Swift Customer Security Controls Framework v2024

Customer Security Programme

Detailed Description

This document establishes a set of mandatory and advisory security controls for the operating environment of Swift users. Mandatory security controls build on existing guidance and establish a security baseline for the entire user community. Advisory controls are optional best practices that Swift recommends each user to implement in the operating environment. This document must be read in conjunction with the CSP FAQ Swift Knowledge Centre article 5021823 which provides additional valuable information.

07 July 2023

Table of Contents

Executive Summary

Launched in 2016 in response to the sophisticated cyber-attacks on Swift users, the Customer Security Programme (CSP) seeks to maintain appropriate cyber-security hygiene across all users, reduce the risk of cyber-attacks and minimise the financial impact of fraudulent transactions. There has been a continued evolution since 2016, with enhanced digitalisation or cloudification and Swift users facing attacks of increasing levels of sophistication. Modus operandi, the Tactics, Techniques, and Procedures (TTPs) have progressed and changed as institutions strengthen security measures. The persistence of such threats emphasises the importance of remaining vigilant and proactive in the long term. While users are responsible for protecting their own environments and accesses to Swift, the CSP has been introduced to support customers and drive industry-wide collaboration in the fight against cyber fraud. The CSP establishes a common set of security controls known as the Customer Security Controls Framework (CSCF) which is designed to help customers to secure their own environments and to foster a more secure financial ecosystem.

The Swift CSCF consists of both mandatory and advisory security controls which are based on industry-standard frameworks, such as NIST, ISO 27000 and PCI-DSS. Mandatory security controls establish a security baseline for the entire community and must be implemented by all users on their Swift infrastructure. Swift prioritises the mandatory controls to set a realistic goal for short-term, tangible security gains, as well as risk reduction. Advisory controls are based on best practices that Swift recommends users to implement. Gradually over time, the mandatory controls may change through the evolving threat landscape, and some advisory controls may become mandatory.

Swift details all controls around the following three overarching objectives:

- secure your environment
- know and limit access
- detect and respond

The controls have been developed based on a continuous Swift's analysis of cyber-threat intelligence and in conjunction with industry expert and user feedback. The control definitions are also intended to align with existing information security industry standards.

The controls detailed in this document represent general product-agnostic controls. The controls should not be considered exhaustive or all-inclusive, and do not replace a well-structured security and risk framework that covers the end-to-end transaction chain, sound judgement, or compliance with the latest security best practices.

Given the evolving nature of cyber-threats, the introduction of new technologies, and updated Swift strategic initiatives, controls will be regularly assessed, refined, and expanded with the changes published in new versions of this document. Consequently, Swift recommends users to always consult the latest version of this document through the Swift Knowledge Centre (KC).

To support the adoption of the security controls, Swift has developed a process that requires users to attest compliance against the mandatory (and optional advisory) security controls. Swift requests users to submit an attestation into the KYC Security Attestation (KYC-SA) application. By the end of each year, users must attest compliance against the mandatory (and optional advisory) security controls as documented in the CSCF effective at that time. Generally, a new version of the CSCF is published in July, listing the mandatory and advisory controls users must attest against (as of July of the following year when implemented in the KYC-SA). That is, users must attest between July 2024 and December 2024 against the security controls listed in the CSCF v2024 published in mid-2023.

All users retain control over their own data and are able to grant (automatically or not) access to allow counterparties to view their attestation data. This fosters transparency and creates peer-driven momentum to improve security practices by allowing other users on the network to apply risk-based decision-making efforts concerning their business relationships. For more information about the attestation and reporting process, see the *Swift Customer Security Controls Policy (available in* KC*)*.

The CSP is designed to be a collaborative effort between Swift and the users to strengthen the overall security of the financial ecosystem. As part of the CSCF Controls change management process, extensive consultation is undertaken with the community. The CSCF Working Group, established in 2021, is structured around 25 NMGs with 21 of them continue to positively answer to initial consultation requests. The role of the CSCF Working Group is to centralise, prioritise and review all formal and informal feedback from the community and then finalise the recommended changes. The Working Group results are also shared with Swift Oversight to ensure transparency with Swift Oversight. Therefore, all users must read the controls set out in this document carefully, and prepare their own organisation for implementation accordingly.

Overview of changes

The Swift Customer Security Controls Framework (CSCF) version 2024 builds incrementally on last year's version (CSCF v2023). The CSCF Working Group reviewed several possible 'Change Requests' (CRs), including scope changes, guidance clarifications, cosmetic changes, and open questions.

In support of the ramping up of outsourcing and cloudification within our community, **control 2.8 (Outsourced Critical Activity Protection) is turned mandatory** and subject to some clarifications.

To support a phased promotion of control 2.4A (Back Office Data Flow Security) to mandatory, several changes are made to this control to properly identify:

- the servers bridging the back office and the user's secure zone
- the mechanisms securing the data flow exchange, either
  - by end-to-end data protection or
  - by securing each flow segment and the supporting 'bridging servers' which are guardians of such data exchange.

While the control 2.4A remains advisory, Swift recommends to already identify these flows and assess their security posture.

Further minor clarifications or changes have been brought to specific controls or to the overall CSCF framework to improve the usability and comprehension of the document and to help users implement the framework as intended:

- Controls 2.1 (Internal Data Flow Security) and 2.4 (Back Office Data Flow Security) are sharing the same risk drivers
- Guidance related to USB ports protection is now fully in control 2.3 (System Hardening)
- The (common) optional enhancement related to application allowlisting has been moved from controls 1.1, 1.5 and 6.2 to control 2.3 (System Hardening)
- Control 2.9 (Transaction Business Controls) now explicitly mentions business controls can be performed outside the secure zone
- Wording regarding supervision and secure storing of tokens has been aligned in control 3.1 (Physical Security) and 5.2 (Token Management)
- Control 3.1 (Physical Security) now contains recommendations on reasonable sanitisation of disposed or reassigned equipment
- Some controls titles have been aligned for taxonomy purpose (5.4 Password Repository Protection & 2.8 Outsourced Critical Activity Protection)
- Wording regarding software integrity check has been aligned between control 6.2 (Software integrity) and control 2.2 (Security Updates)
- Control 6.4 (Logging and Monitoring) is referred to, when relevant, in other controls requiring to monitor the logs
- Control 7.4 (Scenario-based Risk Assessment) now explicitly mentions reliance can be made on existing Information Security Risk Management processes
- Appendix D (Glossary of Terms) and sections in the introduction revisit and clarify the notions of service providers and third parties including outsourcing agent
- Appendix E (Mapping to Industry Standards) will be aligned with the latest security standard versions
- Appendix F (Services and Components in scope per architecture type) is updated with information about the CREST GUI and Gateway, such information was previously in the related Knowledge Centre article
- The section Scope of Security Controls and Appendices D and F clarify that WebAccess webservers are to be identified and secured as customer connectors by the WebAccess service providers.

Swift recommends users to consult the *CSCF v2024 compared to v2023* version of this document to view full details of changes. This file does not include editorial changes that Swift performs to improve the readability of the document.

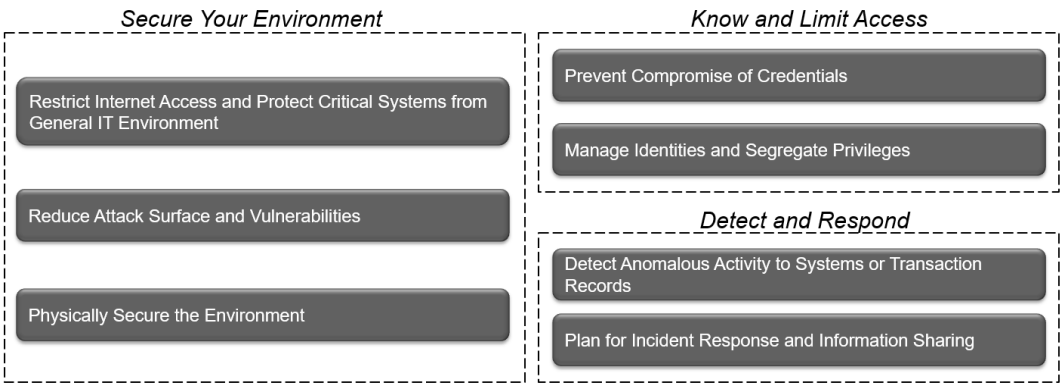
To further ease the performance of independent assessments, Swift reminds users of the following:

- Compliance with control objectives is a risk-based approach. The provided implementation guidelines can be used as a starting point, but cannot be considered as strict audit checklists.
- Users engaging with third parties (including cloud providers) to host or operate (in full or in part) their own Swift infrastructure must obtain reasonable comfort from third parties that the outsourced activities or externally hosted components are protected per the security controls. As such, third parties can rely on their compliance programme that usually builds on SOC 2, PCI-DSS or NIST certifications or assurance to answer users engaging with them and map the CSCF security controls. For example, **Appendix G** presents the shared responsibilities when moving to an Infrastructure as a Service (IaaS) model in the cloud.

Framework Objectives and Principles

Objectives and Principles

The security controls are based on three overarching framework objectives, supported by eight security principles. Objectives are the highest level structure for security within the user's environment. The associated principles elaborate on the highest priority focus areas within each objective. The objectives and corresponding principles include the following:



Drawing 1: Framework Objectives and Principles

The 32 security controls (25 mandatory controls and 7 advisory controls) detailed in this document underpin these objectives and principles where the first two principles, sharing common controls, have been grouped. The controls help mitigate specific cyber-security risks that Swift users face due to the cyber-threat landscape. Within each security control, Swift has documented the most common risk drivers that the control is designed to help mitigate. Addressing these risks aims to prevent or minimise undesirable and potentially fraudulent business consequences, such as the following:

- unauthorised sending or modification of financial transactions
- processing of altered or unauthorised Swift inbound transactions (that is, received transactions)
- business conducted with an unauthorised counterparty
- confidentiality breach (of business data, computer systems, or operator details)

- integrity breach (of business data, computer systems, or operator details).

Ultimately, these consequences represent enterprise-level risks, including the following:

- Financial Risk
- Legal Risk
- Regulatory Risk
- Reputational Risk.

Integration with Security Governance and Risk Management

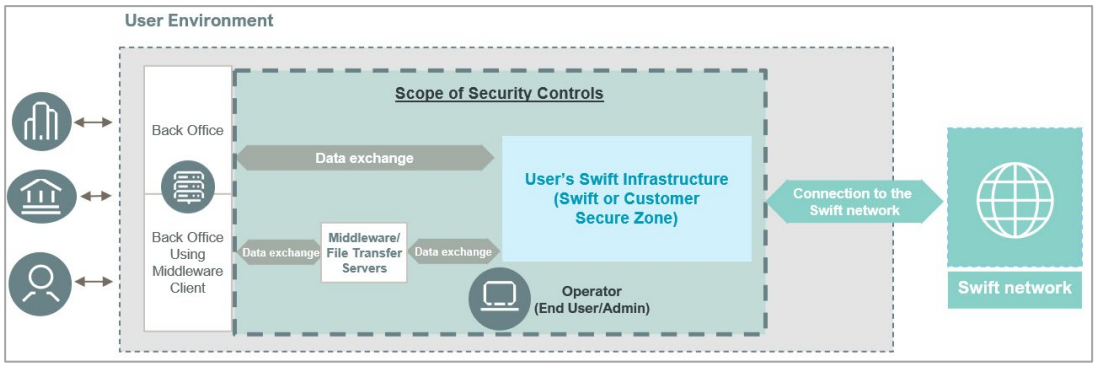
Swift encourages users to consider cyber risk management in the broadest possible terms, including beyond the scope of the user's Swift infrastructure and the Swift security controls. For the most effective management of risk, users should not view the implementation of these security controls as a one-time activity, nor as exhaustive or all-inclusive. Users should, instead, incorporate Swift's controls into an ongoing cyber-security governance and risk programme within their organisation, considering sound judgement and the latest best practices, taking into account user-specific infrastructure and configurations. As a result, users can re-use and benefit from existing policies, procedures, and controls that have been established to manage other areas of cyber risk. To help users in this approach, **Appendix E** contains a mapping of the Swift security controls against three international security standard frameworks: **NIST Cybersecurity Framework v1.1**, **ISO 27002 (2013)**, and **PCI-DSS 3.2.1**. Swift has also published a Getting Started Guide to assist users in assessing counterparty cyber-security risk and incorporating this into the risk management framework.

A holistic approach to cyber risk is most effective in avoiding enterprise-level risk, therefore improving the overall safety of each individual organisation and the wider financial community.

Additionally, users should have the correct level of accountability and oversight for their cyber risk management activities. Generally, a Chief Information Security Officer plays a prominent role in this domain by directing the priorities of the security programme and soliciting the appropriate support and guidance from the Board.

Scope of Security Controls

The scope of security controls in this document encompasses a defined set of components in the user's environment (see Drawing 2).



Drawing 2: Scope of Security Controls

A) The security controls apply to the following **in-scope components**:

- **User's Swift infrastructure –** The collection of - on-premises or remote (that is hosted or operated by a third party, or both), - Swift-specific components managed by or for users, including applications, network devices, tokens and other removable media, and supporting hardware. Examples of user's Swift infrastructure set-up and so called Swift-related components (depending on the user architecture type) are as follows:
  - **Swift Secure Zone**: a segmented zone that separates Swift-related systems from the wider enterprise environment (further detailed in control 1.1). This zone can expand beyond the user's Swift infrastructure and can include non-Swift systems. Its primary purpose is to host the below identified Swift footprints.
  - **Messaging Interface**: a software that supports the use of MT, MX, or ISO 20022 message standards through Swift FIN, InterAct, FileAct, and SWIFTNet Instant messaging services. The software provides the means for users to connect these business applications to Swift messaging services and is typically connected directly to the communication interface. Swift provides messaging interfaces (for example, Alliance Access and Alliance Messaging Hub or Alliance Messaging Hub Instant). Messaging interfaces that hold a Swift-compatible label can also be provided by third-party vendors. A Messaging Interface is considered as a Swift footprint.
  - **Communication Interface**: a software that provides a link between the SWIFTNet network and usually* the Messaging Interface software or a back-office system. Communication interfaces provide centralised, automated, and high-throughput integrations with different in-house financial applications and service-specific interfaces. Swift provides the communication Interfaces (for example, Alliance Gateway or Alliance Gateway Instant). Communication interfaces that hold a Swift-compatible label can also be provided by third-party vendors. A Communication Interface is considered as a Swift footprint. * While Alliance Gateway is usually linked with a Messaging Interface, Alliance Gateway Instant is usually directly linked with a back-office system (unless an explicit Messaging Interface is also integrated in the user infrastructure).
  - **SWIFTNet Link (SNL)**: SNL is a mandatory software product for access to FIN, InterAct, and FileAct messaging services over a secure IP network. This document refers to the SNL, a Swift footprint, as part of the Communication Interface scope.
  - **Connector**: Connectors are local software designed to facilitate communication with an external messaging interface or a communication interface (or both), to a service provider (handling as such the external connection) or an outsourcing agent. When using a connector, interface components are usually offered by a service provider (for example, offered by a service bureau, a hub infrastructure, or Swift).

Swift connector is a connector specifically designed to support Swift business. It is usually provided by Swift (for example, Alliance Cloud SIL, Direct Link, Alliance Lite2 AutoClient, in combination with SIL or not in combination with SIL, or Microgateway). Swift connector, holding a Swift-compatible label, can also be provided by third-party vendors. A Swift connector is considered as a Swift footprint.

Customer connector is typically a commercial off-the-shelf product configured for Swift purposes. It includes generic file transfer solutions or local middleware systems implementations (such as IBM® MQ server, Apache Kafka or Solace server/broker) used to facilitate an external connection with Swift-related components offered by a service provider or an outsourcing agent. Those generic elements not provided by Swift (or not labelled as Swift-compatible) are considered as a non-Swift footprint.

An application developed in-house that implements Swift APIs (basically by either using the specifications of or integrating the Swift Security SDK) to connect and transmit business transactions independently[1] to Swift messaging services[2] exposed by the Swift API Gateway, is also considered as a customer (bespoke API) connector or a non-Swift footprint.

The 'customer facing' webservers operated by Swift WebAccess service providers are also considered as customer connectors.

When used alone, the term ***connector*** refers to both Swift connectors and customer connectors.

- **Customer Secure Zone**: a secure operational (sometime also called production) environment that separates non-Swift footprint, such as the customer connector, from the wider enterprise environment (further detailed in control 1.5). Its primary purpose is to host the customer connector but can also include non-Swift related systems which then also need to be adequately protected. ***Users already having a Swift secure zone may consider adding in that zone a customer connector (in which case there is no need to create a specific customer secure zone for it).*** Note: a customer secure zone can also host systems supporting an existing Swift secure zone such as a storage area network (SAN), a hosted database, a virtualisation platform, authentication services or middleware servers.

  Secure zone: refers to a Swift or a customer secure zone.

- **Swift Hardware Security Modules (HSMs)**, connected and disconnected personal tokens, and smart cards.

  A ***new HSM*** model (the PED-less Luna SA7) will be available as from 2023 and will progressively replace the existing HSM model (the Luna IS6). While the same level of security will be maintained with this replacement, the new HSM will, due to its PED-less nature, be identified as in scope of all the relevant controls. As such:

  - The term "***HSM***", refers to both existing (Luna IS6) and new (Luna SA7) models -no change.
  - The term "***New HSM***" will be used when newly identified as in scope component of several controls to facilitate the transition. Consequently, those controls need to be formally taken into account when installing, using the new HSM model (Luna SA7 as per documentation) and further assessing compliance with those controls.
  - At the end of the HSM migration, the term "new" in "new HSM" will disappear and "HSM" will remain.

[1] Independently means without using a communication interface or a Swift connector such as SIL, Direct Link or Microgateway

[2] Business transactions to messaging services refers to requests introducing or affecting Swift payments (such as creation of MT101, 103, 202, 205, their MX equivalent, pain.001, pacs.008, pacs.009, or cancelling/stopping/recalling/modifying those requests). On the other side, queries on previous transactions (such as through the Basic Tracker), pre-validation, conversion or screening performed before submitting business transactions are not considered affecting messaging services (being ancillary services) and can be considered as out of scope unless they require the same roles/entitlements as business transactions to messaging services (no separation of duties and precautionary principle has to be applied).

- Firewalls, switches, and routers within or surrounding the user's Swift infrastructure (dedicated or shared) referred to generically as network devices protecting the secure zone.
- **Graphical user interface (GUI)**: software that produces the graphical interface for a messaging interface, a communication interface or a connector (for example, Alliance Web Platform Server-Embedded, Swift Microgateway Front-End, and equivalent products).

- **Operators**: Operators are individual end users and administrators that directly interact with the users' Swift infrastructure at the application or OS level.
- **Operator PCs:** the end user's or administrator's computing device (typically a desktop or laptop) used to conduct their duties as an operator (to use, operate, or maintain the user's Swift infrastructure residing on premises or externally hosted) or as a user (to use a remote Swift infrastructure or application operated by a service provider, such as a service bureau, a Lite2 for Business Application or Business Connect provider, a Group Hub, a third party, such as an outsourcing agent, or Swift), or a combination, depending on the architecture type.

  A ***general-purpose operator PC*** is generally located in the general enterprise IT environment and is used for daily business activities including accessing the user's or a remote Swift infrastructure or an application operated by a service provider or an outsourcing agent, depending on the architecture type. It also includes devices (physical or virtual desktop, laptops but also other devices such as tablets or mobiles) managed by the customer and used to interact with the messaging or communication interface, a GUI, a connector, a transaction business control solution, or a Swift-related application exposed by a service provider or an outsourcing agent.

  A ***dedicated operator PC*** is located in the secure zone and is dedicated to interact with components of the secure zone (sometimes also referred to as an *operational console*).

  The term ***operator PC*** alone refers to both general-purpose and dedicated operator PCs.
- **Data exchange layer:** the transport of data between the Swift-related components (in the user's Swift infrastructure, at a service provider or at an outsourcing agent) and a user back-office first hop, at application level, as seen from the Swift-related components. This can be performed directly between the back-office first hop and the component in the secure zone or through **bridging servers** such as (one or a chain of) middleware or file transfer servers.
- **Middleware server:** local middleware systems implementations, such as the IBM® MQ server (including MQ queues manager, MQ appliance, or both), Apache Kafka or Solace server/broker, used for data exchange between the Swift-related components (in the user's Swift infrastructure, at a service provider or at an outsourcing agent) and the user back-office first hop. Usually considered as a bridging server, it must however be considered as a customer connector when used to facilitate an external connection with Swift-related components offered by a service provider (such as a service bureau, or potentially a Business Connect or a Lite2 for Business Application provider) or an outsourcing agent.
- **File transfer server:** solution, such as an sFTP server, used to exchange files between the Swift-related components (in the user's Swift infrastructure, at a service provider or at an outsourcing agent) and the user back-office first hop. Usually considered as a bridging server, it must however be considered as a customer connector when used to facilitate an external connection with Swift-related components offered by a service provider (such as a service bureau, or potentially a Business Connect or a Lite2 for Business Application provider) or an outsourcing agent.

B) The following **components** are **out of scope**:

- **User back office:** the systems responsible for business logic, (financial) transaction generation, and other activities that occur before transmission into the user's Swift infrastructure. For example:

- Back-office implementations such as SAP, General Ledger, or applications that use an MQ, Kafka or Solace kind of client to liaise with the user's Swift infrastructure are out of scope unless co-hosted[3] with an in-scope component.
- An application or system relying on a communication interface, such as Alliance Gateway (see Drawing 3b) or a Swift connector, such as Direct Link or Microgateway (see Drawing 5) or, a customer connector (see Drawing 6b) for API calls to Swift remains a back-office system and is out of scope unless i) co-hosted[3] with the communication interface or the connector or ii) using the specifications of or integrating the Swift Security SDK to connect and transmit business transactions independently to Swift.

Although back-office systems responsible for transactions generation are typically outside of the CSP/CSCF scope, it is strongly recommended to protect them by implementing controls from security framework(s) (potentially using the CSCF principles) to minimise the risks into the broader end-to-end transaction chain.

- **General Enterprise IT environment:** the general IT infrastructure used to support the broader organisation (for example, general-purpose PCs, mail servers, or directory services).
- Connections to the Swift network supplied by Swift Network Partners and Internet connections to the Swift network are also out of scope. **However**, the remotely managed by Swift Alliance Connect SRX VPN boxes are in scope of control **3.1**; they are expected to be in an environment with appropriate physical controls. **Similarly**, the Alliance Connect Virtual VPN instances (hosting systems or machines), only available through the cloud providers approved as per the Swift Cloud Provider Programme, are also in scope of the controls **3.1** and **1.3** that are covered by those cloud providers as per the programme.

Note: **Users must attest** for all in-scope components in their live, back-up, and disaster recovery environment, while **taking into account the** specific but still **comprehensive architecture, specifically declaring the most encompassing or all-embracing architecture type;** that architecture type can be identified using the decision tree.

- Test systems are not considered in scope of the security controls as long as (i) they are fully separated from production or live environment (including separate HSMs) and (ii) they are configured to only support test traffic (for example, by only using lite certificates on test only logical terminals). If the test systems are not fully separated or can be configured for live traffic, then users must take the test systems in scope and make sure that the same security controls are applied as for production or live systems.

Similar to the back office, users should still implement good security hygiene on their test systems to also make sure they cannot be easily reconfigured for live traffic.

- Development systems must not be within the secure zone and are not connected to the Swift network and are therefore, not considered in scope.

C) **Co-hosting components** and turning those components **in scope:**

The primary purpose of a secure zone is to host Swift-related components but it can also include non-Swift related systems as long as they are adequately protected. All components within the secure zone must be protected to an equivalent level of security and trust by applying controls applicable to the Swift-related components (see the CSP FAQ for the relevant controls).

Similarly, the purpose of a Swift-related system is to host or run a Swift-related component turning such system and co-hosted[4] (running on that system) applications in scope, irrespective on the underlying layer (individual/physical system, virtual machine on a virtualisation platform or in the Cloud). The virtualisation and cloud platform must also be

[3] Components deployed or running on a same physical system or virtual machine (created on a virtualisation or cloud platform). Note that containerised applications are considered as co-hosted on the same physical system or virtual machine that hosts the container itself.

[4] Components deployed or running on a same physical system or virtual machine (created on a virtualisation or cloud platform). Note that containerised applications are considered as co-hosted on the same physical system or virtual machine that hosts the container itself.

specifically considered; deployment in the Cloud is simply an abstraction of the virtualisation platform and a similar approach has to be sought in line with the next below note.

D) **Sharing/reusing credentials of components in scope:**

As a rule, ancillary or complementary services, allowing queries on preceding transactions (for instance through the Basic Tracker), pre-validation, conversion or screening and performed before submitting actual business transactions, are considered as out of scope.

However, sharing/reusing credentials and roles/entitlements used for business transactions on systems or components for those ancillary services turns those systems or components in scope.

Appendix F supports the identification of elements or Swift-related components in-scope.

Note: Users that engage with third parties (for example, an external IT provider, or a cloud provider as outsourcing agent – see https://www2.swift.com/go/book/book201561 for more info) or service providers (such as a service bureau, a Business Connect or a Lite2 for Business Application provider which, in this specific case, must be considered as a an outsourcing agent) to host or operate in full or in part the user's own Swift infrastructure must be aware of the following:

- Users are still responsible and accountable to attest for their comprehensive architecture type (as if it was operated on premises).
- Consequently, users must get reasonable comfort[5] from such third parties or service providers that the related activities are protected, at a minimum, to the same standard of care as if operated within the originating organisation and in line with the CSCF security controls. As such, third parties can rely on their compliance programme that usually builds on SOC 2 type 2, PCI-DSS or NIST certifications or assurance to answer users engaging with them.

Appendix G illustrates a typical spread of responsibilities to consider when outsourcing to a cloud provider through an Infrastructure as a Service (IaaS) model.

[5] See the Glossary of Terms for the definition

Architecture Types

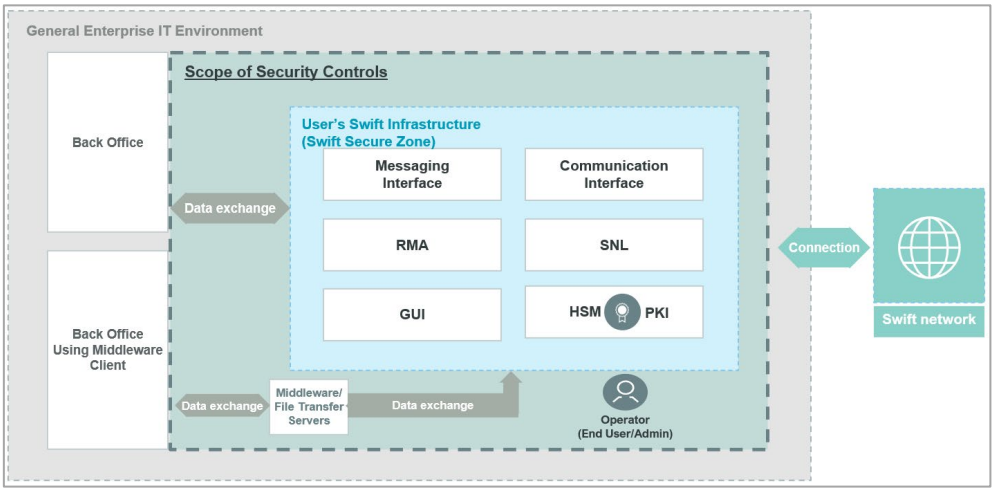
Each user must identify which of the five reference architecture types (Drawings 3-7) most closely matches their own architecture deployment to determine which components are in scope (a CSP architecture decision tree is available). Depending on the architecture type where the most comprehensive one has to be chosen, some security controls may or may not apply.

The five reference architectures are as follows, where component or licence ownership is the key differentiator:

- **Architecture A1** – Users owning the communication interface (and, generally, the messaging interface)

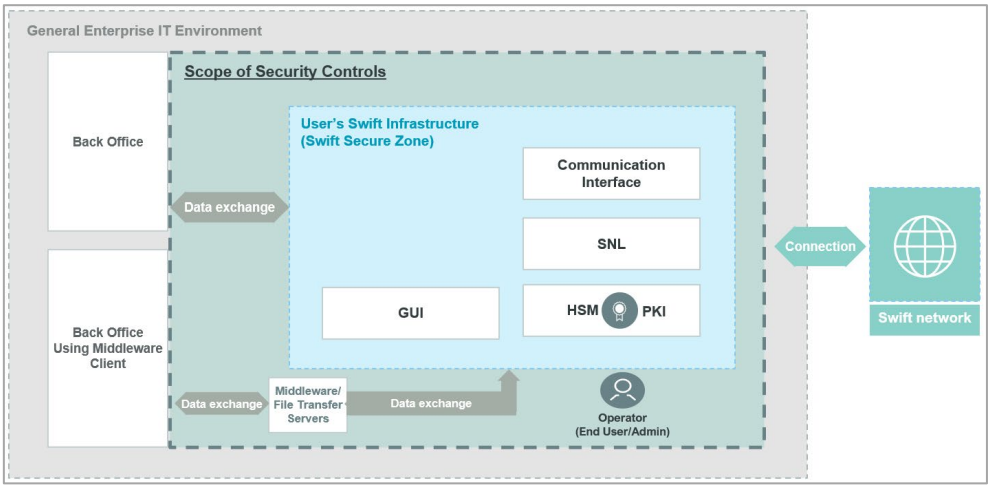
  The communication interface is owned by the user.

  Drawing 3a displays the case where both the messaging interface and the communication interface licences are owned by the user and reside within the environment.



Drawing 3a: Architecture A1 – Interfaces within the user environment

Users that do not own a messaging interface but only own a communication interface (such as in the Drawing 3b below) are also considered as **architecture A1**.



Drawing 3b: Architecture A1 – Communication interface only within the user environment

The **architecture A1** type also includes hosted solutions where the user owns the licence for the communication interface that the user operates on behalf of other users, or the communication interface owned by the user is operated for personal use by a third party within (or hosted) outside the user environment. An Alliance Gateway Instant

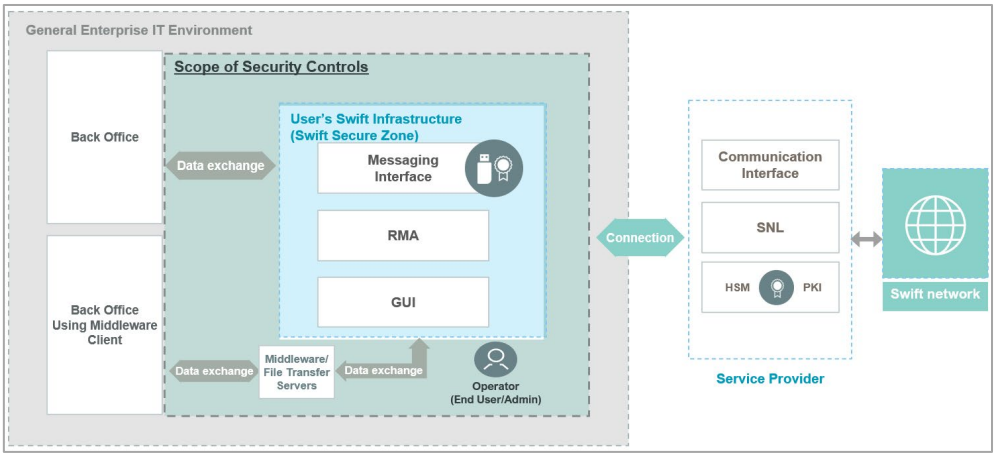
linked with a back-office system without any Messaging Interface illustrates such set-up.

Although back-office systems responsible for transactions generation are typically outside of the CSP/CSCF scope, it is strongly recommended to protect them by implementing controls from security framework(s) (potentially using the CSCF principles) to minimise the risks into the broader end-to-end transaction chain.

