **List of Approved Tools and Components** secondary document created in the foundation phase. During the actual production of the software, it is necessary to verify that the tools and components being used are the ones approved during that process.

[Implementation-2] Configure Build Process to Improve Security

Decrease the number of security vulnerabilities in the software and reduce costs by eliminating vulnerabilities before testing occurs.

Build Process Documentation

Compilers and other tools provide pre-runtime security checking. In order to establish the level of security provided, all security-related tool settings must be enumerated, and their values justified.

[Implementation-3] Use Secure Settings by Default

Help improve the security of the software at installation time, which reduces the likelihood of the software being deployed with weak security settings that would put it at greater risk of compromise.

Secure Settings Document

The most secure application can be compromised because of poorly chosen configuration settings. Such configuration errors may be in access control, database management, cryptographic material handling, or communication. All must be properly chosen, documented, and justified to ensure the greatest level of security appropriate to the task.

[Implementation-4] Reuse Well-Secured Software

Reuse of well-secured (verified) software lowers the costs of development, expedites development, and decreases the likelihood of introducing additional security vulnerabilities.

Component / Version – Product / Version Cross-reference Document

Security-qualified component reuse lowers the costs of software development, expedites software development, and decreases the likelihood of introducing additional security vulnerabilities into the software. This is particularly true for software that implements security functionality, such as cryptographic modules and protocols. By having a component / version – product / version cross-reference, we will be able to both quickly select components for reuse and quickly determine products impacted by issues discovered in these components.

[Implementation-5] Code Securely

Decrease the number of security vulnerabilities in the software and reduce costs by eliminating vulnerabilities during source code creation.

Secure Development

Creating secure code requires the application of multiple techniques. These include applying:

- secure coding best practices
- language-specific best practices
- industry-specific security rules (MISRA / AUTOSAR)
- tool-specific security checkers

[Implementation-6] Deprecate Unsafe Functions

Project teams should analyze all functions and APIs that will be used in conjunction with a software development project and prohibit those that are determined to be unsafe.

Currently Used Deprecated Functions Document

Many commonly used functions and APIs are not secure in the face of the current threat environment. As these present an ongoing threat to security, they should be replaced with secure alternatives where possible. In order to be able to assess the overall risk presented by their continued use, they should be documented, and their presence justified.

[Implementation-7] Static Analysis

Project teams should perform static analysis of source code applying cybersecurity checkers.

Static Analysis Report

Alongside compiler feedback, static code analysis is the fastest turnaround feedback available to developers as to possible security issues within their code. From a traceability perspective, the static analysis report provides not only immediate issues, but also information as to the overall state of cybersecurity culture and can be used as an indicator as to what additional training is needed.

[Implementation-8] Dynamic Program Analysis

Run-time verification of the cybersecurity aspects of software programs is necessary to ensure that a program's functionality works as designed.

Dynamic Analysis Report

Although the quality of feedback from the compiler and static analysis tools have become much better over time regarding security-related issues, there are many situations which are not considered by them. This is where dynamic analysis comes in. Whether code coverage or stack patterns, data gathered during this activity is very helpful in identifying security-related issues.

[Implementation-9] Security Code Review

The security team and security advisors should augment static analysis with other automated or human review as appropriate.

Secure Code Review Summary

Although the quality of feedback from the compiler and static analysis tools have become much better over time regarding security-related issues, there are many situations which are not considered by them. Secure code reviews provide an opportunity for development to take advantage the experience of security SMEs to spot problematic usage and provide suggestions as to secure coding best practices. The secure code review summary captures the outcomes from such reviews.

[Implementation-10] Fuzz Testing

Fuzz testing is a specialized form of dynamic analysis used to induce program failure by deliberately introducing malformed or random data to an application.

Fuzz Testing Report

Although the quality of security-related feedback from the compiler, static and dynamic analysis tools have become much better over time, there are many situations they do not consider. Fuzz testing provides coverage for some of those deficiencies. Although lacking the level of automation of other methods, fuzz testing allows for highly tailored behavior analysis at the unit level.

[Implementation-11] Implementation Gate

Implementation phase exit is conditional (formally gated) on completion of all AVCDL phase requirements and work products for this phase.

Formal Gate Signoff

The formal gate signoff details and documents how the phase's work products have been completed.

Phase Product Linking

When the product is being created to a particular certification, the work products on the phase need to be mapped to the requirements of that certification.

[Verification-1] Penetration Testing

Penetration testing identifies vulnerabilities before software is released so they can be corrected before release, which prevents exploitation.

Penetration Testing Report

Although the quality of feedback from the compiler, static and dynamic analysis tools, and fuzz testing provides a great deal of insight into security-related issues, they generally only expose possible security issues. Penetration testing provides analysis into possible real-world attacks on the system as a whole and exploits only achievable through taking advantage of multiple security deficiencies. The penetration testing report documents those deficiencies.

[Verification-2] Threat Model Review

The threat models should be reviewed to ensure that any design or implementation changes to the system have been accounted for, and that any new attack vectors created as a result of the changes have been reviewed and mitigated.

Updated Threat Model

Many changes can occur between the creation/update of a threat model during the design phase and the verification phase. Reviewing the threat model allows for the verification that issues identified for mitigation have been appropriately dealt with and that no new issues have arisen.