- **Architecture A2** – Users owning the messaging interface, but not the communication interface

  The messaging interface is owned, but a service provider (for example, a service bureau, Swift[6], or a Group Hub) owns the licence for the communication interface.

  Drawing 4 displays the case where the messaging interface is owned by the user and resides within the user environment.



Drawing 4: Architecture A2 – Messaging Interface only within the user environment

This architecture type also includes hosted solutions where the user has the licence for the messaging interface that is hosted or operated on his behalf by a third party or a service provider (that in this case must be considered as an outsourcing agent).

Although back-office systems responsible for transactions generation are typically outside of the CSP/CSCF scope, it is strongly recommended to protect them by implementing controls from security framework(s) (potentially using the CSCF principles) to minimise the risks into the broader end-to-end transaction chain.

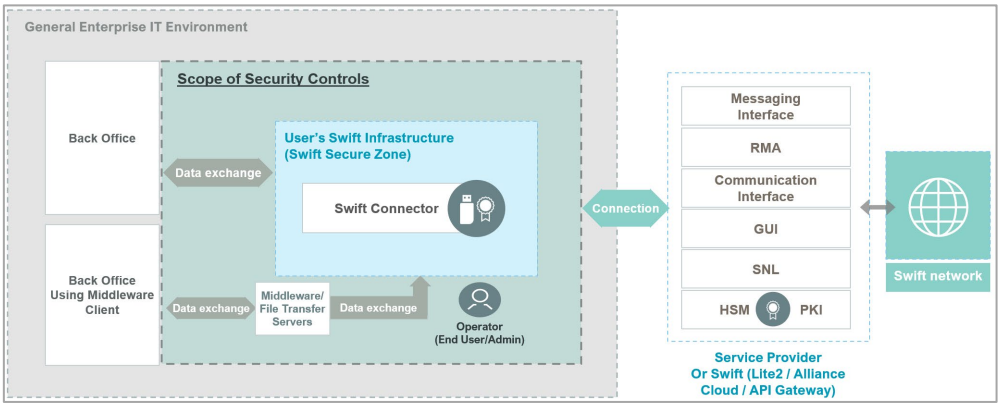
- **Architecture A3** – Swift Connector

  A Swift connector[7] is used (such as in Drawing 5) within the user environment to facilitate an application-to-application communication with an interface at a service provider (for example, a service bureau or a Group Hub) or with Swift services (such as Alliance Cloud or Alliance Lite2) with no (messaging or communication) interface.

  Optionally, this set-up can be used in combination with a GUI solution (user-to-application). In such cases, controls pertaining to the GUI must also be implemented.

[6] In the scope of Alliance Remote Gateway

[7] For example, Alliance Cloud SIL, Direct Link, Alliance Lite2 AutoClient, in combination with SIL or not, or Microgateway



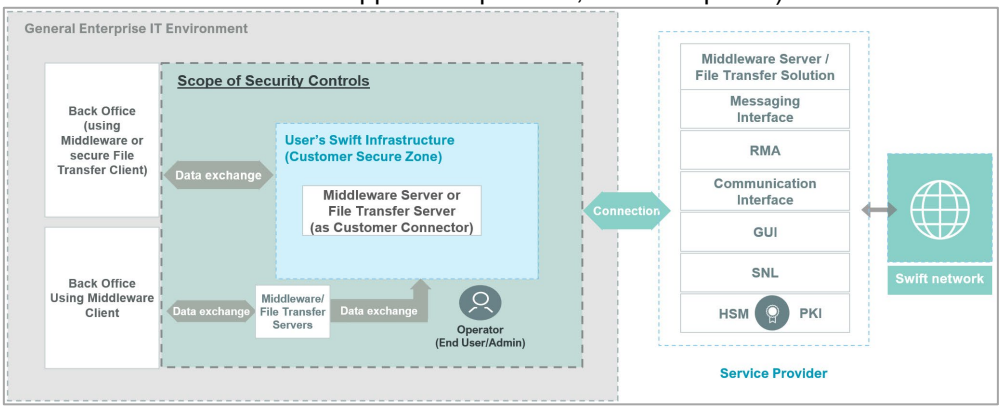
Drawing 5: Architecture A3 – Swift Connector

This architecture type also includes hosted solutions of the user's Swift connector that is hosted or operated on his behalf by a third party or a service provider (that in this case must be considered as an outsourcing agent).

Although back-office systems responsible for transactions generation are typically outside of the CSP/CSCF scope, it is strongly recommended to protect them by implementing controls from security framework(s) (potentially using the CSCF principles) to minimise the risks into the broader end-to-end transaction chain.

- **Architecture A4** – Customer Connector

  When a user does not have any Swift footprint but uses, within its environment[8], a server running a software application (for example, a file transfer solution or a middleware server that is a customer connector, see Drawing 6a) to facilitate an application-to-application external connection with an interface, a Swift-related application or solution at a service provider (for example, a service bureau, a Business Connect or a Lite2 Business Application provider, or a Group Hub).



Drawing 6a: Architecture A4 – Middleware/File Transfer Server as Connector

This architecture type also includes hosted solutions of the user's customer connector that is hosted or operated on its behalf by a third party or by a service provider (that in this case must be considered as an outsourcing agent).

Although back-office systems responsible for transactions generation are typically outside of the CSP/CSCF scope, it is strongly recommended to protect them by implementing controls from security framework(s) (potentially using the CSCF principles) to minimise the risks into the broader end-to-end transaction chain.

This architecture type also includes usage, as a customer connector, of a file transfer server or a middleware server, or both, to connect with i) an outsourcing agent or ii) a remote Group Hub through externally exposed connections (that means, not through a

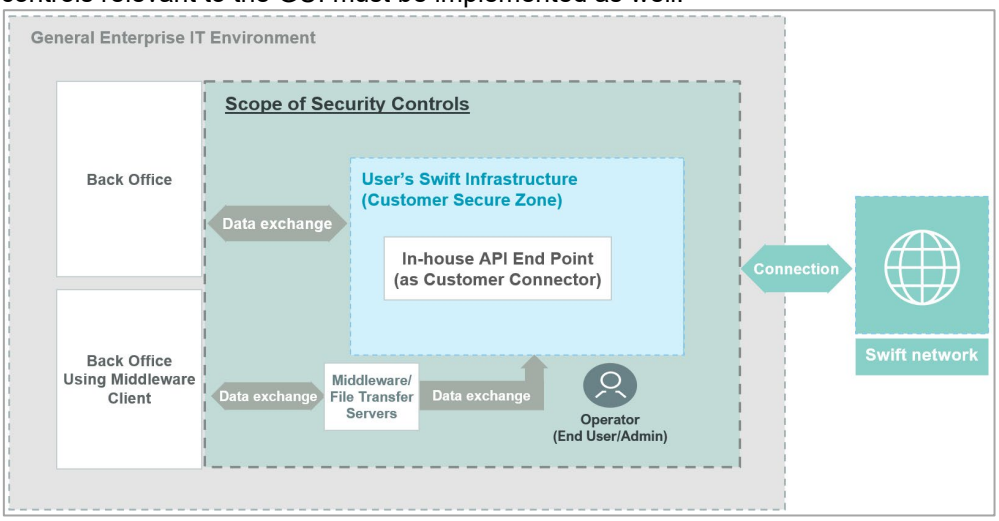
[8] On premises or externally hosted, in the Cloud or not.

wide area network - WAN or extended virtual local area network - VLAN), without any Swift connector.

When additional middleware or file transfer server(s) are securing the data exchange with the back-office first hop, users must also consider the controls applicable to bridging servers.

Architecture A4 also includes, when there is no Swift footprint, customer connectors that are applications used within the user environment[9] that implement Swift APIs (basically to directly connect and independently transmit[10] business transactions to Swift services (a future messaging service[11] or the Transaction Manager exposed by Swift). The implementation of the Swift APIs (basically by using either the specifications of or integrating the Swift Security SDK) makes such applications a custom-made API endpoint referred to as a customer connector or a non-Swift footprint (see Drawing 6b).

This last set-up could also integrate a GUI solution (user-to-application). In such case, controls relevant to the GUI must be implemented as well.



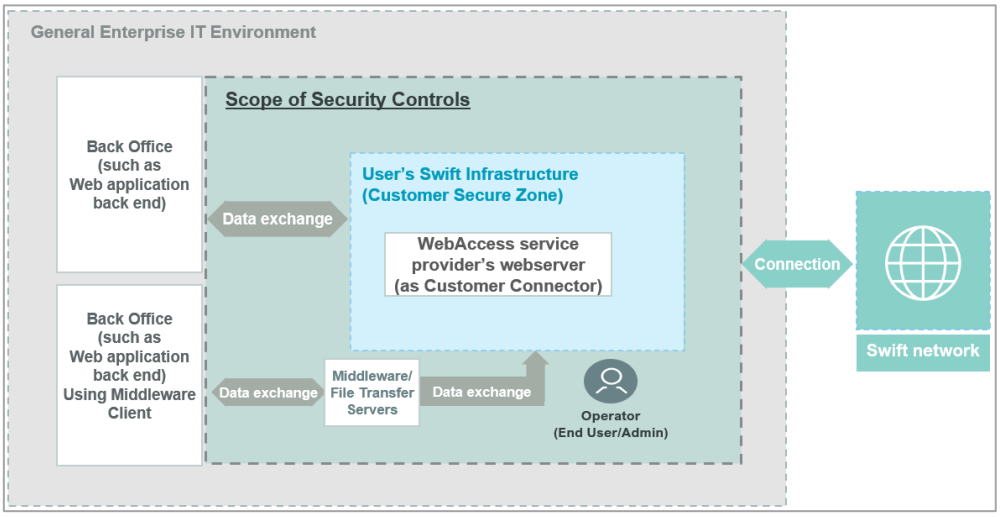
Drawing 6b: Architecture A4 – Customer (in-house API) Connector

This architecture type also includes hosted solutions of the user's customer connector that is hosted or operated on its behalf by a third party or a service provider (that in this case must be considered as an outsourcing agent).

[9] On premises or externally hosted, in the Cloud or not.

[10] Without the usage of a communication interface or a dedicated Swift (API) Connector.

[11] Business transactions to messaging services refers to requests introducing or affecting payments (such as creation of MT101, 103, 202, 205, their MX equivalent, pain.001, pacs.008 and pacs.009, or cancelling/stopping/recalling/modifying those requests). On the other side, queries on previous transactions (such as through the Basic Tracker), pre-validation, conversion or screening performed before submitting business transactions are not considered affecting messaging services.



Drawing 6c: Architecture A4 – WebAccess service provider webserver front-end

This architecture type also includes, when there is not Swift footprint, the webserver front-end, as customer connector, a WebAccess service provider relies on to offer its web application over Swift. In a distinct 2- or 3-tier model, the web application or back end is considered as a back-office system, as illustrated above.

Although back-office systems responsible for transactions generation are typically outside of the CSP/CSCF scope, it is strongly recommended to protect them by implementing controls from security framework(s) (potentially using the CSCF principles) to minimise the risks into the broader end-to-end transaction chain.

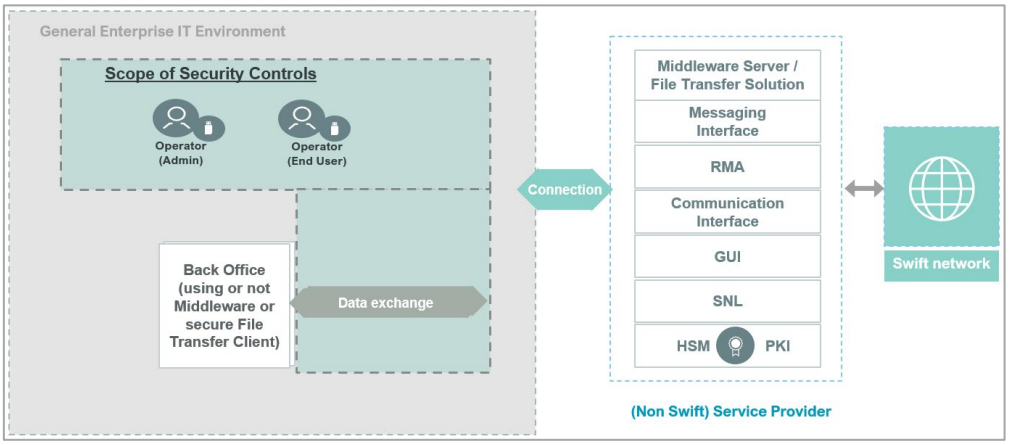
- **Architecture B** – No local user footprint

  No Swift-specific infrastructure component is used within the user environment. The following two types of set-ups are covered by this architecture type:

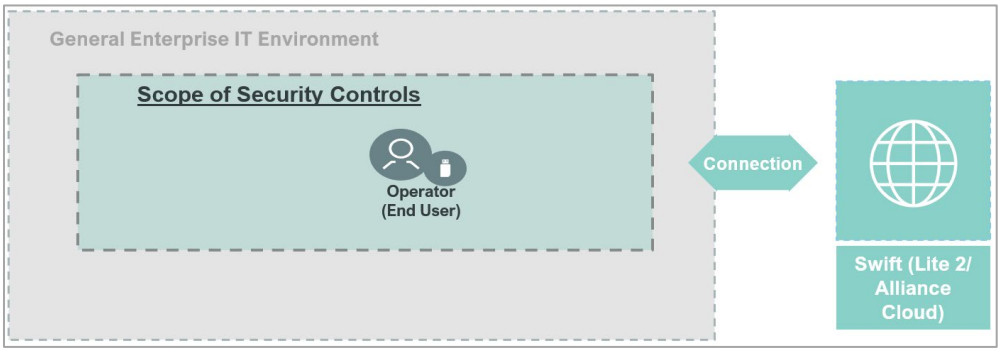
  – Users only access Swift messaging services through a GUI application at the service provider (user-to-application). The PC or device used by those users to submit or affect business transactions must be considered as a general-purpose operator PC and must be protected accordingly.

  – A user's back-office application communicate directly with the service provider (application-to-application) using APIs from the service provider, a middleware client (such as an MQ, Kafka or Solace client/broker) or a secure File Transfer client without connecting or independently transmitting business transactions to Alliance Cloud, a Swift messaging service, the Swift API Gateway[12] or, the Transaction Manager exposed by Swift. In such a case, the service provider must make sure that the security of the environment and the security of the data exchange with the user are aligned with the CSCF controls. Categorising this set-up as **Architecture B** is aligned with the scope of the security controls, which excludes user back-office applications. However, Swift strongly recommends already implementing **Architecture A4** controls on such applications that integrate APIs, a middleware client or a secure file transfer client.

[12] Would otherwise be considered as an architecture type A4 with a customer connector.



Drawing 7a: Architecture B – No user footprint connecting to a Service Provider (other than Swift)



Drawing 7b: Architecture B – Interactive User connecting to Swift

Although back-office systems responsible for transactions generation are typically outside of the CSP/CSCF scope, it is strongly recommended to protect them by implementing controls from security framework(s) (potentially using the CSCF principles) to minimise the risks into the broader end-to-end transaction chain.

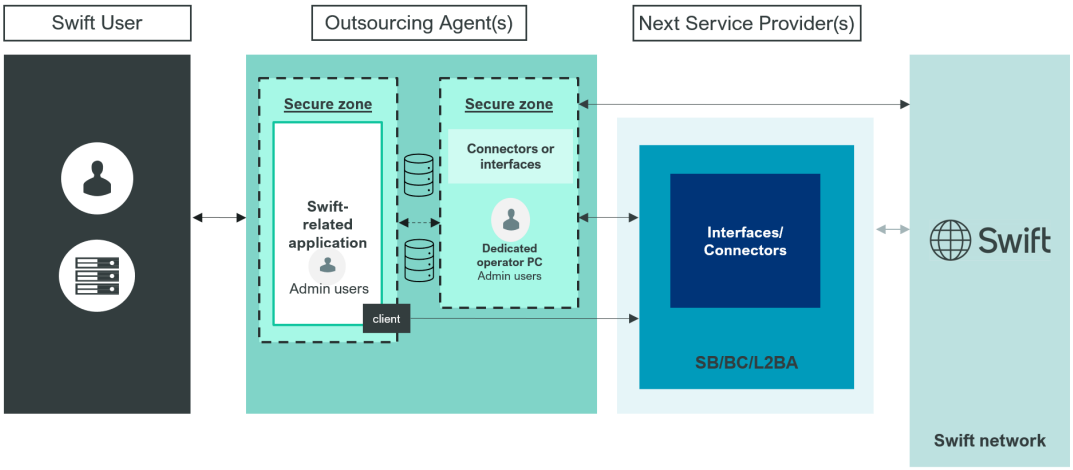
This architecture type also includes users that only access Swift messaging services (user-to-application) with a browser, exposed by Alliance Cloud and Alliance Lite2. PCs used by those users to submit or affect business transactions must be considered as general-purpose operator PCs and must be protected accordingly.

The security controls applicable for Architectures A1, A2, and A3 are identical[13] and fewer controls apply to Architecture A4. These architectures are referenced collectively on the following pages as type A. Fewer security controls apply to users that utilise Architecture type B (for more information, see the Security Controls Summary Table section).

Outsourcing agent reliance

In addition to the user's infrastructure that is, in full or in part, hosted or operated on its behalf by a third party, there are other cases where an outsourcing agent offers applications that connect to Swift through, for instance, a service bureau (SB), a Business Connect (BC) or a Lite2 Business Application (L2BA) provider as per below drawing. In this case, the user has to seek the architecture type (usually A4 or B) and compliance results from his outsourcing agent in order to fill his attestation. More information is available on the Knowledge Centre (swift.com).

[13] Except for Control 6.3 Database Integrity that explicitly does not apply to any architecture A3



Drawing 8: Outsourcing agent connecting a user to Swift through a service provider

Security Controls Structure

Each security control in this document is structured into the following three parts:

- general control information
- control definition
- implementation guidance

General Control Information

- **Control Number and Title:** Each control has a unique number and title. If the control number is suffixed with an *A*, then this indicates that the control is *Advisory*.
- **Control Type:** This identifies the control as *Mandatory* or *Advisory*. Users must implement all applicable Mandatory controls, taking into account the architecture type. Advisory controls are considered as a security best practice and are strongly recommended for additional implementation.
- **Applicability to Architecture types**: Controls are applicable either to users with **Architecture types A1**, **A2**, **A3**, **A4**, **type B**, or a combination of types. As such, users with **Architecture type B** are not required to comply with controls applicable to **Architecture types A1**, **A2**, **A3** and **A4** only.

Control Definition

- **Control Objective**: The security goal to be achieved, irrespective of the implementation method.
- **In-scope Components**: The specific Swift-related components covered by this particular control. (For more information, see Scope of Security Controls. The Controls_Matrix document can also be consulted to have a view of the relevant controls per in-scope component.)

  Note: When extending the scope to new components, the new in-scope components can initially be tagged as *Advisory*[14].
- **Risk Drivers**: The specific risks addressed by this particular control. A full matrix of risks is documented in Appendix A.

Implementation Guidance

- **Control Statement**: The suggested means by which the Control Objective can be fulfilled.
- **Control Context:** Additional introductory background information about this control.
- **Implementation Guidelines**: The Swift-formulated method for control implementation.

Important Users must attest against their compliance with all mandatory control objectives. Additional details about implementation options for compliance are described in the next section. Users can also find additional valuable information in the CSP FAQ (Swift Knowledge Centre article 5021823) and the Security Guidance Document (log in on swift.com required).

[14] The Change Management process makes sure that the Swift community has sufficient time to understand and implement any future changes to the control requirements. Typically, new mandatory controls will be first introduced as advisory, thereby giving all users at least two cycles to plan, budget and implement.

Security Controls Compliance

As per the above-described security controls structure, the objective of a control states the security goal to be achieved irrespective of the implementation method used.

To comply with a CSP security control, users must implement a solution that meets the control definition; namely, the solution:

- meets the stated control objective
- covers the documented in-scope components relevant for the user's architecture
- addresses the risk drivers (see Appendix A for a risk matrix and Appendix C for illustrations of such risks)

The *Control Statement* is the suggested means to fulfil the control objective and the *Implementation Guidelines* are common methods for implementing the control.

Compliance can be obtained by either of the following methods:

A) Implement a solution aligned with the implementation guidance provided in this document.

The *implementation guidance* section should not be considered as a strict audit checklist because each user's implementation can vary. Therefore, in the case that some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be taken into account to properly assess the overall guideline's adherence level.

B) Implement an alternative solution to the Swift-formulated implementation guidance, which equally meets the control definition.

In such a case, deployed controls, their effectiveness, and particular environment specificities must be taken into account to properly assess the control definition compliance of the solution (risk assessment approach).

Both methods are considered as valid and equally robust from a risk perspective.

Users are ultimately responsible for assessing the suitability of Swift-formulated implementation guidance in their environment or determining if they want to adopt alternative implementation solutions.

It is the expectation that only a small subset of users (typically those with a high level of Information Security Risk Management maturity within their organisation) will consider alternative implementation methods for one or more controls to cope with large or complex configurations.

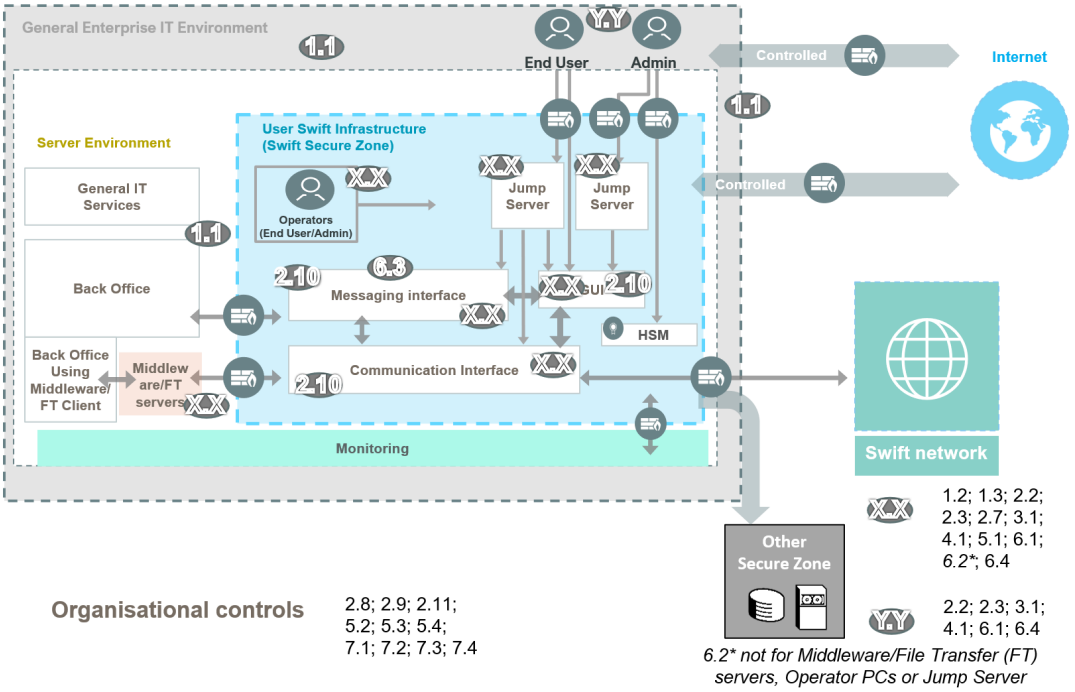
Security Controls Summary Table

The following table provides an overview of all mandatory and advisory security controls, structured according to the principle they support and with reference to the architecture type to which they relate. In addition, the table identifies the relevance of the controls, depending on the architecture type. Advisory controls are notated with an *A* after the control number (for example, *2.4A*) throughout this document, and are shaded in the table below. Likewise, individual shaded cells in the table below are advisory for some specific architecture types even when the control is mandatory for other architecture types.

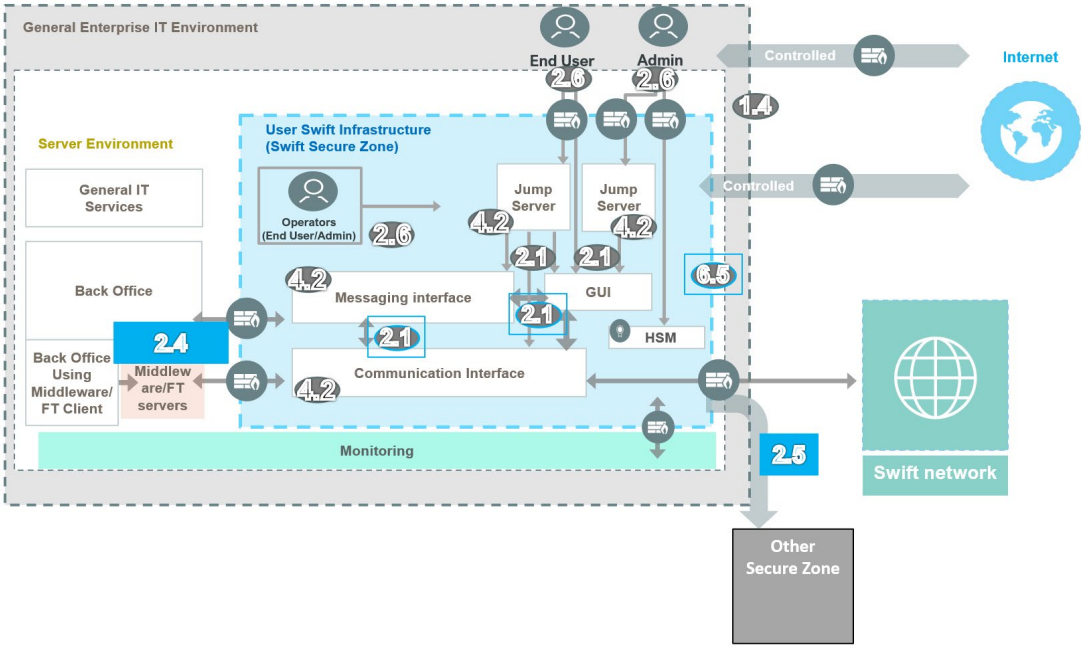
Mandatory and Advisory Security Controls	Architecture Type				
	A1	A2	A3	A4	B
1 Restrict Internet Access and Protect Critical Systems from General IT Environment					
1.1 Swift Environment Protection	•	•	•		
1.2 Operating System Privileged Account Control	•	•	•	•	•
1.3 Virtualisation or Cloud Platform Protection	•	•	•	•	
1.4 Restriction of Internet Access	•	•	•	•	•
1.5 Customer Environment Protection				•	
2 Reduce Attack Surface and Vulnerabilities					
2.1 Internal Data Flow Security	•	•	•		
2.2 Security Updates	•	•	•	•	•
2.3 System Hardening	•	•	•	•	•
2.4A Back Office Data Flow Security	•	•	•	•	•
2.5A External Transmission Data Protection	•	•	•	•	
2.6 Operator Session Confidentiality and Integrity	•	•	•	•	•
2.7 Vulnerability Scanning	•	•	•	•	•
2.8 Outsourced Critical Activity Protection	•	•	•	•	•
2.9 Transaction Business Controls	•	•	•	•	•
2.10 Application Hardening	•	•	•		
2.11A RMA Business Controls	•	•	•	•	•
3 Physically Secure the Environment					
3.1 Physical Security	•	•	•	•	•
4 Prevent Compromise of Credentials					
4.1 Password Policy	•	•	•	•	•
4.2 Multi-Factor Authentication	•	•	•	•	•
5 Manage Identities and Separate Privileges					
5.1 Logical Access Control	•	•	•	•	•
5.2 Token Management	•	•	•	•	•
5.3A Staff Screening Process	•	•	•	•	•
5.4 Physical and Logical Password Storage Protection	•	•	•	•	•
6 Detect Anomalous Activity to Systems or Transaction Records					
6.1 Malware Protection	•	•	•	•	•
6.2 Software Integrity	•	•	•	•	
6.3 Database Integrity	•	•		•	
6.4 Logging and Monitoring	•	•	•	•	•
6.5A Intrusion Detection	•	•	•	•	
7 Plan for Incident Response and Information Sharing					
7.1 Cyber Incident Response Planning	•	•	•	•	•
7.2 Security Training and Awareness	•	•	•	•	•
7.3A Penetration Testing	•	•	•	•	•
7.4A Scenario-based Risk Assessment	•	•	•	•	•

The following two drawings present visually where the controls would apply, using, for reference, one of many ways an **Architecture A1** could be designed. (See *Appendix B* for other reference architectures. The Controls_Matrix document can also be consulted to have a view of the relevant controls per in-scope component.)

Drawing 9 shows the controls applicable at the infrastructure and systems levels and with organisational controls surrounding such environment. Drawing 10 focus on interactive and application flow controls between the Swift-related components and the operator PCs or back-office systems.



Drawing 9: Infrastructure static and organisational controls



Drawing 10: Human/Application to Machine/application flow controls

Detailed Control Descriptions

1 Restrict Internet Access and Protect Critical Systems from General IT Environment

1.1 Swift Environment Protection

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		●	●	●		

Control Definition

Control Objective: Ensure the protection of the user's Swift infrastructure from potentially compromised elements of the general IT environment and external environment.

In-scope components:

- messaging interface
- communication interface
- GUI
- SWIFTNet Link
- Hardware Security Module (HSM)
- Swift or customer connector
- jump server
- dedicated and general-purpose operator PCs

Risk Drivers:

- compromise of enterprise authentication system
- compromise of user credentials
- credential replay
- exposure to internet-based attacks
- unauthorised access

Implementation Guidance

Control Statement:

A separated secure zone safeguards the user's Swift infrastructure from compromises and attacks on the broader enterprise and external environments.

Control Context:

Segmentation between the user's Swift infrastructure and the larger enterprise network reduces the attack surface and has shown to be an effective way to defend against cyber-attacks that commonly involve a compromise of the general enterprise IT environment. Effective segmentation includes network-level separation, access restrictions, and connectivity restrictions.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

a) Overall design goals for implementing environment separation

- Implement a “secure zone” to separate and protect the user’s Swift infrastructure from the compromise of systems and services located outside of the secure zone.
- To the fullest extent possible, passwords and other authenticators that are usable inside the secure zone (especially for privileged accounts) are not stored or used in any form (hashed, encrypted, or plain text) in systems outside of the secure zone. This does not apply to encrypted back-up files. If the authentication services system resides outside of the Swift secure zone, then:
  - Either the system is in another existing secure zone that has similar controls,
  - or the system is only used to filter the connections to the Swift infrastructure component (controlling the connectivity at the boundary of the secure zone). In such a case, logical access to the Swift infrastructure component is ensured by another authentication mechanism residing in the secure zone (another IAM or the accessed component itself).
- The secure zone is appropriately scoped to each user's environment, including the potential reuse of existing secure zones (for example, a secure “production environment”, “back-office environment”, or “payment systems zone”) to include the user’s Swift infrastructure.
- The components within the secure zone are all protected to the same or an equivalent level of security, access control, and trust and may communicate freely within the zone. Primary purpose of a secure zone is to host Swift-related components but can also include non-Swift related systems which then also need to be adequately protected by applying controls applicable to the Swift-related components (see the CSP FAQ for the relevant controls).
- Appendix B contains illustrative architecture diagrams that show samples of the methods a secure zone can be designed.

b) Scope of the secure zone

- The secure zone contains, but is not limited to, all components of the user’s Swift infrastructure. This includes the messaging interface, the communication interface, the browser-based GUI, the SWIFTNet Link, the Hardware Security Module (HSM), the Swift or customer connector, the jump server (see details below), and any applicable operator PCs solely dedicated to the operation or administration of the user’s Swift infrastructure.
  - General-purpose operator PCs are not included in the secure zone.
  - Dedicated operator PCs with Swift-related software installed (that is, “thick client” GUI software) are located in the secure zone, or the software is installed only on the jump server to be accessed by the general-purpose operator PCs outside of the secure zone.
  - Back-office and middleware servers used for data exchange with back-office systems are not necessarily included in the secure zone, but may be considered for inclusion depending on the chosen size and scope of the secure zone.
  - Test systems are not considered in scope of the security controls as long as (i) they are fully separated from production or live environment (including separate HSMs) and (ii) they are configured to only support test traffic (for example, by only using lite certificates on test-only logical terminals). If the test systems are not fully separated or can be configured for live traffic, then users must take the test systems in scope and make sure that the same security controls are applied as for production or live systems. Development systems are not within the secure zone and are not connected to the Swift network.
  - The Alliance Connect SRX VPN boxes or the Alliance Connect Virtual VPN instances (hosting systems or machines) are in a secure environment with appropriate physical controls (aligned with control 3.1).
- The secure zone size and scope are defined in a way that is most appropriate to the user’s environment. Options may include, but are not limited to the following:
  - A Swift secure zone dedicated only for the user’s Swift infrastructure.
  - An expansion of an existing secure area (for example, a secure “production environment” or “payment systems zone”) to include the user’s Swift infrastructure. The size and scope of this zone may vary significantly depending on the existing environment.
- Software, systems, and services within the secure zone are assessed for need and removed from the zone if not supporting the operations or security of the zone (for example, assess the need for e-mail access).

c) Protection of the secure zone

Boundary Protection

- Transport layer stateful firewalls are used to create logical separations at the boundary of the secure zone.
  - Transport layer firewalls creating the secure zone boundary should be physically or virtually dedicated to the protection of the secure zone. If a firewall is shared to separate other zones, then care must be taken for the firewall management to make sure that compromises of the firewall do not affect the protection of the secure zone.
  - Access control lists (ACLs) and application firewalls may be used to provide additional protection for the secure zone but are not sufficient on their own.
- Layer 2 devices (data link layer, such as switches) may be shared between the secure zone and other uses (VLAN separation).
- Administrative access to networking devices is protected using either an out-of-band network or through controlled in-band access (for example, a management VLAN). Administrative access to the firewalls that protect the secure zone does not rely on the enterprise user authentication system, but a system located within an existing secure zone that has similar controls as the Swift secure zone.
- Inbound and outbound connectivity for the secure zone is limited to the fullest extent possible. A process is implemented to analyse, review, and enforce the firewall rules governing the connectivity.
  - No “allow any” firewall rules are implemented, and network flows are explicitly authorised (allowlisting approach). To achieve this, a general enterprise server might initially be used to filter legitimate connectivity access towards the secure zone without losing traceability of such connections.
  - Generally, connectivity crossing the secure zone boundary is restricted to bi-directional communications with back-office applications and MV-SIPN[15] or the Internet, inbound communications from approved general-purpose operator PCs to the jump server, and outbound administration data (data logging, back-ups).
  - Firewall rules are reviewed annually, at least.
  - Connections through the boundary firewalls are logged in line with or as part of the control 6.4.

d) Access to the secure zone systems

d.1 Local Operator (end user and administrator) Access

- The secure zone has implemented one of the following designs for restricting operator access (interactive or command-line sessions) into the secure zone:
  - Operators connect from dedicated operator PCs located within the secure zone (that is, PCs located within the secure zone, and used only for secure zone purposes).
  - Operators connect from their general-purpose operator PC to the secure zone through a jump server (for example, using a Citrix-type solution or Microsoft Terminal Server) located within the Swift secure zone or within another existing secure zone that has similar controls.
    As the entry point into the secure zone, the jump server implements strong security practices, including the following:
    - Make sure all in-scope security controls in this document are implemented (for example security updates, system hardening).
    - Separate jump server for system administrators (with multi-factor authentication) and end users. As an alternative to separate jump servers, only allow temporary access to system administrators with effective approval processes and session activity recording.
    - Restrict access to authorised operators only.
    - Remove any unnecessary software.
    - Restrict risky activity (for example, sending or receiving e-mails).
    - Enable logging in line with or as part of the control 6.4.
  - Operators connect from their general-purpose operator PC and only access the messaging or communication interface with a browser-based GUI (for example, Alliance Web Platform). The following specific security controls apply to this set-up:
    - The browser-based GUI is located in the secure zone and is, at network level, logically separated from the messaging and communication interface.
    - Multi-factor authentication is implemented, where appropriate (on the browser-based GUI, on the messaging interface, or on the communication interface).
    - This set-up cannot be used for operating system administration activities.

[15] Multi-Vendor Secure IP Network

- Swift systems within the secure zone restrict administrative access to only expected ports, protocols, and originating IPs.

d.2 Remote Operator Access (teleworking, "on-call" duties, or remote administration)

- Remote access to the secure zone from outside of the user's network first requires VPN authentication (recommended with multi-factor authentication) to the user's network before accessing the secure zone through the same secured channels as local operators.
- A risk assessment is performed by the user to consider appropriate security controls to be implemented for remote access, such as the use of a virtual desktop infrastructure, dedicated channels for connectivity (for example, dedicated jump servers for remote users, leased lines).

e) Separation from General Enterprise IT Services

- To protect the secure zone from credential theft or a compromise of enterprise authentication (LDAP, RADIUS, Identity Provider, multi-factor) services, or a combination of both, secure zone systems use a separate authentication system from the general enterprise authentication service. For example, secure zone systems are not a member of the corporate directory service, but are instead members of a secure zone directory service.
- Supporting IT infrastructure, such as asset management, databases, data storage, security services (for example, patching), and networking services (for example, DNS, NTP) used within the secure zone is protected from credential compromise within the larger enterprise. Institutions must conduct an analysis of connectivity points which make sure that these systems do not store authenticators (passwords, tokens, and other methods) for systems and accounts in scope in any format (hashed, encrypted, plain text) outside of the secure zone or another existing secure zone that has similar controls. The supporting IT infrastructure should not be exclusive to Swift systems and may be shared within the secure zones.

Optional Enhancement:

- Restrict (through additional separation) the communication between components of the secure zone considering the following:
  - Network ACLs or host-based firewalls that restrict traffic on a host-by-host basis within the secure zone.
  - Individual hardware or network-based firewalls between the components in the secure zone can optionally be used.

Considerations for alternative implementations:

Users with a high level of security programme maturity within the organisation might consider implementing alternative solutions such as those suggested below or others. The alternative solutions must be risk-appropriate to each environment, and must consider the effort required to effectively implement, manage, and maintain the solution.

Potential cases when alternative solution could be envisaged:

- When the secure zone authentication service is not separated from the enterprise authentication service, then it requires implementing (or having implemented) a comprehensive set of defence-in-depth controls to protect from and detect adversaries that cross the secure zone boundary. The set of controls must consider following elements:
  - locating the authentication service within an existing secure zone with similar controls as those applicable to the Swift secure zone,
  - limiting trust relationships between the larger enterprise environment and the secure zone (such as one-way trust relationships),
  - restricting operator and administrative access by implementing strong privileged access controls and read-only access where feasible,
  - enabling verbose logging, and implementing centralised active monitoring and detective capabilities.
- When general enterprise IT services (for example, vulnerability scanning and boundary firewall management) are shared between the secure zone and other environments, then any credentials used across the environment should be monitored to make sure they are only used when and where expected.
- When a general enterprise server is initially used to reach the secure zone, then that server is only used to filter legitimate connectivity access (as a concentrator or gateway to ease access filtering to the secure zone). Identity and access management to the secure zone components or the jump server (or both) still rely on authentication services that reside within the Swift secure zone or another existing secure zone that has similar controls.

- When the secure zone has dependencies on enterprise shared functions (such as directory services, servers, or networks) that are outside the scope, then the user must make sure that any compromise of such functions will not compromise the security of the in-scope components.

1.2 Operating System Privileged Account Control

Control Type: Mandatory / Advisory for B	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Restrict and control the allocation and usage of administrator-level operating system accounts.

In-scope components:

Administrator-level accounts defined on the following components:

- systems or virtual machines (VMs) hosting a Swift-related component (including interface, GUI, Swift or customer connector, jump server)
- dedicated operator PCs
- network devices protecting the secure zone
- on-premises or remote (that is hosted or operated by a third party, or both) virtualisation or cloud platform that hosts Swift-related VMs
- New HSM
- [Advisory: bridging servers (such as middleware or file transfer servers other than customer connectors) used for and guardian of the secure data exchange between back-office and Swift-related components]
- [Advisory: General-purpose operator PCs]

Risk Drivers:

- deletion of logs and forensic evidence
- excess privilege or access
- lack of traceability
- unauthorised system changes
- HSM management misused

Implementation Guidance

Control Statement:

Access to administrator-level operating system accounts is restricted to the maximum extent possible. Usage is controlled, monitored, and only permitted for relevant activities such as software installation and configuration, maintenance, and emergency activities. At all other times, an account with the least privilege access is used.

Control Context:

Tightly protecting administrator-level accounts within the operating system reduces the opportunity for an attacker to use the privileges of the account as part of an attack (for example, executing commands or deleting evidence).

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- Examples of administrator-level accounts are defined as follows:
  - Windows: built-in administrator account and members of groups with administrator privileges (for example, accounts with debug or file system privileges). Typically, Enterprise Admins group, Domain Admins group, and Local Administrator group.
  - Linux/Unix: root account (User ID = 0) and members of the root group.

- Mainframe: system administrator or system programmer role.
- Network devices: accounts like admin, root, telco, su or cisco.
- New HSM: admin accounts.
- Virtualisation or cloud platform: administrator, super admin, root user or the like accounts.

- Access to administrator-level operating system accounts is restricted to the maximum extent possible unless needed to install, configure, maintain, operate, or support emergency activities. The use of the administrator-level account is limited to the duration of the activity (for example, maintenance windows).
- Logins with built-in administrator-level accounts are not permitted, except to perform activities where such accounts are specifically needed (for example, system configuration) or in emergency situations (break-glass account). Individual accounts with administrator-level privileges or accounts with the ability to escalate to administrative access, (like “sudo”) are used instead.
- Individual administrator-level account access and usage are logged, in line with or as part of the control 6.4, so that activities can be reconstructed to help determine the root-cause of incidents.
- Administrator-level passwords are tightly controlled with physical access controls when physically recorded.

Optional Enhancements:

- Systems are configured to disallow logins of built-in administrator-level accounts, except through a maintenance mode (for example, single user mode or safe mode). This effectively prohibits logins to the account as a service, batch job, through remote desktop services, or by escalating privileges from another account.
- Apply the control to all bridging servers (such as middleware or file transfer servers other than customer connectors) used for data exchange between back-office and Swift-related components.

Considerations for alternative implementations:

New models are emerging to enhance the user experience but also availability as observed with the pandemic. Alternative implementations are raising to give users flexible access to the institutions’ environment: not necessarily through fully managed devices[16] but incorporating also individuals own devices to reach resources located on premises or in the cloud. That implies moving from a controlled on-premises environment (for which the CSCF mainly provides guidance) to a zero-trust environment requiring to assess and control appropriately each type of access.

Such alternatives have to be considered individually and specifically by users from a risk-based point of view taking into consideration potential risks if some elements are compromised. Those alternatives cannot be described here but would require to consider elements such as those identified below for a secure virtual desktop infrastructure:

- Defining a virtual desktop infrastructure (Citrix or other workspace) with OS privileged account managed centrally and not possibly activated or used by the end users can meet the control (considering the virtual infrastructure is protected in line with control 1.3 and the virtual desktops themselves are protected as a physical general-purpose operator PC in line with control such as 2.2, 2.3, 2.7, 4.1, 5.1; 6.1; 6.4, 6.5A).
- The risks of end-user device compromise must be considered, analysed and appropriate controls deployed to protect the virtual desktop infrastructure and further accessed resources. Those controls must ensure proper authentication, activities authorisation (requesting sometimes additional independent factors) but also appropriate prompt reaction, also involving the end users, in case of end-user device compromise, loss or theft to block potential accesses through such device.
- Confidentiality and integrity of the sessions established towards the virtual infrastructure must also be ensured in line with the standard operator session depicted in control 2.6.

[16] Fully managed device is a company-owned device with features that give IT admins control of the device settings and policy configuration, including potential remote-wipe in case of theft/loss. At the opposite of the spectrum, there is usage of unmanaged (by a company) individuals own device to access the company resources, network or applications hosted in the cloud or by a service provider. Intermediary models can exist where managed (sand-boxed) application(s) (by a company) are deployed on users own device (potentially with the capability to remote-wipe those applications in case of device theft/loss). All those models have to be properly analysed for proper and secure usage.

1.3 Virtualisation or Cloud Platform Protection

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	

Control Definition

Control Objective: Secure the virtualisation or cloud platform and virtual machines (VMs) that host Swift-related components to the same level as physical systems.

In-scope components:

- on-premises or remote (that is hosted or operated by a third party, such as a cloud provider, or both) virtualisation or cloud platform and VMs used to host any of the following Swift-related components:
  - messaging interface
  - communication interface
  - GUI
  - SWIFTNet Link
  - Swift and customer connector
  - jump server
  - dedicated and general-purpose operator PCs
  - firewalls
  - Alliance Connect Virtual VPN instance
  - [Advisory: bridging servers (such as middleware or file transfer servers other than customer connectors) used for and guardian of the secure data exchange between back-office and Swift-related components]

***Note**:* This requirement is not applicable when there is no on-premises and remote virtualisation or cloud platform and no VMs used to host the referred Swift-related components.

Note: It is reminded that containerised applications have to be considered as co-hosted on the same system that hosts the container itself.

Risk Drivers:

- unauthorised access
- uncontrolled proliferation of systems and data

Implementation Guidance

Control Statement:

Secure the virtualisation or cloud platform, virtualised machines, and the supporting virtual infrastructure (such as firewalls) to the same level as physical systems.

Control Context:

Security controls that apply to non-virtualised (physical) systems are equally applicable to virtual systems. The additional virtualisation layer needs extra attention from a security perspective. The uncontrolled proliferation of VMs could lead to unaccounted machines with the risk of unmanaged, unpatched systems open to unauthorised access to data.

If appropriate controls have been implemented to this underlying layer, then Swift does not limit the use of virtual technology for any component of the user's Swift infrastructure or the associated supporting infrastructure (for example, virtual firewalls).

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be

considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- The same security requirements apply to the virtualisation or cloud platform, virtual machines, and supporting virtual infrastructure as for all other infrastructure systems and components. Those security requirements cover, for example, the location in an existing secure zone that has similar controls as those applicable to the Swift or customer secure zone, privileged access restrictions, login and password policies, installation of security updates, and restriction of internet access. Those controls have the virtualisation or cloud platform identified in the *In-scope Components* section.
- Vulnerability scanning is performed on Swift-related VMs and, when technically possible, on the virtualisation or cloud platform.
- The virtualisation or cloud platform systems are subject to physical protection, which prevents unauthorised physical access.
- VM isolation is ensured on the virtualisation or cloud platform to prevent the lateral move out of a virtual machine to access or interact with other VMs (or the underlying hypervisor) or to bypass normal network controls that filter or inspect connections to the Swift environment (or a combination of both).
    - Filtering and expected inspections of the network flows that reach the Swift-related VMs are performed preferably using resources (such as firewalls, packet inspections, or content filtering) external to the virtualisation platform or must be enforced at the hypervisor level.
    - If isolation is ensured on the virtualisation or cloud platform, then the hosted VMs can maintain their security classification and can be individually secured accordingly (as such, they do not inherit the classification of the Swift-related VMs and are not subject to all Swift-related controls).
- When multi-factor authentication is implemented for interactive access to the Swift-related VM operating systems (and in-line with control 4.2) to also prevent direct access to those VMs from the hypervisor layer, then multi-factor authentication is not mandatory at the virtualisation platform management level.
- When relying on a third party for the underlying virtualisation or cloud platform, the user must engage with the third party to obtain reasonable comfort that the control objective is met.

Optional Enhancement:

- Apply the control to all bridging servers (such as middleware or file transfer servers other than customer connectors) used for data exchange between back-office and Swift-related components.

1.4 Restriction of Internet Access

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Control/Protect Internet access from operator PCs and systems within the secure zone.

In-scope components:

- dedicated and general-purpose operator PCs
- jump server
- messaging interface
- communication interface
- GUI
- SWIFTNet Link
- Swift and customer connector
- New HSM
- [Advisory: bridging servers (such as middleware or file transfer servers other than customer connectors) used for and guardian of the secure data exchange between back-office and Swift-related components]
- [Advisory: on-premises or remote (that is hosted or operated by a third party, or both) virtualisation or cloud platform and their management PCs]

Risk Drivers:

- exposure to internet-based attacks

Implementation Guidance

Control Statement:

All general-purpose and dedicated operator PCs, as well as systems within the secure zone, have controlled direct internet access in line with business[17].

Control Context:

Direct access to the Internet raises exposure to internet-based attacks. Risk is even higher in case of human interactions (browsing, e-mails, or other social network activities being permitted). Once compromised, those systems can be an entry point that allows lateral movements or injection of command and control elements (or a combination of both).

If reducing the attack surface and vulnerabilities of those systems (as per the relevant controls identified in this document) is primordial, then limiting and controlling direct Internet accesses is crucial.

On top of (general) operator PCs that connect Swift-related services or applications offered by outsourcing agents or service providers (such as Swift in the case of Alliance Lite2 or Alliance Cloud, a Service Bureau, a Business Connect or an L2BA provider), due diligence must be taken to secure (general) operator PCs used to access on-premises interfaces or GUIs. Insecurely combining access to the “production environment” and the Internet could be abused by attackers.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user’s implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be

[17] Purpose is not to prohibit internet access but to limit/control connectivity where it is relevant for business related reasons (such as to access external service provider or outsourcing agent resources).

considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

a) Internet access from the secure zone

- General-purpose internet browsing (including e-mail activities) from systems within the Swift or customer secure zone is not permitted.
- Internet access from systems within the secure zone (for example, dedicated operator PCs or other Swift-related components) is highly restricted and ideally should be blocked.
  - When possible, activities that require the Internet are conducted outside of the secure zone. Example activities may include conducting daily business on swift.com, or downloading security updates for secure transfer into the secure zone.
  - If internet access is needed from within the secure zone, then access should be granted only to allowlisted URL destinations through a proxy with content inspection and adequate blocking or filtering controls. Connections are only permissible if they are initiated in the outbound direction.
- As the entry point into the secure zone, the jump server (located within the secure zone or another existing secure zone that has similar controls) does not have internet access.

b) Internet access from general-purpose operator PCs

- Control internet access provided on the general-purpose operator PCs used with the following purposes:
  - Connect to an application at the service provider or outsourcing agent (user-to-application) to process financial transactions[18].
  - Access a messaging or communication interface through a browser-based GUI (for example, Alliance Web Platform).

  Control access through one of the following options:
    - internet access through a remote desktop or virtual machine solution
    - internet access from the general-purpose operator PC to only allowlisted URL destinations through a proxy with content inspection, in combination with adequate blocking or filtering controls and permitting only outbound initiated connections
    - internet access from the general-purpose operator PC through a Web Gateway (with content inspection, in combination with blocking or filtering controls) using maintained denylisted URL destinations
- Even if Swift strongly recommends controlling the internet access, another method to meet the control objective on those PCs accessing the user's Swift infrastructure is to enforce the usage of a jump server that has no internet access combined with multi-factor authentication (in line with control 4.2) implemented on the individual Swift-related components/systems or at the jump server.

c) Internet access from other components (middleware servers or the virtualisation platform – Advisory)

- When used, internet access from the middleware server or the virtualisation platform underlying system (also referred to as the hypervisor) is highly restricted and ideally blocked.
  - When possible, activities that require the Internet are conducted from other systems. Examples of such activities include conducting daily business, or downloading security updates for secure transfer into the target system.
- If internet access is needed from those systems, then access should be granted only to allowlisted URL destinations through a proxy with content inspection and adequate blocking or filtering controls. Connections are only permissible if they are initiated in the outbound direction.

Optional Enhancement:

- Apply the control to all bridging servers (such as middleware or file transfer servers other than customer connectors) used for data exchange between back-office and Swift-related components.

[18] Such as , creating, submitting, approving or modifying messaging transactions or updating entitlements. Read-only/queries kind of access can be waived if entitlements cannot be changed from such operator PC's.

1.5 Customer Environment Protection

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
					•	

Control Definition

Control Objective: Ensure the protection of the customer's connectivity infrastructure from external environment and potentially compromised elements of the general IT environment.

In-scope components:

- customer connector
- dedicated and general-purpose operator PCs
- jump server
- HSM when used with customer connector

Risk Drivers:

- compromise of enterprise authentication system
- compromise of user credentials
- credential replay
- exposure to internet-based attacks
- unauthorised access

Implementation Guidance

***Note**:* This control focuses on the customer connector and expects separation between (i) operational (or production) environment where the customer connector resides and (ii) the wider or general IT environment.

Control Statement:

A separated secure zone safeguards the customer's infrastructure used for external connectivity from external environments and compromises or attacks on the broader enterprise environment.

Control Context:

Segmentation between the customer's connectivity infrastructure and its larger enterprise network reduces the attack surface and has shown to be an effective way to defend against cyber-attacks that commonly involve compromise of the general enterprise IT environment. Effective segmentation will include network-level separation, access restrictions, and connectivity restrictions.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

a) Overall design goals for implementing environment separation

- Implement a secure zone (or restricted operational zone) to separate and protect the customer connectivity infrastructure from the compromise of systems and services located outside of the secure zone.
- To the fullest extent possible, passwords and other authenticators that are usable inside the secure zone (especially for privileged accounts) are not stored or used in systems outside of the secure zone. This does not apply to encrypted back-up files.
- If the authentication services system resides outside of the secure zone, then:

  - Either the system is in another existing secure zone that has similar controls,
  - or the system is only used to filter the connections to the customer's connectivity infrastructure (controlling the accesses at the boundary of the secure zone). In such a case, logical access to the component is ensured by another authentication mechanism residing in the secure zone (by another IAM or is performed by the accessed component itself).
- The secure zone is appropriately scoped to each user's environment, including the potential reuse of existing secure zones (for example, a secure "production environment", "back-office environment", or "payment systems zone") to include the connectivity infrastructure.
- The components within the secure zone are all protected to the same or an equivalent level of security and trust and those components may communicate freely within the zone. Primary purpose of a secure zone is to host Swift-related components but can also include non-Swift related systems which then also need to be adequately protected by applying controls applicable to the Swift-related components (see the CSP FAQ for the relevant controls).

b) Scope of the secure zone

- The secure zone contains, but is not limited to, all components of the connectivity infrastructure. This includes the Hardware Security Module (when relevant), the customer connector, the jump server (see details below), and any applicable operator PCs solely dedicated to the operation or administration of the connectivity infrastructure. As such:
  - General-purpose operator PCs are not included in the secure zone.
  - Dedicated operator PCs with "thick client" GUI software are located in the secure zone, or the software is installed on the jump server and accessed by the general- purpose operator PCs residing outside of the secure zone.
  - Back-office systems are not necessarily included in the secure zone, but may be considered for inclusion depending on the chosen size and scope of the secure zone.
  - Test systems are not considered in scope of the security controls as long as (i) they are fully separated from production or live environment (including separate HSMs, when used) and (ii) they are configured to only support test traffic (for example, by only using lite certificates on test only logical terminals). If the test systems are not fully separated or can be configured for live traffic, then users must take the test systems in scope and make sure that the same security controls are applied as for production or live systems. Development systems are not within the secure zone and are not connected to the Swift network.
  - When used, the Alliance Connect SRX VPN boxes or the Alliance Connect Virtual VPN instances (hosting systems or machines) are in a secure environment with appropriate physical controls (aligned with control 3.1).
- The secure zone size and scope are defined in a way that is most appropriate to the user's environment. Options may include, but are not limited to the following:
  - A specific secure zone dedicated only for the connectivity infrastructure.
  - An expansion of an existing secure area (for example, a secure "production environment" or "payment systems zone") to include the connectivity infrastructure. The size and scope of this zone may vary significantly depending on the existing environment.
- Software, systems, and services within the secure zone are assessed for need and removed from the zone if not supporting the operations or security of the zone (for example, assess the need for e-mail access).

c) Protection of the secure zone

Boundary Protection

- Transport layer stateful firewalls are used to create logical separations at the boundary of the secure zone.
  - Transport layer firewalls creating the secure zone boundary should be physically or virtually dedicated to the protection of the secure zone. If a firewall is shared to separate other zones, then care must be taken for the firewall management to make sure that compromises of the firewall do not affect the protection of the secure zone.
  - Access control lists (ACLs) and application firewalls may be used to provide additional protection for the secure zone, but are not sufficient alone.
- Layer 2 devices (data link layer, such as switches) may be shared between the secure zone and other uses (VLAN separation).

- Administrative access to networking devices is protected using either an out-of-band network or through controlled in-band access (for example, a management VLAN). Administrative access to the firewalls that protect the secure zone does not rely on the enterprise user authentication system, but a system located within an existing secure zone that has similar controls.
- Inbound and outbound connectivity for the secure zone is limited to the fullest extent possible. A process is implemented to analyse, review, and enforce the firewall rules governing the connectivity.
  - No “allow any” firewall rules are implemented, and network flows are explicitly authorised (allowlisting approach). To achieve this, a general enterprise server might initially be used to filter legitimate connectivity access towards the secure zone without losing traceability of such connections.
  - Generally, connectivity crossing the secure zone boundary is restricted to bi-directional communications with back-office applications and MV-SIPN[19] or the Internet, inbound communications from approved general-purpose operator PCs to the jump server, and outbound administration data (data logging, back-ups).
  - Firewall rules are reviewed annually, at least.
  - Connections through the boundary firewalls are logged in line with or as part of the control 6.4.

d) Access to the secure zone systems

d.1 Local Operator (end user and administrator) Access

- The secure zone has implemented one of the following designs for restricting operator access (interactive or command-line sessions) into the secure zone:
  - Operators connect from dedicated operator PCs located within the secure zone (that is, PCs located within the secure zone, and used only for secure zone purposes).
  - Operators connect from their general-purpose operator PC to the secure zone through a jump server (for example, using a Citrix-type solution or Microsoft Terminal Server) located within the secure zone or within another existing secure zone that has similar controls.
    As the entry point into the secure zone, the jump server implements strong security practices, including the following:
    - Make sure all in-scope security controls in this document are implemented (for example security updates, system hardening).
    - Separate jump server for system administrators (with multi-factor authentication) and end users. As an alternative to separate jump servers, only allow temporary access to system administrators with effective approval processes and session activity recording.
    - Restrict access to authorised operators only.
    - Remove any unnecessary software.
    - Restrict risky activity (for example, sending or receiving e-mails).
    - Enable logging in line with or as part of the control 6.4.
  - Operators connect from their general-purpose operator PC and only access the customer connector offering interactive access with a browser-based GUI. The following specific security controls apply to this set-up:
    - The browser-based GUI is located in the secure zone and is ideally, at network level, logically separated from the customer connector.
    - Multi-factor authentication is implemented, where appropriate (on the browser-based GUI or on the customer connector).
    - This set-up cannot be used for operating system administrative activities.
- Customer connectivity systems within the secure zone restrict administrative access to only expected ports, protocols, and originating IPs.

d.2 Remote Operator Access (teleworking, “on-call” duties, or remote administration)

- Remote access to the secure zone from outside of the user network first requires VPN authentication (recommended with multi-factor authentication) to the network before accessing the secure zone through the same secured channels as local operators.
- A risk assessment is performed by the user to consider additional security controls to be implemented for remote access, such as the use of a virtual desktop infrastructure, dedicated channels for connectivity (for example, dedicated jump servers for remote users, leased lines).

[19] Multi-Vendor Secure IP Network

e) Separation from General Enterprise IT Services

- To protect the secure zone from credential theft or a compromise of enterprise authentication (LDAP, RADIUS, Identity Provider, multi-factor) services, or a combination of both, secure zone systems use a separate authentication system from the general enterprise authentication service. For example, secure zone systems are not a member of the corporate directory service, but are instead members of a secure zone directory service.
- Supporting IT infrastructure, such as asset management, databases, data storage, security services (for example, patching), and networking services (for example, DNS, NTP) used within the secure zone is protected from credential compromise within the larger enterprise. Institutions must conduct an analysis of connectivity points which make sure that these systems do not store authenticators (passwords, tokens, and other methods) for systems and accounts in scope in any format (hashed, encrypted, plain text) outside of the secure zone or another existing secure zone that has similar controls. The supporting IT infrastructure should not be exclusive to Swift systems and may be shared within the secure zones.

Optional Enhancements:

- Restrict (through additional separation) the communication between components of the secure zone considering the following:
  - Network ACLs or host-based firewalls that restrict traffic on a host-by-host basis within the secure zone.
- Individual hardware or network-based firewalls between the components in the secure zone can optionally be used.

Considerations for alternative implementations:

Institutions with a high level of security programme maturity within the organisation might consider implementing alternative controls such as those suggested below or others. The alternative solutions must be risk-appropriate to each environment, and must consider the effort required to effectively implement, manage, and maintain the solution.

- Not separating secure zone authentication services from the enterprise authentication service will require implementing a comprehensive set of defence-in-depth controls to protect from and detect adversaries that cross the secure zone boundary. Controls may include locating the authentication service within an existing secure zone with similar controls as those applicable to the secure zone, limiting trust relationships between the larger enterprise environment and the secure zone (such as one-way trust relationships), restricting operator and administrative access, implementing strong privileged access controls, implementing read-only access where feasible, enabling verbose logging, and implementing centralised active monitoring and detective capabilities.
- If general enterprise IT services (for example, vulnerability scanning and boundary firewall management) are shared between the secure zone and other environments, then any credentials used across the environment should be monitored to make sure they are only used when and where expected.
- If a general enterprise server is initially used to reach the secure zone, then that server is only used to filter legitimate connectivity access (as a concentrator or gateway to ease access filtering to the secure zone). Identity and access management for secure zone components or the jump server (or both) still rely on authentication services that reside within the secure zone or another existing secure zone that has similar controls.
- If the secure zone has dependencies on enterprise shared functions (such as directory services, servers, or networks) that are outside the scope, then the user must make sure that any compromise of such functions will not compromise the security of the in-scope components.

2 Reduce Attack Surface and Vulnerabilities

2.1 Internal Data Flow Security

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		●	●	●		

Control Definition

Control Objective: Ensure the confidentiality, integrity, and authenticity of application data flows between 'user's Swift-related components.

In-scope components:

- On-premises or remote (hosted and or operated by a third party, or both) user's Swift infrastructure and related components (including new HSM)

Risk Drivers:

- loss of sensitive data confidentiality
- loss of sensitive data integrity
- unauthenticated system traffic
- unauthorised access

Implementation Guidance

Control Statement:

Confidentiality, integrity, and authentication mechanisms are implemented to protect Swift-related component-to-component or system-to-system data flows.

Control Context:

The protection of internal data flows safeguards against unintended disclosure, modification, and access of the data while in transit.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- All data flows between Swift-related components are protected using a secure mechanism (for example, by using Local Authentication (LAU) in combination with a confidentiality protection[20] or by using two-way TLS) to support the confidentiality, integrity, and mutual authentication of the data flows. This includes the following data flows:
  - RMA application to messaging interface
  - GUI to messaging interface
  - GUI to communication interface
  - messaging interface to communication interface

[20] Such as one-way TLS

- Secure protocols use current, commonly accepted cryptographic algorithms (for example, AES[21] and ECDHE[22]) with key lengths in line with the current best practices. For more information about cryptographic algorithms that support secure protocols, see Swift Knowledge Centre article 5021566.
- Credentials and private keys used, and usually stored, by the applications to secure the flows are protected (large spectrum of protection, from definition and usage of secure coding guidelines to usage of specific solutions, can be envisaged based on user's risk management).