[Verification-3] Attack Surface Analysis Review

The attack surface analysis should be reviewed to ensure that any design or implementation changes to the system have been accounted for, and that any new attack vectors created as a result of the changes have been reviewed and mitigated.

Updated Attack Surface Analysis

Many changes can occur between the creation/update of an attack surface analysis during the design phase and the verification phase. Performing another attack surface analysis allows for the verification that issues identified for mitigation have been appropriately dealt with and that no new issues have arisen.

[Verification-4] Verification Gate

Verification phase exit is conditional (formally gated) on completion of all AVCDL phase requirements and work products for this phase.

Formal Gate Signoff

The formal gate signoff details and documents how the phase's work products have been completed.

Phase Product Linking

When the product is being created to a particular certification, the work products on the phase need to be mapped to the requirements of that certification.

[Release-1] Final Security Review

The Final Security Review (FSR) is a deliberate examination of all the security activities performed on a software application prior to release.

Final Security Review Report

This is a deliberate examination of all the security activities performed on a software application prior to release. The FSR is performed by the security advisor with assistance from the regular development staff and the security and privacy team leads. The FSR is not a “penetrate and patch” exercise, nor is it a chance to perform security activities that were previously ignored or forgotten. Regressions discovered at this stage indicate a failure in the verification phase.

[Release-2] Archive

All pertinent information and data must be archived to allow for post-release servicing of the software.

Archive Manifest

It is critical that all elements and information necessary to regenerate a product are archived and that a manifest of these materials is created. This serves to ensure that the product release can be updated should the need arise and provides a mechanism to allow for the quick determination of vulnerability for any of the elements used in the product’s creation.

[Release-3] Release Gate

Release phase exit is conditional (formally gated) on completion of all AVCDL phase requirements and work products for this phase.

Formal Gate Signoff

The formal gate signoff details and documents how the phase’s work products have been completed.

Phase Product Linking

When the product is being created to a particular certification, the work products on the phase need to be mapped to the requirements of that certification.

[Operation-1] Identify and Confirm Vulnerabilities

Help ensure vulnerabilities are identified more quickly so they can be remediated more quickly, reducing the window of opportunity for attackers.

Cybersecurity Incident Report

A cybersecurity incident report is the culmination of a series of activities. There will be times when the report is abbreviated as one or more of the possible activities may be determined as unnecessary. This work product contributes the result of the issue ingest and confirmation activities.

[Operation-2] Assess and Prioritize the Remediation

Help ensure vulnerabilities are remediated as quickly as necessary, reducing the window of opportunity for attackers.

Cybersecurity Incident Report

A cybersecurity incident report is the culmination of a series of activities. There will be times when the report is abbreviated as one or more of the possible activities may be determined as unnecessary. This work product contributes the result of the issue assessment and prioritization activities.

[Operation-3] Root Cause Vulnerabilities

Help reduce the frequency of vulnerabilities in the future.

Cybersecurity Incident Report

A cybersecurity incident report is the culmination of a series of activities. There will be times when the report is abbreviated as one or more of the possible activities may be determined as unnecessary. This work product contributes the result of the issue root cause activities.

[Operation-4] Secure Deployment

Software must be deployed in a secure manner and the details of the deployment recorded.

Software Deployment Report

It is critical that all software and configuration information necessary to the operation a product are archived and that a manifest of these materials is created. This serves to ensure that the product release can be updated should the need arise and provides a mechanism to allow for the quick determination of vulnerability for any of the elements used in the product's creation. It can also serve as a baseline for determination of attempted alteration of the system.

[Decommissioning-1] Apply Decommissioning Protocol

The decommissioning protocol specified in the decommissioning plan should be applied to the system coming out of service. Although both return for servicing (RMA) and end-of-life entail taking the product out of the field, each requires unique security activities. When it is necessary to return a component for failure analysis, there is a need to maintain as much of the relevant state as possible. This poses many issues as the relevant information may be entangled with other security-relevant information which we desire not to expose. Things are less complicated when the desire is to permanently remove a vehicle from service as it is far easier to scrub a system entirely than selectively remove sensitive information

Decommissioning Report

The decommissioning report details the implementation of the decommissioning protocol set out in the decommissioning plan created during the foundation phase.

References

1. **UNECE trans WP.29 GRVA 2019 2 - World Forum for Harmonization of Vehicle Regulations: Proposal for a Recommendation on Cyber Security**
   https://unece.org/fileadmin/DAM/trans/doc/2019/wp29grva/ECE-TRANS-WP29-GRVA-2019-02e.pdf
2. **Software Package Data Exchange® (SPDX®)**
   https://spdx.dev/wp-content/uploads/sites/41/2017/12/spdxversion2.1.pdf
3. **ISO 19770-2:2015 Information technology - IT asset management - Part 2: Software identification tag**
   https://www.iso.org/standard/65666.html
4. **NIST IR 8060 Guidelines for the Creation of Interoperable Software Identification (SWID) Tags**
   https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8060.pdf
5. **NIST SP 800-181 NICE Cybersecurity Workforce Framework (NCWF)**
   https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-181r1.pdf
6. **Responsibility Assignment Matrix**
   https://en.wikipedia.org/wiki/Responsibility_assignment_matrix
7. **Capability Maturity Model**
   https://en.wikipedia.org/wiki/Capability_Maturity_Model
8. **Software Bill of Materials**
   https://www.ntia.gov/sbom