[21] Advanced Encryption Standard

[22] Elliptic Curve Diffie-Hellman Ephemeral

2.2 Security Updates

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Minimise the occurrence of known technical vulnerabilities on operator PCs and within the user's Swift infrastructure by ensuring vendor support, applying mandatory software updates, and applying timely security updates aligned to the assessed risk.

In-scope components:

Hardware and all software running on the following systems, virtual machines, hosts, servers or devices:

- physical systems or virtual machines (VMs) hosting a Swift-related component (including interface, GUI, Swift or customer connector)
- dedicated and general-purpose operator PC
- jump server
- on-premises or remote (that is hosted or operated by a third party, or both) virtualisation or cloud platform hosting Swift-related VMs and their management PCs
- network devices protecting the secure zone
- New HSM
- [Advisory: bridging servers (such as middleware or file transfer servers other than customer connector) used for and guardian of the secure data exchange between back-office and Swift-related components]

Risk Drivers:

- exploitation of known security vulnerabilities

Implementation Guidance

Control Statement:

All hardware and software inside the secure zone and on operator PCs are within the support life cycle of the vendor, have been upgraded with mandatory software updates, and have had security updates promptly applied.

Control Context:

The closure of known security vulnerabilities is effective in reducing the various pathways that an attacker may use during an attack. A security update process that is comprehensive, repeatable, and implemented in a timely manner is necessary to continuously close these known vulnerabilities when security updates are available.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- Vendor support
  - All software (including operating systems) and hardware (including network devices) are within the actively supported product life-cycle window of the vendor (including extended support), if applicable.
  - Maintenance or licensing contracts are in place for access to updates, minor upgrades, and other critical maintenance functions.
- Mandatory software updates
  - Mandatory releases or updates that are applicable to a user's Swift component are installed within the deadline specified by the vendor.

- Application of security updates
  - A risk assessment process is in place to determine the most appropriate treatment of vendor security updates. Risk assessment considerations may include the vendor-reported criticality of the update, user exposure and vulnerability, mitigating controls, and operational impact.
  - User-defined deployment timelines are established for applying updates based on criticality, system type, and required update testing.
  - In the absence of established internal processes and timelines, Swift recommends the use of the Common Vulnerability Scoring System (CVSS) Version 3 as a guideline for criticality, with the following update deployment targets:
    - Critical (9.0+ score): applied within one month of release
    - High (7.0 – 8.9 score): applied within three months of release
    - Low / Medium (< 7.0 score): user defined

    Note: It is common practice that operating system security updates are automatically pushed and applied on the Operator PCs shortly after their publication by the provider.
- Source and integrity validation of software and security updates.
- Before applying the software and security updates, the legitimate source is validated and integrity checks (for example, checksum validation) are performed when technically possible.

Optional Enhancement:

- Apply the control to all bridging servers (such as middleware or file transfer servers other than customer connectors) used for data exchange between back-office and Swift-related components.

2.3 System Hardening

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Reduce the cyber-attack surface of Swift-related components by performing system hardening.

In-scope components:

- dedicated and general-purpose operator PC
- jump server
- systems (physical or VMs) hosting a Swift-related component (including interface, GUI, Swift and customer connectors)
- on-premises or remote (that is hosted or operated by a third party, or both) virtualisation or cloud platform (also referred to as the hypervisor) hosting Swift-related VMs and their management PCs
- network devices protecting the secure zone
- [Advisory: bridging servers (such as middleware servers other than customer connectors) used for data exchange between back-office and Swift-related components]

***Note**:* Swift HSMs are FIPS 140-2 Level 3 compliant with hardened underlying OS and are out of the scope of this control.

Risk Drivers:

- excess attack surface
- exploitation of insecure system configuration

Implementation Guidance

Control Statement:

Security hardening is conducted and maintained on all in-scope components.

Control Context:

System hardening applies the security concept of “least privilege” to a system by disabling features and services that are not required for normal system operations. This process reduces the system capabilities, features, and protocols that a malicious person may use during an attack.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user’s implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- All in-scope systems are hardened, considering one or more of the following:
  - vendor security configuration guidance
  - industry-standard security configuration guidance (for example,[23] CIS , DISA STIG, NIST)
  - a local or regulator’s standard security configuration, or controls set of the same rigour as the vendor or industry guidance

[23] Center for Internet Security; Defense Information Systems Agency - Security Technical Implementation Guide; National Institute of Standards and Technology

- The selected hardening configuration (set of rules) can be overruled by application-specific configuration requirements to maintain a proper operational state for Swift-related systems.
- At a minimum, the hardening process should do the following:
  - Change default passwords.
  - Disable or remove unnecessary user accounts.
  - Disable or restrict unnecessary services, ports, and protocols.
  - Remove unnecessary software.
  - Restrict physical ports (for example, USBs or serial bus) as appropriate, by disabling them to the maximum extent possible, while still supporting operations (for example, when USB tokens are required to authenticate users, for message operations or other operational tasks).
  - Set, when technically possible, auto-lock options (such as activating an operator PC screen saver requiring a login after an inactivity time-out or when turned to sleep mode). A 15-minute inactivity time-out is recommended.
  - Adjust any default configurations known to be vulnerable.

  The vendor and industry standards listed above can provide detailed guidance to accomplish these minimum targets.
- Deviations from the selected hardening configuration are documented along with justification for the deviation and potential mitigations applied.
- Systems are maintained secure, as follows:
  - by checking regularly (at least twice per year) the systems against the secure settings identified as per preceding guidance to take any relevant corrective actions
  - by regularly applying the identified secure settings to the systems.

Optional Enhancement:

- Systems within the secure zone implement application allowlisting on the operating system which allows only known and trusted applications to be executed.
- Apply the control to all bridging servers (such as middleware or file transfer servers other than customer connectors) used for data exchange between back-office and Swift-related components.

2.4A Back Office Data Flow Security

Control Type: Advisory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Ensure the confidentiality, integrity, and mutual authenticity of data flowing between on-premises or remote Swift infrastructure components and the back-office first hops they connect to.

In-scope components:

- Business transaction data exchanged between the on-premises or remote (that is hosted or operated by a third party, or both) Swift-related components (interfaces, GUI, connectors) and the back-office first hops they are connected to either directly or through one or several bridging servers (such as middleware or file transfer servers).

Appendix H must be consulted to apprehend and visualise the various elements and options.

Such exchanges can be protected

- **end-to-end** between Swift-related components in the secure zone and a back-office first hop, through:
  - a) a secure mechanism ensuring authentication, integrity and confidentiality at the data (or application) level, OR
  - b) when there is a direct (point to point) connectivity between the Swift-related component in the secure zone and the back-office first hop, a secure protocol or mechanism ensuring the authentication, integrity and confidentiality of the connection.
- **or transitively (at each leg or segment in the flow)**, when one or several servers bridge the secure zone with a back-office first hop and there is no end-to-end protection of the data, through the implementation of:
  - 1) a secure protocol or mechanism for the direct (point to point) exchange between the Swift-related component(s) in the secure zone and the bridging server, that ensures authentication, integrity and confidentiality of the data exchanged, AND
  - 2) the CSCF controls on the bridging server(s) that is/are guardian of the data exchange security, AND
  - 3) when relevant, a secure protocol or mechanism for the exchange between the bridging servers themselves that ensures authentication, integrity and confidentiality of the data exchanged, AND
  - 4) a secure protocol or mechanism for the direct (point to point) exchange between the back-office first hop and the bridging server that ensures authentication, integrity and confidentiality of the data exchanged.

***Note**:* Swift expects this control to be turned Mandatory gradually in two phases as described below. Therefore, we strongly recommend users to already identify the relevant flows and assess their data exchange security (that is end-to-end or transitive segments as per the above) for proper prioritisation and planification of potential next steps.

[Advisory Phase 1 – in a future CSCF version: **Protect the new point to point flows and the bridging servers that are guardian of the data exchange**

- Protect end-to-end the new direct flows (point to point connections) created between a back-office first hop and components in the secure zone (security by design as from day one)

- Protect the flows that are not end-to-end protected and which rely on bridging servers for their security, through the implementation of:

  1) a secure protocol or mechanism for the direct (point to point) exchange between the secure zone and the bridging server, that ensures authentication, integrity and confidentiality of the data exchanged, AND

  2) the CSCF controls on the bridging server(s) that is/are guardian of the data exchange security, AND

  3) when relevant, a secure protocol or mechanism for the exchange between the bridging servers themselves that ensures authentication, integrity and confidentiality of the data exchanged]

[Advisory Phase 2 – in a subsequent CSCF version: **Protect the legacy flows**

- Protect end-to-end the legacy direct flows (point to point connections) between back-office first hops and the Swift-related components in the secure zone

- Protect the legacy flows that are not end-to-end protected and which rely on bridging servers for their security, through the implementation of:

4) a secure protocol or mechanism for the direct (point to point) exchange between the back-office first hops and the bridging server, that ensures authentication, integrity and confidentiality of the data exchanged]

- New HSM (direct flows from back-office hops should be avoided; flows to New HSM should only be permitted from the SNL or the customer connector using application profiles).

Risk Drivers:

- loss of sensitive data confidentiality
- loss of sensitive data integrity
- unauthenticated system traffic
- unauthorised access

Implementation Guidance

Control Statement:

Confidentiality, integrity, and authentication mechanisms (at system, transport, message or data level) are implemented to protect data flows between Swift infrastructure components and the back-office first hops they connect to.

Control Context:

Protection of data flows or connections between the back-office first hops as seen from the Swift or customer secure zone and the Swift infrastructure safeguards against person-in-the-middle attack, unintended disclosure, modification, and data access while in transit.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- As this control is expected to become Mandatory gradually in a future version of the CSCF, the following guidelines are provided to reach compliance:
  - Set up an inventory of data flows between Swift-related components and the first back-office hops.
  - Assess the security of each flow. For flows that are not properly protected, start defining a plan, considering a risk based approach (supported by a risk assessment) for their prioritisation, to:
    - Implement secure mechanisms or protocols offered by the interfaces, connectors, or middleware server
    - Migrate opportunistically legacy and less regular flows to secure mechanisms or protocols
    - Mitigate (in line with the risk assessment) the risk of back-office system spoofing or message injections through systems or network connectivity means.
- Implement an end-to-end protection between a back-office first hop and a component in the secure zone; that can be achieved through the implementation of:
  - a secure mechanism (for example AES-GCM based Local Authentication, XML DSIG, or the like) ensuring integrity, confidentiality and authentication at the data level where the authentication and authorisation are validated by the recipient. Prevention or detection of possible replay attacks (when same protected data is resubmitted) must be considered.
  - a secure protocol protecting the direct (point to point) exchange between a component in the secure zone and the back-office first hop by ensuring the authentication, integrity and confidentiality of the

data exchange. This can be achieved for example, by using Local Authentication (LAU) in combination with a confidentiality protection[24], authenticated API calls over TLS or by using two-way TLS.

- In the absence of end-to-end protection and where there is one or several servers bridging the back-office first hop and a component in the secure zone:
  - Each leg or segment in the flow (that is, back-office first hop to bridging server; inter bridging servers connections; bridging server to component in the secure zone) is protected for integrity, confidentiality and authentication, the authentication and authorisation are validated by the bridging server. That protection can be obtained through different ways such as a secure mechanism or secure protocol as per the above,

    AND
  - The bridging server(s), guardian of the data exchange, must themselves be protected considering the controls listing 'bridging server' in their in-scope components section.
- The data exchange or end-to-end flow between the back office-first hop and the component in the secure zone must be considered and assessed globally as several methods of protection can be combined. For instance, LAU can be set between the back-office first hop and the component in the secure zone (through for instance a bilateral key) to support authentication and integrity of the data, while the various segments in the flow are protected through a secure protocol covering confidentiality and integrity of the data while it is in transit.
- Secure protocols use current, commonly accepted cryptographic algorithms (for example, AES[25] or ECDHE[26]) with key lengths in line with the current best practices. For more information about cryptographic algorithms that support secure protocols, see Swift Knowledge Centre article 5021566.
- Credentials and private or bilateral keys used, and usually stored, by the applications to protect the flows are secured where a large spectrum of protection, from proper coding guidelines to the usage of specific solutions, can be envisaged based on the user's risk management.

[24] Such as one-way TLS where the sender validates the genuine recipient

[25] Advanced Encryption Standard

[26] Elliptic Curve Diffie-Hellman Ephemeral

2.5A External Transmission Data Protection

Control Type: Advisory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	

Control Definition

Control Objective: Protect the confidentiality of Swift-related sensitive data transmitted or stored outside of the secure zone as part of operational processes.

In-scope components:

- Swift-related secure zone sensitive data (such as back-ups, business transaction details, and credentials)

Risk Drivers:

- compromise of trusted back-up data
- loss of sensitive data confidentiality

Implementation Guidance

Control Statement:

Sensitive Swift-related data that leaves the secure zone as a result of operating system/application back-ups for recovery purposes, business transaction data replication for archiving, or extraction for offline processing is protected when stored outside of a secure zone and is encrypted while in transit to the first storage location.

Control Context:

While **2.4A** covers the back-office application flows with the Swift-related components, this control covers the underlying Swift-related data that resides in the cloud or is exported from the secure zone and manipulated as per operational activities (such as back-ups or manual/automated data extraction/copies).

Operating system or applications back-ups and the replication of business transaction data can provide useful information to prepare fraudulent transactions. The initial transfer, handling, and storage outside of secure zones (when, for example, using the SAN/NAS[27] technology) must therefore be secured to prevent unauthorised access. Flow or data encryption are usual means to protect such data in transit.

Back-up encryption, encryption of data at rest, or appropriate authorisation and access controls are usual means to protect stored data.

Offline processing covers, for example, processing performed for support activities, additional analysis, or business intelligence activities.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- Replicated or extracted Swift-related sensitive data (business transaction data that reveals details such as involved debtors, creditors, accounts, amounts, trade information), passwords, and other authenticators are as follows:
  - Protected from unauthorised access when stored outside of the Swift or customer secure zone or another secure zone that has similar controls as the Swift or customer secure zone. Such replicated or extracted data is also ideally encrypted when stored outside of a secure zone (this can be achieved either at the data, file, application, or system level).

[27] Storage Area Network / Network Attached Storage both providing network storage solutions[28] Advanced Encryption Standard

  - Encrypted when in transit between secure zones (for example, between data centres) or transferred outside of a secure zone (Swift or another zone that has similar controls). Encryption can be applied on the data or at the network/communication/transport layer.
- When relying on a remote virtualisation or cloud platform (that is hosted or operated by a third party, or both) it is recommended to ensure the encryption of the data. This can be obtained at the subscription level or at the storage level, expected to be offered by the third party to provide a guarantee in regard to access to stored data.
- Encryption protocols or mechanisms use a current, commonly accepted cryptographic algorithm (for example, AES[28] or ECDHE[29]) with key lengths in line with current best practices. For more information about cryptographic algorithms that currently support secure protocols, see Swift Knowledge Centre article 5021566.
- Encryption mechanisms comply with applicable laws and regulations[30].
- If the cryptography protecting Swift-related sensitive data has been compromised, then a process should be established to apply new cryptography and secure or destroy any compromised copies of the data.

***Note*:** It is expected that back-ups kept for business or system recovery are maintained in a secure zone that has similar controls to the Swift or customer secure zone.

[28] Advanced Encryption Standard

[29] Elliptic Curve Diffie-Hellman Ephemeral

[30] Such as those identified by Global Partner Digital (https://www.gp-digital.org/world-map-of-encryption/)

2.6 Operator Session Confidentiality and Integrity

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Protect the confidentiality and integrity of interactive operator sessions that connect to the on-premises or remote (operated by a service provider or outsourcing agent) Swift infrastructure or to a service provider or outsourcing agent Swift-related applications.

In-scope components:

Interactive user, operator or management sessions performed from

- dedicated and general-purpose operator PC
- jump server

Any another intermediate system accessed or used from any of the above to connect to

- jump server or any other intermediate system accessed or used from any of the above
- systems hosting a Swift-related component (including interface, GUI, Swift and customer connectors)
- network devices protecting the secure zone
- management console of a virtualisation or cloud platform hosting Swift-related components (including Swift and customer connector)
- interface applications, GUI and Swift or customer connector in the secure zone
- applications at the service provider or outsourcing agent
- HSM
- [Advisory: bridging servers (such as middleware servers other than customer connectors) used for and guardian of the secure data exchange between back-office and Swift-related components]

Risk Drivers:

- loss of operational confidentiality
- loss of operational integrity
- password theft

Implementation Guidance

Control Statement:

The confidentiality and integrity of interactive operator sessions that connect to service provider or outsourcing agent Swift-related applications or into the user's secure zone are safeguarded.

Control Context:

Operator sessions, through the jump server when accessing the on-premises or remote (that is hosted or operated by a third party, or both) Swift infrastructure, pose a unique threat because unusual or unexpected activity is more difficult to detect during interactive sessions than it is during application-to-application activity. Therefore, it is important to protect the integrity and confidentiality of these operator sessions to reduce any opportunity for misuse or password theft. When used, access to the virtualisation layer (virtualisation or cloud management console) must be similarly protected.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- All interactive sessions are protected by a cryptographic protocol (for example, ssh, https with one-way TLS).
- Protocols use a current, commonly accepted cryptographic algorithm (for example, AES[31] or ECDHE[32]), with key lengths in line with the current best practices. More guidelines on cryptographic algorithms that support secure protocols can be found in Swift Knowledge Centre article 5021566.
- Operator sessions and other session types (for example, admin or maintenance) possess an inactivity lock-out feature that limits the session to the minimal time frame necessary to perform business-as-usual duties.
- If the inactivity lock-out is not implemented at the application level, then it should be implemented at the operating system level of the application, or on the jump server.
- The communication between the jump server (when used) and the Swift-related components or underlying systems, is protected using a secure mechanism (for example, one-way or two-way TLS) to support the confidentiality and integrity of the user's connection to the applications or the underlying systems.

Optional Enhancement:

- Apply the control to all bridging servers (such as middleware or file transfer servers other than customer connectors) used for data exchange between back-office and Swift-related components.

[31] Advanced Encryption Standard

[32] Elliptic Curve Diffie-Hellman Ephemeral

2.7 Vulnerability Scanning

Control Type: Mandatory / Advisory for B	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Identify known vulnerabilities within the user's Swift environment by implementing a regular vulnerability scanning process and act upon results.

In-scope components:

- jump server
- dedicated operator PCs
- all systems hosting a Swift-related component (including interface, GUI, Swift and customer connectors)
- [Advisory: on-premises or remote (that is hosted or operated by a third party, or both) virtualisation or cloud platform hosting Swift-related VMs and their management PCs as per the optional enhancement]
- [Advisory: bridging servers (such as middleware or file transfer servers other than customer connector used for data exchange between back-office and Swift-related components]
- [Advisory: General-purpose operator PCs as per the optional enhancement]

Risk Drivers:

- exploitation of known security vulnerabilities
- Unknown security vulnerabilities or security misconfigurations

Implementation Guidance

Control Statement:

Secure zone (including dedicated operator PC) systems are scanned for vulnerabilities using an up-to-date, reputable scanning tool and results are considered for appropriate resolving actions.

Control Context:

The detection of known vulnerabilities allows vulnerabilities to be analysed, treated, and mitigated. The mitigation of vulnerabilities reduces the number of pathways that a malicious actor can use during an attack. A vulnerability scanning process that is comprehensive, repeatable, and performed in a timely manner is necessary to continuously detect known vulnerabilities and to allow for further action.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- Vulnerability scanning is performed at least annually or after any significant change to the environment (for example, introduction of new servers or components, and network design changes that modify or increase the range of in-scope components).
  - Vulnerability scanning tools are from a reputable vendor and are updated with scan profiles within one month prior to scanning.
  - The most appropriate type of vulnerability scanning (such as using credentials) is selected for the environment. Any administrative credentials used for scanning are appropriately protected.
  - Sufficient risk-based safeguards are in place to minimise any operational impact (for example, running scans in safe mode, or omitting systems that may be negatively affected from the scan).
- Beyond vulnerability identification through scanning, all penetration tests or effective vulnerability tests on or through Swift-related services and products are consistent with the Swift Customer Testing Policy.

- The outcome of the vulnerability scanning is documented (with restricted access) and analysed for appropriate action and remediation (such as applying security updates in line with control **2.2**).
- Once per quarter, month, or real-time (preferred) scanning is recommended.

Optional Enhancements:

- Vulnerability scanning includes network devices protecting the secure zone (such as routers and switches).
- Vulnerability scanning includes the general-purpose operator PCs used to connect to the user's, outsourcing agent's or service provider's Swift infrastructure. As an alternative, security updates are regularly applied on the general-purpose operator PCs. In the latter case, only supported and regularly patched applications are deployed on those PCs.
- Vulnerability scanning possibly includes the on-premises or remote (that is hosted or operated by a third party, or both) Virtualisation or cloud platform that hosts the Swift-related VMs.
- Apply the control to all bridging servers (such as other middleware or file transfer servers other than customer connectors) used for data exchange between back-office and Swift-related components.

2.8 Outsourced Critical Activity Protection

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Ensure the protection, in line with the CSCF, of the user's Swift infrastructure from risks exposed by the outsourcing of critical activities.

In-scope components:

- this organisational control is applicable when critical Swift-related activities are outsourced to a third party or a service provider.
- as per the "Scope of Security Controls" section, the 'user's Swift infrastructure' encompasses the collection of, on-premises or remote (that is hosted or operated by a third party, or both), Swift-components managed by or for users by a third party, including applications, network devices, tokens and other removable media, and supporting hardware.

Note:

- Users that engage with third parties (for example, an external IT provider, a cloud provider or an outsourcing agent) or service providers (such as a service bureau, a Business Connect or a Lite2 for Business Application provider) to host or operate in full or in part the user's Swift infrastructure must:
  - attest for their comprehensive architecture type (as if it was operated on premises) and therefore,
  - obtain reasonable comfort from such third parties or services providers that the outsourced activities are protected, at a minimum, to the same standard of care as if operated within the originating organisation and in line with the CSCF security controls. Such third parties and service providers are allowed to rely on their compliance programme that usually builds on maintained certification(s) or assurance, and to map these with the CSCF controls when providing comfort to users they serve.
- This control remains strongly recommended even when the outsourced activities are not critical.

Risk Drivers:

- exposure to sub-standard security practices

Implementation Guidance

Control Statement:

Critical outsourced activities are protected, at a minimum, to the same standard of care as if operated within the originating organisation.

Control Context:

When critical activities are outsourced to third parties (for example, external IT provider, cloud provider or outsourcing agent) or service providers (a service bureau, a Business Connect or a Lite2 for Business Application provider), it is essential that at a minimum, the original standard of care for security is maintained (in addition to adherence to this customer security control framework) to make sure that no new weaknesses or vulnerabilities are introduced.

Note:

- Swift defines the following activities related to the user's Swift infrastructure, at a minimum, as critical:
  - security management and change management of the hardware, including HSM, and software (including applications, operating system, and underlying virtualised platform or infrastructure) supporting the user's Swift infrastructure
  - RMA and Business Transaction controls related operations (in support of controls 2.9 and 2.11)
  - accessing sensitive user data (for example, message content) processed by the user's Swift infrastructure
  - monitoring of events generated by the user's Swift infrastructure that contain sensitive user data
  - network management and configuration of the user's Swift infrastructure
  - Swift-related transaction operations (for example, creation or modification of a financial transaction message within the messaging interface or a connector)

  - security administration of the users entitlements, their tokens or (private) keys needed to perform Swift-related transaction operations
  - Ancillary services operations when sharing/reusing, for those ancillary services, credentials and roles/entitlements (accounts) used for Swift-related transaction operations. Otherwise, such separated/independent ancillary services accesses are not to be considered as in scope,
- External contractors are not always considered as third party resources:
  - if those external contractors are members of the of the institution task force as contributors, then they are considered user's employees
  - however, if external contractors manage or operate user's components remotely from their company location or equipment, as part of a managed service contractual agreement, they are considered as third-party resources.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- When hosting or outsourcing its own Swift infrastructure (or a part of it) to a third party (such as an external IT provider or a cloud provider as outsourcing agent) or a service provider acting on their behalf, the users remain responsible for the conformance with the security controls of this framework and must seek compliance from that third party or service provider to complement their attestation.
- When the third party provides shared services to connect non-related Swift users, the third party must be registered for the Shared Infrastructure Programme (SIP) or the Alliance Lite2 for Business Applications (L2BA) programme. Users remain responsible for their own infrastructure, organisation, for implementing secure data flows toward the provider in line with the provider's specifications and for implementing multi-factor authentication in line with control 4.2 when accessing the provider's resources. Users are also responsible for monitoring the provider's compliance with the relevant SIP or L2BA programme[33]:
  - Service bureaux registered and compliant under the SIP are listed in the Swift Partner Programme Service Bureau Directory.
  - L2BA providers registered and compliant under the related programme are listed in the Lite2 Business Applications Providers Directory.
- When relying on an outsourcing agent to collect, process and further submit Swift-related transactions to a service bureau, an L2BA provider or even directly to Swift (through for instance Alliance Cloud, Lite2, Transaction Manager or another Swift channel), the users remain responsible for the conformance with the security controls and must seek compliance from that outsourcing agent to complement their attestation.
- Service Level Agreements (SLAs) and a Non-disclosure Agreement (NDA) are established with any third party or service provider, including outsourcing agent, when critical activities have been outsourced. These SLAs define the standard of care under which those critical operations are carried out by the third party or the service provider.
- A security risk assessment supporting the outsourcing to the third party is conducted at the start of the engagement, and is reviewed on a regular basis thereafter.

[33] A provider remains listed as long as it is compliant. Should it be de-listed, it would be listed again once compliance is regained.

2.9 Transaction Business Controls

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Ensure outbound transaction activity within the expected bounds of normal business.

In-scope components:

- GUI
- messaging interface
- communication interface
- Swift and customer connector

Note: Components are mentioned as the vector for outbound transaction business, not necessarily where the controls are performed (these controls can be business controls performed outside of the secure zone). Transaction activity refers to payment instructions. Reliance on other relevant recent (business) assessment, audit or regulator answers to confirm effectiveness of the control is an option[34].

Risk Drivers:

- undetected anomalies or suspicious activity

Implementation Guidance

Control Statement:

Implement transaction detection, prevention, or validation controls, or a combination of them to ensure outbound transaction activity within the expected bounds of normal business.

Control Context:

Implementing business controls that restrict Swift transactions to the fullest extent possible reduces the opportunity for the sending (outbound) and, optionally, receiving (inbound) of fraudulent transactions. These restrictions are best determined through an analysis of normal business activity. Parameters can then be set to restrict business to acceptable thresholds based on “normal” activity.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user’s implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- Implement controls that will detect, prevent, or additionally validate the flow of transactions against the expected bounds of normal business. **Examples** of potential measures can include any or a combination of the following five elements:
  - (1) limiting traffic outside of business hours

    Note: This measure may not be applicable to some users: as business hours are organisational and business unit-specific, multiple start and finish times (business hours) may need to be supported or no specific range can be defined for systems used on a 24-hour basis. In cases of 24-hour centralised Swift processing, limit or monitor transactions as appropriate to support business as usual considering the other elements.

[34] Although reliance on recent certification, audit or assessment is allowed for any CSP control as per the Independent Assessment Framework (IAF), it is even more relevant in this control.

- Consider restricting Swift transaction submissions and approvals outside of normal business hours[35].
- Consider enabling active FIN sessions to business hours only (for example, using automated logical terminal sessions log out at the end of the business day).

  Note: Suspicious messages can be blended in with legitimate traffic during business hours. Therefore, always limit or monitor transactions as appropriate, to support business as usual activities considering the next elements.

– (2) limiting traffic beyond normal business amount ranges
  - Consider restricting Swift transactions outside of user-defined amount limits. Such limits can be specified globally, per region, currency, transaction type or known correspondents, in line with functionalities offered by the used Swift-related interface, application or service. Putting on-hold restricted transactions for additional/off-line validation and approval (in line with separation of duties as per control 5.1) is deemed a valid control.

– (3) performing end-of-day and (possibly) intra-day validations or reconciliations through any or a combination of the following
  - Consider implementing a process to issue and check confirmation messages (for example, to check that the MT 900 and MT 910, or their MX camt.054 equivalent confirmations match the transactions that have occurred on the accounts or through potential online queries, for intra-day Nostro reconciliation).
  - Consider reconciling the entity's accounting records with end-of-day statement messages (for example, using MT 940 and MT 950, their MX camt.053 equivalent, or through potential online queries for end-of-day Nostro reconciliation).
  - Consider reconciling daily (and possibly intra-day) messages that are sent to/from the back office and to/from the Swift Network.

– (4) performing central checks on payments to spot potential abnormal behaviour
  - Consider tracking session numbers within the messaging interface to make sure that the sequential session numbering is intact with no unexpected gaps.
  - Consider monitoring uncharacteristic transactions (for example, exceptionally high amounts or cumulative amounts, unusual beneficiaries, senders, or currencies) based on self-determined criteria.

– (5) performing independent reconciliation with transaction data securely obtained from a secondary source (either internal or external, such as reports from service providers) or verifying that the transaction is genuine with the emitter or the recipient (or both).

Optional Enhancements:

- Application and operating system accounts are restricted from login attempts that occur outside of the expected role-specific operational hours.
- Implement controls to ensure inbound transaction activity within the expected bounds of normal business.
- Implement controls to other sensitive transactions not limited to payments.
- Consider combining several types of controls (considering (1) and (2) as preventive control; (4) as on-line detective control; (3) and (5) as post-event validation control)

[35] Limiting or dully controlling sessions outside of normal business hours can introduce delays allowing to intercept/recall fraudulent transactions before their potential immediate processing and ultimately cash-out.

2.10 Application Hardening

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		●	●	●		

Control Definition

Control Objective: Reduce the attack surface of Swift-related components by performing application hardening on the Swift-compatible messaging and communication interfaces, the Swift connector and related applications.

In-scope components:

- messaging interface
- communication interface
- GUI
- SWIFTNet Link
- Swift connector

Risk Drivers:

- excess attack surface
- exploitation of insecure application configuration

Implementation Guidance

Control Statement:

All messaging interfaces and communication interfaces products within the Swift secure zone are Swift-compatible. Application security hardening is conducted and maintained on all in-scope components.

Control Context:

Application hardening applies the security concept of “least privilege” to an application by disabling features and services that are not required for normal operations. This process reduces the application capabilities, features, and protocols that may be used during an attack. The process also makes sure that potential default credentials are changed.

In addition, Swift runs a Compatible Interface Programme to make sure interfaces are aligned with current practices and to give the customer additional assurance, guarantees, and better visibility regarding individual product capabilities. Upon the successful validation of the test results by the Swift Test Authority, the interface is published in the Compatible Register. As per the Swift General Terms and Conditions, customers must use a Swift-compatible interface.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user’s implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- Make sure the messaging and communication interfaces are Swift-compatible (the list of compatible interfaces is published in the Compatible Register on www.swift.com).
  - The Swift-compatible interface should meet all the security conformance requirements (mandatory and advisory) defined in the Swift Compatible Interface Programme.
    - If some security conformance requirements are yet to be met, then the user should upgrade to a Swift-compatible interface by implementing at least the minimum mandatory security conformance requirements.
    - The interface provider should be contacted in case of doubts regarding the availability of some security functionalities or their proper configuration and usage.
- All in-scope applications are hardened considering one or more of the following:

  - vendor security, operational or configuration guidance (such as the Alliance Security Guidance)
  - a local or a regulator's standard security configuration, or controls set of the same rigour as the vendor guidance
- At a minimum, the application hardening process should do the following:
  - Change default existing passwords.
  - Disable or remove unnecessary user accounts.
  - Disable or restrict unnecessary components, adaptors, or connectivity methods.
  - Securely configure the adapters, connectivity methods, or remote connections.
  - Remove unnecessary packages.
  - Adjust any default configurations known to be vulnerable.
- Deviations from the selected hardening configuration (that is, a set of rules) are documented along with the justification for the deviation.

Optional Enhancement:

- Additional applications installed on the systems that host in-scope components and handle Swift-related data are also subject to considered application hardening as per the vendor recommendations.

2.11A RMA Business Controls

Control Type: Advisory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Restrict transaction activity to validated and approved business counterparties.

In-scope components:

- GUI
- dedicated and general-purpose operator PCs
- messaging interface
- Swift and customer Connector

Note: GUI, connectors, and messaging interface are mentioned as potential vectors for Relationship Management Application (RMA) exchange and reporting. Operator PCs used to access the RMA Portal for central management are also mentioned as potential vectors.

Risk Drivers:

- business conducted with an unauthorised counterparty

Implementation Guidance

Control Statement:

Implement RMA controls to restrict transaction activity with effective business counterparties.

Control Context:

Implementing business controls that restrict Swift transactions to the fullest extent possible reduces the opportunity for both the sending and receiving of fraudulent transactions. These restrictions are best determined through an analysis of effective business relationships where RMA is a mechanism to prevent unwanted traffic on a service by controlling who can send traffic and what type of messages can be exchanged through Relationship Management Application Plus (RMA+).

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- RMA
  - Appropriate know-your-customer principles and due diligence is performed during the creation and maintenance of RMA relationships.
  - RMA relationships are reviewed annually (at least) to make sure that obsolete (unused, dormant, or unwanted) relationships are analysed and removed or revoked in a timely manner.

Optional Enhancement:

- RMA+
  - Restrict the valid RMA relationships to the specific message types that are agreed with the counterparty.

3 Physically Secure the Environment

3.1 Physical Security

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Prevent unauthorised physical access to sensitive equipment, workplace environments, hosting sites, and storage.

In-scope components:

- dedicated and general-purpose operator PC, including removable equipment
- jump server
- on-premises or remote (that is hosted or operated by a third party, or both) hardware, including HSM, hosting a Swift-related component (including interface, GUI, Swift and customer connectors)
- on-premises or remote (that is hosted or operated by a third party, or both) hardware supporting virtualisation or cloud platform and hosting Swift-related VMs
- Alliance Connect SRX VPN boxes and Alliance Connect Virtual VPN instances
- [Advisory: bridging servers (such as middleware or file transfer servers other than customer connectors) used for and guardian of the secure data exchange between back-office and Swift-related components]

Note: Alliance Connect SRX VPN boxes and Alliance Connect Virtual VPN instances (hosting systems or machines) must also be in an environment with appropriate physical controls as described below.

Risk Drivers:

- lack of traceability
- unauthorised physical access

Implementation Guidance

Control Statement:

Physical security controls are in place to protect access to sensitive equipment, hosting sites, and storage.

Control Context:

Implementing physical security controls protects against insider and external threats, and reduces opportunistic attacks enabled by access to physical systems.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- Security of Removable Equipment
  - Sensitive removable equipment, such as PIN Entry Devices (PEDs), PED keys, Swift-related smart cards, USB Tokens, and (time-based) one-time password ((T)OTP) Devices, is supervised (that is, it always remains under the control of its owner) or is securely stored. For example, it does not remain plugged in a device that is unattended.
  - Sensitive removable equipment required for normal continuous operations (for example, hot swappable disks or HSM devices) are hosted in a data centre or, at a minimum, in a locked room.

- Back-up media (for example, tapes) is physically secured.

- Security of the Workplace Environment
  - Operator PCs are located in a secured workplace environment where access is controlled and granted only to employees and other authorised workers and visitors. A separate physical area for operator PCs to access Swift systems is not required.
  - Printers used for Swift transactions are located in a secured workplace environment and their access is restricted.
- Security for Remote Workers (for example, teleworkers or "on call" operations staff)
  - A security policy is established to support expected use cases for remote workers. The following items are considered when establishing the policy:
    - physical security of the expected teleworking environment
    - rules for personal equipment used for Swift business purposes (for example, personal PCs cannot be used to access the Swift infrastructure, however personal mobile devices can be used as a second authentication factor)
    - security during use in public environments
    - security during public and private transport
    - equipment storage
    - unauthorised access to equipment (for example, from family or friends)
    - remote access requirements (recommended VPN with multi-factor authentication)
    - protection of mobile devices used for authentication, such as (T)OTP (recommend enabling password and auto-lock features)
    - compensating controls (for example, virtual desktop preventing local storage, full-disk encryption)
    - reporting of security incidents (for example, theft) while working remotely
- Security of the Server Environment
  - Servers are hosted in a data centre or, at a minimum, in a locked room with limited and controlled access (for example, using access control cards or biometrics).
    - Ideally, servers are rack-mounted. A risk assessment is conducted to determine if a separate and exclusive rack, or the locking of the rack, is appropriate based on the existing data centre physical access controls.
  - The server environment has video surveillance with movement detection and recording equipment. The implementation of video surveillance recording and retention of images comply with applicable laws and regulations[36]. Ideally, images are retained for at least three months.
  - No physical reference to Swift on servers (for example, labels).
- Equipment Disposal
  - Consider sanitisation of the embarked data (such as secure deletion, overwriting or encryption on storage media, or manufacturer's reset) when in-scope equipment is to be disposed or reused.
- Physical Access Logging and Review
  - Physical access to sensitive equipment areas (for example, data centre, secured storage) is logged.
  - Physical access logs are available for audit and investigations, and are retained for a minimum of 12 months and in compliance with applicable laws and regulations.
  - Physical access is promptly revoked (or modified) when an employee changes roles or leaves the organisation.
  - Physical access control lists are reviewed annually (at least).

Optional Enhancement:

- Apply the control to all bridging servers (such as middleware or file transfer servers other than customer connectors) used for data exchange between back-office and Swift-related components.

[36] Such as the "Guidelines 3/2019 on processing of personal data through video devices", local Data Protection Act/code of practice or Laws related to video surveillance

4 Prevent Compromise of Credentials

4.1 Password Policy

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Ensure passwords are sufficiently resistant against common password attacks by implementing and enforcing an effective password policy.

In-scope components:

Passwords defined on the following components:

- dedicated and general-purpose operator PCs
- jump server
- Swift-related components (including interfaces, GUI, Swift and customer connectors, new HSM)
- systems hosting Swift-related components
- network devices protecting the secure zone
- on-premises or remote (that is hosted or operated by a third party, or both) virtualisation or cloud platform hosting Swift-related VMs and their management PCs
- personal tokens and personal mobile devices used as possession factor for multi-factor authentication (considered as software tokens) (see control **4.2**)
- [Advisory: bridging servers (such as middleware or file transfer servers other than customer connectors) used for and guardian of the data exchange between back-office and Swift-related components]

Risk Drivers:

- password cracking, guessing, or other computational compromise

Implementation Guidance

Control Statement:

All application and operating system accounts enforce passwords with appropriate parameters such as length, complexity, validity, and the number of failed login attempts. Similarly, personal tokens and mobile devices enforce passwords or a Personal Identification Number (PIN) with appropriate parameters.

Control Context:

Implementing a password policy that protects against common password attacks (for example, guessing and brute force) is effective for protecting against account compromise. Attackers often use the privileges of a compromised account to move laterally within an environment and progress the attack. Another risk is the compromise of local authentication keys to tamper with the integrity of transactions.

However, it is important to recognise that passwords alone are generally not sufficient in the current cyber-threat landscape. Users should consider this control in close relationship with the multi-factor authentication requirement.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- A password policy that also covers PIN settings is established, aligned to current industry standards or industry best practices, and defines the following criteria:
  - password expiration
  - password length, composition, complexity, and other restrictions
  - password re-use
  - lock out after failed authentication attempts (and remedy)
  - password requirements modified as necessary for the following specific use cases:
    - in combination with a second factor (for example, one-time password)
    - authentication target (for example, operating system, application, mobile device, or token)
    - type of account (general operator, privileged operator, application-to-application account, or local authentication keys)

  For additional best practice guidelines about password and PIN parameter settings, see Swift Knowledge Centre articles 5021567 and 5022038.
- The password policy is developed in consideration of known password-based vulnerabilities in the computing environment. For example, requiring a password of 15 or more characters for Windows systems prevents Windows from computing the highly vulnerable LAN Manager (LM) password hash.
- The established password policy is enforced through technical means (for example, through an Active Directory group policy, or within application settings), when possible.
- Effectiveness of the password policy is reviewed regularly (annually, by recommendation).
- System settings related to password management and storage are aligned to industry and vendor best practices (for example, enabling the “NoLMHash” registry setting in Windows).
- Passwords used for secure zone systems are significantly more exposed if the passwords are stored in authentication systems outside of the secure zone (for example, an enterprise Active Directory). Instead, passwords for secure zone systems are, to the fullest extent possible, stored only within the zone (for example, in an Active Directory for production systems) as described in the guidance for the design of the secure zone or another existing secure zone that has similar controls.

Optional Enhancement:

- Apply the control to all bridging servers (such as middleware or file transfer servers other than customer connectors) used for data exchange between back-office and Swift-related components.

***Note*:** Users should implement strong passwords and preferably strong authentication mechanisms for all systems used within the end-to-end transaction chain, and not limit these controls to the Swift infrastructure only.

4.2 Multi-Factor Authentication

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Prevent that a compromise of a single authentication factor allows access into Swift-related systems or applications by implementing multi-factor authentication.

In-scope components (depending on implementation):

- dedicated operator PC login
- access to jump server and to new HSM
- login process to the messaging interface (including a related hosted database), communication interface, Swift and customer connector (including a related hosted database) or a service provider or outsourcing agent Swift-related application
- login process to (operating) systems hosting the messaging interface (including a hosted database), Swift and customer connector (including a related hosted database) and communication interface or a service provider or outsourcing agent Swift-related application
- access to the remote Swift infrastructure (that is hosted or operated by a third party, or both)

Risk Drivers:

- credential replay
- password cracking, guessing, or other computational compromise
- password theft

Implementation Guidance

Control Statement:

Multi-factor authentication is used for interactive user access to Swift-related components or applications and operating system accounts.

Control Context:

Multi-factor authentication requires the presentation of two or more of the following common authentication factors:

- knowledge factor: something the operator knows (for example, a password)
- possession factor: something the operator has (for example, connected USB tokens or smart cards, or disconnected tokens such as a (time based) one-time password- (T)OTP- generator or application storing a cryptographic private key that runs on another device like operator's mobile phone considered as a software token, RSA token, 3-Skey Digital and its mobile version considered as a software token, or Digipass)
- inherence factor: something the operator is (for example, biometrics such as fingerprints, retina scans, or voice recognition)

Implementing multi-factor authentication provides an additional layer of protection against common authentication attacks (for example, shoulder surfing, password re-use, or weak passwords) and provides further protection from account compromises for malicious transaction processing. Attackers often use the privileges of a compromised account to move laterally within an environment and to progress an attack.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- When implementing multi-factor authentication, the following principles apply:
  - When based on a knowledge factor (typically a password) combined with a possession factor (a mobile device), the device used for the second factor must not be the same as the device used to enter the first factor. As such, using an app to generate the second factor on the same device/PC used to enter the first factor (password) is not sufficient to access the user's Swift systems.
    - Second factor solutions based on a possession factor include (but are not limited to) TOTP, RSA SecurID, Digipass, Mobile App, Transaction Authentication Number (TAN) Table, 3-Skey Digital and its mobile version, and personal USB token. The solution should be selected per the user's risk management, considering NIST SP 800-63B Authenticator Assurance Level 2 as guidance).
  - An inherence factor is more safely combined with a possession factor than with a knowledge factor.
- Multi-factor authentication is implemented on one authentication stage/step (at minimum) encountered by the system administrator or the end user when accessing a Swift application, the new HSM or the hosting system.
  - Operating system administrators when accessing the hosting system or new HSM:
    - at the secure zone boundary (jump server)
    - at the dedicated operator PC login (within the secure zone)
  - End users (in descending order of security robustness) when accessing the Swift application or new HSM:
    - on the individual Swift applications (the browser-based GUI, the messaging interface, or the communication interface)
    - at the secure zone boundary (jump server)
    - at the dedicated operator PC login (that is, within the secure zone)
- Multi-factor authentication is implemented for remote user administrative access, generally for VPN authentication.
- Multi-factor authentication systems are significantly more exposed if the authentication credentials are stored outside of the secure zone (for example, within an enterprise Active Directory). If possible, then the authentication system that supports the multi-factor solution is located within the secure zone.
- The presented authentication factors are individually assigned and support the individual accountability of access to services, operating systems, and applications.
- If single sign-on (for example, SAML) is implemented, then a second factor is still required at the login or at a later stage.
- When accessing (at least for transaction processing[37]) a Swift-related service, application, or component that is operated by a service provider or a third party (such as a service bureau, a Business Connect or an L2BA provider, or an outsourcing agent), multi-factor authentication must be used.

***Note*:** All Swift and Swift-compatible third-party vendor messaging and communication interfaces must support or embed multi-factor authentication.

Considerations for alternative implementations:

When the device used for the second factor is the same as the one used to enter the first factor, additional mitigations must be identified and implemented in line with a user risk assessment (considering NIST SP 800-63B SP Authenticator Assurance Level 2 as guidance)

The objective of the risk assessment is to evaluate and keep potential risks under user's risk appetite when combining factors in case of loss, theft or compromise of such device. Mitigations can include technical measures (such as enforcing a PIN or a password to unlock an application linked with the registered device, application that generates a one-time string; limiting the accessed functions depending on the device level of trust; requiring additional factor, such as a biometric factor to unlock cryptographic private key(s) used as possession factor, for most sensitive functions,…) and include as well complementary procedural requirements (such as Policy asking end user to immediately contact a security operations centre -SOC- to block, potentially remote-wipe, or put on-hold any potential access or transaction performed through this lost, stolen or potentially compromised device).

[37] such as creating, submitting, approving, or modifying transactions or user entitlements.

5 Manage Identities and Separate Privileges

5.1 Logical Access Control

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Enforce the security principles of need-to-know access, least privilege, and separation of duties for operator accounts.

In-scope components:

User, operator or management accounts defined on the following components:

- on-premises or remote virtualisation or cloud platform and their management PCs, hosting Swift-related VMs and on those VMs themselves
- jump server
- dedicated operator PCs
- operating systems hosting interfaces, GUI, Swift and customer connectors, service provider or outsourcing agent Swift-related applications
- interfaces, GUI, connectors, service provider or outsourcing agent Swift-related applications
- HSM
- network devices protecting the secure zone
- SWIFTNet Online Operations Manager (O2M) on swift.com
- [Advisory: All operator accounts on the bridging servers (such as middleware or file transfer servers other than customer connectors), used for and guardian of the secure data exchange between back-office and Swift-related components]

Note: Sharing/reusing credentials and roles/entitlements (accounts) for business transactions on other systems or components used for ancillary services will turn those systems or components in scope of the CSCF controls.

Risk Drivers:

- excess privilege or access
- separation of duty violation
- unauthorised access
- HSM management misused

Implementation Guidance

Control Statement:

Accounts are defined according to the security principles of need-to-know access, least privilege, and separation of duties.

Control Context:

Applying the security principles of (1) need-to-know, (2) least privilege, and (3) separation of duties is essential to restricting access to the user's Swift infrastructure. Effective management of operator accounts reduces the opportunities for a malicious person to use these accounts as part of an attack.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be

considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

A logical access control policy is documented and enforced based on the following principles:

- Need-to-know.
  - Only operators (end users and administrators) who have a continuing requirement to access the secure zone are permitted to have accounts within the secure zone.
  - Privileges are only assigned to an operator with a validated need-to-know (for example, system set-up makes sure that operators only have access to the information, files, and system resources necessary for their defined tasks). Access to other system functions is disabled.
- Least Privilege.
  - The system set-up makes sure that user and administrator privileges are controlled in a way that allows all privileges to be tailored to individual needs.
  - Accounts are granted only to privileges that are required for normal, routine operation. Additional privileges are only granted on a temporary basis.
- Separation of Duties and Four-Eyes.
  - Vendor documented guidance on role separation is followed in vendor-specific documentation.
  - Sensitive duties are separated. This means that some roles cannot be represented by the same individual, such as:
    - Transaction submission and transaction approval
    - Application Administrator and security officer roles
    - Network and operating system administrators.
  - Sensitive permissions are separated to prevent by-passing the Four-Eyes principle. At a minimum, this requirement applies to access control and security configuration operations on the following components: Messaging and Communication Interface, GUI, Swift and customer connectors, service provider or outsourcing agent Swift-related applications, HSMs, SWIFTNet Online Operations Manager, and Secure Channel.
- Account Review and Revocation
  - Privileges (including those delegated to providers) are promptly revoked when an employee changes roles or leaves the organisation (or the provider). Privileges assignment must ensure continuous accountability and traceability.
  - Accounts (including those delegated to providers) are reviewed at least annually and adjusted as required to continuously ensure accountability and traceability of accounts assignment.
- An emergency procedure to access privileged accounts is documented for use when authorised people are unavailable due to unexpected circumstances:
  - Any operational use of the procedure is logged.
  - Access to the emergency privileged accounts is controlled. Usage is logged, in line with or as part of the control 6.4, ensuring accountability and traceability and the password is changed after emergency use.

Optional Enhancement:

- Apply the control to the operator accounts on all the bridging servers (such as middleware or file transfer servers other than customer connectors) used for data exchange between back-office and Swift-related components.

5.2 Token Management

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Ensure the proper management, tracking, and use of connected and disconnected hardware authentication or personal and software tokens (when tokens are used).

In-scope components:

- connected and disconnected hardware authentication or personal tokens used for Swift operations or secure zone access
- software tokens used as second authentication factor (when explicitly identified for their assignment and management in support of control 4.2)
- PIN Entry Device (PED) used for HSM operations

Risk Drivers:

- authentication token theft
- lack of traceability
- HSM management misused

Implementation Guidance

Control Statement:

Connected and disconnected hardware authentication or personal tokens are managed appropriately during their assignment, distribution, revocation, use, and storage.

Control Context:

The protection of connected and disconnected hardware authentication, personal tokens or software tokens is essential to safeguarding the related operator or system account. It also reinforces good security practice by providing an additional layer of protection from attackers.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- A controlled process is used for the assignment and distribution of connected and disconnected hardware, personal tokens or software tokens used for Swift operations (for example USB token, HSM token, smart card, (T)OTP- generator or application storing a cryptographic private key that runs on another device like operator's mobile phone).
- Hardware and software token assignment, including those delegated to a provider, is reviewed at least annually (more frequently is recommended).
- Personally assigned hardware or software tokens, including those delegated to a provider, are revoked when the individual no longer requires access and should be recalled (for possible disposal or reassignment as appropriate).
- A record is maintained of assigned hardware token ownership.
- Hardware token is supervised (it always remains under the control of its owner) or it is securely stored. As such, it does not stay plugged in a device unattended by the token owner. .
- When a remote PED is used, the following security practices apply:

- PED keys must be stored and only accessible by relevant staff (originals and copies should be stored in a safe with access tracking)
- Although the HSM PED keys are not personally assigned, usage should be controlled, tracked and monitored. In case a PIN is set on the PED keys and a person with access to these keys and PIN is leaving the company, the PIN codes should be changed
- The flows to the HSM must be secured as per the Alliance Security Guidance considering also the CSP FAQ (Swift Knowledge Centre article 5021823 related to control 5.2 remote PED keys) to properly establish and manage the connection.

5.3A Staff Screening Process

Control Type: Advisory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: To the extent permitted and practicable, ensure the trustworthiness of staff operating the user's Swift environment by performing regular staff screening.

In-scope components:

- All staff (such as employees, agents, consultants and contractors) with operational (maintenance or administration) access to Swift-related systems, HSMs, Swift and customer connector or middleware servers and on-premises or remote virtualisation or cloud platform hosting Swift-related VMs, Swift and customer connector VMs or middleware server VMs.

Risk Drivers:

- untrustworthy staff or system operators

Implementation Guidance

Control Statement:

Staff operating the user's Swift infrastructure are screened prior to initial appointment in that role and periodically thereafter.

Control Context:

A staff screening process with internal or external clearance, provides additional assurance that operators or administrators of the user's Swift infrastructure are trustworthy, and reduces the risk of insider threats.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

To the extent permitted under applicable laws and regulations[38], and considering local practices and available information, the following guidelines and specified verifications are recommended:

- All in-scope staff are screened at least every 5 years.
  - For those already in the role and not yet screened, a catch-up process is gradually organised as part of the periodic screening (sometimes also referred to as re-screening)
- The screening process for initial employment includes the following verifications:
  - Identity verification
  - Confirmation of full details of qualifications
  - Confirmation of previous employment history
  - Details of any past or pending civil or criminal proceedings against the employee
  - Validation of any involvement in external businesses that could result in a conflict of interest
  - Financial credit verification
- The periodic screening process includes the following verifications:
  - Details of any pending civil or criminal proceedings against the employee
  - Validation of any involvement in external businesses that could result in a conflict of interest

[38] Including, where applicable, social concertation

- Financial credit verification

Note: in case of staff not directly employed by the Swift user (such as agents, contractors or consultants), the screening can fall under contractual obligation between the Swift user and the employer.

5.4 Password Repository Protection

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Protect physically and logically the repository of recorded passwords.

In-scope components:

Repository recording accounts and passwords defined on the following components:

- dedicated and general-purpose operator PC
- jump server
- Swift-related components (including interfaces, GUI, Swift and customer connectors)
- systems or virtual machines hosting Swift-related components
- HSM and related tokens
- network devices protecting the secure zone
- on-premises or remote (that is hosted or operated by a third party, or both) virtualisation or cloud platform hosting Swift-related VMs
- SWIFTNet Online Operations Manager (O2M) on swift.com
- [Advisory: bridging servers (such as middleware or file transfer servers other than customer connectors used for data exchange between back-office and Swift-related components]

Risk Drivers:

- password theft

Implementation Guidance

Control Statement:

Recorded passwords are stored in a protected physical or logical location, with access restricted on a need-to-know basis.

Control Context:

The secure storage of recorded passwords (repository) makes sure that passwords are not easily accessible to others, thereby protecting against simple password theft. Common unsecure methods include, but are not limited to: recording passwords in a spreadsheet or a text document saved in cleartext on a desktop, or in a shared directory, or a server, saved on a mobile phone, written/printed on a post-it or a leaflet.

This control covers the storage of emergency, privileged or any other account passwords. All accounts have to be considered because (i) combination of compromised, not-privileged, accounts, such as transaction creator account and approver account can be damageable, and (ii) even monitoring accounts provide valuable information during the reconnaissance time.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- Passwords written on physical media are protected through:
  - placing inside a sealed, tamper-evident security envelope
  - storing in a safe
  - logging the access to the storage location and which account passwords have been accessed.
- Passwords stored logically (digitally) are protected through:

  - Encryption-at-rest or obfuscation (that is, no plain text storage),
  - Authenticated access to the storage location, ideally with access logging in line with or as part of the control 6.4.
- Passwords are not recorded in user manuals or other operational means unless the password is stored in line with the guidance above.
- If emergency access is granted to an operator who, under normal conditions, would not have access, then the password is changed immediately thereafter, and optionally, also the combination to the storage safe.
- Passwords are not hardcoded in scripts or other software code.

Optional Enhancement:

- The safe is certified through, for example, Underwriters Laboratories (UL) Class TL or EN-1143-1 certification.
- Apply the control to all bridging servers (such as middleware or file transfer servers other than customer connectors) used for data exchange between back-office and Swift-related components.

6 Detect Anomalous Activity to Systems or Transaction Records

6.1 Malware Protection

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		●	●	●	●	●

Control Definition

Control Objective: Ensure that the user's Swift infrastructure is protected against malware and act upon results.

In-scope components:

Anti-malware software is implemented on Windows operating systems of the below components:

- dedicated and general-purpose operator PC
- jump server
- management PCs of an on-premises or remote (that is hosted or operated by a third party, or both) virtualisation or cloud platform
- systems hosting a Swift-related components (including interface, GUI, Swift or customer connector)
- [Advisory: bridging servers (such as middleware or file transfer servers other than customer connectors used for data exchange between back-office and Swift-related components]

Risk Drivers:

- execution of malicious code
- exploitation of known security vulnerabilities

Implementation Guidance

Control Statement:

Anti-malware software from a reputable vendor is installed, kept up-to-date on all systems, and results are considered for appropriate resolving actions.

Control Context:

Malware is a general term that includes many types of intrusive and unwanted software, including viruses. Anti-malware technology (a broader term for anti-virus) is effective in protecting against malicious code that has a known digital or behaviour profile.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- On-access anti-malware scanning (also known as real-time or background scanning) is performed on all in-scope systems. On-demand full scanning is scheduled at least on a weekly basis for operator PCs (ideally on a daily basis). On-demand full scanning should be scheduled regularly for servers in line with business and operational constraints. For performance reasons full scans are performed at times of low usage, outside of business hours, or both.
- The scope of the scanning should include all files of the systems in scope. Exclusion of elements or directory from scanning is subject to risk assessment considering user's infrastructure set-up, internal security requirements and policies, the product capabilities and the following principles:
  - Software (such as exe, libraries, scripts) and static data (such as configuration files) are expected to be scanned on-access or at installation, and regularly thereafter, when complemented with a run-time

integrity mechanism (in line with the software integrity check depicted in control 6.2) allowing the identification of file changes or unexpected additions.

- Database server content (data files) can be excluded from the scanning when the data has been checked, validated, and scanned at least once before being stored.

- Anti-malware software from a reputable vendor is installed on all computing platforms and updated in line with the scanning frequency.
- Systems that fail to update their profiles or run scheduled scans are detected and corrected.
- Anti-malware software is tested for compatibility with the operational environment.
- Anti-malware software is configured in prevent mode if possible, after assessing for operational impact. It is recommended to configure the anti-malware software to quarantine suspicious files and to raise an alarm to the user's security department instead of immediately deleting them. This allows the user's security department to investigate the alert and possibly prevent future 'false positives' while allowing the recovery of files if it is confirmed that they are legitimate.
- Files to be sent should be scanned at least once at any stage/step of their internal processing and, ideally, as close as possible to their transfer into the Swift network. This is to make sure that such files do not contain viruses or malware that may create risks for the sender, for Swift, or for the receiver.
- Endpoint Protection Platform (EPP) solution, combined or not with Endpoint Detection and Response (EDR) offering similar control on the infrastructure can be considered as a valid implementation.
- The resulting alerts are analysed for appropriate action and remediation in line with or as part of control the 6.4.

Optional Enhancements:

- Anti-malware systems use a combination of signature-based and heuristic-based capabilities.
- Anti-malware or other EPP solutions are implemented, when technically possible, on all General-purpose operator PCs.
- Anti-malware solutions are, when technically possible, implemented on non-Windows systems.
- 'On-demand full scanning' on servers is scheduled to be performed at least on a weekly basis.

6.2 Software Integrity

Control Type: Mandatory / Advisory for A4	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	

Control Definition

Control Objective: Ensure the software integrity of the Swift-related components and act upon results.

In-scope components:

- Swift connector
- GUI to the messaging and communication interface
- messaging interface
- communication interface
- RMA
- SNL
- New HSM (only for the checking of the firmware update)
- [Advisory: Customer connector]

Risk Drivers:

- unauthorised system changes
- HSM management misused

Implementation Guidance

Control Statement:

A software integrity check is performed at regular intervals on messaging interface, communication interface, and other Swift-related components and results are considered for appropriate resolving actions. Origin and integrity of the software is ensured at download and at deployment time.

Control Context:

Software integrity checks provide a detective control against unexpected modification to operational software.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- Software integrity checks are conducted on in-scope components upon start-up, and additionally at least once per day.

  Options for implementation:
  - Integrated in the product
  - Third-party file integrity monitoring (FIM) tool
- Before applying downloaded software, operators should validate the legitimate source/site and perform, when technically possible, integrity checks such as checksum validation to support the change and release management process up to the deployment of the software.
- The resulting alerts are analysed for appropriate action and remediation in line with or as part of the control 6.4.

Optional Enhancements:

- An integrity check is performed in memory.
- An integrity check is performed at the operating system level.
- File Integrity Monitoring covers the products with integrated mechanisms.

6.3 Database Integrity

Control Type: Mandatory / Advisory for A4	Applies to architecture:	A1	A2	A3	A4	B
		•	•		•	

Control Definition

Control Objective: Ensure the integrity of the database records for the Swift messaging interface or the customer connector and act upon results.

In-scope components:

- databases for messaging interface products, including a related hosted database
- databases for customer connector, including a related hosted database

Note: this requirement is not applicable for Architecture A1 if the infrastructure does not include a messaging interface and for Architecture A4 if there is no database linked to the customer connector.

Risk Drivers:

- loss of sensitive data integrity

Implementation Guidance

Control Statement:

A database integrity check is performed at regular intervals on databases that record Swift transactions and results are considered for appropriate resolving actions.

Control Context:

Database integrity checks allow unexpected modification to records stored within the database to be detected.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- Database integrity check functionality is enabled to make sure integrity at record level (checksum or signature of the records) and confirm that there are no gaps in sequential transaction numbering.

  Options for implementations:

  – Integrated into the messaging interface application

  – Integrated into the database product where the related hosted database, including its supporting server, is protected similarly to a Swift-related component (see the CSP FAQ – Swift Knowledge Centre article 5021823- for the relevant controls to consider).

- The resulting alerts are analysed for appropriate action and remediation in line with or as part of the control 6.4.

Optional Enhancements:

- A full database integrity check is performed at regularly timed intervals, ideally every two weeks.
- The integrity check performs a full referential check on all records (for example, no orphan records between tables) and searches for any unexpectedly deleted records.
- A dedicated database instance is used for Swift purposes.

6.4 Logging and Monitoring

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Record security events, detect and respond to anomalous actions and operations within the user's Swift environment.

In-scope components:

- Data exchange layer: network
- Operating system of a dedicated and general-purpose operator PC
- jump server
- Swift-related components (including interfaces, GUI, Swift and customer connectors)
- systems or virtual machines hosting Swift-related components
- network devices protecting the secure zone and HSM
- database linked to a messaging interface or a customer connector
- authentication or authorisation servers, or both, controlling accesses to the secure zone
- on-premises or remote (that is hosted or operated by a third party, or both) virtualisation or cloud platform hosting Swift-related VMs
- [Advisory: bridging servers (such as middleware or file transfer servers other than customer connectors) used for and guardian of the secure data exchange between back-office and Swift-related components]

Risk Drivers:

- lack of traceability
- undetected anomalies or suspicious activity

Implementation Guidance

Control Statement:

Capabilities to detect anomalous activity are implemented, and a process or tool is in place to keep and review logs.

Control Context:

Developing a logging and monitoring plan is the basis for effectively detecting abnormal behaviour and potential attacks and support further investigations. As the operational environment becomes more complex, so will the logging and monitoring capability needed to perform adequate detection. Simplifying the operational environment will enable simpler logging and monitoring.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- Overall goals for logging and monitoring:
  - Implement a plan to log security-relevant activities and configure alarms for suspicious security events (when supported by the application).
  - Implement a plan to monitor security events in logs and to monitor other data (for example, real-time business activities through the GUI), and establish a plan to treat reported alarms.

  - Support investigations and forensics in case of potential breach through log retention, in line with applicable laws and regulations.
- Logging:
  - Logging capabilities are implemented to detect and support analysis of abnormal usage within the secure zone and any attempts to undermine the effectiveness of controls within the secure zone.
  - Logs provide traceability of account usage to the appropriate individual.
  - Minimum logs to be recorded include:
    - Command-line history for privileged operating system accounts on servers
    - Messaging and communication interface application and operating system logs which detail abnormal system behaviour (for example, activity outside normal business hours, multiple failed login attempts, authentication errors, changes to user groups)
    - Firewall logs
    - Database logs (if available, and as a minimum in the case of hosted database solutions).
- Monitoring:
  - Procedures are in place to identify suspicious login activities into any privileged operating system or application accounts within the secure zone.
  - Monitoring processes are in place to review server, application, and database monitoring data of the secure zone either daily through human review or through automated monitoring with alerting.
  - Monitoring processes are in place to review network-monitoring data on a regular basis.
  - Unusual or suspicious activity is reported for further investigation to the appropriate security team.
- Log retention:
  - All logging and monitoring activities comply with applicable laws, regulations, and employment contracts which supersede other implementation guidance.
  - Messaging and communication interface application audit logs are retained for no less than 12 months and are sufficiently protected from an enterprise administrator-level compromise (for example, log files are transferred to a separate system with different system administrator credentials).
  - Operator PC, firewall, and database audit logs are retained for no less than 31 days (it is recommended to extend firewall and database audit logs retention to three months and possibly 12 months to support longer investigations).
  - Audit logs captured on other identified in-scope components, at application level or system level, are retained for no less than 12 months.
  - Prevent audit log loss by considering a range of configurable choices when log storage is to be exhausted. As examples, such choices can include log rotation, degraded mode or ignoring some events.

Optional Enhancements:

- A centralised logging capability is implemented, minimising the number of log locations to be inspected.
- Session recording is implemented to record all activity conducted by privileged accounts on Swift secure zone servers.
- Apply the control to all bridging servers (such as middleware or file transfer servers other than customer connectors) used for data exchange between back-office and Swift-related components.

6.5A Intrusion Detection

Control Type: Advisory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	

Control Definition

Control Objective: Detect and contain anomalous network activity into the on-premises or remote Swift environment.

In-scope components:

- network (data exchange layer reaching the Swift-related components)
- remote (that is hosted or operated by a third party, or both) virtualisation or cloud platform supporting the user Swift environment

Risk Drivers:

- undetected anomalies or suspicious activity

Implementation Guidance

Control Statement:

Intrusion detection is implemented to detect unauthorised network access and anomalous activity.

Control Context:

Intrusion detection systems are most commonly implemented on a network (NIDS)[39] – establishing a baseline for normal operations and sending notifications when abnormal activity on the network is detected. As an operational network becomes more complex (for example, systems communicating to many destinations, internet access), so will the intrusion detection capability needed to perform adequate detection. Therefore, simplifying network behaviour is a helpful enabler for simpler and more effective intrusion detection solutions.

Host intrusion detection systems (HIDS) are intended to protect the individual system on which they are implemented and to detect network packets on its network interfaces, similar to the way an NIDS operates.

Intrusion detection systems (NIDS or HIDS) often combine signature- and anomaly-based detection methods. Some systems can respond to any detected intrusion (for example, terminating the connection).

Endpoint and extended detection and response (EDR and XDR) are part of an emerging technology that addresses the need for continuous monitoring and response to advanced threats by detecting suspicious activities and (traces of) other problems on systems, and on endpoints. This technology is more frequently combined with endpoint protection platform (EPP) that operates at the device level while others expand their monitoring capabilities through XDR.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- The intrusion detection system is configured to detect anomalous activity and at the boundary of the secure zone and ideally within the secure zone. This can be achieved through NIDS, HIDS, EDR, XDR, or a combination of those techniques depending on the network configuration. For example, large VLAN would better benefit from NIDS; isolated island separating systems may benefit from HIDS. The EDR or XDR solution can also be considered, in particular within secure zones.
- Network activity to be tracked for intrusion detection analysis may include:
  - Inbound and outbound connections during non-business hours

[39] Network Intrusion and Detection System

  - Unexpected connections from/to the secure zone towards/from systems outside of the perimeter of the Swift or customer secure zone and ideally between systems within the secure zone
  - Unexpected port or protocol use (for example, P2P)
- The system has a repeatable process to regularly update known intrusion signatures or its abnormalities detection methods.
- If an intrusion is detected, then an alarm is raised and, if the tool permits, a defence mechanism can be triggered manually or automatically.
- Detected intrusions are managed through the standard incident response process.

Optional Enhancement:

- Intrusion detection systems can inspect encrypted flows.
- Abnormal network activity detection and containment is performed within the secure zone

Considerations for alternative implementations:

Institutions with a high level of security information and event management (SIEM) maturity within their organisation may consider extending, as stated in the control 6.4, their SIEM for real-time analysis of network and systems intrusion.

7 Plan for Incident Response and Information Sharing

7.1 Cyber Incident Response Planning

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Ensure a consistent and effective approach for the management of cyber incidents.

In-scope components:

- organisational control

Risk Drivers:

- excess harm from deficient cyber readiness

Implementation Guidance

Control Statement:

The user has a defined and tested cyber incident response plan.

Control Context:

Availability and adequate resilience are of key importance to the business. In this respect, defining and testing a cyber incident response plan is a highly effective way of reducing the impact and duration of a real cyber incident. As lessons are learnt either by testing this plan, or through real incidents, it is essential to apply these learnings and improve the plan. Planning for the sharing of threat and incident information is also critical in helping the broader financial community to implement effective protection against cyber-attacks.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- The user has developed and annually updates a cyber incident response plan. A formal back-up and recovery plan exists for all critical business lines to support incident response activities.
  - The cyber incident response plan includes up-to-date contact details (internal and external when using third parties or service providers) and escalation timers. Such a plan is based, as a guide, on:
    - The Cyber Security Incident – Recovery roadmap that provides a non-exhaustive list of steps or actions that a customer must follow in case of a cyber-security breach including the need to revert to Swift Support. Details are outlined in the Swift-ISAC Bulletin #10047.
    - Internal security policies, laws, and regulations within a user's jurisdiction must be adhered to and considered when planning a cyber incident response.
- As a minimum, the plan is reviewed on an annual basis, and tested at least every two years to make sure safe recovery of critical business operations with minimised outage time after a cyber-security incident.
- The cyber incident response plan includes steps to:
  - Promptly notify the appropriate internal stakeholders and leadership.
  - Promptly notify the relevant external organisational stakeholders (typically, regulator(s), supervisor(s), law enforcement authorities).

  - Promptly notify the Swift Customer Support Centre through the default channel and to comply with other obligations applicable to users in case of a security incident including the obligation to cooperate and provide forensic support as may be required by Swift.
  - Promptly contain or isolate the impacted system to limit the exposure of the attack while still being able to identify rogue activities.
  - Involve skilled cyber-security professionals to identify and address the cyber incident. It is the user's responsibility to take prompt corrective action to investigate, clean the full infrastructure, and resume secure operations as soon as possible.
  - Review the correctness of the user current attestation(s) and, as applicable under the Swift Security Controls Policy, invalidate such attestation(s) and submit new attestation(s).
  - Conduct post-incident problem analysis to identify and remediate vulnerabilities.
  - Fully document the incident.
- The user has a documented plan for the timely sharing of threat information to intelligence-sharing organisations, law enforcement, local regulators (as required in each user's jurisdiction) and to Swift. Sharing threat information may potentially support root cause analysis and sharing anonymous Indicators of Compromises (IOC) with the community.
- Information to be shared is first evaluated to make sure compliance with applicable laws and regulations (for example, privacy of personal data, confidentiality of investigations) and protects against the unintended sharing of sensitive data or data not relevant to the incident.
- The user can consume threat intelligence shared by Swift, for example in the form of IOCs.
- The user has procedures in place to:
  - Make sure the information is distributed to the correct contacts within the organisation,
  - Block traffic to/from IP-addresses/URLs mentioned in the IOCs.

Optional Enhancement:

- The user integrates the Swift ISAC automated feed solution in the environment.

7.2 Security Training and Awareness

Control Type: Mandatory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Ensure all staff are aware of and fulfil their security responsibilities by performing regular awareness activities, and maintain security knowledge of staff with privileged access.

In-scope components:

- All staff (such as employees, agents, consultants and contractors) with access to Swift-related systems (as user or for maintenance or administration)
- All staff (such as employees, agents, consultants and contractors) with privileged access to Swift-related systems (for maintenance or administration)

Risk Drivers:

- increased security risk from improperly trained staff

Implementation Guidance

Control Statement:

Annual security awareness sessions are conducted for all staff members with access to Swift-related systems. All staff with privileged access maintain knowledge through specific training or learning activities when relevant or appropriate (at management's discretion).

Control Context:

A security training and awareness programme encourages conscious and appropriate security behaviour of employees and administrators, and generally reinforces good security practice. In addition, it is particularly important that privileged access users have and maintain appropriate knowledge and expertise.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- All staff with access to Swift-related systems complete annual security awareness or training. Topics may include:
  - Swift-related products and services training (for example, through SWIFTSmart which is available to all users)
  - Cyber-security threat awareness within the financial services industry or relevant to the staff member's role and responsibilities
  - Risks related to internet usage or deployment in the cloud
  - Password security and management
  - Device security
  - Safe operating habits (for example, spam and phishing, including "spear[40]" phishing identification, downloading files, browsing practices)
  - Reporting of suspicious events and activities
  - Internal or external programme that optionally allows staff to obtain and maintain certification.

[40] Spear phishing is an e-mail or electronic communications scam targeted towards a specific individual, organisation or business.

- In addition, all staff with privileged access maintain their knowledge and expertise in line with their role and responsibilities by considering training or other learning activities that may include topics like:
  - Cyber risks awareness linked to their technologic or Swift-related environment (for example, through IOCs published by Swift) to develop best practice and processes
  - Administering and securing devices and other used systems
  - Detection and response to cyber incidents in line with the organisation's response plan
  - Internal or external programme that optionally allows staff to obtain and maintain certification.
- Training is delivered through the most appropriate channel, including computer-based training, classroom training, and webinars.

Optional Enhancement:

- Social engineering testing, including fake phishing e-mail campaigns, is performed to challenge and enhance security awareness.

7.3A Penetration Testing

Control Type: Advisory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Validate the operational security configuration and identify security gaps by performing penetration testing.

In-scope components:

- General-purpose operator PCs or, when used, jump servers used to access the secure zone
- Dedicated operator PCs
- Data exchange layer (the entry points to the secure zone or flows established to the secure zone components should be considered)
- Swift-related components (including interfaces, GUI, HSM, Swift and customer connectors)
- systems or virtual machines hosting Swift-related components
- network devices protecting the secure zone
- remote (operated by a third party) virtualisation or cloud platform hosting Swift-related VMs and the related management PCs

Note: Tests are performed in line with the Swift Customer Testing Policy. As such, Swift-specific applications and Swift-central services such as SWIFTNet InterAct, FileAct, FIN, SWIFTNet Instant or WebAccess are not to be tested.

Risk Drivers:

- Unknown security vulnerabilities or security misconfigurations

Implementation Guidance

Control Statement:

Application, system and network penetration testing is conducted towards the secure zone and the operator PCs or, when used, the jump server.

Control Context:

Penetration testing is based on simulated attacks that use similar technologies to those deployed in real attacks. It is used to determine the pathways that attackers might use, and the depth to which the attackers may be able to access the targeted environment. Conducting these simulations is an effective tool for identifying weaknesses in the environment which may require correction, improvement, or additional controls.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- The organisation uses a risk-based approach to determine the preferred scope (for example, the secure zone, or a specific server including potential other services supporting the secure zone), method (for example by sharing or not the internal structure, design or implementation) and attack origin (for example, internal, from within or outside the secure zone, or external attack) for the test.
- Penetration testing is performed at least every 2 years, and ideally as well after significant changes to the environment (for example, introduction of new/different servers, new operating systems, underlying technology such as virtualisation or new network device technology, network design change).
- Penetration testing is carefully planned and performed to avoid potential availability or integrity impacts.

- Penetration testing is performed by expert staff independent from the team in charge of the Swift infrastructure (internal Red Team or external resources).
- Network devices and system penetration testing (for example, rule bases and configurations review) are performed in the service production environment or in a pre-production environment replicating the live environment.
- Sufficient safeguards are in place to minimise any operational impact from conducting the penetration test.
- The outcome of the penetration testing is documented (with restricted access) and used as an input for the security update process.

Note: The CSP FAQ (Swift Knowledge Centre article 5021823) provides additional details on the scoping and the testing scenarios to consider. It is not expected that all components, underlying systems and environments are fully tested at each penetration testing exercise. However, all relevant systems must be considered as part of the overall penetration testing programme. This programme can also rely on previous internal exercises covering part of the overall infrastructure subject to the penetration testing exercise (such as network, physical environment or authentication services).

Optional Enhancement:

- Penetration testing is performed on Swift-specific applications while adhering to the Swift Customer Testing Policy. This Swift-specific application penetration testing is performed in the testing environment to avoid potential availability or integrity impacts.

7.4A Scenario-based Risk Assessment

Control Type: Advisory	Applies to architecture:	A1	A2	A3	A4	B
		•	•	•	•	•

Control Definition

Control Objective: Evaluate the risk and readiness of the organisation based on plausible cyber-attack scenarios.

In-scope components:

- Organisational control (people, processes, and infrastructure) to be also met by a third party operating a remote virtualisation or cloud platform that hosts Swift-related VMs.

Risk Drivers:

- excess harm from deficient cyber readiness
- unidentified sensitivity to cyber exposure

Implementation Guidance

Control Statement:

Scenario-based risk assessments are conducted regularly to improve incident response preparedness and to increase the maturity of the organisation's security programme.

Control Context:

Scenario-based risk assessments include cyber wargames, test attacks on existing systems and processes targeting the user's Swift infrastructure. Scenario-based risk assessments also include technical and business driven exercises performed as part of an institution risk management or Information Security Risk Management (ISRM) process.

These assessments include the following threats: end-user impersonation, message tampering, message eavesdropping, third-party software weaknesses, compromising systems or Denial of Service (DoS) attacks affecting service availability. Results of the assessment and existing mitigations help identify areas of risks that may require future actions, risk mitigations or an update of the cyber incident response plan.

Identified resulting actions, mitigations, or updates must be reported and closed in line with their criticality as per the internal or organisation's Information Security Risk Management (ISRM) process.

Several ISRM frameworks exist and can be consulted[41] to define the user's proper ISRM and resources (such as CIS-Critical Security Controls). These frameworks can be used to start implementing a basic risk management process to be further enhanced to address user's specific risks.

Implementation Guidelines:

The implementation guidelines are common methods to apply the relevant control. The guidelines are a helpful way to begin an assessment but should never be considered as an audit checklist as each user's implementation may vary. Therefore, in cases where some implementation guidelines elements are not present or partially covered, mitigations as well as particular environment specificities must be considered to properly assess the overall compliance adherence level (as per the suggested guidelines or as per the alternatives).

- A scenario-based risk assessment and planning activity is conducted to:
  - identify possible methods for adversaries to gain unauthorised access to the user's Swift infrastructure based upon observed adversary techniques or plausible adversary techniques inferred from adversaries' motivations and capabilities
  - analyse the effectiveness of existing prevention and detection controls to mitigate anticipated adversary techniques to gain unauthorised access to the environment

[41] For example, on NIST, ENISA, COBRA or ISO sites or from a local or regulator's standard or controls set to the same level of rigour as industry guidance.

- analyse the probability and impact of significant and plausible attack vectors given existing controls
- analyse the effectiveness of existing response controls to limit impact of significant and plausible attack vectors given existing controls
- Identify the need for additional preventive or detective controls

  All those activities can be covered through the organisation's ISRM.

- Assessment and planning activity is conducted at least annually, and updated through ongoing risk management activities, when significant technology changes occur, or when threat intelligence indicates relevant changes in an applicable adversary's capabilities or motivations.
- Current threat intelligence and observed or likely attacks (vectors, techniques, actors,) are used as the basis for scenarios.
- Each asset class (end-user devices, servers, network devices) is assessed against threats on a regular basis and when changes are introduced or when new threats are identified.

Appendix A: **Risk Driver Summary Matrix**

The matrix below is a summary of the risk drivers in this document, mapping the security controls to the documented risks they are intended to help mitigate.

Swift Security Controls / Risk Drivers	1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4A	2.5A	2.6	2.7	2.8	2.9	2.10	2.11A	3.1	4.1	4.2	5.1	5.2	5.3A	5.4	6.1	6.2	6.3	6.4	6.5A	7.1	7.2	7.3A	7.4A
Authentication token theft																					X										
Business conducted with an unauthorised counterparty																X															
Compromise of enterprise authentication system	X				X																										
Compromise of trusted back-up data										X																					
Compromise of user credentials	X				X																										
Credential replay	X				X														X												
Deletion of logs and forensic evidence		X																													
Excess attack surface								X							X																
Excess harm from deficient cyber readiness																													X			X
Excess privilege or access		X																		X											
Execution of malicious code																								X								
Exploitation of insecure system configuration								X							X																

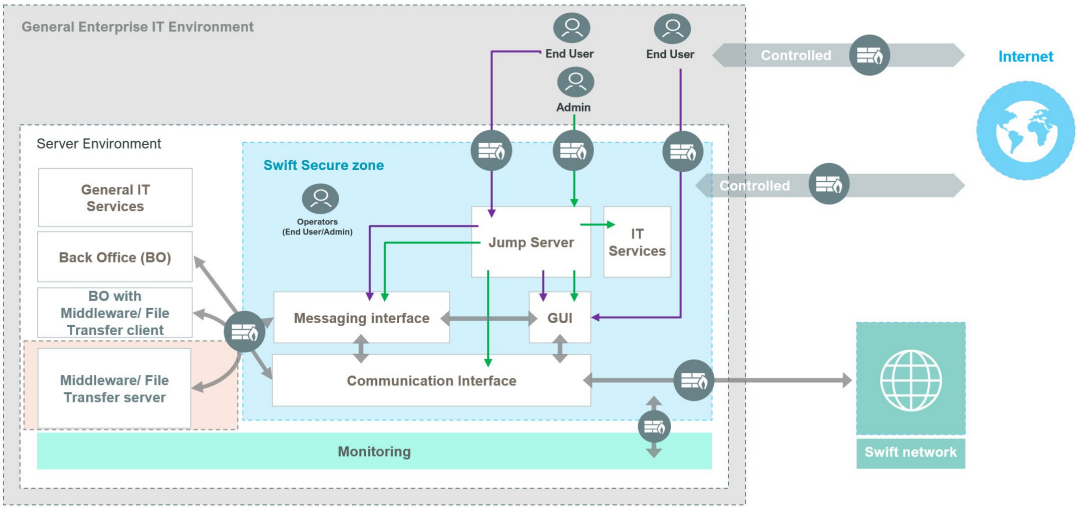
Swift Security Controls / Risk Drivers	1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4A	2.5A	2.6	2.7	2.8	2.9	2.10	2.11A	3.1	4.1	4.2	5.1	5.2	5.3A	5.4	6.1	6.2	6.3	6.4	6.5A	7.1	7.2	7.3A	7.4A
Exploitation of known security vulnerabilities							X					X												X							
Exposure to internet-based attacks	X			X	X																										
Exposure to sub-standard security practices													X																		
HSM management misused		X																		X	X				X						
Increased security risk from improperly trained staff																														X		
Lack of traceability		X															X				X						X				
Loss of operational confidentiality											X																				
Loss of operational integrity											X																				
Loss of sensitive data confidentiality						X			X	X																					
Loss of sensitive data integrity						X			X																	X					
Password cracking, guessing, or other computational compromise																		X	X												
Password theft						X					X								X				X								
Separation of duty violations																				X											
Unauthorised access	X		X		X	X														X											
Unauthorised physical access																	X														

Swift Security Controls / Risk Drivers	1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4A	2.5A	2.6	2.7	2.8	2.9	2.10	2.11A	3.1	4.1	4.2	5.1	5.2	5.3A	5.4	6.1	6.2	6.3	6.4	6.5A	7.1	7.2	7.3A	7.4A
Unauthorised system changes		X																							X							
Unauthenticated system traffic						X			X																							
Uncontrolled proliferation of systems and data			X																													
Undetected anomalies or suspicious activity														X													X	X				
Unidentified sensitivity to cyber exposure																																X
Unknown security vulnerabilities or security misconfigurations												X																			X	
Untrustworthy staff or system operators																						X										

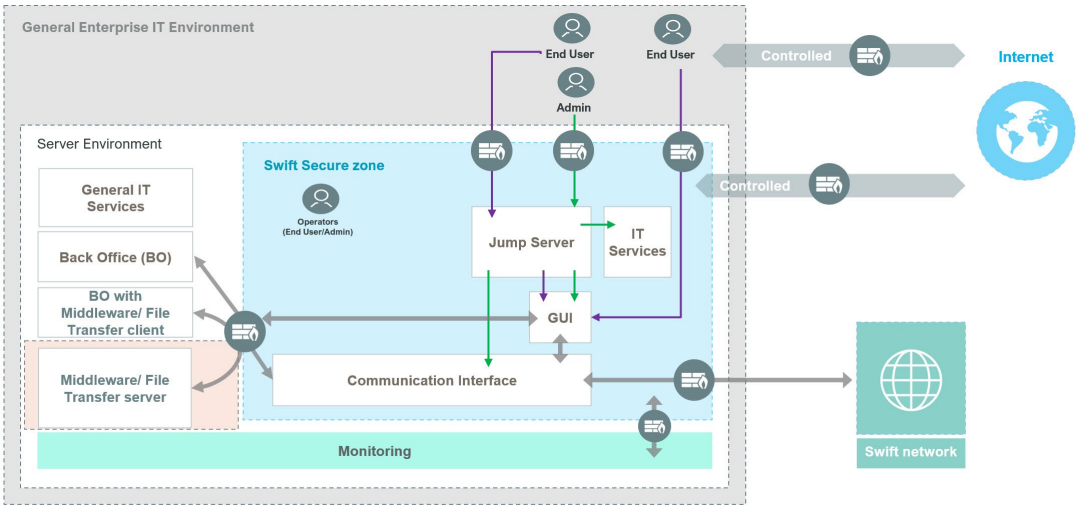
Appendix B: Secure Zone Reference Architectures

The following diagrams are for reference only, and describe one of many ways for the secure zones to be designed for each architecture (A1, A2, A3, A4, B); the CSP architecture decision tree can be used to identify the architecture type that matches best (taking also into account the Knowledge Centre article https://www2.swift.com/go/book/book201561 when an outsourcing agent is used).

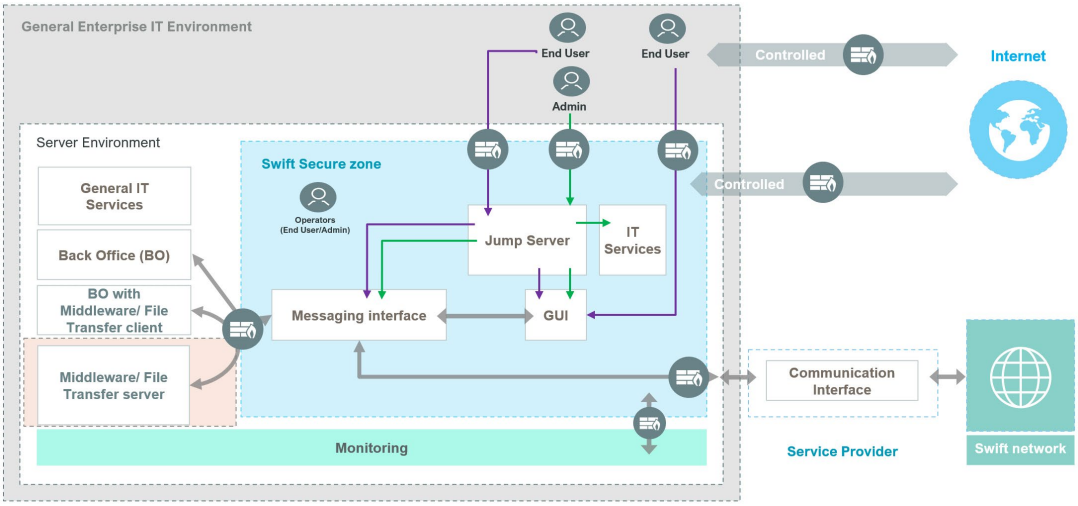
Drawing 11a: Secure Zones Example for Architecture A1 Interfaces within the user environment (on premises or in the Cloud)



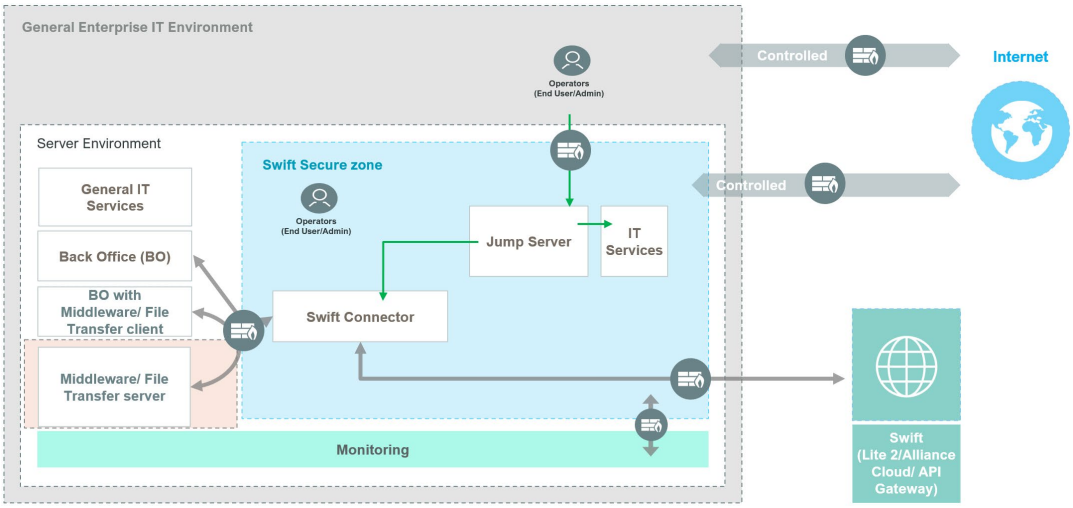
Drawing 11b: Secure Zones Example for Architecture A1 Communication interface only within the user environment (on premises or in the Cloud)



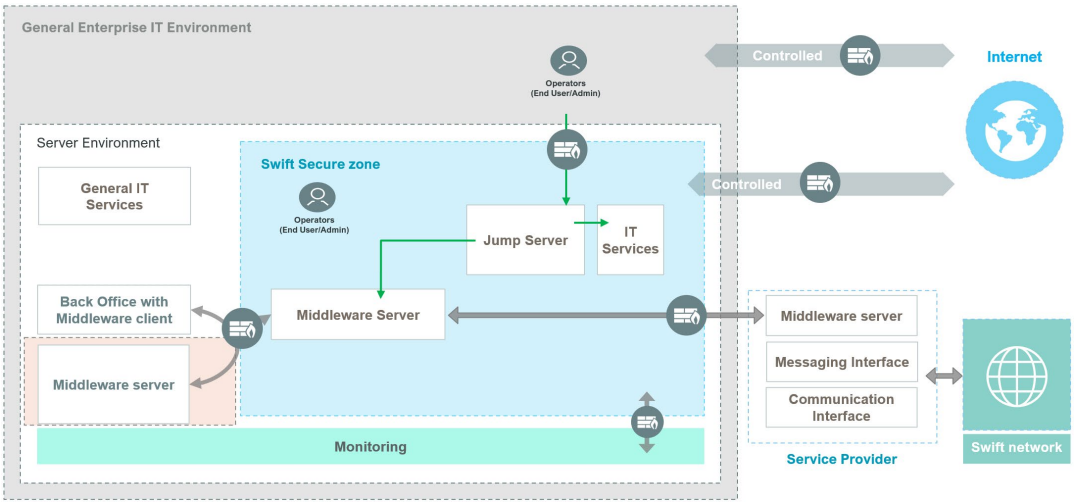
Drawing 12: Secure Zone Example for Architecture A2
Messaging interface only within the user environment (on premises or in the Cloud)



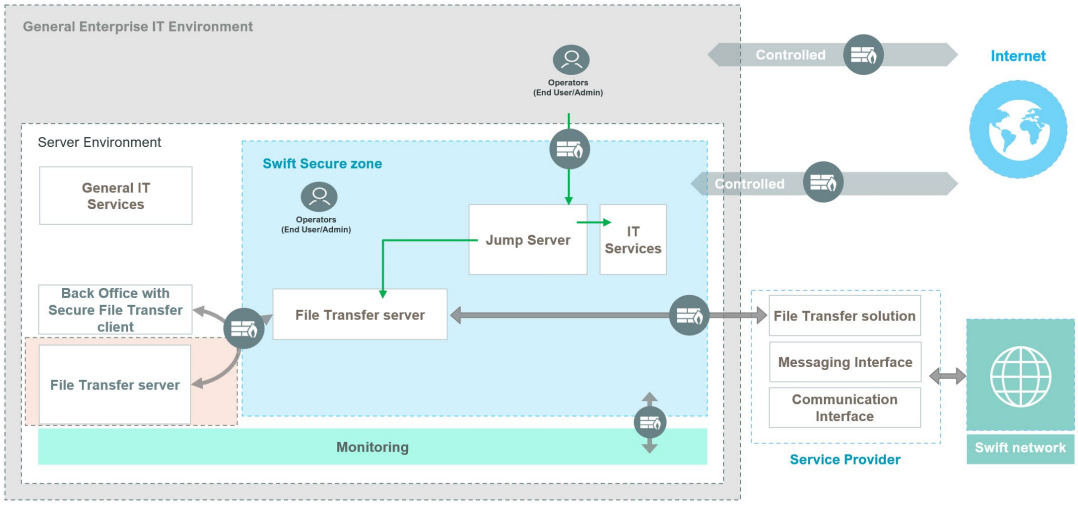
Drawing 13a: Secure Zone Example for Architecture A3
Swift connector



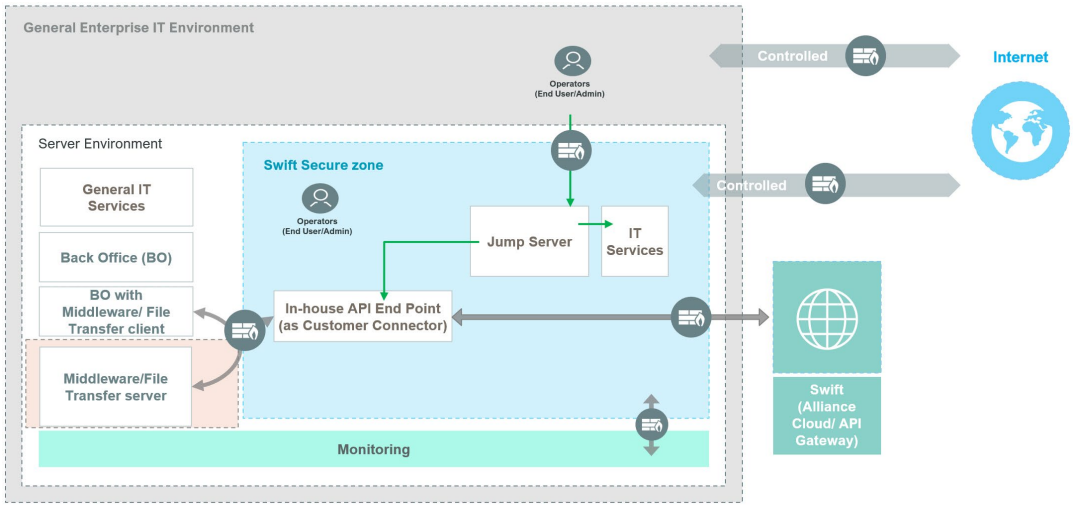
Drawing 13b: Secure Zone Example for Architecture A4
Middleware server as Customer Connector



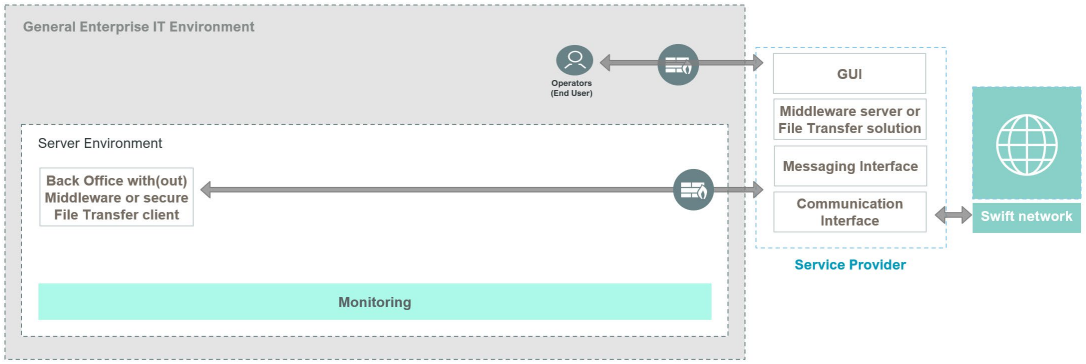
Drawing 13c: Secure Zone Example for Architecture A4 File Transfer server as Customer Connector



Drawing 13d: Secure Zone Example for Architecture A4 In-house customer API End Point as Customer Connector



Drawing 14: Architecture B No local footprint



Appendix C: **Sample Threat Scenarios**

The following scenarios are examples to help users to understand the types of cyber-threats that each security control is intended to help mitigate. These scenarios are non-exhaustive and provided for context and educational purposes only. The likelihood and impact of each scenario may differ significantly based on variables within the user environment.

1.1 Swift Environment Protection

- Attackers compromise credentials of the system administrator of the enterprise Active Directory, thereby granting the attackers access to all login credentials stored in the directory.
- Attackers compromise supporting IT infrastructure (for example, scanning server, patching server), located in the general IT environment. The objective is to steal system credentials and subsequently access the user's Swift infrastructure.
- Attackers gain administrative access to an operator's PC, allowing the attacker to compromise the local account database and reuse the stored hashes to access other systems.

1.2 Operating System Privileged Account Control

- A system administrator using the root account in Linux performs unauthorised actions (for example, changing security configurations, causing an intentional system crash), which are not traceable to an individual operator.
- An operator with excess administrative privileges deletes logs and other forensic evidence to hide unauthorised actions.

1.3 Virtualisation or Cloud Platform Protection

- An attacker with access to the hypervisor or the cloud platform could comprise the confidentiality, integrity and availability of virtual machines hosting Swift services.
- An attacker with access to the hypervisor or the cloud platform provisioning function could create new virtual machines to support the attack. For example, they can create fake application services to cause users to divulge sensitive information or download malware.
- An attacker benefits of vulnerabilities or insecure configuration of the hypervisor or the cloud platform to breach separation between hosted virtual machine domains to access virtual machines hosting Swift services.

1.4 Restriction of Internet Access

- Attackers compromise supporting IT infrastructure (for example, middleware server or virtualisation platform server), to steal system credentials and subsequently access the user's Swift infrastructure.
- Attackers gain administrative access to an operator's PC, allowing the attacker to compromise the local account database and reuse the stored hashes to access other systems.
- An operator clicks a malicious link in an e-mail or web page, unknowingly downloading malware which compromises the operator's PC or server.

1.5 Customer Environment Protection

- Attackers compromise credentials of the system administrator of the enterprise Active Directory, thereby granting the attackers access to all login credentials stored in the directory.
- Attackers compromise supporting IT infrastructure (for example, scanning server, patching server), located in the general IT environment. The objective is to steal system credentials and subsequently access the customer infrastructure.
- Attackers gain administrative access to an operator's PC, allowing the attacker to compromise the local account database and reuse the stored hashes to access other systems.

2.1 Internal Data Flow Security

- An attacker with network access to the secure zone compromises the integrity of the transactions in transit between the messaging interface and communication interface.
- An attacker with network access to the secure zone is able to monitor unencrypted traffic between user's Swift components and record confidential transactions.

2.2 Security Updates

- An attacker uses a known and unpatched vulnerability to gain access to a server hosting a Swift-related component.
- The operating system has aged beyond the vendor's support life-cycle window, resulting in persistent open vulnerabilities with no available remediation from the vendor.

2.3 System Hardening

- An attacker uses the default username and password to access the administration interface of a network firewall.
- An attacker uses a vulnerability associated with an unused network protocol (for example, telnet) to gain access to a Swift server.

2.4A Back Office Data Flow Security

- An attacker positioned on the used middleware server or between the back office and messaging interface injects unauthenticated transactions.
- An attacker creates a "person-in-the-middle" attack to change the beneficiary accounts of valid Swift transactions.
- An attacker positioned on the used middleware server or between the back office and messaging interface can monitor unencrypted traffic and record confidential transactions.

2.5A External Transmission Data Protection

- A data back-up location is compromised, and unencrypted Swift back-ups and credential hashes are accessed, providing the attacker with valuable information about Swift operators and typical activity within the user's environment.
- Unencrypted back-ups of Swift servers are transmitted over an insecure network connection, resulting in an attacker gaining read-access to all recent messaging traffic records.

2.6 Operator Session Confidentiality and Integrity

- An operator leaves his desk and no timed screen lock-out is implemented, allowing an unauthorised person access to the operator's account and the Swift messaging interface.
- An attacker is able to perform surveillance on an unencrypted operator session, and learns from unencrypted information to plan a future attack.
- An attacker is able to perform surveillance on an unencrypted operator session, and steals credentials to create a fraudulent Swift transaction.
- An attacker intercepts a transaction sent between the browser and the web application, modifies the transaction content, and forwards it to the web application.
- An attacker is able to hijack an open session or bypass an authentication scheme due to unsafe settings to capture or create fraudulent Swift transactions.

2.7 Vulnerability Scanning

- A discoverable vulnerability is left unidentified and untreated, allowing an attacker to exploit the vulnerability to gain access to a Swift-related system.

2.8 Outsourced Critical Activity Protection

- An outsourced provider does not properly separate Swift systems from other low-security systems, resulting in a virus spreading across environments and affecting the integrity of the Swift systems.
- An outsourced provider does not properly enforce access control, resulting in an unauthorised employee gaining access to the Swift messaging interface or other Swift-related components or systems.

2.9 Transaction Business Controls

- Daily reconciliation is not performed, resulting in a fraudulent transaction going unnoticed until after the settlement date.
- Transactions are not limited to normal business hours or amounts, resulting in an unnoticed fraudulent transaction.
- Transactions are not monitored or regularly reconciled, resulting in an unnoticed fraudulent transaction.

2.10 Application Hardening

- Default accounts or passwords can be used by attackers to gain unauthorised access to the application.
- Excessive privileges given to application users can be abused by attackers to perform unauthorised actions on the application.
- An attacker uses a vulnerability associated with an unused network protocol (for example, telnet) or functionality provided by unnecessary packages to gain access to a Swift server.

2.11A RMA Business Controls

- RMA relationships are not properly managed, resulting in the processing of a transaction from an unscreened or dormant counterparty.

3.1 Physical Security

- Poor log retention results in the inability to fully investigate which staff had physical access to the safe after a set of Swift HSM tokens were discovered to be missing.

- Weak data centre access control provides unauthorised staff with physical access to perform a physical-based attack on the Swift servers.

4.1 Password Policy

- A password policy is established, but not enforced, resulting in operators using weak passwords that are easily cracked during a cyber-attack.
- A password of insufficient length allows the computation of a weak password hash, which an attacker steals from the PC's memory and allows him to recompute the original password.
- The same passwords are used by an administrator for systems inside and outside the secure zone, resulting in an adversary compromising the more exposed password and re-using this knowledge to gain access to the secure zone.

4.2 Multi-factor Authentication

- Multi-factor authentication is not implemented to access applications, resulting in an adversary using a stolen password to gain full access to the Swift messaging interface.
- Multi-factor authentication is not implemented to access the operating system of the messaging interface, resulting in an adversary using a stolen password to gain full administrative access to the system.

5.1 Logical Access Control

- "Least privilege" controls are not enforced, allowing an operator who only requires read-only access the ability to create and send Swift transactions.
- Separation of duty controls are not enforced, allowing a single operator to create and approve a Swift transaction, conflicting with the user's transaction approval policy.
- Account access is not promptly revoked, resulting in a recently transferred employee using their residual access to modify records on the Swift messaging interface.

5.2 Token Management

- Poor record keeping during assignment of connected and disconnected hardware or personal tokens results in the inability to revoke the correct tokens after staff members leave the organisation, allowing unknown and uncontrolled access.
- A token is left inserted in an operator's PC when not in use, allowing an attacker to use the token as an authentication credential as part of an attack.

5.3A Staff Screening Process

- A new employee with a previous judicial record for financial fraud is not screened before being granted operator access, resulting in an untrustworthy individual being placed in a position of trust.
- Current employees are not periodically screened, resulting in the organisation not having knowledge of an employee who has taken a part-time job with another financial institution and now has a significant conflict-of-interest.

5.4 Password Repository Protection

- A Swift operator stores his passwords on a piece of paper at his work area, allowing any staff member with physical access to the area to view the recorded password.
- A Swift application administrator stores his administrative passwords in a plain text file on his PC, thus allowing any PC system administrator access to the passwords.

6.1 Malware Protection

- Anti-malware software is not installed on the operator PC, resulting in a common malware executable compromising the PC after clicking a phishing e-mail.
- Anti-malware software on the Swift servers is not regularly updated, resulting in an otherwise detectable malicious executable causing harm to the servers.

6.2 Software Integrity

- An advanced attacker modifies the executable of the messaging interface and is not detected because software integrity checking has not been implemented.
- A malicious version of a software update is installed because the checksum was not verified at time of download.

6.3 Database Integrity

- A lack of database integrity checking allows targeted malware to delete database records while performing unauthorised transactions.
- A lack of database integrity checking allows an attacker to modify database records to hide evidence.
- A lack of database integrity checking allows a gap in sequential record numbering to remain undetected.

6.4 Logging and Monitoring

- Poor system logging results in the inability to trace malicious privileged commands to a specific individual during a cyber incident investigation.
- Logs are collected, but not monitored, resulting in abnormal activity going undetected until significant financial harm has occurred.

6.5A Intrusion Detection

- A lack of intrusion detection capabilities results in unusual traffic outside normal business hours going undetected.
- A lack of intrusion detection capabilities results in unexpected protocol traffic for a given port going undetected.
- The intrusion detection system is not properly configured or monitored, resulting in discoverable intrusions remaining undetected because of the high number of false alarms.

7.1 Cyber Incident Response Plan

- An untested cyber incident response plan results in a poor and uncoordinated response to a serious cyber intrusion, resulting in significant and avoidable financial harm.
- The failure to notify Swift during a cyber incident results in incomplete sharing of information, leading to similar cyber incidents at other institutions that could have been avoided.
- The inability to act upon cyber-threat intelligence leads to cyber intrusions that could have been avoided.

7.2 Security Training and Awareness

- Swift operators are not trained on best security practices, resulting in staff clicking malicious phishing e-mail links.
- Swift application administrators are not trained on security awareness related to their role and, as a result, do not detect or report suspicious activity on the Swift systems.
- Swift security officers lack knowledge related to their role and, as a result, do not properly assign privileges for operators, allowing the bypass of the separation of duties principle.

7.3A Penetration Testing

- Penetration testing is not conducted in the Swift environment, and thus excessively permissive firewall rules are not discovered and corrected.
- Penetration testing is conducted by unqualified staff who are unable to simulate a typical financial industry attacker, which results in a false sense of security and low commitment to needed security improvements.

7.4A Scenario-based Risk Assessment

- Realistic risk scenarios are not tested within the organisation, resulting in an incorrect estimation of likelihood, impact, and overall cyber risk.
- Risk scenarios are tested without involvement of the business units and appropriate management, resulting in poor overall value of the activity and low commitment to needed security improvements.

Appendix D: **Glossary of Terms**

Term	Definition
Administrator	May refer to: Application Administrators – responsible for configuring, maintaining, and conducting privileged activities through an application interface. System Administrators – responsible for configuring, maintaining, and conducting other privileged activities through operating systems or other direct (non front-end) access.
Ancillary services	Complementary services, such as queries on previous transactions (for instance through the Basic Tracker), pre-validation, conversion or screening performed before submitting business transactions. They are usually considered as out of scope, However, sharing/reusing credentials and roles/entitlements used for business transactions on systems or components used for those ancillary services turns those systems or components in scope.
Application account	Logons designated for an application. They are not meant to be used by a human, through a graphical user interface or interactive access. Application accounts have a password that is stored, retrieved, and used automatically by the application. An application account is typically used for integration purposes (for example, calling of API) or to support STP (straight-through-processing).
Asset class	A category of computing asset (for example, databases, servers, applications).
Back-office system	A system responsible for business logic, transaction generation, and other activities occurring before transmission into the user's Swift infrastructure.
Bridging server	Server (such as middleware or file transfer server) used to support the data exchange between back-office first hop and Swift-related components in the user's Swift infrastructure.
Business Connect provider	A provider that is subject to the Enabler programme and is offering a Business Connect solution to facilitate users' connectivity to Swift messaging services.
Co-hosted components	Components deployed or running on a same physical system or virtual machine (created on a virtualisation or cloud platform). Note that containerised applications are considered as co-hosted on the same physical system or virtual machine that hosts the container itself.
Communication interface	The software providing a link between the SWIFTNet network and Messaging Interface software or a back-office system. Communication interfaces provide centralised, automated, and high-throughput integration with different in-house financial applications and service-specific interfaces. Communication Interfaces are provided by Swift (for example, Alliance Gateway or Alliance Gateway Instant). Communication interfaces holding a Swift-compatible label can also be provided by third-party vendors.

Term	Definition
Connector	A local software designed to facilitate communication with an external messaging or communication interface, both or to a service provider. When using a connector, interface components are usually offered by a service provider (for example, by a service bureau, hub infrastructure, or Swift). In some cases, a service provider, including a Business Connect provider or a third party such as an outsourcing agent, can also run a connector.
Customer connector	File transfer solutions or middleware servers (such as IBM® MQ, Apache Kafka or Solace servers/brokers) used for external communication or connection are considered customer connectors as opposed to Swift-compatible products (such as communication and messaging interfaces or connector) delivered by Swift or related third-party vendors that are considered as Swift connector. In the future, an application integrating all functionalities to directly and independently connect to the Swift API Gateway to process transactions will also be considered as a customer (in-house API) connector. The 'customer facing' webservers used by WebAccess service providers to support their Swift WebAccess service, are also considered as customer connectors.
Common Vulnerability Scoring System (CVSS)	An open industry standard for assessing the severity of software vulnerabilities by assigning severity scores to these vulnerabilities, allowing for prioritisation of responses and resources in line with the threat.
Cyber-security incident	Any malicious act or suspicious event that compromises, or was an attempt to compromise, a computing environment.
Data exchange layer	The transporting of data between the Swift-related components (in the user's Swift infrastructure or at a service provider) and a user back-office first hop, at application level, as seen from the Swift-related components.
Dedicated operator PC	An operator PC located in the secure zone and dedicated to interact with components of the secure zone.
Endpoint Detection and Response (EDR)	An emerging technology that addresses the need for continuous monitoring and a response to advanced threats by detecting suspicious activities and (traces of) other problems on systems/endpoints.
Endpoint Protection Platform (EPP)	An emerging solution to address attack prevention. More frequently combines with EDR.
End User	An individual requiring interactive access to the application (for example, for business transactions, monitoring, and access control). This includes security officers and application administrators responsible for configuring and maintaining the application.
Four-Eyes principle	A security principle whereby two individuals must approve an action before it can be taken. This principle is also known as two-person rule.
General (enterprise) IT environment	The infrastructure used to support the broad organisation. This includes general IT services and general-purpose operator PCs.

Term	Definition
General IT services	The supporting IT infrastructure, such as authentication services, asset management, databases, data storage, security services (for example, patching) and networking services (for example, DNS, NTP).
General-purpose operator PCs	A PC or device located in the general enterprise environment and used for daily business activities: - as an operator: to use, operate, or maintain the user's Swift infrastructure residing on premises or externally hosted) or - as a user: to use a remote Swift infrastructure or application operated by a service provider (such as a service bureau, a Business Connect or a Lite2 for Business Application provider, or a Group Hub), a third party (such as an outsourcing agent) or Swift or - a combination, depending on the architecture type.
Graphical user interface (GUI)	A software that produces the graphical interface for a messaging interface, a communication interface or a connector (for example, Alliance Web Platform, Swift Microgateway Front-End, and equivalent products).
Group Hub	A Swift user or a non-Swift user organisation connecting Swift users within its corporate group. A Group Hub is considered as a service provider for the related users and not a third party.
Hardware token	A USB token, smart card, or similar device.
Hardware Security Module (HSM)	Swift Hardware Security Modules store the users private keys necessary for the signing (encrypting) operations. A new HSM model (Luna SA7) will be available as from 2023 and will progressively replace the existing HSM model (Luna IS6). While such replacement will maintain the same level of security, the HSM will clearly be identified as in scope of all the relevant controls. As such: - The term "HSM", refers to both existing (Luna IS6) and new models (Luna SA7) –no change. - The term "New HSM" will be used as heads-up when newly identified as in scope component of several controls. This to facilitate the transition while it is expected they are already covered with the existing model. Consequently, those controls need to be formally taken into account when installing, using the new HSM model (Luna SA7 as per documentation) and further assessing compliance with those controls. - At the end of the HSM migration, the term "new" in "new HSM" will disappear and "HSM" will remain.
Hosted database	The terminology used when a (user) database server or infrastructure (such as Oracle or the like) is used as opposed to an "embedded database" incorporated in the messaging interface itself.
Indicators of compromise (IOC)	Artefacts that can be observed on a network or operating system that might indicate system compromise.
Interactive login / session	The session model that indicates an exchange of data (for example, when a user enters data or a command and the system returns data).

Term	Definition
IT services	A set of components in support of business processes inside the secure zone, such as a release and patching deployment platform, Active Directory.
Jump server	A server used to provide access to the user secure zone from the user's corporate network (for example, Citrix or Remote Desktop).
Local Authentication (LAU)	The mechanism that provides integrity and authentication of files exchanged between applications. Local Authentication requires that the sending and receiving entity use the same key to compute a Local Authentication file signature.
Messaging interface	A software supporting the use of MT, MX, or ISO 20022 message standards through Swift FIN, InterAct, FileAct, and SWIFTNet Instant messaging services. The software provides the means for users to connect business applications to Swift messaging services and is typically connected directly to the communication interface. Messaging interfaces are provided by Swift (for example, Alliance Access or Alliance Messaging Hub). Messaging interfaces holding a Swift-compatible label can also be provided by third-party vendors.
Middleware	A software that enables two separate programs to interact or to exchange data with each other (for example, IBM® MQ, BizTalk, ConnectDirect, Apache Kafka, Solace PubSub+).Usually composed of a Server and Clients running on the various interconnected systems (Client-Server model). In the case of peer-to-peer model without central server, connectivity can be considered as being direct between the systems (so not through middleware).
Middleware server	Local middleware systems implementations, such as IBM® MQ server (including MQ queues manager, MQ appliance or both), Apache Kafka or Solace server/broker, used for data exchange between the Swift-related components (in the user's Swift infrastructure or at a service provider) and a user back-office first hop as seen from the Swift-related components.
Multi-factor authentication	A method of user authentication where at least two different components are required to authenticate a user. The following authentication factors can be selected: • Knowledge factor (something the user knows), for example, a PIN or a password • Possession factor (something the user has), for example, an HSM token, a Digipass, mobile phone, 3-Skey Digital and its mobile version, or an RSA One Time Password device • Human factor (something the user is), for example, finger print or any other biometric
Network access control list (ACL)	The list of rules that are applied to port numbers or IP addresses for controlling traffic in and out. These lists are available on a network device.
Network devices	Components used to assist in the management, routing, and security of the network (for example, routers, switches, firewalls).

Term	Definition
Non-Swift footprint	A component deployed in user environment to link with Swift messaging services, Swift Transaction Manager, or a service provider, and that is not a messaging interface, a communication interface or a connector delivered by Swift or a related third-party vendor. Examples are File server solutions, middleware servers or customer (in-house API) connector.
Operating system (OS) accounts	User accounts on a server or PC that are used for direct access to the operating system.
Operator	The term that collectively refers to both individual types below: End users – individuals requiring interactive access to the application (for example, for business transactions, monitoring, and access control). This includes security officers and application administrators responsible for configuring and maintaining the application. Operating System Administrators – responsible for configuring, maintaining, and conducting other privileged activities on the operating systems hosting the user's Swift infrastructure.
Operator PC	A PC or device used for daily business duties: - as an operator: to use, operate, or maintain the user's Swift infrastructure residing on premises or externally hosted or - as a user: to use a remote Swift infrastructure or application operated by a service provider (such as a service bureau, a Business Connect or a Lite2 for Business Application provider, an outsourcing agent, a Group Hub), a third party (such as an outsourcing agent) or Swift, or - a combination, depending on the architecture type.
Outsourcing agent	An outsourcing agent is a third-party organisation to whom Swift users delegate one or more of the following tasks: hosting, installation, operation and maintenance of components involved in their Swift connectivity, all components being subject to the CSP program. Those components may cover interfaces, Swift or customer connectors or a Swift-related application as defined in the Customer Security Controls Framework (CSCF). See also Knowledge Centre (swift.com) for more details.
Personal Identification Number (PIN)	A secret number that acts like a password preventing others from gaining unauthorised access to or using a token, mobile device or card.
Privileged account	An account on an operating system or application that gives elevated access beyond that of a typical user. Includes administrator accounts on operating systems, and security officer or application owner accounts on applications.

Term	Definition
Reasonable comfort	A level of comfort that Management can obtain from internal or external subject matter experts (SME) when: - appropriate level of independence and objectivity of the SME is ensured, - fair validation by the SME of control design and implementation, confirms mitigation of risks as in the control objective, and - noted deviations do not materially impact the control's ability to mitigate the risk, or alternative controls compensate for such deviations. External assessments and certifications (such as against Systems and Organizations Controls (SOC) 2 or any industry standard identified in Appendix E like NIST or PCI-DSS) that cover CSCF controls, can give Management such reasonable comfort about the appropriateness of the controls as well as their operating effectiveness. The scope of this evaluation and the approach used for control evaluation in the context of such external assessments or certifications must be understood before relying on them either partially or fully.
Relationship Management Application (RMA)	A filter that enables the user to limit the correspondents from which messages can be received and the type of messages which can be received. The use of the Relationship Management Application mechanism is mandatory for the FIN service. It is available on an optional basis for SCORE FileAct and Generic FileAct.
Remote access	The access to a computer from outside the user's network. For example, from home or from another organisation's network.
Remote login	A login to a system initiated over a network connection rather than directly from the user's network.
Secure zone	A secure operational (sometimes also called production) zone on user premises separated from the general enterprise. The secure zone contains Swift-related systems (for example, messaging interface, communication interface, connectors), and, optionally, other protected systems.
Server Environment	A data centre or other secured physical location hosting servers.
Service bureau	A Swift user or non-user organisation that provides services to connect Swift users. The services offered by a service bureau typically include sharing, hosting, or operating Swift connectivity components, logins to the infrastructure, or managing sessions or security on behalf of Swift users. Service bureaux are subject to the Shared Infrastructure Programme.
Service provider	An organisation that usually provides services to Swift users regarding the day-to-day operation of the connection to Swift. The services offered typically include sharing, or operating Swift connectivity components, logins to the infrastructure, or managing sessions or security for Swift users. Such organisations include service providers (for example, service bureau, Business Connect or Lite2 Business Application provider subject to respectively the SIP, the Enabler and L2BA programme, or Group Hub).

Term	Definition
Simple Object Access Protocol (SOAP)	An XML-based messaging protocol for exchanging information among computers.
Single user or safe mode	The protected mode of operation that limits the privileges of the user.
Software token	An authentication token in logical (software) form.
Staff	All individual people who collectively work for the same organisation (such as employees, agents, consultants and contractors).
Swift connector	A connector provided by Swift (for example, Swift Integration Layer (SIL) Direct Link, Alliance Lite2 AutoClient or Microgateway). A connector holding a Swift-compatible label provided by a related third-party vendor.
Swift footprint	A messaging interface, a communication interface or a connector provided by Swift or holding a Swift-compatible label and provided by a third-party vendor.
Swift-related application	Swift-related components or a (web) application exposed by a service provider or an outsourcing agent to support transactions or payments related activities (creation, modification, approval, deletion of Swift transactions). A Treasury Management System (TMS) is an example of such exposed (web) application.
Swift-related component	A software, product or element, usually deployed in a user's infrastructure or secure zone, that supports Swift messaging and transactions services. Messaging and communication interfaces, SNL, Swift and customer connector, HSM, tokens, GUI as defined in the Scope of Security Controls section are examples of Swift-related components. Network devices protecting the secure zone can by extension also be considered as Swift-related components but they are specifically mentioned when expected to be in-scope of a control.
Swift-related system	A system (physical box or virtual machine) running a Swift-related component.
Thick client	A software program installed and executed on the local operator PC, rather than using a browser interface.
Third party	An entity independent of the Swift user or user's Swift connectivity provider. For example, an outsourced or external IT provider or cloud provider. By default, service bureaux, Business Connect and L2BA providers are considered service provider and not as third party, unless the Swift user specifically engages with them to host or to operate, or both, in full or in part the user's Swift infrastructure (still owned by the user).
Transaction Authentication Number (TAN)	A type of single-use password generally used with a standard ID and password. Initially presented in a list or table.

Term	Definition
Transaction business control solution	A transaction business controls solution is used to screen (for AML, sanctions or other compliance purpose) or detect potentially fraudulent payments. The solution is also used to review the identified suspicious transaction payments and decide whether they can be cancelled or released. Note: The general-purpose operator PCs used to either configure the solution, cancel or release payments or a combination of those actions, are part of the CSP scope.
(Swift) Transaction Manager	The platform deployed centrally by Swift to offer complete transaction management in line with the strategy endorsed by the Board in March 2020.
Transport Layer Security (TLS)	A cryptographic protocol that secures communications by offering confidentiality, integrity and protection against replay attacks measures.
(Swift) User	An organisation that Swift has admitted under the Corporate Rules as a duly authorised user of Swift services and products. The eligibility criteria to become a Swift user are set out in the Corporate Rules.
User application accounts	User accounts established at the applications layer to grant access and permissions to the application (that is, not operating system accounts).
User's Swift infrastructure	The collection of, on-premises or remote (that is hosted or operated, or both by a third party), Swift-specific components managed by of for the user. Those components include applications, network devices, tokens and other removable media, and supporting hardware. Also known as the Swift or customer secure zone.
Extended Detection and Response (XDR)	New technology approach to threats detection and response that extends beyond end points by collecting and correlating activity data across multiple security layers (endpoints and other servers, across networks, clouds, and applications).

Appendix E: Mapping to Industry Standards

The Swift Knowledge Centre article 5026074 maps the Swift security controls against different international security standard frameworks (such as NIST Cybersecurity Framework v1.1, ISO 27002 2022 or PCI DSS 4.0). The article is expected, going forward, to be enriched with latest versions of standards or with new Standards, such as the SOC2 TS 2017 or the Unified Compliance Framework (UCF) standards.

The mapping table referred to in the article provides further details on how the Swift security controls map to similar controls in those industry standards. If users are certified against any of these standards and under the condition that the Swift infrastructure is in the scope of this certification, then the table indicates how the controls from these standards map to the Swift security controls.

As to other standards, Swift suggests using the informative references provided by NIST in the document Framework Core of their Cybersecurity Framework v1.1 (Appendix A) provided in the referred mapping table.

Important Note:

Meeting requirements from these industry security standards does not automatically imply full compliance with the Swift security controls. Some aspects of the controls might not be covered in the standard. It remains the ultimate responsibility of the user to assess whether and to which extent the compliance with one of these industry standards is suitable to assess the compliance with the Swift security controls.

Appendix F: Services and Components in scope per architecture type

To help users to identify the most important elements, the table below (version 2021_1.0) presents the services and components expected to be in scope and the usually related architecture type. It also presents, for informational purposes only, usual elements that are not in scope.

Note: This table will continuously be updated through the year. It is recommended to always use the latest version that can be found in the Swift Knowledge Centre article 5024040 (CSP Components).

The following elements have to be considered when using the table:

- If multiple components are owned by a user and these components have different architecture types, then the user has to attest in KYC-SA against the most comprehensive architecture type using the decision tree, if needed.
- A component that is co-hosted with a component in scope is considered as in scope.
- All the components located in a secure zone have to be secured to the same level.
- <CTRL F> represents the most convenient way to locate a product.
- Components are listed alphabetically within each category.
- <effective date> refers to the date as of which a newly introduced component must be taken into account for the CSP Assessment. If empty, then it means it must already be taken into account.
- A, B – The Architecture type can be any “A” or “B”.
- N/a – Not applicable.
- Swift recommends protecting components that are out of scope as if they were in scope.

In Scope/Not in CSP Scope	Category	Component Name (alphabetical within each category)	Description/Remark	Likely CSP Architecture Type(s)	Effective date
	Interfaces and Related Applications	Communication Interface (for example, Alliance Gateway, Alliance Gateway Instant (AGI))	List of compatible communication interface: here Also includes API connectivity functionality based on AGI or Alliance Gateway	A1	
		Graphical user interface (GUI)	Products such as Alliance Web Platform (AWP), CREST Gateway, Swift gpi GUI. Swift does not ensure compatibility of GUI provided by Vendors	A	
		IPLA – Alliance Access Integration Platform	Built on top of Alliance Access	A1/A2	
		IPLA – Connector For Sanctions Screening	Connector For Sanctions Screening is an IPLA component on Alliance Access	A1/A2	
		IPLA – gpi Connector	gpi Connector on IPLA is a vertical solution running as a set of components inside of Alliance Access IPLA infrastructure (runs within Alliance Access)	A1/A2	
		IPLA – TARGET2 Connector	TARGET2 Connector is an IPLA component on Alliance Access	A1/A2	
		IPLA – TARGET2 for Securities (T2S) Connector	T2S Connector is an IPLA component on Alliance Access	A1/A2	
		Messaging Interface (for example, Alliance Access, Alliance Messaging Hub (AMH))	List of compatible messaging interface: here Some back-office systems can be considered as Messaging Interface (See KB article 5021823)	A1/A2	
		MQHA in relax/strict mode	Back-office system or application using the MQHA in Relax and Strict mode are considered as Messaging Interface	A1/A2	
		RAHA in relax/strict mode	Back-office system or application using the RAHA in Relax and Strict mode are considered as Messaging Interface	A1/A2	
		Relationship Management Application (RMA)	Can be stand-alone or integrated in the Alliance Access or AMH or vendor product –(see also here). The RMA Portal operated by Swift is out of scope as such. When the RMA Portal is used, the equipment (such as the operators PCs and tokens) used to connect to it for central management of the relationships is in scope.	A1/A2	
		SwAP Proxy (gpi Connector) on AMH	gpi Connector on AMH is an AMH service running within AMH infrastructure (runs within AMH)	A1/A2	
		Remote File Handler (RFH) – FileAct	This component is used to transfer files and is installed in the messaging interface. Sometimes also referred by its executable name: SWFA Handler	A1/A2	
		Alliance Access used to connect to Alliance Remote Gateway (ARG)	This is an Alliance Access solution hosted on customer premises and accessing the Alliance Cloud solution at Swift.	A2	
		SWIFTNet Link (SNL)	Can be either included in the Communication Interface or stand-alone.	A1	
		Swift Translator (embedded)	when Embedded (that is integrated in the Messaging interface); **Out of Scope when stand-alone**	A1/A2	
	Swift Connector	Alliance Lite2 AutoClient	This is the File base solution interfacing with Swift Lite servers	A3	
		Swift Integration Layer (SIL) – Alliance Cloud	This is the Swift New Lite solution that replaces the AutoClient solution based on Direct Link.	A3	
		Swift Integration Layer (SIL) – Connector For Sanctions Screening	Connector For Sanctions Screening is a stand-alone component (discontinued as of January 2022)	A3	

In Scope/Not in CSP Scope	Category	Component Name (alphabetical within each category)	Description/Remark	Likely CSP Architecture Type(s)	Effective date
		Swift Integration Layer (SIL) – gpi Connector (aka gpi Connector Stand-alone)	gpi Connector on SIL is a vertical solution running as a set of components inside of SIL which is a stand-alone software (For example, gSRP or gCASE)	A3	
		Swift API Connector	Includes products such as Direct Link/SIL or Swift Microgateway.	A3	
		Swift Microgateway	Provides API connectivity functionality.	A3	
	Customer Connector	Customer API connector	Customer in-house API connector including API connectivity functionality, based on Swift API Security SDK or their specifications	A4	
		sFTP or FTPs solutions (servers)	Secure File transfer solutions used to facilitate communication with Swift-related components offered by a service provider or an outsourcing agent.	A4	
		Middleware Server	Includes local middleware systems implementations, such as IBM® MQ, Apache Kafka or Solace server/broker, used for data exchange/connectivity with a service provider or an outsourcing agent.	A4	
		WebAccess servers at provider side	The 'customer facing' webservers hosted and operated by Swift WebAccess service providers. The Swift WebAccess service providers have to consider those webservers as **customer connectors**, protect them accordingly and include them in the scope of their KYC-SA attestation and in the supporting assessment. Such webservers can only be descoped from the CSP if i) the supported WebAccess services are read-only (that is, used for queries only) **and** ii) the disclosure of transactions to external bodies or to other not involved participant(s) of the service does not affect the confidentiality and consequently the reputation of the service provider. Any descoping must be confirmed by the independent assessor.	A	
	Hardware Components	Alliance Connect SRX VPN boxes or Alliance Connect Virtual VPN instances (hosting systems or machines)	Only the CSP control 3.1 (Physical security) applies.	A, B	
		Connected and disconnected hardware authentication or personal tokens	Connected and disconnected hardware authentication or personal tokens used for Swift operations or secure zone access and PIN Entry Device (PED) used for HSM operations. **Includes the 3SKey personal tokens when used for Swift services (such as FIN, InterAct, FileAct in direct or through Alliance Cloud, Lite2 and in the future a messaging service or the Transaction Manager exposed by Swift).**	A, B	
		Hardware Security Module (HSM)	Typically combined with SWIFTNet Link SNL.	A1	
		Network devices protecting the secure zone(s)	Includes firewalls and routers.	A	
		Virtualisation Platform (Hypervisor)	Underlying layer on premises or with cloud providers hosting Swift-related virtual machines (VMs).	A, B	
		Dedicated operator PC	An operator PC located in the secure zone and dedicated to interact with components of the secure zone.	A	

In Scope/Not in CSP Scope	Category	Component Name (alphabetical within each category)	Description/Remark	Likely CSP Architecture Type(s)	Effective date
	Operator PCs and Operators	General-purpose operator PC accessing the on-premises or remote Swift infrastructure and the operators	An operator PC located in the general enterprise environment and used for daily business activities.	A, B	
		(Dedicated or) General-purpose operator PC used by CREST GUI users	An operator PC that uses the CREST Network Service Layer (NSL) stack. It is also known as a CREST GUI and used to interact with the CREST Gateway which is an Alliance Access based software.	A	
		General-purpose operator PC used to access Swift Messaging Services hosted and operated at a service provider	General-purpose operator PC used to access Swift Messaging Services hosted and operated onsite at a service provider (such as a service bureau, a Business Connect or an L2BA provider, a Group Hub), a third party (such as an outsourcing agent) or Swift and when those PCs are used to submit or affect business transactions.	B	
		General-purpose operator PC used by Alliance Cloud or Lite 2 GUI Users	GUI users only do not have a connector.	B	
		General-purpose operator PC used by Business Connect or L2BA GUI Users	This covers PC that remotely connect to a front-end application (or solution) operated by a Business Connect or an L2BA provider	B	
		General-purpose operator PC used by ESMIG user-to-application users	These are PCs connecting to the European Single Market Infrastructure Gateway (ESMIG) application over the Swift Network	B	
		General-purpose operator PC connecting to Swift Sanctions Screening Service cloud solution (SSS)	Sanction Screening Service uses central copy service. The solution is used to review the transactions in blocking or non-blocking mode. Where blocking mode is selected, the service is used to abort or release the transaction. The general-purpose operator PCs used to either configure the solution, abort or release payments or a combination of those actions, are part of the CSP scope.	A, B	
		General-purpose operator PC connecting to Swift Transaction Screening Service cloud solution (TSS)	Transactions Screening Service uses central copy service. The solution is used to review the transactions in blocking or non-blocking mode. Where blocking mode is selected, the service is used to abort or release the transaction. The general-purpose operator PCs used to either configure the solution, abort or release payments or a combination of those actions, are part of the CSP scope.	A, B	
		General-purpose operator PC connecting to Swift Payment Controls Service cloud solution (PCS)	Payment Controls Service uses central copy service. The solution is used to review payment instructions identified as potentially fraudulent or out-of-policy according to user-defined criteria. This can be done in blocking or non-blocking mode. Where blocking mode is selected, the service is used to abort or release the payment. The general-purpose operator PCs used to either configure the solution, abort or release payments or a combination of those actions, are part of the CSP scope.	A, B	
		General-purpose operator PC connecting to a transaction business control solution (on-premises, remote or in the cloud)	Other ancillary transaction business control solutions may be used to screen transactions for AML, fraud, sanctions or other compliance purposes. These solutions can also be used to review and make disposition decisions on whether the transaction, payment is aborted or released. The general-purpose operator PCs used to either configure the solution, abort or release payments or a combination of those actions, are part of the CSP scope.	A, B	
		General-purpose operator PC connecting to WebAccess services	Using Web Platform/Alliance Gateway/SNL - over MV-SIPN	A1	

In Scope/Not in CSP Scope	Category	Component Name (alphabetical within each category)	Description/Remark	Likely CSP Architecture Type(s)	Effective date
		"	Using Browser/Tokens - over MV-SIPN	B	
		"	Using Browser/Tokens - over the Internet	B	
		General-purpose operator PC accessing the gpi Tracker or the Central RMA Portal with management/update role(s)	Using Web Platform/Alliance Gateway/SNL - over MV-SIPN	A1	
		"	Using Browser/Tokens - over MV-SIPN	A, B	
		"	Using Browser/Tokens - over the Internet	B	
	MI products footprint used for Specific Swift Service	CRNet in Alliance Access	The CRNet component provides the user with a number of controls over the network connection from Alliance Access to the CRNet host application. It contains the underlying processes required for file transfer and interactive services.	A2	
		Euclid Connector Client (ECC) - for Swift traffic	Delivered by Swift to Euclid users and used for Swift traffic.	A1	
		Euclid Connector Host (ECH)	Delivered to EuroClear by Swift - only located at EuroClear premises.	Does not affect the architecture of the user	
		MI Channel for Continuous Link Settlement (CLS)	Market Infrastructure (MI) Channel is a messaging channel designed to enable customers to access large market infrastructures in an efficient manner. MI Channel relies on the SWIFTNet store-and-forward platform, and optimises the exchange of large amounts of data between the market infrastructure and their participants, while offering a simplified mode of operation and facilitating integration. MI Channel functionality is integrated within the existing communication Interface: SWIFTNet Link and Alliance Gateway.	A1	
		MI Channel for T2S	Software that manages the full communication stack for connecting to the T2S gateway in the Swift OPC specific to EuroClear.	A1	
		Minimum Foot print (MFP)	This solution is offered in two flavours: (i) embedded in SNL or (ii) as stand-alone, replacing the Alliance Access-Alliance Gateway-SWIFTNet Link, in both cases, they are in scope of the CSP.	A1	
		Transaction Delivery Agent (TDA)	The transaction delivery agent is an application, running on top of Alliance Gateway. It provides the transfer of messages between institutions. This transfer method offers a single guaranteed delivery of messages in sequence, without any duplication. The transaction delivery agent interface used to communicate with the applications of the institutions is based on the standard IBM WebSphere MQ messaging middleware.	A1	
	Others	Data Exchange Layer	The transport of data between the Swift-related components (in the on-premises user's Swift infrastructure or onsite at a service provider or at an outsourcing agent) and a user back-office first hop as seen from the Swift-related components. Applicable controls: 2.4A, 6.4, 6.5A, 7.3A.	A, B	
		Jump Server giving access to the secure zone(s)	A server used to provide access to the user secure zone from the user's corporate network (for example, Citrix or Remote Desktop).	A	

In Scope/Not in CSP Scope	Category	Component Name (alphabetical within each category)	Description/Remark	Likely CSP Architecture Type(s)	Effective date
		SOAP/API to connect from a back-office application to the Messaging Interface at a service provider	The SOAP connection method enables the exchange of MT, XML-based messages, and FileAct messages between Alliance Access and back-office applications.	B	
Not In scope		3Skey	A Swift personal identity solution based on PKI technology.3SKey tokens can be used with all banks to sign and approve transactions. 3SKey can be used on any electronic banking channel including in-house cash or treasury management systems, web banking, local and proprietary networks and Swift. You can use it to sign electronically banking instructions or connect securely to your banking application. **Note that 3Key personal tokens used for Swift services are in scope (refer to Hardware Components category above)**	N/a	
		Alliance WareHouse (AWH)	AWH is an application used to store long term messages and does not allow messaging operations. **Note that while not in scope of the CSCF as such when it is hosted outside a secure zone, the feeding of AWH (from the messaging interface) is subject to control 2.5A External Transmission Data Protection**.	N/a	
		Australia New Payments Platform (AU-NPP) and Go Local India (GLI) users	Not considered as Swift users.	N/a	
		The back office	The systems responsible for business logic, transaction generation, and other activities occurring before transmission into the user's Swift infrastructure. For example, back-office implementations such as SAP and General Ledger are out of scope.	N/a	
		Business Intelligence systems (for example, Swift Scope)	Although globally out of scope, **Swift recommends the Business Intelligence systems defined as destination for transmitted sensitive data to be included in the control '2.5A External Transmission Data Protection'.**	N/a	
		Connections to the Swift network supplied by Swift Network Partners	This includes the (i) Connection to the four Swift providers (BT Global Services, Orange Business Services, AT&T and Colt) behind the VPN Boxes, (ii) Internet connections and (iii) connections through Alliance Connect Virtual established by a Cloud Provider approved as per the Swift Cloud Provider Programme.	N/a	
		Euclid Client Connector (ECC) - Not for Swift traffic	Delivered by Swift. Swift provides the connector but BT Radianz provides the network connectivity.	N/a	
		Euclid PC	Delivered by EuroClear to EuroClear customers.	N/a	
		Euclid Server	Delivered by EuroClear.	N/a	
		sFTP or FTPs solutions (client) on back-office system	Secure File transfer solutions through a client only, used to facilitate communication with Swift-related components offered by a service provider or an outsourcing agent.	B	
		General Enterprise IT environment	The general IT infrastructure used to support the general organisation (for example, general-purpose PCs, mail server, directory services)	N/a	

In Scope/Not in CSP Scope	Category	Component Name (alphabetical within each category)	Description/Remark	Likely CSP Architecture Type(s)	Effective date
		General-purpose operator PC accessing the gpi Basic Tracker or the Central RMA Portal with viewer/read only role	When using swift.com accounts only for Basic Tracker functionalities. It is not used for Stop and Recall. Similarly, viewer role only on RMA Portal doesn't allow for any update. The RMA Portal operated by Swift is out of scope as such. When the RMA Portal is used, the equipment (such as the operators PCs and tokens) used to connect to it for central management of the relationships is in scope.	N/a	
		MQ, Kafka or Solace Client on back-office system	This is a software component that enables an application running on a system to issue calls to a queue manager (MQ Server) or a broker (Kafka/Solace) running on another system. The output from the call is sent back to the client, which passes it to the application.	N/a	
		MQHA in Basic mode	Back-office application using the MQHA in **Basic** mode are considered as back-office.	N/a	
		RAHA in Basic mode	Back-office application using the RAHA in **Basic** mode are considered as back-office.	N/a	
		Payment Gateway/Domestic Messaging Channel (PAG/DMC) for AU-NPP	Not considered as Swift users.	N/a	
		Swift Payment Control Service (PCS)	A solution used to identify and review potentially fraudulent or out-of-policy outgoing payment instructions in blocking or non-blocking mode (without capability to initiate or modify payments). **Swift recommends the tokens associated to this service to be covered in the control 5.2 Token Management.** **Note: The general-purpose operator PCs used to either configure the solution, abort or release payments or a combination of those actions, are part of the CSP scope.**	N/a	
		Pre-validation for Swift gpi	The gpi pre-validation detects payment problems before payments are sent for execution. There is consequently no specific risk in terms of CSP. The gpi pre-validation uses the Swift API Gateway. It must authenticate with the Swift API platform, which can be facilitated by using dedicated technology, such as the connector for Swift gpi. It can be based on SDK, SDK + gpi stand-alone connector, or by using interfaces and an embedded gpi connector.	N/a	
		Pre-validation gpi webserver	Used for queries only and does not impact the integrity of the transactions.	N/a	
		Swift Sanctions Screening Service (SSS) using central copy service - cloud solution	A solution used to screen transactions for sanctions compliance where blocked payments can be reviewed in Sanctions Screening and aborted or released. End users do not initiate or modify payments in Sanctions Screening. If a payment is released then it means that it has passed all transaction controls in the messaging interface (for example, Four-Eyes, Six-Eyes) and this is just an additional check for compliance purposes with those regulatory lists (UN, States, and other lists). End users use WebAccess to access the Sanctions Screening GUI in the cloud. **Swift recommends that the tokens associated to this service be covered in the control 5.2 Token Management.**	N/a	

In Scope/Not in CSP Scope	Category	Component Name (alphabetical within each category)	Description/Remark	Likely CSP Architecture Type(s)	Effective date
			Note: The general-purpose operator PCs used to either configure the solution, abort or release payments or a combination of those actions, are part of the CSP scope.		
		Swift Scope	A business intelligence solution providing full and immediate visibility on an organisation's daily cash reporting.	N/a	
		Swift SDK on back-office application (when relying on Swift footprint or customer connector)	Not in scope when relying on other Swift-related components to connect to Swift Messaging/Transaction Services (using a communication interface or a Swift (API) connector as Swift footprint and basically not integrating the Swift Security SDK or its specification).	N/a	
		Swift Translator (stand-alone)	Out of scope when stand-alone. **In-scope when embedded in the Messaging Interface.**	N/a	
		Swift Transaction Screening Service (TSS) using central copy service - cloud solution	A solution used to screen transactions for sanctions compliance where blocked payments can be reviewed in Transaction Screening and aborted or released. End users do not initiate or modify payments in Transaction Screening. If a payment is released then it means that it has passed all transaction controls in the messaging interface (for example, Four-Eyes, Six-Eyes) and this is just an additional check for compliance purposes with those regulatory lists (UN, States, and other lists). End users use WebAccess to access the Transaction Screening GUI in the cloud. **Swift recommends that the tokens associated to this service be covered in the control 5.2 Token Management.** **Note: The general-purpose operator PCs used to either configure the solution, abort or release payments or a combination of those actions, are part of the CSP scope.**	N/a	
		Other transaction business control solutions	Other ancillary transaction business control solutions may be used to screen transactions for AML, fraud, sanctions or other compliance purposes. These solutions can also be used to review and make disposition decisions on whether the transaction, payment is aborted or released (without capability to initiate or modify payments). **Note: The general-purpose operator PCs used to either configure the solution, abort or release payments or a combination of those actions, are part of the CSP scope.**	N/a	

Appendix G: Shared Responsibilities in an IaaS Cloud Model

Users engaging with third parties (such as an external IT provider, a cloud provider, or an outsourcing agent) or service providers (such as a service bureau, a Business Connect or a Lite2 Business Application provider that are, in this case, considered as third parties) in order to host or operate their own Swift infrastructure in full or in part, must get reasonable comfort from those third parties that the related activities are protected in line with CSCF security controls. As such, third parties can rely on their compliance programme that usually builds on SOC 2, PCI-DSS or NIST certifications or assurance to answer users engaging with them and to map the CSCF security controls. As an example, cloud providers approved as per the Swift Cloud Provider Programme, are, per the programme, providing a similar conformance statement to their subscribing users. The user remains responsible and accountable for the attestation they need to fill taking into account the deployed controls and those deployed by the involved third parties and service providers.

Not all outsourcing models can be covered here. Therefore, to illustrate and trigger users' choice when considering the outsourcing model, the table below presents the typical sharing of responsibilities when an Infrastructure as a Service (IaaS) model in the cloud is selected. In such scenario, the user subscribes to a virtualised environment set up by selected cloud providers (CP) on the cloud infrastructure. The user remains responsible for the deployment, management of the various stacks (systems and applications) in the subscribed environment and, therefore, of the related controls. The HSM has to be physically hosted (on-premises or in a co-location data centre). If the majority of the systems or components of an architecture A1 are hosted with the cloud provider, then the user still has on premises some equipment, as a minimum the operator PCs, that they need to protect.

It is also recommended to consult the outsourcing agent documentation that is available on the Knowledge Centre (swift.com).

Control	User	CP	Relevance for the Cloud Provider (CP)
1.1 Swift environment protection.	X	X	Segregated virtualised user environment [mainly through 1.1.c by design, network & operations]
1.2 OS privileged accounts control.	X	X	On the virtualisation infrastructure/environment set up by the CP
1.3 Virtualisation platform protection	X[if used]	X	Supporting the virtualisation infrastructure/environment set up by the CP.
1.4 Restriction of internet access.	X	X	Protection of the virtualisation infrastructure/environment set up by the CP
1.5 Customer environment protection.	X	X	Segregated virtualised user environment [mainly through 1.5.c by design, network & operations]
2.1 Internal data flow security	X		
2.2 Security updates.	X	X	On the virtualisation infrastructure/environment (and admin desktop)
2.3 System hardening	X	X	On the virtualisation infrastructure/environment (and admin desktop)
2.4A Back Office data flow security.	X	X	Secure exchange with the virtualisation infrastructure/environment and subscription of the user
2.5A External transmission data protection.	X	X	Virtualisation infrastructure/environment back-ups and transfers (between virtual stacks). Protect data storage (ideally through encryption – data at rest or environment/subscription)
2.6 Operator session confidentiality, integrity.	X	X	Limited to virtualisation infrastructure/environment and dedicated operator PCs
2.7 Vulnerability scanning	X	X	On the virtualisation infrastructure/environment
2.8 Outsourced critical activity protection	X		To be considered by user depending on outsourced model (HSM) – VPN is managed by Swift Potential access to data to be covered in contract (if possible through virtualised environment)
2.9 Transaction business controls	X		
2.10 Application hardening	X		
2.11A RMA business controls	X		

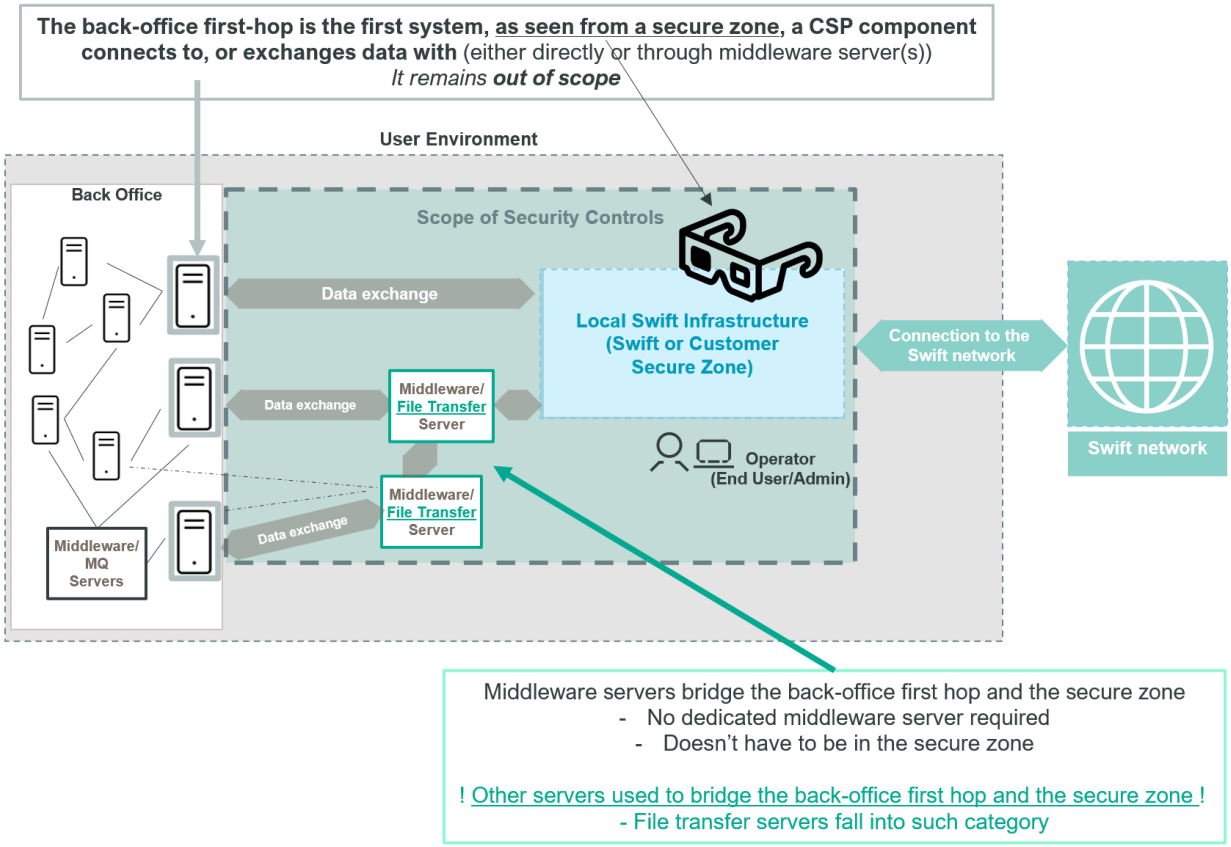
Virtualisation infrastructure/environment includes Network/FW devices

Control	User	CP	Relevance for the Cloud Provider (CP)
3.1 Physical security	X	X	Of the virtualisation infrastructure/environment.
4.1 Password policy	X	X	On the virtualisation infrastructure/environment and the subscription set up for the user
4.2 Multi-factor authentication.	X	X	Support secure access to the virtualisation infrastructure/environment set up for the user
5.1 Logical access control	X	X	On the virtualisation infrastructure/environment and the subscription set up for the user
5.2 Token management.	X	X	Solution dependent (HSM or others used by CP to access the virtualised infrastructure)
5.3A Personnel vetting process	X	X	For operators of the virtualisation infrastructure/environment and subscription set up for the user
5.4 Password repository protection	X	X	For the virtualisation infrastructure/environment, subscription and solution dependent (HSM or ?)
6.1 Malware protection.	X	X	Solution dependent on the virtualisation infrastructure/environment and operator PCs
6.2 Software integrity	X		
6.3 Database integrity	X		
6.4 Logging and monitoring	X	X	On the virtualisation infrastructure/environment and the subscription set up for the user
6.5A Intrusion detection	X	X	On the virtualisation infrastructure/environment and the subscription set up for the user (RACI)
7.1 Cyber incident response planning.	X	X	To be incorporated in customer incident response plan
7.2 Security training and awareness.	X	X	For operators of the virtualisation infrastructure/environment and subscription set up for the user
7.3A Penetration testing	X	X	On the virtualisation infrastructure/environment supporting the subscription set up for the user.
7.4A Scenario risk assessment	X	X	On the virtualisation infrastructure/environment supporting the subscription set up for the user.

Virtualisation infrastructure/environment includes Network/FW devices

Appendix H: Back-Office First Hop Flows Protection

The purpose of the drawings below is to visualise and support the journey to mandatory for the control 2.4 to secure the data exchange with the back-office first hop as seen from the secure zone.



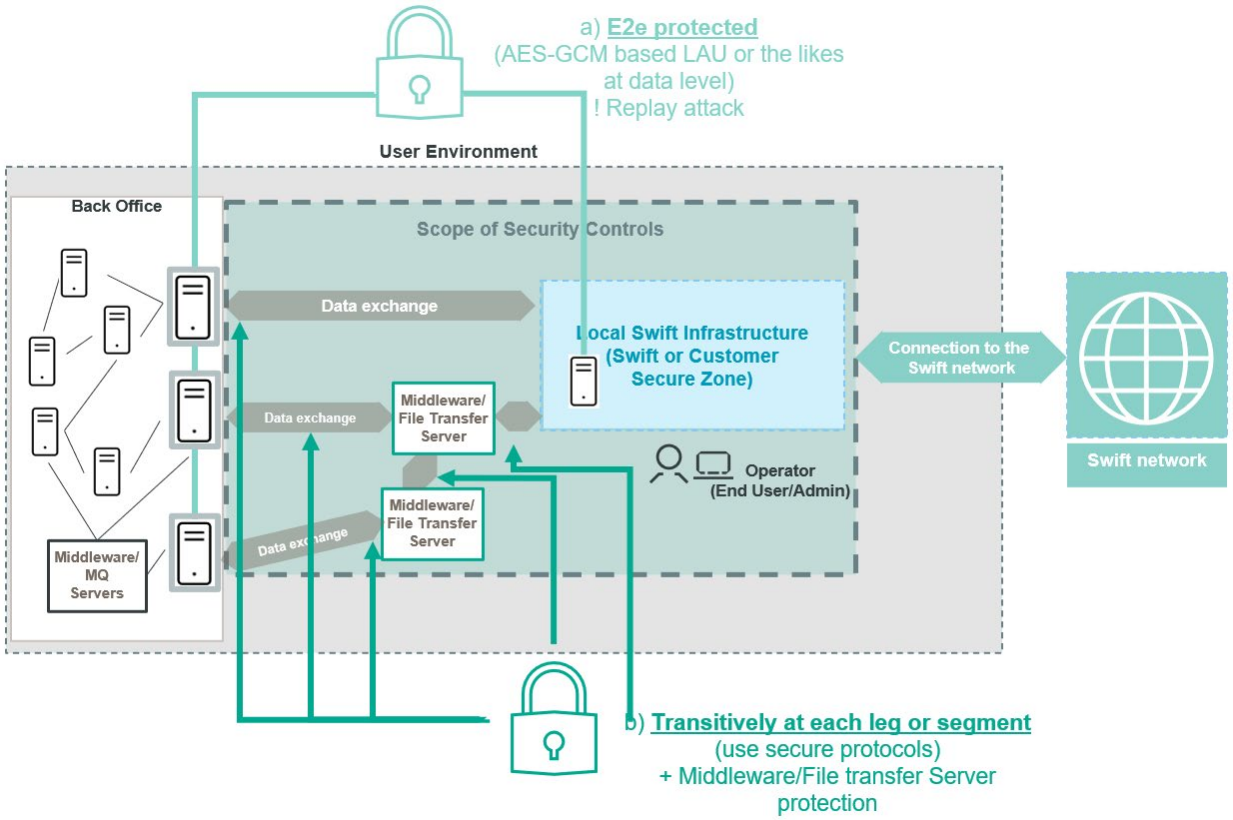
Drawing 15: The back-office first hops and servers bridging them with the secure zone

The drawing above depicts back-office first hops considered as first systems as seen from a secure zone component. As such, when not co-hosted with a Swift-related component and not in the secure zone, these back-office systems remain out of scope.

The data exchange between a secure zone component (first element) and the back-office first hop (second element) can be:

- directly performed between those two elements (point to point connection), or
- through a single server (such as a middleware/file transfer server) bridging those two elements, or
- through a series of servers (such as middleware/file transfer servers) bridging those two elements.

It is reminded that servers bridging a back-office first hop with the secure zone do not need to be in the secure zone, neither do they need to be dedicated to such bridging.



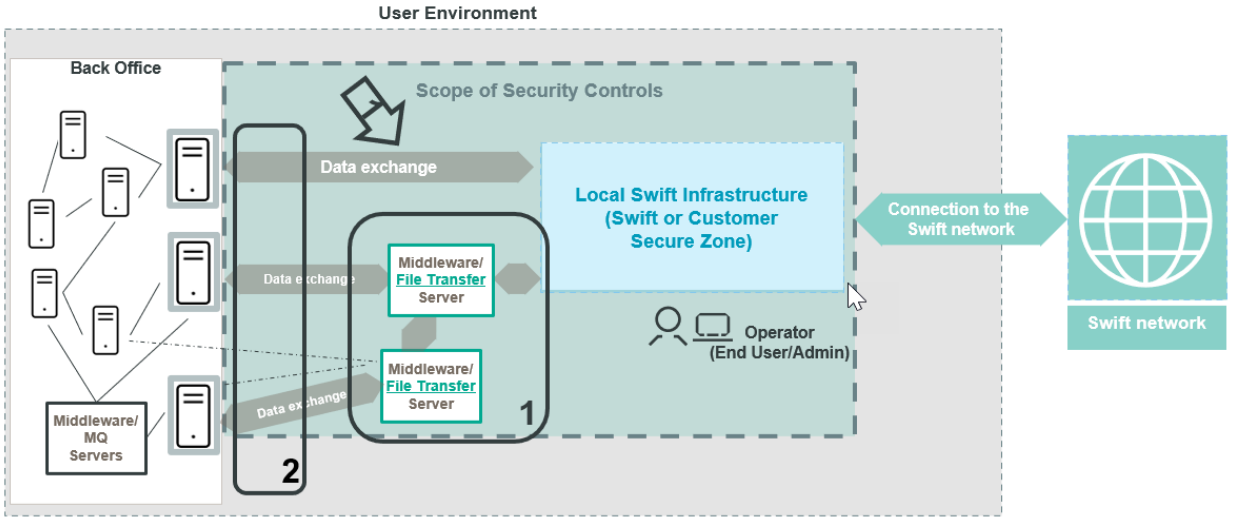
Drawing 16: Alternative ways of protecting the data exchange between the secure zone and the back-office first hops

The protection of the data exchange can be done

- end-to-end at data level, or
- through secure protocols on each segment, between the bridging servers, and by protecting the bridging servers themselves.

When an end-to-end protection exists at data level, combined with replay attack prevention, the protection of the in-between bridging servers is of less importance. Indeed, the data, individually authenticated and protected by the back office, that is validated by a component in the secure zone, cannot be modified during the data exchange.

Different methods can be used for the various data exchange or flows. Therefore, we strongly recommend supporting the journey by identifying those flows and by assessing their security posture. Then, appropriate actions should be taken to achieve the two phases depicted below.



Drawing 17: The two phases

Phase 1 (in a future CSCF version) - Protect the new point to point flows and the bridging servers that are guardian of the data exchange

- For the flows that rely, for the security of the exchanged data, on bridging servers (such as middleware or file transfer servers):
    - Consider those bridging servers as mandatory in-scope components of the various controls and not anymore as Advisory in-scope
    - Secure the data exchange between those bridging servers and the components in the secure zone
- When introducing new direct flows, between a new back-office first hop and a secure zone component, then protect, through security by design, the data exchange in line with control 2.4, such as by using i) end-to-end data protection or ii) a (native) secure protocol between the two elements.

Phase 2 (in a subsequent CSCF version) - Protect the legacy flows

- For the flows that rely, for the security of the exchanged data, on bridging servers (such as middleware or file transfer servers):
    - Secure the data exchange between those bridging servers and the back-office first hop (legacy flows)
- Secure the remaining direct data exchange between the secure zone components and the back-office first hops (legacy flows).

To prepare the phases, we recommend supporting the prioritisation of those various flows through a risk-based assessment.

Legal Notices